Less is better?

Igor Davydenko
Kyiv.py#11

Django...

Django Dash

- Annual competition, started in 2008
- 48 hours from idea to result
- Team of <= 3 members
- Django is required
- Very fun to participate

Django Dashers

- **Read the Docs** (Django Dash 2010)
- Any other working and good projects?
- I tried to be Dasher 2 times, but without success

The main problem

- Django Dash idea needs more than 48 hours
- Or idea is not good at all
- Sometimes you don’t need Django to build web-application :)

WSGI...

WSGI Wrestle

- Competition started in 2013
- Same rules as Django dash instead of Django

WSGI Wrestle 2013

- Let’s build Google Reader in 48 hours
- Why it’s not real?

Google Reader tasks

- OAuth login (Google, Twitter, Facebook)
- Import subscriptions from OPML
- Add subscription from web form
- Background task to fetch feed entries
- Frontend to show available entries by user query

Let’s do it!

- In plain WSGI (sort of)
- Without frameworks (no Flask, Pyramid, etc)
- Without ORM (no SQLAlchemy, Peewee, etc)
- Is it still real? :)

WSGI Application

- Should be a callable object (function or class)
- Receive two arguments
  - environ (Dict with environment vars for current request)
  - start_response (Response handler)
- Should pass status code and headers to response handler
- Should return iterable or yield generator

WSGI Application

```
from wsgiref.simple_server import make_server


def app(environ, start_response):
    start_response('200 OK', [('Content-Type', 'text/html')])
    yield 'Hello, world!'


if __name__ == '__main__':
    server = make_server('0.0.0.0', 8000, app)

    try:
        print('Running on http://0.0.0.0:8000/')
        server.serve_forever()
    except KeyboardInterrupt:
        print
```

Problems

Routing

- AKA Django's urlpatterns or Flask's @app.route
- environ['PATH_INFO'] contains current user path
- How to process it? With regular expressions, without?
- How to pass necessary information to views?
- How to execute view due current path?
- Questions, questions, questions

Database, cache

- AKA Django's ORM, cache and SQLAlchemy, Flask-Cache
- No problem here
- Hail to psycopg2 (PostgreSQL Python library)
  psycopg2 has ThreadConnectionPool from the box
- And redis-py (Redis Python library)
  redis.StrictRedis.from_url makes you happy

Session

- AKA Django's session, Flask's session
- Where to store session
- Which session to use (cookie-based, cache-based, etc)
- Is Beaker good enough? (No Redis backend, no clear configuration, but has WSGI middleware)

OAuth

- AKA django-social-auth, Flask-OAuth
- Mom, I don’t wanna to build one more bicycle :(
- Which library to use oauth, oauthlib or oauth2?
- What about python-social-auth?
- Questions, questions...

Import from OPML

- No problem here
- lxml is only way

Background tasks

- No problem here
- Use Celery if you want to do a serious business
- rq if you’re not so serious

Building your own web-framework

Routing

- Op, op, welcome **WebOb**
- Rour, rour, welcome **routr**
- Oo, oo, oo, welcome **rororo**

Routing

```
from rororo.app import create_app
from rororo.routes import GET
from rororo.schema import qs


ROUTES = (
    GET('/', 'views.index', renderer='index.html'),
    GET('/browser/subscriptions',
        qs(limit=int),
        'views.subscriptions',
        renderer='subscriptions.html'),
    POST('/import',
         'views.import_subscriptions',
         renderer='import_subscriptions'),
)

app = create_app(__name__)
```

Routing

```
def index():
    return {...}


def subscriptions(limit=None):
    return {...}


def import_subscriptions(request):
    return {...}
```

Database, cache

```
from psycopg2.extensions import UNICODE, UNICODEARRAY, register_type
from psycopg2.pool import ThreadedConnectionPool
from redis import StrictRedis

from .utils import db_url_to_params


DATABASE_URL = '...'
DATABASE_POOL_OPTIONS = {...}
REDIS_URL = '...'

register_type(UNICODE)
register_type(UNICODEARRAY)

app.db = ThreadedConnectionPool(
    dict(DATABASE_POOL_OPTIONS, **db_url_to_params(DATABASE_URL))
)
app.redis = StrictRedis.from_url(REDIS_URL)
```

Session

```
from beaker.middleware import SessionMiddleware


SESSION_OPTIONS = {...}

app.wsgi_app = SessionMiddleware(app.wsgi_app, SESSION_OPTIONS)
```

Session

```
def index(request):
    user = request.environ['beaker.session'].get('user')
    ...
```

OAuth

- Em...
- Wasted so many time, better to use Dummy User login

Background tasks

```
# app.py
from rq import Queue

QUEUE_ENTRIES = ‘readthestuff_entries’
QUEUES = (QUEUE_ENTRIES, )

queue_entries = Queue(QUEUE_ENTRIES, connection=redis)

# rqworker
$ rqworker -c app
```

Total result?

- 4 hours till end
- NFL Sunday games already started
- But, I’m not giving up

Total result?

- http://readthestuff.com
- https://github.com/playpauseandstop/readthestuff

Questions?