



Modern Django

with Dynamic Frontend & awesome features



Cookiecutter

کوکی کاتر یک ابزار متن باز است که برای ایجاد پروژه های جدید بر اساس الگوهای از پیش تعیین شده به کار می رود. این ابزار به توسعه دهنده گان اجازه می دهد تا به سرعت ساختار و تنظیمات اولیه پروژه های خود را بدون نیاز به تکرار کارهای دستی ایجاد کنند.

تاریخچه :Cookiecutter-Django

-  **شروع پروژه** - در سال ۲۰۱۳ توسط Daniel Roy Greenfeld به عنوان یک ابزار متن باز برای ایجاد سریع پروژه های مبتنی بر الگو توسعه یافت.
-  **محبوبیت در جامعه توسعه دهنده گان** - به سرعت در میان توسعه دهنده گان Python، به ویژه برای فریمورک های مانند Flask و Django محبوب شد.
-  **گسترش به زبان ها و فریمورک های مختلف** - پس از موفقیت اولیه، Cookiecutter برای پشتیبانی از پروژه های متنوع در زبان ها و فریمورک های مختلف توسعه یافت.
-  **ابزار استاندارد در توسعه پروژه های مدرن** - به مرور زمان، این ابزار به یکی از ابزارهای اصلی در استانداردسازی پروژه ها، به ویژه در محیط های تیمی و صنعتی، تبدیل شد.

چرا؟

- صرفه‌جویی در زمان: بدون Cookiecutter، باید برای ساختار، ابزارها، و تنظیمات پروژه تصمیم‌گیری کنید که زمان بر است.
- پاییندی به Best practices: کوکی کاتر الگوهایی با بهترین شیوه‌های توسعه ارائه می‌دهد و از مشکلات احتمالی ناشی از تنظیمات دستی جلوگیری می‌کند.

محدودیت‌ها

- محدودیت در شرایط خاص: Cookiecutter ممکن است برای پروژه‌های بسیار سفارشی یا غیر معمول مناسب نباشد، زیرا الگوها بر اساس یک tech stack مشخص طراحی شده‌اند.

استفاده‌های رایج

- ایجاد پروژه‌های جدید: شروع سریع پروژه‌های جنگو با تنظیمات آماده و بهینه.
- توسعه در تیم‌ها: هماهنگ‌سازی اعضای تیم با ساختارهای یکپارچه و استاندارد.
- استفاده در محیط production: با تنظیمات Docker و CI/CD، پروژه‌ها به سادگی به محیط‌های تولید منتقل می‌شوند.

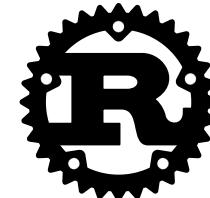
ابزارهای استفاده شده در Cookiecutter-Django

Ruff

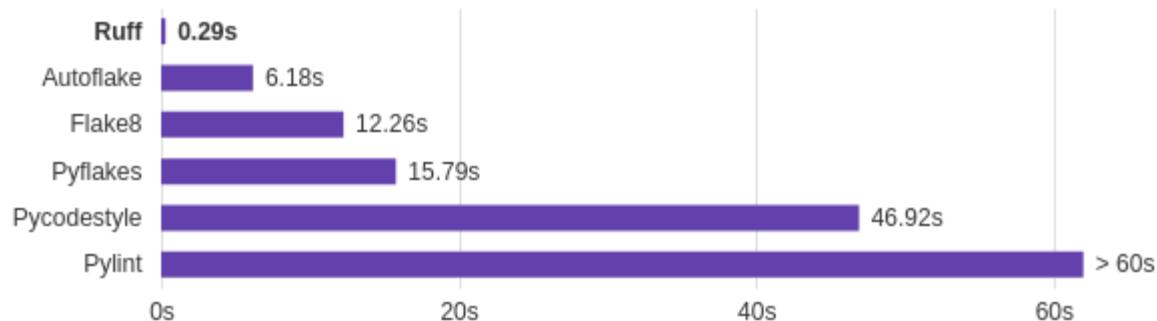
یک ابزار linting بسیار سریع برای Python است که به بررسی و تصحیح کدهای شما کمک می‌کند. این ابزار می‌تواند انواع مشکلات کدنویسی و سبک کد را تشخیص دهد و پیشنهادهایی برای بهبود آن‌ها ارائه کند. با استفاده از Ruff، می‌توانید از کدهای تمیز و باکیفیت اطمینان حاصل کنید.



- نوشته شده با زبان راست



- فوق سریع



DjLint

یک ابزار DjLint مخصوص linting Django است که به بررسی کدهای تمپلیت Django و HTML می‌پردازد. این ابزار به طور خاص برای توسعه‌دهندگان Django طراحی شده است تا اطمینان حاصل شود که تمپلیت‌ها و ساختار HTML پروژه با استانداردهای کدنویسی مطابقت دارند.

- جلوگیری از اشکالات تمپلیت
- خوانا شدن تمپلیت
- best practice
- رعایت

Pre-commit

به شما اجازه می‌دهد تا قبل از هر commit، اسکریپت‌های خاصی را اجرا کنید. این اسکریپت‌ها می‌توانند برای بررسی کیفیت کد، اجرای تست‌ها، یا هر اقدام دیگری استفاده شوند. با Pre-commit، می‌توانید از ورود کدهای مشکل‌دار به مخزن کد جلوگیری کنید.



- اجرای اسکریپت‌های خاص
- اجرای تست‌ها
- اجرای فرمت‌بندی کد
- اطمینان از کیفیت کد برای تمام تیم

```
$ cd my_project
# install in venv
$ python3 -m pip install pre-commit
# make sure git is already initialized
$ pre-commit install
```

Docker

Docker ابزاری است که به شما اجازه می‌دهد پروژه‌های خود را در کانتینرها جدآگانه توسعه دهید و اجرا کنید. این کانتینرها تضمین می‌کنند که محیط توسعه شما مستقل از سیستم عامل و سایر پروژه‌ها باشد. با Docker، می‌توانید پروژه‌های خود را به راحتی بین توسعه‌دهندگان مختلف به اشتراک بگذارید و به محیط production منتقال دهید.



- ابزار و استاندارد جهانی

ابزارهای پیشنهادی ما

Justfile

به شما امکان می‌دهد که دستورات متداول و پیچیده را در قالب ساده ذخیره کنید. با استفاده از Justfile، می‌توانید وظایف متداول مانند ساخت پروژه، اجرای تست‌ها، یا شروع سرور را تنها با یک دستور اجرا کنید. این ابزار بهره‌وری شما را افزایش می‌دهد و کارها را ساده‌تر می‌کند. ✨

```
# Build the Docker images
build:
    docker compose -f docker-compose.local.yml build

migrations:
    docker compose -f docker-compose.local.yml run --rm django python manage.py makemigrations

# Run the Docker containers
run:
    docker compose -f docker-compose.local.yml up
```

htmx (Dynamic Frontend)

-یک کتابخانه (Library) کوچک جاوا اسکریپت است.
-به ما این اجازه را میدهد که وب اپ های داینامیک بسازیم (Dynamic web application).
-هسته‌ی HMTX ایده‌ای بود برای ساخت وب اپ ها با استفاده از Hypermedia driven Approach.
-رویکرد هایپرمدیا در HMTX بر این ایده استوار است که خود HTML و لینک های آن به جای استفاده از JS میتوانند برای ارتباطات، تعاملات و داینامیک بودن سایت استفاده شوند.
HTML به خودی خود یک هایپرمدیاست یا حداقل زبان است (البته که زبان محسوب نمیشود صرفا برای هایپرمدیا اینو گفتیم) که شامل مجموعه‌ای از کنترل های هایپرمدیاست از جمله لینک ها و فرم ها. به طور مثال یک لینک به صورت زیر از طریق href آدرس دهی میشه:

```
# link in html  
<a href="/some/path">click me</a>
```

و این کنترل ها به نوبه‌ی خود اجازه‌ی تعامل بین مرورگر و سرور را میدهند.
-به این دلیل که HTML شامل تعداد زیادی از هایپرمدیا کنترل ها نیست که ما بتونیم با سرور تعامل داشته باشیم، ولی زمانی که از HMTX استفاده میکنیم این مدل رو گسترش میدهد به طوری که تقریبا هر عنصر HTML میتوانه به یک هایپرمدیا کنترل عالی تبدیل بشه که میتوانه با سرور تعامل داشته باشه و HMTX این کار رو با ارایه‌ی ویژگیهای Attribute اضافی انجام میدهد که میتوانیم روی عناصر HTML اعمال کنیم تا کارهایی مثل راه اندازی Request HTTP رو انجام بدیم.

ادامه(نمونه ای از کد HTMX)

به طور مثال با استفاده از HTMX، به همین سادگی یک قسمت رو داینامیک میکنیم:

```
<!-- بارگذاری پاسخ در div با id="content" -->
<button hx-get="/load-content" hx-target="#content">
  Load Content
</button>

<div id="content">
  <!-- محتوا اینجا بارگذاری میشود -->
</div>
```

مزایا و معایب نسبت به Vue.js

سبکی و کارایی: HMTX برخلاف Vue.js نیازی به یک پکیج بزرگ جاوا اسکریپت ندارد و مستقیماً به عناصر HTML متصل می‌شود. برای سایتهايی که پیچیدگی زیادی ندارند، این باعث می‌شود سایت سریع‌تر بارگذاری شود و تجربه کاربری بهتری داشته باشد.

تعاملات سمت سرور: HMTX به سادگی با backend مثل Django سازگار است و به شما امکان می‌دهد بدون مدیریت یک وضعیت پیچیده در فرانت‌اند، با سمت سرور تعامل برقرار کنید. در مقابل، Vue.js برای تعاملات پیچیده‌تر سمت کاربر و حالت‌های زیادی که باید مدیریت شوند، گزینه بهتری است.

قابلیت گسترش و مدیریت وضعیت: اگر پروژه به تعاملات پیچیده یا کامپوننت‌های بسیار پیچیده نیاز داشته باشد، Vue.js با سیستم مدیریت وضعیت خود (Vuex) و ابزارهایی مثل روت برای صفحات پیچیده، بهتر عمل می‌کند.

معایب و محدودیت‌ها: HMTX برای پروژه‌های پیچیده مناسب نیست، زیرا به نسبت Vue.js انعطاف‌پذیری کمتری در ساخت اپلیکیشن‌های پیچیده دارد.

یادگیری و راهاندازی ساده‌تر: HMTX به دلیل وابستگی کم به کدنویسی جاوا اسکریپت و تمرکز بر HTML یادگیری ساده‌تری نسبت به Vue.js دارد.

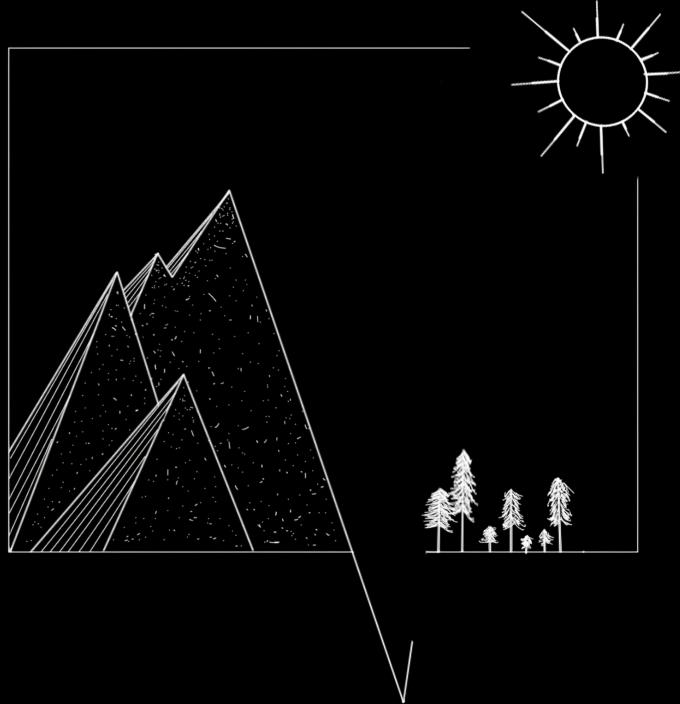
نصب و استفاده

```
$ pip install "cookiecutter>=1.7.0"
```

```
$ cookiecutter https://github.com/cookiecutter/cookie
```

پس از این مرحله درباره پروژه سوالاتی پرسیده میشود که میتوانید پاسخ دهید و یا از مقادیر پیشفرض استفاده کنید.

گیتهاب پروژه cookiecutter-django



ساخت یک پروژه برای تست



1. تعریف نیازمندی های پروژه
2. طراحی نسبی سیستم
3. ایجاد پروژه با Cookiecutter

با تشکر از توجه شما

Powered by  Sliderv