

Django

Django هو إطار عمل Framework مجاني و مفتوح المصدر لتطوير تطبيقات الويب
يسهل عملية بناء التطبيقات الويب عن طريق توفير هياكل قابلة لإعادة
الاستخدام مثل إدارة المصادقة والتعامل مع قواعد البيانات ومعالجة الطلبات
والاستجابات مما يجعل تطوير الويب أسرع وأكثر كفاءة.
- يوفر مرونة عالية

- يعد Framework مجاني لا deadline
- آمن .. سهل .. سريع

لماذا استخدم Django Framework دوافع باستخدام Frameworks

- سرعة في التطوير (نظام كامل مع إدارة ...)

- Admin panel جاهزة

- سهل التعامل مع قواعد البيانات

- نظام توثيق (ملاحظات) جاهز

- بيئة عمل منظمة "Don't Repeat Yourself"

- MTV (Model, Template, View)

- له مجتمع كبير documents ممتازة

- تطبيقات مشهورة: Instagram

Pinterest

Mozilla

سلسلة تعتبر تقبل سبياً (لا يمكننا استقلاله) مع مشاريع صغيرة أو
بسيطة "Micro services"

لا يفضل استقلاله مع التطبيقات التي تعتمد على Real time

نفضل استقلاله مع التطبيقات الكبيرة التي تتطلب تنظيم للمهام
التي تحتاج وقتاً طويلاً لتنفيذها
ALADIB

MVT أو MTV

M: Model هيكلية البيانات (الجدول ... الخ)
 T: Template العاجزات html ، المتغيرات التي تمر عليها
 V: View business logic (حلقة العمل بين Model و Template)
 والي يحدد كيف يتصرف هذا الموقع

FastAPI

Flask

Django

الأكثر صغراً
 أنظمتين Flask و FastAPI
 بغير خيار DRF
 أفضل مدونة لبيئات realtime
 أخف سرعة من FastAPI
 API "بدوي"
 أفضل مدونة
 متوسط
 سريع جداً
 مدج اجباري بـ تشغيل API
 أكثر ملاءمة

التطبيق العمل:

البيئة الافتراضية: Virtual Environment

كل مشروع بـ Versions الخاصة فيه

1] إنشاء بيئة افتراضية
 python3 -m venv venv
 اسم المجلد

2] لتفعيل البيئة الافتراضية
 cd venv/scripts/activate

3] تنزيل Django داخل البيئة الافتراضية
 pip install django

4] إنشاء مشروع "project"
 django-admin startproject myproject
 اسم المجلد المستهدف

5] الانتقال إلى مجلد المشروع " myproject "

6] لا نعرف الصفحة على د browser
python3 manage.py runserver
أد المشروع .

وظيفة الملفات التي تم إنشاؤها داخل مجلد المشروع :

- manage.py الملف الرئيسي ، سيعالج مع أوامر ال power shell مثل run server

- settings.py إعدادات Configuration المشروع

مثل لغة التطبيق ، الاتصال مع قاعدة البيانات ، ... الخ
urls.py النقطة المركزية لرباط المشروع كله .

- wsgi.py و asgi.py إعدادا للرفع على د server (سنعرف عليهم لاحقاً)

تحت د project في Django يوجد العنصر د apps

7] إنشاء app جديد تحت الproject blog
python3 manage.py startapp blog
اسم مجلد د app

ملاحظة مهمة : د django لا تعرف على هذا د app إلى أن يتم إضافة اسم د app
في ملف settings.py تحت INSTALLED_APPS []

8] ضمن ملف views.py
from django.http import HttpResponse
التي تحت الapp blog

def home (request):

return HttpResponse ("Welcome to Django")

[9] `urls.py` الذي قمنا بإضافته بيانياً داخل المجلد `blog app` #ملف جديد

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('', views.home),
]
```

داخل ملف `urls.py` الذي تحت مجلد المشروع نضيف ما يلي:

```
from django.urls import path, include
[
    path('', include('blog.urls')),
]
```

[10] تحت مجلد `blog app` نضع مجلد جديد `templates` ونضع فيه ملف `home.html`

```
<h1>
    welcome to my site
</h1>
<P1>
</P1>
```

[11] # الصفحة السابقة نطلب تعديل `views.py`

```
from django.shortcuts import render
def home(request):
    return render(request, 'home.html')
```

Models : كلاسز بقايع البيانات Model

[12] `models.py` داخل ملف `blog` app :
`from django.db import models`

```
class Post(models.Model):  
    title = models.CharField(max_length=100)  
    content = models.TextField()  
    created_at = models.DateTimeField(auto_now_add=True)
```

```
def __str__(self):  
    return self.title
```

لايفال ال model كجول ضمتاعة لبيانات :

[13] `python3 manage.py makemigrations` `migration` ملف [13]

[14] `python3 manage.py migrate` `بيشتر` الجول [14]

Control panel

[15] `python3 manage.py createsuperuser` `admin` `user` [15]

→ username :
→ email :
→ password :
→ re-enter password :
→ N/Y → Y

[16] `python3 manage.py runserver` `127.0.0.1/admin` `الصفحة` [16]
`admin` `للمشور` `الصفحة` `127.0.0.1/admin` `نه فدي`

علامة ملاحظة: لم يغير اد Post model
لأننا لم نقم بإضافة أي admin.py

17

17 داخل ملف admin.py :

```
from models import Post
```

```
admin.site.register(PostPost)
```

فكرة سنقوم بإتمام العملية القادمة
داخل views.py :

```
def post_list(request):
```

```
    post = Post.objects.all()
```

```
    return render(request, 'Post-list.html', {'posts': posts})
```

↓
تحويل صفحة

html

↓

إرسال اد Post، ببني

عن طريق المتغير posts