

Python Software Engineer

Flask

Antonio Pisanello | automne 2022



Sommaire

- Introduction Flask
- Jinja



Introduction Flask

Flask ?

- Flask est un micro framework web écrit en python
- Ultra léger
- Dynamique : permet de créer des sites web ainsi que des API complexes
- Flexible : peut être étendu via des librairies et modules
- Maintenu par une grande communauté



Introduction Flask

Qui utilise FLASK

- Samsung
- Netflix
- Uber
- Zalando
- Airbnb
- Reddit
- Trivago
- Et vous bientôt ;P



Introduction Flask

REST API / Web Server

- Routing
- HTML Methods
- HTML Errors



Introduction Flask

REST API / Web Server

- Routing
- HTML Methods
 - GET; HEAD; POST; PUT; DELETE; CONNECT; OPTION; TRACE; PATCH
 - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>
- HTML Errors



Introduction Flask

REST API / Web Server

- Routing
- HTML Methods
- HTML Errors
 - 1xx: Information
 - 2xx: Successful
 - 3xx: Redirection
 - 4xx: Client Errors
 - 5xx: Server Errors
 - https://en.wikipedia.org/wiki/List_of_HTTP_status_codes



Introduction Flask

Installation de Flask

- ``pip install flask``
- L'installation installe Flask et Jinja



Introduction Flask

Lancement de Flask

- export FLASK_ENV=development
- export FLASK_APP=server_file.py
- flask run
- `python3 server_file.py`

```
from flask import Flask

app = Flask(__name__)

# implementation des routes

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



Introduction Flask

Le serveur web

- Par défaut Flask se lance comme suit :
 - <http://127.0.0.1:5000>
 - Uniquement accessible par « localhost »
 - Le port 5000 est le port par défaut de Flask



Introduction Flask

Le serveur web

- Ces paramètres peuvent être modifié dans l'appel de fonction ``run()``
- ``app.run(host='0.0.0.0', port='8080', debug=True)``
 - `http://0.0.0.0:8080`
 - Accessible par d'autres ordinateurs du même réseau, public
 - `Debug=True`, signifie que le serveur est en mode debug (auto-relaunch on saving)
 - Le port 8080 est maintenant choisi.
 - Si on veut mettre le port 80, qui est le port officiel pour les serveurs web, c'est possible, il faut juste lancer python avec sudo



Introduction Flask

Les routes

- `@app.route('/')``
- Important de spécifier l'URL
- Le nom de la fonction, utilisée plus tard avec `url_for()`
- Return statement.
 - JSON
 - HTML
 - TEXT

```
@app.route('/')  
def index():  
    return "<h1>Hello World</h1>"
```



Introduction Flask

Les routes dynamique

- `@app.route('/allow/<variable_name>')`
- `@app.route('/allow/<converter:variable_name>')`
- Si le converter n'est pas respecté -> erreur

Converters	
string	(Défaut), texte sans slash '/'
int	Nombres entiers positifs
float	Nombres flottants positifs
path	Comme string, sauf que cette fois les slash sont acceptés
uuid	uuid string



Introduction Flask

Les routes dynamique

- `@app.route('/allow/<variable_name>')`
- `@app.route('/allow/<converter:variable_name>')`
- On récupère le paramètre, dans la signature de la fonction
- `def allow(variable_name)`

```
@app.route('/allow/<int:Number>')
def allow(Number:int):
    if Number < 25:
        return f'You have been allowed to enter because your number is {str(Number)}'
    else:
        return f'You are not allowed'
```



Introduction Jinja

Template engine

- Installer par défaut avec Flask
- Variables
- Statements
- Blocks
- Filtres | Tests
- Macros

<https://jinja.palletsprojects.com/en/3.1.x/templates/>



Introduction Jinja

Variables

- Passée par flask en paramètre
- Peuvent être accédée par `.` ou `[]`
- Sont affichée à l'aide de `{{ .. }}`

`{% ... %}` for Statements

`{{ ... }}` for Expressions to print to the template output

`{# ... #}` for Comments not included in the template output



Introduction Jinja

Statements

- IF
- FOR
- Sont affichées à l'aide de `{% .. %}`

`{% ... %}` for Statements

`{{ ... }}` for Expressions to print to the template output

`{# ... #}` for Comments not included in the template output



Introduction Jinja

Blocks

- Concept de template
- Diminution de code

```
Login.html

{%extends 'includes/base.html'%}
{%block title%}
Login
{%endblock%}
{%block content%}
<form action="#" method="POST">
  <div class="form-group">
    <label for="uName">Username</label>
    <input type="text" name="uName" id="uName" required placeholder="username" />
  </div>
  <div class="form-group">
    <label for="uPwd">Password</label>
    <input type="password" name="uPwd" id="uPwd" placeholder="password" />
  </div>
  <div class="form-group">
    <input type="submit" name="btnSubmit" value="S'enregistrer" />
  </div>
</form>
{%endblock%}
```



Introduction Jinja

Filtres et Tests

List of Builtin Filters

<u>abs()</u>	<u>forceescape()</u>	<u>map()</u>	<u>select()</u>	<u>unique()</u>
<u>attr()</u>	<u>format()</u>	<u>max()</u>	<u>selectattr()</u>	<u>upper()</u>
<u>batch()</u>	<u>groupby()</u>	<u>min()</u>	<u>slice()</u>	<u>urlencode()</u>
<u>capitalize()</u>	<u>indent()</u>	<u>pprint()</u>	<u>sort()</u>	<u>urlize()</u>
<u>center()</u>	<u>int()</u>	<u>random()</u>	<u>string()</u>	<u>wordcount()</u>
<u>default()</u>	<u>items()</u>	<u>reject()</u>	<u>striptags()</u>	<u>wordwrap()</u>
<u>dictsort()</u>	<u>join()</u>	<u>rejectattr()</u>	<u>sum()</u>	<u>xmlattr()</u>
<u>escape()</u>	<u>last()</u>	<u>replace()</u>	<u>title()</u>	
<u>filesizeformat()</u>	<u>length()</u>	<u>reverse()</u>	<u>tojson()</u>	
<u>first()</u>	<u>list()</u>	<u>round()</u>	<u>trim()</u>	
<u>float()</u>	<u>lower()</u>	<u>safe()</u>	<u>truncate()</u>	

List of Builtin Tests

<u>boolean()</u>	<u>even()</u>	<u>in()</u>	<u>mapping()</u>	<u>sequence()</u>
<u>callable()</u>	<u>false()</u>	<u>integer()</u>	<u>ne()</u>	<u>string()</u>
<u>defined()</u>	<u>filter()</u>	<u>iterable()</u>	<u>none()</u>	<u>test()</u>
<u>divisibleby()</u>	<u>float()</u>	<u>le()</u>	<u>number()</u>	<u>true()</u>
<u>eq()</u>	<u>ge()</u>	<u>lower()</u>	<u>odd()</u>	<u>undefined()</u>
<u>escaped()</u>	<u>gt()</u>	<u>lt()</u>	<u>sameas()</u>	<u>upper()</u>



Introduction Jinja

Macros

- Réutilisation de code
- Diminution de code
- Generation HTML

```
_formHelper.html

{% macro afficherInput(field, showErrors="true") %}
    <div class="form-group">
        {{field.label}}
        {{field(**kwargs) | safe}}
        {% if showErrors="true" and field.errors %}
            <div class="invalid-feedback">>
                {% for error in field.errors %}
                    <span class="help-inline">{{ error }}</span>
                {% endfor %}
            </div>
        {% endif %}
    </div>
{% endmacro %}
```

