

### Шаг 1: Установка зависимостей

- Откройте терминал или командную строку и перейдите в папку с кодом.
- Выполните команду `npm install` для установки всех необходимых зависимостей.

### Шаг 2: Запуск кода

- Выполните команду `npm run dev` для запуска кода.

### Шаг 3: Настройка подключения к базе данных

- Откройте файл `.env` в папке с кодом и заполните ее своими данными:

...

`PORT` = порт, на котором будет работать сервер.

`DB_NAME` = имя базы данных.

`DB_USER` = имя пользователя базы данных.

`DB_PASSWORD` = пароль пользователя базы данных.

`DB_HOST` = хост базы данных.

`DB_PORT` = порт базы данных.

`JWT_ACCESS_SECRET` = секретный ключ для токенов доступа.

`JWT_REFRESH_SECRET` = секретный ключ для токенов обновления.

...

## Шаг 5: Примеры запросов в Postman

### Регистрация

...

POST /register HTTP/1.1

Content-Type: application/json

```
{  
  "email": "john.doe@example.com",  
  "password": "password123",  
  "avatar": "https://example.com/avatar.jpg"  
}
```

### Авторизация

...

POST /login HTTP/1.1

Content-Type: application/json

```
{  
  "email": "john.doe@example.com",  
  "password": "password123"  
}
```

### **Создание предложения**

...

POST /createfeedback HTTP/1.1

Authorization: Bearer access token

Content-Type: application/json

```
{  
  "title": "баг",  
  "description": "описание бага",  
  "category": "Bug ",  
  "status": "In Progress "  
}
```

...

### **Голосование за предложение**

...

POST /upvote HTTP/1.1

Authorization: Bearer access token

Content-Type: application/json

```
{  
  "postId": 1  
}
```

...

### **Получение предложения**

...

GET /receivefeedback HTTP/1.1

...

### **Получение списка категорий**

...

GET /categories HTTP/1.1

...

### **Получение списка статусов**

...

GET /statuses HTTP/1.1

...

## Получение предложений с фильтрацией, сортировкой и пагинацией

### Параметры фильтрации

- **category:** категория предложения (например, "Feature", "Planned", etc.)
- **status:** статус предложения (например, "status1", "status2", etc.)
- **sortBy:** поле, по которому нужно сортировать предложения (например, "votes", "createdAt", etc.)
- **order:** порядок сортировки (например, "ASC", "DESC")
- **page:** номер страницы (например, 1, 2, etc.)
- **limit:** количество предложений на странице (например, 10, 20, etc.)

### Сортировка по количеству голосов в порядке убывания

...

GET /feedbackposts?sortBy=votes&order=DESC HTTP/1.1

...

### Сортировка по количеству голосов в порядке возрастания

...

GET /feedbackposts?sortBy=votes&order=ASC HTTP/1.1

...

### Сортировка по дате создания в порядке убывания

...

GET /feedbackposts?sortBy=createdAt&order=DESC HTTP/1.1

...

### Сортировка по названию в порядке убывания

...

GET /feedbackposts?sortBy=title&order=DESC HTTP/1.1

...

### Примеры использования команд сортировки

Получить все предложения, отсортированные по количеству голосов в порядке убывания:

...

GET /feedbackposts?sortBy=votes&order=DESC HTTP/1.1

...

Получить все предложения, отсортированные по дате создания в порядке возрастания:

...

GET /feedbackposts?sortBy=createdAt&order=ASC HTTP/1.1

...

Получить все предложения, отсортированные по названию в порядке убывания:

``GET /feedbackposts?sortBy=title&order=DESC HTTP/1.1``