

پروژه تک نیوز

مقدمه:

پروژه‌ی «تک نیوز» یک سامانه‌ی جمع‌آوری، ذخیره‌سازی و نمایش اخبار از سایت زومیت است. در این پروژه با استفاده از ابزارهایی مانند Django، Celery، Scrapy و Docker، یک سیستم ساختیم که اخبار را از سایت Zoomit جمع‌آوری کرده، آن‌ها را پردازش کرده و از طریق یک API در اختیار کاربران قرار می‌دهد.

ابزارهای استفاده‌شده:

ابزار	کاربرد
Python 3.12	زبان برنامه‌نویسی اصلی
Django	فریم‌ورک اصلی برای ساخت API و مدیریت داده
Scrapy	فریم‌ورک کراول و استخراج اخبار از وب‌سایت
Playwright	اجرای کراول مبتنی بر مرورگر برای صفحات دینامیک
Celery	اجرای تسک‌ها، مانند کراول کردن و ذخیره اخبار بصورت خودکار
Celery Beat	زمان‌بندی اجرای خودکار تسک‌های Celery
Redis و	Celery Message broker برای Celery
Flower	مانیتورینگ تسک‌های Celery
Docker	کانتینری‌سازی کل پروژه برای توسعه و اجرا در محیط ایزوله
Docker Compose	مدیریت چند سرویس Docker با یک فایل مشترک

مدل‌های ساخته‌شده:

1. مدل خبر که شامل title, content, tags, reference می‌شود.
2. مدل tag که شامل نام این دسته بندی می‌شود (مانند ورزشی، اقتصاد، ...)

فرایند جمع‌آوری خبر:

1. اجرای دستور `python manage.py runserver` ابتدا پروژه را راه اندازی میکنیم.
2. اجرای دستور `celery -A roshanNewsProject worker -l info` ورکر ها شروع به کار میکنند.

3. سپس در ترمینال دیگر با اجرای دستور `celery -A roshanNewsProject beat -l info` سلری بیت هم شروع به کار کرده و به صورت دوره ای در بازه های زمانی مشخص اخبار را از وبسایت زومیت کرا ل کرده و در دیتابیس ذخیره میکند.

4. در سمت سرور، این درخواست در یک تسک Celery مدیریت شده و خبر در دیتابیس ذخیره می شود.

5. تمامی اخبار از طریق `api` زیر در دسترس هستند:

["http://127.0.0.1:8000/api/news/"](http://127.0.0.1:8000/api/news/)

همچنین امکان فیلتر کردن اخبار بر اساس 3 شاخص `tag`, `keyword`, `exclude_keyword` وجود دارد

["http://127.0.0.1:8000/api/news/?keyword=\[عبارت مورد نظر\]"](http://127.0.0.1:8000/api/news/?keyword=[عبارت مورد نظر])

["http://127.0.0.1:8000/api/news/?tags=\[عبارت مورد نظر\]"](http://127.0.0.1:8000/api/news/?tags=[عبارت مورد نظر])

["http://127.0.0.1:8000/api/news/?exclude_keyword=\[عبارت مورد نظر\]"](http://127.0.0.1:8000/api/news/?exclude_keyword=[عبارت مورد نظر])

و در نهایت امکان ترکیب کردن فیلتر ها نیز وجود دارد

[http://127.0.0.1:8000/api/news/?exclude_keyword=\[عبارت مورد نظر\]&keyword=\[عبارت مورد نظر\]](http://127.0.0.1:8000/api/news/?exclude_keyword=[عبارت مورد نظر]&keyword=[عبارت مورد نظر])

حتما توجه کنید که برای استفاده از این `api` و همچنین فیلتر های این `api` باید از متود `GET` استفاده کنید

6. برای ذخیره سازی اخبار جدید گفته شد از `api` زیر استفاده میکنیم

["http://127.0.0.1:8000/api/news/insert/"](http://127.0.0.1:8000/api/news/insert/)

حتما توجه کنید که برای استفاده از این `api` از متود `POST` استفاده کنید

همچنین داده ارسال شده باید دیتا به فرمت زیر ارسال شود:

```
Data = {  
    "title": [عنوان خبر],  
    "content_news": [متن خبر],  
    "tags": [موضوع خبر],  
    "reference": [منبع خبر]  
}
```

برای مثال:

```
{  
  
  "title": "اخبار نقل و اتصالات",  
  
  "content_news": "وبرتر به لیورپول پیوست",  
  
  "tags": ["فوتبال", "ورزشی", "نقل و انتقال"],  
  
  "reference": "www.varzesh3.com"  
}
```

تسک‌های Celery:

فایل tasks.py شامل تسک مربوط به کرال و ذخیره‌سازی اخبار است. این تسک‌ها با اجرا کردن دستور scrapy crawl zoomit در بازه زمانی ثابت توسط Beat اجرا می‌شوند تا فشار پردازش از مسیر اصلی Django برداشته شود.

زمان‌بندی (Celery Beat):

با Celery Beat، می‌توانیم جمع‌آوری اخبار را هر ساعت یا هر زمان خاصی به‌صورت خودکار انجام دهیم. تنظیمات مربوطه در [settings.py](#) موجود است و میتوان مدت زمان را در آنجا مشخص کرد و اینکه با هر بار شروع دوباره تسک چه پیامی نیز داده شود مشخص میشود.

Flower – مانیتورینگ Celery:

در انتها با استفاده از celery flower امکان مانیتور کردن تسک‌ها از طریق ادرس زیر خواهیم داشت:

<http://localhost:5555>

اجرای پروژه با Docker:

1. ساخت و اجرا:

`docker compose up --build -d`

2. برای مشاهده لاگ‌های سرویس‌های مختلف نیز میتوان از دستورات زیر استفاده کرد

```
docker compose logs -f web
```

```
docker compose logs -f celery
```

```
docker compose logs -f celery-beat
```

3. متوقف سازی:

```
docker compose down
```