به نام خدا

عنوان: طراحی بکند وبسایت با فریم ورک جنگو

فهرست

* مقدمه
* جنگو
* معماری جنگو
* URLS
* VIEWS
* MODELS

مقدمه:

در این پروژه می‌خواهیم با استفاده از فریم ورک جنگو بکند وبسایت آموزشی در خصوص کتاب و تعدادی از کتابخانه های آن مربوط به اینترنت اشیا و... بسازیم.

در این گزارش ابتدا به توضیح مختضری از فریم ورک جنگو، معماری آن و توضیح بخش آن می‌پردازیم.

فریم ورک جنگو بر خلاف مدت کمی که از زمان تولیدش می‌گذرد به فریم ورکی محبوب تبدیل شده است به طوری که تعداد ستاره های آن در گیت هاب به حدود 56 هزار ستاره رسیده است که نشان دهنده محبوبیت این فریم ورک در بین فریم ورک ها میباشد.

بعلاوه اینکه سایت ها و شرکتی های بزرگی همچون اینستاگرام، موزیلا، pinterest و... از این فریم ورک استفاده می‌کنند که قدرت و اهمیت این آن را بالا می‌برد.

آخرین نسخه این فریم ورک نسخه 3.1 می‌باشد که در این فریم از آن استفاده شده است، آخرین نسخه LTS[[1]](#footnote-1) نسخه 2.2 از جنگو 2 می‌باشد که طبق گفته سایت رسمی سایت djangoproject.com تا حدودآوریل سال 2022 پشتبانی خواهدشد. نسخه LTS از نسخه 3 نسخه 3.2 خواهد بود که تا سال 2024 پشتیبانی خواهد شد.

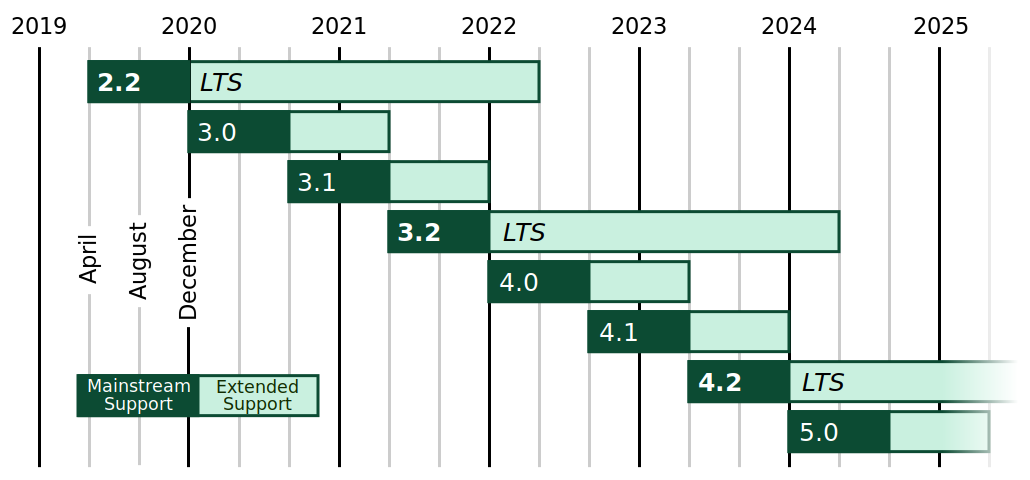


Figure 1 نسخه های جنگو

جنگو:

جنگو فریم ورک open-source تحت وب پایتون است که اولین انتشار رسمی آن در سال 2005 بوده است که توسط فردی به نام Adrian Holovatyساخته شده است. از مزیت های قابل توجه این فریم ورک می‌توان به توسعه سریع، مقدار کد کمنر و اصل (Don’t repeat yourself) توجه کرد. علاوه بر اینها جنگو خود به خود از حملاتی مثل SQLInjection، [password cracking](https://en.wikipedia.org/wiki/Password_cracking" \o "Password cracking) و ... جلوگیری می‌کند.

جنگو مستقیماً از دیتابیس استفاده نمی‌کند و با استفاده از ORM خود این کار را انجام می‌دهد که از مزیت های این کار می‌توان به سرعت بالای در کار با دیتابیس و جلوگیری از حملاتی مثل SQL Injection اشاره کرد.

جنگو امکان کار با API ها را نیز دارد که برای این کار می‌بایست از کتابخانه Django-rest-frame نیز استفاده کرد.

معماری جنگو:

معماری جنگو MVT استفاده می‌کند که در واقع مخفف model-view-template می‌باشد که به معنی این است که مدل ها(دیتابیس ها) و تمپلیت ها که فایل های html های در بخش view ها با هم ترکیب می‌شوند و به کاربر نشان داده می‌شوند.

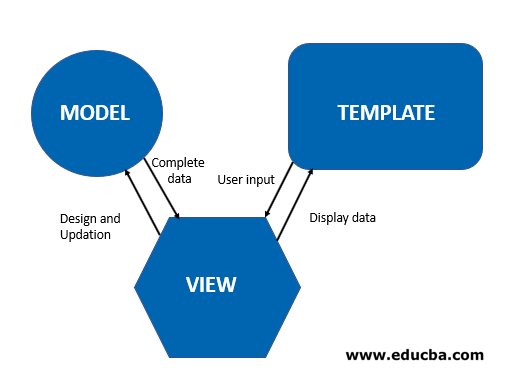


Figure 2 MVT Architecture

و یا به یا به عبارتی دیگر معماری MVC استفاده می‌کند که مخفف عبارت model-view-controller است که در واقع تعریفی کامل تر از معماری MVT است

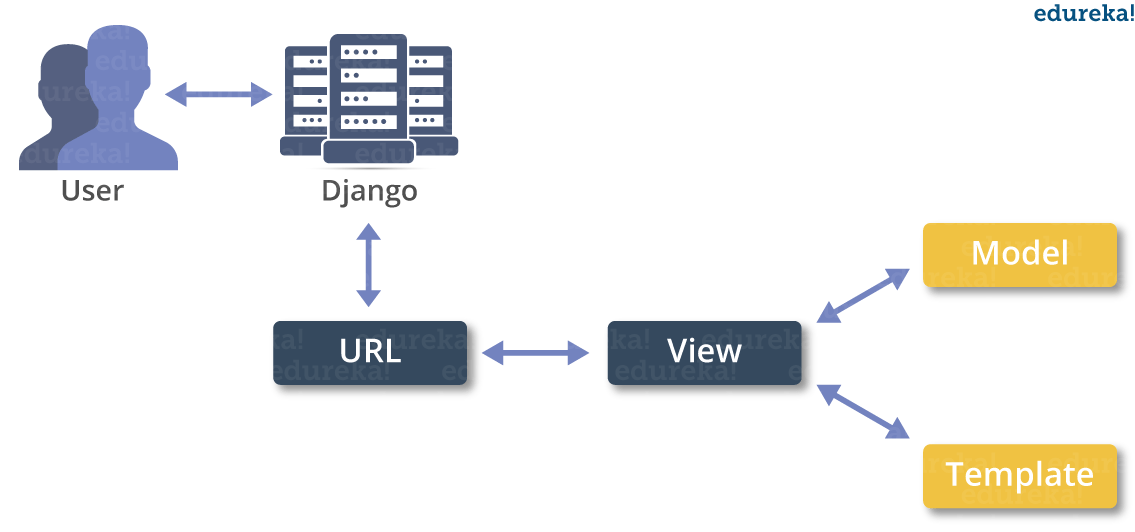


Figure 3 MVC Architecture

همانطور که در تصویر شماره 3 وقتی کاربر با استفاده از جنگو یک آدرس را وارد می‌کتد و جنگو ابتدا این url را به viewsها ارسال می‌کند و بعد از آن را با modelها (که در واقع دیتای مربوط به هر request است) و templateها ترکیب میکند و به کاربر بر می‌گرداند

:URLs

آدرس هایی هستند که به ازای هر کدام از آنها یک view وجود دارد که به ازای وارد کردن هر کدام از آنها یک request ساخته می‌شود و به سمت view می‌رود و بعد از تولید response به کاربر نمایش داده می‌شود

به عنوان مثال به ازای وارد کردن آدرس سایت برای این پروژه در واقع prefix روت به ویو landing که صفحه اصلی سایت است که از آنجا می‌توان به ویوو های دیگر که پست ها و پست های ویدویی هستند رفت که این کار هم با urlها انجام می شود و روند توضیح داده شده برای صفحه landing برای صفحات دیگر هم اتفاق می‌افتد.

:VIEWs

همانطور که پیش تر نیز توضیح داده شد، وظیفه این بخش ترکیب اطلاعاتی که با استفاده از ORM از دیتابیس دریافت کرده ایم و تمپیلت هاست ک بخشی از این کار نیز با template engine ها انجام می‌شود که از جمله آنها می‌توان به jinja و jinja2 استفاده کرد که جنگو به صورت default از jinja استفاده می‌کند که از لحاظ شیوه نوشتار(syntax) تفاوت هایی با jinja 2 دارد.

در ویو ها یک دسته ای از ویو ها هستند به صورت آماده به منظور کار های خاصی استفاده میشوند که از جمله آنها می‌توان به (list-view) ها اشاره کرد که برای نمایش نوعی از ویو که لیستی از دیتا را می‌خواهیم نمایش دهیم استفاده می‌شود.

:MODELs

همانطور که پیش تر اشاره شد مدل ها دیتا بیس های هر اپ جنگو می‌باشند که به صورت کلاس های پایتونی تعریف می‌شوند که هر کدام ار آنها یک table برای دیتابیس های sql base یا collection برای دیتابیس های no sql را تشکیل می‌دهند.

برای ذخیره اظلاعات در دیتابیس از فیلدهایی استفاده می‌شود که تعدادی از آنها عبارتنداز:

:CharField برای ذخیره دیتا های متنی مثل اسم، فامیلی و... استفاده میشود.

IntegerField: برای ذخیره اعداد صحیح استفاده می‌شود و نباید برای شماره تلفن از آن استفاده کرد.

EmailField: دقیقآً مثل Charfield اما با این تفاوت که ایمیل بودن ورودی را چک می‌کند.

URLField: دقیقا مثل Charfield اما با این تفاوت که یک آدرس اینترنتی را در خود ذخیره می‌کند

FileFIeld: فایل را به آدرس تعیین شده در داخل پرانتز مربوط به هر فیلد تعریف شده است آپلود می‌کند و آدرس آن در دیتابیس ذخیره می‌شود و از آن آدرس قابل دسترسی خواهد بود.

ImageField: دقیقا مثل FileField عمل می‌کند اما با این تفاوت که برای استفاده از این مورد نیاز به کتابخانه pillow داریم.

همچنین می‌توانید بین دیتا بیس ها و table های مختلف ارتباط برقرار کرد که برای می‌بایست از فیلد های زیر بهره ببریم:

foreignKey: یک آبجکت می‌تواند با چند آبجکت های از table های مختلف ارتباط داشته که بک مثال برای نوضیح این ارتباط، ارتباط یک پدر با چند فرزند باشد.

OnetoOneField: بک آبکجت می‌تواند با یک آبجکت از یک table دیگر ارتباط داشته باشد. مثل ارتباط یک زن و یک شوهر

ManytoManyFiled: چند آبجکت از یک دیتا بیس می‌توانند با چند آبجکت از table های مختلف ارتباط داشته باشد. مثل ارتباط چند برادر خواهر با هم

Template Engines :

Template engine ها به منظور مدیریت و داینامیک سازی تمپلیت ها که فایل های html به کار می‌روند. به عنوان مثال هنگامی که با رفتن به صفحه پست ها آخرین پست ها را می‌بنیم، نشان از داینامیک بودن صفحه است. برای توضیح اینکه داینامیک سازی دقیقا چگونه کار میکند ابتدا باید نگاه دیگری به مبحث view ها بیندازیم.

همانطور که قبلا توضیح داده شد، هر view هنگامی که فراخوانی می‌شود ابتدا تمپلیت و دیتای مربوط به خود(که توسط مدل ها انجام می‌شود) را توسط الگویی که برنامه‌نویس به آنها داده است با هم ترکیب می‌کند و بعد به صورت response به کاربر بازمی‌گرداند.

برای مثال در صفحه پست های این پروژه، ابتدا هنگامی که بر روی دسته بندی مربوطه کلیک می‌کنیم، دیتای مربوط به هر دسته بندی فراخوانی می‌شود و سپس به نمپیلیت ارسال و با حلقه for مربوط به جینجا، دیتای مورد نظر رندر شده و به در نهایت به کاربر نشان داده میشود.

به طور کلی ما template engine داریم که جینجا و جینجا2 هستند که تفاوت syntax کمی دارند، و جنگو به صورت default از جینجا استفاده می‌کند که میتوان با مراجعه به فایل settings پروژه آن را تغییر داد.

تعدادی از دستورات در جینجا عبارتند:

حلفه for

ساختار شرطی if

تگ load static

نگ استاتیک

و...

که به کاربرد for و if قبلا اشاره شده اما برای توضیح static tags می‌بایست به کمک به استفاده راحت تر از فایل های استاتیک پروژه اشاره کرد. و تگ load static به منظور فعال کردن و صدور دسترسی برای استفاده از نگ های استاتیک به کار می‌رود.

1. LTS(LONG\_TERM SUPPORT):به نسخه ای گفته می‌شود که برای مدت طولانی تری نسبت به بقیه ورژن های این نسخه پشتیبانی می‌شود [↑](#footnote-ref-1)