

## # Домашнее задание 02: Разработка REST API сервиса

**Цель работы:** Получить практические навыки разработки REST API сервиса с использованием принципов REST, обработкой HTTP запросов, реализацией аутентификации и документированием API.

### ## Задание

Выберите нужный вариант из файла `homework\_variants.pdf` (варианты 1–24) и выполните следующие задачи:

#### 1. Проектирование REST API

- Изучите выбранный вариант задания
- Спроектируйте REST API endpoints для всех операций из вашего варианта
- Используйте правильные HTTP методы (GET, POST, PUT, DELETE, PATCH)
- Используйте правильные HTTP статус-коды
- Спроектируйте структуру URL (ресурсы, вложенные ресурсы)
- Определите структуру Request/Response для каждого endpoint

#### 2. Реализация REST API сервиса

- Реализуйте REST API сервис на выбранном языке и фреймворке (Python/Flask, Node.js/Express, Java/Spring Boot и т.д.)
- Реализуйте минимум 5 API endpoints из вашего варианта задания
- Используйте in-memory хранилище (списки, словари) или простую БД (SQLite)
- Реализуйте обработку ошибок с правильными HTTP статус-кодами
- Используйте DTO (Data Transfer Objects) для передачи данных

#### 3. Реализация аутентификации

- Реализуйте простую аутентификацию (можно использовать JWT токены или session-based)
- Защитите минимум 2 endpoint с помощью аутентификации
- Реализуйте endpoint для регистрации/логина пользователя
- Добавьте middleware для проверки аутентификации

#### 4. Документирование API

- Создайте OpenAPI/Swagger спецификацию для вашего API
- Опишите все endpoints с параметрами, request/response схемами
- Добавьте примеры запросов и ответов
- Если возможно, добавьте Swagger UI для интерактивного тестирования API

#### 5. Тестирование

- Создайте простые тесты для основных endpoints (можно использовать curl, Postman или unit-тесты)
- Протестируйте успешные сценарии
- Протестируйте обработку ошибок (невалидные данные, отсутствующие ресурсы и т.д.)

### ## Результат

Результат должен быть оформлен в виде следующих файлов, размещенных в вашем GitHub репозитории:

- Исходный код REST API сервиса
- `openapi.yaml` или `swagger.yaml` – OpenAPI спецификация
- `README.md` – с описанием проекта, инструкциями по запуску и примерами использования API
- Тесты (если реализованы)
- Dockerfile и docker-compose.yaml с контейнером, запускающим API

**\*\*Критерии оценки:\*\***

- Соответствие принципам REST
- Корректность использования HTTP методов и статус-кодов
- Качество реализации API endpoints
- Работоспособность аутентификации
- Полнота и качество OpenAPI документации
- Качество кода и структура проекта