**Création de projet Django**

Voici les étapes détaillées pour créer un projet Django :

NB : “**ProjetDjango**” et “**AppDjango**” seront respectivement les noms par défaut de notre projet et de notre application Django

1. Installer Django :

* **"python -m pip install Django"**

1. Créer un projet Django :

* **"python -m django startproject ProjetDjango"**

1. Démarrer le serveur de développement :

* **"cd ProjetDjango"**
* **"python manage.py runserver"**

1. Étapes pour créer une application dans un projet Django :Créer l'application : **a**. Dans le **répertoire principal** de ton projet Django (là où se trouve manage.py), exécute cette commande :

* **"python manage.py startapp AppDjango"**

**b**. Ajouter l'application dans **settings.py** : Ouvre le fichier **settings.py** et ajoute l’application dans la liste **INSTALLED\_APPS** :

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'**AppDjango**', # Ajoute cette ligne

]

**c**. Créer une vue dans views.py : Ouvre le fichier **AppDjango**/**views.py** et crée une vue basique :

from django.http import HttpResponse

def index(request):

return HttpResponse("Bienvenue dans l'application")

**d**. Ajouter les URLs de l'application : Crée ou modifie le fichier **AppDjango**/**urls.py** pour lier la vue à une URL :

from django.urls import path

from . import views

urlpatterns = [path('', views.index, name='index')]

**e**. Inclure les URLs de l'application dans urls.py du projet principal : Ouvre le fichier **urls.py** dans le répertoire principal du projet et ajoute cette ligne pour inclure les URLs de "**AppDjango**"

from django.contrib import admin

from django.urls import include, path

urlpatterns = [

path('admin/', admin.site.urls),

path('mangalib/', include('mangalib.urls')),

]

f. Appliquer les migrations (si nécessaire) : Si tu as des modèles dans l'application, exécute les migrations pour les synchroniser avec la base de données :

- “**python manage.py makemigrations AppDjango**”

- “**python manage.py migrate”**

1. Lancer le serveur et tester : Relance le serveur de développement avec la commande suivante :

* **“python manage.py runserver”**

Ensuite, va sur **http://127.0.0.1:8000/AppDjango/** pour voir ta vue en action.

Ajouter d’un Template à notre rendu dans Django

Pour utiliser des **Template** au lieu d'afficher un simple message dans une vue Django, voici les étapes à suivre. Ce guide explique comment créer et intégrer un Template dans une application Django.

* + - 1. **Créer un dossier pour les Template**

Dans le répertoire de votre application Django, créez un dossier Template. Ensuite, à l’intérieur, créez un sous-dossier portant le nom de l’application pour éviter des conflits entre les Template.

Exemple : **AppDjango** / templates/ **AppDjango** / index.html

* + - 1. **Créer le fichier HTML du Template**

Dans le dossier **templates/ AppDjango** /, créez un fichier index.html. Voici un exemple simple de code HTML que vous pouvez utiliser :

<!DOCTYPE html>

<html *lang*="fr">

<head>

    <meta *charset*="UTF-8">

    <title>MangaLib</title>

</head>

<body>

  <h1 *style*="color: red">{{titre}}</h1>

  <p>{{message}}<p/>

</body>

</html>

* + - 1. **Vérifier la configuration des templates dans settings.py**

Dans votre fichier **settings.py**, vérifiez que Django est configuré pour chercher les templates dans les dossiers d’application. L’option **APP\_DIRS: True** doit être activée pour que Django trouve automatiquement les fichiers dans les dossiers templates des applications :

* + - 1. **Modifier la vue pour utiliser un Template**

Utilisez la fonction render de Django pour rendre le Template et passer les variables nécessaires. Modifiez votre fichier views.py comme suit :

from django.shortcuts import render

def index(*request*):

  contexte = {'titre': 'Bienvenue sur Mangalib',

              'message': 'MangaLib est un site web destiner aux amoureux du Mangas'

  }

  return render(*request*, 'mangalib/index.html', contexte)

Ici, **render** récupère automatiquement le Template **AppDjango /index.html** et insère les données du **contexte** dans le Template.

* + - 1. **Vérifier le mode DEBUG pour le débogage**

Si vous ne voyez pas les changements ou si des erreurs apparaissent, assurez-vous que le mode DEBUG est activé dans le fichier **settings.py**. Cela permet d’afficher des messages d’erreurs détaillés dans le navigateur : **DEBUG = True**

Cela vous aidera à repérer les erreurs éventuelles dans vos fichiers ou vos configurations.