

Journée de partage de pratiques

Démarrer avec Flask

31 mai 2023

1. Installation

2. 1^{er} exemple

3. Les routes

4. Méthode GET

5. Les modèles

6. Lire un contenu et l'afficher

7. Se connecter

8. Redirections

9. Répertoire static

10. Lien avec MySQL

```
1 david@coolermaster:~$ mkdir mon_projet
2 david@coolermaster:~$ cd mon_projet/
3 david@coolermaster:~/mon_projet$ python3 -m venv .venv
4 david@coolermaster:~/mon_projet$ source .venv/bin/activate
5 (.venv) david@coolermaster:~/mon_projet$ pip3 install flask
6 Collecting flask
7   Using cached Flask-2.3.2-py3-none-any.whl (96 kB)
```

Fichier hello.py

```
from flask import Flask

app = Flask(__name__)

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

*Dans cette situation, __name__ a pour valeur hello
Le principe est de renvoyer du HTML*

De la doc. officielle

1. Installation

2. 1^{er} exemple

3. Les routes

4. Méthode GET

5. Les modèles

6. Lire un contenu et l'afficher

7. Se connecter

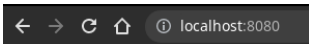
8. Redirections

9. Répertoire static

10. Lien avec MySQL

```
18 (.venv) david@coolermaster:~/mon_projet$ flask --app hello.py run -p 8080
19 * Serving Flask app 'hello.py'
20 * Debug mode: off
21 WARNING: This is a development server. Do not use it in a production deployment.
22 Use a production WSGI server instead.
23 * Running on http://127.0.0.1:8080
24 Press CTRL+C to quit
25 127.0.0.1 - - [29/May/2023 12:33:32] "GET / HTTP/1.1" 200 -
```

par défaut le numéro de port est 5000



Hello, World!

En intégrant dans le fichier `hello.py` :

```
if __name__ == "__main__":
    app.run(port=8080)
```

On peut utiliser `$ python3 hello.py`

Routes de hello

1. Installation

2. 1^{er} exemple

3. Les routes

4. Méthode
GET

5. Les
modèles

6. Lire un
contenu et
l'afficher

7. Se
connecter

8.
Redirections

9. Répertoire
static

10. Lien avec
MySQL

```
66 (.env) david@coolermaster:~/mon_projet$ flask --app hello.py routes
67 hello
68 Endpoint      Methods  Rule
69 -----
70 hello_world   GET      /
71 static        GET      /static/<path:filename>
```

*Une route static est présente :
ainsi s'il existe seulement un fichier ./static/a/test :*

- *localhost:8080/static renverra 404*
- *localhost:8080/static/a renverra 404*
- *localhost:8080/static/a/test accédera au fichier test
(le téléchargera)*

Si le fichier est ./static/a/test.html, l'accès à localhost:8080/static/a/test.html interprétera le contenu du fichier.

*La route est prévue par défaut mais Flask ne la crée pas pour autant :
le répertoire static n'existe pas au démarrage.*

Routes de hello

```
1 from flask import Flask
2 from flask import render_template
3
4 app = Flask(__name__)
5
6
7 @app.route("/")
8 def hello_world():
9     return "<p>Hello, World!</p>"
10
11
12 @app.route("/mapage")
13 def vers_ma_page():
14     return render_template("mapage.html")
```

*Flask cherche le fichier mapage.html
dans le répertoire templates :*

```
136 (.venv) david@coolermaster:~/mon_projet$ tree .
137 .
138 ├── hello.py
139 ├── __pycache__
140 │   ├── hello.cpython-311.pyc
141 │   └── templates
142 │       └── mapage.html
143
144 3 directories, 3 files
```

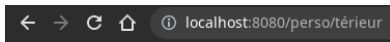
Routes variables

```
17 @app.route("/perso/<nom>")
18 def vers_page_user(nom):
19     nom = nom.capitalize()
20     return render_template("pageperso.html",
21                           lenom=nom)
```

dans le fichier hello.py

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Page perso de {{ lenom }}</title>
6   </head>
7   <body>
8     <h1>Nom : {{ lenom }}</h1>
```

dans le fichier templates/pageperso.html



Nom : Térieur

Les modèles

```
1 <!DOCTYPE html>
2 <html>
3   <head lang="fr">
4     <meta charset="UTF-8">
5     <title>Mon application Flask</title>
6   </head>
7   <body>
8     {% block contenu %} {% endblock %}
9     <hr />
10    <adress>
11      <a href="mailto:david.cobac@ac-nantes.fr">david cobac</a>
12    </adress>
13  </body>
14 </html>
```

dans le fichier templates/monmodele.html

```
1 {% extends 'monmodele.html' %}
2
3 {% block contenu %}
4 <h1>Bonjour {{ leprenom }} {{ lenom }}</h1>
5 {% endblock %}
```

dans le fichier templates/pageperso.html



Bonjour David Sanitaire

1. Installation

2. 1^{er} exemple

3. Les routes

4. Méthode
GET

5. Les
modèles

6. Lire un
contenu et
l'afficher

7. Se
connecter

8.
Redirections

9. Répertoire
static

10. Lien avec
MySQL

```
1 {% extends 'monmodele.html' %}
2
3 {% block contenu %}
4
5 {% if leprenom == "David" %}
6 <h1>Bonjour Concepteur</h1>
7 {% else %}
8 <h2>Bonjour {{ leprenom }} {{ lenom }}</h2>
9 {% endif %}
10
11 {% for lettre in leprenom %}
12 <div class="lettre"> {{ lettre }} </div>
13 {% endfor %}
14
15 {% endblock %}
```

```
1 from flask import Flask, render_template
2 import random
3
4 app = Flask(__name__)
5
6
7 def donnees():
8     r = list(range(100))
9     random.shuffle(r)
10    return r
11
12
13 @app.route("/")
14 def home():
15     return render_template("mapage.html")
16
17
18 @app.route("/data")
19 def affiche_donnees():
20     return render_template("data.html",
21                             tab=donnees())
```

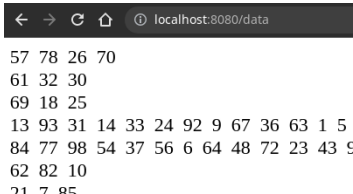
hello.py

`Voir les données`

dans mapage.html

```
1 {% extends "monmodele.html" %}
2
3 {% block contenu %}
4
5 {% for entier in tab %}
6 {{entier}}&nbsp;
7 {% if entier % 5 == 0 %}
8 <br />
9 {% endif %}
10 {% endfor %}
11
12 {% endblock %}
```

data.html



localhost:8080/data

57 78 26 70
61 32 30
69 18 25
13 93 31 14 33 24 92 9 67 36 63 1 5
84 77 98 54 37 56 6 64 48 72 23 43 9
62 82 10
21 7 85

Logique de d'authentification

1. Installation
2. 1^{er} exemple
3. Les routes
4. Méthode GET
5. Les modèles
6. Lire un contenu et l'afficher
7. Se connecter
8. Redirections
9. Répertoire static
10. Lien avec MySQL

```
8 @app.route("/")
9 def home():
10     if not session.get("est_connecte"):
11         return render_template("login.html")
12     else:
13         return render_template("mapage.html")
```

La page `login.html` contient un formulaire utilisant la méthode POST et appelant l'URL `/login`.

La page `mapage.html` contient un lien hypertexte vers l'URL `/logout`.

Logique d'authentification

```
16 @app.route("/login", methods=["POST"])
17 def connexion():
18     if request.form['id'] == 'admin' \
19         and request.form["mdp"] == "password":
20         session["est_connecte"] = True
21     return home()
```

Vérification des champs envoyés avec la méthode POST.
On retourne sur la page d'accueil.

Logique d'authentification

```
16 @app.route("/login", methods=["POST"])
17 def connexion():
18     if request.form['id'] == 'admin' \
19         and request.form["mdp"] == "password":
20         session["est_connecte"] = True
21     return home()
```

Vérification des champs envoyés avec la méthode POST.
On retourne sur la page d'accueil.

Bien sûr, il est hors de question d'avoir
dans un fichier le mot de passe en clair.

Logique d'authentification

1. Installation

2. 1^{er} exemple

3. Les routes

4. Méthode
GET

5. Les
modèles

6. Lire un
contenu et
l'afficher

7. Se
connecter

8.
Redirections

9. Répertoire
static

10. Lien avec
MySQL

```
24 @app.route("/logout")
25 def deconnexion():
26     session["est_connecte"] = False
27     return home()
```

On se déconnecte et on retourne sur la page d'accueil.

Logique d'authentification

- 1. Installation
- 2. 1^{er} exemple
- 3. Les routes
- 4. Méthode GET
- 5. Les modèles
- 6. Lire un contenu et l'afficher
- 7. Se connecter
- 8. Redirections
- 9. Répertoire static
- 10. Lien avec MySQL

```
1 from flask import Flask, render_template, request
2 from flask import session
3
4 app = Flask(__name__)
5 app.secret_key = "ma phrase de passe"
6
7
8 @app.route("/")
9 def home():
10     if not session.get("est_connecte"):
11         return render_template("login.html")
12     else:
13         return render_template("mapage.html")
14
15
16 @app.route("/login", methods=["POST"])
17 def connexion():
18     if request.form['id'] == 'admin' \
19        and request.form["mdp"] == "password":
20         session["est_connecte"] = True
21         return home()
22
23
24 @app.route("/logout")
25 def deconnexion():
26     session["est_connecte"] = False
27     return home()
28
29
30 if __name__ == "__main__":
31     app.debug = True
32     app.run(port=8080)
```

hello.py

```
1 {% extends 'monmodele.html' %}
2
3 {% block contenu %}
4 <form action="/login" method="POST">
5     <input type="username"
6           name="id"
7           placeholder="utilisateur">
8     <input type="password"
9           name="mdp"
10           placeholder="mot de passe">
11     <input type="submit" value="Se connecter">
12 </form>
13 {% endblock %}
```

login.html

```
<a href="/logout">Se déconnecter</a>
```

dans mapage.html

redirect et url_for

1. Installation

2. 1^{er} exemple

3. Les routes

4. Méthode
GET

5. Les
modèles

6. Lire un
contenu et
l'afficher

7. Se
connecter

8.
Redirections

9. Répertoire
static

10. Lien avec
MySQL

```
1 from flask import Flask, render_template, request
2 from flask import session, redirect, url_for
3
4 app = Flask(__name__)
5 app.secret_key = "ma phrase de passe"
6
7
8 @app.route("/")
9 def home():
10     if not session.get("est_connecte"):
11         return render_template("login.html")
12     else:
13         return render_template("mapage.html")
14
15
16 @app.route("/login", methods=["POST"])
17 def verif_connexion():
18     if request.form['id'] == 'admin' and request.form["mdp"] == "password":
19         session["est_connecte"] = True
20     return redirect(url_for("home"))
21
22
23 @app.route("/logout")
24 def deconnexion():
25     session["est_connecte"] = False
26     return redirect(url_for("home"))
27
28
29 if __name__ == "__main__":
30     app.debug = True
31     app.run(port=8080)
```

La route vers **static** est déjà connue (voir 3)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <link rel="stylesheet"
6       href="{{ url_for('static', filename='page.css') }}">
7     <title>Ma page</title>
8     <script type="text/javascript"
9       src="{{ url_for('static', filename='page.js') }}"></script>
```

extrait d'un fichier templates/page.html