

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 FastAPI, C++ Poco, Yandex Userver)
- 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

Варианты

1 Социальная сеть <https://facebook.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Стена
- Сообщения чата (P2P)

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Добавление записи на стену
- Загрузка стены пользователя
- Отправка сообщения пользователю
- Получение списка сообщения для пользователя

2 Магазин <https://ozon.ru/>

Приложение должно содержать следующие данные:

- Пользователь
- Товар
- Корзина

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание товара
- Получение списка товаров
- Добавление товара в корзину
- Получение корзины для пользователя

3 Сайт конференции <https://www.eventboost.com/ru-RU/>

Приложение должно содержать следующие данные:

- Пользователь
- Доклад
- Конференция

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание доклада
- Получение списка всех докладов
- Добавление доклада в конференцию
- Получение списка докладов в конференции

4 Сайт заказа услуг <https://profiru.ru/>

Приложение должно содержать следующие данные:

- Пользователь
- Услуга
- Заказ

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание услуги
- Получение списка услуг
- Добавление услуг в заказ
- Получение заказа для пользователя

5 Мессенджер <https://slack.com/intl/en-gb/>

Приложение должно содержать следующие данные:

- Пользователь
- Групповой чат
- PtP Чат

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание группового чата
- Добавление пользователя в чат
- Добавление сообщения в групповой чат
- Загрузка сообщений группового чата
- Отправка PtP сообщения пользователю
- Получение PtP списка сообщения для пользователя

6 Сервис доставки <https://www.cdek.ru/ru/>

Приложение должно содержать следующие данные:

- Пользователь
- Посылка
- Доставка

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание посылки
- Получение посылок пользователя
- Создание доставки от пользователя к пользователю
- Получение информации о доставке по получателю
- Получение информации о доставке по отправителю

7 Сервис поиска попутчиков <https://www.blablacar.ru/>

Приложение должно содержать следующие данные:

- Пользователь
- Маршрут
- Поездка

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание маршрута
- Получение маршрутов пользователя
- Создание поездки
- Подключение пользователей к поездке
- Получение информации о поездке

8 Управление проектами <https://www.atlassian.com/ru/software/jira>

Приложение должно содержать следующие данные:

- проект
- задача
- исполнитель

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание проекта
- Поиск проекта по имени
- Поиск всех проектов
- Создание задачи в проекте
- Получение всех задач в проекте
- Получение задачи по коду

9 Электронная почта <https://outlook.live.com/>

Приложение должно содержать следующие данные:

- почтовая папка
- сообщение
- пользователь

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание новой почтовой папки
- Получение перечня всех папок
- Создание нового письма в папке
- Получение всех писем в папке
- Получение письма по коду

10 Планирование задач <https://www.leadertask.ru/>

Приложение должно содержать следующие данные:

- цель
- задача
- исполнитель

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание новой цели
- Получение списка всех целей
- Создание новой задачи на пути к цели
- Получение всех задач цели
- Изменение статуса задачи в цели

11 Хранение файлов <https://360.yandex.ru/disk/>

Приложение должно содержать следующие данные:

- папка
- файл
- пользователь

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание новой папки
- Получение списка всех папок
- Создание файла в папке
- Получение файла по имени
- Удаление файла
- Удаление папки

12 Бюджетирование <https://about.coinkeeper.me/>

Приложение должно содержать следующие данные:

- планируемый доход
- планируемый расход
- пользователь

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создать планируемый доход
- Получить перечень планируемых доходов
- Создать планируемый расход
- Получить перечень планируемых расходов
- Посчитать динамику бюджета за период

13 Система бронирования отелей <https://www.booking.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Отель
- Бронирование

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание отеля
- Получение списка отелей
- Поиск отелей по городу
- Создание бронирования
- Получение бронирований пользователя
- Отмена бронирования

14 Фитнес-трекер <https://www.myfitnesspal.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Тренировка
- Упражнение

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание упражнения
- Получение списка упражнений
- Создание тренировки
- Добавление упражнения в тренировку
- Получение истории тренировок пользователя
- Получение статистики тренировок за период

15 Система управления библиотекой <https://www.librarything.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Книга
- Выдача

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Добавление книги в библиотеку
- Поиск книги по названию
- Поиск книги по автору
- Создание выдачи книги пользователю
- Получение списка выданных книг пользователя
- Возврат книги

16 Система заказа такси <https://www.uber.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Водитель
- Поездка

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Регистрация водителя
- Создание заказа поездки
- Получение активных заказов
- Принятие заказа водителем
- Получение истории поездок пользователя
- Завершение поездки

17 Система управления рестораном <https://www.opentable.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Блюдо
- Заказ

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Добавление блюда в меню
- Получение меню ресторана
- Поиск блюд по категории
- Создание заказа
- Добавление блюда в заказ
- Получение заказов пользователя
- Изменение статуса заказа

18 Система управления складом <https://www.zoho.com/inventory/>

Приложение должно содержать следующие данные:

- Пользователь
- Товар
- Поступление

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Добавление товара на склад
- Поиск товара по названию
- Получение остатков товаров
- Создание поступления товара
- Получение истории поступлений
- Списание товара со склада

Домашнее задание по курсу «Архитектура программных систем»

19 Система управления обучением <https://www.moodle.org/>

Приложение должно содержать следующие данные:

- Пользователь
- Курс
- Урок

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание курса
- Получение списка курсов
- Добавление урока в курс
- Получение уроков курса
- Запись пользователя на курс
- Получение курсов пользователя
- Отметка о прохождении урока

20 Система управления медицинскими записями <https://www.epic.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Пациент
- Запись

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Регистрация пациента
- Поиск пациента по ФИО
- Создание медицинской записи
- Добавление записи к пациенту
- Получение истории записей пациента
- Получение записи по коду

21 Система управления арендой автомобилей <https://www.hertz.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Автомобиль
- Аренда

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Добавление автомобиля в парк
- Получение списка доступных автомобилей
- Поиск автомобилей по классу
- Создание аренды
- Получение активных аренд пользователя
- Завершение аренды
- Получение истории аренд

22 Система управления событиями <https://www.eventbrite.com/>

Домашнее задание по курсу «Архитектура программных систем»

Приложение должно содержать следующие данные:

- Пользователь
- Событие
- Участник

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание события
- Получение списка событий
- Поиск событий по дате
- Регистрация пользователя на событие
- Получение участников события
- Получение событий пользователя
- Отмена регистрации на событие

23 Система управления рецептами <https://www.allrecipes.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Рецепт
- Ингредиент

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание рецепта
- Получение списка рецептов
- Поиск рецептов по названию
- Добавление ингредиента в рецепт
- Получение ингредиентов рецепта
- Получение рецептов пользователя
- Добавление рецепта в избранное

24 Система управления недвижимостью <https://www.zillow.com/>

Приложение должно содержать следующие данные:

- Пользователь
- Объект недвижимости
- Просмотр

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Добавление объекта недвижимости
- Поиск объектов по городу
- Поиск объектов по цене
- Запись на просмотр объекта
- Получение записей на просмотр объекта
- Получение объектов пользователя
- Изменение статуса объекта

