

# گزارش ادغام پروژه youtube\_automation\_v3 در مونوریپو aladdin-sandbox

## مقدمه

این گزارش به بررسی نحوه ادغام پروژه youtube\_automation\_v3 در مونوریپو aladdin-sandbox ، به طور خاص در زیرپوشه apps/youtube\_automation می‌پردازد. هدف این است که محتویات youtube\_automation\_v3 به گونه‌ای سازگار و منتقل شود که با ساختار و استانداردهای مونوریپو aladdin-sandbox همخوانی داشته باشد و امکان توسعه و استقرار یکپارچه را فراهم آورد.

## ساختار فعلی youtube\_automation\_v3

ریپازیتوری youtube\_automation\_v3 دارای ساختار زیر است:

Plain Text

```
youtube_automation_v3/
├── CHANGELOG.md
├── DEPLOY_GUIDE.md
├── LICENSE.txt
├── README.md
├── VERSION
├── docs/
├── infra/
├── install.sh
├── scripts/
└── youtube_automation_v3/
    ├── DEPLOYMENT_GUIDE_EN.md
    ├── DEPLOYMENT_GUIDE_FA.md
    ├── Dockerfile
    ├── __init__.py
    ├── api/
    ├── core/
    ├── docker-compose.yml
    ├── main.py
    ├── models/
    ├── plugins/
    ├── providers/
    ├── repositories/
    ├── requirements.txt
    ├── schemas/
    ├── services/
    └── tasks/
```

```
└─ test_api.py
└─ tests/
```

همانطور که مشاهده می‌شود، هسته اصلی برنامه در زیرپوشه `youtube_automation_v3/youtube_automation_v3` قرار دارد. همچنین، فایل‌های مستندات، اسکریپت‌های نصب و فایل‌های مربوط به زیرساخت (`infra`) در سطح ریشه ریپازیتوری اصلی `youtube_automation_v3` قرار گرفته‌اند.

## ساختار هدف در `aladdin-sandbox`

در مونوریپو `aladdin-sandbox`، پروژه‌ها در زیرپوشه `/apps` قرار می‌گیرند. بنابراین، پروژه `youtube_automation_v3` باید به `aladdin-sandbox/apps/youtube_automation` منتقل شود. ساختار مورد انتظار برای این پروژه در مونوریپو به شرح زیر است:

Plain Text

```
aladdin-sandbox/
└─ apps/
  └─ youtube_automation/
    └─ api/
    └─ core/
    └─ docs/ # مستندات پروژه خاص
    └─ infra/ # زیرساخت پروژه خاص (مانند Dockerfile)
    └─ models/
    └─ plugins/
    └─ providers/
    └─ repositories/
    └─ requirements.txt
    └─ schemas/
    └─ services/
    └─ tasks/
    └─ tests/
    └─ main.py
    └─ test_api.py
    └─ ... (سایر فایل‌های مربوط به پروژه)
└─ config/
└─ infra/
└─ libs/
└─ reports/
└─ ... (سایر پوشه‌ها و فایل‌های مونوریپو)
```

مراحل ادغام

برای ادغام موفقیت‌آمیز `youtube_automation_v3` در `aladdin-sandbox/apps/youtube_automation` ، مراحل زیر باید دنبال شوند:

## 1. انتقال فایل‌های اصلی برنامه

تمامی محتویات پوشه `youtube_automation_v3/youtube_automation_v3` باید به `aladdin-sandbox/apps/youtube_automation` منتقل شوند. این شامل پوشه‌های `api/` ، `core/` ، `models/` ، `plugins/` ، `providers/` ، `repositories/` ، `schemas/` ، `services/` ، `tasks/` ، `tests` و فایل‌هایی مانند `main.py` ، `requirements.txt` ، `test_api.py` ، `Dockerfile` ، `docker-compose.yml` و فایل‌های مستندات خاص پروژه ( `DEPLOYMENT_GUIDE_EN.md` ، `DEPLOYMENT_GUIDE_FA.md` ) است.

## 2. مدیریت مستندات

- **مستندات پروژه خاص:** فایل‌های `DEPLOYMENT_GUIDE_EN.md` و `DEPLOYMENT_GUIDE_FA.md` که در `youtube_automation_v3/youtube_automation_v3/docs` قرار دارند، باید به یک پوشه `/docs` در داخل `aladdin-sandbox/apps/youtube_automation` منتقل شوند. این کار به سازماندهی مستندات خاص هر پروژه کمک می‌کند.
- **مستندات عمومی:** فایل‌های `LICENSE.txt` و `CHANGELOG.md` ، `DEPLOY_GUIDE.md` ، `README.md` که در ریشه `youtube_automation_v3` قرار دارند، باید بررسی شوند. اگر این مستندات مربوط به کل مونوریپو نیستند و فقط به `youtube_automation` مربوط می‌شوند، باید به `aladdin-sandbox/apps/youtube_automation/docs` منتقل شوند. در غیر این صورت، ممکن است نیاز به ادغام یا بازنویسی در مستندات عمومی مونوریپو ( `/aladdin-sandbox/docs` ) داشته باشند.

## 3. مدیریت اسکریپت‌ها و زیرساخت

- **اسکریپت‌های نصب:** فایل `install.sh` و پوشه `/scripts` در ریشه `youtube_automation_v3` باید بررسی شوند. اگر این اسکریپت‌ها برای نصب یا اجرای `youtube_automation` به صورت مستقل طراحی شده‌اند، باید به `aladdin-sandbox/apps/youtube_automation/scripts` منتقل شوند و در صورت لزوم برای کار در محیط مونوریپو اصلاح گردند. اگر این اسکریپت‌ها عمومی هستند، باید با اسکریپت‌های موجود در `aladdin-sandbox/scripts` ادغام شوند.
- **فایل‌های زیرساخت:** پوشه `/infra` در ریشه `youtube_automation_v3` باید بررسی شود. اگر حاوی فایل‌های زیرساختی خاص `youtube_automation` است، باید به `aladdin-sandbox/apps/youtube_automation/infra` منتقل شود. `Dockerfile` و `docker-compose.yml` که در `youtube_automation_v3/youtube_automation_v3` قرار دارند، باید به `aladdin-sandbox/apps/youtube_automation` منتقل شوند و در صورت نیاز برای کار در مونوریپو و هماهنگی با سیستم CI/CD موجود (مانند `mamos_runner.py` ) اصلاح گردند.

## 4. به‌روزرسانی `requirements.txt`

فایل `requirements.txt` در `aladdin-sandbox/apps/youtube_automation` باید شامل تمامی وابستگی‌های پایتون مورد نیاز برای `youtube_automation_v3` باشد. اطمینان حاصل شود که هیچ وابستگی تکراری یا متناقضی با سایر پروژه‌های مونوریپو وجود ندارد.

## 5. به روزرسانی config/projects.yaml

فایل `aladdin-sandbox/config/projects.yaml` باید به روزرسانی شود تا پروژه `youtube_automation` را با مسیر صحیح و دستورات `build` و `test` مناسب برای آن تعریف کند. مثال:

YAML

```
- name: youtube_automation
  path: apps/youtube_automation
  build: "pip install -r requirements.txt"
  test: "pytest"
  deploy:
    - environment: Test
      render_service_id: "youtube-automation-test-service-id"
    - environment: Staging
      render_service_id: "youtube-automation-staging-service-id"
    - environment: Production
      render_service_id: "youtube-automation-production-service-id"
```

## 6. به روزرسانی GitHub Actions Workflow

فایل `aladdin-sandbox/.github/workflows/youtube_automation.yml` باید به روزرسانی شود تا با ساختار جدید پروژه و `mamos_runner.py` سازگار باشد. اطمینان حاصل شود که `paths` در `on: push` به `**/apps/youtube_automation` اشاره می کند و دستورات `build` و `test` در `mamos_runner.py` به درستی فراخوانی می شوند.

## 7. بررسی مسیرهای داخلی و واردات (Imports)

پس از انتقال فایل ها، ممکن است نیاز باشد مسیرهای داخلی (`internal paths`) و دستورات `import` در کد پایتون `youtube_automation_v3` اصلاح شوند تا با ساختار جدید در `aladdin-sandbox/apps/youtube_automation` سازگار باشند. به عنوان مثال، اگر `youtube_automation_v3` خود را به عنوان یک پکیج در نظر می گرفت، اکنون باید به عنوان یک زیرپکیج در `apps.youtube_automation` رفتار کند.

## 8. تست جامع

پس از انجام تمامی مراحل فوق، یک تست جامع برای پروژه `youtube_automation` در مونوریپو ضروری است. این شامل اجرای `mamos_runner.py --project youtube_automation` برای اطمینان از موفقیت آمیز بودن مراحل ساخت و تست، و همچنین بررسی عملکرد برنامه در محیط توسعه محلی است.

## نتیجه گیری

با پیروی از این مراحل، پروژه `youtube_automation_v3` می تواند به طور موثر در مونوریپو `aladdin-sandbox` ادغام شود. این ادغام نه تنها به سازماندهی بهتر کد کمک می کند، بلکه از مزایای سیستم CI/CD

یکپارچه مونوریپو نیز بهره‌مند خواهد شد و فرآیند توسعه، تست و استقرار را کارآمدتر می‌سازد.