

Flask? No thanks!

by Alexey Popravka

About me

- 6+ years of Python
- Python developer / Tech Lead at Prom.ua
- Contributor of github.com/aio-libs
- github.com/popravich

What this talk is about?

- What is Flask
- What Flask is not (what is wrong)
- Alternatives
- Questions

What is Flask

“Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions.”

What is Flask

“Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions.”

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World!'

if __name__ == '__main__':
    app.run()
```

What is Flask

- *Jinja2* templates out of the box

```
from flask import render_template

@app.route('/hello/')
@app.route('/hello/<name>')
def hello(name=None):
    return render_template('hello.html', name=name)
```

```
<!doctype html>
<title>Hello from Flask</title>
{% if name %}
    <h1>Hello {{ name }}!</h1>
{% else %}
    <h1>Hello World!</h1>
{% endif %}
```

What is Flask

- *Jinja2* templates
- Sessions

```
from flask import Flask, session, escape

app = Flask(__name__)

@app.route('/')
def index():
    if 'username' in session:
        return 'Logged in as %s' % escape(session['username'])
    return 'You are not logged in'
```

What is Flask

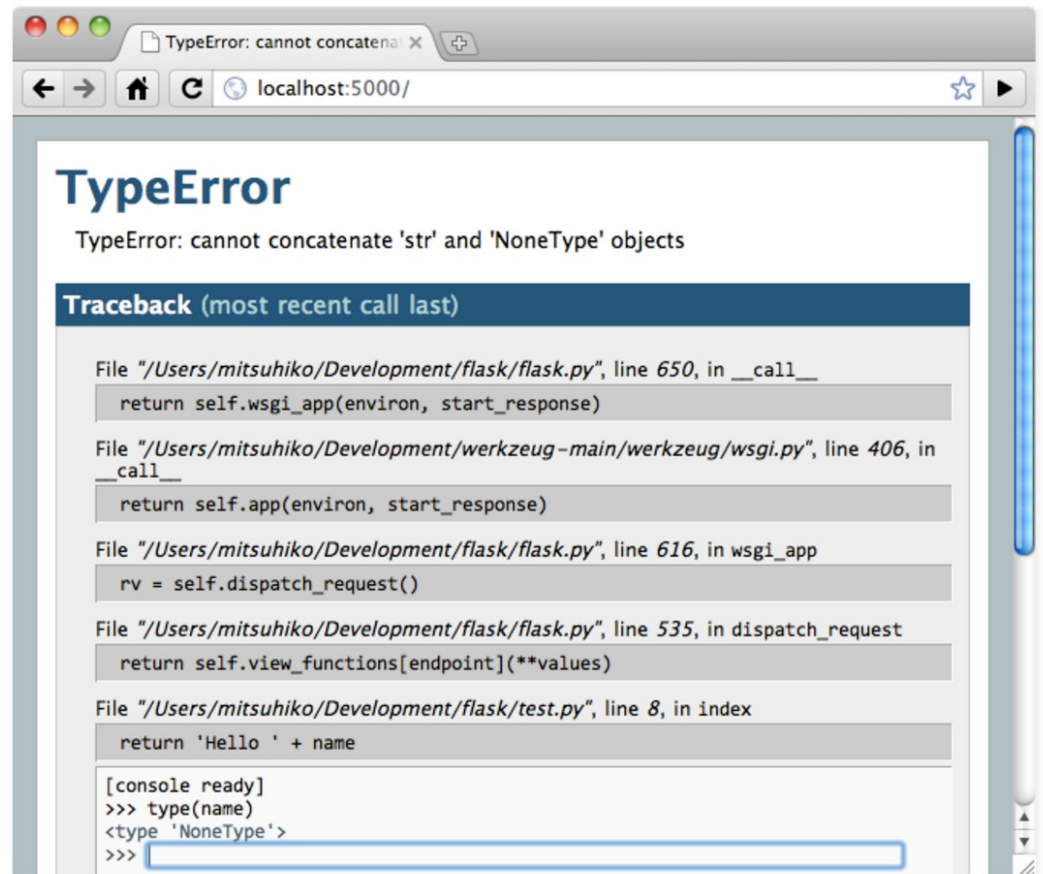
- *Jinja2* templates
- Sessions
- Static files serving

```
url_for('static', filename='style.css')
```


What is Flask

- *Jinja2* templates
- Sessions
- Static files serving
- Debugger

```
app.run(debug=True)
```



What is Flask

- *Jinja2* templates
- Sessions
- Static files serving
- Debugger
- Many extensions

```
$ pip search flask | wc -l  
1020
```

What is wrong with: routes

- Preferred way – using decorator

```
# app/view.py

@app.route('/')
def index():
    return "Hello world"

bp = Blueprint()

@bp.route('/')
def hello():
    return "Hello world"
```

What is wrong with: routes

- Preferred way – using decorator

Must have application instance ->

```
# proj/view.py
from proj import app

@app.route('/')
def index():
    return "Hello world"
```

```
# proj/__init__.py
from flask import Flask

app = Flask('proj')
```

We begin with already runnable app.

But, wait!

- * Did we have time to configure it?
- * Did we have time to configure something else (DB, etc)?

What is wrong with: routes

- Preferred way – using decorator?

How to control? ->

```
# proj/view.py
from proj import app

if app.debug:
    @app.route('/docs/')
    def show_some_docs():
        pass
```

What is wrong with: composite apps

Say you need several apps not messing their code:

```
# proj/app/__init__.py
from flask import Flask

app = Flask('proj.app')

@app.route('/')
def index():
    return render_template(
        'index.html')
```

```
# proj/admin/__init__.py
from flask import Flask

app = Flask('proj.admin')

@app.route('/')
def index():
    return render_template(
        'admin_index.html')
```

What is wrong with: composite apps

How to reuse apps' URLs?

??? ->

```
# proj/admin/__init__.py
from flask import Flask
from flask import url_for

app = Flask('proj.admin')

@app.route('/')
def index():

    url_for('app.index') #???

    return render_template(
        'admin_index.html')
```

** Blueprints can help, but application must be only one*

What is wrong with: jinja

Abused!

```
from celery import Celery
from flask import render_template

celery_app = Celery()

@celery_app.task
def send_periodic_email():
    to = get_address()
    html_body = render_template('mails/annoy.html')
    send_mail(to, html_body)
```

Web Worker == Celery Worker

What is wrong with: global vars

- Magical request object
thread-local proxy object

Need request instance ->

```
from celery import Celery
from flask import render_template, session

celery_app = Celery()

@celery_app.task
def send_periodic_email():
    to = get_address()
    user_name = session['user_name']
    html_body = render_template(
        'mails/annoy.html',
        {user_name: user_name})
    send_mail(to, html_body)
```

What is wrong with: global vars

- Magical request object

```
from flask import Flask, request
from proj.db import session
from proj.redis import redis
# ... import other global instance

app = Flask()

@app.route('/')
def index():
    uid = redis.get(request.cookies.get('cid'))
    user_row = session.execute("select * from ...")
    # ...
```

Alternatives

- Enough using WSGI, start using asyncio/aiohttp

Alternatives

- Can't cope asyncio? At least try using Flask less self-destructive:
 - Get rid of *app.route* decorator

```
def index():  
    return "Hello world"  
  
# ... more views  
  
def setup_routes(app):  
    app.add_url_rule('/', 'index', index)  
    app.add_url_rule(...)
```

```
from flask import Flask  
from . import views  
  
def main():  
    app = Flask()  
    views.setup_routes(app)  
    # more configuration
```

Alternatives

- Can't cope asyncio? At least try using Flask less self-destructive:
 - Use any kind of dependencies instead of global (thread-local proxy) variables:

```
from flask import Flask, current_app

class MyApp:
    def __init__(self, *args, redis, db_conn **kw):
        super().__init__(*args, **kw)
        self.redis = redis
        self.db_conn = db_conn

def get_user_view():
    redis = current_app.redis
    ....
```

Alternatives

- Still want to use Flask — use Werkzeug!

The End

Questions?

What Flask is not — it is not “micro framework” it is a “Werkzeug wrapper”