

راه اندازی یک سرویس File Uploader

۱- توضیحات

هدف از این پروژه تستی ایجاد یک ساختار API ساده برای ارتباط بین سرویس MinIO و وب سرویس جانگو با استفاده از Rest Framework است. الزامات مورد نیاز برای انجام این پروژه آشنایی با موارد زیر است:

- Django
- Django Rest Framework
- MinIO Object Storage
- Docker
- Unit & Integration Testing
- Postgresql
- JWT Token Management System

به این صورت که در این پروژه کوچک هدف این است تا یک وب سرویس که مشخصات آن ارائه خواهد شد، پیاده سازی شود که از طریق API فایل را از کاربر دریافت کند این اطلاعات را بر روی سرور ارسال کند و این فایل ها در S3 ذخیره شده و بتوان آن ها را مدیریت نمود. ضمن اینکه تمامی موارد نیاز است تا یک تست برای آن ها توسعه داده شود.

۲- جزئیات اپ مورد نظر

در این وب سرویس که هدف کلی آن ارائه یک سرویس Uploader است نیاز است تا یک سرویس با نام uploader ایجاد شود که دارای یک table با اطلاعات زیر است (اسم مدل ترجیحا UploadedFile در نظر گرفته شود).

- کاربر ارسال کننده درخواست
- فایل ارسال شده در MinIO
- نوع فایل
- حجم فایل
- یک تابع در مدل که در زمان حذف آیتم فایل اصلی در MinIO را نیز حذف کند.

۳- وب سرویس‌ها

که وب سرویس‌های زیر نیاز است تا ارائه شوند:

۳-۱- وب سرویس ایجاد فایل

- وب سرویس برای ایجاد فایل با متد POST که از ورودی کاربر فایل را دریافت می‌کند و این فایل را در MinIO ذخیره می‌کند و اطلاعات آن مثل حجم و نوع فایل را در دیتابیس ذخیره می‌کند. ضمن اینکه اطلاعات کاربر را از طریق توکن ارسال می‌کند و به هیچ عنوان کاربر نباید بتواند از طریق ارسال user_id خود در ورودی data درخواست فایل را به اسم کاربری دیگر ارسال و در سیستم ذخیره کند.

POST /api/upload/

Input data: file <formData>

Output Json: { 'id': <id>, 'file_url': 'Address to MinIO file path for download', 'type': 'pdf', 'size': 12000 <in byte> }

۳-۲- وب سرویس مشاهده فایل‌های آپلود شده

- وب سرویس که لیست فایل‌های ایجاد شده توسط کاربر را دریافت می‌کند.

GET /api/upload/

۳-۳- وب سرویس حذف فایل

که id فایل در مدل UploadedFile را به صورت زیر دریافت کرده و فایل را از مدل جانگو و از روی سرویس MinIO پاک می‌کند.

DELETE /api/upload/<id>/

۴- تست‌های مورد نیاز

برای توسعه این سرویس خروجی مورد نظر از طریق تست‌های ارائه شده بررسی می‌شود. به همین جهت نیاز است تا test case‌های زیر توسعه داده شوند:

۴-۱- تست ایجاد فایل

که در این تست یک فایل به صورت sample ایجاد می‌شود برای سرور ارسال می‌شود و این فایل باید ذخیره شود و چک شود که آیا این آیتم ایجاد شده و در دیتابیس postgres ذخیره شده و از طریق ایا اطلاعات آبجکت در MinIO نیز درست است یا نه و در نهایت چک شود که آیا مسیر فایل آپلود شده در خود MinIO در دسترس هست یا نه.

۲-۴- تست مشاهده لیست فایل ها

یک سری فایل ایجاد شود و چک شود که ایا اطلاعات این فایل ها از طریق API در دسترس هست یا نه

۳-۴- تست برای حذف فایل

یک تست که ابتدا یک فایل برای کاربر ایجاد کند و سپس کاربر این فایل را حذف کند و تست که ایا از روی مدل های دیتابیس و MinIO به درستی پاک شده است یا نه.

۴-۴- Integration Test

علاوه بر تست های بالا یک تست Integration هم به صورت ساده برای چک کردن اتصال جانگو با سرویس MinIO نوشته شود.

۵- خروجی های مورد انتظار

به منظور دریافت خروجی نهایی نیاز است تا موارد زیر ارائه شوند:

- کدهای سرویس ارائه شده

- فایل داکر های پروژه (docker compose اختیاری است) و اگر به صورت جدا هر کدام از داکر های MinIO و سرویس توسعه داده شده و postgres اجرا می شوند همان دستور های اجرای هر کدام به صورت مجزا ارائه شود کافی است.

- ارائه test case های ارائه شده

- خروجی Swagger برای API ها

مدت زمان تقریبی انجام پروژه برای یک شخص با سابقه بیش از ۳ سال برنامه نویسی جانگو: **۶ ساعت**

با تشکر