

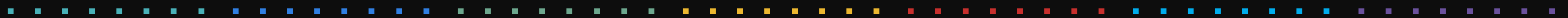
WMNBE-2203 | BACK-END DEVELOPMENT

Flask #9

Blueprints / Web APIs



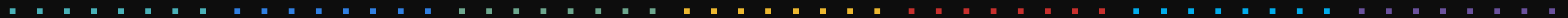
ΠΕΡΙΕΧΟΜΕΝΑ



Περιεχόμενα

- Blueprints
- Web APIs
 - REST

BLUEPRINTS



Blueprints

Modularity

Με τη χρήση των **Blueprint**, μπορούμε να κάνουμε την εφαρμογή μας πιο *modular*, δηλαδή να χωρίσουμε το project σε μικρότερα τμήματα.

Κάθε τέτοιο τμήμα μπορεί να συνοδεύεται και από το δικό του **URL prefix**.

Blueprints

Instantiation

Ένα **Blueprint** χρησιμοποιείται σχεδόν με τον ίδιο τρόπο όπως και ένα **Flask app**, χωρίς όμως να είναι ένα ολοκληρωμένο **app**.

Είναι στην ουσία μια συλλογή από επί μέρους στοιχεία που μπορούν να ενσωματωθούν σε μία εφαρμογή.

```
bp = Blueprint('main', __name__, url_prefix='/')

@bp.get('/')
def home():
    return 'Hello from a Blueprint!'
```

Blueprints

Registration

Για να ενσωματώσουμε ένα **Blueprint** σε μία εφαρμογή αρκεί να το εγγράψουμε σε αυτή.

```
app.register_blueprint(bp)
```

Από την έκδοση **2.0** του **Flask** και μετά υποστηρίζονται και εμφωλευμένα **Blueprint**.

```
parent = Blueprint('parent', __name__, url_prefix='/parent')
child = Blueprint('child', __name__, url_prefix='/child')
parent.register_blueprint(child)
app.register_blueprint(parent)
```

Blueprints

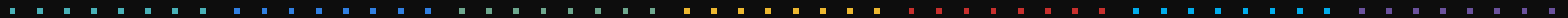
URLs

Μια αλλαγή που πρέπει να κάνουμε, σε περίπτωση που μετατρέψουμε μία εφαρμογή μας ώστε να χρησιμοποιεί **Blueprint**, είναι η εξής.

Πρέπει στη μέθοδο **url_for** να συμπεριλαμβάνουμε και το όνομα του **Blueprint**.

```
url_for('main.home')
```


WEB APIS



Web APIs

What is it?

Ένα *Web API* (*Application Programming Interface*) είναι μια διεπαφή που επιτρέπει σε συστήματα να επικοινωνούν μεταξύ τους, μέσω του πρωτοκόλλου **HTTP** / **HTTPS**.

Επιτρέπει τη διαλειτουργικότητα μεταξύ εφαρμογών, κυρίως μέσω **JSON** ή **XML**.

Πολλές γνωστές εφαρμογές (*Google, Facebook, Twitter*), παρέχουν κάποιου είδους **Web API** (πολύ συχνά αναφέρεται απλά ως **API**).

Web APIs

What is it meant for?

- Για να παρέχουμε πρόσβαση στα δεδομένα της εφαρμογής μας σε τρίτους.
- Για την επικοινωνία διακριτών συστημάτων μεταξύ τους.
- Για το backend κομμάτι μιας *SPA* ή *mobile* εφαρμογής.

Web APIs

REST

Ένα *Web API* μπορεί να δομηθεί με ποικίλους τρόπους και συχνά χρειάζεται ένα σύνολο οδηγιών / καλών πρακτικών για να αναπτυχθεί σωστά.

Μια δημοφιλή αρχιτεκτονική, για το σκοπό αυτό, είναι η αρχιτεκτονική *REST (Representational state transfer)*.

Ένα *Web API* χαρακτηρίζεται ως *RESTful* όταν ακολουθεί τις οδηγίες αυτές.

Web APIs

REST

Σύμφωνα με τη διατριβή του *Roy Fielding* (όπου πρωτοπαρουσιάστηκε η ιδέα του REST):

An application or architecture considered RESTful or REST-style is characterized by:

- State and functionality are divided into distributed resources
- Every resource is uniquely addressable using a uniform and minimal set of commands (typically using HTTP commands of GET, POST, PUT, or DELETE over the Internet)
- The protocol is client/server, stateless, layered, and supports caching

Web APIs

Super simple Web API with Flask

Για να δημιουργήσουμε ένα απλό **Web API** με **Flask**, χωρίς τη χρήση κάποιας επέκτασης, θα ακολουθήσουμε τις παρακάτω οδηγίες:

- Σε ξεχωριστό **Blueprint**, με δικό του *URL Prefix*, ορίζουμε τα **routes** / **endpoints** του *API*.
 - Μέσα στο *prefix*, καλό είναι να υπάρχει και πρόβλεψη για versioning.
- Με κατάλληλη διαχείριση των **HTTP Verbs / Methods** εκτελούμε τις λειτουργίες **CRUD**.
- Ένα *view function* αρκεί να επιστρέξει ένα **Dictionary** ώστε αυτό να μετατραπεί σε *JSON*.
 - Αν θέλουμε να επιστρέψουμε μία λίστα από στοιχεία, χρησιμοποιούμε τη μέθοδο **jsonify**.

Web APIs

Παράδειγμα

```
from flask import Blueprint, jsonify

...

bp = Blueprint('api', __name__, url_prefix='/api/v1')

users = [
    { 'username': 'alice', 'Firstname': 'Alice' },
    { 'username': 'bob', 'Firstname': 'Bob' }
]

@bp.route('/users/', methods=['GET'])
def get_users():
    return jsonify(users)
```

Χρήσιμα links

📄 Blueprints and Views — Flask Documentation (2.2.x)
<https://flask.palletsprojects.com/en/2.2.x/tutorial/views/>

📄 Modular Applications with Blueprints — Flask Documentation (2.2.x)
<https://flask.palletsprojects.com/en/2.2.x/blueprints/>

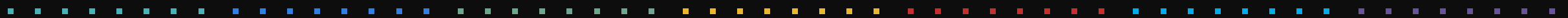
📄 Creating Web APIs with Python and Flask | Programming Historian
<https://programminghistorian.org/en/lessons/creating...>

SA Representational State Transfer (REST)
<https://www.service-architecture.com/articles/web-s...>

📄 What is REST? | Codecademy
<https://www.codecademy.com/articles/what-is-rest>

M HTTP request methods - HTTP | MDN
<https://developer.mozilla.org/en-US/docs/Web/HTTP/...>

🌐 HTTP Status Codes
<https://restfulapi.net/http-status-codes/>



Extra info

🐾 Understanding SOAP vs REST: Basics And Differences
<https://smartbear.com/blog/soap-vs-rest-whats-the-di...>

📖 Flask-RESTful — Flask-RESTful 0.3.8 documentation
<https://flask-restful.readthedocs.io/en/latest/>

📖 Flask-Marshmallow: Flask + marshmallow for beautiful APIs — Flask-Marshmallow 0.14.0 documentation
<https://flask-marshmallow.readthedocs.io/en/latest/#...>

📖 marshmallow: simplified object serialization — marshmallow 3.17.1 documentation
<https://marshmallow.readthedocs.io/en/stable/>

🔗 Building Restful APIs With Flask and SQLAlchemy (Part 1) | by Babatunde Koiki | Better Programming
<https://betterprogramming.pub/building-restful-apis-...>

🔗 Python REST APIs With Flask, Connexion, and SQLAlchemy – Real Python
<https://realpython.com/flask-connexion-rest-api/>

THANK YOU!