Flask Tutorial

Rafael Castro G. Silva

Department of Computer Science
University of Copenhagen

20/05/2025

Learning Goals

Learning Goals

1. Learn what is Flask
2. Learn how to bootstrap a Flask web application
3. Learn some basic Flask concepts
4. Learn the Model View Control (MVC) pattern
5. Learn basic Docker

Flask Introduction

What is Flask?

What is Flask?

- Flask is a micro framework for web applications

What is Flask?

- Flask is a micro framework for web applications
  - Framework: a set of modules and libraries that provide basic functionalities
    May or may nor enforce some standards (e.g. directory structures, MVC pattern, etc)
  - Micro: enforces very little
  - In/for Python

What is Flask?

- Flask is a micro framework for web applications
  - Framework: a set of modules and libraries that provide basic functionalities
    May or may nor enforce some standards (e.g. directory structures, MVC pattern, etc)
  - Micro: enforces very little
  - In/for Python

Other frameworks out there

- Ruby on Rails
- Django (Python)
- Java Spring
- Symfony (PHP)

Who uses Flask?

Who uses Flask?

airbnb
reddit
Uber

What do I need know?

What do I need know?

- Very basic Python
  - Variables, functions, indentation, lists, maps (dictionaries), if-then-else, for loops, class (OO programming), import libraries
- Terminal commands: `ls`, `cd`, `pwd`...
- Basic HTML
- CSS (optional)
- SQL

Flask Concepts

The App Variable

- Our app is an instance of the Flask class
- __name__ is a default configuration telling that the files of the project are in the current directory

```python
from flask import Flask

app = Flask(__name__)
```

Routing

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"
```

- `@app.route("/")` is Python decorator: extends the behavior of a function
    - Tells Flask what URL should trigger our function
    - The `"/"` is the root of our web site domain
      `http://127.0.0.1:5000` takes us to this route
    - We can have other routes: `@app.route("/register")`
      `http://127.0.0.1:5000/register` takes us to this other route

Routing

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"
```

- `@app.route("/")` is Python decorator: extends the behavior of a function
  - Tells Flask what URL should trigger our function
  - The `"/"` is the root of our web site domain
    `http://127.0.0.1:5000` takes us to this route
  - We can have other routes: `@app.route("/register")`
    `http://127.0.0.1:5000/register` takes us to this other route

What else to learn:

- Variable rules:
  `https://flask.palletsprojects.com/en/stable/quickstart/#variable-rules`
- HTTP methods:
  `https://flask.palletsprojects.com/en/stable/quickstart/#http-methods`

Templates

File ./app.py:

```
from flask import render_template

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

File ./templates/hello.html:

```
<!doctype html>
<title>Hello from Flask</title>
{% if person %}
  <h1>Hello {{ person }}!</h1>
{% else %}
  <h1>Hello, World!</h1>
{% endif %}
```

Templates

File ./app.py:

```python
from flask import render_template

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

File ./templates/hello.html:

```html
<!doctype html>
<title>Hello from Flask</title>
{% if person %}
  <h1>Hello {{ person }}!</h1>
{% else %}
  <h1>Hello, World!</h1>
{% endif %}
```

- Import render_template
  (Jinja2 template engine)
- 2 routes for the same function
- The function has a named argument

Templates

File ./app.py:

```python
from flask import render_template

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

File ./templates/hello.html:

```html
<!doctype html>
<title>Hello from Flask</title>
{% if person %}
  <h1>Hello {{ person }}!</h1>
{% else %}
  <h1>Hello, World!</h1>
{% endif %}
```

- Import render_template (Jinja2 template engine)
- 2 routes for the same function
- The function has a named argument

- render_template passes the named arguments
- The template has access to the Python variables.
  Inside of {% %} is Jinja2 code
  Inside of {{ }} is to print as HTML

What else to learn about templates?

- If statements: `https://jinja.palletsprojects.com/en/stable/templates/#if`
- For loops: `https://jinja.palletsprojects.com/en/stable/templates/#for`
- Template Inheritance
  `https://jinja.palletsprojects.com/en/stable/templates/#template-inheritance`

MVC

The Bank Example Project

Let's write our web apps!