

Python flask.url_for() Examples

The following are code examples for showing how to use `flask.url_for()`. 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](#) 7 vc

```
def test_dotted_names(self):
    frontend = flask.Blueprint('myapp.frontend', __name__)
    backend = flask.Blueprint('myapp.backend', __name__)

    @frontend.route('/fe')
    def frontend_index():
        return flask.url_for('myapp.backend.backend_index')

    @frontend.route('/fe2')
    def frontend_page2():
        return flask.url_for('.frontend_index')

    @backend.route('/be')
    def backend_index():
        return flask.url_for('myapp.frontend.frontend_index')

    app = flask.Flask(__name__)
    app.register_blueprint(frontend)
    app.register_blueprint(backend)

    c = app.test_client()
    self.assertEqual(c.get('/fe').data.strip(), b'/be')
    self.assertEqual(c.get('/fe2').data.strip(), b'/fe')
    self.assertEqual(c.get('/be').data.strip(), b'/fe')
```

Example 2

Project: *everclass-server* Author: *everclass* File: [views.py](#) [Mozilla Public License 2.0](#) 7 vc

```
def register():
    """注册：第一步：输入学号"""
    if request.method == 'GET':
        return render_template('user/register.html')
    else:
        if not request.form.get("xh", None): # 表单为空
            flash(MSG_EMPTY_USERNAME)
            return redirect(url_for("user.register"))

        _session_save_student_to_register_(request.form.get("xh", None))

        # 如果输入的学号已经注册，跳转到登录页面
        if User.exists(session[SESSION_STUDENT_TO_REGISTER].sid_orig):
            flash(MSG_ALREADY_REGISTERED)
            return redirect(url_for('user.login'))

        return redirect(url_for('user.register_choice'))
```

Example 3

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [helpers.py](#) [Apache License 2.0](#) 6 vc

```
def test_url_with_method(self):
    from flask.views import MethodView
    app = flask.Flask(__name__)
    class MyView(MethodView):
        def get(self, id=None):
            if id is None:
                return 'List'
            return 'Get %d' % id
        def post(self):
            return 'Create'
    myview = MyView.as_view('myview')
    app.add_url_rule('/myview/', methods=['GET'],
                     view_func=myview)
    app.add_url_rule('/myview/<int:id>', methods=['GET'],
                     view_func=myview)
    app.add_url_rule('/myview/create', methods=['POST'],
                     view_func=myview)

    with app.test_request_context():
        self.assertEqual(flask.url_for('myview', _method='GET'),
                         '/myview/')
        self.assertEqual(flask.url_for('myview', id=42, _method='GET'),
                         '/myview/42')
        self.assertEqual(flask.url_for('myview', _method='POST'),
                         '/myview/create')
```

Example 4

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#) 6 vc

```
def test_build_error_handler(self):
    app = flask.Flask(__name__)

    # Test base case, a URL which results in a BuildError.
    with app.test_request_context():
        self.assertRaises(BuildError, flask.url_for, 'spam')

    # Verify the error is re-raised if not the current exception.
    try:
        with app.test_request_context():
            flask.url_for('spam')
    except BuildError as err:
        error = err
    try:
        raise RuntimeError('Test case where BuildError is not current.')
    except RuntimeError:
        self.assertRaises(BuildError, app.handle_url_build_error, error, 'span')

    # Test a custom handler.
    def handler(error, endpoint, values):
        # Just a test.
        return '/test_handler/'
    app.url_build_error_handlers.append(handler)
    with app.test_request_context():
        self.assertEqual(flask.url_for('spam'), '/test_handler/')
```

Example 5

```
def test_inject_blueprint_url_defaults(self):
    app = flask.Flask(__name__)
    bp = flask.Blueprint('foo.bar.baz', __name__,
                        template_folder='template')

    @bp.url_defaults
    def bp_defaults(endpoint, values):
        values['page'] = 'login'
    @bp.route('/<page>')
    def view(page): pass

    app.register_blueprint(bp)

    values = dict()
    app.inject_url_defaults('foo.bar.baz.view', values)
    expected = dict(page='login')
    self.assertEqual(values, expected)

    with app.test_request_context('/somepage'):
        url = flask.url_for('foo.bar.baz.view')
        expected = '/login'
        self.assertEqual(url, expected)
```

Example 6

```
def test_route_decorator_custom_endpoint(self):
    app = flask.Flask(__name__)
    app.debug = True

    @app.route('/foo/')
    def foo():
        return flask.request.endpoint

    @app.route('/bar/', endpoint='bar')
    def for_bar():
        return flask.request.endpoint

    @app.route('/bar/123', endpoint='123')
    def for_bar_foo():
        return flask.request.endpoint

    with app.test_request_context():
        assert flask.url_for('foo') == '/foo/'
        assert flask.url_for('bar') == '/bar/'
        assert flask.url_for('123') == '/bar/123'

    c = app.test_client()
    self.assertEqual(c.get('/foo/').data, b'foo')
    self.assertEqual(c.get('/bar/').data, b'bar')
    self.assertEqual(c.get('/bar/123').data, b'123')
```

Example 7

```

def test_dotted_names(self):
    frontend = flask.Blueprint('myapp.frontend', __name__)
    backend = flask.Blueprint('myapp.backend', __name__)

    @frontend.route('/fe')
    def frontend_index():
        return flask.url_for('myapp.backend.backend_index')

    @frontend.route('/fe2')
    def frontend_page2():
        return flask.url_for('.frontend_index')

    @backend.route('/be')
    def backend_index():
        return flask.url_for('myapp.frontend.frontend_index')

    app = flask.Flask(__name__)
    app.register_blueprint(frontend)
    app.register_blueprint(backend)

    c = app.test_client()
    self.assertEqual(c.get('/fe').data.strip(), b'/be')
    self.assertEqual(c.get('/fe2').data.strip(), b'/fe')
    self.assertEqual(c.get('/be').data.strip(), b'/fe')

```

Example 8

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [blueprints.py](#) [Apache License 2.0](#) 6 vc

```

def test_dotted_names_from_app(self):
    app = flask.Flask(__name__)
    app.testing = True
    test = flask.Blueprint('test', __name__)

    @app.route('/')
    def app_index():
        return flask.url_for('test.index')

    @test.route('/test/')
    def index():
        return flask.url_for('app_index')

    app.register_blueprint(test)

    with app.test_client() as c:
        rv = c.get('/')
        self.assertEqual(rv.data, b'/test/')

```

Example 9

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [helpers.py](#) [Apache License 2.0](#) 6 vc

```

def test_url_with_method(self):
    from flask.views import MethodView
    app = flask.Flask(__name__)
    class MyView(MethodView):
        def get(self, id=None):
            if id is None:
                return 'List'
            return 'Get %d' % id

```

```

    def post(self):
        return 'Create'
myview = MyView.as_view('myview')
app.add_url_rule('/myview/', methods=['GET'],
                 view_func=myview)
app.add_url_rule('/myview/<int:id>', methods=['GET'],
                 view_func=myview)
app.add_url_rule('/myview/create', methods=['POST'],
                 view_func=myview)

with app.test_request_context():
    self.assertEqual(flask.url_for('myview', _method='GET'),
                     '/myview/')
    self.assertEqual(flask.url_for('myview', id=42, _method='GET'),
                     '/myview/42')
    self.assertEqual(flask.url_for('myview', _method='POST'),
                     '/myview/create')

```

Example 10

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

2.0

6 vc

```

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 11

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

6 vc

```

def test_build_error_handler(self):
    app = flask.Flask(__name__)

    # Test base case, a URL which results in a BuildError.
    with app.test_request_context():
        self.assertRaises(BuildError, flask.url_for, 'spam')

    # Verify the error is re-raised if not the current exception.
    try:
        with app.test_request_context():
            flask.url_for('spam')
    except BuildError as err:
        error = err

```

```

try:
    raise RuntimeError('Test case where BuildError is not current.')
except RuntimeError:
    self.assertRaises(BuildError, app.handle_url_build_error, error, 'spam')

# Test a custom handler.
def handler(error, endpoint, values):
    # Just a test.
    return '/test_handler/'
app.url_build_error_handlers.append(handler)
with app.test_request_context():
    self.assertEqual(flask.url_for('spam'), '/test_handler/')

```

Example 12

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#) 6 vc

```

def test_route_decorator_custom_endpoint(self):
    app = flask.Flask(__name__)
    app.debug = True

    @app.route('/foo/')
    def foo():
        return flask.request.endpoint

    @app.route('/bar/', endpoint='bar')
    def for_bar():
        return flask.request.endpoint

    @app.route('/bar/123', endpoint='123')
    def for_bar_foo():
        return flask.request.endpoint

    with app.test_request_context():
        assert flask.url_for('foo') == '/foo/'
        assert flask.url_for('bar') == '/bar/'
        assert flask.url_for('123') == '/bar/123'

    c = app.test_client()
    self.assertEqual(c.get('/foo/').data, b'foo')
    self.assertEqual(c.get('/bar/').data, b'bar')
    self.assertEqual(c.get('/bar/123').data, b'123')

```

Example 13

Project: *everyclass-server* Author: *everyclass* File: [views.py](#) [Mozilla Public License 2.0](#) 6 vc

```

def register_by_email():
    """注册: 第三步: 使用邮箱验证注册"""
    if not session.get(SESSION_STUDENT_TO_REGISTER, None): # 步骤异常, 跳回第一步
        return redirect(url_for('user.register'))

    sid_orig = session[SESSION_STUDENT_TO_REGISTER].sid_orig

    if User.exist(sid_orig):
        return render_template("common/error.html", message=MSG_ALREADY_REGISTERED)

    request_id = IdentityVerification.new_register_request(sid_orig, "email", ID_S

    with tracer.trace('send_email'):
        try:

```

```

        rpc_result = Auth.register_by_email(request_id, sid_orig)
    except Exception as e:
        return handle_exception_with_error_page(e)

    if rpc_result['acknowledged']:
        return render_template('user/emailSent.html', request_id=request_id)
    else:
        return render_template('common/error.html', message=MSG_INTERNAL_ERROR)

```

Example 14

Project: *pycos* Author: *xmnlab* File: *views.py* GNU General Public License v3.0

6 vc

```

def register_view(self):
    form = RegistrationForm(request.form)
    if helpers.validate_form_on_submit(form):
        user = User()

        form.populate_obj(user)
        # we hash the users password to avoid saving it as plaintext in the db
        # remove to use plain text:
        user.password = generate_password_hash(form.password.data)

        db_session.add(user)
        db_session.commit()

        login.login_user(user)
        return redirect(url_for('.index'))
    link = '<p>Already have an account? <a href="' + url_for('.login_view') +
    self._template_args['form'] = form
    self._template_args['link'] = link
    return super(PySOSAdminView, self).index()

```

Example 15

Project: *reroils-data-legacy* Author: *rero* File: *utils.py* GNU General Public License v2.0

6 vc

```

def delete_location(record_type, pid, record_indexer, parent_pid):
    """Save a record into the db and index it.

    If the location does not exists, it well be created
    and attached to the parent member.
    """
    location = Location.get_record_by_pid(pid)
    persistent_idenfier = location.persistent_identifier
    member = MemberWithLocations.get_record_by_pid(parent_pid)
    organisation = OrganisationWithMembers.get_organisation_by_memberid(
        member.id
    )
    member.remove_location(location, delindex=True)
    organisation.reindex()

    _next = url_for('invenio_records_ui.memb', pid_value=parent_pid)
    return _next, persistent_idenfier

```

Example 16

Project: *reroils-data-legacy* Author: *rero* File: *utils.py* GNU General Public License v2.0

6 vc

```
def save_location(data, record_type, fetcher, minter,
                 record_indexer, record_class, parent_pid):
    """Save a record into the db and index it.

    If the item does not exists, it well be created
    and attached to the parent document.
    """
    member = MemberWithLocations.get_record_by_pid(parent_pid)
    pid = data.get('pid')
    if pid:
        location = Location.get_record_by_pid(pid)
        location.update(data, dbcommit=True, reindex=True)
    else:
        location = Location.create(data, dbcommit=True, reindex=True)
        member.add_location(location, dbcommit=True, reindex=True)
    organisation = OrganisationWithMembers.get_organisation_by_memberid(
        member.id
    )
    organisation.reindex()

    _next = url_for('invenio_records_ui.memb', pid_value=parent_pid)
    return _next, location.persistent_identifier
```

Example 17

Project: [reroils-data-legacy](#) Author: [rero](#) File: [utils.py](#) GNU General Public License v2.0 6 vc

```
def delete_item(record_type, pid, record_indexer, parent_pid):
    """Remove an item from a document.

    The item is marked as deleted in the db, his pid as well.
    The document is reindexed.
    """
    item = Item.get_record_by_pid(pid)
    document = get_document(parent_pid, item.id)
    persistent_identifier = item.persistent_identifier
    if document:
        document.remove_item(item, delindex=True)
        _next = url_for('invenio_records_ui.doc', pid_value=document.pid)
    else:
        item.delete(delindex=True)
        _next = url_for('invenio_records_ui')
    return _next, persistent_identifier
```

Example 18

Project: [reroils-data-legacy](#) Author: [rero](#) File: [utils.py](#) GNU General Public License v2.0 6 vc

```
def save_item(data, record_type, fetcher, minter,
             record_indexer, record_class, parent_pid):
    """Save a record into the db and index it.

    If the item does not exists, it well be created
    and attached to the parent document.
    """
    item_pid = data.get('pid')
    document = None
    if item_pid:
        item = Item.get_record_by_pid(item_pid)
        document = get_document(parent_pid, item.id)
        item.update(data, dbcommit=True)
```



```

else:
    item = Item.create(data, dbcommit=True)
    document = get_document(parent_pid, item.id)
    if document:
        document.add_item(item, dbcommit=True)
if document:
    document.reindex()
if parent_pid:
    _next = url_for('invenio_records_ui.doc', pid_value=parent_pid)
else:
    _next = url_for('invenio_records_ui.item', pid_value=item.pid)
return _next, item.persistent_identifier

```

Example 19

Project: *video2commons* Author: *toolforge* File: *redisession.py* GNU General Public License v3.0

6 vc

```

def save_session(self, app, session, response):
    """Save session to Redis."""
    domain = self.get_cookie_domain(app)
    path = url_for('main', _external=False)

    if not session:
        self.redis.delete(self.prefix + session.sid)
        if session.modified:
            response.delete_cookie(app.session_cookie_name,
                                   domain=domain, path=path)
    else:
        redis_exp = self.get_redis_expiration_time(app, session)
        cookie_exp = self.get_expiration_time(app, session)
        if session.modified:
            val = self.serializer.dumps(dict(session))
            self.redis.setex(self.prefix + session.sid,
                             int(redis_exp.total_seconds()), val)
        else:
            self.redis.expire(self.prefix + session.sid,
                              int(redis_exp.total_seconds()))
        response.set_cookie(app.session_cookie_name, session.sid,
                            expires=cookie_exp, httponly=True,
                            domain=domain, path=path, secure=True)

```

Example 20

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

6 vc

```

def test_dotted_names(self):
    frontend = flask.Blueprint('myapp.frontend', __name__)
    backend = flask.Blueprint('myapp.backend', __name__)

    @frontend.route('/fe')
    def frontend_index():
        return flask.url_for('myapp.backend.backend_index')

    @frontend.route('/fe2')
    def frontend_page2():
        return flask.url_for('.frontend_index')

    @backend.route('/be')
    def backend_index():
        return flask.url_for('myapp.frontend.frontend_index')

```

```

app = flask.Flask(__name__)
app.register_blueprint(frontend)
app.register_blueprint(backend)

c = app.test_client()
self.assertEqual(c.get('/fe').data.strip(), b'/be')
self.assertEqual(c.get('/fe2').data.strip(), b'/fe')
self.assertEqual(c.get('/be').data.strip(), b'/fe')

```

Example 21

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

6 vc

```

def test_url_with_method(self):
    from flask.views import MethodView
    app = flask.Flask(__name__)
    class MyView(MethodView):
        def get(self, id=None):
            if id is None:
                return 'List'
            return 'Get %d' % id
        def post(self):
            return 'Create'
    myview = MyView.as_view('myview')
    app.add_url_rule('/myview/', methods=['GET'],
                     view_func=myview)
    app.add_url_rule('/myview/<int:id>', methods=['GET'],
                     view_func=myview)
    app.add_url_rule('/myview/create', methods=['POST'],
                     view_func=myview)

    with app.test_request_context():
        self.assertEqual(flask.url_for('myview', _method='GET'),
                         '/myview/')
        self.assertEqual(flask.url_for('myview', id=42, _method='GET'),
                         '/myview/42')
        self.assertEqual(flask.url_for('myview', _method='POST'),
                         '/myview/create')

```

Example 22

Project: *flask-template* Author: *pwgraham91* File: [login_view.py](#) MIT License

5 vc

```

def dev_login(user_id):
    if ENVIRONMENT == 'dev':
        login_user(db.session.query(User).get(user_id))
    return redirect(url_for('index'))

```

Example 23

Project: *flask-template* Author: *pwgraham91* File: [login_view.py](#) MIT License

5 vc

```

def logout():
    logout_user()
    return flask.redirect(flask.url_for('index'))

```

Example 24

Project: *flaskit* Author: *elston* File: [views.py](#) MIT License

5 vc

```
def login():
    # ..
    form = LoginForm(request.form)
    if request.method == 'POST':
        if form.validate_on_submit():
            login_user(form.user)
            redirect_url = request.args.get('next') or url_for('admin.index')
            return redirect(redirect_url)
        else:
            flash_errors(form)
    # ..
    return render_template('accounts/login.html', form=form)
```

Example 25

Project: *flaskit* Author: *elston* File: [views.py](#) MIT License

5 vc

```
def logout():
    logout_user()
    flash('You are logged out.', 'info')
    return redirect(url_for('admin.index'))
```

Example 26

Project: *Mastering-Python-Networking-Second-Edition* Author: *PacktPublishing*

File: [chapter9_6.py](#) MIT License

5 vc

```
def get_url(self):
    return url_for('get_device', id=self.id, _external=True)
```

Example 27

Project: *Mastering-Python-Networking-Second-Edition* Author: *PacktPublishing*

File: [chapter9_7.py](#) MIT License

5 vc

```
def get_url(self):
    return url_for('get_device', id=self.id, _external=True)
```

Example 28

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

2.0

5 vc

```
def test_default_endpoint_name(self):
    app = flask.Flask(__name__)
    mod = flask.Module(__name__, 'frontend')
    def index():
        return 'Awesome'
    mod.add_url_rule('/', view_func=index)
    app.register_module(mod)
    rv = app.test_client().get('/')
    self.assertEqual(rv.data, b'Awesome')
    with app.test_request_context():
        self.assertEqual(flask.url_for('frontend.index'), '/')
```

Example 29

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

5 vc

```

def test_templates_and_static(self):
    app = moduleapp
    app.testing = True
    c = app.test_client()

    rv = c.get('/')
    self.assertEqual(rv.data, b'Hello from the Frontend')
    rv = c.get('/admin/')
    self.assertEqual(rv.data, b'Hello from the Admin')
    rv = c.get('/admin/index2')
    self.assertEqual(rv.data, b'Hello from the Admin')
    rv = c.get('/admin/static/test.txt')
    self.assertEqual(rv.data.strip(), b'Admin File')
    rv.close()
    rv = c.get('/admin/static/css/test.css')
    self.assertEqual(rv.data.strip(), b'/* nested file */')
    rv.close()

    with app.test_request_context():
        self.assertEqual(flask.url_for('admin.static', filename='test.txt'),
                        '/admin/static/test.txt')

    with app.test_request_context():
        try:
            flask.render_template('missing.html')
        except TemplateNotFound as e:
            self.assertEqual(e.name, 'missing.html')
        else:
            self.assert_true(0, 'expected exception')

    with flask.Flask(__name__).test_request_context():
        self.assertEqual(flask.render_template('nested/nested.txt'), 'I\'m ne

```

Example 30

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

```

def test_blueprint_url_processors(self):
    bp = flask.Blueprint('frontend', __name__, url_prefix='<lang_code>')

    @bp.url_defaults
    def add_language_code(endpoint, values):
        values.setdefault('lang_code', flask.g.lang_code)

    @bp.url_value_preprocessor
    def pull_lang_code(endpoint, values):
        flask.g.lang_code = values.pop('lang_code')

    @bp.route('/')
    def index():
        return flask.url_for('.about')

    @bp.route('/about')
    def about():
        return flask.url_for('.index')

    app = flask.Flask(__name__)
    app.register_blueprint(bp)

```

```

c = app.test_client()

self.assertEqual(c.get('/de/').data, b'/de/about')
self.assertEqual(c.get('/de/about').data, b'/de/')

```

Example 31

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

[5 vc](#)

```

def test_url_for_with_anchor(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return '42'
    with app.test_request_context():
        self.assertEqual(flask.url_for('index', _anchor='x y'),
                        '/#x%20y')

```

Example 32

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

[5 vc](#)

```

def test_url_for_with_scheme(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return '42'
    with app.test_request_context():
        self.assertEqual(flask.url_for('index',
                                      _external=True,
                                      _scheme='https'),
                        'https://localhost/')

```

Example 33

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

[5 vc](#)

```

def test_url_for_with_scheme_not_external(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return '42'
    with app.test_request_context():
        self.assertRaises(ValueError,
                          flask.url_for,
                          'index',
                          _scheme='https')

```

Example 34

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

[5 vc](#)

```

def test_subdomain(self):
    @self.app.route('/', subdomain='<company_id>')
    def view(company_id):
        return company_id

    url = flask.url_for('view', company_id='xxx')
    response = self.client.get(url)

```

```
self.assert_equal(200, response.status_code)
self.assert_equal(b'xxx', response.data)
```

Example 35

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [testing.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_nosubdomain(self):
    @self.app.route('/<company_id>')
    def view(company_id):
        return company_id

    url = flask.url_for('view', company_id='xxx')
    response = self.client.get(url)

    self.assert_equal(200, response.status_code)
    self.assert_equal(b'xxx', response.data)
```

Example 36

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_url_generation(self):
    app = flask.Flask(__name__)
    @app.route('/hello/<name>', methods=['POST'])
    def hello():
        pass
    with app.test_request_context():
        self.assert_equal(flask.url_for('hello', name='test x'), '/hello/test
        self.assert_equal(flask.url_for('hello', name='test x', _external=True
        'http://localhost/hello/test%20x')
```

Example 37

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_static_files(self):
    app = flask.Flask(__name__)
    app.testing = True
    rv = app.test_client().get('/static/index.html')
    self.assert_equal(rv.status_code, 200)
    self.assert_equal(rv.data.strip(), b'<h1>Hello World!</h1>')
    with app.test_request_context():
        self.assert_equal(flask.url_for('static', filename='index.html'),
        '/static/index.html')
    rv.close()
```

Example 38

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [appctx.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_basic_url_generation(self):
    app = flask.Flask(__name__)
    app.config['SERVER_NAME'] = 'localhost'
    app.config['PREFERRED_URL_SCHEME'] = 'https'

    @app.route('/')
    def index():
        pass
```

```

with app.app_context():
    rv = flask.url_for('index')
    self.assertEqual(rv, 'https://localhost/')

```

Example 39

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [appctx.py](#) [Apache License 2.0](#) 5 vc

```

def test_url_generation_requires_server_name(self):
    app = flask.Flask(__name__)
    with app.app_context():
        with self.assertRaises(RuntimeError):
            flask.url_for('index')

```

Example 40

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [appctx.py](#) [Apache License 2.0](#) 5 vc

```

def test_url_generation_without_context_fails(self):
    with self.assertRaises(RuntimeError):
        flask.url_for('index')

```

Example 41

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [blueprints.py](#) [Apache License 2.0](#) 5 vc

```

def test_templates_and_static(self):
    app = moduleapp
    app.testing = True
    c = app.test_client()

    rv = c.get('/')
    self.assertEqual(rv.data, b'Hello from the Frontend')
    rv = c.get('/admin/')
    self.assertEqual(rv.data, b'Hello from the Admin')
    rv = c.get('/admin/index2')
    self.assertEqual(rv.data, b'Hello from the Admin')
    rv = c.get('/admin/static/test.txt')
    self.assertEqual(rv.data.strip(), b'Admin File')
    rv.close()
    rv = c.get('/admin/static/css/test.css')
    self.assertEqual(rv.data.strip(), b'/* nested file */')
    rv.close()

    with app.test_request_context():
        self.assertEqual(flask.url_for('admin.static', filename='test.txt'),
                         '/admin/static/test.txt')

    with app.test_request_context():
        try:
            flask.render_template('missing.html')
        except TemplateNotFound as e:
            self.assertEqual(e.name, 'missing.html')
        else:
            self.assert_true(0, 'expected exception')

    with flask.Flask(__name__).test_request_context():
        self.assertEqual(flask.render_template('nested/nested.txt'), 'I\'m ne

```

Example 42

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

2.0

5 vc

```
def test_blueprint_url_processors(self):
    bp = flask.Blueprint('frontend', __name__, url_prefix='<lang_code>')

    @bp.url_defaults
    def add_language_code(endpoint, values):
        values.setdefault('lang_code', flask.g.lang_code)

    @bp.url_value_preprocessor
    def pull_lang_code(endpoint, values):
        flask.g.lang_code = values.pop('lang_code')

    @bp.route('/')
    def index():
        return flask.url_for('.about')

    @bp.route('/about')
    def about():
        return flask.url_for('.index')

    app = flask.Flask(__name__)
    app.register_blueprint(bp)

    c = app.test_client()

    self.assertEqual(c.get('/de/').data, b'/de/about')
    self.assertEqual(c.get('/de/about').data, b'/de/')
```

Example 43

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

5 vc

```
def test_url_for_with_scheme(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return '42'
    with app.test_request_context():
        self.assertEqual(flask.url_for('index',
                                       _external=True,
                                       _scheme='https'),
                         'https://localhost/')
```

Example 44

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

5 vc

```
def test_url_for_with_scheme_not_external(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return '42'
    with app.test_request_context():
        self.assertRaises(ValueError,
                          flask.url_for,
                          'index',
                          _scheme='https')
```


Example 45

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [testing.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_nosubdomain(self):
    @self.app.route('/<company_id>')
    def view(company_id):
        return company_id

    url = flask.url_for('view', company_id='xxx')
    response = self.client.get(url)

    self.assertEqual(200, response.status_code)
    self.assertEqual(b'xxx', response.data)
```

Example 46

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_url_generation(self):
    app = flask.Flask(__name__)
    @app.route('/hello/<name>', methods=['POST'])
    def hello():
        pass
    with app.test_request_context():
        self.assertEqual(flask.url_for('hello', name='test x'), '/hello/test
        self.assertEqual(flask.url_for('hello', name='test x', _external=True
        'http://localhost/hello/test%20x')
```

Example 47

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_static_files(self):
    app = flask.Flask(__name__)
    app.testing = True
    rv = app.test_client().get('/static/index.html')
    self.assertEqual(rv.status_code, 200)
    self.assertEqual(rv.data.strip(), b'<h1>Hello World!</h1>')
    with app.test_request_context():
        self.assertEqual(flask.url_for('static', filename='index.html'),
        '/static/index.html')
    rv.close()
```

Example 48

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [basic.py](#) [Apache License 2.0](#) [5 vc](#)

```
def test_url_processors(self):
    app = flask.Flask(__name__)

    @app.url_defaults
    def add_language_code(endpoint, values):
        if flask.g.lang_code is not None and \
            app.url_map.is_endpoint_expectng(endpoint, 'lang_code'):
            values.setdefault('lang_code', flask.g.lang_code)

    @app.url_value_preprocessor
    def pull_lang_code(endpoint, values):
```

```

        flask.g.lang_code = values.pop('lang_code', None)

@app.route('/<lang_code>/')
def index():
    return flask.url_for('about')

@app.route('/<lang_code>/about')
def about():
    return flask.url_for('something_else')

@app.route('/foo')
def something_else():
    return flask.url_for('about', lang_code='en')

c = app.test_client()

self.assertEqual(c.get('/de/').data, b'/de/about')
self.assertEqual(c.get('/de/about').data, b'/foo')
self.assertEqual(c.get('/foo').data, b'/en/about')

```

Example 49

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [appctx.py](#) [Apache License 2.0](#) [5 vc](#)

```

def test_basic_url_generation(self):
    app = flask.Flask(__name__)
    app.config['SERVER_NAME'] = 'localhost'
    app.config['PREFERRED_URL_SCHEME'] = 'https'

    @app.route('/')
    def index():
        pass

    with app.app_context():
        rv = flask.url_for('index')
        self.assertEqual(rv, 'https://localhost/')

```

Example 50

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: [appctx.py](#) [Apache License 2.0](#) [5 vc](#)

```

def test_url_generation_requires_server_name(self):
    app = flask.Flask(__name__)
    with app.app_context():
        with self.assertRaises(RuntimeError):
            flask.url_for('index')

```