Python flask.flash() Examples

The following are code examples for showing how to use *flask.flash()*. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

Example 1

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *signals.py* Apache License 2.0 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
                self.assert_equal(message, 'This is a flash message')
                self.assert_equal(category, 'notice')
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *signals.py* Apache License 2.0 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
```

```
            self.assert_equal(message, 'This is a flash message')
            self.assert_equal(category, 'notice')
    finally:
        flask.message_flashed.disconnect(record, app)
```

Example 3

Project: *flasky* Author: *RoseOu* File: *signals.py* MIT License 6 vo

```
def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')
    def index():
        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')

    recorded = []
    def record(sender, message, category):
        recorded.append((message, category))

    flask.message_flashed.connect(record, app)
    try:
        client = app.test_client()
        with client.session_transaction():
            client.get('/')
            self.assert_equal(len(recorded), 1)
            message, category = recorded[0]
            self.assert_equal(message, 'This is a flash message')
            self.assert_equal(category, 'notice')
    finally:
        flask.message_flashed.disconnect(record, app)
```

Example 4

Project: *Flask_Blog* Author: *sugarguo* File: *signals.py* GNU General Public License v3.0 6 vo

```
def test_flash_signal(self):
    app = flask.Flask(__name__)
    app.config['SECRET_KEY'] = 'secret'

    @app.route('/')
    def index():
        flask.flash('This is a flash message', category='notice')
        return flask.redirect('/other')

    recorded = []
    def record(sender, message, category):
        recorded.append((message, category))

    flask.message_flashed.connect(record, app)
    try:
        client = app.test_client()
        with client.session_transaction():
            client.get('/')
            self.assert_equal(len(recorded), 1)
            message, category = recorded[0]
            self.assert_equal(message, 'This is a flash message')
            self.assert_equal(category, 'notice')
```

```
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 5

Project: *fixmynotes.com* Author: *mariowr2* File: *__init__.py* MIT License 6 vo

```
def uploaded_file(filename, splitting_mode):


        print "sending file with mode "+str(splitting_mode)

        output_filename = split_pdf.process_pdf(filename, file_input_location_absc
        print_debug_msg("returned filename is "+output_filename)


        if allowed_filename(output_filename):
                return redirect(url_for('serve_file', output_filename=output_filer
        else:
                flash(output_filename)
                return redirect(url_for('unsuccesful'))



#serve the file with the new name as part of the url for
```

Example 6

Project: *platzi-hello-gae* Author: *xertica-cloud* File: *signals.py* GNU General Public License v2.0 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
                self.assert_equal(message, 'This is a flash message')
                self.assert_equal(category, 'notice')
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 7

Project: *github-stats* Author: *lipis* File: *feedback.py* MIT License 6 vo

```
def feedback():
  if not config.CONFIG_DB.feedback_email:
    return flask.abort(418)

  form = FeedbackForm(obj=auth.current_user_db())
  if not config.CONFIG_DB.has_anonymous_recaptcha or auth.is_logged_in():
    del form.recaptcha
  if form.validate_on_submit():
    body = '%s\n\n%s' % (form.message.data, form.email.data)
    kwargs = {'reply_to': form.email.data} if form.email.data else {}
    task.send_mail_notification('%s...' % body[:48].strip(), body, **kwargs)
    flask.flash('Thank you for your feedback!', category='success')
    return flask.redirect(flask.url_for('welcome'))

  return flask.render_template(
    'feedback.html',
    title='Feedback',
    html_class='feedback',
    form=form,
  )
```

Example 8

Project: *github-stats* Author: *lipis* File: *auth.py* MIT License 6 vo

```
def signin_user_db(user_db):
  if not user_db:
    return flask.redirect(flask.url_for('signin'))
  flask_user_db = FlaskUser(user_db)
  auth_params = flask.session.get('auth-params', {
    'next': flask.url_for('welcome'),
    'remember': False,
  })
  flask.session.pop('auth-params', None)
  if flask_login.login_user(flask_user_db, remember=auth_params['remember']):
    user_db.put_async()
    if user_db.github:
      return flask.redirect(flask.url_for('gh_account', username=user_db.github))
    return flask.redirect(util.get_next_url(auth_params['next']))
  flask.flash('Sorry, but you could not sign in.', category='danger')
  return flask.redirect(flask.url_for('signin'))
```

Example 9

Project: *tesismometro* Author: *joapaspe* File: *signals.py* MIT License 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
```

```
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
                self.assert_equal(message, 'This is a flash message')
                self.assert_equal(category, 'notice')
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 10

Project: *LuckyCAT* Author: *fkie-cad* File: *Jobs.py* GNU General Public License v3.0 6 vo

```
def delete_job(job_id):
    if job_id is None:
        flask.abort(400, description="Invalid job ID")
    if flask.request.method == 'POST':
        job = Job.objects.get(id=job_id)
        if job:
            if not can_do_stuff_with_job(current_user, job.owner):
                logging.error('User %s can not delete job with id %s' %
                              (current_user.email, str(job.id)))
                flask.flash('You are not allow to delete this job.')
            else:
                job.delete()
                crashes = Crash.objects(job_id=job_id)
                crashes.delete()
        return flask.redirect('/jobs/show')
    else:
        return flask.render_template('jobs_delete.html', id=job_id)
```

Example 11

Project: *LuckyCAT* Author: *fkie-cad* File: *Jobs.py* GNU General Public License v3.0 6 vo

```
def jobs_download(job_id):
    # FIXME may crash if no crashes available
    if job_id is None:
        flask.flash("Invalid job ID")
        return flask.redirect('/jobs/show')

    job = Job.objects.get(id=job_id)
    if not can_do_stuff_with_job(current_user, job.owner):
        flask.flash('User is not allowed to download job.')
        return flask.redirect('/jobs/show')

    job_crashes = Crash.objects(job_id=job_id)
    if job_crashes:
        imz = InMemoryZip()
        summary = {}
        for c in job_crashes:
            summary[str(c.id)] = _get_summary_for_crash(c)
            imz.append("%s" % str(c.id), c.test_case)
        imz.append("summary.json", json.dumps(summary, indent=4))

        filename = os.path.join('/tmp', '%s.zip' % job_id)
        if os.path.exists(filename):
            os.remove(filename)
        imz.writetofile(filename)
        return flask.send_file(filename, as_attachment=True)
```

Example 12

Project: *neo4j-social-network* Author: *bestvibes* File: *signals.py* MIT License 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
                self.assert_equal(message, 'This is a flash message')
                self.assert_equal(category, 'notice')
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 13

Project: *neo4j-social-network* Author: *bestvibes* File: *signals.py* MIT License 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
                self.assert_equal(message, 'This is a flash message')
                self.assert_equal(category, 'notice')
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 14

6 vo

Project: *gae-init-magic* Author: *gae-init* File: *property.py* MIT License

```
def property_delete(project_id, model_id, property_id):
  user_key = auth.current_user_key()
  project_db = model.Project.get_by_id(project_id)
  if not project_db or project_db.user_key != user_key:
    flask.abort(404)

  model_db = model.Model.get_by_id(model_id, parent=project_db.key)
  if not model_db:
    flask.abort(404)

  property_db = model.Property.get_by_id(property_id, parent=model_db.key)
  if not property_db:
    flask.abort(404)

  property_db.key.delete()
  flask.flash('Property "%s" deleted.' % property_db.name, category='success')
  return flask.redirect(flask.url_for('model_view', project_id=project_db.key.id()
```

Example 15

Project: *gae-init-magic* Author: *gae-init* File: *feedback.py* MIT License 6 vo

```
def feedback():
  if not config.CONFIG_DB.feedback_email:
    return flask.abort(418)

  form = FeedbackForm(obj=auth.current_user_db())
  if not config.CONFIG_DB.has_anonymous_recaptcha or auth.is_logged_in():
    del form.recaptcha
  if form.validate_on_submit():
    body = '%s\n\n%s' % (form.message.data, form.email.data)
    kwargs = {'reply_to': form.email.data} if form.email.data else {}
    task.send_mail_notification('%s...' % body[:48].strip(), body, **kwargs)
    flask.flash('Thank you for your feedback!', category='success')
    return flask.redirect(flask.url_for('welcome'))

  return flask.render_template(
    'feedback.html',
    title='Feedback',
    html_class='feedback',
    form=form,
  )
```

Example 16

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *basic.py* Apache License 2.0 5 vo

```
def test_flashes(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testkey'

        with app.test_request_context():
            self.assert_false(flask.session.modified)
            flask.flash('Zap')
            flask.session.modified = False
            flask.flash('Zip')
            self.assert_true(flask.session.modified)
            self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

Example 17

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *basic.py* Apache License 2.0 5 vo

```
def test_flashes(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testkey'

        with app.test_request_context():
            self.assert_false(flask.session.modified)
            flask.flash('Zap')
            flask.session.modified = False
            flask.flash('Zip')
            self.assert_true(flask.session.modified)
            self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

Example 18

Project: *flasky* Author: *RoseOu* File: *basic.py* MIT License 5 vo

```
def test_flashes(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testkey'

        with app.test_request_context():
            self.assert_false(flask.session.modified)
            flask.flash('Zap')
            flask.session.modified = False
            flask.flash('Zip')
            self.assert_true(flask.session.modified)
            self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

Example 19

Project: *ActivityManager* Author: *yichengchen* File: *uploader.py* GNU General Public License v3.0 5 vo

```
def upload(activity, act, current_user):
    form = Forms.UploadFile()
    filename = None
    if form.validate_on_submit():
        ext_name = secure_filename(form.works.data.filename).split('.')[-1]

        print(current_user.name)

        filename = "{}_{}_{}.{}".format(act.title, current_user.stu_code, current_
        # try:
        directory = 'uploads/{}/'.format(activity)
        if not os.path.exists(directory):
            os.makedirs(directory)
        print(filename)
        form.works.data.save(directory + filename)
        file_size = "{0}k".format(os.path.getsize(directory + filename) / 1000)

        data = UploadHistory(current_user.sid, activity, file_size)
        db.session.add(data)
        db.session.commit()
        # except Exception as err:
            # flash("错误:" + str(err))
            # return jsonify(success=False,status="错误:" + str(err))
        print("flash upload success")
```

```
        return jsonify(success=True,status="上传成功!")
        # flash("上传成功!", 'info')
    else:
        if request.method == "POST":
            return jsonify(success=False,status="上传失败，请检查文件格式。或刷新网页 / 联
            # print("validdate fail")
            # flash("validdate fail")
    last_time = UploadHistory.query.filter_by(sid=current_user.sid, activity=activ


    if not last_time:
        last_time_msg = '还未上传过文件'
    else:
        last_time_msg = '上次上传时间: {0} , 大小: {1}'.format(last_time.time[:-7],

    return render_template('upload.html', user=current_user, form=form, filename=f
```

Example 20

Project: *automlk* Author: *pierre-chaville* File: *views_textset.py* MIT License 5 vo

```
def create_text():
    # form to create a new textset
    form = CreateTextsetForm()
    if request.method == 'POST':
        if form.validate():
            # try:
            if form.mode_file.data == 'upload':
                # check and upload a file
                filename = form.file_text.data.filename
                if filename == '' or filename.split('.')[-1].lower() != 'txt':
                    flash('file %s type must be txt' % filename)
                else:
                    form.filename.data = get_uploads_folder() + '/' + str(uuid.uui
                        -1].lower()
                    form.file_text.data.save(form.filename.data)

            create_textset(name=form.name.data,
                           description=form.description.data,
                           source=form.source.data,
                           url=form.url.data,
                           filename=form.filename.data)
            return redirect('/textset_list')
            # except Exception as e:
            #     flash(e)
        else:
            flash(", ".join([key + ': ' + form.errors[key][0] for key in form.err

    return render_template('create_text.html', form=form, config=get_config())
```

Example 21

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def index():
    # return flask.send_from_directory('./static/', 'index.html')
    flask.flash(str((fan.speed, fan.brightness)))
    return flask.render_template("index.html")
```

```
# Light Functions
```

Example 22

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def toggle_light():
    fan.light_toggle()
    flask.flash("Toggling Light")
    return flask.redirect(flask.url_for("index"))
```

Example 23

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def light_off():
    fan.light_powered_on = False
    flask.flash("Turning Light Off")
    return flask.redirect(flask.url_for("index"))
```

Example 24

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def light_on():
    fan.light_powered_on = True
    flask.flash("Turning light On")
    return flask.redirect(flask.url_for("index"))
```

Example 25

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def light_level(level):
    fan.brightness = int(level)
    flask.flash("Set light level to {}".format(level))
    return flask.redirect(flask.url_for("index"))
```

Example 26

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def dec_light():
    fan.dec_brightness()
    flask.flash("Decreased Light Level")
    return flask.redirect(flask.url_for("index"))


# Fan Functions
```

Example 27

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def inc_speed():
    fan.inc_speed()
```

```
    flask.flash("Increased Fan Speed")
    return flask.redirect(flask.url_for("index"))
```

Example 28

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def dec_speed():
    fan.dec_speed()
    flask.flash("Decreased Fan Speed")
    return flask.redirect(flask.url_for("index"))
```

Example 29

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def set_speed(speed):
    fan.speed = int(speed)
    flask.flash("Set fan speed to {}".format(speed))
    return flask.redirect(flask.url_for("index"))
```

Example 30

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def fan_toggle():
    fan.fan_toggle()
    flask.flash("Toggling Fan")
    return flask.redirect(flask.url_for("index"))
```

Example 31

Project: *SenseMe* Author: *TomFaulkner* File: *flask_app.py* GNU General Public License v3.0 5 vo

```
def fan_on():
    fan.fan_powered_on = True
    flask.flash("Turning Fan On")
    return flask.redirect(flask.url_for("index"))
```

Example 32

Project: *gitlab-tools* Author: *Salamek* File: *index.py* GNU General Public License v3.0 5 vo

```
def schedule_sync_mirror(mirror_id: int):
    # Check if mirror exists or throw 404
    found_mirror = PushMirror.query.filter_by(id=mirror_id, user=current_user).fir
    if not found_mirror.project_id:
        flask.flash('Project mirror is not created, cannot be synced', 'danger')
        return flask.redirect(flask.url_for('push_mirror.index.get_mirror'))
    task = sync_push_mirror.delay(mirror_id)
    log_task_pending(task, found_mirror, sync_push_mirror, InvokedByEnum.MANUAL)

    flask.flash('Sync has been started with UUID: {}'.format(task.id), 'success')
    return flask.redirect(flask.url_for('push_mirror.index.get_mirror'))
```

Example 33

Project: *gitlab-tools* Author: *Salamek* File: *index.py* GNU General Public License v3.0 5 vo

```
def schedule_delete_mirror(mirror_id: int):
    mirror_detail = PushMirror.query.filter_by(id=mirror_id, user=current_user).fi
    mirror_detail.is_deleted = True
    db.session.add(mirror_detail)
    db.session.commit()

    delete_push_mirror.delay(mirror_detail.id)

    flask.flash('Push mirror was deleted successfully.', 'success')

    return flask.redirect(flask.url_for('push_mirror.index.get_mirror'))
```

Example 34

Project: *gitlab-tools* Author: *Salamek* File: *index.py* GNU General Public License v3.0 5 vo

```
def new_fingerprint():
    form = NewForm(
        flask.request.form
    )
    if flask.request.method == 'POST' and form.validate():
        flask.flash('New fingerprint was added.', 'success')
        return flask.redirect(flask.url_for('fingerprint.index.get_fingerprint'))

    return flask.render_template('fingerprint.index.new.html', form=form)
```

Example 35

Project: *gitlab-tools* Author: *Salamek* File: *index.py* GNU General Public License v3.0 5 vo

```
def get_new_rsa_key():

    current_user.is_rsa_pair_set = False
    current_user.gitlab_deploy_key_id = None
    db.session.add(current_user)
    db.session.commit()

    create_rsa_pair.delay(current_user.id)
    flask.flash('New RSA pair key has been requested!', 'success')
    return flask.redirect(flask.url_for('home.index.get_home'))
```

Example 36

Project: *gitlab-tools* Author: *Salamek* File: *index.py* GNU General Public License v3.0 5 vo

```
def schedule_sync_mirror(mirror_id: int):
    # Check if mirror exists or throw 404
    found_mirror = PullMirror.query.filter_by(id=mirror_id, user=current_user).fir
    if not found_mirror.project_id:
        flask.flash('Project mirror is not created, cannot be synced', 'danger')
        return flask.redirect(flask.url_for('pull_mirror.index.get_mirror'))
    task = sync_pull_mirror.delay(mirror_id)
    log_task_pending(task, found_mirror, sync_pull_mirror, InvokedByEnum.MANUAL)

    flask.flash('Sync has been started with UUID: {}'.format(task.id), 'success')
    return flask.redirect(flask.url_for('pull_mirror.index.get_mirror'))
```

Example 37

Project: *gitlab-tools* Author: *Salamek* File: *index.py* GNU General Public License v3.0 5 vo

```
def schedule_delete_mirror(mirror_id: int):
    mirror_detail = PullMirror.query.filter_by(id=mirror_id, user=current_user).fi
    mirror_detail.is_deleted = True
    db.session.add(mirror_detail)
    db.session.commit()

    delete_pull_mirror.delay(mirror_detail.id)

    flask.flash('Pull mirror was deleted successfully.', 'success')

    return flask.redirect(flask.url_for('pull_mirror.index.get_mirror'))
```

Example 38

Project: *Flask_Blog* Author: *sugarguo* File: *basic.py* GNU General Public License v3.0 5 vo

```
def test_flashes(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testkey'

        with app.test_request_context():
            self.assert_false(flask.session.modified)
            flask.flash('Zap')
            flask.session.modified = False
            flask.flash('Zip')
            self.assert_true(flask.session.modified)
            self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

Example 39

Project: *fixmynotes.com* Author: *mariowr2* File: *__init__.py* MIT License 5 vo

```
def upload_pdf():
        if request.method == 'POST':

                splitting_mode = request.form['mode'] # get the radio button selec
                print "SPLITTING MODE SET TO "+str(splitting_mode)

                if 'pdf' in request.files:
                        pdf_file = request.files['pdf']
                        if not pdf_file.filename == '':
                                if pdf_file and allowed_filename(pdf_file.filename
                                        filename = secure_filename(pdf_file.filena
                                        if filename:
                                                pdf_file.save(os.path.join(app.cor
                                                return redirect(url_for('uploaded_
                                        else:
                                                flash("There seems to be somethin
                                                return redirect(url_for('unsuccesf
                                else:
                                        clear_uploaded_file(pdf_file.filename) # c
                                        flash("This webapp only works with pdf fi
                                        return redirect(url_for('unsuccesful'))
                        else:
                                flash("No file was selected.")
                                return redirect(url_for('unsuccesful'))
                else:
                        flash("Failed to upload file.")
                        return redirect(url_for('unsuccesful'))
```

```
        return render_template('upload.html') # if not a post request, show the ht


#process pdf, verify successful and then send it to a custom url
```

Example 40

Project: *fixmynotes.com* Author: *mariowr2* File: *__init__.py* MIT License 5 vo

```
def handle_request_too_large(e):
        flash("Terrible error ocurred. Maximum file size is "+str(MAX_FILE_SIZE)+
        return redirect(url_for('unsuccesful'))
```

Example 41

Project: *fixmynotes.com* Author: *mariowr2* File: *__init__.py* MIT License 5 vo

```
def handle_bad_request(e):
        flash("Terrible error ocurred. (Bad Request)")
        return redirect(url_for('error'))
```

Example 42

Project: *fixmynotes.com* Author: *mariowr2* File: *__init__.py* MIT License 5 vo

```
def handle_not_found(e):
        flash("4 0 4")
        return redirect(url_for('error'))
```

Example 43

Project: *platzi-hello-gae* Author: *xertica-cloud* File: *basic.py* GNU General Public License v2.0 5 vo

```
def test_flashes(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testkey'

        with app.test_request_context():
            self.assert_false(flask.session.modified)
            flask.flash('Zap')
            flask.session.modified = False
            flask.flash('Zip')
            self.assert_true(flask.session.modified)
            self.assert_equal(list(flask.get_flashed_messages()), ['Zap', 'Zip'])
```

Example 44

Project: *github-stats* Author: *lipis* File: *profile.py* MIT License 5 vo

```
def profile_password():
  if not config.CONFIG_DB.has_email_authentication:
    flask.abort(418)
  user_db = auth.current_user_db()
  form = ProfilePasswordForm(obj=user_db)

  if not user_db.password_hash:
    del form.old_password
```

```
  if form.validate_on_submit():
    errors = False
    old_password = form.old_password.data if form.old_password else None
    new_password = form.new_password.data
    if new_password or old_password:
      if user_db.password_hash:
        if util.password_hash(user_db, old_password) != user_db.password_hash:
          form.old_password.errors.append('Invalid current password')
          errors = True

      if not (form.errors or errors):
        user_db.password_hash = util.password_hash(user_db, new_password)
        flask.flash('Your password has been changed.', category='success')

    if not (form.errors or errors):
      user_db.put()
      return flask.redirect(flask.url_for('profile'))

  return flask.render_template(
    'profile/profile_password.html',
    title=user_db.name,
    html_class='profile-password',
    form=form,
    user_db=user_db,
  )
```

Example 45

Project: *github-stats* Author: *lipis* File: *user.py* MIT License 5 vo

```
def user_verify(token):
  user_db = auth.current_user_db()
  if user_db.token != token:
    flask.flash('That link is either invalid or expired.', category='danger')
    return flask.redirect(flask.url_for('profile'))
  user_db.verified = True
  user_db.token = util.uuid()
  user_db.put()
  flask.flash('Hooray! Your email is now verified.', category='success')
  return flask.redirect(flask.url_for('profile'))


###############################################################################
# User Forgot
###############################################################################
```

Example 46

Project: *github-stats* Author: *lipis* File: *user.py* MIT License 5 vo

```
def user_reset(token=None):
  user_db = model.User.get_by('token', token)
  if not user_db:
    flask.flash('That link is either invalid or expired.', category='danger')
    return flask.redirect(flask.url_for('welcome'))

  if auth.is_logged_in():
    flask_login.logout_user()
    return flask.redirect(flask.request.path)

  form = UserResetForm()
```

```
  if form.validate_on_submit():
    user_db.password_hash = util.password_hash(user_db, form.new_password.data)
    user_db.token = util.uuid()
    user_db.verified = True
    user_db.put()
    flask.flash('Your password was changed succesfully.', category='success')
    return auth.signin_user_db(user_db)

  return flask.render_template(
    'user/user_reset.html',
    title='Reset Password',
    html_class='user-reset',
    form=form,
    user_db=user_db,
  )


###############################################################################
# User Activate
###############################################################################
```

Example 47

Project: *github-stats* Author: *lipis* File: *user.py* MIT License 5 vo

```
def user_activate(token):
  if auth.is_logged_in():
    flask_login.logout_user()
    return flask.redirect(flask.request.path)

  user_db = model.User.get_by('token', token)
  if not user_db:
    flask.flash('That link is either invalid or expired.', category='danger')
    return flask.redirect(flask.url_for('welcome'))

  form = UserActivateForm(obj=user_db)
  if form.validate_on_submit():
    form.populate_obj(user_db)
    user_db.password_hash = util.password_hash(user_db, form.password.data)
    user_db.token = util.uuid()
    user_db.verified = True
    user_db.put()
    return auth.signin_user_db(user_db)

  return flask.render_template(
    'user/user_activate.html',
    title='Activate Account',
    html_class='user-activate',
    user_db=user_db,
    form=form,
  )


###############################################################################
# User Merge
###############################################################################
```

Example 48

Project: *github-stats* Author: *lipis* File: *linkedin.py* MIT License 5 vo

```
def linkedin_authorized():
  id_token = linkedin.authorize_access_token()
  if id_token is None:
    flask.flash('You denied the request to sign in.')
    return flask.redirect(util.get_next_url())

  me = linkedin.get('people/~:(id,first-name,last-name,email-address)')
  user_db = retrieve_user_from_linkedin(me.json())
  return auth.signin_user_db(user_db)
```

Example 49

Project: *github-stats* Author: *lipis* File: *microsoft.py* MIT License 5 vo

```
def microsoft_authorized():
  id_token = microsoft.authorize_access_token()
  if id_token is None:
    flask.flash('You denied the request to sign in.')
    return flask.redirect(util.get_next_url())
  me = microsoft.get('me')
  user_db = retrieve_user_from_microsoft(me.json())
  return auth.signin_user_db(user_db)
```

Example 50

Project: *github-stats* Author: *lipis* File: *auth.py* MIT License 5 vo

```
def signin_oauth(oauth_app, scheme=None):
  try:
    flask.session.pop('oauth_token', None)
    save_request_params()
    return oauth_app.authorize_redirect(flask.url_for(
      '%s_authorized' % oauth_app.name, _external=True, _scheme=scheme
    ))
  except authlib.client.OAuthError:
    flask.flash(
      'Something went wrong with sign in. Please try again.',
      category='danger',
    )
    return flask.redirect(flask.url_for('signin', next=util.get_next_url()))
```