

Разработка REST API для управления задачами

Задача:

Создание REST API для взаимодействия с модулем управления задачами в веб-приложении. API обеспечивает возможность создания, редактирования, удаления и получения информации о задачах, а также работы с проектами, пользователями и статусами задач.

Цели:

1. Разработать удобный REST API, поддерживающий все основные операции CRUD.
2. Реализовать безопасную и масштабируемую архитектуру API.
3. Обеспечить соответствие стандартам REST.
4. Подготовить документацию для использования API.

Решение:

Функциональные требования:

1. CRUD для задач:
 - Создание новой задачи: POST /tasks
 - Получение задачи по ID: GET /tasks/{id}
 - Обновление задачи: PUT /tasks/{id}
 - Удаление задачи: DELETE /tasks/{id}
2. Работа с проектами:
 - Получение списка проектов: GET /projects
 - Получение задач по проекту: GET /projects/{id}/tasks
3. Управление статусами:
 - Получение списка статусов: GET /statuses
4. Пользователи:
 - Назначение исполнителя для задачи: PATCH /tasks/{id}/assign
5. Фильтрация и поиск:
 - Фильтрация задач по статусу, приоритету, исполнителю:
GET /tasks?status=completed&priority=high&assignee=3
6. Аутентификация и авторизация:
 - Использование JWT для проверки подлинности пользователей.
 - Ограничение доступа к задачам на основе ролей пользователей.

Технические детали:

Архитектура: RESTful

Технологии:

- Backend: Python
- База данных: PostgreSQL
- Аутентификация: JWT (JSON Web Tokens)

Методы HTTP: GET, POST, PUT, DELETE, PATCH

Коды статусов HTTP:

- 200 — Успешный запрос
- 201 — Создано
- 400 — Неверный запрос
- 401 — Неавторизован
- 404 — Не найдено
- 500 — Ошибка сервера

Документация API:

```
1 openapi: 3.0.0
2 info:
3   title: API для управления задачами
4   version: 1.0.0
5   description: API для управления задачами.
6
7 paths:
8   /tasks:
9     get:
10       summary: Получить список задач
11       responses:
12         '200':
13           description: Успешный запрос
14           content:
15             application/json:
16               schema:
17                 type: array
18                 items:
19                   $ref: '#/components/schemas/Task'
20       post:
21         summary: Создать задачу
22         requestBody:
23           description: Данные для новой задачи
24           required: true
25           content:
26             application/json:
27               schema:
28                 $ref: '#/components/schemas/Task'
29         responses:
30           '201':
31             description: Задача создана
32
33 components:
34   schemas:
```

API для управления задачами

1.0.0 OAS 3.0

API для управления задачами.

default

GET /tasks Получить список задач

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Успешный запрос	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
33 components:
34   schemas:
35     Task:
36       type: object
37       properties:
38         id:
39           type: integer
40           description: Уникальный идентификатор задачи
41         title:
42           type: string
43           description: Заголовок задачи
44         description:
45           type: string
46           description: Описание задачи
47         priority:
48           type: integer
49           description: Приоритет задачи (1 - высокий, 2 - средний, 3 -
50             низкий)
```

POST /tasks Создать задачу

Parameters Try it out

No parameters

Request body required application/json

Данные для новой задачи

Example Value | Schema

```
{
  "id": 0,
  "title": "string",
  "description": "string",
  "priority": 0
}
```

Responses

Code	Description	Links
201	Задача создана	No links

Результаты:

1. Полностью реализованный REST API для управления задачами.
2. Подключение API к модулю управления проектами.
3. Автоматические тесты на основные операции API.
4. Полноценная документация для разработчиков.

Алексей Лужков
Системный аналитик
89040877877@ya.ru