

第三章 SpringMVC(四)

-AngularJS+BS+SpringMVC异步交互2018/11/14 [泽林.王峰]

授课目标

- 1、 上章回顾
- 2、 AngularJS框架介绍
- 3、 AngularJS基本用法
- 4、 SpringMVC+AngularJS+BootStrap完成学生管理系统

授课内容

1、 上章回顾

2、 AngularJS框架介绍

2.1 AngularJS简介

AngularJS 诞生于2009年，由Misko Hevery 等人创建，后为Google所收购。是一款优秀的前端JS框架，已经被用于Google的多款产品当中。AngularJS有着诸多特性，最为核心的是：MVC、模块化、自动化双向数据绑定、依赖注入等等。



2.2 AngularJS四大特征

2.2.1 MVC 模式

Angular遵循软件工程的MVC模式,并鼓励展现，数据，和逻辑组件之间的松耦合.通过依赖注入（ dependency injection ），Angular为客户端的Web应用带来了传统服务端的服务，例如独立于视图的控制。因此，后端减少了许多负担，产生了更轻的Web应用。



Model:数据,其实就是angular变量(\$scope.XX);

View: 数据的呈现,Html+Directive(指令);

Controller:操作数据,就是function,数据的增删改查;

2.2.2双向绑定

AngularJS是建立在这样的信念上的：即声明式编程应该用于构建用户界面以及编写软件构建，而指令式编程非常适合来表示业务逻辑。框架采用并扩展了传统HTML，通过双向的数据绑定来适应动态内容，双向的数据绑定允许模型和视图之间的自动同步。因此，AngularJS使得对DOM的操作不再重要并提升了可测试性。



2.2.3依赖注入

依赖注入(Dependency Injection,简称DI)是一种设计模式, 指某个对象依赖的其他对象无需手工创建，只需要“吼一嗓子”，则此对象在创建时，其依赖的对象由框架来自动创建并注入进来,其实就是最少知识法则;模块中所有的service和provider两类对象，都可以根据形参名称实现DI.

2.2.4模块化设计

高内聚低耦合法则

1)官方提供的模块 ng、ngRoute、ngAnimate

2)用户自定义的模块 angular.module('模块名',[])

3、 AngularJS基本用法

3.3入门小Demo

3.3.1 表达式

```
1 <html>
2 <head>
3   <title>入门小Demo-1</title>
4   <script src="angular.min.js"></script>
5 </head>
6 <body ng-app>
7   {{100+100}}
8 </body>
9 </html>
```

表达式的写法是{{表达式}} 表达式可以是变量或是运算式

ng-app 指令 作用是告诉子元素一下的指令是归angularjs的,angularjs会识别的

ng-app 指令定义了 AngularJS 应用程序的 根元素。

ng-app 指令在网页加载完毕时会自动引导（自动初始化）应用程序。

3.3.2 双向绑定

```
1 <html>
2 <head>
3   <title>入门小Demo-1 双向绑定</title>
4   <script src="angular.min.js"></script>
5 </head>
6 <body ng-app>
7   请输入你的姓名：<input ng-model="myname">
8   <br>
9   {{myname}},你好
10 </body>
11 </html>
```

ng-model 指令用于绑定变量,这样用户在文本框输入的内容会绑定到变量上，而表达式可以实时地输出变量。

1.3.3 初始化指令

我们如果希望有些变量具有初始值，可以使用ng-init指令来对变量初始化

```

1 <html>
2 <head>
3     <title>入门小Demo-3  初始化</title>
4     <script src="angular.min.js"></script>
5 </head>
6 <body ng-app    ng-init="myname='陈大海'">
7 请输入你的姓名：<input ng-model="myname">
8 <br>
9 {{myname}},你好
10 </body>
11 </html>

```

3.3.4 控制器

```

1 <html>
2 <head>
3     <title>入门小Demo-3  初始化</title>
4     <script src="angular.min.js"></script>
5     <script>
6         var app=angular.module('myApp',[]); //定义了一个叫myApp的模块
7         //定义控制器
8         app.controller('myController',function($scope){
9             $scope.add=function(){
10                 return parseInt($scope.x)+parseInt($scope.y);
11             }
12         });
13     </script>
14 </head>
15 <body ng-app="myApp" ng-controller="myController">
16 x:<input ng-model="x" >
17 y:<input ng-model="y" >
18 运算结果：{{add()}}
19 </body>
20 </html>

```

ng-controller用于指定所使用的控制器。

理解 \$scope :

*scope*的使用贯穿整个 *AngularJS* 应用,它与数据模型相关联,同时也是表达式执行的上下文.有了 *scope* 就在视图和控制器之间建立了一个通道,基于作用域视图在修改数据时会立刻更新 *scope*,同样的 *scope* 发生改变时也会立刻重新渲染视图.

3.3.5 事件指令

```

1 <html>
2 <head>
3     <title>入门小Demo-5  事件指令</title>
4     <script src="angular.min.js"></script>
5     <script>
6         var app=angular.module('myApp',[]); //定义了一个叫myApp的模块

```

```

7      //定义控制器
8      app.controller('myController',function($scope){
9          $scope.add=function(){
10             $scope.z= parseInt($scope.x)+parseInt($scope.y);
11         }
12     });
13     </script>
14 </head>
15 <body ng-app="myApp" ng-controller="myController">
16 x:<input ng-model="x" >
17 y:<input ng-model="y" >
18 <button ng-click="add()">运算</button>
19 结果: {{z}}
20 </body>
21 </html>

```

ng-click 是最常用的单击事件指令，再点击时触发控制器的某个方法

3.3.6 循环数组

```

1 <html>
2 <head>
3     <title>入门小Demo-6  循环数据</title>
4     <script src="angular.min.js"></script>
5     <script>
6         var app=angular.module('myApp',[]); //定义了一个叫myApp的模块
7         //定义控制器
8         app.controller('myController',function($scope){
9             $scope.list= [100,192,203,434 ]; //定义数组
10        });
11    </script>
12 </head>
13 <body ng-app="myApp" ng-controller="myController">
14 <table>
15 <tr ng-repeat="x in list">
16     <td>{{x}}</td>
17 </tr>
18 </table>
19 </body>
20 </html>

```

这里的ng-repeat指令用于循环数组变量。

3.3.7 循环对象数组

```

1 <html>
2 <head>
3     <title>入门小Demo-7  循环对象数组</title>
4     <script src="angular.min.js"></script>
5     <script>
6         var app=angular.module('myApp',[]); //定义了一个叫myApp的模块

```

```

7      //定义控制器
8      app.controller('myController',function($scope){
9          $scope.list= [
10             {name:'张三',shuxue:100,yuwen:93},
11             {name:'李四',shuxue:88,yuwen:87},
12             {name:'王五',shuxue:77,yuwen:56}
13         ];//定义数组
14     });
15     </script>
16 </head>
17 <body ng-app="myApp" ng-controller="myController">
18 <table>
19 <tr>
20     <td>姓名</td>
21     <td>数学</td>
22     <td>语文</td>
23 </tr>
24 <tr ng-repeat="entity in list">
25     <td>{{entity.name}}</td>
26     <td>{{entity.shuxue}}</td>
27     <td>{{entity.yuwen}}</td>
28 </tr>
29 </table>
30 </body>
31 </html>

```

3.3.8 内置服务

我们的数据一般都是从后端获取的，那么如何获取数据呢？我们一般使用内置服务\$http来实现。注意：以下代码需要在tomcat中运行。

```

1 <html>
2 <head>
3     <title>入门小Demo-8  内置服务</title>
4     <meta charset="utf-8" />
5     <script src="angular.min.js"></script>
6     <script>
7         var app=angular.module('myApp',[]); //定义了一个叫myApp的模块
8         //定义控制器
9         app.controller('myController',function($scope,$http){
10             $scope.findAll=function(){
11                 $http.get('data.json').success(
12                     function(response){
13                         $scope.list=response;
14                     }
15                 );
16             }
17         });
18     </script>
19 </head>
20 <body ng-app="myApp" ng-controller="myController" ng-init="findAll()">
21 <table>

```

```

22 <tr>
23     <td>姓名</td>
24     <td>数学</td>
25     <td>语文</td>
26 </tr>
27 <tr ng-repeat="entity in list">
28     <td>{{entity.name}}</td>
29     <td>{{entity.shuxue}}</td>
30     <td>{{entity.yuwen}}</td>
31 </tr>
32 </table>
33 </body>
34 </html>

```

建立文件 data.json

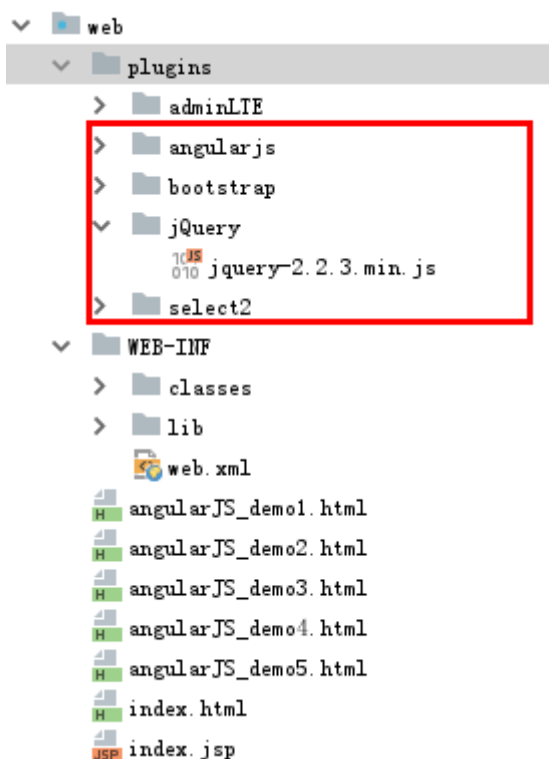
```

1 [
2     {"name": "张三", "shuxue": 100, "yuwen": 93},
3     {"name": "李四", "shuxue": 88, "yuwen": 87},
4     {"name": "王五", "shuxue": 77, "yuwen": 56},
5     {"name": "赵六", "shuxue": 67, "yuwen": 86}
6 ]

```

4、 SpringMVC+AngularJS+BootStrap完成学生管理系统

4.1)引入Angularjs及其相关的库：



4.2)定义index.html页面完成CRUD操作：

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>展示学生列表</title>
6     <link rel="stylesheet" href="plugins/bootstrap/css/bootstrap.min.css">
7     <style>
8         .table{
9             text-align: center;
10        }
11    </style>
12    <script src="plugins/jquery/jquery-2.2.3.min.js"></script>
13    <script src="plugins/bootstrap/js/bootstrap.min.js"></script>
14    <!--引入angularjs库-->
15    <script src="plugins/angularjs/angular.min.js"></script>
16    <script>
17        //1.定义模块
18        var app = angular.module("myApp",[]);
19        //2.在此模块下定义控制器
20        app.controller("myController",function($scope,$http){
21            $scope.reloadList=function(){
22                //1.得到所有的学生信息
23                $http.get("student/list.action").success(
24                    function(response){
25                        $scope.students = response;
26                    })
27                //2.得到所有的班级信息
28                $http.get("classes/list.action").success(
29                    function (response) {
30                        $scope.classes = response;
31                    }
32                )
33            }
34            //点击保存按钮进行添加或修改
35            $scope.save=function(){
36                //添加或修改的思路：如果$scope.entity.sid（表单隐藏域）有值，则执行修改，否则，执行添加
37                var sid = $scope.entity.sid;
38                var url = "student/add.action";
39                if(sid){
40                    url = "student/update.action"
41                }
42                $http.post(url,$scope.entity).success(
43                    function(response){
44                        if(response.success){ //如果添加成功，就刷新列表
45                            $scope.reloadList();
46                        }else{ //如果添加失败，就打印出错信息
47                            alert(response.message);
48                        }
49                        // 关闭对话框
50                        $('#myModal').modal("hide");
51                    }
52                )

```



```

53     }
54     //执行点击“修改”按钮操作
55     $scope.update=function(stud){
56         //1. 打开保存对话框
57         $("#myModal").modal("show");
58         //2.将得到的学生对象赋值给 entity对象
59         stud.birth = $scope.conterDate(stud.birth);
60         $scope.entity = stud;
61     }
62     //点击添加按钮
63     $scope.add=function(){
64         //1.将entity对象清空
65         $scope.entity = {};
66         //2.打开对话框
67         $("#myModal").modal("show");
68     }
69     //根据id号删除学生
70     $scope.delete=function(sid){
71         $http.get("student/delete.action?sid="+sid).success(
72             function (response) {
73                 if(response.success){ //如果删除成功，就刷新列表
74                     $scope.reloadList();
75                 }else{ //如果删除失败，就打印出错信息
76                     alert(response.message);
77                 }
78             }
79         )
80     }
81     //将得到的时间毫秒值转换为对应的日期
82     $scope.conterDate=function(time){
83         var dt = new Date(time);
84         return dt.getFullYear() + "-" + (dt.getMonth()+1) + "-" +
dt.getDate();
85     }
86 })
87 </script>
88 </head>
89 <body ng-app="myApp" ng-controller="myController" ng-init="reloadList()">
90 <div class="container">
91     <div class="panel panel-primary">
92         <div class="panel-heading">
93             <h4 class="glyphicon glyphicon-user">学生管理系统</h4>
94         </div>
95         <table class="table table-bordered table-hover">
96             <tr class="bg-info">
97                 <td>学号</td>
98                 <td>姓名</td>
99                 <td>性别</td>
100                <td>年龄</td>
101                <td>住址</td>
102                <td>生日</td>
103                <td>所在班级</td>
104                <td>操作</td>

```

```

105         </tr>
106
107         <tr ng-repeat="stud in students">
108             <td>{{stud.sid}}</td>
109             <td>{{stud.sname}}</td>
110             <td>{{stud.sex}}</td>
111             <td>{{stud.age}}</td>
112             <td>{{stud.addr}}</td>
113             <td>{{conterDate(stud.birth)}}</td>
114             <td>{{stud.classes.cname}}</td>
115             <td>
116                 <a href="#" class="btn btn-sm btn-success" ng-
click="update(stud)">修改</a>
117                 <a href="#" class="btn btn-sm btn-info" ng-
click="delete(stud.sid)">删除</a>
118             </td>
119         </tr>
120
121     </table>
122     <div class="panel-footer clearfix" style="text-align:right;padding:10px;">
123         <a href="javascript:;" class="btn btn-sm btn-warning pull-left glyphicon
glyphicon-plus-sign"
124             ng-click="add()">添加学生</a>
125         泽林信息版权所有 2000-2018.
126     </div>
127 </div>
128 </div>
129
130 <!-- 模态框（添加或修改学生） -->
131 <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-
labelledby="myModallabel">
132     <div class="modal-dialog" role="document">
133         <div class="modal-content">
134             <div class="modal-header">
135                 <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
136                 <h4 class="modal-title" id="myModallabel">编辑学生信息</h4>
137             </div>
138             <div class="modal-body">
139                 <form class="form-horizontal" id="form1">
140                     <input type="hidden" ng-model="entity.sid" id="sid">
141                     <div class="form-group">
142                         <label class="col-sm-2 control-label">学生姓名:</label>
143                         <div class="col-sm-10">
144                             <input type="text" class="form-control" placeholder="输
入姓名" ng-model="entity.sname" id='sname'>
145                         </div>
146                     </div>
147                     <div class="form-group">
148                         <label class="col-sm-2 control-label">性别:</label>
149                         <div class="radio">
150
151                             <label>

```

```

152         <input type="radio" ng-model="entity.sex"
value="男" >男
153     </label>
154     <label>
155         <input type="radio" ng-model="entity.sex"
value="女">女
156     </label>
157 </div>
158 </div>
159 <div class="form-group">
160     <label class="col-sm-2 control-label">学生年龄:</label>
161     <div class="col-sm-10">
162         <input type="text" class="form-control" placeholder="输
入年龄" ng-model="entity.age" id="age">
163     </div>
164 </div>
165 <div class="form-group">
166     <label class="col-sm-2 control-label">学生住址:</label>
167     <div class="col-sm-10">
168         <input type="text" class="form-control" placeholder="输
入住址" ng-model="entity.addr" id="addr">
169     </div>
170 </div>
171 <div class="form-group">
172     <label class="col-sm-2 control-label">生日:</label>
173     <div class="col-sm-10">
174         <input type="text" class="form-control" placeholder="输
入生日" ng-model="entity.birth" id="birth">
175     </div>
176 </div>
177 <div class="form-group">
178     <label class="col-sm-2 control-label">班级:</label>
179     <div class="col-sm-10">
180         <select class="form-control" ng-model="entity.cid"
id="cid">
181             <option ng-repeat="c in classes" value="{{c.cid}}">
{{c.cname}}</option>
182         </select>
183     </div>
184 </div>
185
186 </form>
187 </div>
188 <div class="modal-footer">
189     <button type="button" class="btn btn-default" data-dismiss="modal">
关闭</button>
190     <button type="button" class="btn btn-primary" id="btn_save" ng-
click="save()">保存</button>
191 </div>
192 </div>
193 </div>
194 </div>
195 </body>

```

```
196 </html>
197
```

4.3)定义StudentController.java对象完成CRUD操作：

```
1  package com.zelin.web.controller;
2
3  import com.zelin.pojo.Student;
4  import com.zelin.service.StudentService;
5  import com.zelin.utils.Result;
6  import com.zelin.web.exception.MyException;
7  import org.springframework.beans.factory.annotation.Autowired;
8  import org.springframework.stereotype.Controller;
9  import org.springframework.util.StringUtils;
10 import org.springframework.web.bind.annotation.PathVariable;
11 import org.springframework.web.bind.annotation.RequestBody;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.ResponseBody;
14
15 import javax.servlet.ServletException;
16 import javax.servlet.http.HttpServletRequest;
17 import javax.servlet.http.HttpServletResponse;
18 import javax.servlet.http.HttpSession;
19 import java.io.IOException;
20 import java.util.List;
21
22 @Controller
23 @RequestMapping("/student")
24 public class StudentController {
25     @Autowired
26     private StudentService studentService;
27
28     //查询所有的学生
29     @RequestMapping("/list")
30     @ResponseBody
31     public List<Student> findAll() {
32         return studentService.findAll();
33     }
34
35     //添加学生
36     @RequestMapping("/add")
37     @ResponseBody
38     public Result add(@RequestBody Student student) {
39         try {
40             studentService.add(student);
41             return new Result(true, "添加成功");
42         } catch (Exception e) {
43             e.printStackTrace();
44             return new Result(false, "添加失败");
45         }
46     }
47     //修改学生
```

```

48     @RequestMapping("/update")
49     @ResponseBody
50     public Result update(@RequestBody Student student) {
51         try {
52             studentService.update(student);
53             return new Result(true, "修改成功");
54         } catch (Exception e) {
55             e.printStackTrace();
56             return new Result(false, "修改失败");
57         }
58     }
59     //删除学生
60     @RequestMapping("/delete")
61     @ResponseBody
62     public Result delete(String sid) {
63         try {
64             studentService.delete(sid);
65             return new Result(true, "删除成功");
66         } catch (Exception e) {
67             e.printStackTrace();
68             return new Result(false, "删除失败");
69         }
70     }
71 }
72

```

4.4)运行效果如下：

<div>  学生管理系统 <div> 按 F11 即可退出全屏模式 </div> </div>							
学号	姓名	性别	年龄	住址	生日	所在班级	操作
1	张小三	男	25	广州123	2018-9-6	1302班	修改 删除
2	王小五	男	28	广州	2010-10-27	1301班	修改 删除
3	李小红	女	190	杭州	2008-8-21	1302班	修改 删除
4	王二小	男	12	岳阳	2010-2-16	1302班	修改 删除
5	张小二	男	28	上海123	2018-10-30	1303班	修改 删除
6	黄家驹	男	54	香港	1995-3-12	1303班	修改 删除
7	罗成	男	22	邵阳	1997-1-18	1301班	修改 删除
8	魏征	男	28	洛ab阳	2000-3-16	1301班	修改 删除
9	徐达	男	29	江苏无锡	2010-2-23	1302班	修改 删除
13	孙宾	男	35	上海浦东	1987-6-23	1302班	修改 删除
15	赵本山	男	65	东北大街	1954-11-17	1301班	修改 删除
16	宋小宝	男	54	东北	1963-7-18	1302班	修改 删除

编辑学生信息

学生姓名:

李强

性别:

☒ 男 ☐ 女

学生年龄:

23

学生住址:

香港九龙

生日:

1995-3-12

班级:

1302班

关闭

保存

5、分页查询

5.1) 定义前台分页相关的配置信息：

```
1 <script>
2     //1.定义模块
3     //1.1) 如果要分页，需要引入pagination模块
4     var app = angular.module("myApp",["pagination"]);
5     //2.在此模块下定义控制器
6     app.controller("myController",function($scope,$http){
7         //分页第三步：引入分页的配置对象属性
8         /*
9         currentPage：当前页码
10        totalItems:总条数
11        itemsPerPage:每页的记录数
12        perPageOptions：页码选项
13        onChange：更改页面时触发事件 */
14        $scope.paginationConf={
15            currentPage : 1,    //代表当前页
16            itemsPerPage : 5,    //代表每页的记录数
17            totalItems : 100,    //代表总记录数
18            perPageOptions : [5,10,15,20,25,30], //代表分页的选项列表
19            onChange:function(){ //窗体一加载就执行
20                $scope.loadList($scope.paginationConf.currentPage,
21                    $scope.paginationConf.itemsPerPage);
22            }
23        }
```

```

24      //定义loadList ( 进行分页 )
25      $scope.loadList=function(page,pageSize){
26          $http.get("student/listpage.action?
page="+page+"&pageSize="+pageSize).success(
27              function (response) {
28                  $scope.paginationConf.totalItems=response.total;
29                  $scope.students=response.rows;
30              }
31          )
32      }
33      $scope.reloadList=function(){
34          //1.得到所有的学生信息
35          $http.get("student/list.action").success(
36              function(response){
37                  $scope.students = response;
38              })
39          //2.得到所有的班级信息
40          $http.get("classes/list.action").success(
41              function (response) {
42                  $scope.classes = response;
43              }
44          )
45      }
46      //点击保存按钮进行添加或修改
47      $scope.save=function(){
48          //添加或修改的思路：如果$scope.entity.sid ( 表单隐藏域 ) 有值，则执行修改，否
则，执行添加
49          var sid = $scope.entity.sid;
50          var url = "student/add.action";
51          if(sid){
52              url = "student/update.action"
53          }
54          $http.post(url,$scope.entity).success(
55              function(response){
56                  if(response.success){ //如果添加成功，就刷新列表
57                      $scope.reloadList();
58                  }else{ //如果添加失败，就打印出错信息
59                      alert(response.message);
60                  }
61                  // 关闭对话框
62                  $('#myModal').modal("hide");
63              }
64          )
65      }
66      //执行点击“修改”按钮操作
67      $scope.update=function(stud){
68          //1.打开保存对话框
69          $('#myModal').modal("show");
70          //2.将得到的学生对象赋值给 entity对象
71          stud.birth = $scope.conterDate(stud.birth);
72          $scope.entity = stud;
73      }
74      //点击添加按钮

```

```

75     $scope.add=function(){
76         //1.将entity对象清空
77         $scope.entity = {};
78         //2.打开对话框
79         $("#myModal").modal("show");
80     }
81     //根据id号删除学生
82     $scope.delete=function(sid){
83         $http.get("student/delete.action?sid="+sid).success(
84             function (response) {
85                 if(response.success){    //如果删除成功，就刷新列表
86                     $scope.reloadList();
87                 }else{                  //如果删除失败，就打印出错信息
88                     alert(response.message);
89                 }
90             }
91         )
92     }
93     //将得到的时间毫秒值转换为对应的日期
94     $scope.conterDate=function(time){
95         var dt = new Date(time);
96         return dt.getFullYear() + "-" + (dt.getMonth()+1) + "-" +
dt.getDate();
97     }
98 })
99 </script>

```

5.2) 定义前台分页导航信息：

```

1 <tfoot>
2     <tr>
3         <td colspan="8">
4             <!--2、 定义分页导航 -->
5             <tm-pagination conf="paginationConf"></tm-pagination>
6         </td>
7     </tr>
8 </tfoot>

```

5.3) 定义后台控制器相关：

```

1 //进行分页
2 @RequestMapping("/listpage")
3 @ResponseBody
4 public PageResult<Student> listPage(int page,int pageSize){
5     return studentService.findByPage(page,pageSize);
6 }

```

5.4) 定义服务层相关：


```

1  @Override
2      public PageResult<Student> findByPage(int page, int pageSize) {
3          //1.开始分页
4          PageHelper.startPage(page,pageSize);
5          //2.进行分页查询
6          StudentExample example = new StudentExample();
7          Page<Student> students = (Page<Student>)
studentMapper.selectByExample(example);
8          //3.构造分页的PageResult对象
9          return new PageResult<>(students.getTotal(),students.getResult());
10     }

```

5.5) 运行效果如下：

学生管理系统							
学号	姓名	性别	年龄	住址	生日	所在班级	操作
1	张小三	男	25	广州123	2018-9-6	1302班	修改 删除
2	王小五	男	28	广州	2010-10-27	1301班	修改 删除
3	李小红	女	190	杭州	2008-8-21	1302班	修改 删除
4	王二小	男	12	岳阳	2010-2-16	1302班	修改 删除
5	张小二	男	28	上海123	2018-10-30	1303班	修改 删除

[«](#)
[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[»](#)
第 1 页 每页 5 条 /共21条

[添加学生](#)
泽林信息版权所有 2000-2018.