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 2.0

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 2

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *regression.py* Apache License 2.0

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 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 2.0 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 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 2.0

6 vo

```
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" License

6 vo

```
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: *picourl* Author: *PradheepShrinivasan* File: *views.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
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: *SmartBlug* 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: *SmartBlug* 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](targe

    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 2.0

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)

        @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 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 2.0 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)

        @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 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 38

Project: *flasky* Author: *RoseOu* File: *blueprints.py* MIT License 5 vc

```
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 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 License 2.0 5 vo

```
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" License 5 vo

```
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_r
            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 currer
                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" License 5 vo

```
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" License 5 vo

```
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" License 5 vo

```
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)
```