R209
TP2

https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-fr

afin d'éviter des problèmes avec le proxy il faut exécuter les commandes suivante avant de commencer

export http_proxy=cache-etu.univ-artois.fr:3128
export https_proxy=cache-etu.univ-artois.fr:3128

Attention faire une machine virtuelle ubuntu bridge et lire le TP en entier 👍

Mise en place et configuration du serveur

Création du serveur

il faut d'abord installer flask pour cela on fait les commandes suivantes:

sudo apt update

ensuite il faut installer python 3 avec pip:

sudo apt install python3-pip

puis flask avec:

pip install flask

```
administrateur@rt-mv:~/projet209$ pip install flask
Defaulting to user installation because normal site-packages is not writeable
Collecting flask
  Downloading Flask-2.3.2-py3-none-any.whl (96 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 96.9/96.9 KB 177.7 kB/s eta 0:00:00
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Jinja2>=3.1.2
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.1/133.1 KB 236.4 kB/s eta 0:00:00
Collecting blinker>=1.6.2
  Downloading blinker-1.6.2-py3-none-any.whl (13 kB)
Collecting Werkzeug>=2.3.3
  Downloading Werkzeug-2.3.4-py3-none-any.whl (242 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 242.5/242.5 KB 230.8 kB/s eta 0:00:00
Collecting click>=8.1.3
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 96.6/96.6 KB 365.1 kB/s eta 0:00:00
Requirement already satisfied: MarkupSafe>=2.0 in /usr/lib/python3/dist-packages (from Jinja2>=3.1.2->flask) (2.0.1)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
  WARNING: The script flask is installed in '/home/administrateur/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.2 Werkzeug-2.3.4 blinker-1.6.2 click-8.1.3 flask-2.3.2 itsdangerous-2.1.2
```

Créer un dossier sur le PC qui va contenir le projet

ensuite dans ce dossier on va créer un fichier app.py

Configuration du serveur

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello():
    return "Hello world !"
app.run(host='localhost', port=5000)
```

Faire un cd dans le dossier créé

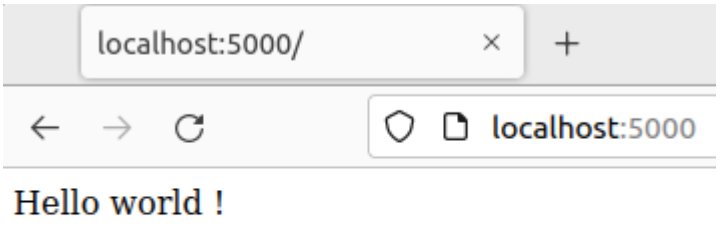
```
export FLASK_APP=app.py
export FLASK_ENV=development
```

```
administrateur@rt-mv:~/R209$ set FLASK_APP=app.py
administrateur@rt-mv:~/R209$ set FLASK_ENV=development
administrateur@rt-mv:~/R209$ export FLASK_APP=app.py
administrateur@rt-mv:~/R209$ export FLASK_ENV=development
```

sudo apt install python3-flask
flask run

```
administrateur@rt-mv:~/TP$ flask run
 * Ignoring a call to 'app.run()' that would block the current 'flask' CLI command.
   Only call 'app.run()' in an 'if __name__ == "__main__"' guard.
 * Serving Flask app 'app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

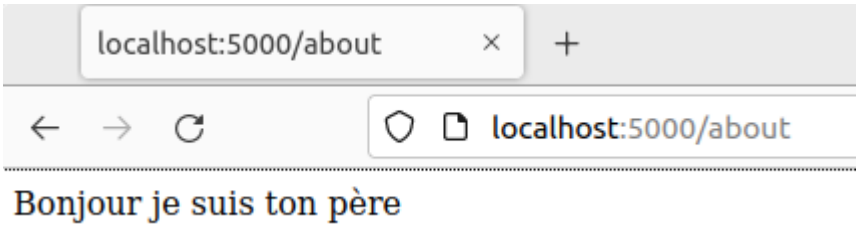
http://localhost:5000/



Création des premières routes

Route about

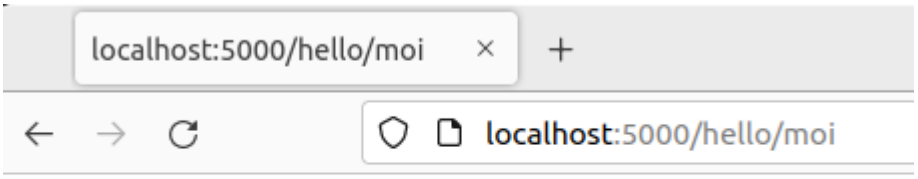
```
  GNU nano 6.2
from flask import Flask
app = Flask(__name__)

@app.route('/')

def hello():
        return "Hello world !"

@app.route('/about')

def about():
        return  "Bonjour, je suis noir !"

app.run(host='localhost', port=5000)
```

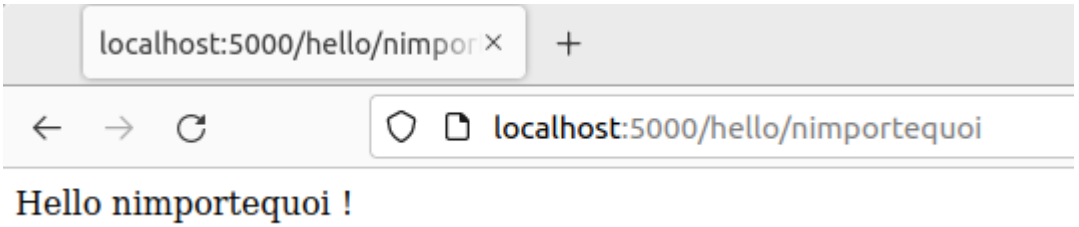


Route hello

```
from flask import Flask
app = Flask(__name__)
@app.route('/')

@app.route('/about')
def about():
        return 'Bonjour je suis Baptiste'

@app.route('/hello')
@app.route('/hello/<name>')
def hello(name = None):
        if name is None:
                return 'Hello world !'
        return 'Hello ' + name + ' !'
app.run(host='localhost', port=5000)
```

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

@app.route('/')

@app.route('/about')

def about():
        return 'Bonhour je suis Baptiste'

@app.route('/hello')
@app.route('/hello/<name>')

def hello(name = None):
        if name is None:
                return 'Hello world !'
        return 'Hello ' + name + ' !'

app.run(host='localhost', port=5000)
```





API et Database
Installation package db, importation et configuration

pip install mysql-connector-python

```
administrateur@rt-mv:~/projet209$ pip install mysql-connector-python
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.0.33-cp310-cp310-manylinux1_x86_64.whl (27.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 27.4/27.4 MB 709.1 kB/s eta 0:00:00
Requirement already satisfied: protobuf<=3.20.3,>=3.11.0 in /usr/lib/python3/dist-packages (from mysql-connector-python)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-8.0.33
```

Pour savoir si ça a marché :

import mysql.connector as MC

try:

conn = MC.connect(host= 'localhost', database = 'python', user = 'root', password = 'root')
except MC.Error as mcErr:
print (mcErr)

```
from flask import Flask
app = Flask(__name__)
import mysql.connector as MC

@app.route('/')

@app.route('/about')

def about():
        return 'Bonhour je suis Baptiste'

@app.route('/hello')
@app.route('/hello/<name>')

def hello(name = None):
        if name is None:
                return 'Hello world !'
        return 'Hello ' + name + ' !'

try:
        conn = MC.connect(host= 'localhost', database = 'python', user = 'root', password = 'root')
except MC.Error as mcErr:
        print (mcErr)

app.run(host='localhost', port=5000)
```

Création de notre db

sudo apt install mysql-server

- id : int auto_increment primary key
- pseudo : varchar(25)
- password : varchar(255)

Si vous souhaitez vous connectez à mysql via un invité de commande voici la commande :

```
mysql -u root -p
```

```
# connexion à la DB
conn = MC.connect(host= 'localhost', port = 3307, database = 'python', user =
'root', password = 'root')

# création du cursor
cursor = conn.cursor()
```

```
# création de notre requête
req = 'SELECT * FROM user'

# lancement de la requête
cursor.execute(req)

# affichage des résultats
results = cursor.fetchall()for res in results: print ('Pseudo : {}'.format(res[1]))
```

```
import mysql.connector as MC
from flask import Flask
app = Flask(__name__)
@app.route('/')

@app.route('/about')
def about():
        return 'Bonjour je suis ton père'

@app.route('/hello')
@app.route('/hello/<name>')
def hello(name = None):
        if name is None:
                return 'Hello world !'
        return 'Hello ' + name + ' !'


# connexion à la DB
conn = MC.connect(host= 'localhost', port = 3306, database = 'mysql', user = 'mathys', password = '1234')
# création du cursor
cursor = conn.cursor()
# création de notre requête
req = 'SELECT * FROM user'
# lancement de la requête
cursor.execute(req)
# affichage des résultats
results = cursor.fetchall()
for res in results:
        print ('Pseudo : {}'.format(res[1]))



app.run(host='localhost', port=5000)
```

Ubuntu Software

```
administrateur@rt-mv:~/R209$ sudo flask run
Pseudo : debian-sys-maint
Pseudo : mathys
Pseudo : mysql.infoschema
Pseudo : mysql.session
Pseudo : mysql.sys
Pseudo : root
Pseudo : votre_utilisateur
 * Ignoring a call to 'app.run()' that would block the current 'flask' CLI comma
nd.
   Only call 'app.run()' in an 'if __name__ == "__main__"' guard.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

```
F 1 <!DOCTYPE html>
2 <html lang="en">
<! - - CV- -> «section class="content-section bg-primary text-white
text-center" id="services">
<div class="container px-4 px-lg-5">
```

```
<div class=" content-section-heading">
© IN 7
8
<h2 class="mb-5">Mon CV</h2>
<img src=" (f url_for ('static', filename= 'Images/CV.png') ]]"
alt="CV"
class="image-size">
10
] Te
11
12
日V
13
<center>
<a href-"{f url_for ('static', filename='files/cV.pdf') j]" download>
<img sre=" (f url_for('static' , fillename= 'Images/DL.png') \J"
class="icone-size"/
```

https://www.php.net/downloads.php

https://httpd.apache.org/download.cgi

```
wget https://archive.apache.org/dist/httpd/httpd-2.4.46.tar.gz
tar -xzvf httpd-2.4.46.tar.gz


wget https://www.php.net/distributions/php-7.4.23.tar.gz

tar -xzvf php-7.4.23.tar.gz

sudo apt update
sudo apt install php

sudo dnf install php
```