Démarrer un projet Django

Un projet peut contenir plusieurs applications

# Installation de python

## Téléchargement et installation sur Windows

Télécharger python version 3.6.x (du moins c’est la version que j’utilise avec mes projets et en raison de compatibilité avec MySQL que je vais décrire un plus loin dans ce tutoriel.) sur le site officiel de python et l’installer sur Windows

## Vérification

Vérifier à l’aide de la console si python a été bien installé en y tapant la commande suivante :

python -V

Vérifier si pip est présent aussi, en tapant la commande suivante, normalement, pip est déjà présent dans les versions de python >3.4

pip -V

NB : pip permet d'installer facilement le framework Django, c’est un gestionnaire de paquet comme npm pour angular par exemple.

# Installation du Framework Django

## Création d’un environnement virtuel

Premièrement, il faut créer un dossier ou par exemple on veut placer tous les projets Django,

On peut tout de suite passer à l’installation de Django mais ce sera difficile de gérer si on veut utiliser plusieurs versions de Django dans différents projets.

La bonne pratique c’est de créer un environnement virtuel pour chaque version de Django qu’on veut utiliser. Après la création d’un environnement virtuel, il faudra d’abord l’activer avant de se lancer dans l’installation d’une version de Django. L’emplacement de l’environnement virtuel n’est pas forcément dans le dossier qui contient le projet Django.

### 1ère alternative pour la création d’un environnement virtuel :

Lancer la commande suivante, et un dossier au nom de l’env virt sera créée dans le dossier spécifié pour l’installation

python -m venv nom\_environnement\_virtuel

Cette commande permet de lancer une commande python qui va charger un module "venv" et qui va permettre l'utilisation de notre environnement virtuel.

### 2ème alternative :

Si la commande précédente ne marche pas, on a la possibilité d’utiliser cette autre commande, avec pip installé :

Pip install virtualenv

Une fois le module « virtualenv » installé, lancer cette commande pour créer un environnement virtuel :

Virtualenv nom\_environnement\_virtuel

## Activation de l’environnement virtuel

Après avoir créé l’environnement virtuel, il faut l’activer avec la commande suivante,

Cette commande sera lancée depuis l’emplacement du dossier contenant l’env virt

nom\_environnement\_virtuel\Scripts\activate

et pour désactiver l’environnement, il faut juste taper la commande suivante

deactivate

## Installation de django

Lorsque l’environnement virtuel est activé, on peut installer Django avec la commande suivante.

pip install Django

Pour vérifier que Django a bien été installé, on peut exécuter les commandes suivantes dans un invite de commande

python

>>> import django

>>> print(django.get\_version())

Une fois django installé, il faut se mettre dans le répertoire (à l’aide de la console ou naviguer avec l’explorateur de windows) qu’on veut placer le projet, lancer la commande suivante pour créer un nouveau projet Django.

django-admin startproject nom\_site

* Le **mysite/**répertoire racine externe est simplement un conteneur pour votre projet. Son nom n'a pas d'importance pour Django; vous pouvez le renommer comme bon vous semble.
* **manage.py**: Un utilitaire de ligne de commande qui vous permet d’interagir de différentes manières avec ce projet Django. Vous pouvez lire tous les détails sur **manage.py**dans [django-admin et manage.py](https://docs.djangoproject.com/en/2.1/ref/django-admin/) .
* Le **mysite/**répertoire interne est le package Python réel de votre projet. Son nom est le nom du paquet Python que vous devrez utiliser pour importer n'importe quoi à l'intérieur (par exemple **mysite.urls**).
* **mysite/\_\_init\_\_.py**: Un fichier vide qui indique à Python que ce répertoire doit être considéré comme un paquet Python. Si vous êtes un débutant en Python, en savoir [plus sur les paquets](https://docs.python.org/3/tutorial/modules.html#tut-packages) dans la documentation officielle de Python.
* **mysite/settings.py**: Paramètres / configuration pour ce projet Django. [Les paramètres Django](https://docs.djangoproject.com/en/2.1/topics/settings/) vous diront tout sur le fonctionnement des paramètres.
* **mysite/urls.py**: Les déclarations d'URL pour ce projet Django; une "table des matières" de votre site propulsé par Django. Vous pouvez en savoir plus sur les URL dans le [répartiteur d'URL](https://docs.djangoproject.com/en/2.1/topics/http/urls/) .
* **mysite/wsgi.py**: Un point d’entrée pour les serveurs Web compatibles WSGI pour servir votre projet. Voir [Comment déployer avec WSGI](https://docs.djangoproject.com/en/2.1/howto/deployment/wsgi/) pour plus de détails.

Le serveur de développement [**¶**](https://docs.djangoproject.com/en/2.1/intro/tutorial01/#the-development-server)

Vérifions votre projet Django fonctionne. Passez dans le **mysite** répertoire externe, si ce n'est déjà fait, et exécutez les commandes suivantes :

python manage.py runserver

tutoriel site Django

Module -> fichier avec extension .py ou on peut grouper des opérations, fonctions..

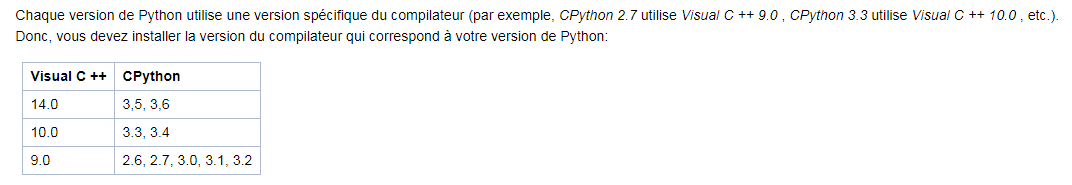
Tout fichier python est un module, on peut regrouper des classes …

Package -> un dossier dans le quelle on doit forcément trouver le fichier \_\_init\_\_.py

Création de projet mysite (ce dossier contient un fichier python manage.py, c’est la racine de notre projet)

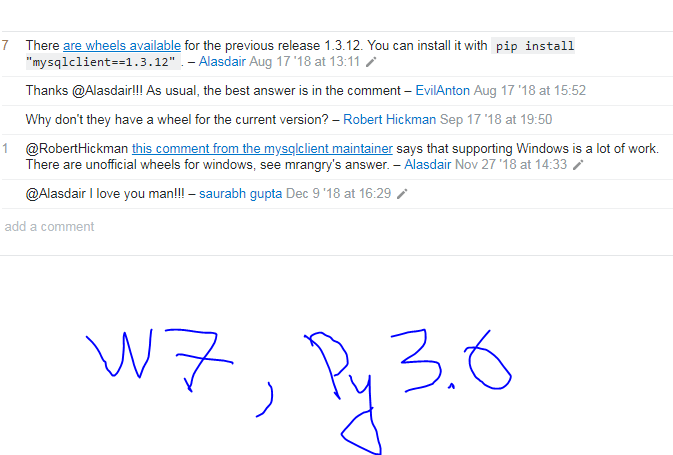
Utilisation base de données mysql dans django

Version de compilateur C++ utilisé pour une version de python



Il faut installer python 3.6 et puis installer mysqlclient par la commande suivante :

pip install "mysqlclient==1.3.12



Base de données avec MySql

1. CREER UNE NOUVELLE BASE DE DONNEES.

Voir l’url suivant pour le tutoriel complet : <https://www.leeboonstra.com/Django/setup-a-django-project-with-a-mysql-database/>

Il faut installer mysqlclient pour que Django puisse se connecter à MySql

NB : après installation de wamp ou xamp, et après sa mise en marche, il faut installer mysqlclient avec la commande pip, pour que django puisse se connecter à la base de données mysql,

Pip install mysqlclient

Ou si cette commande génère une erreur, on peut exécuter la commande suivante

pip install --only-binary :all: mysqlclient



La création de base de données peut se faire soit depuis l’interface de phpmyadmin soit depuis une console mysql. Si on veut créer un utilisateur et de lui conférer tous les privilèges sur une base de données existante, il faut passer par la console mysql. Sinon, l’utilisateur par défaut de wamp est « root » et le mot de passe est nul.

* Créer une base de données + utilisateur avec tous les privilèges sur la base, à executer dans une console mysql

CREATE DATABASE mydatabase;  
CREATE USER 'username'@'localhost' IDENTIFIED BY 'your\_password';  
GRANT ALL PRIVILEGES ON mydatabase.\* TO 'username'@'localhost';  
FLUSH PRIVILEGES;

NB : il ne faut pas utiliser ‘admin’ ou ‘root’ comme username , ainsi que le mot de passe. Si jamais durant la création de base de données, il y une ou des erreurs, il vaut mieux de refaire la démarche jusqu’à ce que cela se termine avec succès.

* Une fois la base de données créée, il faut renseigner dans settings.py le nom de la base de données ainsi que les paramètres nécessaires pour le fonctionnement.
* Cela fait, créer des modèles dans models.py et ensuite une par une les commandes suivantes depuis prompt (cmd)

python manage.py makemigrations nom\_appli

python manage.py sqlmigrate nom\_appli 0001

python manage.py migrate

* Vérifier ensuite dans la base MySQL si la création ou, modification a bien été prise en compte.

1. MODIFIER UN MODELE EXISTANT.

Ce sera toujours les mêmes commandes qu’on utilise, mais cette fois-ci, il ne faut plus créer une base donnée, après modification des modèles, il faut juste lancer les commandes suivantes :

python manage.py makemigrations nom\_appli

python manage.py sqlmigrate nom\_appli 0001

python manage.py migrate

ou python manage.py migrate --run-syncdb

1. UTILISATION D’UNE BD EXISTANTE.

La base de données est déjà existante et elle possède des tables,

Commande permettant d’inspecter la base de données qu’on a créée depuis phpmyadmin ou autre SGBD

python manage.py inspectdb

Commande permettant d’enregistrer l’inspection effectuée depuis la base de données dans un module models.py,

python manage.py inspectdb > models.py

Enregistrer et appliquer les modifications effectuées pour les models,

NB : Il faut voir la documentation officielle pour plus d’explication.

1. MODIFICATION D’UN MODELE CONTENANT DEJA DES DONNEES.

« Il me semble qu'il n'y a pas de problème pour ajouter des tables lorsque tu crées un nouveau modèle. Par contre si tu modifies un modèle, il te faut sauvegarder tes données (dumpdata), supprimer ta base puis la recréer (syncdb) et reinserer tes données (loaddata). »

<https://docs.djangoproject.com/fr/2.1/ref/django-admin/>

django-admin dumpdata [app\_label[.ModelName] [app\_label[.ModelName] ...]][**¶**](https://docs.djangoproject.com/fr/2.1/ref/django-admin/#django-admin-dumpdata)

Résumé processus d’un nouveau projet Django

Raccourci pour monter tous les commandes django :

Python manage.py

Supposons qu’on a déjà crée un dossier pour mettre le projet django et que python et pip sont déjà installé sur l’ordinateur, ainsi que wamp ou autre

Création de virtual environnement

Python -m venv nom\_virtuel\_environnement

Activation venv

Nom\_venv\_\scripts\activate

Désactivation de venv

Nom\_venv\scripts\deactivate

Installation de django (la dernière version sera installé)

Pip install django

Installation de django, une version précise

Pip install django==2.1.7

Installation de djangorestframework (si on veut utiliser cela)

Pip install djangorestframework

Install de connecteur mysql

Pip install mysqlclient

S’il y a erreur due par le manque de Visual C++, exécuter la commande suivante :

pip install --only-binary :all: mysqlclient

Création de nouveau projet

Django-admin statproject nom\_projet

Création nouvelle application dans le projet

Python manage.py startapp nom\_app

Les packages à installer pour des utilisations spécifiques

Pour pouvoir manipler image : pip install pillow

Editeur de texte riche en html : pip install ckeditor

Effacer fichier lors suppression instance modèle : pip install django\_cleanup

Imprimer pdf xhtml2pdf

12/09/2019

Git add .

Git commit -m "commentaire de modification"

Git push