Rapport Django

Introduction

Django est un framework web gratuit et open-source écrit en Python. Il s'agit d'un cadre de haut niveau, ce qui signifie qu'il fournit de nombreuses fonctionnalités prêtes à l'emploi, ce qui facilite le développement d'applications Web. Django est également un framework mature, avec une communauté nombreuse et active. Cela signifie qu'il existe de nombreuses ressources disponibles pour nous aider à apprendre et à utiliser Django.
Dans un premier temps nous allons expliquer comment installer Django ,comment créer un projet Django et une application sur Django.
En deuxième nous allons expliquer la composition des fichiers Django.

I.Comment installer Django ?

Pour installer Django, nous pouvons utiliser le gestionnaire de paquets pip.
pip install django

Une fois Django installé, nous pouvons démarrer un nouveau projet Django en exécutant la commande suivante :

```
django-admin startproject myproject
```

Cela créera un nouveau répertoire appelé myproject. Dans ce répertoire, nous trouverons un certain nombre de fichiers et de répertoires. Les fichiers les plus importants sont :

- settings.py : Ce fichier contient les paramètres de votre projet Django.
- urls.py : Ce fichier contient les URL de votre projet Django.
- wsgi.py : Ce fichier est utilisé pour démarrer le serveur de développement Django.

a.**Créer une application Django**

Une application Django est une collection de code Django utilisée pour effectuer une tâche spécifique. Par exemple, nous pouvons créer une application Django pour gérer les comptes

d'utilisateurs ou pour stocker et récupérer des données à partir d'une base de données.

Pour créer une nouvelle application Django, nous pouvons utiliser la commande suivante :

python manage.py startapp myapp
Cela créera un nouveau répertoire appelé myapp dans le répertoire de notre projet Django. Dans ce répertoire, nous trouverons un certain nombre de fichiers et de répertoires. Les fichiers les plus importants sont :

- models.py : Ce fichier contient les modèles de votre application Django.
- views.py : ce fichier contient les vues de votre application Django.
- urls.py : Ce fichier contient les URL de votre application Django.

Configuration de la base de données

Django prend en charge un certain nombre de bases de données différentes, notamment MySQL, PostgreSQL et SQLite. Pour configurer une base

de données pour notre projet Django, nous devrons créer une base de données puis configurer Django pour l'utiliser.

Exemples de models:

Un modèle est une représentation d'un objet du monde réel dans Django. Par exemple, nous pouvons créer un modèle pour représenter un utilisateur ou un produit. Les modèles sont utilisés pour stocker des données dans une base de données.

Migrations

Les migrations sont utilisées pour suivre les modifications apportées à vos modèles Django. Lorsque nous apportons des modifications à nos modèles, nous devons effectuer des migrations pour mettre à jour la base de données.

Vues

Une vue est une fonction qui gère une demande d'un utilisateur. Par exemple, une vue peut être

utilisée pour afficher une liste de produits ou pour créer un nouveau compte utilisateur.

Modèles ou templates

Un modèle est un fichier utilisé pour afficher des pages HTML. Les modèles Django sont écrits dans un langage spécial appelé Django Template Language (DTL).

URL

Une URL est un identifiant unique pour une ressource sur votre projet Django. Les URL sont utilisées pour acheminer les demandes des utilisateurs vers les vues.

Testez l'application

Une fois que nous avons développé notre application Django, nous devons la tester pour nous assurer qu'elle fonctionne correctement. Django fournit un certain nombre d'outils de test que nous pouvons utiliser pour tester notre application.

Intergiciel

Le middleware est un moyen d'ajouter des fonctionnalités aux applications Django. Le middleware est exécuté entre une requête et une réponse.

Formes

Les formulaires sont utilisés pour collecter des données auprès des utilisateurs. Django fournit un certain nombre de classes de formulaires que nous pouvons utiliser pour créer des formulaires.

Séances

Les sessions sont utilisées pour stocker des données sur l'interaction d'un utilisateur avec une application Django. Les sessions sont stockées dans une base de données.

Sécurité

Django fournit un certain nombre de fonctionnalités qui peuvent être utilisées pour

améliorer la sécurité de nos applications Django. Ces fonctionnalités incluent :

- Protection contre les injections SQL
- Prévention des scripts inter sites (XSS)
- Protection contre la falsification des requêtes inter sites (CSRF)
- Protection contre la force brute
- Authentification et autorisation
- La gestion des erreurs

Administration

Django inclut une interface d'administration intégrée qui peut être utilisée pour gérer nos applications Django. L'interface d'administration nous permet de créer et de modifier des modèles, de gérer des utilisateurs et d'effectuer d'autres tâches.

Utilisation de SQL dans Django

Django peut être utilisé pour accéder et manipuler des données dans une base de données à l'aide de SQL. Django fournit un certain nombre d'outils

qui facilitent l'utilisation de SQL dans les applications Django.

Conclusion

En résumé, Django est un choix solide pour le développement d'applications web robustes, évolutives et sécurisées. Que ce soit pour créer des sites web simples ou des applications web complexes, Django offre les outils nécessaires pour accélérer le processus de développement et garantir la qualité et la maintenabilité du code. Avec sa communauté active et son écosystème riche, Django reste un choix populaire parmi les développeurs web.