Manuel technique

développement de site web en python à l’aide du cadriciel Django

**SOMMAIRE:**

Technologies utilisées et versions 2

Hébergement accès et login du projet 2

Architecture MVT de django 3

Installation et déploiement 4

Installation et configuration d’un projet django 4

Installation et configuration d’un projet django sur pc sans droit d’administration 5

Déploiement sur pythonanywhere 6

Standard de développement 8

Technologies utilisées et versions:

**Langage de programmation:**

-python 3.7

**Libraries python installées (pip install):**

-psycopg2

-mysqlclient

-Pillow (nécessaire pour l’utilisation de l’attribut “ImageField”)

-shopping-cart-django

**Langages informatiques:**

-html5

-css3

**Système de gestion de base de données:**

-MySQL

**Cadriciel web:**

-Django 3.1.1

**Hébergement & accès:**

-pythonanywhere (http://sgperformancescustoms.pythonanywhere.com)

-compte administrateur: email->[sgperformancescustoms@gmail.com](mailto:sgperformancescustoms@gmail.com) mot de passe->Motdepasse1234

-compte client: email-> mot de passe->Motdepasse1234

**Architecture MVT de Django**

**Models, Views, Templates**



Static



**Templates**





Client



**Vues**

Requête http



Réponse http

**Modèles**

Serveur web

Urls

ORM



Base de données

**Installation et déploiement:**

Environnement de développement intégré : pycharm professional edition 2020.2 (**NB**: licence professionnelle nécessaire)

marche à suivre d’installation de django

## Créer un environnement virtuelle

Installation : pip install virtualenv

Création : virtualenv -p $env :python3 env ou py -m venv env

## Activé l’env

env\scripts\activate.bat

## Désactivé l env

deactivate

## Installer django

Pip install django

## Pour créer la structure d’un project

django -admin startproject <NomDuProjet>

## Pour installer la librairie qui permet de travailler avec MySQL (psycopg2)

pip install psycopg2

pip install mysqlclient

changer les paramètres de la databases dans settings.py (port: 3306, host: 127.0.0.1)

Déterminer le chemin :

* set PATH=%PATH%; (puis le chemin ci dessous)
* C:\Users\micha\Downloads\mysql-8.0.20-winx64\mysql-8.0.20-winx64\bin(varie en fonction de ton ordi)

## démarrer le serveur

* Copier les information dans settings.py
* Aller à la racine du project et taper la commande : python manage.py migrate
* Ensuite lancer le serveur : python manage.py runserver

## Organisation d’un projet django

* **\_\_init\_\_.py** : Déclare ce répertoire comme étant un module. Cela vous permettra d'importer les réglages ailleurs.
* **settings.py** : Les réglages généraux du projet. Nous détaillerons ce fichier juste après.
* **urls.py** : Les routes utilisées dans votre projet. Nous expliciterons ce concept plus tard dans le cours mais retenez que tous les "chemins" que vous entrez dans votre navigateur pour accéder à une page de votre application sont regroupés ici.
* **wsgi.py** : Réglages du serveur web.

## Créer une application

django-admin startapp nom

## Pour utiliser l’invite de commande mysql dans pycharm

mysql -u *nom d’utilisateur* -p

## Opération courantes sur la database (commandes)

<https://code.uni-deal.com/techniques/linux/62-mysql-commandes-utiles.html>

Marche à suivre d’installation de django sur pc sans droit d'administrateur (PC de l’école):

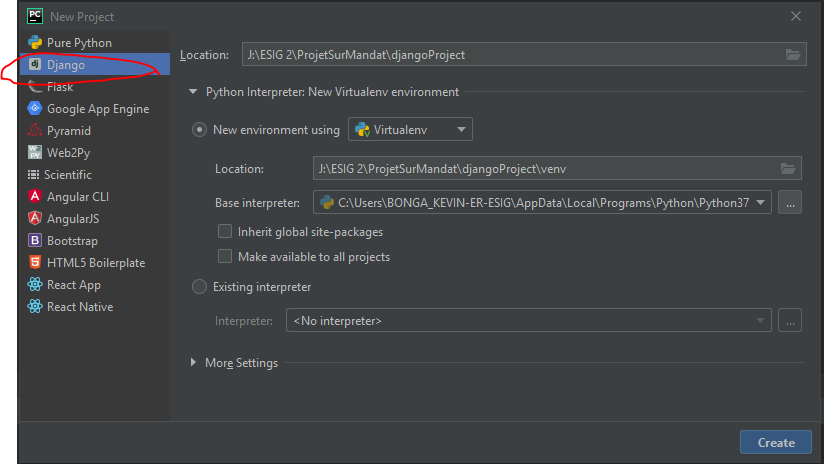
Logiciels recommandés

-PyCharm professional edition -> <https://www.jetbrains.com/pycharm/download/#section=windows>

-MySQL workbench

-Python 3.7

Créer un nouveau projet

-Sélectionner django

-Sélectionner un emplacement local sur le disque (dossier dans lequel le projet sera enregistré)

-Sélectionner l’option « New environment using » puis « Virtualenv »

-Cliquer sur les 3 points de « Base interpreter »

-Sélectionner le SDK dans Users->Dossier->Personnel->ApplicationData (Renommer « ApplicationData » en « AppData »)->Local->Programs->Python->Python37->Python.exe

Déploiement du projet:

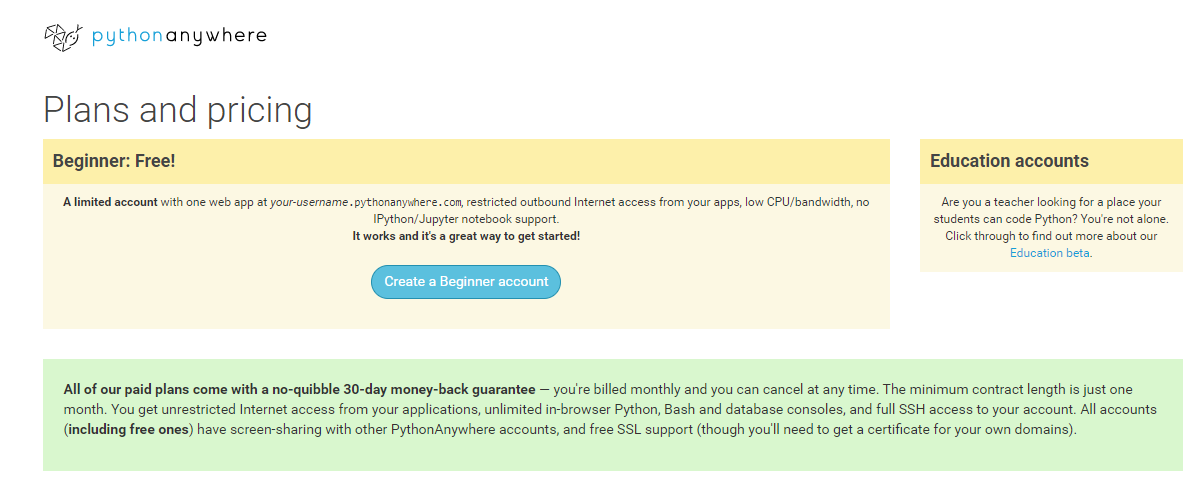
-déploiement pour la v1.0 -> hébergement gratuit avec pythonanywhere pour la démonstration de l’application web

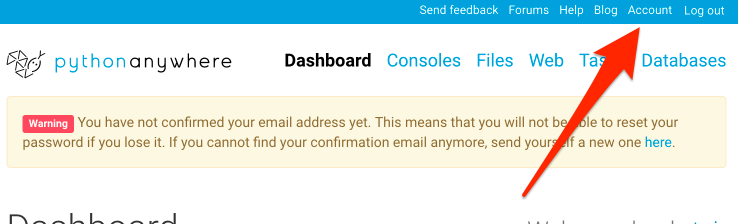
-déploiement pour la v2.0 -> hébergement gratuit avec pythonanywhere pour la démonstration de l’application web

-déploiement pour la v2.1 -> serveur non managé sur Infomaniak (si mandant répond) solution B hébergement gratuit sur pythonanywhere

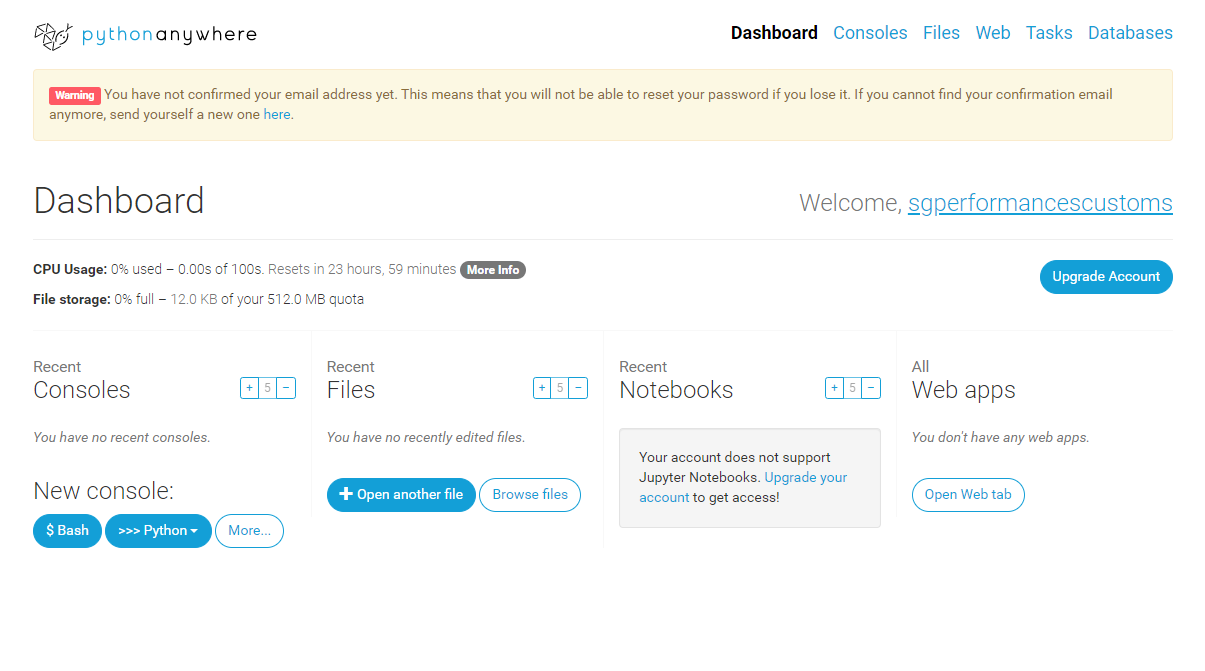
Marche à suivre pour le déploiement (**NB:** zipper le projet django au préalable)

-créer un compte gratuit sur <https://www.pythonanywhere.com/pricing/>

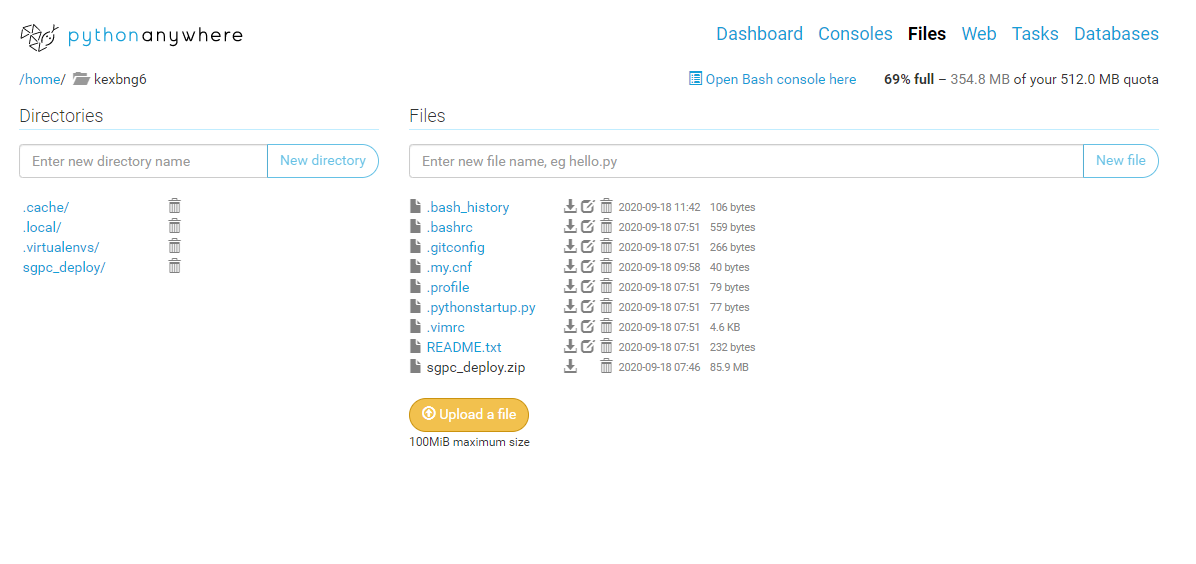


-générer un jeton API dans la partie API token sous l’onglet Account

-ouvrir une nouvelle console bash dans l’interface



-uploader le fichier zippé contenant le projet django

-créer un environnement virtuel -> “mkvirtualenv <NomDeL'Environnement> --python=/usr/bin/python3.7”

-taper la commande “unzip <nomDuFichierZippé>”

-dans la partie Virtualenv du même onglet, insérer le nom de l’environnement virtuel

-une fois que le dossier est dézippé, se déplacer dans le dossier du projet dans la console bash (cd <NomDuProjet>) puis utiliser la commande “pip install -r requirements.txt”

-dans l’onglet web dans la partie code cliquer sur le fichier de configuration wsgi et introduire le code suivant:

# This file contains the WSGI configuration required to serve up your

# Django app

import os

import sys

# Add your project directory to the sys.path

settings\_path = '/home/<username>/<cheminJusquAManage.py/'

sys.path.insert(0, settings\_path)

# Set environment variable to tell django where your settings.py is

os.environ['DJANGO\_SETTINGS\_MODULE'] = <NomDuProjetDjango>.settings'

# Set the 'application' variable to the Django wsgi app

from django.core.wsgi import get\_wsgi\_application

application = get\_wsgi\_application()

-dans le dashboard cliquer sur l’onglet web, puis cliquer sur le bouton reload, puis sur le lien du site web pour se rendre sur le site web déployé en ligne

## Standard de développement:

**Gestion de version :** Git au travers des interfaces graphiques Source Tree et Bitbucket

Chaque membre du groupe développe les éléments du projet dont il est responsable en “local”. Lorsqu’il faut intégrer ces éléments sur le repository du projet nous utilisons deux variantes:

1. copier-coller le code d’un membre du groupe sur un pc centralisé qui sert de machine principale et qui sera utilisé pour les commit et les push (méthode permettant d’éviter les conflits)
2. push son code avec source tree et le communiquer aux autres membres pour qu’ils puissent pull. Les autres membres ne doivent modifier aucun fichier avant de pull sous peine de générer des conflits (méthode peu efficace, fastidieuse et pas optimale)

Au cours de la prochaine phase de construction, nous avons l’intention d’utiliser gitignore pour les fichier de cache ainsi que pour le fichier workplace.xml. Ceci nous permettra de push nos commit sans générer de conflits pour les autres membres du groupe.

**Conventions de nommage:**

***-vues:*** le nom des fonctions est écrit en camelCase

***-modélisation:*** les attributs des classes/tables commencent par les trois premières lettres de la classe/table

***-templates:*** le nom des pages html est écrit en camelCase

***-urls:*** les espaces sont remplacés par un tiret du bas