

Техническое задание: Разработка REST API для управления списком задач (To-Do List)

1. Введение

Необходимо разработать REST API для управления списком задач. API должно позволять пользователям создавать, редактировать, просматривать и удалять задачи.

2. Технологический стек

- Разработчик может выбрать один из следующих стеков технологий:
- Backend: Node.js (Express) или PHP (чистый или с использованием Laravel)
- База данных: JSON-файл, SQLite, MySQL или MongoDB

3. Функционал API

Методы API:

POST /tasks – Создать новую задачу

Входные параметры (JSON):

- title (string) – Заголовок задачи (обязательный)
- description (string) – Описание задачи (необязательный)
- status (string) – Статус задачи (pending, in progress, completed)

Выходные данные: Объект созданной задачи с id

GET /tasks – Получить список всех задач

Выходные данные: массив задач

GET /tasks/{id} – Получить конкретную задачу

Входные параметры: id задачи

Выходные данные: объект задачи

PUT /tasks/{id} – Обновить задачу

Входные параметры (JSON):

- title (string) – Заголовок (обновляемый)
- description (string) – Описание (обновляемый)
- status (string) – Статус (обновляемый)

Выходные данные: обновленный объект задачи

DELETE /tasks/{id} – Удалить задачу

Входные параметры: id задачи

Выходные данные: подтверждение удаления

4. Frontend

- Простое отображение списка задач с возможностью добавления, редактирования и удаления.
- Можно использовать любой фреймворк (React, Vue, Angular) или чистый HTML + JavaScript.
- Минимальное количество CSS, оформление не критично.

5. Дополнительные требования

Структурированный код:

Разделить логику на файлы (контроллеры, маршруты, сервисы, модели).

Аутентификация (необязательно, если сложно):

- Реализовать JWT-аутентификацию (по желанию), чтобы пользователи могли получать и изменять только свои задачи.
- Если JWT сложен, можно обойтись без аутентификации.

Роли пользователей (дополнительно, если возможно):

- Администратор – Полный доступ ко всем задачам.
- Обычный пользователь – Доступ только к своим задачам.

6. Требования к коду

- Чистота и читаемость кода.
- Обработчики ошибок.
- Минимальная документация (README с инструкцией по запуску).