

Todo-be-do-be-do with Flask

Pradeep Gowda

Feb 26, 2013

About me

- ▶ Pradeep Gowda – @btbytes

About me

- ▶ Pradeep Gowda – @btbytes
- ▶ CIO at ENthEnergy LLC

About me

- ▶ Pradeep Gowda – @btbytes
- ▶ CIO at ENthEnergy LLC
- ▶ Python user since the last millenium(1999)

About me

- ▶ Pradeep Gowda – @btbytes
- ▶ CIO at ENthEnergy LLC
- ▶ Python user since the last millenium(1999)
- ▶ Using Python professionally since 2002

- ▶ A micro framework by Armin Ronacher

- ▶ A micro framework by Armin Ronacher
- ▶ Based on Werkzeug, Jinja2 and good intentions (and taste)

- ▶ A micro framework by Armin Ronacher
- ▶ Based on Werkzeug, Jinja2 and good intentions (and taste)
- ▶ Clean way to write small apps

- ▶ Started out as an April fool's joke

Inspirations

- ▶ Started out as an April fool's joke
 - ▶ `http://denied.immersedcode.org/deny.py`

Inspirations

- ▶ Started out as an April fool's joke
 - ▶ `http://denied.immersedcode.org/deny.py`
- ▶ Name was a play on the bottle framework

Inspirations

- ▶ Started out as an April fool's joke
 - ▶ `http://denied.immersedcode.org/deny.py`
- ▶ Name was a play on the bottle framework
 - ▶ Since then has grown in popularity

A Micro framework

- ▶ Simplicity and Small

A Micro framework

- ▶ Simplicity and Small
- ▶ Entire app in handful of modules

A Micro framework

- ▶ Simplicity and Small
- ▶ Entire app in handful of modules
- ▶ Freedom to choose libraries you like

Key features of Flask

- ▶ Easy to learn

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults
- ▶ Excellent documentation

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults
- ▶ Excellent documentation
- ▶ Flask-Extensions

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults
- ▶ Excellent documentation
- ▶ Flask-Extensions
- ▶ Blueprints

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults
- ▶ Excellent documentation
- ▶ Flask-Extensions
- ▶ Blueprints
 - ▶ Application components

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults
- ▶ Excellent documentation
- ▶ Flask-Extensions
- ▶ Blueprints
 - ▶ Application components
 - ▶ Reuse common patterns

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults
- ▶ Excellent documentation
- ▶ Flask-Extensions
- ▶ Blueprints
 - ▶ Application components
 - ▶ Reuse common patterns
 - ▶ (eg: `/admin`, `/user`, `/reports`)

Key features of Flask

- ▶ Easy to learn
- ▶ Good, tasteful defaults
- ▶ Excellent documentation
- ▶ Flask-Extensions
- ▶ Blueprints
 - ▶ Application components
 - ▶ Reuse common patterns
 - ▶ (eg: `/admin`, `/user`, `/reports`)
- ▶ Flask-Snippets

Popular combinations

- ▶ Flask + Jinja2 + SQLAlchemy

Popular combinations

- ▶ Flask + Jinja2 + SQLAlchemy
- ▶ Flask + Mako + SQLAlchemy

Popular combinations

- ▶ Flask + Jinja2 + SQLAlchemy
- ▶ Flask + Mako + SQLAlchemy
- ▶ Flask + Jinja2 + Peewee

Popular combinations

- ▶ Flask + Jinja2 + SQLAlchemy
- ▶ Flask + Mako + SQLAlchemy
- ▶ Flask + Jinja2 + Peewee
- ▶ Flask + CouchDB

How to Install

- ▶ `pip install Flask`

How to Install

- ▶ `pip install Flask`
- ▶ <http://flask.pocoo.org>

Hello world

```
from flask import Flask

app = Flask(__name__)

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

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

Visit the app at <http://localhost:5000>.

Defaults

- ▶ Look under `templates` for templates (default Jinja2)

Defaults

- ▶ Look under `templates` for templates (default Jinja2)
- ▶ Look under `static` for assets (CSS, js etc.,)

The todo app

- ▶ Flask

The todo app

- ▶ Flask
- ▶ Sqlite for persistence

The todo app

- ▶ Flask
- ▶ Sqlite for persistence
- ▶ PeeWee for ORM

The todo app

- ▶ Flask
- ▶ Sqlite for persistence
- ▶ PeeWee for ORM
- ▶ Flask-Peewee wrapper + Authentication

The todo app

- ▶ Flask
- ▶ Sqlite for persistence
- ▶ PeeWee for ORM
- ▶ Flask-PeeWee wrapper + Authentication
- ▶ Bootstrap for CSS

App Demo

Todo-be-do-be-do

When to Flask?

- ▶ Think App vs Application

When to Flask?

- ▶ Think App vs Application
- ▶ Think standalone Services (JSON/REST)

When to Flask?

- ▶ Think App vs Application
- ▶ Think standalone Services (JSON/REST)
- ▶ Think web apps interacting over HTTP vs lib calls.

Thank you

The code for the application and the presentation are available at

- ▶ <http://github.com/indypy/todoflask>

Questions to pradeep@btbytes.com or @btbytes.