**Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации  
Ордена Трудового Красного Знамени  
Федеральное государственное бюджетное образовательное учреждение высшего образования**

Кафедра «Математической кибернетики и информационных технологий»

Отчет по курсовой работе

по дисциплине «Введение в информационные технологии»

на тему:

«Akka HTTP REST сервис на языке программирования Scala»

Выполнила: студентка БВТ1903

Звягина Ольга Владимировна

Проверила:

Мосева Марина Сергеевна

Москва

2021

# Оглавление

[Оглавление](#_heading=h.jjjooarprxri) 2

[Цель работы](#_heading=h.t8oo1xke64im) 3

[Задачи](#_heading=h.rhvgnsa2m0rw) 3

[Ход работы](#_heading=h.xka9ae76w58) 4

[Вывод](#_heading=h.fe9epyhk4mb6) 11

# Цель работы

Запустить и тестировать HTTP-приложение Akka, получить предварительный обзор того, как маршруты упрощают обмен данными по HTTP.

# Задачи

* Изучить литературу по основам работы Akka;
* развернуть на локальном устройстве HTTP-приложение Akka;
* протестировать систему посредством совершения к ней запросов по протоколу HTTP;
* тестирование маршрутов;
* составить отчет о проделанной работе, прикрепив необходимые снимки экрана и листинг кода.

# 

# Ход работы

Развертывание Akka HTTP сервиса на локальной машине.

Файл *JsonFormats.scala* преобразует данные JSON из запросов в типы Scala и наоборот.

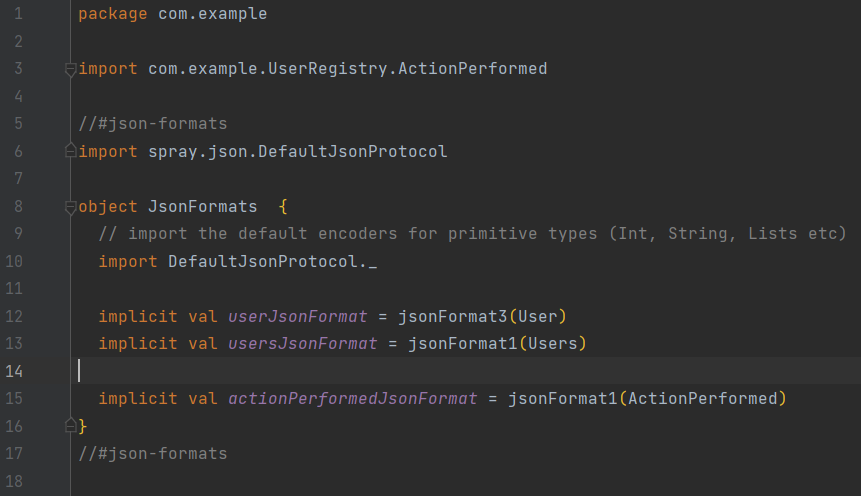


Рисунок 2 – Содержимое файла *JsonFormats.scala*

Файл *QuickstartApp.scala* содержит основной метод начальной загрузки приложения.

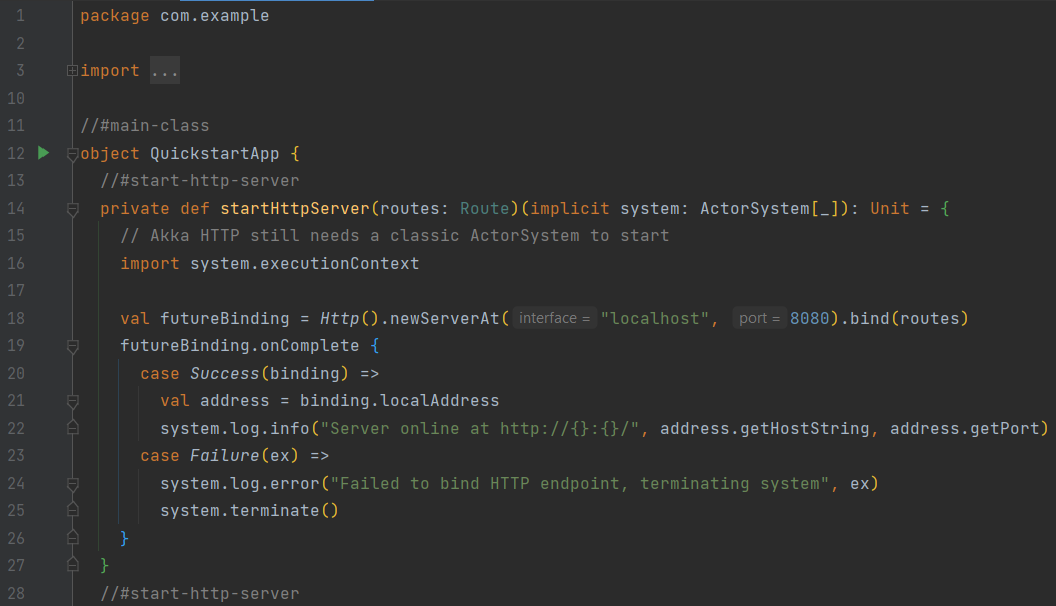


Рисунок 2 – Содержимое файла *QuickstartApp.scala*



Рисунок 3 – Содержимое файла *QuickstartApp.scala*

Файл *UserRegistry.scala* – актор, обрабатывающий запросы на регистрацию.

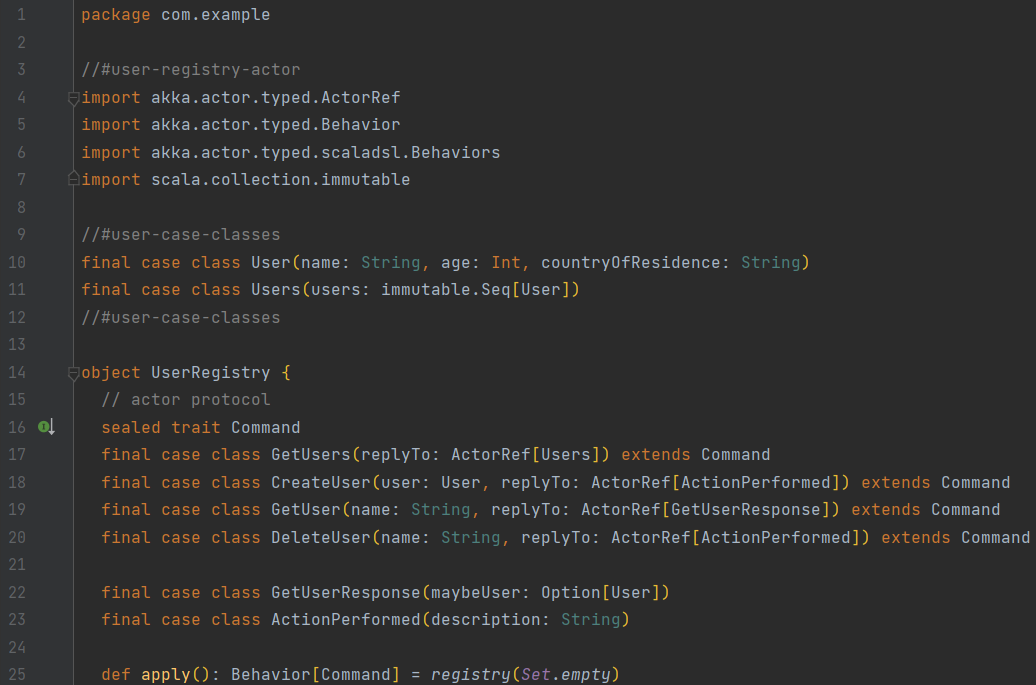


Рисунок 4 – Содержимое файла *UserRegistry.scala*

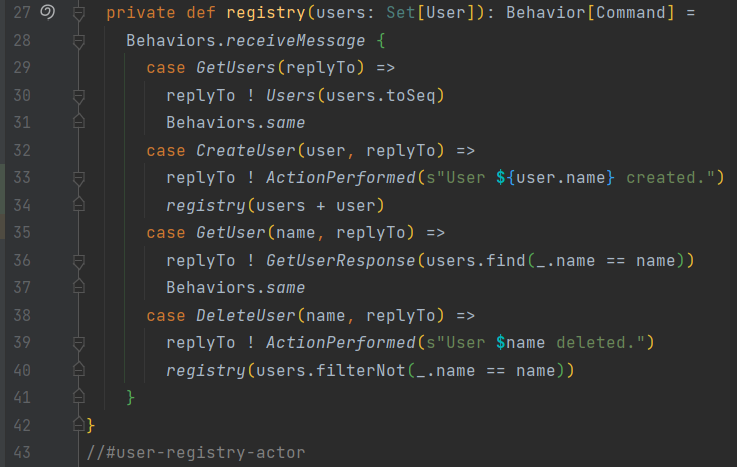


Рисунок 5 – Содержимое файла *UserRegistry.scala*

Файл *UserRoutes.scala* отвечает за HTTP-маршруты Akka, определяющие открытые эндпоинты.

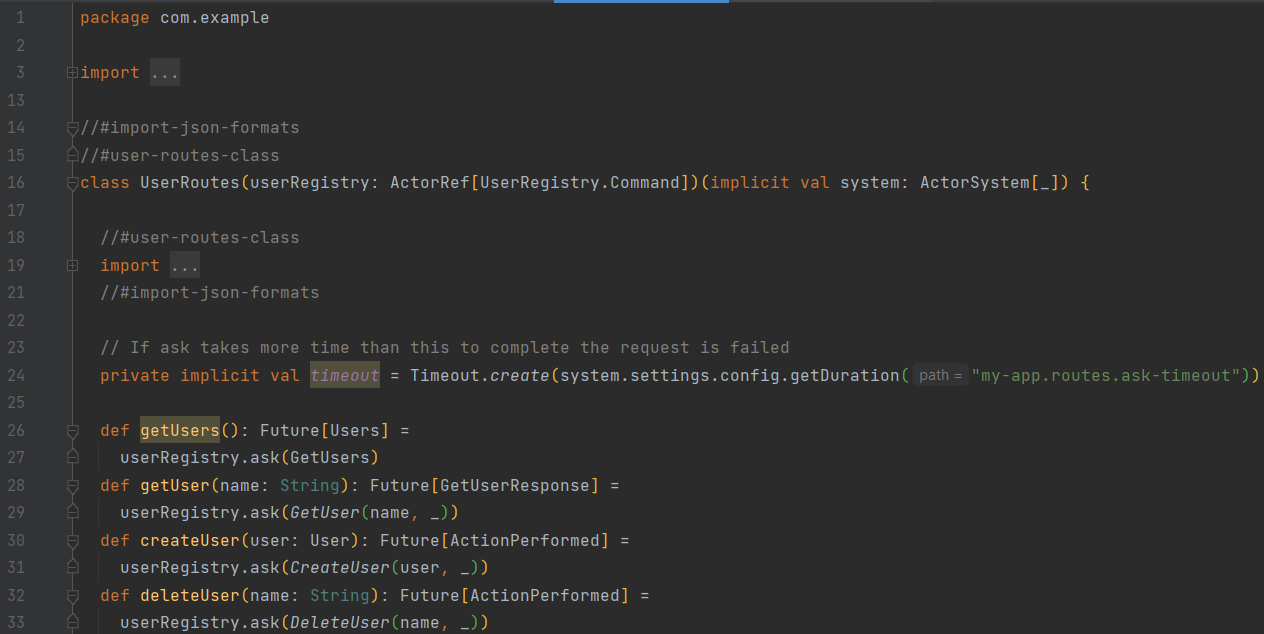


Рисунок 6 – Содержимое файла *UserRoutes.scala*



Рисунок 7 – Содержимое файла *UserRoutes.scala*



Рисунок 8 – Содержимое файла *UserRoutes.scala*

Запуск и тестирование приложения.

Для запуска просто добавляем Run/Debug конфигурацию и указываем как точку входа com.example.QuickstartApp.

Запускаем проект и ждем появления в открывшейся консоли сообщения “...Server online at ...”.

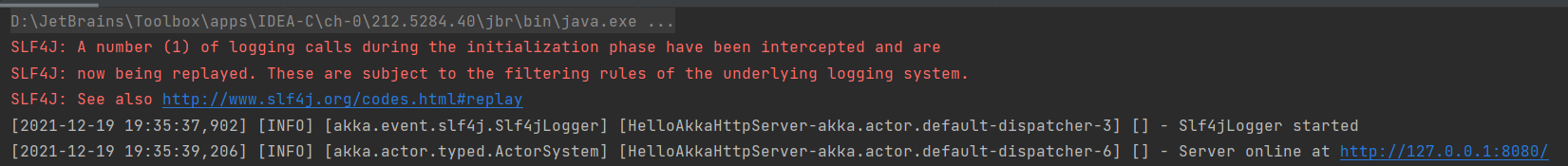


Рисунок 8 – Консоль с сообщением

Теперь протестируем HTTP приложение с помощью инструмента Postman.

Отправим POST запрос приложению поместив в тело запроса в формате JSON параметры пользователя, которого мы хотим добавить.

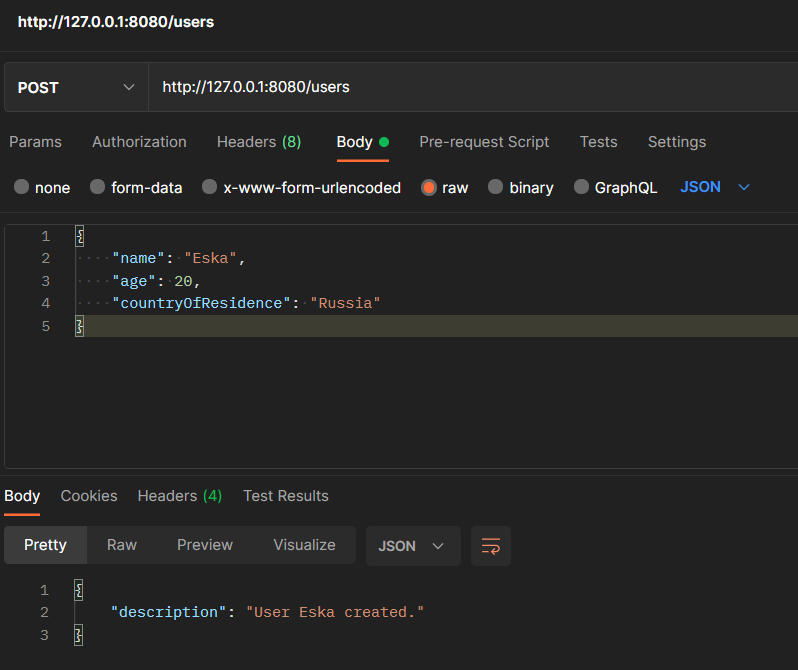


Рисунок 9 – Добавление пользователя

Также можем получить список всех «Зарегистрированных» пользователей, для этого обратимся к серверу через GET запрос.

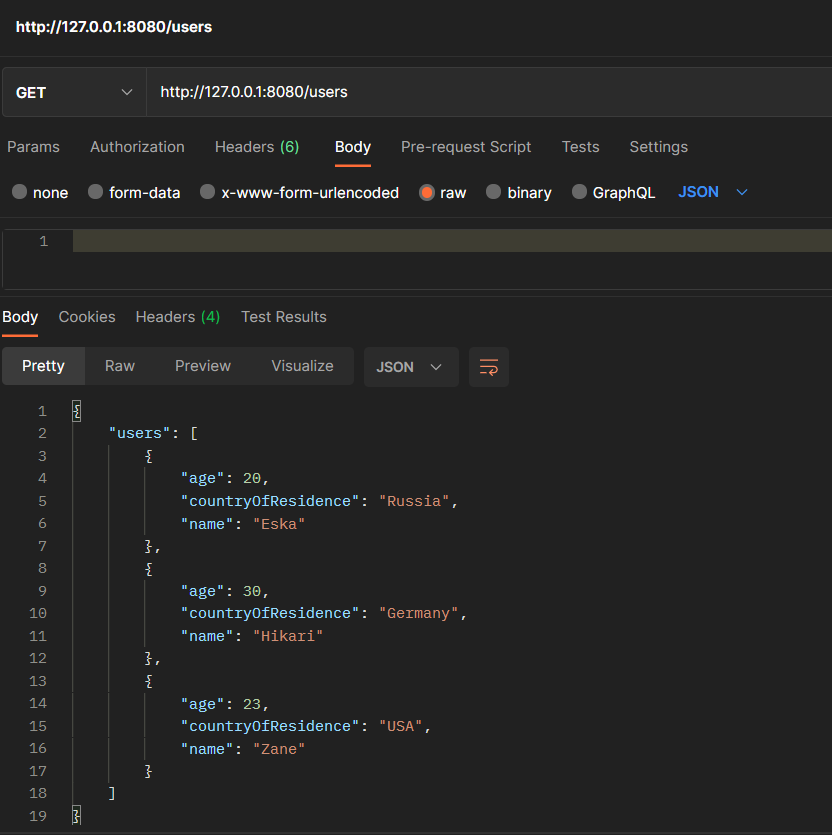


Рисунок 10 – Просмотр списка зарегистрированных пользователей

Помимо добавления новых пользователей и просмотра данных по уже созданным можно удалять пользователей, обратившись к серверу с DELETE запросом и указанием имени пользователя.

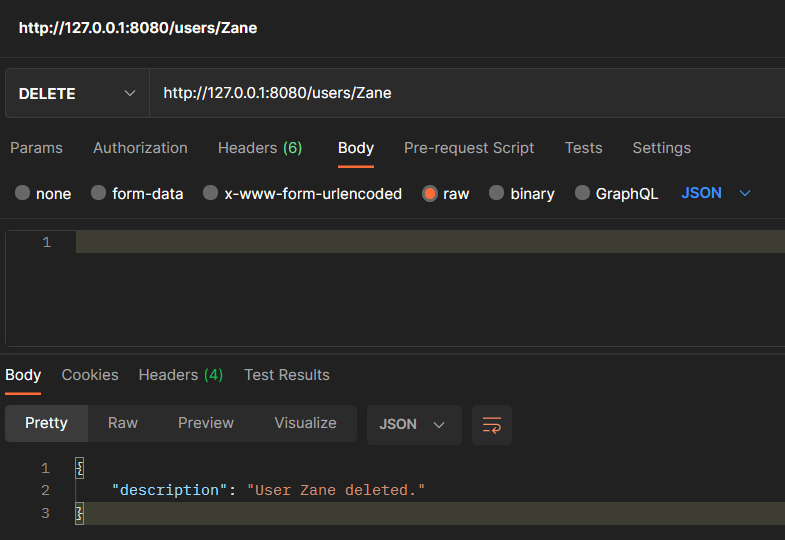


Рисунок 10 – Удаление пользователя

Маршруты и их тестирование.

Все маршруты выделены в отдельный класс. Это сделано для того, чтобы отделить код инфраструктуры от маршрутов, которые должны декларировать только то, с чем им нужно работать, и, следовательно, могут быть немного больше сосредоточены на своей задаче. Это улучшает тестируемость, позволяя производить ее модульно.

Ниже приведен код *unit теста*.

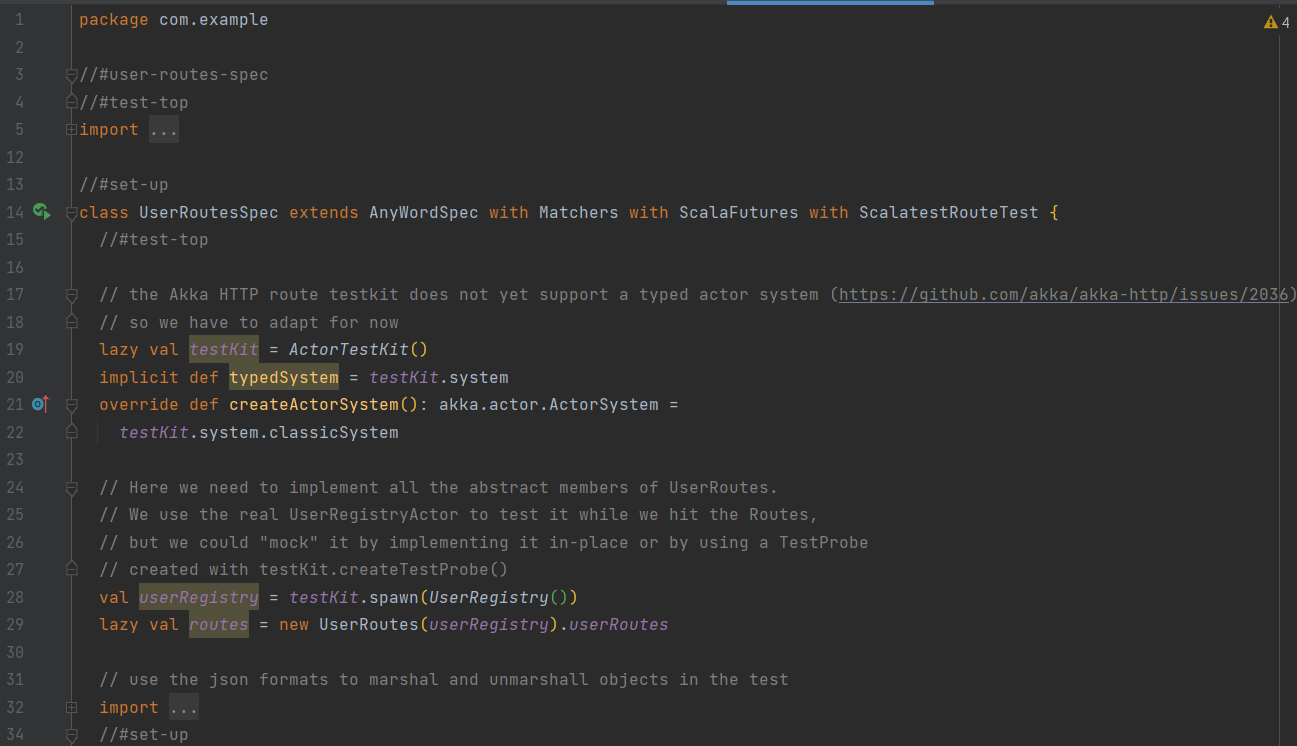


Рисунок 11 – Тестирование

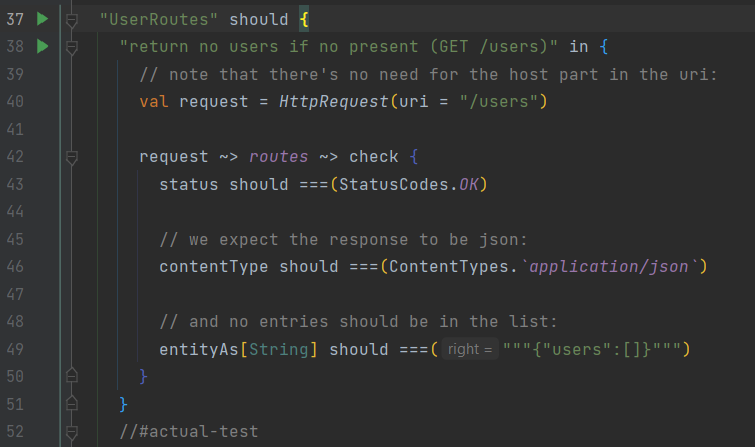


Рисунок 12 – Тестирование. Просмотр пользователей

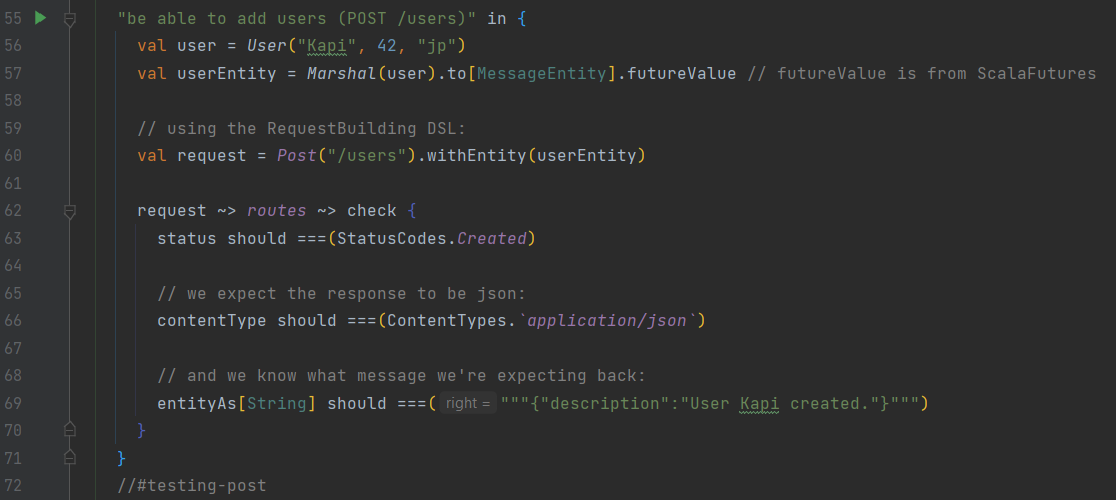


Рисунок 13 – Тестирование. Добавление пользователя

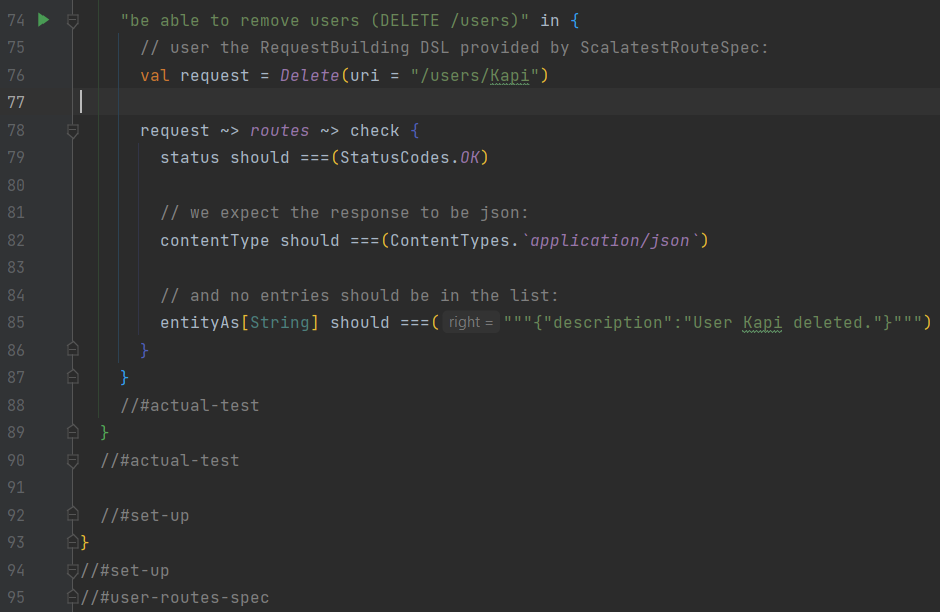


Рисунок 14 – Тестирование. Удаление пользователя

# 

# Вывод

В ходе выполнения данной курсовой работы я получила базовые навыки работы с Akka.

Я запустила и тестировала HTTP-приложение Akka, получила предварительный обзор того, как маршруты упрощают обмен данными по HTTP.