课程目标:

1. (重点)Springmvc实现异步(springmvc返回异步数据)

* 利用out对象输出异步数据
* 利用@ResponseBody注解返回json数据

1. 实现异步加载学生信息展示
2. 优化学生添加功能(采用异步方式加载年级信息)

一、(知识点：掌握)Springmvc实现异步技术:控制器返回异步数据

* 回顾Ajax的工作原理



XMLHttpRequest

* 控制器利用out对象输出返回数据

|  |
| --- |
| *//方式一:利用out对象输出返回的json数据* @RequestMapping(**"/getData"**) **public void** getData(HttpServletResponse response) **throws** IOException {  *//设置响应的编码* response.setContentType(**"text/html;charset=utf-8"**);  PrintWriter out=response.getWriter();  *//1.返回字符串  //out.print("hello");  //2.返回json* out.print(**"[\"张三\",\"李四\",\"麻子\"]"**); } |

* **利用@ResponseBody注解实现返回异步数据的步骤**

**3.1导入jackson的依赖**

|  |
| --- |
| *<!-- 添加jackson支持 start-->* <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-annotations</**artifactId**>  <**version**>2.9.0</**version**> </**dependency**> <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-core</**artifactId**>  <**version**>2.9.5</**version**> </**dependency**> <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-databind</**artifactId**>  <**version**>2.9.5</**version**> </**dependency**> *<!-- 添加jackson支持 end -->* |

* **修改springmvc配置文件添加mvc注解驱动**

<**mvc:annotation-driven**></**mvc:annotation-driven**>

**3.3 编写控制器利用@Reponsebody返回数据**

|  |
| --- |
| //方式二:使用@ResponseBody注解返回异步数据 //注意:@responsebody注解表示将返回的内容输出到网页、 //支持默认采用jackson工具包将java对象转化为json字符串返回 @RequestMapping(**"/getData2"**) @ResponseBody **public** List<Student> getData2(){  *//调用业务获取所有的学生信息* StudentService studentService=**new** StudentServiceImpl();  List<Student> list=studentService.getAllStudent();  *//将java对象转化为json进行输出* **return** list; } |

二、使用springmvc+mybatis+ajax+mysql技术实现学生信息的展示



* 编写获据所有学生的业务

略

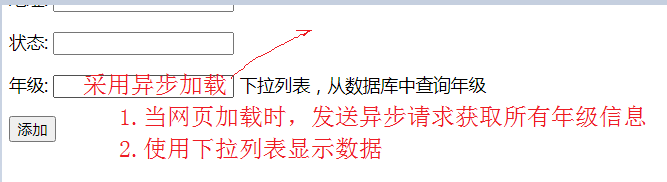
* 编写控制器(如何返回所有学生数据呢?)

|  |
| --- |
| *//方式二:使用@ResponseBody注解返回异步数据 //注意:@responsebody注解表示将返回的内容输出到网页、 //支持默认采用jackson工具包将java对象转化为json字符串返回* @RequestMapping(**"/getData2"**) @ResponseBody **public** List<Student> getData2(){  *//调用业务获取所有的学生信息* StudentService studentService=**new** StudentServiceImpl();  List<Student> list=studentService.getAllStudent();  *//将java对象转化为json进行输出* **return** list; } |

* 编写前端:使用jquery的ajax方法发送异步请求展示数据

|  |
| --- |
| <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %> <**html**> <**head**>  <**title**>Title</**title**>  <**script language="JavaScript" type="text/javascript" src="js/jquery-1.8.3.min.js"**></**script**>  <**script language="JavaScript" type="text/javascript"**>  *//发送异步请求获取学生信息展示* **$**(**function**(){ *//加载事件  //点击事件* **$**(**"#but1"**).click(**function**(){  *//发送异步请求获取数据展示* **$**.post(**"getData2"**,**null**,**function**(data){  *//以表格的方式显示数据* **var** table=**'<table border=2><tr><td>学号</td><td>姓名</td><td>年龄</td><td>性别</td></tr>'**;  **for**(**var** i=0;i<data.**length**;i++){  **var** obj=data[i];  table=table+**'<tr><td>'**+obj.**xh**+**'</td><td>'**+obj.**name**+**'</td><td>'**+obj.**age**+**'</td><td>'**+obj.sex+**'</td></tr>'**;  }  table=table+**"</table>"**;  *//显示* **$**(**"#showInfo"**).html(table);  },**"json"**);   });  });   </**script**> </**head**> <**body**>  <**input id="but1" type="button" name="ss" value="加载学生信息"**>  <**div id="showInfo"**></**div**> </**body**> </**html**> |

1. 优化添加功能（采用异步加载年级信息展示）



>>1编写服务器返回所有年级数据

>>业务

|  |
| --- |
| **public interface** GradeService {  *//查询所有的学生* List<Grade> getAllGrade(); }  **public class** GradeServiceImpl **implements** GradeService {  @Override  **public** List<Grade> getAllGrade() {  *//调用dao层* **try** {  SqlSession sqlSession=MyBatisUtil.*getSession*();  GradeMapper gradeMapper=sqlSession.getMapper(GradeMapper.**class**);  **return** gradeMapper.selectByExample(**new** GradeExample());  }**catch**(Exception e){  e.printStackTrace();  }**finally** {  MyBatisUtil.*closeSession*();  }  **return null**;  } } |

>>控制器

|  |
| --- |
| @Controller *//@RestController = @Controller+ @ResponseBody//所有控制器方法返回的都是异步数据* **public class** GradeController {  *//调用业务* GradeService **gradeService**=**new** GradeServiceImpl();   @RequestMapping(**"/getAllGrade"**)  @ResponseBody  **public** List<Grade> getAllGrade(){  List<Grade> list=**gradeService**.getAllGrade();  **return** list;  } } |

>>2 在页面中编写代码加载数据

|  |
| --- |
| <**script**>  **$**(**function**(){  *//发送异步请求获取数据* **$**.get(**"getAllGrade"**,**null**,**function**(data){  *//console.log(data);* **for**(**var** i=0;i<data.**length**;i++){  *//创建option节点* **var** option=**$**(**"<option value="**+data[i].**gid**+**">"**+data[i].gname+**"</option>"**);  *//追加节点* **$**(**"#gid"**).append(option);  }  },**"json"**);  }); </**script**>  <select id="gid" name="gid"></select> |

四、其它

* 解决springmvc静态资源访问



答:修改springmvc配置文件添加以下代码

*<!--springmvc配置文件中, 启用默认Servlet-->*<**mvc:default-servlet-handler**/>