

راهنمای پروژه YouTube Automation در مونوریپو Aladdin-sandbox

مقدمه

این سند یک راهنمای جامع برای پروژه `youtube_automation` است که در مونوریپو `aladdin-sandbox` ادغام شده است. این راهنما شامل دستورالعمل‌های نصب، نحوه اجرای تست‌ها، فرآیند دیپلوی و نکات مربوط به توسعه این پروژه در چارچوب سیستم CI/CD موجود است.

1. ساختار پروژه

پروژه `youtube_automation` در مسیر `aladdin-sandbox/apps/youtube_automation` قرار دارد. ساختار اصلی آن به شرح زیر است:

Plain Text

```
aladdin-sandbox/  
└─ apps/  
    └─ youtube_automation/  
        ├── __init__.py  
        ├── main.py  
        ├── requirements.txt  
        ├── tests/  
        │   └─ test_main.py  
        └─ ... (سایر فایل‌ها و ماژول‌های پروژه)
```

- `main.py`: نقطه ورود اصلی برنامه.
- `requirements.txt`: لیست وابستگی‌های پایتون پروژه.
- `tests/`: حاوی تست‌های واحد و یکپارچه‌سازی پروژه.

2. نصب و راه‌اندازی لوکال

برای نصب و راه‌اندازی پروژه `youtube_automation` به صورت لوکال، مراحل زیر را دنبال کنید:

1. پیمایش به دایرکتوری پروژه: ابتدا به دایرکتوری پروژه `youtube_automation` بروید:
2. ایجاد محیط مجازی (اختیاری اما توصیه می‌شود):
3. نصب وابستگی‌ها: تمامی وابستگی‌های پایتون مورد نیاز پروژه را نصب کنید:
4. پیکربندی متغیرهای محیطی: پروژه `youtube_automation` ممکن است به متغیرهای محیطی خاصی (مانند کلیدهای API یوتیوب) نیاز داشته باشد. این متغیرها را باید در یک فایل `env` در ریشه پروژه `youtube_automation` یا به عنوان متغیرهای محیطی سیستم عامل تنظیم کنید.

- مثال فایل `env`:

5. اجرای پروژه: پس از نصب وابستگی‌ها و پیکربندی متغیرهای محیطی، می‌توانید پروژه را اجرا کنید:

3. اجرای تست‌ها

سیستم CI/CD به طور خودکار تست‌ها را اجرا می‌کند، اما می‌توانید تست‌ها را به صورت لوکال نیز اجرا کنید:

1. پیمایش به دایرکتوری پروژه: به دایرکتوری `aladdin-sandbox/apps/youtube_automation` بروید.
2. فعال‌سازی محیط مجازی (اگر ایجاد کرده‌اید):
3. نصب `pytest` (اگر قبلاً نصب نشده است):
4. اجرای تست‌ها: دستور تست تعریف شده در `config/projects.yaml` برای این پروژه `pytest` است. آن را اجرا کنید:

4. فرآیند دیپلوی (استقرار)

دیپلوی پروژه `youtube_automation` از طریق سیستم CI/CD و GitHub Actions مدیریت می‌شود. این فرآیند شامل مراحل زیر است:

1. **تریگر Workflow:** هر `push` به شاخه `main` (یا برنچ‌های پیکربندی شده دیگر) که شامل تغییرات در مسیر `**/apps/youtube_automation` باشد، Workflow `YouTube Automation CI/CD` را فعال می‌کند.
2. **مراحل بیلد و تست:** Workflow ابتدا پروژه را بیلد کرده و تست‌های آن را اجرا می‌کند. این مرحله تضمین می‌کند که کد جدید پایدار و بدون خطا است.
3. **استقرار در محیط Test:** پس از موفقیت‌آمیز بودن بیلد و تست، پروژه به طور خودکار در محیط `Test` مستقر می‌شود.
4. **استقرار در محیط Staging (با تایید دستی):** پس از تایید در محیط `Workflow` ، `Test` به مرحله استقرار در محیط `Staging` می‌رسد. این مرحله نیاز به تایید دستی در GitHub Actions دارد.
5. **استقرار در محیط Production (با تایید دستی):** پس از تایید در محیط `Workflow` ، `Staging` به مرحله استقرار در محیط `Production` می‌رسد. این مرحله نیز نیاز به تایید دستی دارد.

نکات کلیدی برای دیپلوی:

- برای هر محیط (`Test`, `Staging`, `Production`) `render_service_id` مطمئن شوید که: `config/projects.yaml` تنظیم شده است `youtube_automation` به درستی در این فایل برای پروژه (`Production`).
- **GitHub Secrets:** (مانند) و سایر اطلاعات حساس مورد نیاز برای دیپلوی API کلیدهای `GitHub Secrets` باید به عنوان (یوتیوب API و کلیدهای `RENDER_API_KEY` باشند.
- **Environment Secrets:** در `Environment Secrets` برای متغیرهای محیطی خاص هر محیط، از `GitHub Actions` استفاده کنید.

5. توسعه پروژه

برای توسعه و بهبود پروژه `youtube_automation`، نکات زیر را در نظر بگیرید:

1. **ایجاد برنج جدید:** همیشه برای توسعه ویژگی‌های جدید یا رفع اشکالات، یک برنج جدید از `main` ایجاد کنید (مثلاً `feature/new-feature` یا `bugfix/fix-issue`).
2. **تغییرات کد:** تغییرات لازم را در فایل‌های پروژه اعمال کنید.
3. **نوشتن تست‌ها:** برای هر ویژگی جدید یا رفع اشکال، تست‌های مربوطه را بنویسید یا به‌روزرسانی کنید. این کار به حفظ کیفیت کد کمک می‌کند.
4. **اجرای تست‌های لوکال:** قبل از `push` کردن تغییرات، تست‌ها را به صورت لوکال اجرا کنید تا از صحت تغییرات خود مطمئن شوید.
5. **Commit و Push:** `commit` خود تغییرات خود را `push` کرده و به برنج خود `commit` کنید.
6. **ایجاد Pull Request:** یک Pull Request به شاخه `main` ایجاد کنید. این کار Workflow CI/CD را فعال می‌کند و کنترل‌های خودکار (بیلد و تست) بر روی کد شما اعمال می‌شود.
7. **بازبینی کد:** منتظر بازبینی کد توسط همکاران خود باشید و بر اساس بازخوردها، تغییرات لازم را اعمال کنید.
8. **ادغام با main:** پس از تایید PR و موفقیت‌آمیز بودن تمامی کنترل‌های PR، CI/CD را با `main` ادغام کنید. این کار فرآیند دیپلوی خودکار را آغاز خواهد کرد.

6. بهبود و گسترش CI/CD برای YouTube Automation

- **افزودن ابزارهای کیفیت کد:** می‌توانید ابزارهایی مانند `Flake8` یا `Black` را به Workflow CI/CD اضافه کنید تا کیفیت و فرمت کد به طور خودکار بررسی شود.
- **اسکن امنیتی:** ادغام ابزارهای اسکن امنیتی برای شناسایی آسیب‌پذیری‌ها در کد یا وابستگی‌های پروژه.
- **تست‌های عملکرد:** در صورت نیاز، تست‌های عملکرد را برای این پروژه پیاده‌سازی کنید تا از عملکرد بهینه آن اطمینان حاصل شود.

این راهنما به شما کمک می‌کند تا به طور موثر با پروژه `youtube_automation` در `monorepo` `aladdin-sandbox` کار کنید.