

10 نقطه دام جنگو برای تازه کران

جنگو یک فریم ورک وب قوی است، اما استفاده از آن برای کاربران تازه کار ممکن است کمی طاقت فرسا باشد. من این مشکل را چند سال پیش بهش بر خوردم، و حتی برای درک مفهوم های ساده چندین ساعت صرف کردم. اگر در جنگو تازه کار هستید، (یا به طور کلی با فریم ورک های MVC)، شاید باید طرز فکرتان را کمی برای استفاده از این فریم ورک تغییر دهید. در زیر چند تا از مشکلاتی که برای من پیش آمد و درک آنها چندین ساعت وقت بردار می بینید. خوشبختانه، این لیست چندین ساعت را برای شما ذخیره می کند.

1. جنگو یک CMS نیست

جنگو CMS نیست. جنگو نمیتواند کلرهای را که wordpress و drupal به راحتی انجام می دهند، انجام دهد. با این وجود، می توانید کلرهای بیشتری با جنگو انجام دهید چون یک CMS نیست.

اگر زمینه کار با، برای مثال، wordpress دارید، زمانی زیادی طول می کشد تا به ساخت دستی url و تغییر تنظیمات عادت کنید. بیشتر تنظیمات بدون صفحه گرافیکی انجام می شوند، بلکه آنها با فایل های پایتون تنظیم می شوند.

گرچه جنگو با یک محیط مدیریتی میاد، تنظیم جنگو به راحتی یک CMS استاندارد نیست.

2. محیط نصب محلی

یکی از بزرگترین مشکلاتی که کاربران برای تنظیم جنگو روی محیط محلی دارند. برای بیشتر تازه کران، نصب محلی فریم ورک بیشتر از ساخت اولین برنامه یشان وقت می برد. فریم ورک های PHP مثل CakePHP تقریباً بعد از دانلود برای استفاده در سیستم محلی آماده اند.

نصب جنگو و تنظیم آن برای اجرا روی سیستم محلی کمی سخت است، مخصوصاً اگر با مفهوم هایی مثل symlinks آشنا نباشید. مستندات نصب جنگو تقریباً کامل است، اما اگر آنها برای شما کارساز نبودند میتوانید چند ویدیو آموزشی را جست و جو کنید یا به دنبال مستندات مخصوص پلتفرم خود بگردید.

3. پروژه ها در برابر برنامه ها

اگر در جنگو تازه کار هستید (یا فریم ورک های MVC به صورت کلی)، ممکنه کلمه های "پروژه ها" و "برنامه ها" کمی گیج کننده باشد.

در جنگو، یک پروژه فقط مجموعه ای از برنامه هاست. به عنوان مثال داریم یک سایتی شبیه به YouTube میسازیم؛ یک پروژه می سازید، "MyTube"، که حاوی برنامه هایی مثل "ویدیو ها"، "کامنت ها"، "رتبه ها"، "بلاگ ها" و غیره.

برنامه ها شبیه ماژول های یک CMS هستند، فقط آنها را از صفر می سازید. آنها خیلی کوچک تر هستند، و معمولاً فقط یک عمل اصلی انجام می دهند. برای مثال "MyTube"، شما فقط یک برنامه برای آپلود ویدیو توسط کاربران، نوشتن کامنت برای آنها، رتبه دهی آنها، و هر چیز دیگر نمیسازید. شما میخواهید برنامه هایتان کلرای کوچک داشته باشند، و فقط یک یا دو عمل را انجام دهند.

4. وراثت قالب

وراثت قالب یکی از بهترین قابلیت های جنگوست، اما فهم آن کمی زمان بر خواهد بود.

با یک CMS سنتی، شاید قسمت های قابل استفاده دوباره را از قالب جدا کنید و به فایل های خودشان برآرید (مثل یک فایل سرتیتر)، آنها را در هر یک از قالب ها شامل کنید. با جنگو، شما یک قالب پایه Base.html می سازید، و شما بلوک های دلای قابلیت رونویسی در قالب کودک معین می کنید. این کار کدرا بیشتر دلای قابلیت استفاده دوباره می کند.

پس، برای مثال، شاید یک قالب پایه Base.html شبیه زیر داشته باشید:

```
<div>
  {% block content %}
    <p>Here is our content that will be overwritten</p>
  {% endblock %}
  <p>Other stuff that won't be overwritten.</p>
</div>
```

بعد در قالبی دیگر شاید بخواهید بلوک `{%content%}` را رونویسی کنید. به جای شامل کردن یک فایل فرعی به طرح، به راحتی کاری شبیه زیر می کنیم:

```
{% extends base.html %}

{%block content%}
I'd like a different content block!
{%endblock%}
```

این Base.html را طولانی تر می کند. این روش باعث کنترل بیشتر بر روی قالب ها می شود، و اطمینان این را حاصل می کند که کدهایتان کپی نشده باشند. برای یادگیری بیشتر درباره قدرت وراثت قالب، سری به این مقاله درباره [قدرت وراثت قالب جنگو بزنید](#).

5. جدایی کد و فایل های رسانه ای

به دلیل اینکه خود جنگو فایل های رسانه ای به کار نمی برد، دوست دارد که فایل های رسانه ای (عکس ها، اسکریپت های جاوا، و غیره) را در یک مسیر دیگر جدا از برنامه ها قرار دهد. این به شما اجازه می دهد که چگونه از آن فایل ها بهره ببرید. می توانید یک سرور وب جدا یا سرویس ابری مثل [Amazon S3](#) را برای فایل های خود به کارگیرید.

فایل های پروژه شما شاید چیزی شبیه به این باشد:

```
MyProject
  -app
  -app2
  -media
    -javascript
    -stylesheets
    -images
  .....

```

6. مهاجرت ها

مهاجرت های پایگاه داده یکی از قسمت های مشکل برای تازه کاران است. این وقتی که شما از قبل در پایگاه داده خود فیلد (field) دارید اطلاق می افتد (توسعه یا تولید)، و شما چیزی اضافه می کنید که پایگاه داده را تغییر می دهد.

دیگر فریم ورک ها مثل [Rails](#) مهاجرت های پایگاه داده را در خود دارند. در جنگو، اگرچه، مگر اینکه از یک راه حل third-party استفاده کنید، باید مهاجرت ها را به صورت دستی (غیر خودکار) انجام دهید. خوشبختانه، یک ابزار third-party خوب وجود دارد که کارهای مهاجرت را انجام می دهد: [South](#)

South به شما اجازه انجام مهاجرت پایگاه داده را می دهد، تقریباً مثل اون شکلی که Rails به شما این اجازه را می دهد، و وارد کردن آن به پروژه های جاری خودتان آسان است. South قوی و ساده است، و مشکلات مهاجرت پایگاه داده ای که جنگو دارد را حل می کند. در اینجا یک [آموزش](#) قرار دارد که درای مثال هایی از چگونگی کاربرد South است.

7. محیط محلی در برابر محیط های توسعه

جنگو با SQLite می آید، یک پایگاه داده ساده که نیازی به تنظیمات ندارد. این آزمایش نمونه های اولیه را ساده و سریع تر می کند.

اگرچه، زمانی که پروژه خود را به محیط توسعه بردید، شاید مجبور شوید از پایگاه های داده قوی تر مثل PostgreSQL یا MySQL استفاده کنید. این به این معناست که شما درای دو محیط جداگانه هستید: توسعه و تولید.

جنگو منطقی برای تعویض راحت محیط ها بدون تغییر فایل settings.py ندارد. برای مثال، داشتن تنظیمات

```
DEBUG = True
```

برای محیط های محلی مفید است، به دلیل اینکه وقتی کد شما مشکلی داشته باشد پیام های کاربردی نشان داده می شوند. اما وقتی در محیط تولید هستید، هیچ وقت نمیخواهید که کاربران پیام های خطا را ببینند، پس باید این را معین کنید:

```
DEBUG = False
```

یک راه برای اتوماتیک کردن تعویض بین این دو محیط هست، با استفاده از [این تغییر کوچک](#):

```
import socket
if socket.gethostname() == '<a href="http://productionserver.com"
target="_blank">productionserver.com</a>':
    DEBUG = False
else:
    DEBUG = True
```

شما می توانید از بیت سوکت در قسمت های مختلف settings.py استفاده نمایید.

8. نوشتن تگ های قالب سفارشی

تگ های قالب یکی از پر کاربردترین خصوصیت در جنگو است و در حقیقت، جنگو از قبل با تعداد زیادی [تگ های قالب آماده](#) می آید.

اگرچه، اگر می خواهید [تگ های قالب سفارشی خودتان](#) را بنویسید، این فرایند می تواند کمی گیج کننده باشد. در اینجا یک مثال از چگونگی [نوشتن تگ های قالب](#) هست که باید به شما در فهم بیشتر و چگونگی نوشتن تگ های قالب خودتان کمک کند.

9. سندیت کاربر جنگو

جنگو به همراه یک سیستم سندیت کاربر در پروژه مدیر می آید که می توانید از آن برای مدیریت کاربران استفاده کنید. اگرچه، سیستم سندیت کاربر روش ساده ای برای دسترسی کاربران غیر مدیر برای ساخت حساب های کاربری نمی دهد. CMS هایی، مثل دروپال، این کار را به خوبی انجام می دهند.

چند گزینه خوب برای سیستم سندیت کاربر عمومی هست که این کمبود را جبران می کند. بهترین راه third-party احتمالاً

راه حل [django-registration](#) است. ساده و انعطاف پذیر، و تمام چیزی را که برای ثبت نام کاربر جدید نیاز دارید برای شما فراهم می آورد.

10. دیدگاه های عمومی

دیدگاه های عمومی خیلی قوی هستند، و می توانند زمان نوشتن دیدگاه برای برنامه جنگو شمارا کاهش دهند. اگرچه، خیلی از تازه کران نمی دانند چقدر این ابزار کاربردی است.

برای مثال، اگر بخواهید یک صفحه تخت برای صفحه خانه خود درست کنید، اما می خواهید حالت آن متفاوت باشد، می توانید یک قالب رومسیر قالب flatpages بسازید که از دیدگاه `direct_to_template` استفاده می کند.

به غیر از ساخت قالب مناسب، باید فایل `urls.py` را تغییر دهید تا به آن اشاره کند:

```
urlpatterns = patterns(' ',
    (r'^$', 'django.views.generic.simple.<WBR>direct_to_template', {'template':
'homepage.html'}),
    .....
```

حال، ما یک صفحه خانگی با طراحی سفارشی داریم که نیازی به نوشتن هیچ دیدگاه سفارشی ندارد. من این تکنیک را بلرها به جای [نوشتن دیدگاه های اضافه](#) استفاده کردم.

جمع بندی

وقتی در تمام قسمت هایی که برای تازه کران سخت است کار کردید، جنگو اثبات می کند که ابزاری قوی و سریع برای نوشتن برنامه های وب است. اگر در اوایل راه گیج شدید اصلا نگران نباشید. همیشه می توانید در کانال IRC جنگو کمک بخواهید؛ [Stackoverflow](#) مکان مناسب دیگری برای پرسیدن سوال است.

مترجم : علی مرادی

تماس با من : adeadmarshal@gmail.com

سایت : <http://tarjomeebook.ir>