**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №1

Выполнил:

студент группы ИУ5-55Б Кожиев Т.К.

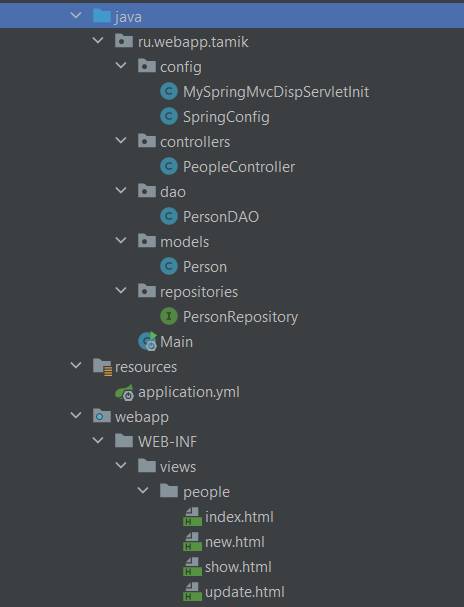
Преподаватель:

Гапанюк Ю.Е

2021

(Сделал проект на Java\*)

**Cтруктура:**

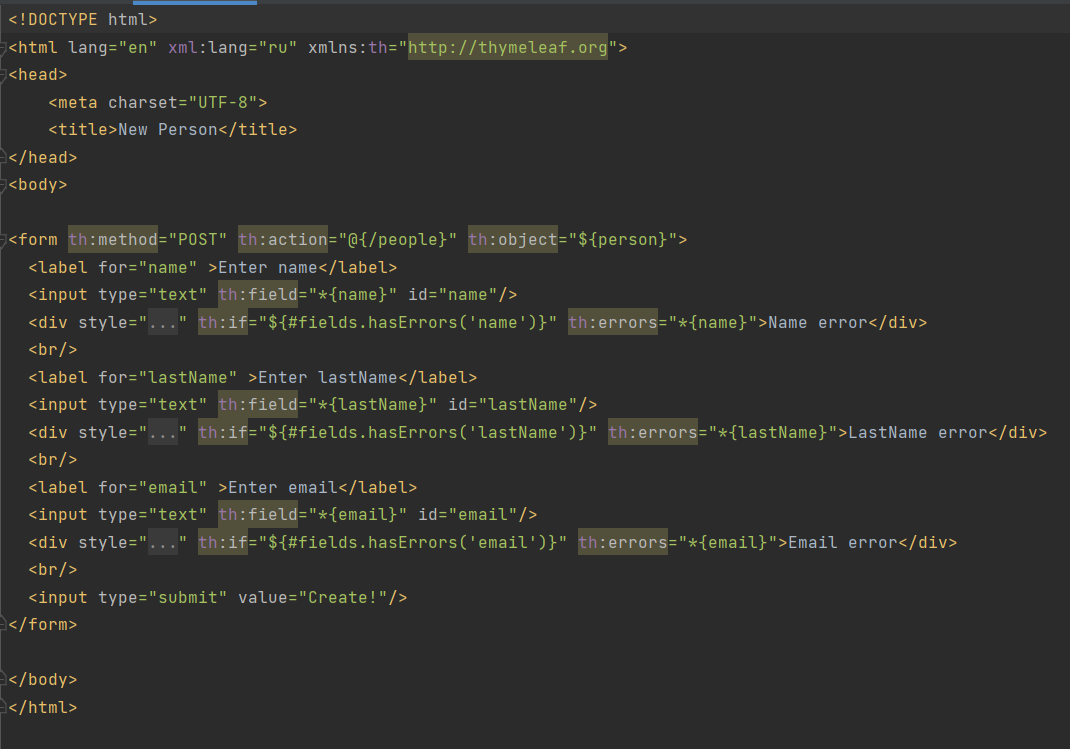
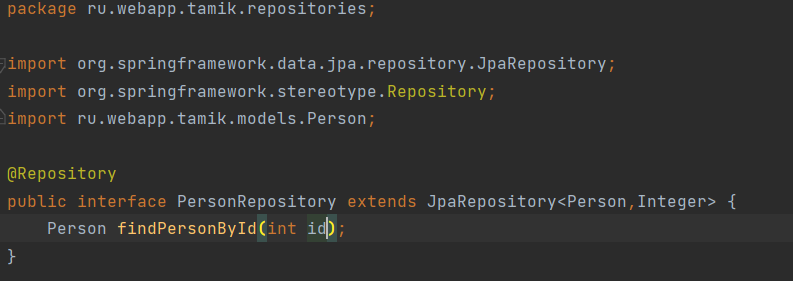
****

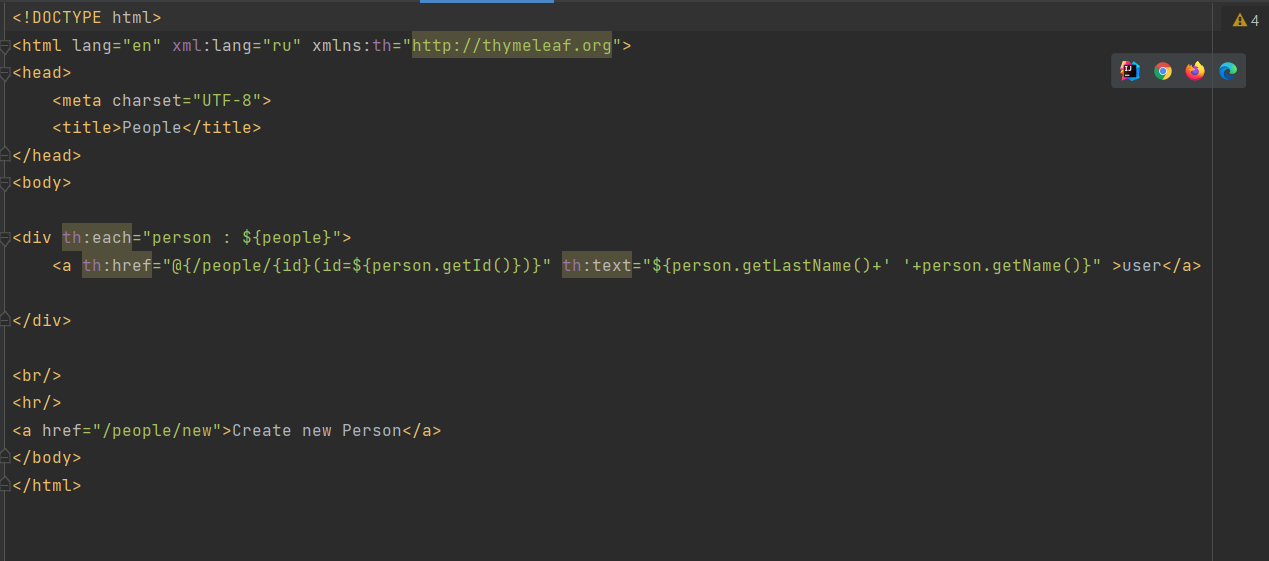
**Код:**

*PeopleController.java:*

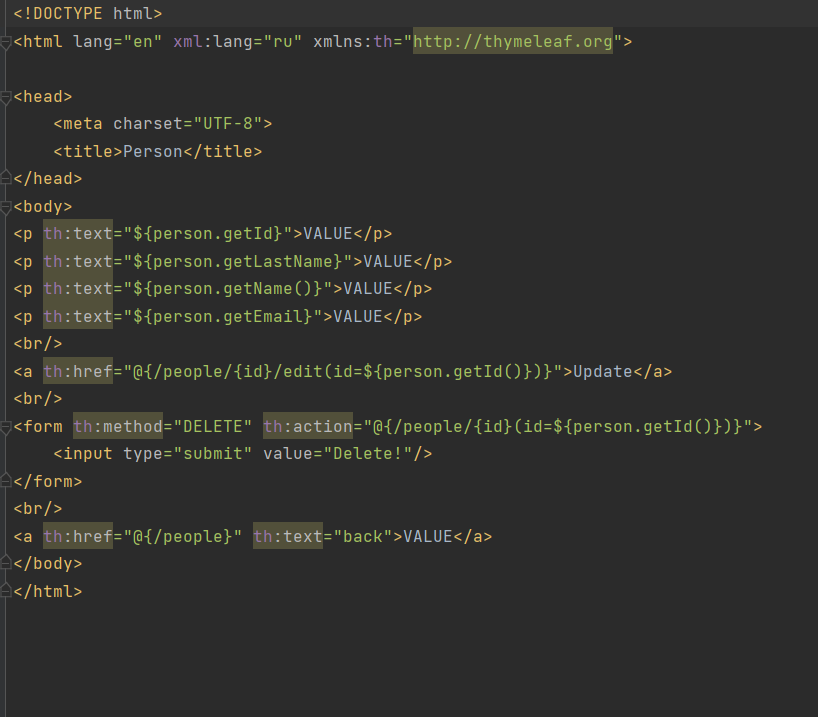
package ru.webapp.tamik.controllers;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.transaction.annotation.Transactional;  
import org.springframework.ui.Model;  
import org.springframework.validation.BindingResult;  
import org.springframework.web.bind.annotation.\*;  
import ru.webapp.tamik.dao.PersonDAO;  
import ru.webapp.tamik.models.Person;  
  
import javax.validation.Valid;  
  
@Controller  
@RequestMapping("/people")  
public class PeopleController {  
  
 private final PersonDAO personDAO;  
  
 public PeopleController(PersonDAO personDAO) {  
 this.personDAO = personDAO;  
 }  
  
 @GetMapping()  
 public String index(Model model) {  
 model.addAttribute("people", personDAO.index());  
 return "people/index";  
 }  
  
 @GetMapping("/{id}")  
 public String show(@PathVariable("id") int id, Model model) {  
 model.addAttribute("person", personDAO.show(id));  
 return "people/show";  
 }  
  
 @GetMapping("/new")  
 public String newPerson(Model model) {  
 model.addAttribute("person", new Person());  
 return "people/new";  
 }  
  
 @PostMapping()  
 @Transactional  
 public String create(@ModelAttribute("person") @Valid Person person, BindingResult bindingResult) {  
 if (bindingResult.hasErrors())  
 return "people/new";  
 personDAO.save(person);  
 return "redirect:/people";  
 }  
  
 @GetMapping("/{id}/edit")  
 public String edit(@PathVariable("id") int id, Model model) {  
 model.addAttribute("person", personDAO.show(id));  
 return "people/update";  
 }  
  
 @PatchMapping("/{id}")  
 public String update(@PathVariable("id") int id, @ModelAttribute("person") @Valid Person person,  
 BindingResult bindingResult) {  
 if(bindingResult.hasErrors())  
 return "people/update";  
 personDAO.update(id, person);  
 return "redirect:/people";  
 }  
  
 @DeleteMapping("/{id}")  
 public String delete(@PathVariable("id") int id) {  
 personDAO.delete(id);  
 return "redirect:/people";  
 }  
  
}

*new.html:*

*PersonRepository.java:* *index.html:*



*show.html:*



*Person.java:*

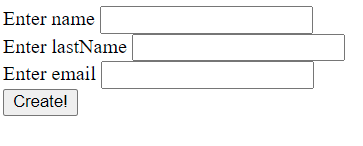
package ru.webapp.tamik.models;  
  
import lombok.Data;  
  
import javax.persistence.\*;  
  
import javax.validation.constraints.Email;  
import javax.validation.constraints.NotEmpty;  
import javax.validation.constraints.Size;  
  
  
@Data  
@Entity  
@Table(name="Person")  
public class Person {  
 @Id  
 @Column(name="id")  
 @GeneratedValue  
 Integer id;  
  
 @NotEmpty(message= "Имя не может быть пустым")  
 @Size(min=2,max=30,message="Имя должно быть не меньше 2 и не больше 30 букв")  
 @Column(name="name")  
 private String name;  
  
 @NotEmpty(message= "Фамилия не может быть пустой")  
 @Size(min=2,max=30,message="Фамилия должна быть не меньше 2 и не больше 30 букв")  
 @Column(name="lastName")  
 private String lastName;  
  
 @Email(message="Емаил введен неправильно")  
 @Column(name="email")  
 private String email;  
  
 public Person() {  
 }  
  
 public Person(String name, String lastName, String email) {  
 this.name = name;  
 this.lastName = lastName;  
 this.email = email;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

**Демонстрация:**

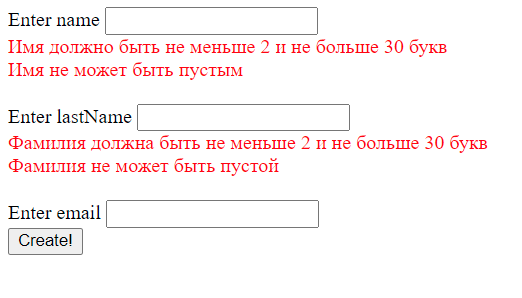
HomePage



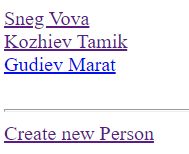
Create New Person:



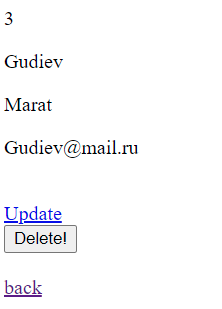
Валидация



Добавили:



Посмотрели:



Можно изменить или удалить.

DataBase:

