

Comment utiliser Django avec Angular 15 ?

Guide Complet

09/12/2022

Django est le Framework **Python** le plus connu et le plus utilisé.

Dans ce guide complet je vais vous expliquer comment créer une **API** avec Django.

Nous verrons de quel façon nous pourrons l'utiliser avec le **Framework Angular**.



Ce que nous allons faire ?

- **Qu'est ce que Django ?**
C'est un Framework Python nous verrons ce que cela veut dire.
- **Installation de Django et première Application**
Il a été écrit par en 1989 pour simplifier la vie du programmeur.
- **Installation de python sur votre poste de travail**
- Comment installer Python sur Windows et linux et quel éditeur de code choisir ?
- **Création des API**
Une simple application en quelques minutes.
- **Code source**
Le code complet du projet angular-app-multi-api sur github.

Django mais c'est qui, c'est quoi?

Si vous êtes amateur de cinéma comme moi, la première fois que vous avez entendu le nom de Django vous avez sans doute pensé à ça.



Deux sacrés personnages.

Et autant dire que ceux-là faut pas les chercher sinon ça dépote.

Mais Django c'est aussi un nom très connu dans le monde du développement.

Django et Python

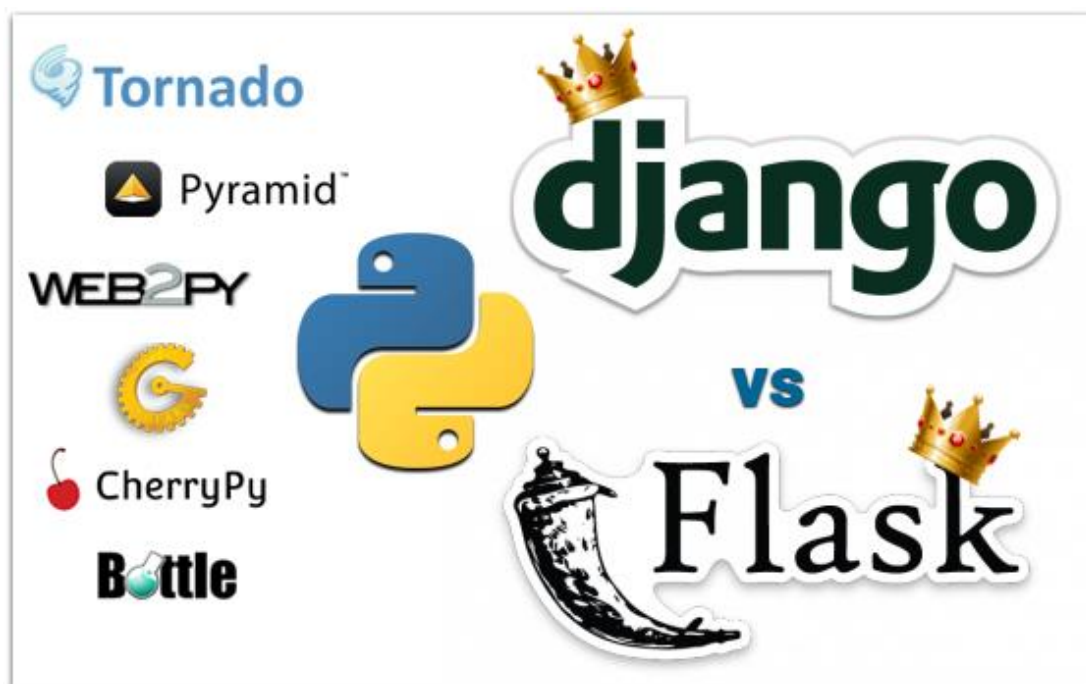
Python est le langage qui connaît la plus forte progression ces dernières années.

Vous avez peut-être entendu parler de Flask, Tornado, CherryPy, Web2Py, Bottle ...

Il existe plusieurs Frameworks Python qui essaient de conquérir le cœur des développeurs.

Mais deux Frameworks règnent actuellement sans partage.

Les rois du monde Python



Django et **Flask** sont les deux Frameworks Python open source les plus connus et les plus utilisés.

Faites un petit tour sur indeed et vous verrez qu'il n'y a pas photo à l'arrivée.

Alors si nous voulons travailler avec Python notre choix est fait.

Dans ce tutoriel nous testerons Django.

Si c'est Flask qui vous intrigue je vous ai concocté un tutoriel dédié c'est ici

- <https://www.ganatan.com/tutorials/flask-avec-angular>

Les origines

Les origines

- **Django**

Il a été développé en **2003** pour le journal local de Lawrence (Kansas)
Par **Adrian Holovaty** et **Simon Willison** qui voulaient passer de PHP à Python.

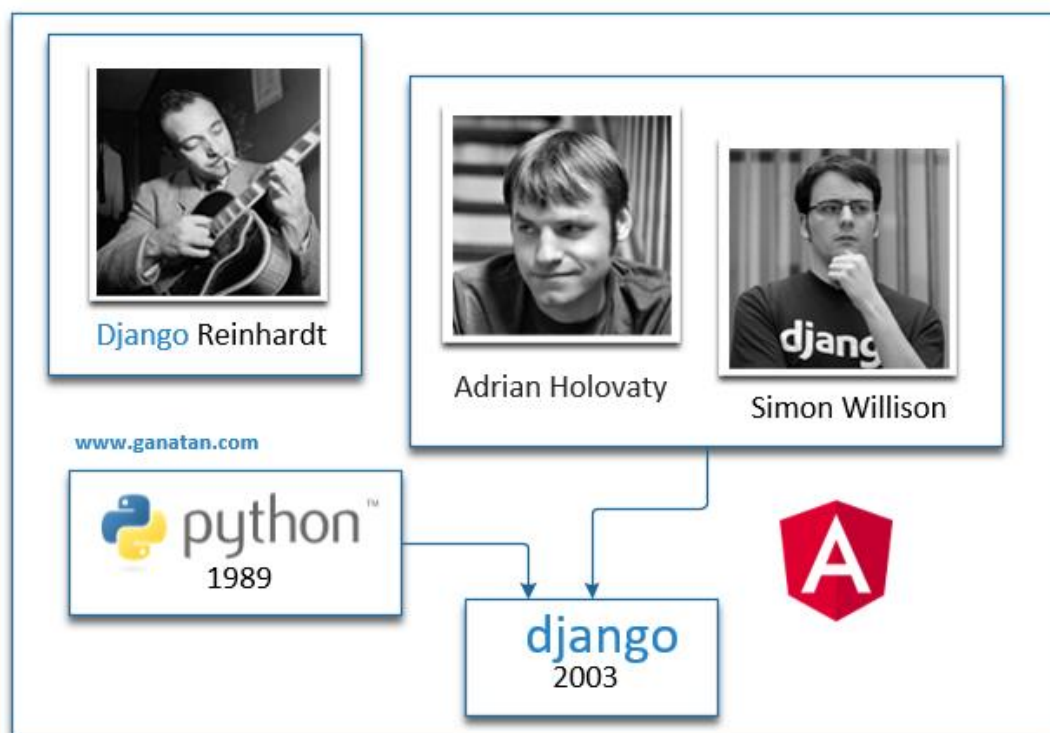
En 2005 il est publié sous licence BSD.

En 2008 il est développé par la Django Software Foundation.

Le nom « **Django** » a été choisi en souvenir de Django Reinhardt.

Ce Django là était sans doute le meilleur guitariste manouche de tous les temps.

Django les origines



Que dit **github** ?

- **Django**
Star : **50800**, Fork : **22000** ,Commits **28528**
<https://github.com/django/django>

Le site officiel

- **Django**
<https://www.djangoproject.com/>

Présentation de Django

Django est donc un framework Python.

Il est open source ce qui signifie que son utilisation est gratuite.

Nous l'utiliserons pour les raisons suivantes.

- C'est le Framework Web Python le plus populaire
- C'est un Framework Full stack (Flask est un microframework)
- C'est un Framework **MVC**
MVC (**M**odel **V**ue **C**ontroller) est un **design pattern** (patron de conception) qui permet de mieux structurer votre code.
- Il dispose d'une administration flexible.

La documentation en Français est accessible ici.

- <https://docs.djangoproject.com/fr/3.0/>

Installation Django

Commençons par installer Django.

Les détails de l'installation sont sur le site officiel
<https://www.djangoproject.com/download/>

Mais c'est simplement comme n'importe quelle bibliothèque Python.
Nous utiliserons le gestionnaire de paquets Python : **pip**

Vous pouvez installer une version spécifique ou la dernière version disponible.

Il suffit enfin de vérifier si django est installé.

```
# Installation d'une version spécifique
pip3 install Django==3.0.8

# Installation de la dernière version disponible
pip3 install Django

# Test de la version de Django installée
python -m django --version
```

Première application Django

Notre but est de créer au final une API avec Django.
Nous utiliserons cette API avec notre frontend Angular.

Nous l'appellerons **api_django**

Nous allons créer notre premier projet.

1. Créer le projet avec la commande django-admin
2. Executer le projet
3. Tester le projet
4. Appliquer la migration
5. Retester le projet sans erreurs

```
# Créez un répertoire demo (le nom est ici arbitraire)
mkdir demo
```

```
# Allez dans ce répertoire
cd demo
```

```
# créer un projet appelé arbitrairement api_django
django-admin startproject api_django
```

```
# Aller dans le répertoire créé
cd api_django
```

```
# Executer le serveur avec l'interpréteur
py manage.py runserver
```

```
# Tester le serveur
http://127.0.0.1:8000/
```

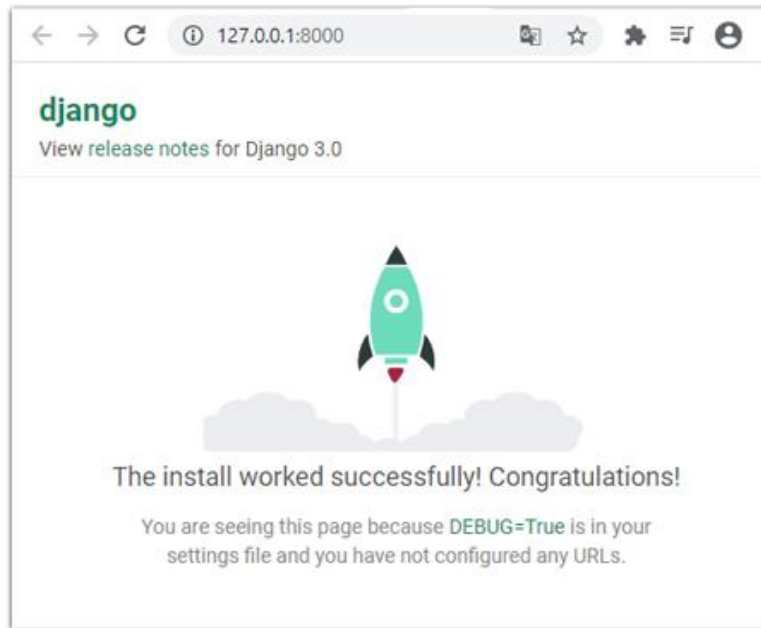
```
# Message d'erreur dans la console
# You have 17 unapplied migration(s)
```

```
# Application de la commande migrate
python manage.py migrate
```

```
# Réexecuter le serveur avec l'interpréteur
py manage.py runserver
```

```
# Retester le serveur
http://127.0.0.1:8000/
```


Résultat dans le navigateur



Passons aux explications.

- La commande **startproject**
Cette commande permet de créer un projet django.
Elle crée automatiquement une structure de répertoires propre au framework django.

Retrouvez plus de détails sur le site officiel

<https://docs.djangoproject.com/fr/3.0/ref/django-admin/#startproject>

- La commande **manage**
Elle a installé une base de données **sqlite**.
La base de données est dans notre exemple dans le fichier **db.sqlite3**

Cette base permettra notamment de gérer l'administration de Django.

Pour accéder à l'administration vous passez par

<http://127.0.0.1:8000/admin>

Si vous voulez manipuler facilement cette base de données (de type sqlite) je vous conseille l'outil suivant.

- **DB Browser for SQLite**
<https://sqlitebrowser.org/dl/>

Architecture Django

Le principe de fonctionnement de django est

- Un **projet** (single project) correspond à un **site web** .
- Un projet peut contenir de multiple **applications (apps)**.
- Une **application** correspond à un **aspect particulier** de notre site web.

Nous allons créer plusieurs applications.

```
django-admin startapp home
```

```
django-admin startapp about
```

```
django-admin startapp contact
```

```
# Création d'un projet  
django-admin startapp projet
```