ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| кандидат технических наук |  |  |  | П.А. Степанов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5  Разработка приложения с использованием thymeleaf |
| по дисциплине: Технология разработки серверных информационных систем |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Самарин Д.В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Задание на лабораторную работу.

# Реализуйте задачи из лабораторной работы 1 средствами Thymeleaf

# Обратите внимание на локализацию приложения. Русский язык должен поддерживаться.

# Вариант:

Управление доступом к комнатам

Листинг<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"  
 xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">  
<head>  
 <meta charset="utf-8"></meta>  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"></meta>  
 <meta name="viewport" content="width=device-width, initial-scale=1"></meta>  
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"></meta>  
 <title>Title</title>  
 <link href="../static/css/bootstrap.min.css" th:href="@{/css/bootstrap.min.css}" rel="stylesheet">  
  
 <script src="http://code.jquery.com/jquery-latest.js"></script>  
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.3/angular.min.js"></script>  
</head>  
  
<body>  
 <div class="navbar navbar-inverse navbar-fixed-top" th:fragment="header">  
 <div class="container">  
 <div class="navbar-header">  
 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".nav-collapse">  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 </button>  
 </div>  
 <div class="navbar-collapse collapse">  
 <ul class="nav navbar-nav">  
 <li th:classappend="${module == 'home' ? 'active' : ''}">  
 <a href="#" th:href="@{/}">Home</a>  
 </li>  
  
 </ul>  
 <ul class="nav navbar-nav navbar-right">  
 <li th:if="${#authorization.expression('!isAuthenticated()')}">  
 <a href="/signin" th:href="@{/login}">  
 <span class="glyphicon glyphicon-log-in" aria-hidden="true"></span>&nbsp;Sign in  
 </a>  
 </li>  
 <li th:if="${#authorization.expression('isAuthenticated()')}">  
 <a href="/logout" th:href="@{/logout}" onclick="$('#form').submit();">  
 <span class="glyphicon glyphicon-log-out" aria-hidden="true"></span>&nbsp;Logout  
 </a>  
 <form style="visibility: hidden" id="form" method="post" action="#" th:action="@{/logout}"></form>  
  
 </li>  
 </ul>  
 </div>  
 </div>  
 </div>  
 <br>  
 <br>  
 <div class="content" style="margin-left: auto; margin-right: auto; width: fit-content">  
 <h1 th:text = "#{ts.ts}">Траснпортники </h1>  
  
 <table style="border: 1px solid;width: 100%;border-collapse: collapse;">  
 <thead>  
 <tr>  
 <th>Type</th><th>Number</th>  
 </tr>  
 </thead>  
 <tr th:each="transporter: ${transporters}">  
 <td th:text="${transporter.type}" style=" background: linear-gradient(rgb(170,170,170), lightgray );"></td>  
 <td th:text="${transporter.name}" style=" background: linear-gradient(rgb(170,170,170), lightgray );"></td>  
 </tr>  
 </table>  
 </div>  
  
</body>  
</html>

package io.swagger.model;  
  
import io.swagger.models.auth.In;  
import lombok.Getter;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
@Entity  
@Table(name = "transporters")  
public class transporters {  
 public transporters(Integer id, String name, Integer type){  
 this.id = id;  
 this.name = name;  
 this.type = type;  
 }  
 @Id  
 @GeneratedValue (strategy = GenerationType.*IDENTITY*)  
 @Column(name = "id")  
 private Integer id;  
  
 public transporters() {  
  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public Integer GetId(){  
 return id;  
 }  
  
  
 @Getter  
 @Column(name = "name")  
 private String name;  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Getter  
 @Column(name = "type")  
 private Integer type;  
  
 public void setType(Integer type) {  
 this.type = type;  
 }  
}

package io.swagger.api;  
  
import io.swagger.service.transportersService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.servlet.ModelAndView;  
  
@Controller  
public class transportersController {  
 @Autowired  
 transportersService tService;  
  
 @RequestMapping(value = "/ts/{id}", method = RequestMethod.*GET*)  
 public ModelAndView transporters(@PathVariable("id") Integer id){  
  
 ModelAndView mav = new ModelAndView("ts");  
 mav.addObject("transporters", tService.findByType(id));  
 return mav;  
 }  
}