УДК 004

Автозаполнение пользовательского интерфейса JQuery с загрузкой Spring

Семченко Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается возможность создания функции автозаполнения с помощью пользовательского интерфейса JQuery. Практическим результатом является web-приложение Spring.
Ключевые слова: Spring Boot, Java, Thymeleaf

JQuery UI autocomplete with Spring Boot

Semchenko Regina Viktorovna
Sholom-Aleichem Priamursky State University
Student

Erovlev Pavel Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article explores the ability to create an autocomplete function using the jQuery UI. The bottom line is a Spring web application.
Keywords: Spring Boot, Java, Thymeleaf

JQuery UI - одна из популярных библиотек javascript. Можно использовать JQuery UI для многих операций, которые обеспечивают визуальные эффекты, виджеты UI и другое.

Цель данной статьи реализовать функцию автозаполнения с помощью пользовательского интерфейса JQuery.

В своей работе А.Б.Джемалетдинов, А.А.Шевченко рассмотрели вопросы создания тестов для Spring Boot mvc контроллеров [1]. В.И.Зарайский провел обзор на разработку модуля автоматизации работы с конференциями в кафедральном приложении [2]. Р.И.Ибраимов продемонстрировал процесс создания Docker-образа для Spring Boot проекта и развернул его на платформе AWS EC2[3]. Е.О.Кабардинский, А.Г.Ивашко

провели сравнительный анализ сервисных шин предприятия, а так же сравнили некоторые ESB, одна из которых Spring Boot [4]. Так же Р.И.Ибраимов, А.Р.Зайчик, Н.С.Минзатров разработали генеалогическое дерево на языке Java с использованием фреймворка Spring Boot b ,b,kbjntrb gedcom4j[5].

Создадим приложение Spring Boot с необходимыми зависимостями. Для этого необходимы зависимости «spring-boot-starter-web» и «spring-boot-starter-thymeleaf» для поддержки работы web-MVC и использования шаблона «Thymeleaf» соответственно (рис.1).

```
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
        <exclusions>
            <exclusion>
                <groupId>org.junit.vintage</groupId>
                <artifactId>junit-vintage-engine</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
</dependencies>
```

Рисунок 1 – pom.xml

Далее создадим класс контроллера «SpringWebMVC» с именем «AutoCompleteController.java»

Этот контроллер добавляет список жестко закодированных значений «String», который содержит имена в атрибут «Model», и отображает страницу «home.html».

```java
import ...

@Controller
public class AutoCompleteController {

    @GetMapping("/")
    public String home(Model model) {
        List<String> names = new ArrayList<>();
        names.add("Павел");
        names.add("Иван");
        names.add("Дмитрий");
        names.add("Егор");
        names.add("Андрей");
        names.add("Олег");
        names.add("Александр");
        names.add("Денис");
        names.add("Ибрагим");
        names.add("Петр");
        model.addAttribute( s: "names", names);
        return "home";
    }
}
```

Рисунок 2 - AutoCompleteController.java

Следом создадим страницу «home.html» в каталоге «/src/main/respources/templates/» и добавим следующий контент (рис.3).

```
<html xmlns:th="http://www.thymeleaf.org">
<head>
 <title>Auto-complete example JQuery</title>
<script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<link rel="stylesheet"
    href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/themes/smoothness/jquery-ui.css">
<script
    src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>
<script th:inline="javascript">
    var real_data = /*[[${names}]]*/'noValue';
    $(document).ready(function() {
        $("#nameInput").autocomplete({
            minLength : 1,
            source : real_data,
            select : function(e, ui) {
                $("#nameOutput").text('You have selected : ' + ui.item.value);
                return ui.item.value;
            }
        })
    });
</script>
</head>
<body>
    <div class="ui-widget">
        <label for="nameInput">search Name: </label><input id="nameInput" />
        <h3 id="nameOutput"></h3>
    </div>
</body>
</html>
```

Рисунок 3 - home.html

Здесь следует обратить внимание на следующие важные моменты:

- В строке 10 идет присваивание значение атрибута «Model» переменной «javascript» с именем «real_data».
- В строке номер 12 настраиваем функцию автозаполнения JQuery UI, вызывая функцию «autocomplete()»
- Элемент HTML <input> с идентификатором «nameInput» - это элемент, который будет вызывать функцию автозаполнения.
- minLength: установили минимальную длину ввода в 1 символ.
- select: вызывает указанную функцию при выборе любого из значений автозаполнения.

Теперь запустим приложение и проверим ее работоспособность на локальном сервере (рис.4-5).

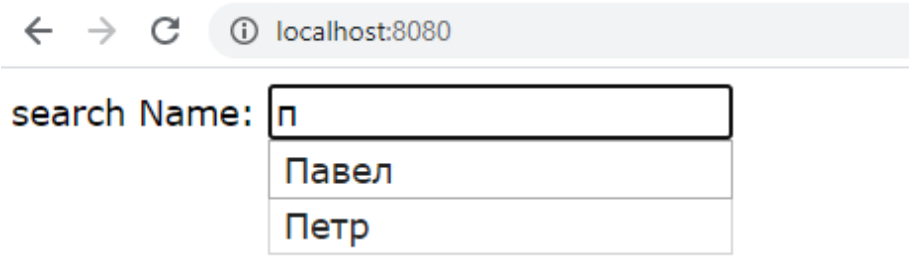

Рисунок 4 – Проверка приложения

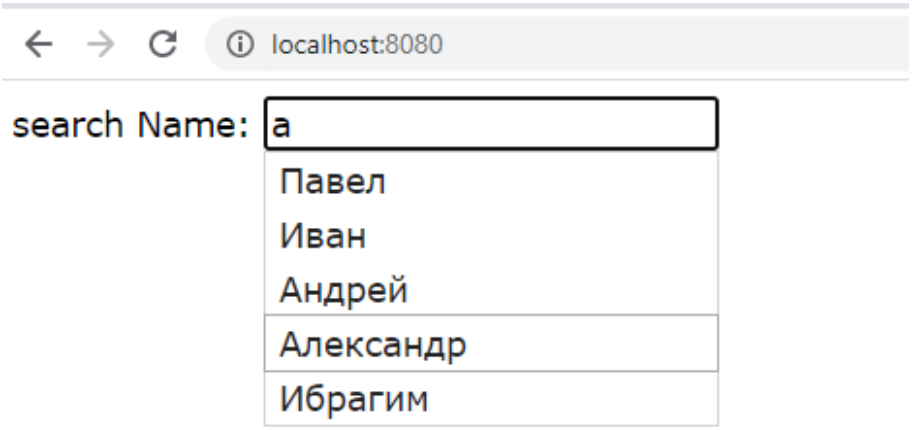

Рисунок 5 – Проверка приложения

В данной статье была создана функция автозаполнения в приложении Spring Web MVC с помощью Spring Boot и Thymeleaf.

Библиографический список

1. Джемалетдинов А.Б., Шевченко А.А. Spring boot: создание тестов для spring mvc контроллеров // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. №4(18). С. 104-111.
2. Зарайский В.И. Разработка модуля автоматизации работы с конференциями в кафедральном приложении // Вестник Ульяновского государственного технического университета. 2019. №3. С. 74-82.
3. Ибраимов Р.И. Развертывание spring приложения с помощью сервиса aws ec2 и docker-контейнеров // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2020. №1(27). С. 138-147.
4. Кабардинский Е.О., Ивашко А.Г. Сравнительный анализ сервисных шин предприятия (ESB) // Математическое и информационное моделирование. 2017. №10. С. 177-185.
5. Ибраимов Р.И., Зайчик А.Р., Минзатров Н.С. Разработка генеалогического дерева средствами фреймвока spring boot // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. №4(18). С. 18-23.