

Web app in Flask



- Giorgio Colomban <gcolomban@gmail.com>

Potete trovare queste slide all'indirizzo:
(<https://slides.poul.org/2018/python/flask>)

Cos'è Flask?

Cos'è Flask?

Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's BSD licensed!

EH?

Un web framework è un insieme di tecnologie utili per lo sviluppo di web app

Perché proprio Flask?

Esistono altri web framework per Python, perché utilizzare proprio Flask?

- E' semplice da imparare
- Ha un nucleo leggero (come si può intuire dal “micro” in “microframework”), ma facilmente espandibile
- Ha un'ottima **documentazione**
- **E' totalmente open source!**

Installazione

Installazione

Ci sono due modi principali per **installare** Flask:

- Tramite **virtualenv**, andando quindi ad installare Flask in un ambiente virtualizzato “isolato”, per evitare conflitti con altre versioni o librerie di Python
- Direttamente su **tutto il sistema**, rischiando di riscontrare i conflitti sopracitati

virtualenv

Qui seguiremo passo passo l'installazione su sistemi Linux e MacOSX, potete trovare le informazioni per l'installazione su Windows nella [documentazione di Flask](#)

- Installiamo virtualenv tramite il comando

```
$ sudo pip install virtualenv
```

- Creiamo il nostro ambiente all'interno di una cartella dedicata

```
$ mkdir corso_flask  
$ cd corso_flask  
$ virtualenv venv
```

virtualenv

- Ogni volta che vorremo lavorare al progetto, basterà eseguire il comando

```
$ source venv/bin/activate
```

- O, in alternativa

```
$ . venv/bin/activate
```

- Una volta dentro l'ambiente venv possiamo installare Flask tramite il comando

```
$ pip install Flask
```

- Per uscire dell'environment basta eseguire

```
$ deactivate
```

System wide installation

L'installazione su tutto il sistema è ancora più semplice: è sufficiente eseguire il comando

```
$ sudo pip install Flask
```

Hello, world!

Hello, world!

Cominciamo a scrivere un po' di codice!

```
from flask import Flask
app = Flask(__name__)

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

Hello, world!

Ok, cosa abbiamo appena fatto?

- Abbiamo per prima cosa importato la **classe** Flask
- Dopo abbiamo creato un'**istanza** di questa classe, inserendo come argomento il nome del modulo dell'applicazione. Quando si utilizza un singolo modulo, come in questo caso, va utilizzato come argomento `__name__`
- Il **decoratore** `route()` indica l'URL che attiverà la nostra funzione
- Alla funzione viene dato un **nome**, che verrà anche utilizzato per generare gli URL relativi alla funzione stessa

Hello, world!

Controller

- Una qualsiasi funzione in Flask diventa un **controller** decorata (come `@app.route()` nel nostro esempio)
- La decorazione associa una parte di URL, detta **route**, ad una funzione
- Il corpo della risposta è costituito da tutto ciò che viene ritornato dal controller

Hello, world!

Deploy dell'applicazione in locale

Per verificare che tutto funzioni, effettueremo il deploy dell'applicazione in locale

```
$ export FLASK_APP=hello.py  
$ flask run
```

Hello, world!

Deploy dell'applicazione in locale

Basterà andare su `http://127.0.0.1:5000/` per vedere il nostro hello, world

Templating

Templating

Flask utilizza la libreria **Jinja2** per generare pagine web dinamiche.

```
from Flask import render_template

@app.route('/hello/')
@app.route('/hello/<name>')
def hello(name=None):
    return render_template('hello.html', name=name)
```

Templating

Sintassi

La sintassi dei template è semplice sintassi HTML

```
<!-- templates/hello.html -->
<!doctype.html>
<title>Hello from Flask</title>
{% if name %}
    <h1>Hello {{ name }}!</h1>
{% else %}
    <h1>Hello World!</h1>
Testo prova

{% endif %}
```

Templating

Riempimento dei templates

- Per riempire i templates va utilizzata la funzione `render_template`
- Le variabili vanno passate a `render_template` come `chiave=valore`

```
@app.route('/')  
def index() :  
    var1 = stuff()  
    var2 = more_stuff()  
    return render_template('page.html', var1=var1, var2=var2)
```

DEMO

(Potete trovare il codice completo [qui](#))

Domande?

Fonti

- Documentazione di Flask
- **Slides** del corso Python 2016 di Emanuele Santoro

Thank you!

