Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
|  |
| Кафедра «Информатика» |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №4**

|  |
| --- |
| Spring MVC |
| Тема |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  |  |  | А.С. Черниговский |
|  | |  |  |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ19-16/1б 031939175 | | |  |  |  | А.Д. Непомнящий |
|  | номер группы, зачетной книжки | | |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2021

1. Цель работы

Цель состоит в ознакомлении с шаблоном MVC в Spring Framework и тем, как он используется при создании веб-приложений.

1. Задачи

Задача работы – изменить практическую работу №3 таким образом, чтобы она представляла собой веб-приложение.

Веб-приложение должно иметь следующие страницы.

1. Главная страница, содержит приветствие и ссылки на другие страницы.
2. Страница просмотра таблицы записей.
3. Страница добавления новой записи в таблицу.
4. Страница редактирования записи.
5. Страница удаления записи из таблицы БД.
6. Страница просмотра записей согласно некоторому критерию.

Помимо этого, должно быть осуществлено не менее двух проверок входных данных, сопровождающихся соответствующими сообщениями об ошибках.

Вариант 10. Одежда.

1. Ход работы

В ходе работы было изменено в соответствии с заданием приложение, написанное в ходе третьей работы. В листингах приведен код некоторых файлов.

Листинг 1 – код класса AddController

package com.github.durakin.isdlabs.lab4.controller;

import com.github.durakin.isdlabs.lab4.form.ApparelForm;

import com.github.durakin.isdlabs.lab4.service.impl.ApparelServiceImpl;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import javax.validation.Valid;

@Controller

@RequestMapping("/add")

public class AddController {

private final ApplicationContext context;

private final ApparelServiceImpl apparelService;

@Autowired

public AddController(ApplicationContext applicationContext) {

this.context = applicationContext;

this.apparelService = context.getBean("apparelServiceImpl", ApparelServiceImpl.class);

}

@GetMapping

public String getForm(Model model) {

model.addAttribute("apparelForm", new ApparelForm());

return "add";

}

@PostMapping

public String addByForm(@Valid ApparelForm apparelForm, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

return "add";

}

var id = apparelService.add(

apparelForm.getType(),

apparelForm.getPrice(),

apparelForm.getInStock(),

apparelForm.getSize(),

apparelForm.getSex());

return "redirect:/find?id=" + id;

}

}

Листинг 2 – Код класса FindController

package com.github.durakin.isdlabs.lab4.controller;

import com.github.durakin.isdlabs.lab4.entity.Apparel;

import com.github.durakin.isdlabs.lab4.form.IdForm;

import com.github.durakin.isdlabs.lab4.form.MaxPriceForm;

import com.github.durakin.isdlabs.lab4.service.impl.ApparelServiceImpl;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import javax.validation.Valid;

import java.util.ArrayList;

import java.util.List;

@Controller

@RequestMapping("/find")

public class FindController {

private final ApplicationContext context;

private final ApparelServiceImpl apparelService;

@Autowired

public FindController(ApplicationContext applicationContext) {

this.context = applicationContext;

this.apparelService = context.getBean("apparelServiceImpl", ApparelServiceImpl.class);

}

@GetMapping

public String viewByIdOrAll(Model model) {

model.addAttribute("idForm", new IdForm());

model.addAttribute("maxPriceForm", new MaxPriceForm());

var apparels = apparelService.findAll();

model.addAttribute("apparels", apparels);

return "find";

}

@GetMapping(params = "id")

public String viewByIdOrAll(@Valid IdForm idForm, BindingResult bindingResultIdForm, MaxPriceForm maxPriceForm, Model model) {

if (bindingResultIdForm.hasErrors()) {

return "find";

}

if (idForm.getId() == 0) {

return "redirect:/find";

}

List<Apparel> apparels = new ArrayList<>();

var apparel = apparelService.findById(idForm.getId());

if(apparel != null) {

apparels.add(apparelService.findById(idForm.getId()));

}

model.addAttribute("apparels", apparels);

return "find";

}

Окончание листинга 2

@GetMapping(params = "price")

public String viewByType(@Valid MaxPriceForm maxPriceForm, BindingResult bindingResultTypeForm, IdForm idForm, Model model) {

if (bindingResultTypeForm.hasErrors()) {

return "find";

}

var apparels = apparelService.findByMaxPrice(maxPriceForm.getPrice());

model.addAttribute("apparels", apparels);

return "find";

}

}

Листинг 3 – Код html-документа add.html

<!DOCTYPE HTML>

<html xmlns:th="https://www.thymeleaf.org" lang="en">

<head>

<title>Add | RKIS\_Lab4</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<header>

<h1>Add apparel</h1>

</header>

<article>

<form class="form" th:object="${apparelForm}" method="post">

<div>

<p>Type</p>

<label>

<input type="text" name="type" th:field="\*{type}" placeholder="type"/>

</label>

<span th:if="${#fields.hasErrors('type')}" th:errors="\*{type}">type error</span>

</div>

<div>

<p>Size</p>

<label>

<input type="text" name="size" th:field="\*{size}" placeholder="size"/>

</label>

<span th:if="${#fields.hasErrors('size')}" th:errors="\*{size}">size error</span>

</div>

<div>

<p>Sex</p>

<label>

<input type="text" name="sex" th:field="\*{sex}" placeholder="sex"/>

</label>

<span th:if="${#fields.hasErrors('sex')}" th:errors="\*{sex}">sex error</span>

</div>

<div>

<p>Price</p>

<label>

Окончание листинга 3

<input type="number" name="price" th:field="\*{price}" placeholder="price"/>

</label>

<span th:if="${#fields.hasErrors('price')}" th:errors="\*{price}">price error</span>

</div>

<div>

<p>In stock</p>

<label>

<input type="number" name="inStock" th:field="\*{inStock}" placeholder="inStock"/>

</label>

<span th:if="${#fields.hasErrors('inStock')}" th:errors="\*{inStock}">inStock error</span>

</div>

<div class="buttons">

<br>

<button type="submit">Add</button>

</div>

</form>

</article>

<footer>

<hr>

<a href="Lab4\_war/">Home</a>

</footer>

</body>

</html>

Листинг 4 – Код html-документа find.html

<!DOCTYPE HTML>

<html xmlns:th="https://www.thymeleaf.org" lang="en">

<head>

<title>Find | RKIS\_Lab4</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<header>

<h1>Check apparels</h1>

</header>

<article>

<section>

<form th:object="${idForm}" method="get">

<div>

<p>ID (0 - all)</p>

<label>

<input type="number" name="id" th:field="\*{id}" placeholder="id"/>

</label>

<span th:if="${#fields.hasErrors('id')}" th:errors="\*{id}">id error</span>

</div>

<div>

<br>

<button type="submit">Find by ID</button>

</div>

</form>

<form th:object="${maxPriceForm}" method="get">

Окончание листинга 4

<div>

<p>Maximal price</p>

<label>

<input type="number" name="price" th:field="\*{price}" placeholder="price"/>

</label>

<span th:if="${#fields.hasErrors('price')}" th:errors="\*{price}">maxPrice error</span>

</div>

<div>

<br>

<button type="submit">Find by maximal price</button>

</div>

</form>

</section>

<hr>

<section>

<p th:text="'Found: ' + ${apparels != null ? apparels.size() : 0}">Number: </p>

<table>

<thead>

<tr>

<th>Id</th>

<th>Type</th>

<th>Price</th>

<th>In stock</th>

<th>Size</th>

<th>Sex</th>

</tr>

</thead>

<tbody>

<tr th:each="apparel, itemStat : \*{apparels}">

<td><p th:text="\*{apparel.getId()}"></p></td>

<td><p th:text="\*{apparel.getApparelType()}"></p></td>

<td><p th:text="\*{apparel.getPrice()}"></p></td>

<td><p th:text="\*{apparel.getInStock()}"></p></td>

<td><p th:text="\*{apparel.getSize()}"></p></td>

<td><p th:text="\*{apparel.getSex()}"></p></td>

</tr>

</tbody>

</table>

</section>

</article>

<footer>

<hr>

<a href="Lab4\_war/">Home</a>

</footer>

</body>

</html>

1. Вывод

Было произведено ознакомление с шаблоном MVC в Spring Framework и тем, как он используется при создании веб-приложений.