

# Combine multiple **Flask** applications at one **Heroku** host

Igor Davydenko  
2012 IT Jam

# Flask is...

- <http://flask.pocoo.org/>
- The best Python web-microframework
- Core is Werkzeug for handling requests
- And Jinja2 for rendering templates (making responses)
- With lot of helpers for web-applications (json, sessions, etc)
- Tons of extensions (SQLAlchemy, peewee, Mongo, MongoEngine, Redis, Login, Social, Security, Script, Dropbox, Test, etc, etc, etc)
- Good quality user community

# Heroku is...

- <http://www.heroku.com/>
- One of the easiest cloud application platform
- Support deploying applications in Ruby, Node.js, Clojure, Java, Python and Scala
- Include command-line toolbelt to perform any action with your host via command line, like `$ heroku create`, `$ heroku logs`, `$ heroku scale`
- Has several plans for databases (from \$50/month)
- Support tons of addons (Mongo, Redis, Sentry, etc, etc)

# Flask application

```
$ cat app.py
from flask import Flask

import settings

app = Flask(__name__)
app.config.from_object(settings)

@app.route('/')
def index():
    return 'Hello, world!'

if __name__ == '__main__':
    app.run()
```

# Prepare and deploy app to Heroku

```
$ tree .
```

```
├── app.py  
├── Procfile  
├── settings.py  
└── requirements.txt
```

```
$ cat requirements.txt
```

```
Flask==0.9
```

```
gunicorn==0.14.6
```

```
$ cat Procfile
```

```
web: gunicorn app:app -b 0.0.0.0:$PORT -w 4
```

```
$ heroku create --name itjam2012-demo
```

```
...
```

```
$ git push heroku master
```

```
...
```

```
$ heroku scale web=1
```

```
...
```

# Flask multi application

```
from flask import Flask
from werkzeug.serving import run_simple
from werkzeug.wsgi import DispatcherMiddleware

import settings

def init_app(name):
    app = Flask(name)
    app.config.from_object(settings)

    @app.route('/')
    def index():
        return 'Hello, {0}!'.format(name)

    return app

main_app = init_app('world')
child_app = init_app('moon')

multi_app = DispatcherMiddleware(main_app, {
    '/moon': child_app,
})
```

# Redeploy to Heroku

```
$ cat Procfile
web: gunicorn app:multi_app -b 0.0.0.0:$PORT -w 4
$ git add .
...
$ git commit -m 'Implement simple Flask multi-app.'
...
$ git push heroku master
...
$ curl http://itjam2012-demo.herokuapp.com/
Hello, world!
$ curl http://itjam2012-demo.herokuapp.com/moon
Hello, moon!
```

# Real world examples

```
app = Flask('igordavydenko')
app.config.from_object(settings)
...
far_app.debug, fd_app.debug = app.debug, app.debug
multi_app = DispatcherMiddleware(app, {
    '/Flask-And-Redis': far_app,
    '/Flask-Dropbox': fd_app,
})
```

- *app* - <http://igordavydenko.com/>
- *fd\_app* - <http://igordavydenko.com/Flask-Dropbox>
- *far\_app* - <http://igordavydenko.com/Flask-And-Redis>



# Other solutions

- <http://flask.pocoo.org/docs/patterns/appdispatch/>
- SubdomainDispatcher  
`application = SubdomainDispatcher('domain.com', create_app)`
- PathDispatcher  
`application = PathDispatcher(default_app, create_app)`

# Questions? Solutions?

I am Igor Davydenko  
<https://github.com/playpauseandstop>