

Python `flask.Flask()` Examples

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

Example 1

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

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

7 vc

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

    class Index(flask.views.MethodView):
        def get(self):
            return flask.Response('Blub', headers={
                'X-Method': flask.request.method
            })

    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assertEqual(rv.data, b'Blub')
    self.assertEqual(rv.headers['X-Method'], 'GET')
    rv = c.head('/')
    self.assertEqual(rv.data, b'')
    self.assertEqual(rv.headers['X-Method'], 'HEAD')
```

Example 3

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

    @app.route('/')
    def index():
        return flask.render_template('simple_template.html', whiskey=42)

    recorded = []
    def record(sender, template, context):
        recorded.append((template, context))

    flask.template_rendered.connect(record, app)
    try:
        app.test_client().get('/')
        self.assertEqual(len(recorded), 1)
        template, context = recorded[0]
        self.assertEqual(template.name, 'simple_template.html')
        self.assertEqual(context['whiskey'], 42)
    finally:
        flask.template_rendered.disconnect(record, app)
```

Example 4

```
def test_request_exception_signal(self):
    app = flask.Flask(__name__)
    recorded = []

    @app.route('/')
    def index():
        1 // 0

    def record(sender, exception):
        recorded.append(exception)

    flask.request_exception.connect(record, app)
    try:
        self.assertEqual(app.test_client().get('/').status_code, 500)
        self.assertEqual(len(recorded), 1)
        self.assertTrue(isinstance(recorded[0], ZeroDivisionError))
    finally:
        flask.request_exception.disconnect(record, app)
```

Example 5

```
def test_appcontext_signals(self):
    app = flask.Flask(__name__)
    recorded = []
    def record_push(sender, **kwargs):
        recorded.append('push')
    def record_pop(sender, **kwargs):
        recorded.append('push')

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

```

flask.appcontext_pushed.connect(record_push, app)
flask.appcontext_popped.connect(record_pop, app)
try:
    with app.test_client() as c:
        rv = c.get('/')
        self.assertEqual(rv.data, b'Hello')
        self.assertEqual(recorded, ['push'])
        self.assertEqual(recorded, ['push', 'pop'])
finally:
    flask.appcontext_pushed.disconnect(record_push, app)
    flask.appcontext_popped.disconnect(record_pop, app)

```

Example 6

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

```

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.assertEqual(len(recorded), 1)
            message, category = recorded[0]
            self.assertEqual(message, 'This is a flash message')
            self.assertEqual(category, 'notice')
    finally:
        flask.message_flashed.disconnect(record, app)

```

Example 7

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

```

def test_basic_module(self):
    app = flask.Flask(__name__)
    admin = flask.Module(__name__, 'admin', url_prefix='/admin')
    @admin.route('/')
    def admin_index():
        return 'admin index'
    @admin.route('/login')
    def admin_login():
        return 'admin login'
    @admin.route('/logout')
    def admin_logout():
        return 'admin logout'
    @app.route('/')
    def index():

```

```

        return 'the index'
    app.register_module(admin)
    c = app.test_client()
    self.assertEqual(c.get('/').data, b'the index')
    self.assertEqual(c.get('/admin/').data, b'admin index')
    self.assertEqual(c.get('/admin/login').data, b'admin login')
    self.assertEqual(c.get('/admin/logout').data, b'admin logout')

```

Example 8

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

6 vc

2.0

```

def test_context_processors(self):
    app = flask.Flask(__name__)
    admin = flask.Module(__name__, 'admin', url_prefix='/admin')
    @app.context_processor
    def inject_all_regular():
        return {'a': 1}
    @admin.context_processor
    def inject_admin():
        return {'b': 2}
    @admin.app_context_processor
    def inject_all_module():
        return {'c': 3}
    @app.route('/')
    def index():
        return flask.render_template_string('{{ a }}{{ b }}{{ c }}')
    @admin.route('/')
    def admin_index():
        return flask.render_template_string('{{ a }}{{ b }}{{ c }}')
    app.register_module(admin)
    c = app.test_client()
    self.assertEqual(c.get('/').data, b'13')
    self.assertEqual(c.get('/admin/').data, b'123')

```

Example 9

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

6 vc

2.0

```

def test_endpoint_decorator(self):
    from werkzeug.routing import Submount, Rule
    from flask import Module

    app = flask.Flask(__name__)
    app.testing = True
    app.url_map.add(Submount('/foo', [
        Rule('/bar', endpoint='bar'),
        Rule('/', endpoint='index')
    ]))
    module = Module(__name__, __name__)

    @module.endpoint('bar')
    def bar():
        return 'bar'

    @module.endpoint('index')
    def index():
        return 'index'

```

```

app.register_module(module)

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

```

Example 10

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

6 vc

2.0

```

def test_blueprint_url_definitions(self):
    bp = flask.Blueprint('test', __name__)

    @bp.route('/foo', defaults={'baz': 42})
    def foo(bar, baz):
        return '%s/%d' % (bar, baz)

    @bp.route('/bar')
    def bar(bar):
        return text_type(bar)

    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/1', url_defaults={'bar': 23})
    app.register_blueprint(bp, url_prefix='/2', url_defaults={'bar': 19})

    c = app.test_client()
    self.assertEqual(c.get('/1/foo').data, b'23/42')
    self.assertEqual(c.get('/2/foo').data, b'19/42')
    self.assertEqual(c.get('/1/bar').data, b'23')
    self.assertEqual(c.get('/2/bar').data, b'19')

```

Example 11

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

6 vc

2.0

```

def test_default_static_cache_timeout(self):
    app = flask.Flask(__name__)
    class MyBlueprint(flask.Blueprint):
        def get_send_file_max_age(self, filename):
            return 100

    blueprint = MyBlueprint('blueprint', __name__, static_folder='static')
    app.register_blueprint(blueprint)

    # try/finally, in case other tests use this app for Blueprint tests.
    max_age_default = app.config['SEND_FILE_MAX_AGE_DEFAULT']
    try:
        with app.test_request_context():
            unexpected_max_age = 3600
            if app.config['SEND_FILE_MAX_AGE_DEFAULT'] == unexpected_max_age:
                unexpected_max_age = 7200
            app.config['SEND_FILE_MAX_AGE_DEFAULT'] = unexpected_max_age
            rv = blueprint.send_static_file('index.html')
            cc = parse_cache_control_header(rv.headers['Cache-Control'])
            self.assertEqual(cc.max_age, 100)
            rv.close()
    finally:
        app.config['SEND_FILE_MAX_AGE_DEFAULT'] = max_age_default

```

Example 12

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 13

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 14

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

6 vc

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

    class Index(flask.views.MethodView):
        def get(self):
            1 // 0
        def post(self):
            1 // 0

    class Other(Index):
```

```

    def get(self):
        return 'GET'
    def post(self):
        return 'POST'

view = Index.as_view('index')
view.view_class = Other
app.add_url_rule('/', view_func=view)
self.common_test(app)

```

Example 15

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

6 vc

```

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

    def add_x_parachute(f):
        def new_function(*args, **kwargs):
            resp = flask.make_response(f(*args, **kwargs))
            resp.headers['X-Parachute'] = 'awesome'
            return resp
        return new_function

    class Index(flask.views.View):
        decorators = [add_x_parachute]
        def dispatch_request(self):
            return 'Awesome'

    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assertEqual(rv.headers['X-Parachute'], 'awesome')
    self.assertEqual(rv.data, b'Awesome')

```

Example 16

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

6 vc

```

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

    class Index(flask.views.MethodView):
        def get(self):
            return 'GET'
        def head(self):
            return flask.Response('', headers={'X-Method': 'HEAD'})

    app.add_url_rule('/', view_func=Index.as_view('index'))
    c = app.test_client()
    rv = c.get('/')
    self.assertEqual(rv.data, b'GET')
    rv = c.head('/')
    self.assertEqual(rv.data, b'')
    self.assertEqual(rv.headers['X-Method'], 'HEAD')

```

Example 17

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

6 vc

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

    class Index(flask.views.View):
        methods = ['GET', 'POST']
        def dispatch_request(self):
            return flask.request.method

    app.add_url_rule('/', view_func=Index.as_view('index'))

    with self.assert_raises(AssertionError):
        app.add_url_rule('/', view_func=Index.as_view('index'))

    # But these tests should still pass. We just log a warning.
    self.common_test(app)
```

Example 18

Project: *invenio-openaire* Author: *inveniosoftware* File: [confest.py](#) MIT License

5 vc

```
def app(request):
    """Flask application fixture."""
    # Set temporary instance path for sqlite
    instance_path = tempfile.mkdtemp()
    app = Flask('testapp', instance_path=instance_path)
    app.config.update(
        SQLALCHEMY_DATABASE_URI=os.environ.get(
            'SQLALCHEMY_DATABASE_URI', 'sqlite:///test.db'),
        INDEXER_REPLACE_REFS=True,
        CELERY_ALWAYS_EAGER=True,
        CELERY_RESULT_BACKEND="cache",
        CELERY_CACHE_BACKEND="memory",
        CELERY_EAGER_PROPAGATES_EXCEPTIONS=True,
        JSONSCHEMAS_HOST='inveniosoftware.org',
        OPENAIRE_OAI_LOCAL_SOURCE='invenio_openaire/data/oaire_local.sqlite',
        TESTING=True,
    )

    app.url_map.converters['pid'] = PIDConverter
    app.url_map.converters['pidpath'] = PIDPathConverter

    LoginManager(app)
    InvenioDB(app)
    InvenioIndexer(app)
    InvenioRecords(app)
    InvenioCelery(app)
    InvenioPIDStore(app)
    InvenioOpenAIRE(app)
    InvenioSearch(app)
    InvenioJSONSchemas(app)

    with app.app_context():
        yield app

    shutil.rmtree(instance_path)
```

Example 19

Project: *invenio-openaire* Author: *inveniosoftware* File: [test_invenio_openaire.py](#) MIT License

5 vc


```
def test_init():
    """Test extension initialization."""
    app = Flask('testapp')
    ext = InvenioOpenAIRE(app)
    assert 'invenio-openaire' in app.extensions

    app = Flask('testapp')
    ext = InvenioOpenAIRE()
    assert 'invenio-openaire' not in app.extensions
    ext.init_app(app)
    assert 'invenio-openaire' in app.extensions
```

Example 20

Project: *hydrus* Author: *HTTP-APIs* File: [app_factory.py](#) MIT License

5 vc

```
def app_factory(api_name: str = "api") -> Flask:
    """Create an app object."""

    app = Flask(__name__)
    app.config['SECRET_KEY'] = 'secret key'
    CORS(app)
    app.url_map.strict_slashes = False
    api = Api(app)

    api.add_resource(Index, "{}/".format(api_name), endpoint="api")
    api.add_resource(Vocab, "{}/vocab".format(api_name), endpoint="vocab")
    api.add_resource(
        Contexts,
        "{}/contexts/<string:category>.jsonld".format(api_name),
        endpoint="contexts")
    api.add_resource(
        EntryPoint,
        "{}/contexts/EntryPoint.jsonld".format(api_name),
        endpoint="main_entrypoint")
    api.add_resource(
        ItemCollection,
        "{}/<string:path>".format(api_name),
        endpoint="item_collection")
    api.add_resource(
        Item,
        "{}/<string:path>/<uuid:id>".format(api_name),
        endpoint="item")
    api.add_resource(
        Items,
        "{}/<string:path>/add/<int_list>".format(api_name),
        "{}/<string:path>/add".format(api_name),
        "{}/<string:path>/delete/<int_list>".format(api_name))

    return app
```

Example 21

Project: *hydrus* Author: *HTTP-APIs* File: [socketio_factory.py](#) MIT License

5 vc

```
def create_socketio(app: Flask, session: scoped_session) -> SocketIO:
    socketio.init_app(app, logger=True)
    socketio.on_namespace(SyncNamespace(namespace='/sync', db_session=session))
    return socketio
```

Example 22

```
def create_app():
    flask_sample = Flask('flask_ci_test')

    flask_sample.register_blueprint(users_views.blueprint)

    return flask_sample
```

Example 23

```
def test_request_signals(self):
    app = flask.Flask(__name__)
    calls = []

    def before_request_signal(sender):
        calls.append('before-signal')

    def after_request_signal(sender, response):
        self.assertEqual(response.data, b'stuff')
        calls.append('after-signal')

    @app.before_request
    def before_request_handler():
        calls.append('before-handler')

    @app.after_request
    def after_request_handler(response):
        calls.append('after-handler')
        response.data = 'stuff'
        return response

    @app.route('/')
    def index():
        calls.append('handler')
        return 'ignored anyway'

    flask.request_started.connect(before_request_signal, app)
    flask.request_finished.connect(after_request_signal, app)

    try:
        rv = app.test_client().get('/')
        self.assertEqual(rv.data, b'stuff')

        self.assertEqual(calls, ['before-signal', 'before-handler',
                                'handler', 'after-handler',
                                'after-signal'])
    finally:
        flask.request_started.disconnect(before_request_signal, app)
        flask.request_finished.disconnect(after_request_signal, app)
```

Example 24

```
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 25

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

5 vc

2.0

```

def test_request_processing(self):
    caught = []
    app = flask.Flask(__name__)
    admin = flask.Module(__name__, 'admin', url_prefix='/admin')
    @admin.before_request
    def before_admin_request():
        caught.append('before-admin')
    @admin.after_request
    def after_admin_request(response):
        caught.append('after-admin')
        return response
    @admin.route('/')
    def admin_index():
        return 'the admin'
    @app.before_request
    def before_request():
        caught.append('before-app')
    @app.after_request
    def after_request(response):
        caught.append('after-app')
        return response
    @app.route('/')
    def index():
        return 'the index'
    app.register_module(admin)
    c = app.test_client()

    self.assertEqual(c.get('/').data, b'the index')
    self.assertEqual(caught, ['before-app', 'after-app'])
    del caught[:]

    self.assertEqual(c.get('/admin/').data, b'the admin')
    self.assertEqual(caught, ['before-app', 'before-admin',
                              'after-admin', 'after-app'])

```

Example 26

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

5 vc

2.0

```

def test_late_binding(self):
    app = flask.Flask(__name__)
    admin = flask.Module(__name__, 'admin')
    @admin.route('/')
    def index():
        return '42'

```

```
app.register_module(admin, url_prefix='/admin')
self.assertEqual(app.test_client().get('/admin/').data, b'42')
```

Example 27

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 28

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

2.0

5 vc

```
def test_blueprint_specific_error_handling(self):
    frontend = flask.Blueprint('frontend', __name__)
    backend = flask.Blueprint('backend', __name__)
    sideend = flask.Blueprint('sideend', __name__)

    @frontend.errorhandler(403)
    def frontend_forbidden(e):
        return 'frontend says no', 403

    @frontend.route('/frontend-no')
    def frontend_no():
        flask.abort(403)

    @backend.errorhandler(403)
    def backend_forbidden(e):
```

```

        return 'backend says no', 403

@backend.route('/backend-no')
def backend_no():
    flask.abort(403)

@sideend.route('/what-is-a-sideend')
def sideend_no():
    flask.abort(403)

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

@app.errorhandler(403)
def app_forbidden(e):
    return 'application itself says no', 403

c = app.test_client()

self.assertEqual(c.get('/frontend-no').data, b'frontend says no')
self.assertEqual(c.get('/backend-no').data, b'backend says no')
self.assertEqual(c.get('/what-is-a-sideend').data, b'application itself s

```

Example 29

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 30

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

5 vc

```
def test_empty_url_defaults(self):
    bp = flask.Blueprint('bp', __name__)

    @bp.route('/', defaults={'page': 1})
    @bp.route('/page/<int:page>')
    def something(page):
        return str(page)

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

    c = app.test_client()
    self.assertEqual(c.get('/').data, b'1')
    self.assertEqual(c.get('/page/2').data, b'2')
```

Example 31

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

2.0

5 vc

```
def test_route_decorator_custom_endpoint(self):

    bp = flask.Blueprint('bp', __name__)

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

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

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

    @bp.route('/bar/foo')
    def bar_foo():
        return flask.request.endpoint

    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')

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

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

Example 32

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

2.0

5 vc

```
def test_template_filter(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter()
    def my_reverse(s):
        return s[::-1]
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('my_reverse', app.jinja_env.filters.keys())
    self.assertEqual(app.jinja_env.filters['my_reverse'], my_reverse)
    self.assertEqual(app.jinja_env.filters['my_reverse']('abcd'), 'dcba')
```

Example 33

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

2.0

5 vc

```
def test_add_template_filter(self):
    bp = flask.Blueprint('bp', __name__)
    def my_reverse(s):
        return s[::-1]
    bp.add_app_template_filter(my_reverse)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('my_reverse', app.jinja_env.filters.keys())
    self.assertEqual(app.jinja_env.filters['my_reverse'], my_reverse)
    self.assertEqual(app.jinja_env.filters['my_reverse']('abcd'), 'dcba')
```

Example 34

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

2.0

5 vc

```
def test_template_filter_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter('strrev')
    def my_reverse(s):
        return s[::-1]
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('strrev', app.jinja_env.filters.keys())
    self.assertEqual(app.jinja_env.filters['strrev'], my_reverse)
    self.assertEqual(app.jinja_env.filters['strrev']('abcd'), 'dcba')
```

Example 35

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

2.0

5 vc

```
def test_add_template_filter_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    def my_reverse(s):
        return s[::-1]
    bp.add_app_template_filter(my_reverse, 'strrev')
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('strrev', app.jinja_env.filters.keys())
    self.assertEqual(app.jinja_env.filters['strrev'], my_reverse)
    self.assertEqual(app.jinja_env.filters['strrev']('abcd'), 'dcba')
```

Example 36

2.0

```
def test_template_filter_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter()
    def super_reverse(s):
        return s[::-1]
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    @app.route('/')
    def index():
        return flask.render_template('template_filter.html', value='abcd')
    rv = app.test_client().get('/')
    self.assertEqual(rv.data, b'dcba')
```

Example 37

2.0

```
def test_add_template_filter_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    def super_reverse(s):
        return s[::-1]
    bp.add_app_template_filter(super_reverse)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    @app.route('/')
    def index():
        return flask.render_template('template_filter.html', value='abcd')
    rv = app.test_client().get('/')
    self.assertEqual(rv.data, b'dcba')
```

Example 38

2.0

```
def test_template_filter_with_name_and_template(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_filter('super_reverse')
    def my_reverse(s):
        return s[::-1]
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    @app.route('/')
    def index():
        return flask.render_template('template_filter.html', value='abcd')
    rv = app.test_client().get('/')
    self.assertEqual(rv.data, b'dcba')
```

Example 39

2.0

```
def test_add_template_filter_with_name_and_template(self):
    bp = flask.Blueprint('bp', __name__)
```



```

def my_reverse(s):
    return s[::-1]
bp.add_app_template_filter(my_reverse, 'super_reverse')
app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')
@app.route('/')
def index():
    return flask.render_template('template_filter.html', value='abcd')
rv = app.test_client().get('/')
self.assertEqual(rv.data, b'dcba')

```

Example 40

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

5 vc

2.0

```

def test_template_test(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_test()
    def is_boolean(value):
        return isinstance(value, bool)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('is_boolean', app.jinja_env.tests.keys())
    self.assertEqual(app.jinja_env.tests['is_boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['is_boolean'](False))

```

Example 41

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

5 vc

2.0

```

def test_add_template_test(self):
    bp = flask.Blueprint('bp', __name__)
    def is_boolean(value):
        return isinstance(value, bool)
    bp.add_app_template_test(is_boolean)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('is_boolean', app.jinja_env.tests.keys())
    self.assertEqual(app.jinja_env.tests['is_boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['is_boolean'](False))

```

Example 42

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

5 vc

2.0

```

def test_add_template_test_with_name(self):
    bp = flask.Blueprint('bp', __name__)
    def is_boolean(value):
        return isinstance(value, bool)
    bp.add_app_template_test(is_boolean, 'boolean')
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    self.assert_in('boolean', app.jinja_env.tests.keys())
    self.assertEqual(app.jinja_env.tests['boolean'], is_boolean)
    self.assert_true(app.jinja_env.tests['boolean'](False))

```

Example 43

2.0

```
def test_template_test_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_test()
    def boolean(value):
        return isinstance(value, bool)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    @app.route('/')
    def index():
        return flask.render_template('template_test.html', value=False)
    rv = app.test_client().get('/')
    self.assert_in(b'Success!', rv.data)
```

Example 44

2.0

```
def test_template_test_after_route_with_template(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template('template_test.html', value=False)
    bp = flask.Blueprint('bp', __name__)
    @bp.app_template_test()
    def boolean(value):
        return isinstance(value, bool)
    app.register_blueprint(bp, url_prefix='/py')
    rv = app.test_client().get('/')
    self.assert_in(b'Success!', rv.data)
```

Example 45

2.0

```
def test_add_template_test_with_template(self):
    bp = flask.Blueprint('bp', __name__)
    def boolean(value):
        return isinstance(value, bool)
    bp.add_app_template_test(boolean)
    app = flask.Flask(__name__)
    app.register_blueprint(bp, url_prefix='/py')
    @app.route('/')
    def index():
        return flask.render_template('template_test.html', value=False)
    rv = app.test_client().get('/')
    self.assert_in(b'Success!', rv.data)
```

Example 46

2.0

```
def test_template_test_with_name_and_template(self):
    bp = flask.Blueprint('bp', __name__)
```

```

@bp.app_template_test('boolean')
def is_boolean(value):
    return isinstance(value, bool)
app = flask.Flask(__name__)
app.register_blueprint(bp, url_prefix='/py')
@app.route('/')
def index():
    return flask.render_template('template_test.html', value=False)
rv = app.test_client().get('/')
self.assert_in(b'Success!', rv.data)

```

Example 47

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

2.0

5 vc

```

def test_context_processing(self):
    app = flask.Flask(__name__)
    @app.context_processor
    def context_processor():
        return {'injected_value': 42}
    @app.route('/')
    def index():
        return flask.render_template('context_template.html', value=23)
    rv = app.test_client().get('/')
    self.assertEqual(rv.data, b'<p>23|42')

```

Example 48

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

2.0

5 vc

```

def test_original_win(self):
    app = flask.Flask(__name__)
    @app.route('/')
    def index():
        return flask.render_template_string('{{ config }}', config=42)
    rv = app.test_client().get('/')
    self.assertEqual(rv.data, b'42')

```

Example 49

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

2.0

5 vc

```

def test_request_less_rendering(self):
    app = flask.Flask(__name__)
    app.config['WORLD_NAME'] = 'Special World'
    @app.context_processor
    def context_processor():
        return dict(foo=42)

    with app.app_context():
        rv = flask.render_template_string('Hello {{ config.WORLD_NAME }} '
                                          '{{{ foo }}}')
        self.assertEqual(rv, 'Hello Special World 42')

```

Example 50

[2.0](#)

```
def test_standard_context(self):
    app = flask.Flask(__name__)
    app.secret_key = 'development key'
    @app.route('/')
    def index():
        flask.g.foo = 23
        flask.session['test'] = 'aha'
        return flask.render_template_string('''
            {{ request.args.foo }}
            {{ g.foo }}
            {{ config.DEBUG }}
            {{ session.test }}
        ''')
    rv = app.test_client().get('/?foo=42')
    self.assertEqual(rv.data.split(), [b'42', b'23', b'False', b'aha'])
```