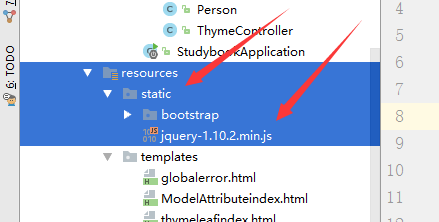
# 1.spring boot 引入thymeleaf模板这里不再讲述

# 2.如何引入资源文件

## 1.解释：根据默认原则，脚步样式，图片等静态资源应该放在static下面



## 2.html文件如何引入

<**link th:href="@{bootstrap/css/bootstrap.min.css}" rel="stylesheet"**/>   
<**link th:href="@{bootstrap/css/bootstrap-theme.min.css}" rel="stylesheet"**/>

<**script type="text/javascript" src="../static/js/bootstrapValidator.js" th:src="@{/js/bootstrapValidator.js}"**></**script**>

# 3.编写实体类person

**public class** Person {  
 **private** String **name**;  
 **private** Integer **age**;  
   
 **public** Person() {  
 **super**();  
 }  
 **public** Person(String name, Integer age) {  
 **super**();  
 **this**.**name** = name;  
 **this**.**age** = age;  
 }  
 **public** String getName() {  
 **return name**;  
 }  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
 **public** Integer getAge() {  
 **return age**;  
 }  
 **public void** setAge(Integer age) {  
 **this**.**age** = age;  
 }

# 2.编写controller

@Controller  
**public class** ThymeController {  
 @RequestMapping(**"/thymeleafindex"**)  
 **public** String index(Model model){  
 Person single = **new** Person(**"aa"**,11);  
 List<Person> people = **new** ArrayList<Person>();  
 Person p1 = **new** Person(**"xx"**,11);  
 Person p2 = **new** Person(**"yy"**,22);  
 Person p3 = **new** Person(**"zz"**,33);  
 people.add(p1);  
 people.add(p2);  
 people.add(p3);  
 model.addAttribute(**"singlePerson"**, single);  
 model.addAttribute(**"people"**, people);  
  
 **return "thymeleafindex"**;  
 }  
  
}

# 3.编写html

<**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h3 class="panel-title"**>访问model</**h3**>  
 </**div**>  
 <**div class="panel-body"**>  
 <**span th:text="${singlePerson.name}"**></**span**>   
 </**div**>  
 </**div**>  
   
 <**div th:if="${not #lists.isEmpty(people)}"**>  
 <**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h3 class="panel-title"**>列表</**h3**>  
 </**div**>  
 <**div class="panel-body"**>  
 <**ul class="list-group"**>  
 <**li class="list-group-item" th:each="person:${people}"**>  
 <**span th:text="${person.name}"**></**span**>  
 <**span th:text="${person.age}"**></**span**>  
 <**button class="btn" th:onclick="'getName(\'' + ${person.name} + '\');'"**>获得名字</**button**>  
 </**li**>  
 </**ul**>  
 </**div**>  
 </**div**>  
</**div**>

<**script th:src="@{jquery-1.10.2.min.js}" type="text/javascript"**></**script**>*<!-- 2 -->* <**script th:src="@{bootstrap/js/bootstrap.min.js}"**></**script**>*<!-- 2 -->* <**script th:inline="javascript"**>  
 **var *single*** = [[${singlePerson}]];  
 **console**.log(***single***.name+**"/"**+***single***.age)  
   
 **function** *getName*(name){  
 **console**.log(name);  
 }  
 </**script**>  
   
 </**body**>  
</**html**>

# 4.代码解释

## 1. 访问model中的数据

<span th:text="${singlePerson.name}"></span>

## 2.数据判断，判断people是否为空（支持>，<，>=等）

## <div th:if="${not #lists.isEmpty(people)}">

<**span th:if="${dispatchDTO.startStatus eq '0'}"**>待审核</**span**>

## 3.迭代显示th:each="person:${people}">

<**li class="list-group-item" th:each="person:${people}"**>  
 <**span th:text="${person.name}"**></**span**>  
 <**span th:text="${person.age}"**></**span**>  
 <**button class="btn" th:onclick="'getName(\'' + ${person.name} + '\');'"**>获得名字</**button**>   
 </**li**>

## 4.JavaScript中访问model

### 1.获取model数据var *single* = [[${singlePerson}]]; 2.

<**button class="btn" th:onclick="'getName(\'' + ${person.name} + '\');'"**>获得名字</**button**>

<**script th:inline="javascript"**>  
 **var *single*** = [[${singlePerson}]];  
 **console**.log(***single***.name+**"/"**+***single***.age)  
   
 **function** *getName*(name){  
 **console**.log(name);  
 }   
 </**script**>

## 5.表单提交th:action="@{/add}" th:method="post" >

<**form id="iform" th:action="@{/add}" th:method="post"** > *<!--th:object="${LargeOrder}"-->* <**table border="1"**>  
 <**tr**>  
 <**th**>auditState</**th**>  
 <**th**>auditPerson</**th**>  
 </**tr**>  
 <**tr**>  
 <**td**><**input type="text" name="auditState"** /></**td**>  
 <**td**><**input type="text" name="auditPerson"** /></**td**>  
 <**td**><**input type="submit" value="添加"**/></**td**>  
 </**tr**>  
 </**table**>  
</**form**>

# 5.thymeleaf 方言

## 1.select（数据字典）

<**select id = "ethnic" name="ethnic"class="form-control"**>  
 <**option th:each="ethnic, ethnicS : ${allEthnics}" th:value="${ethnic.key}" th:text="${ethnic.value}" th:selected="${studentInfo.ethnic == ethnic.key} "**>[[${ethnic.value}]]</**option**>  
</**select**>

## 2.数据字典

<**div th:text="${#pubcode.codeName('ethnic',studentInfo.ethnic)}"**></**div**>

# 6.th:each,轮播图 利用了bootstap前端框架

<**div class="item" th:classappend="${NewsDTO.first}?'active'" th:each="news,NewsDTO:${newsListDTO.banner}"**>  
 <**img th:src="${news.banner}" alt="图一"** />  
</**div**>

# 7. th:utext嵌入html显示内容，cms新闻

|  |
| --- |
| <**section th:utext="${news.content}"**> </**section**> |

# 8.日期，获取thymeleaf值

|  |
| --- |
| <**input readonly class="form-control" id="auditTime" th:value="${reassignApplication.auditTime}" name="auditTime" type="text"** />  <**script th:inline="JavaScript"**>  **var *auditTime*** = [[${reassignApplication.auditTime}]];  **if** (***auditTime***!=**null**) {  **var *auditTimeString*** = ***auditTime***.**year** + **'-'** + ***auditTime***.monthValue + **'-'** + ***auditTime***.dayOfMonth + **' '** + ***auditTime***.hour + **':'** + ***auditTime***.minute;  **$**(**'#auditTime'**).val(***auditTimeString***);  } </**script**> |

# 8.2 $("#principal").val();

|  |
| --- |
| <**input type="hidden" th:value="${principal}" id="principal"**/> <**input type="hidden" th:value="${studentInfo.status}" id="schStatus"**/>  **$**(**document**).ready(**function**(){  **var** principal = **$**(**"#principal"**).val();  **var** schStatus = **$**(**"#schStatus"**).val();  **if**(schStatus==**"01"**){  **$**(**"#auditPass"**).css({**"display"**:**"none"**});  **$**(**"#auditNoPass"**).css({**"display"**:**"none"**});  } |

# 9. th:object=

# 1.controller

@Controller  
**public class** AnimalController {  
  
 @RequestMapping(value = **"/list.html"**, method = RequestMethod.***GET***)  
 **public** String doAdd(Model model, @Valid AnimalForm form, BindingResult result){  
 System.***out***.println(**"动物名："** + form.getOname());  
 System.***out***.println(**"数量："** + form.getOcount());  
 System.***out***.println(**"备注："** + form.getMemo());  
 **if**(result.hasErrors()){  
 model.addAttribute(**"MSG"**, **"出错啦！"**);  
 }**else**{  
 model.addAttribute(**"MSG"**, **"提交成功！"**);  
 }  
 **return "vialidate"**;  
 }  
}

## 2.1解释：

### 2.1.1@Valid AnimalForm form，

@Valid表示要对该form进行验证，具体验证规则就是根据上面【一】里提到；  
spring框架会根据字段名称将页面传递过来的值绑定到animalForm中。

### 2.1.2 BindingResult result，

spring框架会将验证结果设置到该参数，并将该参数放到model传递给页面。  
 springMVC是非常灵活的，以下几种写法可以达到同样的效果：

# 4.html

<**br**/>  
<**div th:text="${MSG}"**>这里是信息提示.</**div**>  
<**br**/>  
<**div th:errors="${animalForm.oname}"**></**div**>  
<**div th:errors="${animalForm.ocount}"**></**div**>  
<**div th:errors="${animalForm.memo}"**></**div**>  
<**form id="iform" th:action="@{/list.html}" th:method="get" th:object="${animalForm}"**>  
 <**table border="1"**>  
 <**tr**>  
 <**th**>动物名称</**th**>  
 <**th**>数量</**th**>  
 <**th**>备注</**th**>  
 <**th**>Action</**th**>  
 </**tr**>  
 <**tr**>  
 <**td**><**input type="text" name="oname" value="" th:value="\*{oname}"**/></**td**>  
 <**td**><**input type="text" name="ocount" value="" th:value="\*{ocount}"**/></**td**>  
 <**td**><**input type="text" name="memo" value="" th:value="\*{memo}"**/></**td**>  
 <**td**><**input type="submit" value="添加"**/></**td**>  
 </**tr**>  
 </**table**>  
</**form**>