

وب سرور چیست؟(نقش Nginx چیست؟)

وب سرور میشه گفت یک نرم افزار یا سخت افزار است که درخواست های HTTP یا HTTPS را از کاربران اینترنت میگیرد و یک صفحه وب یا داده را برای آنها برمیگرداند. اما Nginx یک وب سرور سبک است که علاوه بر سرو کردن فایل های HTML CSS JS میتواند درخواست ها را به برنامه های یک اند منتقل کند و فشار را بین چند سرور تقسیم کند.

ارتباط بین کلاینت و سرور؟

مرورگر یک درخواست http/https را به سرور میفرستد و بعد سرور این درخواست را دریافت کرده و پردازش میکند و پاسخ مناسب آن را بر میگرداند. این ارتباط با استفاده از پروتکل tcp برقرار میشود چون ارتباط بین کلاینت و سرور مهم است و http فقط استاندارد بین درخواست و پاسخ است و مرورگر پاسخ مناسب را به کاربر نمایش میدهد.

وقتی url را باز میکنید چه میشود؟ http چیست؟

پروتکل http چیزی است که تعیین میکند کلاینت و سرور چگونه با هم تبادل داده کنند. وقتی url را وارد میکنید: اول مرورگر دامنه را از طریق dns به IP تبدیل میکند. بعد ارتباط TCP با سرور برقرار میشود. بعد مرورگر یک درخواست HTTP را میفرستد و سرور پاسخ HTTP را برمیگرداند. مرورگر براساس داده نتیجه را به کاربر نمایش میدهد.

جایگاه Django در معماری شبکه

میدانیم که Django یک **بک اند فریمورک** است، یعنی جایی که منطق برنامه، پردازش داده و ارتباط با پایگاه داده انجام میشود. در معماری:

- کاربر → مرورگر → وب سرور (مثل Nginx) → برنامه Django → پایگاه داده
Django پاسخ را (معمولاً HTML یا JSON) تولید میکند و به وب سرور می دهد تا به مرورگر برگرداند

توضیح فرایند کامل

مرورگر → DNS → شبکه → وب سرور → برنامه Django → پایگاه داده

مرورگر: کاربر آدرس را وارد میکند. **DNS:** دامنه به IP سرور تبدیل میشود. **شبکه:** ارتباط TCP/HTTP بین مرورگر و سرور برقرار میشود. **وب سرور:** درخواست را دریافت و در صورت نیاز به Django پاس میدهد. **برنامه Django:** منطق برنامه اجرا میشود، داده ها پردازش یا از پایگاه داده خوانده میشوند. **پایگاه داده:** داده ها ذخیره یا بازیابی میشوند. **پاسخ:** نتیجه به مرورگر فرستاده و به کاربر نمایش داده میشود

تفاوت MVC , MVT در Django

جنگو از الگوی mvt پیروی میکند و تفاوت mvc, mvt در این است که در جنگو نقش Controller را خود فریمورک بر عهده دارد و ما بیشتر روی View و Template تمرکز داریم.

نقش اجزای اصلی Django:

نقش url: مسیر درخواست را شناسایی کرده و اون رو به View مرتبط میکند.

نقش View: منطق پردازش پاسخ ها و درخواست ها.

نقش model: ساختار داده و تعامل با پایگاه داده.

نقش template: قالب html برای نمایش داده ها به کاربر.

محیط مجازی پایتون (virtualenv) چیست؟

یک فضای جداگانه برای نصب کتابخانه ها و اجرای پروژه ، بدون تداخل با سیستم اصلی است ، که این کار باعث میشه هر پروژه نسخه مخصوص خودش رو از کتابخانه داشته باشه و با پروژه های دیگر بدون تداخل اجرا شود، امکان تست نسخه های مختلف یک پکیج و تمیز و قابلیت مدیریت کردن پروژه.

تفاوت Django REST Framework و Django Templates

Django Templates: برای تولید صفحات HTML سمت سرور. مناسب وبسایت هایی که بیشتر رندر سمت سرور دارند. Django REST Framework: برای ساخت API و ارسال داده خام (مثل JSON یا XML) به کلاینت. مناسب اپلیکیشن های موبایل. و از لحاظ کرابرد:

اگر پروژه HTML کامل سمت سرور را می سازد → Templates.
اگر پروژه داده را به اپلیکیشن های دیگر یا فرانت اند جدا می فرستد → DRF.

ایجاد یک پروژه ساده Django

ایجاد محیط مجازی : `python -m venv venv`

نصب جنگو: `pip install django`

ایجاد پروژه جدید : `django-admin startproject myproject`

ورود به پوشه پروژه : `cd myproject`

اجرای سرور توسعه : `python manage.py runserver`

ایجاد یک اپلیکیشن داخل پروژه Django

دستور ایجاد اپلیکیشن: `python manage.py startapp myapp`

اتصال اپلیکیشن به پروژه اصلی:

وارد فایل [setting.py](#) در پوشه پروژه اصلی شوید. بعد در بخش **INSTALLED_APPS**، نام اپلیکیشن را اضافه کنید:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'myapp',  
]
```