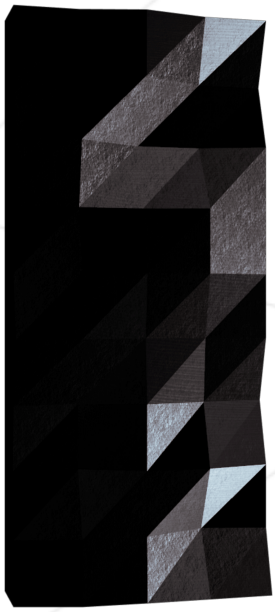

D08 - Mise en production.

English version coming soon!

Summary: *Vous savez maintenant faire un projet django avec toutes les features dont vous rêvez. Il est temps de le partager avec le monde entier!*

Contents

Chapter I

Préambule

Ce préambule sera bref.

...

Voilà.

Chapter II

Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Si aucune information contraire n'est explicitement présente, vous devez partir du principe que les versions des langages et framework utilisés sont les suivantes (ou ultérieures) :
  - `Python 3`
  - `HTML5`
  - `CSS 3`
  - `Javascript EM6`
  - `Django 1.9`
  - `psycopg2 2.6`
- Sauf indication contraire dans le sujet, les fichiers en python de chaque exercice sur `Python` seul (d01, d02 et d03) doivent comporter à leur fin un bloc `if __name__ == '__main__':` afin d'y insérer le point d'entrée dans le cas d'un programme, ou des tests dans le cas d'un module.
- Sauf indication contraire dans le sujet, chaque exercice des journées portant sur `Django` aura sa propre application dans le projet à rendre pour des raisons pédagogiques.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices : seul le travail présent sur votre dépot GIT sera évalué en soutenance.
- Vos exercices seront évalués par vos camarades de piscine.
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement specifiés par les énoncés des exercices.

- Sauf indication contraire dans le sujet vous ne devez pas inclure dans votre rendu :
  - Les dossiers `__pycache__`.
  - Les éventuelles migrations.
    Attention, il vous est tout de même conseillé de rendre le fichier `migrations/__init__.py`, il n'est pas nécessaire mais simplifie la construction des migrations.
    Ne pas ajouter ce fichier n'invalidera pas votre rendu mais vous *devez* être capables de gérer vos migrations en correction dans ce cas.
  - Le dossier créé par la commande `collectstatic` de `manage.py` (avec pour chemin la valeur de la variable `STATIC_ROOT`).
  - Les fichier en bytecode Python (Les fichiers avec une extension en `.pyc`).
  - Les fichiers de base de donnée (notamment avec `sqlite`).
  - Tout fichier ou dossier devant ou pouvant être créé par le comportement normal du travail rendu.

  Il vous est recommandé de modifier votre `.gitignore` afin d'éviter les accidents.
- Lorsque vous devez obtenir une sortie précise dans vos programmes, il est bien entendu interdit d'afficher une sortie précalculée au lieu de réaliser l'exercice correctement.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle `Google / man / Internet / ...`.
- Pensez à discuter sur le forum Piscine de votre Intra !
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet...
- Par pitié, par Thor et par Odin ! Réfléchissez nom d'une pipe !

Chapter III

Règles spécifiques de la journée

- Votre dépôt de rendu sera à cloner en correction avec la commande suivante:

  `git clone <rendu> ~/d08`

  Vous devrez donc vous assurer que votre configuration (Configuration de serveur notamment) fonctionne à cet emplacement.

Chapter IV

Exercice 00

	Exercise 00
Exercice 00: Une application réutilisable.	
Turn-in directory : *ex*00/	
Files to turn in : `Votre projet, les sources de votre application, votre fichier requirements.txt`	
Allowed functions : `L'ensemble des fonctionnalités de Django, setuptools, pillow`	

Le sujet de cette journée est le passage en production de vos projets!

Cependant il faut bien que vous ayez un projet à servir pour démontrer votre maîtrise de la science des serveurs. Le hasard faisant bien les choses, c'est justement le sujet de cet exercice!

Réalisez une *application* django implémentant une unique page, cette page aura une image de votre choix en background, et un formulaire comportant un champ pour soumettre une image et un champ de texte prenant en paramètre un titre pour l'image en question.

Il sera possible pour un visiteur d'uploader une image et de lui donner un titre. L'ensemble des images ainsi uploadées sera affiché sur la page. Chaque image sera accompagnée de son titre.

Vous devrez également créer un *projet* dans lequel vous intégrerez l'application que vous venez de coder. Vous devrez la construire et l'installer sous forme de package avec `pip`.

Votre projet sera testé avec `DEBUG` mis à `True`, mais vous *DEVEZ* le rendre avec la variable mise à `False`.

Il vous est possible d'ajouter du CSS, d'utiliser bootstrap, des chats, d'invoquer les grands Anciens, tout ce qui vous semblera bon. Mais les consignes précédentes devront

être impérativement respectées.

Soyez également judicieux dans votre manière de gérer chaque type de fichier (`MEDIA`, `STATIC`...). Faites en sorte que ces fichiers soient gérés par le serveur de développement si (et uniquement si) la variable `DEBUG` a pour valeur `True`.

La variable `ALLOWED_HOSTS` ne doit *PAS* avoir pour valeur `['*']`.

Il serait dommage de ne pas pouvoir recevoir de points pour les exercices suivants faute d'avoir réussi celui-ci, n'est-ce pas?
Vous ne devez *PAS* rendre le dossier créé par votre collectstatic.

Chapter V

Exercice 01

	Exercise 01
Exercice 01: Servir le projet.	
Turn-in directory : $ex01/$	
Files to turn in : `Votre projet, votre fichier requirements.txt, vos fichiers de conf de serveur, vos éventuels scripts`	
Allowed functions : `L'ensemble des fonctionnalités de Django, setuptools, Nginx, gunicorn, uWSGI`	

Votre projet est fait? Eh bien servez-le maintenant!

Utilisez pour cela un "vrai" serveur (Nginx), n'utilisez *PAS* le serveur de développement de Django.

Commencez par servir le projet en lui-même, sans vous préoccuper des assets (fichiers `MEDIA`, `STATIC`...) pour l'instant.

Vous êtes autorisés à réaliser des petits scripts pour faciliter votre gestion des serveurs. Cependant tous vos processus devront tourner en arrière-plan sur l'ensemble de cette journée.

Chapter VI

Exercice 02

	Exercise 02
Exercice 02: Avec les fichiers “STATIC”.	
Turn-in directory : $ex02/$	
Files to turn in : `Votre projet, votre fichier requirements.txt, vos fichiers de conf de serveur, vos éventuels scripts`	
Allowed functions : `L’ensemble des fonctionnalités de Django, setuptools, Nginx, gunicorn, uWSGI`	

Pas de suspense, faites en sorte de servir vos éléments `STATIC`, comme vos éventuelles feuilles de style par exemple.

Chapter VII

Exercice 03

	Exercise 03
Exercice 03: Avec les fichiers “MEDIA”.	
Turn-in directory : $ex03/$	
Files to turn in : `Votre projet, votre fichier requirements.txt, vos fichiers de conf de serveur`	
Allowed functions : `L’ensemble des fonctionnalités de Django, setuptools, Nginx, gunicorn, uWSGI`	

La même chose maintenant avec les fichiers `MEDIA`.

À ce stade vous devez avoir une conf Nginx vous permettant de servir l’ensemble de votre projet dont la variable `DEBUG` a pour valeur `False`, avec gestion des fichiers `STATIC` et `MEDIA`. Soyez particulièrement attentif à ce que tous les détails fonctionnent.

Chapter VIII

Exercice 04

	Exercise 04
Exercice 04: En HTTPS.	
Turn-in directory : $ex04/$	
Files to turn in : `Votre projet, votre fichier requirements.txt, vos fichiers de conf de serveur`	
Allowed functions : `L'ensemble des fonctionnalités de Django, setuptools, Nginx, gunicorn, uWSGI, SSL.`	

Pour ce dernier exercice faites transiter tout le trafic vers votre site en HTTPS. Vous ouvrirez un port pour le HTTP et un autre pour le HTTPS. Si vous visez le port pour le HTTP, vous serez automatiquement redirigé vers le port en HTTPS.

Un message d'avertissement de la part du navigateur n'est pas gênant dans le cadre de cet exercice si vous utilisez un certificat auto-signé. (Du moment que vous êtes capable d'ajouter une exception).