

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 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'
                   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 vc

```
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.assertEqual(rv.data, b'Blub')
    self.assertEqual(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assertEqual(rv.data, b'')
    self.assertEqual(rv.headers['X-Method'], 'HEAD')
```

Example 3

```
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

```
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

```
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.assertEqual(rv.data, b'GET')
rv = c.head('/')
self.assertEqual(rv.data, b'')
self.assertEqual(rv.headers['X-Method'], 'HEAD')

```

Example 6

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [views.py](#) [Apache License 2.0](#)

6 vc

```

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.assertEqual(rv.data, b'Blub')
    self.assertEqual(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assertEqual(rv.data, b'')
    self.assertEqual(rv.headers['X-Method'], 'HEAD')

```

Example 7

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [views.py](#) [Apache License 2.0](#)

6 vc

```

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.assertEqual(rv.data, b'GET')
    rv = c.head('/')
    self.assertEqual(rv.data, b'')
    self.assertEqual(rv.headers['X-Method'], 'HEAD')

```

Example 8

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#)

6 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 9

Project: [everyclass-server](#) Author: [everyclass](#) File: [views.py](#) [Mozilla Public License 2.0](#)

6 vc

```

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 vc

```

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 vc

```

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', None)):
            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 vc

```

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_time_header"])
    if parse.time.get("start_time") is not None and not parse.streamed_our_response:
        # remote request time should be excluded when calculating total time
        parse.set_extra_resp_header('X-Body-Req-Time', "%.4f" % parse.time["req_time_header"])
        parse.set_extra_resp_header('X-Compute-Time',
                                     "%.4f" % (process_time() - parse.time["start_time"]))
    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

6 vc

```

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 vc

```

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.assertEqual(rv.data, b'Blub')
    self.assertEqual(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assertEqual(rv.data, b'')
    self.assertEqual(rv.headers['X-Method'], 'HEAD')

```

Example 15

Project: *flasky* Author: *RoseOu* File: *views.py* MIT License

6 vc

```

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.assertEqual(rv.data, b'GET')
rv = c.head('/')
self.assertEqual(rv.data, b'')
self.assertEqual(rv.headers['X-Method'], 'HEAD')

```

Example 16

Project: *flasky* Author: *RoseOu* File: [basic.py](#) MIT License

6 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 17

Project: *pixelAntiAdblock* Author: *Mechazawa* File: [application.py](#) GNU General Public License v3.0

6 vc

```

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 vc

```
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 vc

```
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 vc

```
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 vc

```
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 vc

```
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 vc

```
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 24

Project: *hydrus* Author: *HTTP-APIs* File: *resources.py* MIT License

5 vc

```
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".format(
            status_id)
        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 25

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) MIT License

5 vc

```

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 26

Project: *hydrus* Author: *HTTP-APIs* File: [resources.py](#) MIT License

5 vc

```

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_url,
                session=get_session(),
                last_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 27

Project: *hydrus* Author: *HTTP-APIs* File: [auth.py](#) MIT License

5 vc

```

def token_response(token: str) -> Response:
    """
    Return succesful 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 vc

```

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 vc

```

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 vc

```

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 vc

```

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 vc](#)

```

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 vc](#)

```

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 vc](#)

```
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.assertEqual(rv.status_code, 200)
    self.assertIn(b'Response', rv.data)
    self.assertEqual(len(called), 1)
```

Example 35

```
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

```
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 vc

```

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.assertEqual(rv.status_code, 200)
    self.assertIn(b'Response', rv.data)
    self.assertEqual(len(called), 1)

```

Example 38

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#)

5 vc

```

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.assertEqual(rv.status_code, 200)
    self.assertIn(b'Response', rv.data)
    self.assertEqual(len(called), 1)

```

Example 39

Project: *gpu-mux* Author: *google* File: [gpumux.py](#) [Apache License 2.0](#)

5 vc

```

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

```
def enter_maintenance():
    config = get_config()
    auth = request.authorization
    if auth \
        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:
        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 41

```
def generate_simple_resp_page(errormsg=b'We Got An Unknown Error', error_code=500)
    """
    :type errormsg: bytes
    :type error_code: int
    :rtype: Response
    """
    return make_response(errormsg, error_code)
```

Example 42

```
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 vc

```
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_str))

        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 vc

```
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

```
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

```
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, cc
    return cookie_string

# ##### End Server Response Handler #####

# ##### Begin Client Request Handler #####
```

Example 47

```
def crossdomain_xml():
    return Response("""<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.macromedia.com/xml/dtds/cross-don
<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

```
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 vc

```

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 vc

```

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----- 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----- {} ----- \n".format(rv_name)
        dump += attributes(rv)

```

```
    dump += "\n----- {}.headers ----- \n".format(rv_name)
    dump += pformat(list(rv.headers.items()))

    dump += "\n----- end dump ----- \n"

    return dump
```
