**Министерство цифрового развития, связи и массовых коммуникаций**

**Российской Федерации**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Лабораторная работа №6**

по дисциплине

**«**Функциональное программирование**»**

Выполнил:

Студент группы БВТ1901

Смирнов. Д.А

Проверила:

Мосева М.С.

Москва 2021

**Цель работы:** Запустить и протестировать HTTP-приложение Akka, получить предварительный обзор того, как маршруты упрощают обмен данными по HTTP.

**Ход работы:**

Создал и запустил Akka HTTP.

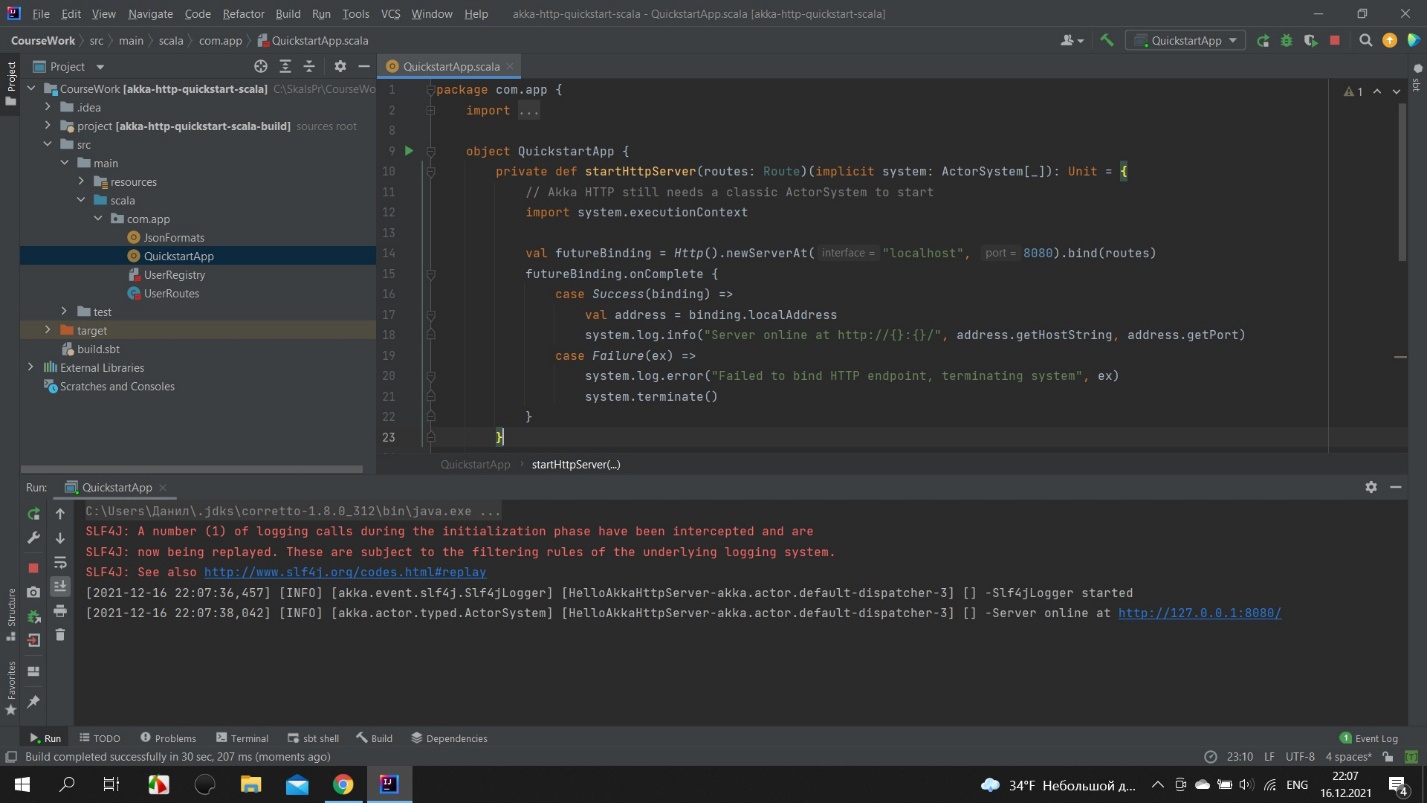


Рисунок 1 – Запуск приложения.

Тестировать с помощью RESTClient

Создание пользователей

Установил для заголовка значение Content-Type: application/json.

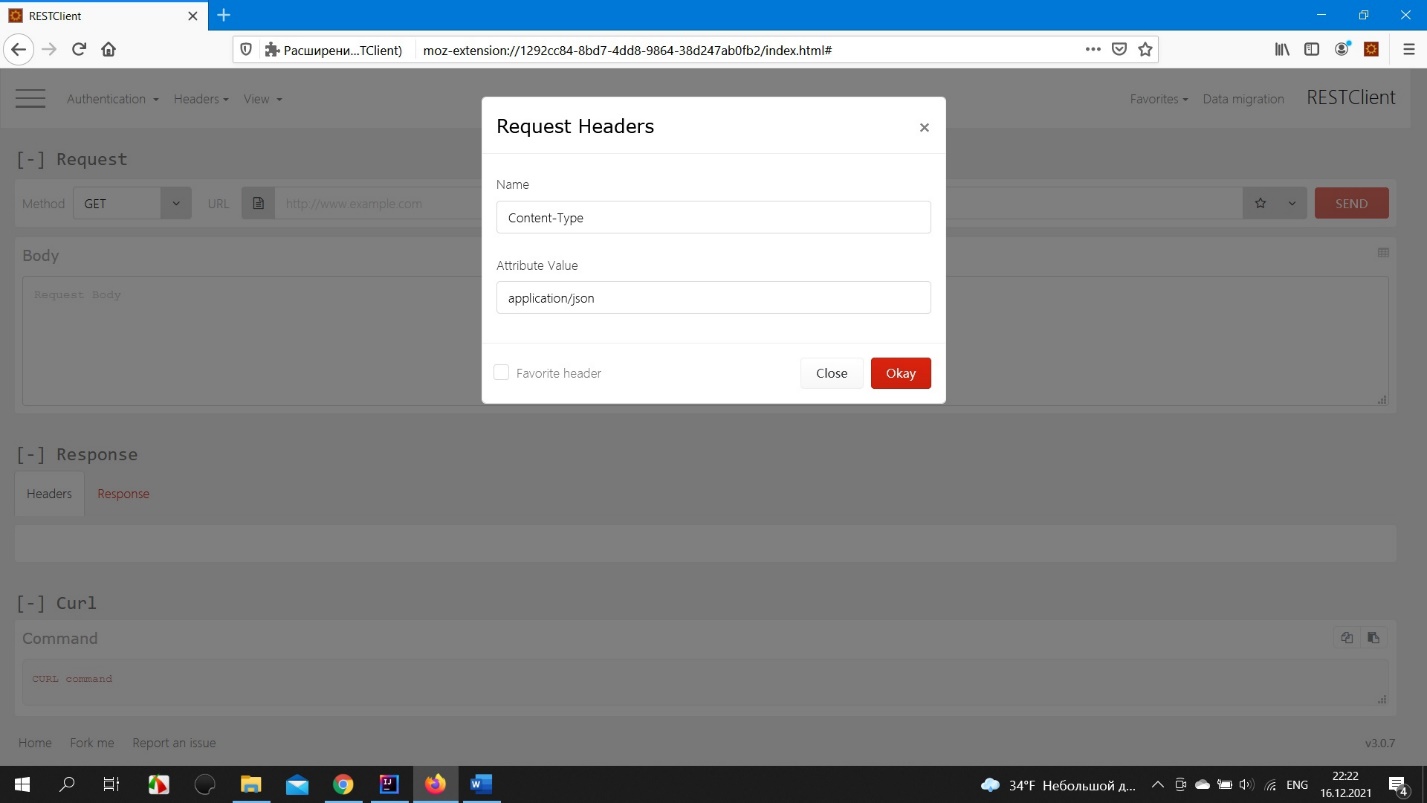


Рисунок 2 – Установка заголовка

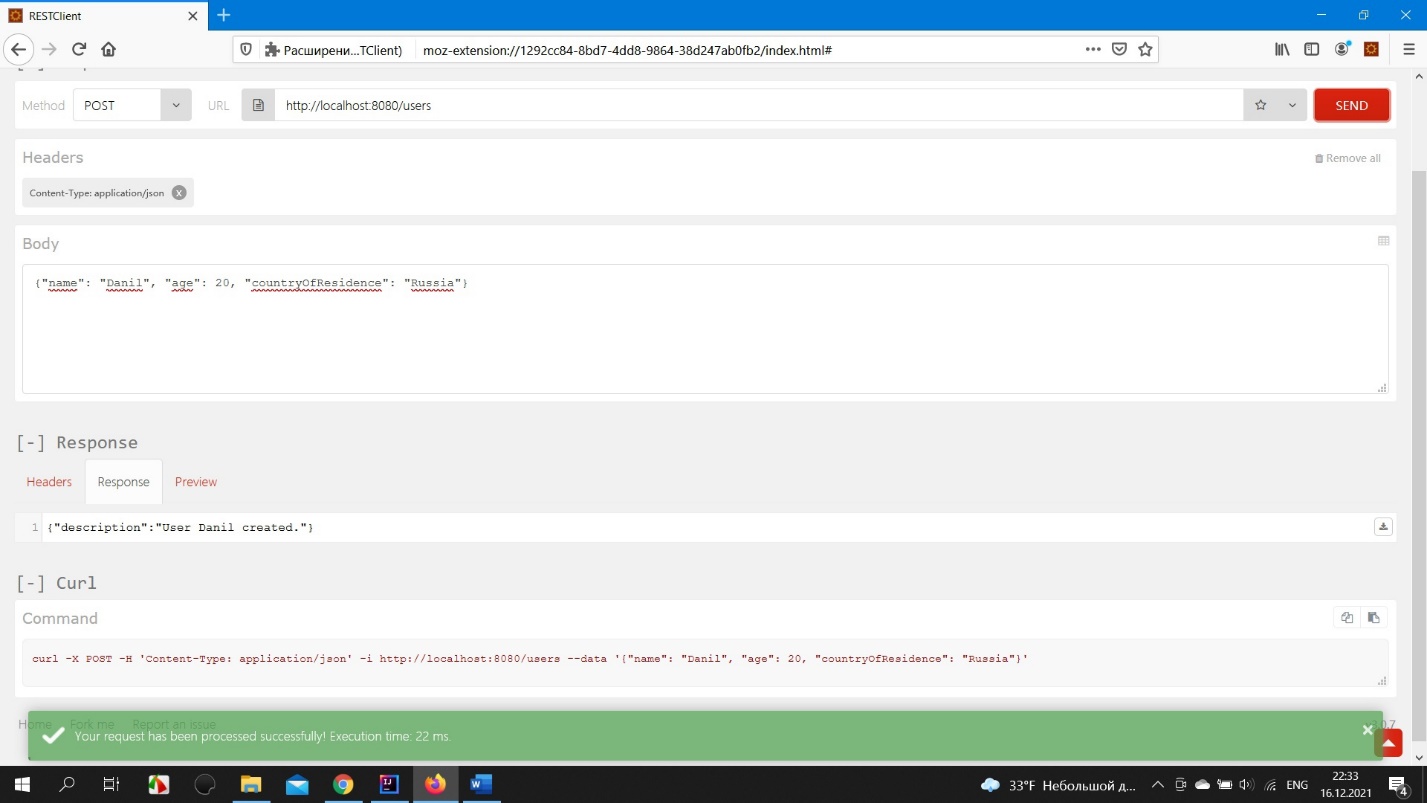
Выбрал метод POST и ввел url. Ввел команды для создания пользователей. 

Рисунок 3 – Создание первого пользователя

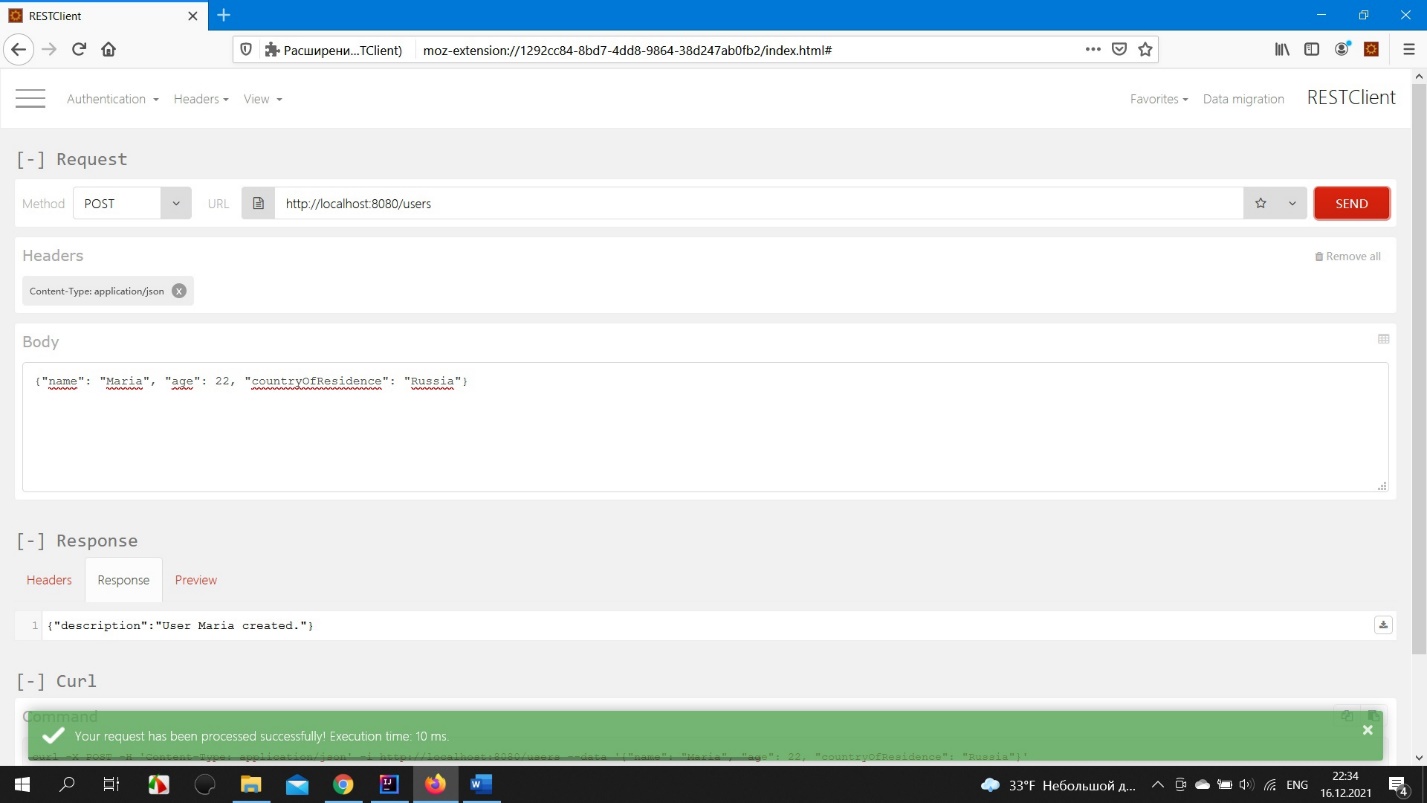


Рисунок 4 – Создание второго пользователя

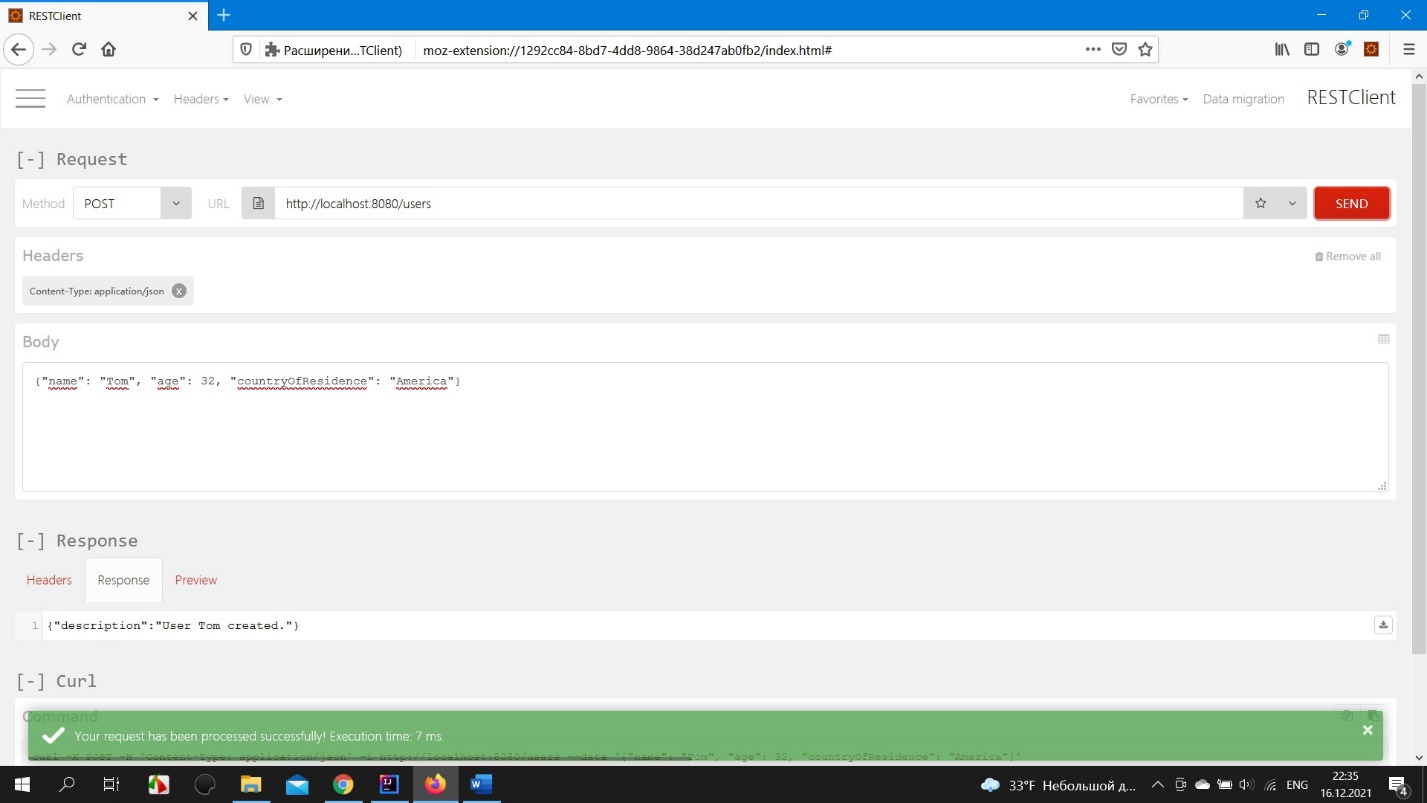


Рисунок 5 – Создание третьего пользователя

После чего получил всех пользователей используя метод GET

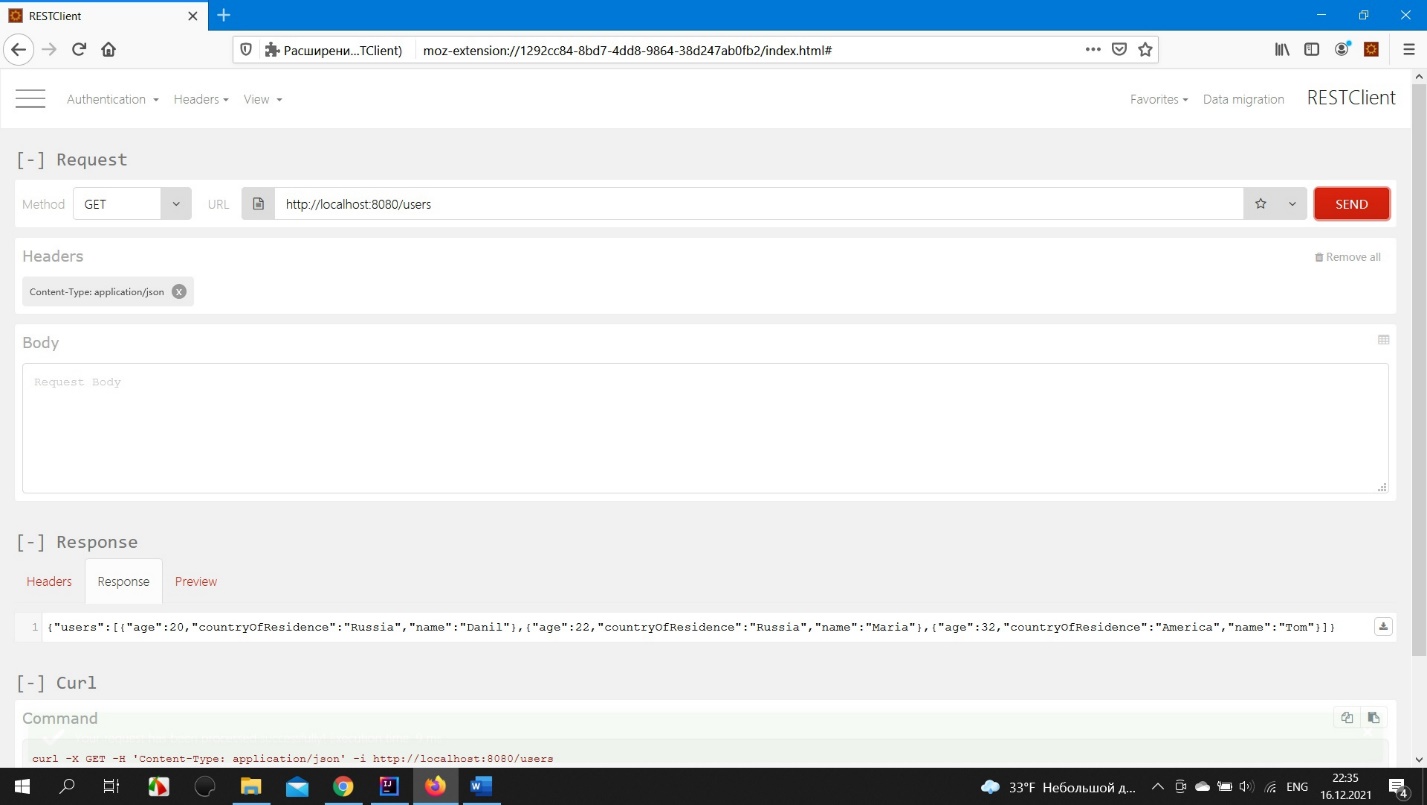


Рисунок 6 – Получение всех пользователей

Далее перезапустил приложение и, попытавшись получить список пользователей, получил пустой список, а попытавшись получить определенного пользователя систем ответила, что он не найден.

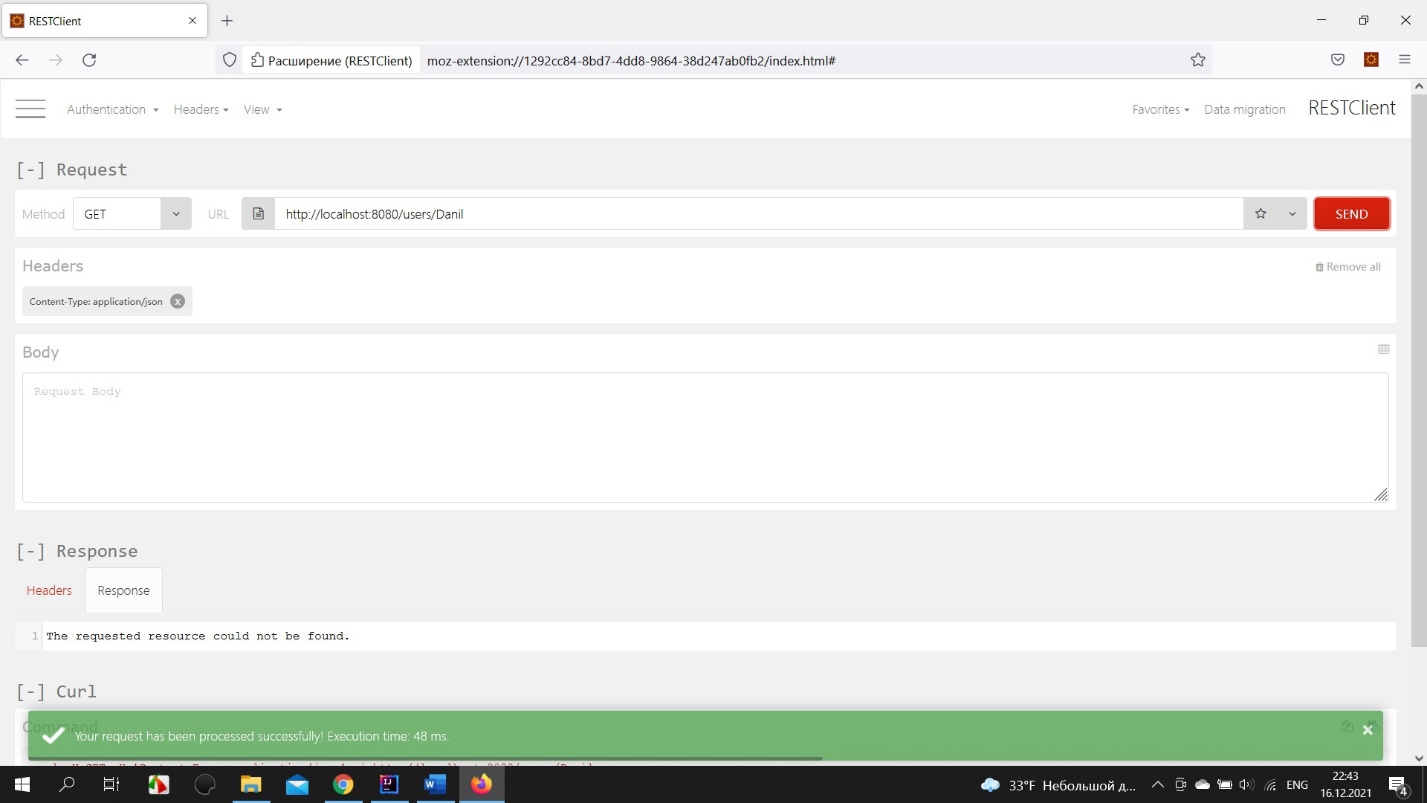


Рисунок 7 – Получение определенного пользователя

Был добавлен новый пользователь и удален с помощью метода DELETE.

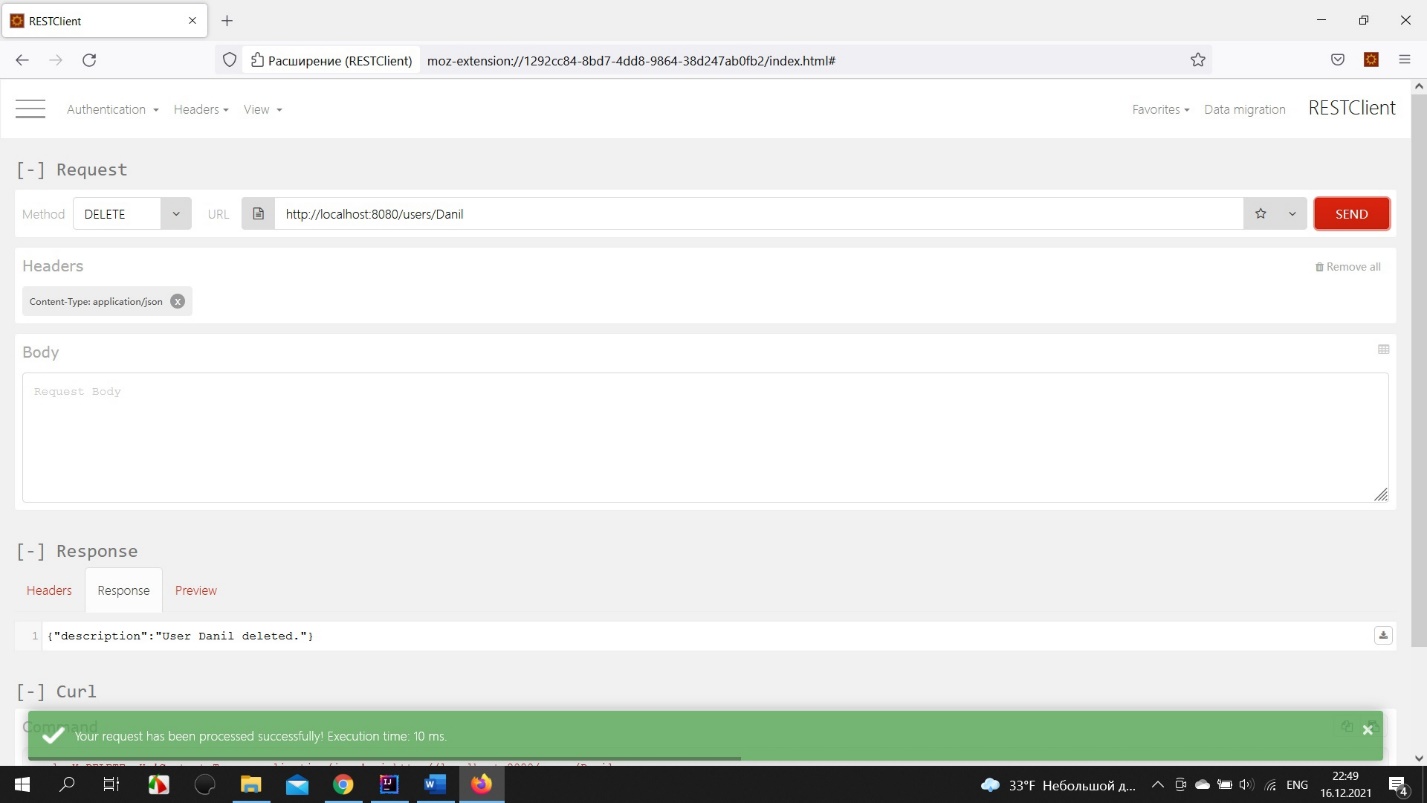


Рисунок 8 – Удаление пользователя

Код модульного теста:

package com.app {  
  
 import akka.actor.testkit.typed.scaladsl.ActorTestKit  
 import akka.http.scaladsl.marshalling.Marshal  
 import akka.http.scaladsl.model.\_  
 import akka.http.scaladsl.testkit.ScalatestRouteTest  
 import org.scalatest.concurrent.ScalaFutures  
 import org.scalatest.matchers.should.Matchers  
 import org.scalatest.wordspec.AnyWordSpec  
  
 class UserRoutesSpec extends AnyWordSpec with Matchers with ScalaFutures with ScalatestRouteTest {  
 lazy val *testKit* = *ActorTestKit*()  
 implicit def typedSystem = *testKit*.system  
 override def createActorSystem(): akka.actor.ActorSystem = *testKit*.system.classicSystem  
  
 val *userRegistry* = *testKit*.spawn(*UserRegistry*())  
 lazy val *routes* = new UserRoutes(*userRegistry*).*userRoutes* import akka.http.scaladsl.marshallers.sprayjson.SprayJsonSupport.\_  
 import JsonFormats.\_  
  
 "UserRoutes" should {  
 "return no users if no present (GET /users)" in {  
 val request = *HttpRequest*(uri = "http://localhost:1234/users")  
 request ~> *routes* ~> check {  
 status should ===(StatusCodes.*OK*)  
 contentType should ===(ContentTypes.*`application/json`*)  
 entityAs[String] should ===("""{"users":[]}""")  
 }  
 }  
  
 "be able to add users (POST /users)" in {  
 val user = *User*("Kapi"**,** 42**,** "jp")  
 val userEntity = *Marshal*(user).to[MessageEntity].futureValue  
 val request = *Post*("http://localhost:8081/users").withEntity(userEntity)  
 request ~> *routes* ~> check {  
 status should ===(StatusCodes.*Created*)  
 contentType should ===(ContentTypes.*`application/json`*)  
 entityAs[String] should ===("""{"description":"User Kapi created."}""")  
 }  
 }  
  
 "be able to remove users (DELETE /users)" in {  
 val request = *Delete*(uri = "http://localhost:8081/users/Kapi")  
 request ~> *routes* ~> check {  
 status should ===(StatusCodes.*OK*)  
 contentType should ===(ContentTypes.*`application/json`*)  
 entityAs[String] should ===("""{"description":"User Kapi deleted."}""")  
 }  
 }  
 }  
 }  
}

Результаты работы модульного теста:

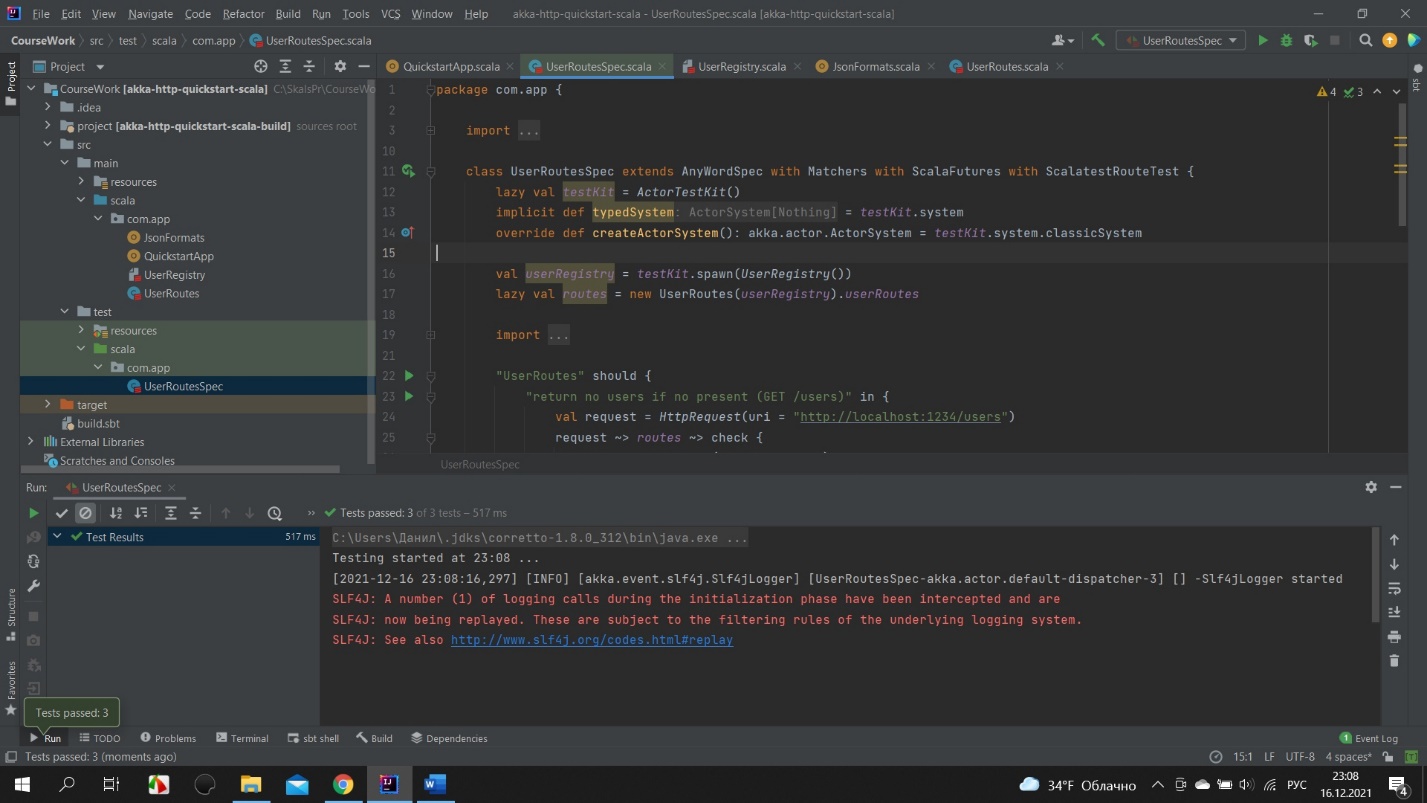


Рисунок 9 – Модульный тест

**Вывод:** был выполнилнен запуск и протестировано HTTP-приложение Akka, получен предварительный обзор того, как маршруты упрощают обмен данными по HTTP.