

Python `flask.Markup()` Examples

The following are code examples for showing how to use `flask.Markup()`. 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: *ras-frontstage* Author: *ONSdigital* File: [create_message.py](#) MIT License

7 vc

```
def create_message(session):
    """Creates and sends a message outside of the context of an existing conversat
    survey = request.args['survey']
    ru_ref = request.args['ru_ref']
    party_id = session['party_id']
    form = SecureMessagingForm(request.form)
    if request.method == 'POST' and form.validate():
        logger.info("Form validation successful", party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for("secure_message_bp.view_conversation",
                             thread_id=sent_message['thread_id']) + "#latest-mess
        flash(Markup('Message sent. <a href={}>View Message</a>'.format(thread_ur
        return redirect(url_for('secure_message_bp.view_conversation_list'))

    else:
        return render_template('secure-messages/secure-messages-view.html',
                               ru_ref=ru_ref, survey=survey,
                               form=form, errors=form.errors, message={})
```

Example 2

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

2.0

6 vc

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])
```

Example 3

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

2.0

6 vc

```
def test_escaping(self):
    text = '<p>Hello World!'
```

```

app = flask.Flask(__name__)
@app.route('/')
def index():
    return flask.render_template('escaping_template.html', text=text,
                                html=flask.Markup(text))
lines = app.test_client().get('/').data.splitlines()
self.assertEqual(lines, [
    b'&lt;p&gt;Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'&lt;p&gt;Hello World!',
    b'<p>Hello World!'
])

```

Example 4

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

6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                    html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 5

Project: *Flask_Blog* Author: *sugarguo* File: [templating.py](#) GNU General Public License v3.0

6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                    html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 6

Project: *platzi-hello-gae* Author: *xertica-cloud* File: [templating.py](#) GNU General Public License v2.0

6 vc

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 7

Project: *tesismometro* Author: *joapaspe* File: [templating.py](#) MIT License

6 vc

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 8

Project: *neo4j-social-network* Author: *bestvibes* File: [run.py](#) MIT License

6 vc

```
def error_handler(error):
    msg = "Request resulted in {}".format(error)
    current_app.logger.warning(msg, exc_info=error)

    if isinstance(error, HTTPException):
        description = error.get_description(request.environ)
        code = error.code
        name = error.name
    else:
        description = ("We encountered an error "
                       "while trying to fulfill your request")
        code = 500
        name = 'Internal Server Error'

    # Flask supports looking up multiple templates and rendering the first
    # one it finds. This will let us create specific error pages
    # for errors where we can provide the user some additional help.
    # (Like a 404, for example).
    templates_to_try = ['errors/{}.html'.format(code), 'errors/generic.html']

```

```
return render_template(templates_to_try,
                        code=code,
                        name=Markup(name),
                        description=Markup(description),
                        error=error)
```

Example 9

Project: *neo4j-social-network* Author: *bestvibes* File: [templating.py](#) MIT License

6 vc

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])
```

Example 10

Project: *neo4j-social-network* Author: *bestvibes* File: [templating.py](#) MIT License

6 vc

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])
```

Example 11

Project: *AneMo* Author: *jspargo* File: [templating.py](#) GNU General Public License v2.0

6 vc

```
def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
```

```

self.assert_equal(lines, [
    b'&lt;p&gt;Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'<p>Hello World!',
    b'&lt;p&gt;Hello World!',
    b'<p>Hello World!'
])

```

Example 12

Project: *oa_qian* Author: *sunqb* File: *templating.py* Apache License 2.0

6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assert_equal(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 13

Project: *fbvoting* Author: *corradomonti* File: *serverutils.py* MIT License

6 vc

```

def configure_app(my_app):
    # configuring the environment
    fbvoting.db.categories.bootstrap_categories()
    socket.setdefaulttimeout(10)

    # configuring flask
    if fbvoting.conf.Config.PROFILE:
        my_app.wsgi_app = ProfilerMiddleware(my_app.wsgi_app, restrictions = [30])
    my_app.wsgi_app = ProxyFix(my_app.wsgi_app)
    my_app.config.from_object('fbvoting.conf.Config')
    my_app.secret_key = fbvoting.conf.MY_SECRET_KEY
    my_app.jinja_env.filters['json'] = lambda v: Markup(json.dumps(v))
    fbvoting.redis.cache.redis_store.init_app(my_app)
    my_debugged_app = DebuggedApplication(my_app)

    #logging
    fbvoting.mylogging.configure_logging(add_handler_to=[my_app.logger])

    return my_app, my_debugged_app

```

decorators

Example 14

```
def build_chart_overview():
    data = commons.get_base_data()
    header = """
        Global Recommendations
        <small>
            Best tunes, from liquid democracy
        </small>
        """

    data.update({
        'active_section': 'charts',
        'header': Markup(header),
        'rank_type': 'chart'
    })

    return render_template('rank.html', **data)
```

Example 15

```
def build_empty_home():
    data = commons.get_base_data()
    data['already_logged'] = False
    data['js_on_loaded'] = Markup("""
        $('#intro-container').load(personal("ajax/intro"),
            function( response, status, xhr ) {
                if ( status != "success" && xhr.status != 0 ) {
                    var msg = "There is a problem communicating to Facebook from c
                    msg+= "Please, try again or <a href='/about#contacts'>contact
                    $('#intro-container').html( msg );
                    $('#intro-container').addClass("alert-message block-message");
                } else {
                    $('#fb-error-msg').hide();
                }
                update_links();
            }
        );
    """);
    return render_template('empty.html', **data)
```

Example 16

```
def ego_petting_msg(userid):
    vote_success = find_success_of_advices(userid, only_above=1)
    if vote_success:
        category, vote, n_listenings = random.choice(vote_success)
        report.mark('ego-petting-msg')
        return Markup("""
<p><div class="alert-message block-message success row span8">
    <div class="column span">
        <h4>
        Your <a href="votes/%s">%s vote</a> <em>%s</em>
        has been listened %i times!
    </h4></div>
    """)
```

```

</div></div></p>
        """ % (category, category, category, dbdocs2str([vote]), n_listenings)
    else:
        return ''

```

Example 17

Project: *Where2Eat* Author: *thetimothyp* File: [templating.py](#) [Creative Commons Zero v1.0 Universal](#) 6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 18

Project: *PennApps2015-Heartmates* Author: *natanlailari* File: [templating.py](#) [Apache License 2.0](#) 6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 19

Project: *noobotkit* Author: *nazroll* File: [templating.py](#) [MIT License](#) 6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [

```

```

        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 20

Project: *xuemo* Author: *skycucumber* File: [templating.py](#) GNU General Public License v2.0

6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 21

Project: *chihu* Author: *yelongyu* File: [templating.py](#) GNU General Public License v3.0

6 vc

```

def test_escaping(self):
    text = '<p>Hello World!'
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('escaping_template.html', text=text,
                                     html=flask.Markup(text))
    lines = app.test_client().get('/').data.splitlines()
    self.assertEqual(lines, [
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'<p>Hello World!',
        b'&lt;p&gt;Hello World!',
        b'<p>Hello World!'
    ])

```

Example 22

Project: *airflow* Author: *apache* File: [utils.py](#) Apache License 2.0

6 vc

```

def task_instance_link(attr):
    dag_id = attr.get('dag_id')
    task_id = attr.get('task_id')
    execution_date = attr.get('execution_date')
    url = url_for(
        'Airflow.task',

```



```

dag_id=dag_id,
task_id=task_id,
execution_date=execution_date.isoformat())
url_root = url_for(
    'Airflow.graph',
    dag_id=dag_id,
    root=task_id,
    execution_date=execution_date.isoformat())
return Markup(
    """
    <span style="white-space: nowrap;">
    <a href="{url}">{task_id}</a>
    <a href="{url_root}" title="Filter on this task and upstream">
    <span class="glyphicon glyphicon-filter" style="margin-left: 0px;"
      aria-hidden="true"></span>
    </a>
    </span>
    """).format(url=url, task_id=task_id, url_root=url_root)

```

Example 23

Project: *cpm* Author: *tzabal* File: [web.py](#) [Apache License 2.0](#)

6 vc

```

def index():
    if request.method == 'POST':
        project_file = request.files['project_file']
        if project_file:
            try:
                project = cpm.validate(project_file)
            except cpm.ProjectValidationException as exc:
                return render_template('index.html', error=exc)
            cpmnet = cpm.CriticalPathMethod(project)
            cpmnet.run_cpm()
            results, images, optimum_solution = cpmnet.get_results('static/results')
            return render_template('results.html',
                results_table=Markup(_get_html_results_table(
                    images=zip(images, range(0, len(images))),
                    force_reload=str(time.time()),
                    optimum_total_cost=optimum_solution[0],
                    optimum_project_duration=optimum_solution[1]))
        else:
            return render_template('index.html', error='No project file has been u

    return render_template('index.html')

```

Example 24

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

5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now

```

```

        flask.session['t'] = (1, 2, 3)
        return response

@app.route('/')
def dump_session_contents():
    return pickle.dumps(dict(flask.session))

c = app.test_client()
c.get('/')
rv = pickle.loads(c.get('/').data)
self.assertEqual(rv['m'], flask.Markup('Hello!'))
self.assertEqual(type(rv['m']), flask.Markup)
self.assertEqual(rv['dt'], now)
self.assertEqual(rv['u'], the_uuid)
self.assertEqual(rv['t'], (1, 2, 3))

```

Example 25

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

[5 vc](#)

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['t'] = (1, 2, 3)
        return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
    self.assertEqual(rv['m'], flask.Markup('Hello!'))
    self.assertEqual(type(rv['m']), flask.Markup)
    self.assertEqual(rv['dt'], now)
    self.assertEqual(rv['u'], the_uuid)
    self.assertEqual(rv['t'], (1, 2, 3))

```

Example 26

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

[5 vc](#)

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):

```

```

        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

@app.route('/')
def dump_session_contents():
    return pickle.dumps(dict(flask.session))

c = app.test_client()
c.get('/')
rv = pickle.loads(c.get('/').data)
self.assertEqual(rv['m'], flask.Markup('Hello!'))
self.assertEqual(type(rv['m']), flask.Markup)
self.assertEqual(rv['dt'], now)
self.assertEqual(rv['u'], the_uuid)
self.assertEqual(rv['b'], b'\xff')
self.assertEqual(type(rv['b']), bytes)
self.assertEqual(rv['t'], (1, 2, 3))

```

Example 27

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

5 vc

```

def recaptcha_html(self, server, query, options):
    html = current_app.config.get('RECAPTCHA_HTML', RECAPTCHA_HTML)
    return Markup(html % dict(
        script_url='%schallenge?%s' % (server, query),
        frame_url='%snoscript?%s' % (server, query),
        options=json.dumps(options, cls=JSONEncoder)
    ))

```

Example 28

Project: *ras-frontstage* Author: *ONSdigital* File: *message_get.py* MIT License

5 vc

```

def view_conversation(session, thread_id):
    """Endpoint to view conversations by thread_id

    NOTE: /thread/<thread_id> endpoint is there for compatability reasons. All ne
    From September 2019 onwards, it should be safe to remove the /thread endpoint
    sure that it's a very small number of people trying to hit it first!).
    """
    party_id = session.get('party_id')
    logger.info("Getting conversation", thread_id=thread_id, party_id=party_id)
    # TODO, do we really want to do a GET every time, even if we're POSTing? Rops
    # way so we can get it working, then get it right.
    conversation = get_conversation(thread_id)
    logger.info('Successfully retrieved conversation', thread_id=thread_id, party_id=party_id)
    try:
        refined_conversation = [refine(message) for message in reversed(conversation)]
    except KeyError as e:
        logger.error('Message is missing important data', thread_id=thread_id, party_id=party_id)
        raise e

    if refined_conversation[-1]['unread']:
        remove_unread_label(refined_conversation[-1]['message_id'])

    form = SecureMessagingForm(request.form)

```

```

form.subject.data = refined_conversation[0].get('subject')

if not conversation['is_closed']:
    if form.validate_on_submit():
        logger.info("Sending message", thread_id=thread_id, party_id=party_id)
        msg_to = get_msg_to(refined_conversation)
        send_message(_get_message_json(form, refined_conversation[0], msg_to=msg_to)
        logger.info("Successfully sent message", thread_id=thread_id, party_id=party_id)
        thread_url = url_for("secure_message_bp.view_conversation", thread_id=thread_id)
        flash(Markup('Message sent. <a href={}>View Message</a>'.format(thread_id)))
        return redirect(url_for('secure_message_bp.view_conversation_list'))

return render_template('secure-messages/conversation-view.html',
                       form=form,
                       conversation=refined_conversation,
                       conversation_data=conversation)

```

Example 29

Project: *SayluaLegacy* Author: *saylua* File: [pagination.py](#) GNU Affero General Public License v3.0 5 vc

```

def render(self):
    return Markup(render_template("pagination.html", current_page=self.current_page,
                                page_count=self.page_count, url_base=self.url_base, url_end=self.url_end,
                                page_buffer=2))

```

Example 30

Project: *SayluaLegacy* Author: *saylua* File: [pagination.py](#) GNU Affero General Public License v3.0 5 vc

```

def render(self):
    return Markup(render_template("pagination.html", current_page=self.current_page,
                                page_count=self.page_count, url_base=self.url_base, url_end=self.url_end,
                                page_buffer=2))

```

Example 31

Project: *SayluaLegacy* Author: *saylua* File: [template_filters.py](#) GNU Affero General Public License v3.0 5 vc

```

def saylua_markdown(text):
    return Markup(md.reset().convert(text))

```

Example 32

Project: *Flask_Blog* Author: *sugarguo* File: [basic.py](#) GNU General Public License v3.0 5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now

```

```

        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

@app.route('/')
def dump_session_contents():
    return pickle.dumps(dict(flask.session))

c = app.test_client()
c.get('/')
rv = pickle.loads(c.get('/').data)
self.assertEqual(rv['m'], flask.Markup('Hello!'))
self.assertEqual(type(rv['m']), flask.Markup)
self.assertEqual(rv['dt'], now)
self.assertEqual(rv['u'], the_uuid)
self.assertEqual(rv['b'], b'\xff')
self.assertEqual(type(rv['b']), bytes)
self.assertEqual(rv['t'], (1, 2, 3))

```

Example 33

Project: *maple-file* Author: *honmaple* File: *admin.py* BSD 3-Clause "New" or "Revised" License

5 vc

```

def _list_thumbnail(view, context, model, name):
    return Markup('' % model.url.replace('photo', 'thumb'))

```

Example 34

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

5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['t'] = (1, 2, 3)
        return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
    self.assertEqual(rv['m'], flask.Markup('Hello!'))
    self.assertEqual(type(rv['m']), flask.Markup)
    self.assertEqual(rv['dt'], now)
    self.assertEqual(rv['u'], the_uuid)
    self.assertEqual(rv['t'], (1, 2, 3))

```

Example 35

5 vc

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
    self.assertEqual(rv['m'], flask.Markup('Hello!'))
    self.assertEqual(type(rv['m']), flask.Markup)
    self.assertEqual(rv['dt'], now)
    self.assertEqual(rv['u'], the_uuid)
    self.assertEqual(rv['b'], b'\xff')
    self.assertEqual(type(rv['b']), bytes)
    self.assertEqual(rv['t'], (1, 2, 3))
```

Example 36

```
def merge_metadata(to_save, meta):
    if to_save['author_name'] == _(u'Unknown'):
        to_save['author_name'] = ''
    if to_save['book_title'] == _(u'Unknown'):
        to_save['book_title'] = ''
    for s_field, m_field in [
        ('tags', 'tags'), ('author_name', 'author'), ('series', 'series'),
        ('series_index', 'series_id'), ('languages', 'languages'),
        ('book_title', 'title')]:
        to_save[s_field] = to_save[s_field] or getattr(meta, m_field, '')
    to_save["description"] = to_save["description"] or Markup(
        getattr(meta, 'description', '')).unescape()
```

Example 37

```
def recaptcha_html(self, public_key):
    html = current_app.config.get('RECAPTCHA_HTML')
    if html:
        return Markup(html)
    params = current_app.config.get('RECAPTCHA_PARAMETERS')
    script = RECAPTCHA_SCRIPT
    if params:
```

```

script += u'?' + url_encode(params)

attrs = current_app.config.get('RECAPTCHA_DATA_ATTRS', {})
attrs['sitekey'] = public_key
snippet = u' '.join([u'data-%s="%s"' % (k, attrs[k]) for k in attrs])
return Markup(RECAPTCHA_TEMPLATE % (script, snippet))

```

Example 38

Project: *neo4j-social-network* Author: *bestvibes* File: *basic.py* MIT License

5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
    self.assertEqual(rv['m'], flask.Markup('Hello!'))
    self.assertEqual(type(rv['m']), flask.Markup)
    self.assertEqual(rv['dt'], now)
    self.assertEqual(rv['u'], the_uuid)
    self.assertEqual(rv['b'], b'\xff')
    self.assertEqual(type(rv['b']), bytes)
    self.assertEqual(rv['t'], (1, 2, 3))

```

Example 39

Project: *neo4j-social-network* Author: *bestvibes* File: *basic.py* MIT License

5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

```

```

@app.route('/')
def dump_session_contents():
    return pickle.dumps(dict(flask.session))

c = app.test_client()
c.get('/')
rv = pickle.loads(c.get('/').data)
self.assertEqual(rv['m'], flask.Markup('Hello!'))
self.assertEqual(type(rv['m']), flask.Markup)
self.assertEqual(rv['dt'], now)
self.assertEqual(rv['u'], the_uuid)
self.assertEqual(rv['b'], b'\xff')
self.assertEqual(type(rv['b']), bytes)
self.assertEqual(rv['t'], (1, 2, 3))

```

Example 40

Project: *AneMo* Author: *jspargo* File: *basic.py* GNU General Public License v2.0

5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
    self.assertEqual(rv['m'], flask.Markup('Hello!'))
    self.assertEqual(type(rv['m']), flask.Markup)
    self.assertEqual(rv['dt'], now)
    self.assertEqual(rv['u'], the_uuid)
    self.assertEqual(rv['b'], b'\xff')
    self.assertEqual(type(rv['b']), bytes)
    self.assertEqual(rv['t'], (1, 2, 3))

```

Example 41

Project: *oa_qian* Author: *sunqb* File: *basic.py* Apache License 2.0

5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

```



```

@app.after_request
def modify_session(response):
    flask.session['m'] = flask.Markup('Hello!')
    flask.session['u'] = the_uuid
    flask.session['dt'] = now
    flask.session['b'] = b'\xff'
    flask.session['t'] = (1, 2, 3)
    return response

@app.route('/')
def dump_session_contents():
    return pickle.dumps(dict(flask.session))

c = app.test_client()
c.get('/')
rv = pickle.loads(c.get('/').data)
self.assertEqual(rv['m'], flask.Markup('Hello!'))
self.assertEqual(type(rv['m']), flask.Markup)
self.assertEqual(rv['dt'], now)
self.assertEqual(rv['u'], the_uuid)
self.assertEqual(rv['b'], b'\xff')
self.assertEqual(type(rv['b']), bytes)
self.assertEqual(rv['t'], (1, 2, 3))

```

Example 42

Project: *oa_qian* Author: *sunqb* File: *widgets.py* [Apache License 2.0](#)

5 vc

```

def recaptcha_html(self, query, options):
    html = current_app.config.get('RECAPTCHA_HTML', RECAPTCHA_HTML)
    server = current_app.config.get(
        'RECAPTCHA_API_SERVER', RECAPTCHA_API_SERVER
    )
    return Markup(html % dict(
        script_url='%schallenge?%s' % (server, query),
        frame_url='%snoscript?%s' % (server, query),
        options=json.dumps(options, cls=JSONEncoder)
    ))

```

Example 43

Project: *rpki-read* Author: *rtrlib* File: *views.py* [MIT License](#)

5 vc

```

def about():
    md_file = codecs.open("../README.md", mode="r", encoding="utf-8")
    md_text = md_file.read()
    content = Markup(markdown.markdown(md_text))
    return render_template("about.html", content=content)

```

dashboard handler

Example 44

Project: *fbvoting* Author: *corrado monti* File: *beennominated.py* [MIT License](#)

5 vc

```

def retrieve_all():
    if 'notifications' in session:
        report.mark('notification-returned')
        code = '\n'.join([n.get_js_code() for n in session['notifications']])

```

```

del session['notifications']

return Markup("""
    <script>
        $('body').on( "onFacebookLoaded", function() {
            """+code+""")
        })
    </script>
""")

return ''

```

Example 45

Project: *fbvoting* Author: *corradomonti* File: [directvote.py](#) MIT License

5 vc

```

def dbdocs2html(docs, category=None):
    return Markup(
        "<table>" +
        ''.join([_dbdoc2html(d, category=category) for d in docs]) +
        "</table>"
    ) if docs else ''

```

Example 46

Project: *fbvoting* Author: *corradomonti* File: [serverutils.py](#) MIT License

5 vc

```

def refresh_token(func):

    def handle_rotten_token():
        data = fbvoting.pagebuilders.common.get_base_data()
        url = request.path
        logger.debug('A rotten token has been identified: redirecting to auth, and
        report.mark('rotten-token')
        data['already_logged'] = False
        data['js_on_loaded'] = Markup("""
            var url = new Url("%s");
            url.query.token = FB.getAuthResponse().accessToken;
            window.location.replace(url.toString());
            """ % url)
        return render_template('empty.html', **data)

    @functools.wraps(func)
    def checking_token_func(*a, **ka):
        try:
            return func(*a, **ka)
        except fbapi.FBException:
            return handle_rotten_token()
        except SSLerror:
            return handle_rotten_token()
        except urllib2.HTTPError as err:
            if err.code == 400:
                return handle_rotten_token()
            else:
                raise err

    return checking_token_func

```

Example 47

```
def bookmark_reader(bookmark_id=None):
    bookmark_id = hashids.decode(str(bookmark_id))[0]
    bookmarkobj = _get_user_object_or_404(Bookmark, bookmark_id, current_user.id)
    if bookmarkobj.readability_html is None:
        return abort(404)
    else:
        readable_html = Markup(bookmarkobj.readability_html)
        return render_template("manager/bookmark_readable.html",
                               readable_html=readable_html,
                               title=bookmarkobj.title,
                               link=bookmarkobj.main_url)
```

Example 48

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
    self.assertEqual(rv['m'], flask.Markup('Hello!'))
    self.assertEqual(type(rv['m']), flask.Markup)
    self.assertEqual(rv['dt'], now)
    self.assertEqual(rv['u'], the_uuid)
    self.assertEqual(rv['b'], b'\xff')
    self.assertEqual(type(rv['b']), bytes)
    self.assertEqual(rv['t'], (1, 2, 3))
```

Example 49

```
def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
```

```

def modify_session(response):
    flask.session['m'] = flask.Markup('Hello!')
    flask.session['u'] = the_uuid
    flask.session['dt'] = now
    flask.session['b'] = b'\xff'
    flask.session['t'] = (1, 2, 3)
    return response

@app.route('/')
def dump_session_contents():
    return pickle.dumps(dict(flask.session))

c = app.test_client()
c.get('/')
rv = pickle.loads(c.get('/').data)
self.assertEqual(rv['m'], flask.Markup('Hello!'))
self.assertEqual(type(rv['m']), flask.Markup)
self.assertEqual(rv['dt'], now)
self.assertEqual(rv['u'], the_uuid)
self.assertEqual(rv['b'], b'\xff')
self.assertEqual(type(rv['b']), bytes)
self.assertEqual(rv['t'], (1, 2, 3))

```

Example 50

Project: *noobotkit* Author: *nazroll* File: *basic.py* MIT License

5 vc

```

def test_session_special_types(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development-key'
    app.testing = True
    now = datetime.utcnow().replace(microsecond=0)
    the_uuid = uuid.uuid4()

    @app.after_request
    def modify_session(response):
        flask.session['m'] = flask.Markup('Hello!')
        flask.session['u'] = the_uuid
        flask.session['dt'] = now
        flask.session['b'] = b'\xff'
        flask.session['t'] = (1, 2, 3)
        return response

    @app.route('/')
    def dump_session_contents():
        return pickle.dumps(dict(flask.session))

    c = app.test_client()
    c.get('/')
    rv = pickle.loads(c.get('/').data)
    self.assertEqual(rv['m'], flask.Markup('Hello!'))
    self.assertEqual(type(rv['m']), flask.Markup)
    self.assertEqual(rv['dt'], now)
    self.assertEqual(rv['u'], the_uuid)
    self.assertEqual(rv['b'], b'\xff')
    self.assertEqual(type(rv['b']), bytes)
    self.assertEqual(rv['t'], (1, 2, 3))

```