АРХИТЕКТУРНЫЙ ФУНДАМЕНТ ДЛЯ АВТОМАТИЗАЦИИ REST API - С ЧЕГО НАЧАТЬ?

Алексей Коледачкин
QA Automation | CEO “QA Playground”
Автор курса AQA PROKA4

Наша цель

Решить проблему с фундаментом, которая часто возникает у новичков и не только, а именно, как начать автоматизацию API и собрать гибкую архитектуру

Основные поинты

1. Избежать спагетти-кода
2. Сделать структуру гибкой и расширяемой
3. Повысить читабельность
4. Увеличить скорость разработки тестов

Общая структура проекта

```
PROJECT
├── config
│   ├── base_test.py
│   ├── headers.py
│   └── stages.py
├── services
│   ├── service_1
│   │   ├── models
│   │   │   └── model_1.py
│   │   ├── api.py
│   │   ├── endpoints.py
│   │   └── payloads.py
│   └── service_2
│       ├── models
│       │   └── model_1.py
│       ├── api.py
│       ├── endpoints.py
│       └── payloads.py
├── tests
│   └── test_example_1.py
├── utils
│   └── helper.py
├── .env
├── conftest.py
└── requirements.txt
```

/config - общие для всех тестов сущности

/services - все api нашего проекта

/services/models - pydantic-модели для валидации

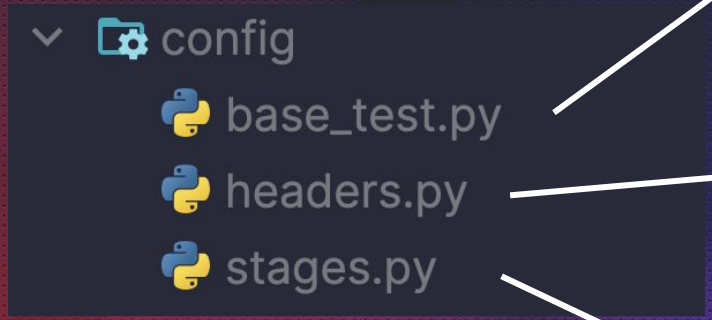
/tests - сами автотесты

/utils - дополнительные библиотеки

.env - файл для хранения чувствительных данных

conftest.py - файл для фикстур

Общие файлы проекта

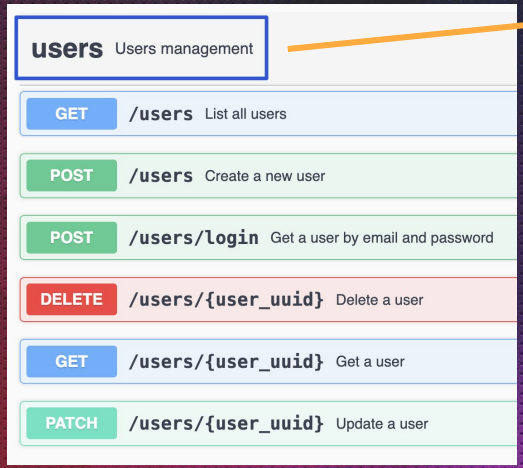


Тут хранятся доступные для всех тестов сущности

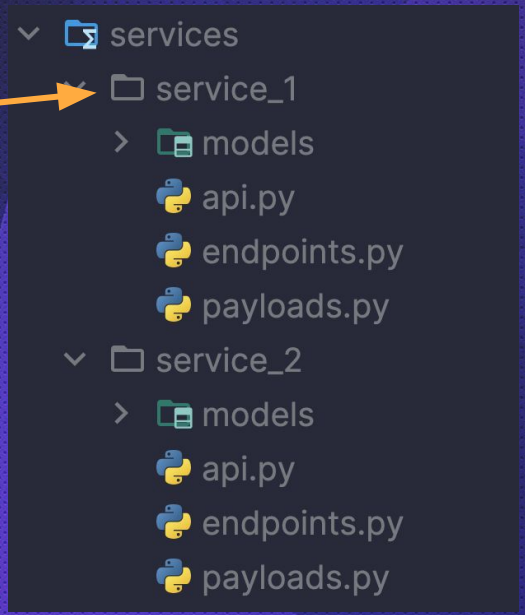
Здесь храним хедеры (статические и динамические)

Тут реализуем управление стейджами проекта

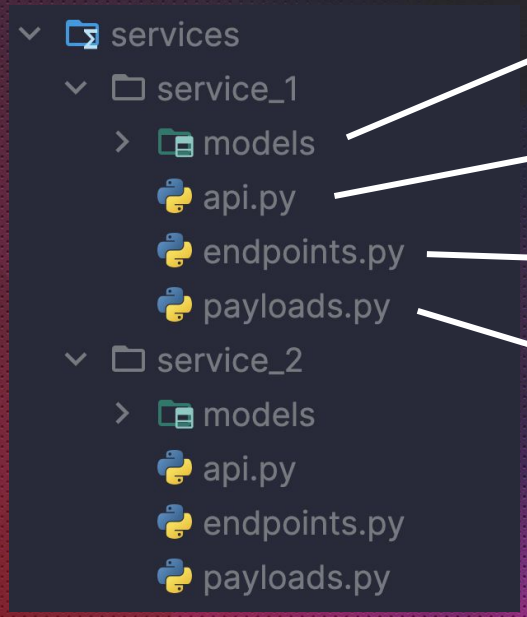
Документация API



Сервисы



API-интерфейс



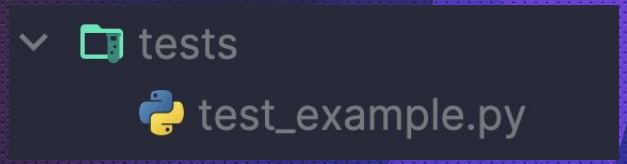
Тут хранятся pydantic-модели для валидации ответов сервера

Здесь реализуем API-методы (шаги тестов)

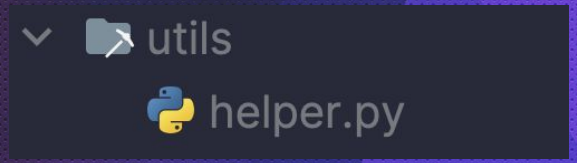
Тут храним статичные и динамические эндпоинты сервиса

Тут храним статические и динамические тела запросов

Сами тесты



Самописные помощники



План

1. Реализовать **headers** и **stages**
2. Расписать 1 сервис
3. Написать первый тест
4. Интегрировать **Allure** и **Pydantic**
5. Написать второй сервис
6. Написать тест с двумя эндпоинтами

Советы

1. Использование сессий
2. Адаптировать тесты под негативные
3. Создание более полезного хелпера
4. Реализовать логирование
5. Развиваться, пробовать и не бояться!