编号：

2020 - 2021学年第 二 学期



**实 验 报 告**

实验课程名称 软件架构与应用开发

专 业 班 级 计算机1803

学 生 姓 号 31801150

学 生 姓 名 张帅

实验指导教师 彭彬

一. 实验目的和要求

**实验要求：**

1.利用Thymetleaf技术实现学生基本信息记录集的翻页功能

2. 利用MVC模式实现学生基本信息展示

3.创建学生信息表单的录入功能，并提供学号长度、性别、身份证（注意身份证号的校验规则）和电子邮件功能，并显示提交成功后的学生信息

二. 实验内容

1.利用Thymetleaf技术实现学生基本信息记录集的翻页功能

首先定义数据结构用于存储学生信息：

@Data  
public class stuEntity {  
  
 @NotNull(message = "编号不为空")  
 private Integer id;  
  
 @NotEmpty(message = "姓名不为空")  
 @Length(min=1,max=6,message = "姓名长度为1-6")  
 private String name;  
  
 @NotNull(message = "年龄不为空")  
 @DecimalMax(value = "100",message = "年龄太大")  
 @DecimalMin(value = "0",message = "年龄太小")  
 private Integer age;  
  
 @NotNull(message = "性别不为空")  
 @DecimalMax(value = "1",message = "性别异常")  
 @DecimalMin(value = "0",message = "性别异常")  
 private Integer sex;  
  
 @NotEmpty(message = "身份证不为空")  
 @Pattern(regexp = "(^[1-9]\\d{5}(19|([23]\\d))\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{3}[0-9Xx]$)|(^[1-9]\\d{5}\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{2}[0-9Xx]$)",  
 message = "身份证格式错误")  
 private String number;  
  
 @Email  
 @NotEmpty(message = "邮箱不为空")  
 private String email;  
  
 public stuEntity(){}  
  
 public stuEntity(Integer id, String name, Integer age, Integer sex, String number, String email) {  
 this.id = id;  
 this.name = name;  
 this.age = age;  
 this.sex = sex;  
 this.number = number;  
 this.email = email;  
 }  
}

2.设置首页，用于初始化数据，每次访问localhost：8080时进行数据的初始化

ArrayList<stuEntity> list = new ArrayList<>();  
@RequestMapping(value = "/")  
String index(){  
  
 if(list.size()==0)  
 {  
 list.add(new stuEntity(1,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
 list.add(new stuEntity(2,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
 list.add(new stuEntity(3,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
 list.add(new stuEntity(4,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
 list.add(new stuEntity(5,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
 list.add(new stuEntity(6,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
 list.add(new stuEntity(7,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
 list.add(new stuEntity(8,"zs",20,1,"33048120000603001X","1016868503@qq.com"));  
  
 }  
  
  
 return "index";  
}

定义/list路径为显示学生列表的页面，接受参数page作为页码，每页3条数据，每次返回当前页码（page），总共页码（sum）判断是否达到最后一页（flag），并且截取数组中指定的元素转存到re列表中，返回给页面

@RequestMapping(value = "/list",method = RequestMethod.*GET*)  
ModelAndView listdata(int page){  
 ModelAndView mav = new ModelAndView("list");  
  
 int flag=0;  
 List<stuEntity> re;  
 if(page \* 3+3 < list.size()) {  
 flag = 0;  
 re = list.subList(page \* 3, page \* 3 + 3);  
 }  
 else  
 {  
 flag = 1;  
 re = list.subList(page \* 3,list.size());  
 }  
 System.*out*.println(list.size());  
 mav.addObject("flag",flag);  
 mav.addObject("msg",re);  
 mav.addObject("sum",(list.size()-1)/3);  
 mav.addObject("index",page);  
 return mav;  
}

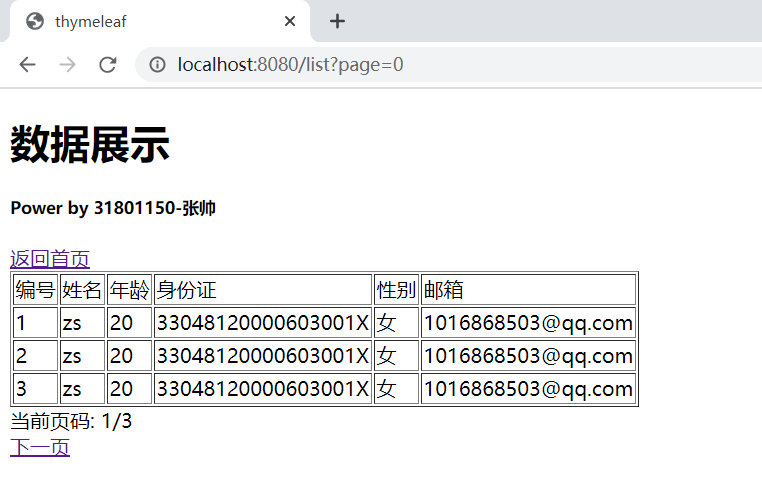
定义index界面，定义链接，访问学生信息列表，并且传递初始化参数0用于显示第一页的数据

<!DOCTYPE HTML>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>thymeleaf</title>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
</head>  
<body>  
<h1>首页</h1>  
<h5>Power by 31801150-张帅</h5>  
  
<a th:href="@{'/list?page=0'}">进入学生信息系统</a>  
<a th:href="@{'/add'}">添加信息</a>  
</body>

定义list界面，显示学生数据，通过当前界面page是否是0和传递来的flag参数，决定是否显示上一页和下一页的按钮，点击按钮时跳转至对应的页码，通过改变传递的参数来决定页码。

<!DOCTYPE HTML>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>thymeleaf</title>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
</head>  
<body>  
<h1>数据展示</h1>  
<h5>Power by 31801150-张帅</h5>  
<a th:href="@{'/'}">返回首页</a>  
<div></div>  
  
<table border="1">  
 <thead>  
 <td >编号</td>  
 <td >姓名</td>  
 <td >年龄</td>  
 <td >身份证</td>  
 <td >性别 </td>  
 <td >邮箱</td>  
 </thead>  
 <tr th:each="item : ${msg}">  
  
  
 <td th:text="${item.id}"></td>  
 <td th:text="${item.name}"></td>  
 <td th:text="${item.age}"></td>  
 <td th:text="${item.number}"></td>  
 <td th:if="${item.sex}==0"> 男</td>  
 <td th:if="${item.sex}==1"> 女</td>  
 <td th:text="${item.email}"></td>  
  
  
 </tr>  
</table>  
  
  
<a th:href="@{'/list?page='+${index - 1}}" th:if="${index}>0">上一页</a>  
<div th:text="'当前页码: '+${index + 1}+'/'+${sum + 1}"></div>  
<a th:href="@{'/list?page='+${index + 1}}" th:if="${flag}==0">下一页</a>  
  
  
</body>

结果：

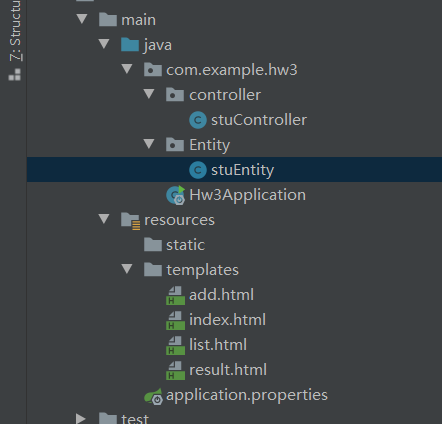






2. 利用MVC模式实现学生基本信息展示

定义了数据的实体类，编写了控制器接受用户的输入并调用模型和视图去完成用户的需求，用户的视图显示在html中。



控制器中，使用ModelAndView封装并返回数据

@RequestMapping(value = "/list",method = RequestMethod.*GET*)  
ModelAndView listdata(int page){  
 ModelAndView mav = new ModelAndView("list");  
  
 int flag=0;  
 List<stuEntity> re;  
 if(page \* 3+3 < list.size()) {  
 flag = 0;  
 re = list.subList(page \* 3, page \* 3 + 3);  
 }  
 else  
 {  
 flag = 1;  
 re = list.subList(page \* 3,list.size());  
 }  
 System.*out*.println(list.size());  
 mav.addObject("flag",flag);  
 mav.addObject("msg",re);  
 mav.addObject("sum",(list.size()-1)/3);  
 mav.addObject("index",page);  
 return mav;  
}

3.创建学生信息表单的录入功能，并提供学号长度、性别、身份证（注意身份证号的校验规则）和电子邮件功能，并显示提交成功后的学生信息

在实体类中，使用validation对数据编写校验规则，email使用@Email注释实现校验，身份证则使用正则表达式进行校验。

@Data  
public class stuEntity {  
  
 @NotNull(message = "编号不为空")  
 private Integer id;  
  
 @NotEmpty(message = "姓名不为空")  
 @Length(min=1,max=6,message = "姓名长度为1-6")  
 private String name;  
  
 @NotNull(message = "年龄不为空")  
 @DecimalMax(value = "100",message = "年龄太大")  
 @DecimalMin(value = "0",message = "年龄太小")  
 private Integer age;  
  
 @NotNull(message = "性别不为空")  
 @DecimalMax(value = "1",message = "性别异常")  
 @DecimalMin(value = "0",message = "性别异常")  
 private Integer sex;  
  
 @NotEmpty(message = "身份证不为空")  
 @Pattern(regexp = "(^[1-9]\\d{5}(19|([23]\\d))\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{3}[0-9Xx]$)|(^[1-9]\\d{5}\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{2}[0-9Xx]$)",  
 message = "身份证格式错误")  
 private String number;  
  
 @Email  
 @NotEmpty(message = "邮箱不为空")  
 private String email;

编写/add路径的方法，使用@Valid注释对传递的数据进行校验，发生错误则不进行跳转并且显示错误，数据均正确则跳转至提交完成界面

@RequestMapping(value = "/add",method = RequestMethod.*GET*)  
String add(stuEntity stu){  
  
 return "add";  
}  
  
@RequestMapping(value = "/result",method = RequestMethod.*GET*)  
String show(){  
  
 return "result";  
}  
  
@RequestMapping(value = "/add",method = RequestMethod.*POST*)  
String addata(@Valid stuEntity stu, BindingResult bindingResult, RedirectAttributes attr){  
  
 if(bindingResult.hasErrors()){  
 return "add";  
 }  
 System.*out*.println(stu.getSex());  
 list.add(stu);  
 attr.addFlashAttribute("stu",stu);  
 return "redirect:/result";  
}

编写add界面，主体是表单，绑定stuEntity数据结构，其中当表单数据校验失败时，显示报错信息，性别由单选框进行数据的绑定

<!DOCTYPE HTML>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>thymeleaf</title>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
</head>  
<style type="text/css"> .warn{color:red} </style>  
<body>  
  
<h1>数据添加</h1>  
<h5>Power by 31801150-张帅</h5>  
<a th:href="@{'/'}">返回首页</a>  
  
  
  
<form th:action="@{/add}" th:object="${stuEntity}" method="post">  
 <div>  
 <div>  
 <span>编号:</span>  
 <span><input type="number" th:field="\*{id}"/></span>  
 <span class="warn" th:if="${#fields.hasErrors('id')}" th:errors="\*{id}">编号有误</span>  
 </div>  
 <div>  
 <span>姓名:</span>  
 <span><input type="text" th:field="\*{name}"/></span>  
 <span class="warn" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}">姓名错误</span>  
 </div>  
 <div>  
 <span>年龄:</span>  
 <span><input type="number" th:field="\*{age}"/></span>  
 <span class="warn" th:if="${#fields.hasErrors('age')}" th:errors="\*{age}">年龄错误</span>  
 </div>  
 <div>  
 <input id="male" name="sex" type="radio" value="0" th:checked="\*{sex==0}"/>  
 <label for="male">男</label>  
 <input id="female" name="sex" type="radio" value="1" th:checked="\*{sex==1}"/>  
 <label for="female">女</label>  
 <span class="warn" th:if="${#fields.hasErrors('sex')}" th:errors="\*{sex}">性别错误</span>  
  
 </div>  
 <div>  
 <span>身份证:</span>  
 <span><input type="text" th:field="\*{number}"/></span>  
 <span class="warn" th:if="${#fields.hasErrors('number')}" th:errors="\*{number}">身份证错误</span>  
 </div>  
 <div>  
 <span>邮箱:</span>  
 <span><input TYPE="text" th:field="\*{email}"/></span>  
 <span class="warn" th:if="${#fields.hasErrors('email')}" th:errors="\*{email}">邮箱错误</span>  
 </div>  
  
 <div>  
 <span>  
 <button type="submit">提交</button>  
 </span>  
 </div>  
 </div>  
</form>  
</body>

编写result界面，当提交数据成功时，来到该页面，显示刚刚提交的数据

<!DOCTYPE HTML>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>thymeleaf</title>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
</head>  
<body>  
<h1>提交成功</h1>  
<h5>Power by 31801150-张帅</h5>  
<a th:href="@{'/'}">返回首页</a>  
<div>  
  
 <div th:text="'编号 '+${stu.id}"></div>  
 <div th:text="'姓名 '+${stu.name}"></div>  
 <div th:text="'年龄 '+${stu.age}"></div>  
 <div th:text="'身份证 '+${stu.number}"></div>  
 <div th:if="${stu.sex}==0">性别 男</div>  
 <div th:if="${stu.sex}==1">性别 女</div>  
 <div th:text="'邮箱 '+${stu.email}"></div>  
</div  
</body>

效果演示：  



