

# Tutoriel bdw-server

Fabien Duchateau (Université Claude Bernard Lyon 1) - 2025



- 1.** Le serveur **bdw-server** permet de développer en local des sites web avec Python, Jinja et PostgreSQL. Suivez les instructions de l'aide (section 3.1) afin d'**installer bdw-server** :

<https://perso.liris.cnrs.fr/fabien.duchateau/bdw>

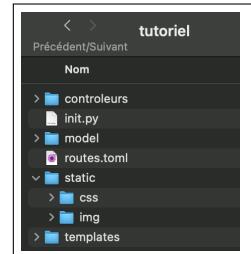
Une fois installé, lancez le serveur avec la commande ci-contre : un message indique qu'il faut fournir un répertoire en paramètre.

```
source .venv/bin/activate # sous
                           linux, macos
                           (.venv\Scripts\activate sous
                           windows)
python server.py
```

```
(venv) $ python server.py
usage: server.py [-h] [-c CONFIG.DB] [-i INIT] [-n] [-p PORT] [-r ROUTES]
                 [-s SCHEM] [-t TEMPLATES] directory
server.py: error: the following arguments are required: directory
```

- 2.** Créons maintenant **un nouveau site**, avec les répertoires et fichiers requis a-minima (*boiler-plate*). Écrivez la commande ci-contre dans un terminal (avec l'environnement virtuel activé). Avec l'option **-b**, le serveur va créer le répertoire du nouveau site, ici **websites/tutoriel**. Ce répertoire contient désormais plusieurs fichiers et répertoires.

```
python server.py -b
websites/tutoriel
```



- 3.** Le serveur se base sur l'architecture **MVC** avec routage. Pour créer une nouvelle page, il est nécessaire d'**ajouter une nouvelle route**. Ouvrez le fichier **routes.toml** et copiez-y le code ci-contre. Il spécifie le chemin des fichiers contrôleur et vue pour l'URL de base.

```
[[routes]]
url = ""
controleur = "controleurs/hello.py"
template = "templates/hello.html"
```

- 4.** Créons maintenant les deux fichiers spécifiés. **Créez le fichier** **controleurs/hello.py** avec la ligne ci-contre. Un message est stocké dans la variable **REQUEST\_VARS** afin qu'il soit accessible dans le template<sup>1</sup>.

**Créez le fichier** **templates/hello.html**, avec les lignes de code correspondantes qui consistent à afficher un titre et le message.

Enfin, vous pouvez **lancer le serveur** en exposant votre site pour visualiser la page. L'option **-n** permet de ne pas se connecter à une BD.

<sup>1</sup>Note : le message "hello world!" aurait pu être codé directement dans le template (et le contrôleur serait vide).

```
REQUEST_VARS['message'] = 'Hello
world !'
```

```
<h1>Tutoriel</h1>
<p>{{ REQUEST_VARS['message'] }}</p>
```

```
python server.py -n
websites/tutoriel
```

# Tutoriel

Hello world !

## En résumé

Ce tutoriel est volontairement limité pour en faciliter la prise en main. Des détails supplémentaires (notamment l'utilisation de la partie modèle) sont donnés dans les diapositives de cours, sur le site démo *serial-critique* fourni avec le serveur, ou dans les TP de programmation web (<https://perso.liris.cnrs.fr/fabien.duchateau/>).