Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 ИУ6-32Б**

О Т Ч Е Т

по лабораторной работе № 10

Название: Аутентификация пользователей с помощью jwt-токена

Дисциплина: Языки интернет-программирования

Студент ИУ6-32Б Кондратов С.Ю.

(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель Шульман В.Д.

(Подпись, дата) (И.О. Фамилия)

Цель работы: получение первичных знаний в области авторизации и аутентификации в контексте веб-приложений

Задание

Переделать сервисы с добавлением jwt токена

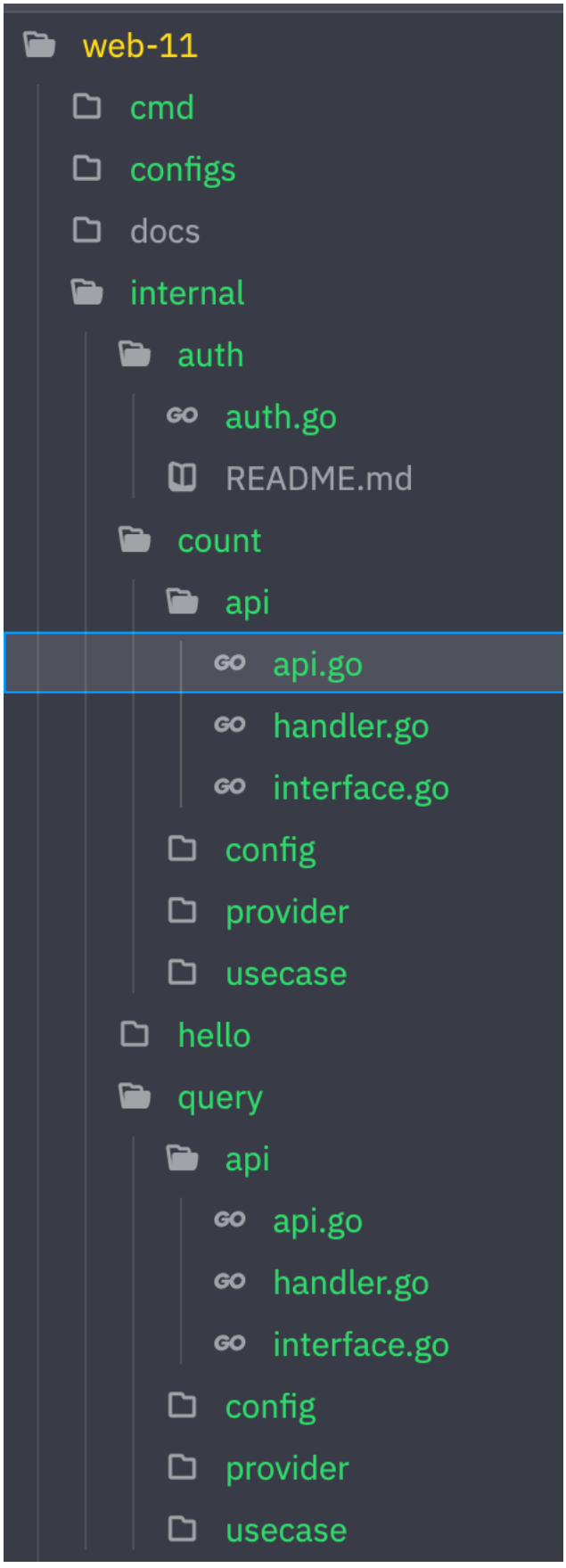
web-11
cmd
configs
docs
internal
auth
auth.go
README.md
count
api
api.go
handler.go
interface.go
config
provider
usecase
hello
query
api
api.go
handler.go
interface.go
config
provider
usecase

Рисунок 1

На рисунке 2 показан пример get запроса для сервиса count.
Для дальнейшей проверки отправим POST запрос с телом JSON

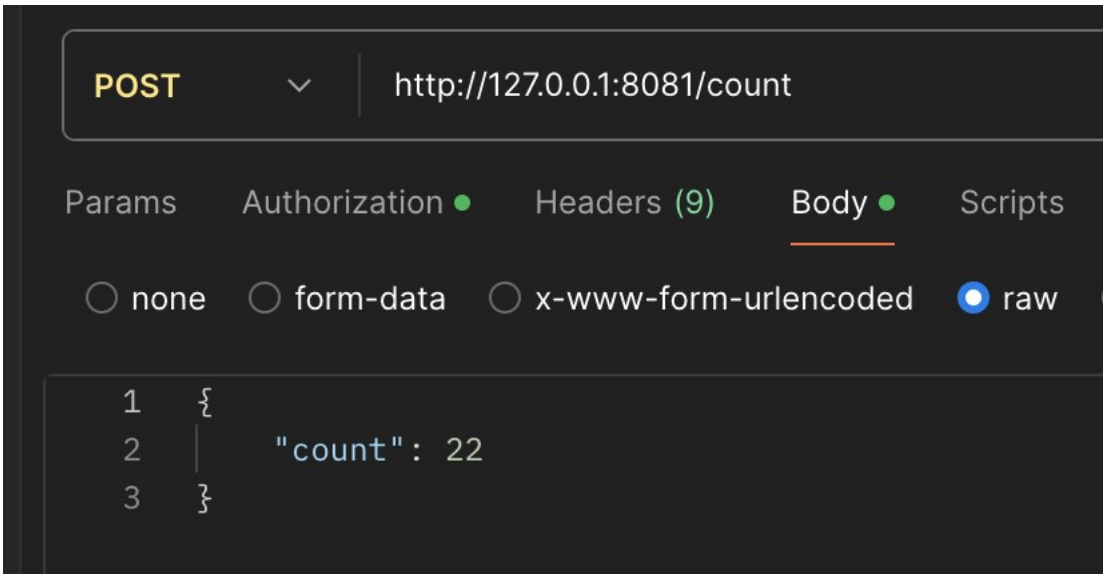

Рисунок 3

	number integer
1	222

Рисунок 4

На рисунке 4 показано то, как это хранится в базе данных

Сервис Hello

Данный сервис должен возвращать Hello, something! На get request.

	message character varying (255)
1	Hello Stepan
2	Hello Emin
3	Hello Mike

Рисунок 4

Здесь показано как возможные сообщения хранятся в бд.
Также я добавил возможность добавления собственных сообщений через POST запрос с передачей JSON объекта.

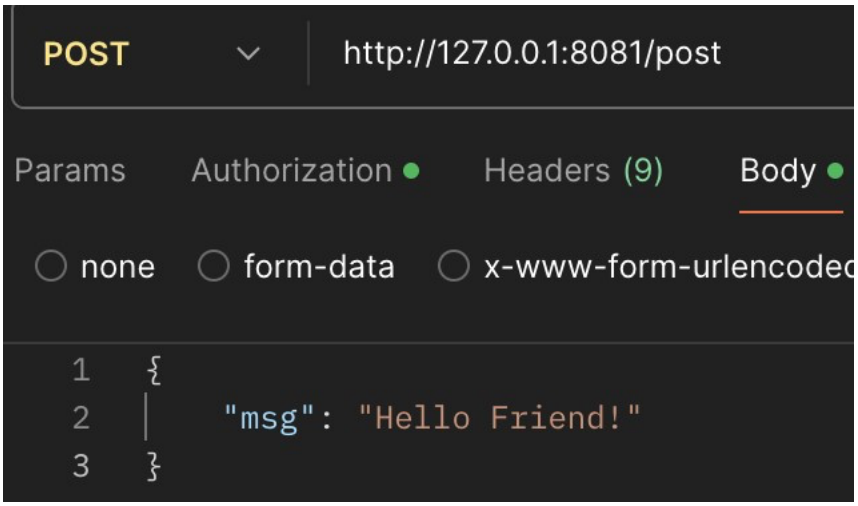

Рисунок 5(пример POST запроса)

На get запрос сервер отправляет случайное приветствующее сообщение

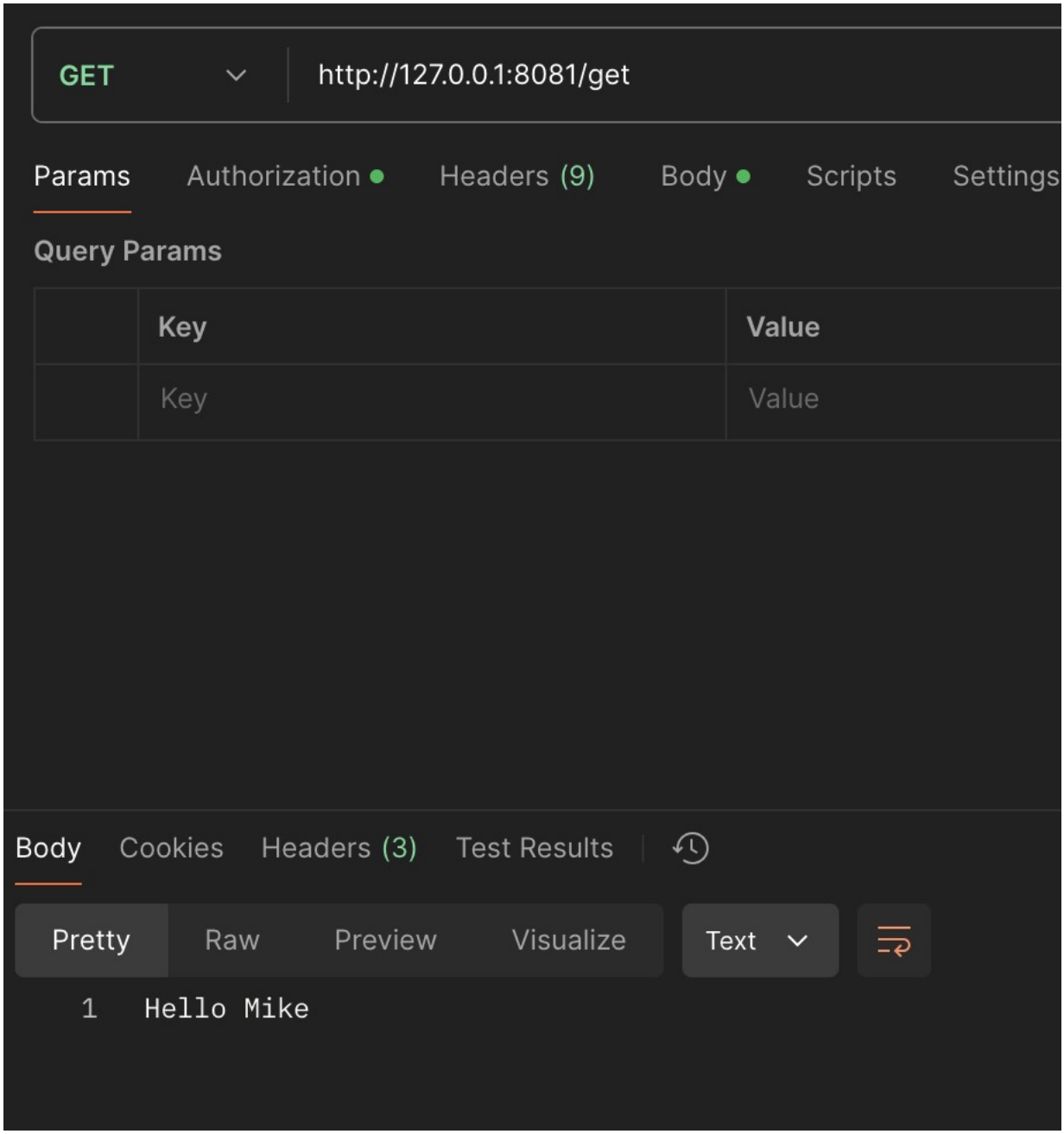

Рисунок 6

Сервис Query

В данном сервисе мы должны возвращать Hello <username> на get запросы пользователя если такой уже существует в бд.

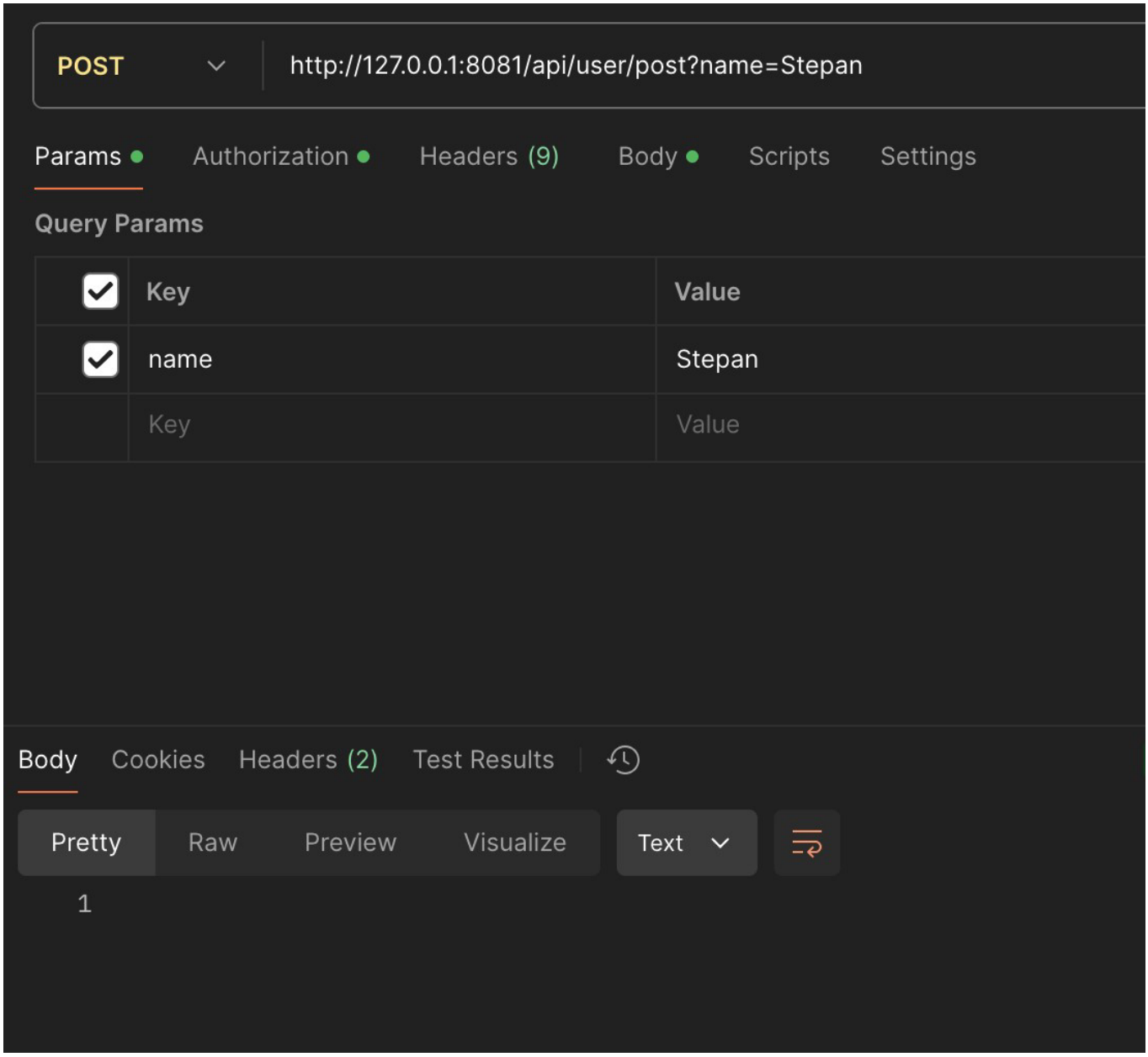

Рисунок 7 (Пример добавления имени в бд)

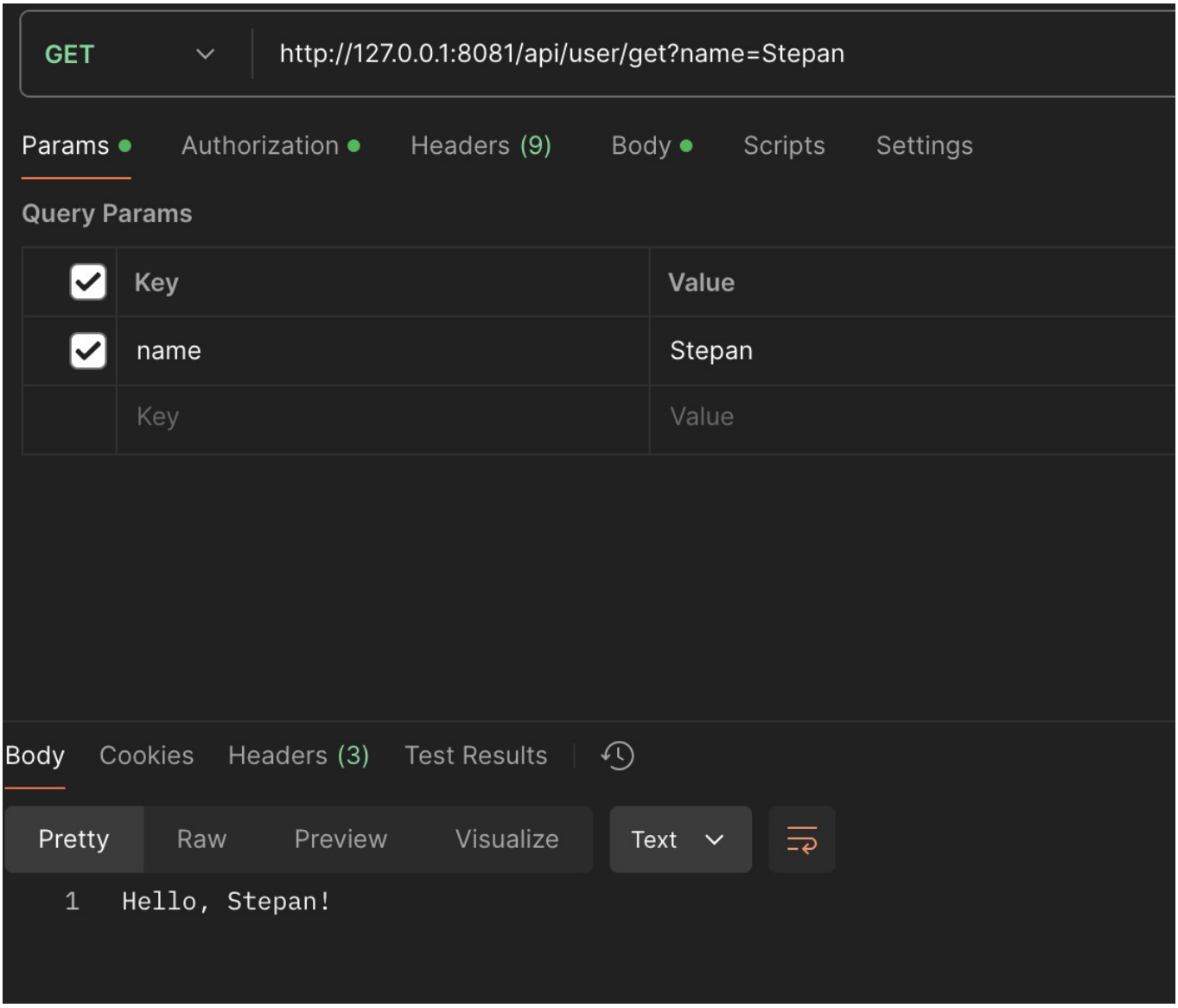

Рисунок 8 (пример запроса GET с тем же параметром имени)

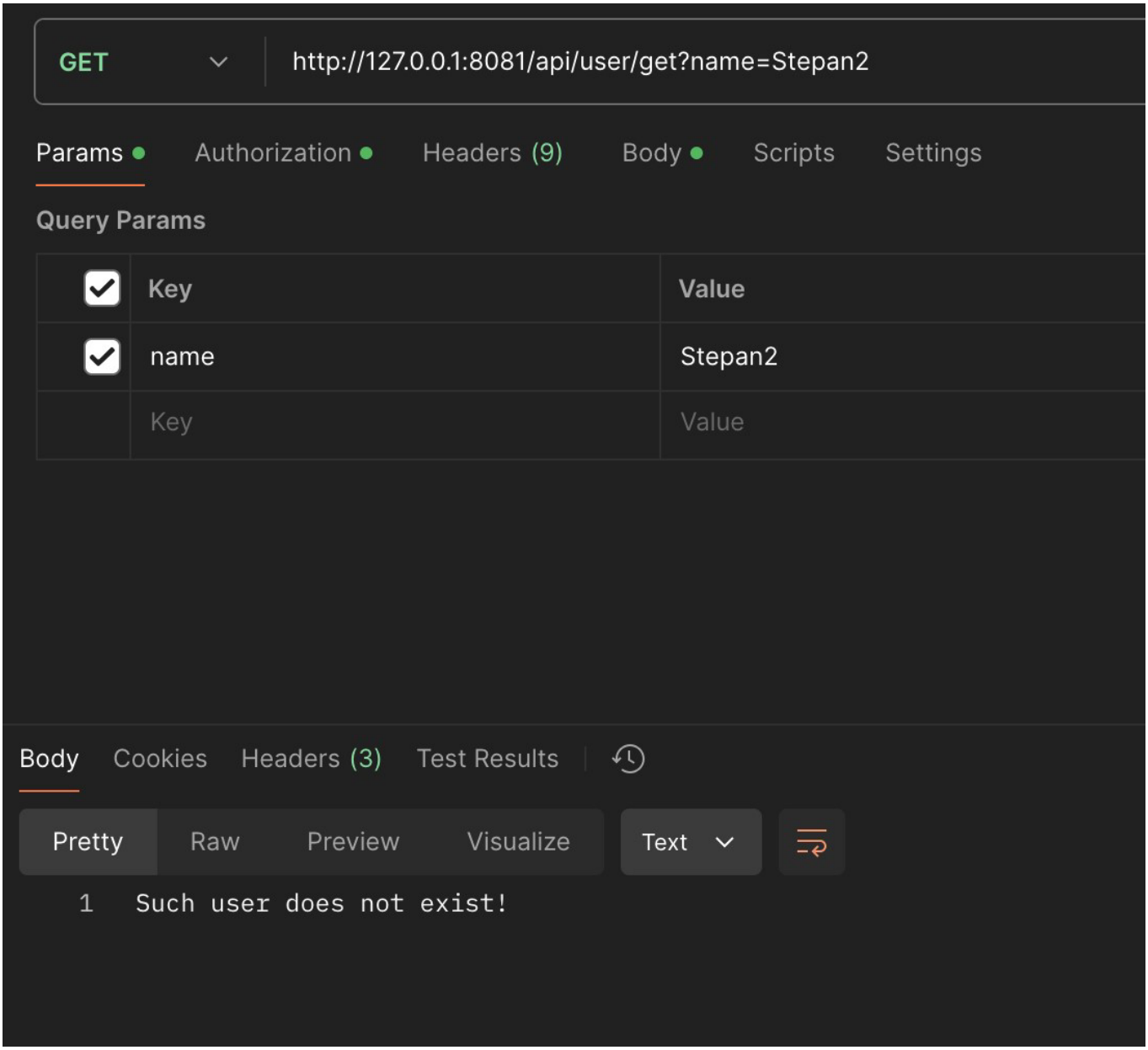

Рисунок 9(В случае отсутствия такого имени в бд, будет выдана ошибка)

JWT Authentication

Для реализации воспользуемся данным нам шаблоном

```
    // Middleware
    e.Use(middleware.Logger())
    e.Use(middleware.Recover())

    // Login route
    e.POST("/login", login)

    // Unauthenticated route
    e.GET("/", accessible)

    // Restricted group
    r := e.Group("/restricted")

    // Configure middleware with the custom claims type
    config := echojwt.Config{
        NewClaimsFunc: func(c echo.Context) jwt.Claims {
            return new(jwtCustomClaims)
        },
        SigningKey: []byte("secret"),
    }
    r.Use(echojwt.WithConfig(config))
    r.GET("", restricted)

    e.Logger.Fatal(e.Start(":1323"))
```

Рисунок 10

Единственное что изменится в проекте по сравнению с 10 лабой это добавление так называемой restricted группы ручек

```
35      api.server.Use(middleware.Logger())
36      api.server.Use(middleware.Recover())
37
38      api.server.POST("/login", api.Login)
39
40      config := echojwt.Config{
41          NewClaimsFunc: func(c echo.Context) jwt.Claims {
42              return new(jwtCustomClaims)
43          },
44          SigningKey: []byte("secret"),
45      }
46      r := api.server.Group("/restricted")
47
48      r.Use(echojwt.WithConfig(config))
49
50      r.GET("/count", api.GetCount)
51      r.POST("/count", api.PostCount)
52
53      api.address = fmt.Sprintf("%s:%d", ip, port)
54
55      return &api
56  }
```

Рисунок 11

Теперь при попытке получить response от сервисов не получив необходимый токен будет выдаваться ошибка

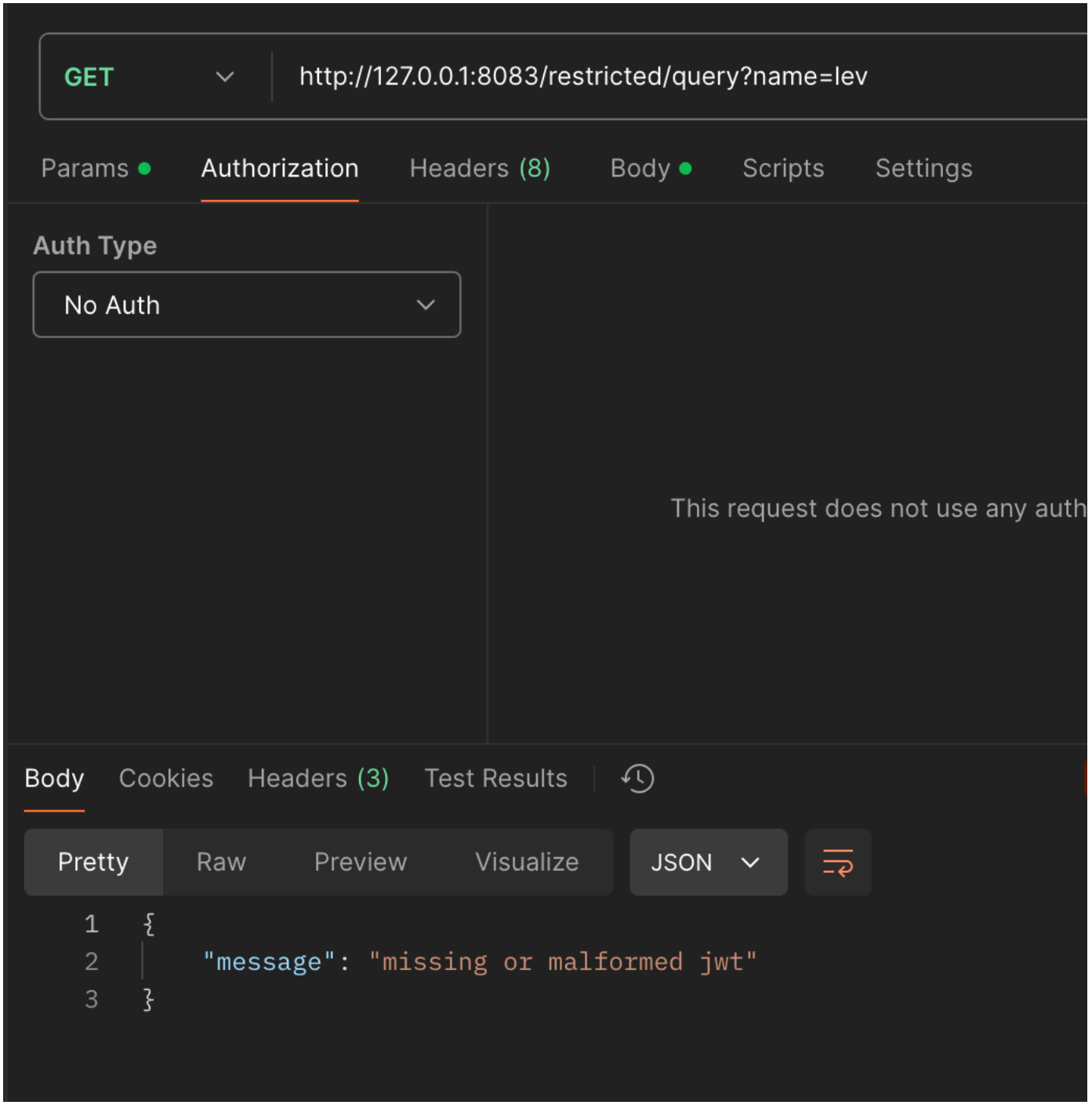

Рисунок 12

Для получения токена мы должны отправить запрос на /login с указанием username и password в форме

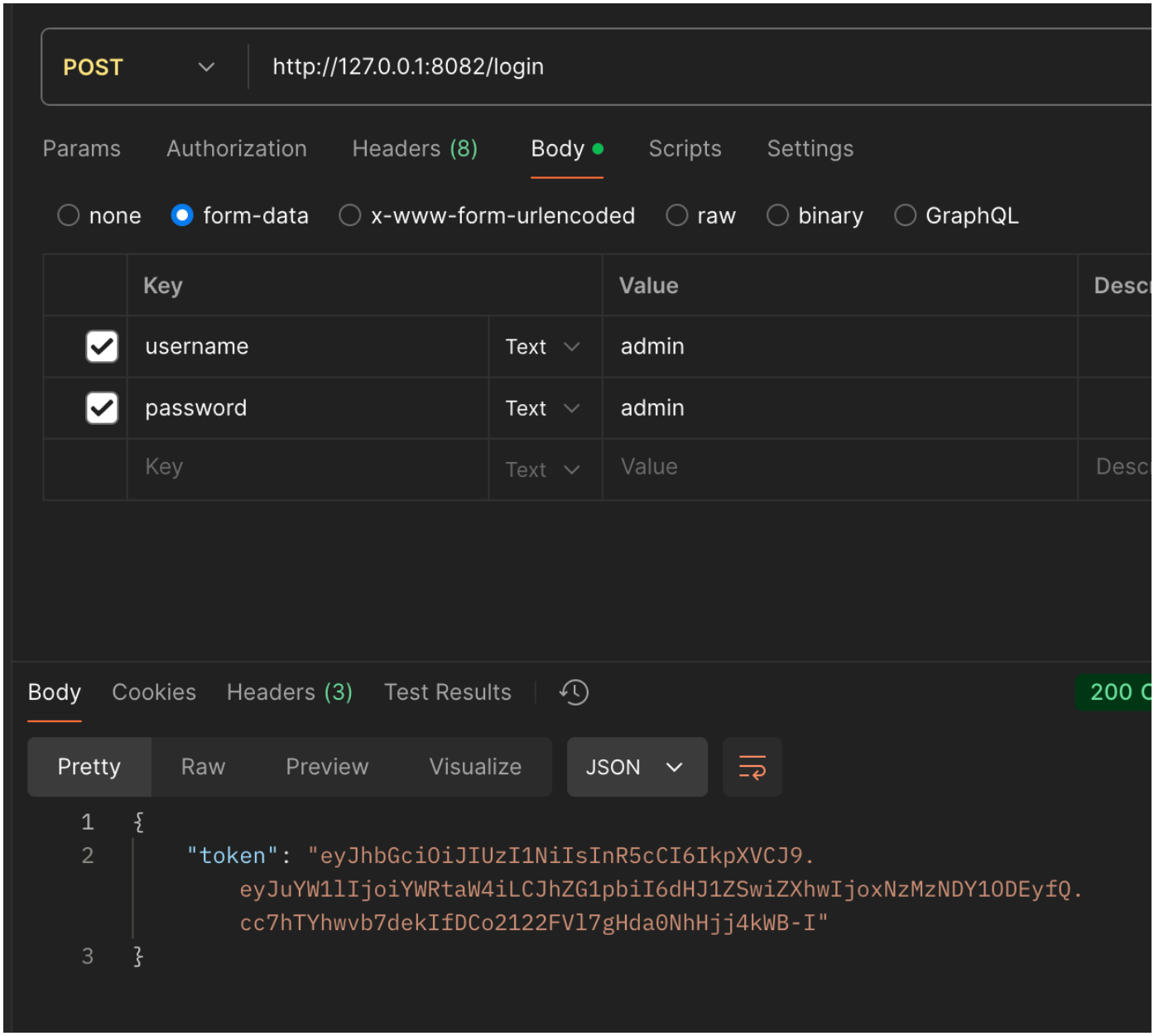

Рисунок 13

Тогда последующие запросы будем выполнять указывая токен аутентификации

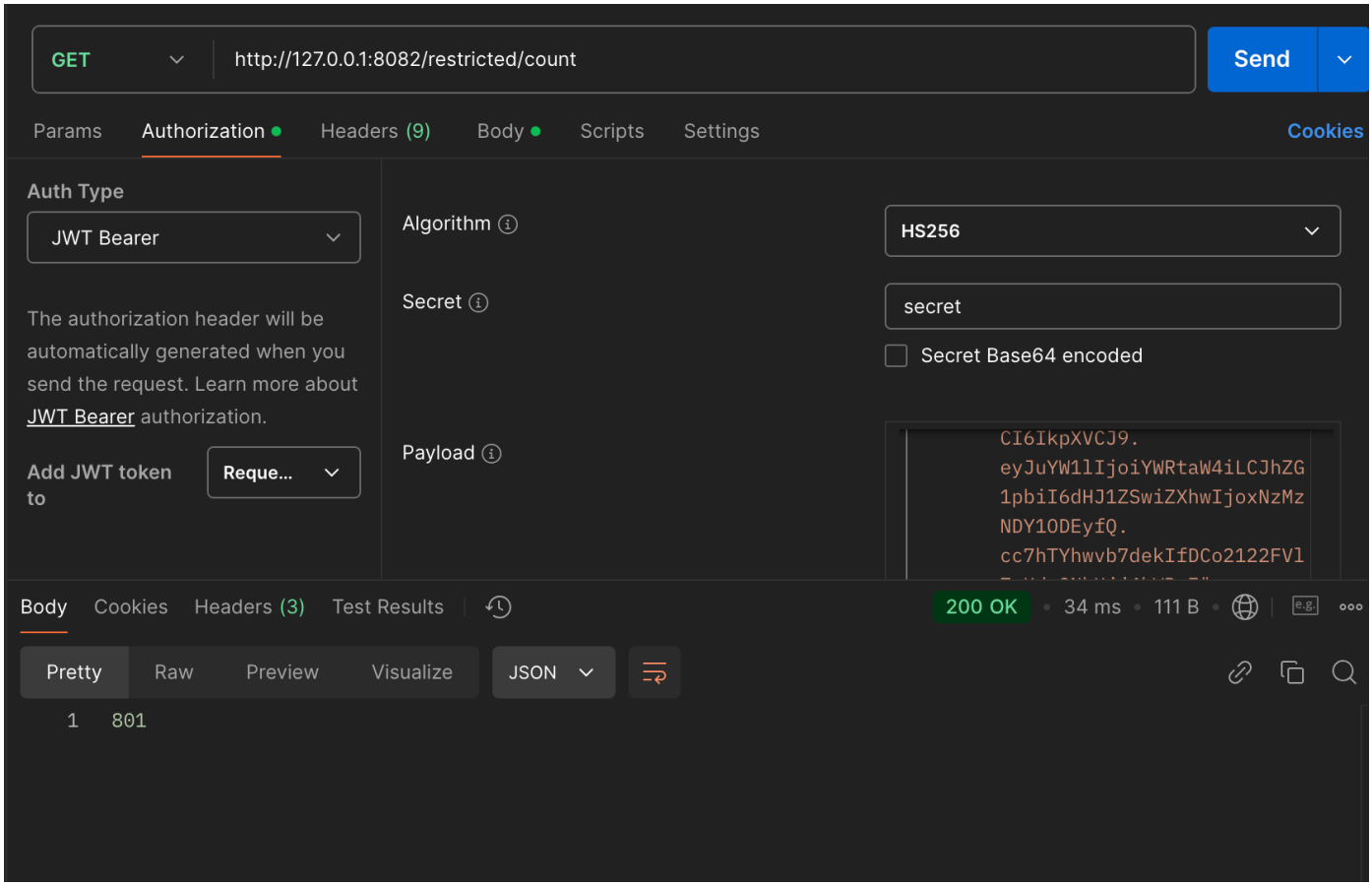

Рисунок 14

Заключение: Я научился создавать веб-сервер и использовать JWT токен для аутентификации.

Москва, 2024