



D08 - Mise en production.

Dernière étape pour un projet complet !

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

Table des matières

I	Préambule	2
II	Consignes	3
III	Règles spécifiques de la journée	4
IV	Exercice 00	5
V	Exercice 01	7
VI	Exercice 02	8
VII	Exercice 03	9
VIII	Exercice 04	10

Chapitre I

Préambule

Ce préambule sera bref.

...

Voilà.

Chapitre II

Consignes

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- 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. L'url de votre dépôt GIT pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un programme appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises surprises.
- Les exercices shell doivent s'exécuter avec `/bin/sh`.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre dépôt de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !


Chapitre 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.

Chapitre IV

Exercice 00

	Exercice : 00
Exercice 00 : Une application réutilisable.	
Dossier de rendu : <code>ex00/</code>	
Fichiers à rendre : <code>Votre projet</code> , les sources de votre application, votre fichier <code>requirements.txt</code>	
Fonctions Autorisées : L'ensemble des fonctionnalités de Django, <code>setuptools</code> , <code>pillow</code>	

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.

Chapitre V

Exercice 01

	Exercice : 01
Exercice 01 : Servir le projet.	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <code>Votre projet</code> , <code>vos fichier requirements.txt</code> , <code>vos fichiers de conf de serveur</code> , <code>vos éventuels scripts</code>	
Fonctions Autorisées : <code>L'ensemble des fonctionnalités de Django</code> , <code>setuptools</code> , <code>Nginx</code> , <code>gunicorn</code> , <code>uWSGI</code>	

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.

Chapitre VI


Exercice 02

	Exercice : 02
Exercice 02 : Avec les fichiers “STATIC”.	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : Votre projet, votre fichier <code>requirements.txt</code> , vos fichiers de conf de serveur, vos éventuels scripts	
Fonctions Autorisées : L’ensemble des fonctionnalités de Django, <code>setuptools</code> , Nginx, gunicorn, <code>uWSGI</code>	

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

Chapitre VII

Exercice 03


	Exercice : 03
Exercice 03 : Avec les fichiers “MEDIA”.	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <code>Votre projet</code> , votre fichier <code>requirements.txt</code> , vos fichiers de conf de serveur	
Fonctions Autorisées : L’ensemble des fonctionnalités de Django, <code>setuptools</code> , Nginx, <code>gunicorn</code> , <code>uWSGI</code>	

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.

Chapitre VIII

Exercice 04

	Exercice : 04
Exercice 04 : En HTTPS.	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : <code>Votre projet</code> , <code>vos fichier requirements.txt</code> , <code>vos fichiers de conf de serveur</code>	
Fonctions Autorisées : L'ensemble des fonctionnalités de Django, <code>setuptools</code> , <code>Nginx</code> , <code>gunicorn</code> , <code>uWSGI</code> , <code>SSL</code> .	

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).