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 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 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></htm
    reply = [{
        'firstName': 'csproot',
        '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': 1500000000000}
    database['licenses'].append(new)
    return json.dumps([get_object_id('licenses', 'ID', '255d9673-7281-43c4-be57-fc
```

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 = nextI
```

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 a
                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"),

    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 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 a
                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 errormsg: 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':
                mediaId = recMsg.MediaId
                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.ar
    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=\"nosh
           "<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)])
```