



Création d'un projet Django

Version 4.1

16 DÉCEMBRE 2021

PAR
TOUTCHY

Le framework Django est un outil puissant et polyvalent pour la création d'applications web. Il est construit sur le langage de programmation Python et permet aux développeurs de créer des applications web à l'aide d'un cadre intuitif et modulaire. Il offre également des fonctionnalités avancées qui peuvent aider à accélérer le développement et à améliorer les performances.

Django est un framework web MVC (Modèle-Vue-Contrôleur). Cela signifie qu'il sépare le modèle (la logique de l'application), la vue (l'interface utilisateur) et le contrôleur (la communication entre les deux). Cela permet aux développeurs de créer des applications web complexes et interactives sans pour autant s'embêter avec des détails techniques complexes.

Sommaire

1. Installation de python et de l'environnement virtuel
2. Installation de Django et des dépendances nécessaires
3. Mise en ligne heroku

Installation de python et de l'environnement virtuel

Python 3 est la dernière version de la langue de programmation populaire et puissante, et est très utile pour créer des applications, des sites Web et des scripts. L'installation de Python 3 et de l'environnement virtuel est un processus relativement facile, mais peut être un peu déroutant pour les débutants. Dans cet article, nous expliquerons en détail comment installer Python 3 et configurer un environnement virtuel sur votre ordinateur.

Tout d'abord, vous devez télécharger la version désirée de Python 3. Vous pouvez le faire à partir du site Web [Python.org](https://www.python.org). Une fois le téléchargement terminé, exécutez le fichier et suivez les instructions pour l'installer sur votre ordinateur. Si vous êtes sous Windows, vous devez cocher la case «Ajouter Python à votre PATH» sinon vous aurez des problèmes pour exécuter vos scripts.

Une fois l'installation terminée, vous devez créer un environnement virtuel. Un environnement virtuel est un espace isolé où vous pouvez installer des modules et bibliothèques sans affecter le système d'exploitation de votre ordinateur. Pour créer un environnement virtuel, ouvrez le terminal et exécutez la commande suivante:

```
python3 -m venv nom-de-votre-environnement
```

Remplacez «nom-de-votre-environnement» par le nom que vous souhaitez donner à votre environnement virtuel. Une fois l'environnement virtuel créé, vous devez l'activer. Pour ce faire, exécutez la commande suivante:

```
source nom-de-votre-environnement/bin/activate
```

Votre environnement virtuel est maintenant activé et prêt à l'emploi. Vous pouvez maintenant installer des modules et bibliothèques sans craindre qu'ils ne soient installés sur votre système d'exploitation. Une fois que vous avez terminé de travailler sur votre environnement virtuel, vous pouvez le désactiver en utilisant la commande deactivate:

```
deactivate
```

C'est tout ! Vous savez maintenant comment installer Python 3 et créer un environnement virtuel sur votre ordinateur. Vous êtes prêt à commencer à coder !

Installation de Django et des dépendances nécessaires

requirements.txt

- asgiref==3.5.2
- cart==1.2.1
- dj-database-url==1.0.0
- Django==4.1
- django-heroku==0.3.1
- Pillow==9.2.0
- pycopg2==2.9.3
- pycryptodome==3.15.0
- pystache==0.6.0
- sqlparse==0.4.2
- static==1.1.1
- whitenoise==6.2.0
- django-taggit==3.0.0
- gunicorn==20.1.0

L'installation de Django et de l'environnement virtuel est une étape importante pour développer et gérer des sites Web efficacement et en toute sécurité. C'est une option très pratique pour les développeurs qui souhaitent séparer leurs projets de manière organisée et sûre. Dans cet article, nous allons vous montrer les étapes nécessaires pour installer Django et l'environnement virtuel.

La première étape est d'installer Python. Python est un langage de programmation très populaire qui est très facile à apprendre et à utiliser. Une fois que Python est installé, vous devez télécharger et installer Django. Vous pouvez le faire en utilisant pip, un outil qui vous permet de télécharger et d'installer les paquets Python.

Une fois que Django est installé, vous devez créer un environnement virtuel. Un environnement virtuel est un environnement séparé qui vous permet d'installer des paquets Python sans affecter les autres projets. Pour créer un environnement virtuel, vous devez utiliser la commande suivante :

```
virtualenv -p python3 myvenv
```

Une fois que l'environnement virtuel est créé, vous devez l'activer. Pour ce faire, vous devez utiliser la commande suivante :

```
source myvenv/bin/activate
```

Une fois que l'environnement virtuel est activé, vous pouvez installer Django. Pour cela, vous devez utiliser la commande suivante :

```
pip install django
```

Une fois que Django est installé, vous pouvez commencer à développer vos projets. Pour vous assurer que votre environnement virtuel est toujours à jour, vous devez vous assurer de mettre à jour Django régulièrement. Pour cela, vous devez utiliser la commande suivante :

```
pip install --upgrade django
```

Vous pouvez également désactiver l'environnement virtuel à tout moment. Pour ce faire, vous devez utiliser la commande suivante :

```
deactivate
```

C'est tout ce qu'il y a à savoir sur l'installation de Django et de l'environnement virtuel. N'hésitez pas à consulter notre documentation et nos tutoriels pour apprendre à utiliser Django et à développer des applications Web.

Mise en ligne avec Heroku python ≥ 3.10

Liste de tâches

- ☐ install requirements
- ☐ Créer le fichier Procfile
- ☐ collectstatic
- ☐ git add/commit/push
- ☐ bash: migrate

requirements.txt

- asgiref==3.5.2
- cart==1.2.1
- dj-database-url==1.0.0
- Django==4.1
- django-heroku==0.3.1
- Pillow==9.2.0
- psycopg2==2.9.3
- pycryptodome==3.15.0
- pystache==0.6.0
- sqlparse==0.4.2
- static==1.1.1
- whitenoise==6.2.0
- django-taggit==3.0.0

- `gunicorn==20.1.0`



Pourquoi et comment passer de Pythonanywhere à Heroku

Heroku est une plateforme de déploiement en ligne qui vous permet de déployer et d'exécuter facilement des applications web. Il offre une solution complète pour vous aider à gérer, déployer, surveiller et maintenir votre application.

Heroku est un service innovant qui offre une variété de fonctionnalités et d'outils pour vous aider à développer et à déployer votre application de manière rapide, facile et efficace. Il s'agit d'une solution complète, car elle vous permet de gérer, de déployer et de surveiller votre application.

Heroku peut fournir des outils tels que des mises à jour automatiques, un système de stockage, des outils de surveillance et des outils de gestion des erreurs.

Quelles différences entre Pythonanywhere et Heroku ?

Si vous êtes un développeur Python qui a l'habitude de coder sur PythonAnywhere, vous avez peut-être entendu parler de Heroku - un service d'hébergement Web qui permet aux développeurs de déployer leurs applications Web rapidement et facilement. Si vous êtes

intéressé par le déploiement de votre application Web sur Heroku, voici un guide complet pour vous aider à passer de PythonAnywhere à Heroku.

Tout d'abord, vous devez comprendre les différences entre PythonAnywhere et Heroku. PythonAnywhere est un service d'hébergement Web qui vous permet de créer et de gérer votre propre serveur Web sur leur plateforme, tandis que Heroku est un service d'hébergement Cloud qui vous permet de déployer facilement et rapidement des applications Web. Heroku est plus facile à utiliser que PythonAnywhere et permet un déploiement plus rapide et plus fiable de vos applications Web.

La prochaine étape consiste à déterminer la manière dont vous allez déployer votre application Web sur Heroku. Vous pouvez le faire en utilisant le service de déploiement intégré d'Heroku ou en configurant un déploiement manuel à l'aide d'un outil comme Git. Si vous choisissez d'utiliser le service de déploiement intégré d'Heroku, vous devrez créer un compte Heroku et vous connecter à votre référentiel de code source (GitHub, Bitbucket, etc.). Une fois que tout est configuré, vous pouvez créer un nouveau projet Heroku et le lier à votre référentiel de code source. Vous pouvez ensuite déployer votre application à l'aide d'Heroku en quelques clics.

Si vous choisissez d'utiliser un déploiement manuel, vous devrez configurer un serveur Git et vous connecter à votre référentiel de code source. Vous devrez ensuite configurer votre environnement Heroku et créer un nouveau projet Heroku. Une fois cela fait, vous pouvez déployer votre application à l'aide de Git en exécutant simplement la commande « `git push heroku master` ».

Après avoir déployé votre application sur Heroku, vous devrez peut-être la relier à une base de données. Heroku offre une variété d'options de base de données, notamment PostgreSQL, MongoDB et Redis. Vous pouvez facilement relier votre application à la base de données de votre choix en utilisant les outils intégrés d'Heroku.

Enfin, vous devrez peut-être paramétrer votre application pour qu'elle s'exécute automatiquement à des moments prédéfinis. Heroku offre un outil appelé « Scheduler » qui vous permet de programmer des tâches qui s'exécuteront automatiquement à des moments prédéfinis.

Python 3 est la dernière version de la langue de programmation populaire et puissante, et est très utile pour créer des applications, des sites Web et des scripts. L'installation de Python 3 et de l'environnement virtuel est un processus relativement facile, mais peut être un peu déroutant pour les débutants. Dans cet article, nous expliquerons en détail comment installer Python 3 et configurer un environnement virtuel sur votre ordinateur.

Tout d'abord, vous devez télécharger la version désirée de Python 3. Vous pouvez le faire à partir du site Web [Python.org](https://www.python.org). Une fois le téléchargement terminé, exécutez le fichier et

suivez les instructions pour l'installer sur votre ordinateur. Si vous êtes sous Windows, vous devez cocher la case «Ajouter Python à votre PATH» sinon vous aurez des problèmes pour exécuter vos scripts.

Une fois l'installation terminée, vous devez créer un environnement virtuel. Un environnement virtuel est un espace isolé où vous pouvez installer des modules et librairies sans affecter le système d'exploitation de votre ordinateur. Pour créer un environnement virtuel, ouvrez le terminal et exécutez la commande suivante:

```
python3 -m venv nom-de-votre-environnement
```

Remplacez «nom-de-votre-environnement» par le nom que vous souhaitez donner à votre environnement virtuel. Une fois l'environnement virtuel créé, vous devez l'activer. Pour ce faire, exécutez la commande suivante:

```
source nom-de-votre-environnement/bin/activate
```

Votre environnement virtuel est maintenant activé et prêt à l'emploi. Vous pouvez maintenant installer des modules et librairies sans craindre qu'ils ne soient installés sur votre système d'exploitation. Une fois que vous avez terminé de travailler sur votre environnement virtuel, vous pouvez le désactiver en utilisant la commande deactivate:

```
deactivate
```

C'est tout ! Vous savez maintenant comment installer Python 3 et créer un environnement virtuel sur votre ordinateur. Vous êtes prêt à commencer à coder !

Bien que l'installation de Django soit généralement assez simple, il est important que les développeurs comprennent ce qu'implique l'installation et quelles sont les dépendances nécessaires pour exécuter le framework.

Tout d'abord, les développeurs doivent s'assurer que leur environnement est prêt à exécuter le framework. Cela signifie qu'ils doivent installer le langage Python sur leur système. Une fois que c'est fait, ils peuvent ensuite commencer à installer Django.

Une fois que Django est installé, les développeurs doivent ensuite installer toutes les dépendances nécessaires pour le faire fonctionner. Cela comprend le package de base de Django, les packages supplémentaires nécessaires, les modules supplémentaires et les bibliothèques.

Une fois que toutes les dépendances sont installées, les développeurs peuvent alors commencer à développer leurs applications web. Cela comprend la configuration de l'environnement, la création de l'application et l'hébergement de l'application web.

Il est important de comprendre que, bien que l'installation de Django soit assez simple, les développeurs ont besoin de comprendre ce qu'implique l'installation et quelles sont les

dépendances nécessaires à l'exécution du framework. Cela leur permettra de tirer le meilleur parti de Django et de créer des applications web robustes et innovantes.