

به نام خدا

## View چیست؟

یک تابع یا کلاس است که web request کاربر را دریافت کرده و web response مناسب را برای کاربر ارسال میکند. این بخش نقش controller را در معماری MVC جنگو بازی میکند. ویوها http response هایی از نوع HTML content, Redirects, Error pages و غیره برمیگردانند.

## function based view Vs class based view ؟

CBV ها ساختار یافته ترند و از رویکر OOP پیروی میکنند، قابلیت inheritance دارند و از این نظر reusability بیشتری دارند اما از سوی دیگر، FBV ها پیچیدگی بسیار کمتری دارند و به راحتی قابل پیاده سازی هستند.

## یک FBV چه آرگومان هایی را به عنوان ورودی میگیرد؟

یک function based view حداقل یک آرگومان ورودی از نوع httprequest میگیرد که نمونه درخواست کاربر است و معمولا Request نامگذاری میشود. این نمونه، تمام اطلاعات مانند method, data, user info را در خود دارد. آرگومان های دیگری که میتواند به عنوان ورودی بگیرد، positional arguments و keyword arguments هستند. اینها از الگوی URL که توسط کاربر POST میشوند، گرفته میشود.

## چطور یک ویو میتواند HTML content یا JSON response بازگرداند؟

زمانی که قرار است ویو یک HTML content برگرداند، معمولا یک template rendering صورت میگیرد و Web page مورد نظر ساخته شده و برای کاربر ارسال میشود اما وقتی قرار است یک JSON response تولید شود، دیتاها به فرمت JSON سریالایز میشوند و سپس ارسال میشوند. این نوع پاسخ معمولا در APIs یا AJAX کاربرد دارند.

## تفاوت `Http response` و `render()` ؟

`Http response` امکان کنترل مستقیم `response content` را به دولوپر میدهد و معمولاً برای داده‌های خام استفاده میشود و در صورتی که قرار است پاسخ به صورت یک `web page` ارسال شود و گاهی از داده‌های داینامیک استفاده شود از تابع `render()` استفاده میشود. این تابع میانبری است که پردازش و ترکیب کردن قالب با داده داینامیک را به صورت اتوماتیک برای ما انجام میدهد و `HTML` رندر گرفته شده را به عنوان یک `Http response` برمیگرداند.

## نقش `template` ها در جنگو؟

`template` در جنگو به عنوان لایه ارائه عمل میکند و بخش طراحی بصری یک `web page` را از بخش `logic` آن جدا میکند. نقش اصلی آن تولید صفحات داینامیک `HTML content` است که برای نمایش به کاربر استفاده میشود.

## چگونه یک تمپلیت بسازیم؟

برای ساخت تمپلیت به دو روش میتوانیم عمل کنیم، میتوانیم در فولدر `app` مورد نظر یک فولدر با نام `templates` بسازیم و در آن، قالب‌های مد نظر برای آن `app` را بسازیم، یا اینکه یک فولدر `templates` به صورت `global` بسازیم و تمامی تمپلیت‌ها را در آن جای دهیم، دقت کنید در هر دو روش باید آدرس فولدر `templates` را به فایل `setting.py` پروژه اضافه کنیم تا پروژه ما، این فولدر را بشناسد. برای استفاده از آن در `view`، کافیست از تابع `render()` استفاده کرده و آدرس تمپلیت مربوطه را به عنوان آرگومان ورودی به `render()` پاس بدهیم.

```
return render(request, 'index.html', context)
```

## مفهوم `template context` چیست؟

`Template context` در واقع یک دیکشنری است که اطلاعات مورد نیاز تکمیل یک `web page`، اعم از داده‌های داینامیک را از ویو یا مدل مربوطه گرفته و به صورت `key: value` به تمپلیت میفرستد تا در تمپلیت با استفاده از `Django Template Language` استخراج و به کار گرفته شود.

## Template tag and filter ؟

تمپلیت تگ ها، ساختار های کنترلی هستند که به ما این امکان را میدهند که یک سری پردازش های منطقی را به قالب خود اضافه کنیم . برای مثال تگ های کنترلی شرطی:

```
{% if %} {% else %} {% endif %}
```

فیلتر ها برای تغییر فرمت نمایش متغیر ها قبل از رندر شدن استفاده میشوند و این امکان را میدهند که مدل نمایش اطلاعات را کنترل و به دلخواه تنظیم کنیم.

## چطور یک قالب را گسترش دهیم به شکلی که از تکرار کد ها پرهیز کنیم؟

میتوانیم در قالب ها، از رویکرد inheritance استفاده کنیم و با ساخت یک قالب بیس و ارث بری کردن سایر قالب ها از این قالب، از نوشتن تکراری بخش های مشترک صفحات مانند header , footer جلوگیری کنیم.

## نقش urls.py در یک پروژه جنگو ؟

فایل urls.py وظیفه map کردن هر درخواست ورودی با view مرتبط با آن درخواست را دارد. این بخش به عنوان مسیر یاب مرکزی یک پروژه جنگو عمل میکند.

## تابع path() چیست، چرا آدرس ها را نامگذاری میکنیم؟

تابع path()، URL ورودی را به عنوان یک آدرس در url patterns کانفیگ میکند تا در مرحله مسیریابی از این آدرس ها در جهت mapping استفاده شود. گاهی اوقات برای اینکه رفرنس دادن به آدرس راحت تر انجام بگیرد ، به آن آدرس یک نام اختصاص میدهیم.

ارائه دهنده: پویا پورعبداله