

## Общее описание

Этот REST API предоставляет функционал для создания, получения, обновления и удаления пользователей. Он поддерживает фильтрацию и пагинацию, а также простую авторизацию на основе имени пользователя и пароля. API использует базу данных PostgreSQL для хранения информации о пользователях, а библиотека go-pg/pg используется для взаимодействия с базой данных.

## Структуры данных

### User

```
// User структура для хранения информации о пользователе
type User struct {
    ID      int    `json:"id"`
    Name    string `json:"name" validate:"required,min=2,max=100"`
    Email   string `json:"email" validate:"required,email"`
    Age     int    `json:"age" validate:"gte=0,lte=130"`
}
```

**ID:** Уникальный идентификатор пользователя.

**Name:** Имя пользователя (обязательное поле, длина от 2 до 100 символов).

**Email:** Адрес электронной почты пользователя (обязательное поле, должно быть валидным email).

**Age:** Возраст пользователя (должен быть в диапазоне от 0 до 130 лет).

### AuthRequest

Структура для авторизации.

```
// AuthRequest структура для хранения данных авторизации
type AuthRequest struct {
    Username string `json:"username"`
    Password string `json:"password"`
}
```

**Username:** Имя пользователя для входа.

**Password:** Пароль пользователя.

## Функции

### connectDB()

Подключается к базе данных PostgreSQL с использованием строки подключения.

### createSchema()

Создает таблицу для хранения пользователей, если она не существует.

### getUsers()

Получает список пользователей с поддержкой фильтрации и пагинации.

### getUser()

Получает пользователя по его уникальному идентификатору (ID).

### createUser()

Создает нового пользователя в базе данных.

### updateUser()

Обновляет информацию о существующем пользователе.

### deleteUser()

Удаляет пользователя из базы данных по его ID.

### loginHandler()

Обрабатывает запросы на авторизацию пользователя.

## Маршруты

Метод	URL	Описание
POST	/login	Авторизация пользователя (login)
GET	/users	Получение списка пользователей с фильтрацией и пагинацией
GET	/users/{id}	Получение пользователя по ID
POST	/users	Создание нового пользователя
PUT	/users/{id}	Обновление данных пользователя
DELETE	/users/{id}	Удаление пользователя