

原 SSM+BootStrap实现增删查改及批量删除操作

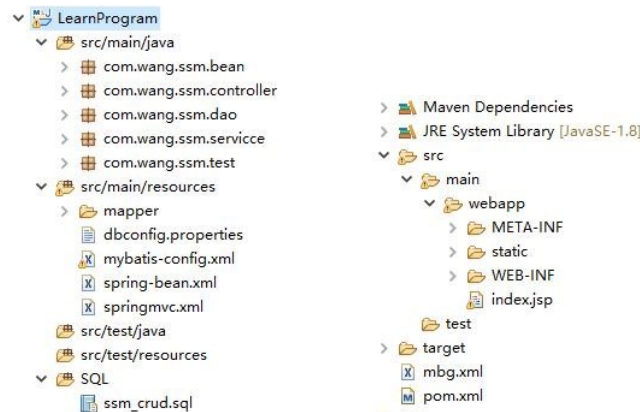
2018年06月05日 20:10:46 阅读数：391 标签：SSM PageHelper CRUD 更多

版权声明：本文为博主原创文章，未经博主允许不得转载。 [https://blog.csdn.net/weixin\\_36279318/article/details/80586493](https://blog.csdn.net/weixin_36279318/article/details/80586493)

## （一）项目结构

注：下载项目源码

注：SSM增删查改源码视频



## SSM演示

添加 删除						
<input type="checkbox"/>	编号	姓名	性别	邮箱	所属部门	操作
<input type="checkbox"/>	201201	张三	男	323458734@qq.com	测试部	编辑  删除
<input type="checkbox"/>	201202	李思	女	2132123@qq.com	业务部	编辑  删除
<input type="checkbox"/>	201203	凯耐	男	23333333@sina.com	测试部	编辑  删除
<input type="checkbox"/>	201204	曹军	男	3262839@qq.com	后勤部	编辑  删除
<input type="checkbox"/>	201205	张薰	女	21878392@qq.com	业务部	编辑  删除

当前第1页，总202页，总1009条记录

### 1.使用Ajax和JQuery实现表格添加数据及分页功能

前端源码：

```
1  //-----1.实现分页功能-----//
2      var totalRecord,currentPageNum;
3      //1.页面加载完成以后，直接发送ajax请求，获取分页数据
4      $(function() {
5          toPage(1);
6      });
7      //跳转页码
8      function toPage(pn) {
9          $.ajax({
10              url : "${path}/emps",
11              data : "pn="+pn,
12              type : "GET",
13              success : function(result) { //回调函数
```

```

14         //console.log(result);
15         //1.解析并显示员工数据
16         build_emps_table(result);
17         //2.解析并显示分页数据
18         build_page_info(result);
19         //3.解析显示分页条数据
20         build_page_nav(result)
21     }
22     });
23 }
24 //2.解析Json并显示员工数据
25 function build_emps_table(result) {
26     //清空table表格
27     $("#emps_table tbody").empty();
28     //解析Json获取员工列表
29     var emps = result.extend.pageInfo.list;
30     $.each(emps, function(index, item) {
31         //alert(item.empName)
32         //把员工数据添加到单元格中
33         var checkBoxTd = $("<td><input type='checkbox' class='check_item' /></td>");
34         var empIdTd = $("<td></td>").append(item.empId);
35         var empNameTd = $("<td></td>").append(item.empName);
36         var genderTd = $("<td></td>").append(
37             item.gender == 'M' ? "男" : "女");
38         var emailTd = $("<td></td>").append(item.email);
39         var deptNameTd = $("<td></td>")
40             .append(item.department.deptName);
41
42         //editbtn和delbtn表示按钮元素及其类容
43         var editbtn = $("<button></button>")
44             .addClass("btn btn-primary btn-sm editbtn")
45             .append("<span></span>")
46             .addClass("glyphicon glyphicon-pencil")
47             .append("编辑")
48             .attr("editid", item.empId);
49
50         var delbtn = $("<button></button>")
51             .addClass("btn btn-danger btn-sm delbtn")
52             .append("<span></span>")
53             .addClass("glyphicon glyphicon-trash")
54             .append("删除")
55             .attr("delid", item.empId); //自定义属性和获取id值
56
57         var operate = $("<td></td>").append(editbtn).append(" ")
58             .append(delbtn);
59         //把解析的Json数据放到表格中
60         //append方法执行完以后继续返回原来的元素
61         //把每个单元格中的数据添加为一行数据，并把它添加到table中
62         $("<tr></tr>").append(checkBoxTd)
63             .append(empIdTd)
64             .append(empNameTd)
65             .append(genderTd)
66             .append(emailTd)
67             .append(deptNameTd)
68             .append(operate)
69             .appendTo("#emps_table tbody");
70     });
71 }
72 //3.解析显示分页信息
73 function build_page_info(result) {
74     $("#page_info_area").empty();
75     //定位到page_info_area元素并把数据添加到此处
76     $("#page_info_area").append(
77         "当前第" + result.extend.pageInfo.pageNum + "页，总"
78         + result.extend.pageInfo.pages + "页，总"
79         + result.extend.pageInfo.total + "条记录");
80     totalRecord=result.extend.pageInfo.total;
81     currentPageNum=result.extend.pageInfo.pageNum;
82 }
83 //4.解析显示分页导航数据
84 function build_page_nav(result) {
85     $("#page_nav_area").empty();
86     //创建ul class="pagination"元素
87     var ul = $("<ul></ul>").addClass("pagination");

```



0



```

88      //创建首页, 前页, 下一页, 末页元素
89      var firstPage = $("- </li>").append($("<a></a>")).append("首页");
90      var prePage = $("- </li>").append($("<a></a>")).append("&lquo;");
91      if (result.extend.pageInfo.hasPreviousPage == false) {
92          firstPage.addClass("disabled");
93          prePage.addClass("disabled");
94      } else {
95          firstPage.click(function() {
96              toPage(1);
97          });
98          prePage.click(function() {
99              toPage(result.extend.pageInfo.pageNum - 1);
100          });
101      }
102
103      var nextPage = $("- </li>")
104          .append($("<a></a>")).append("&raquo;");
105      var lastPage = $("- </li>").append($("<a></a>")).append("末页");
106      if (result.extend.pageInfo.hasNextPage == false) {
107          //不能点击
108          lastPage.addClass("disabled");
109          nextPage.addClass("disabled");
110      } else {
111          lastPage.click(function() {
112              toPage(result.extend.pageInfo.pages);
113          });
114          nextPage.click(function() {
115              toPage(result.extend.pageInfo.pageNum + 1);
116          });
117      }
118      //将首页, 前页li元素添加到ul元素中
119      ul.append(firstPage).append(prePage);
120      //获取要显示的所有页码参数
121      var nums = result.extend.pageInfo.navigatepageNums;
122      $.each(nums, function(index, item) {
123          var num = $("- </li>").append($("<a></a>")).append(item);
124          if (result.extend.pageInfo.pageNum == item)
125              //添加活动标识
126              num.addClass("active");
127          num.click(function() {
128              toPage(item);
129          });
130          ul.append(num);
131      });
132      //添加下一页和末页提示
133      ul.append(nextPage).append(lastPage);
134      $("<nav></nav>").append(ul).appendTo("#page_nav_area");
135  }

```

后台实现分页源码:

```

1  /*
2      * 使用@ResponseBody需要jacksonjar包
3      */
4  @RequestMapping("/emps")
5  @ResponseBody
6  //1. 查询所有员工数据
7  public Msg getEmpsWithJson(@RequestParam(value="pn",defaultValue="1")Integer pn,Model model) {
8      //1. 使用PageHelper分页插件,在查询之前只需传入页码及每页显示的数据记录参数即可!
9      PageHelper.startPage(pn, 5);
10     //2. startPage后紧跟的就是一个分页查询
11     List<Employee> emps=employeeService.getAll();
12     //3. 使用PageInfo封装查询结果, 只需将PageInfo交给页面即可
13     //4. PageInfo封装分页的详细信息
14     //5. PageInfo构造器的5表示连续显示的页码数字
15     PageInfo<Employee> page=new PageInfo<Employee>(emps,5);
16     //6. 处理结果以Json数据格式返回给浏览器
17     return Msg.success().add("pageInfo",page);
18 }
19

```

后台接收分页栏的请求参数源码:

```

1 //4. 查询员工数据
2 @RequestMapping(value="/emp/{id}",method=RequestMethod.GET)
3 @ResponseBody
4 public Msg getEmp(@PathVariable("id")Integer id){
5     Employee employee=employeeService.getEmp(id);
6     return Msg.success().add("emp", employee);
7 }

```

## 2.Ajax实现添加员工数据功能

效果图：

前端源码

```

1 //-----2.实现添加员工数据功能-----//
2 //重置表单数据
3 function resetForm(ele) {
4     $(ele)[0].reset();
5     $(ele).find("*").removeClass("has-error has-success");
6     $(ele).find(".help-block").text("");
7 }
8 //向服务器发送请求获取部门数据
9 function getDepts(ele) {
10     $(ele).empty();
11     $.ajax({
12         url : "${path}/depts",
13         data : "",
14         type : "POST",
15         success : function(result) { //result代表Json数据
16             //构建下拉列表
17             $.each(result.extend.depts, function() { //this表示当前遍历对象
18                 $("</option>").append(this.deptName).attr("value", this.deptId)
19                     .appendTo(ele);
20             });
21         });
22     }
23 });
24 }
25 //校验表单输入的用户名是否重复，当输入数据时立刻发送ajax请求
26 $("#empName_input").change(
27     function() {
28         //获取当前输入框的值
29         var empName = this.value;
30         $.ajax({
31             url : "${path}/checkUser",
32             data : "empName=" + empName,
33             type : "POST",
34             success : function(result) {
35                 //响应码100为可用
36                 if (result.code == 100) {
37                     showValidateMsg("#empName_input", "success", "用户名可以使用");
38                     $("#save_emp_btn").attr("save", "success");
39                 } else if (result.code == 200) {
40                     showValidateMsg("#empName_input", "error",

```

```

41         result.extend.error);
42         $("#save_emp_btn").attr("save", "error");
43     }
44 }
45 })
46 });
47 //添加用户
48 $("#emp_add_btn").click(function() {
49     //弹出模态框前查出部门信息显示在下拉列表
50     getDepts("#deptName_input");
51     //清空表单数据
52     resetForm("#emp_form");
53     //返回添加模态框
54     $('#empAddModal').modal({
55         backdrop : "static"
56     });
57 });
58 //显示校验信息
59 function showValidateMsg(ele, status, msg) {
60     $(ele).parent().removeClass("has-error has-success");
61     $(ele).next("span").text("");
62     if (status == "error") {
63         $(ele).parent().addClass("has-error has-feedback");
64         $(ele).next("span").text(msg);
65     } else {
66         $(ele).parent().addClass("has-success has-feedback");
67         $(ele).next("span").text(msg);
68     }
69 }
70 //校验表单提交数据
71 function validateEmpForm() {
72     var flag = true;
73     //1.获取员工名字并校验
74     var empName = $("#empName_input").val();
75     var regName = /^[a-zA-Z0-9_]{6,16}$|(^[\u2E80-\u9FFF]{2,5})$/;
76     if (!regName.test(empName)) {
77         showValidateMsg("#empName_input", "error", "用户名格式不正确");
78         flag = false;
79     } else {
80         //showValidateMsg("#empName_input", "success", "用户名格式正确");
81     }
82     //2.获取email并校验
83     var email = $("#email_input").val();
84     var regEmail = /^[a-z0-9_\.-]+@([a-z-]+\.)+([a-z]{2,6})$/;
85     if (!regEmail.test(email)) {
86         showValidateMsg("#email_input", "error", "Email格式不正确");
87         flag = false;
88     } else {
89         //showValidateMsg("#email_input", "success", "格式正确");
90     }
91     return flag;
92 }
93 //点击保存按钮
94 $("#save_emp_btn").click(function(){
95     //1.模态框中填写的表单数据提交给服务器保存
96     //2.校验提交表单数据
97     //3.发送ajax请求保存表单数据
98     if (!validateEmpForm())
99         return false;
100     //判断之前ajax判断是否成功! , 如果失败无法向数据库插入数据
101     if ($(this).attr("save") == "error")
102         return false;
103     $.ajax({
104         url: "${path}/emp",
105         type: "POST",
106         data: $("#emp_form").serialize(),
107         success: function(result){
108             //alert(result.msg);
109             //1.员工保存成功: 首先关闭模态框, 其次来到最后一页显示刚保存的数据
110             $('#empAddModal').modal("hide");
111             //保存数据后来到最后一页
112             toPage(totalRecord);
113         }
114     })

```



0



```
115         });
        });
```

后台获取部门数据源码:

```
1  @RequestMapping("/depts")
2  @ResponseBody
3  //获取部门数据
4  public Msg getDeptsWithJson(){
5      List<Department> list=departmentService.getAll();
6      //System.out.println(list.size());
7      return Msg.success().add("depts", list);
8  }
```

后台校验在输入框输入的数据是否合法源码:

```
1  //3.后台校验
2  @RequestMapping(value="/checkUser")
3  @ResponseBody
4  public Msg checkUser(@RequestParam("empName")String empName){
5      //重复校验一遍姓名
6      String regex="^[a-zA-Z0-9_-]{6,16}$|(^[\u2E80-\u9FFF]{2,5}$)";
7      if(!empName.matches(regex)){
8          return Msg.fail().add("error", "用户名格式不正确! ");
9      }
10     //true: 表示可以 false: 表示不可以
11     boolean flag=employeeService.checkUser(empName);
12     if(flag){
13         return Msg.success();
14     }else{
15         return Msg.fail().add("error", "用户名重复, 请输入新的用户名! ");
16     }
17 }
```

后台保存员工数据源码:

```
1  //2.1.JSR303校验表单数据,后保存数据
2  @RequestMapping(value="/emp",method=RequestMethod.POST)
3  @ResponseBody
4  public Msg saveEmps(@Validated Employee employee,BindingResult result){
5      //从前端传来的参数springmvc会自动封装
6      //System.out.println(employee);
7      if(result.hasErrors()){
8          Map<String, String>map=new HashMap<>();
9          List<FieldError> list=result.getFieldErrors();
10         for (FieldError fieldError : list) {
11             map.put(fieldError.getField(), fieldError.getDefaultMessage());
12         }
13         return Msg.fail().add("errorFields", map);
14     }else{
15         employeeService.saveEmp(employee);
16         return Msg.success();
17     }
18 }
```

### 3.Ajax实现员工修改操作

效果如下：



The image shows a modal window titled "员工修改" (Employee Modification). It contains a form with the following fields: "姓名" (Name) with the value "李思", "性别" (Gender) with radio buttons for "男" (Male) and "女" (Female), "邮箱" (Email) with the value "2132123@qq.com", and "所属部门" (Department) with a dropdown menu showing "业务部". At the bottom right of the modal are two buttons: "关闭" (Close) and "修改" (Modify). A watermark URL "https://blog.csdn.net/weixin\_30723118" is visible at the bottom of the modal.

前端源码：

```
1  //-----3.实现修改员工数据功能-----//
2      //绑定编辑按钮（editbtn表示按钮的标识，而不是按钮元素）
3      $(document).on("click", ".editbtn", function(){
4          //1.获取部门数据，填写到id="deptName_update_input"的下拉框
5          getDepts("#deptName_update_input");
6          //2.重置表单
7          resetForm("#emp_update_form");
8          //获取员工id属性
9          getEmp($(this).attr("editid"));
10         //将editid添加到下面元素中
11         $("#update_emp_btn").attr("editid", $(this).attr("editid"));
12         //3.弹出模态框
13         $('#empUpdateModal').modal({
14             backdrop : "static"
15         });
16     });
17     //ajax向服务器发送请求
18     function getEmp(id){
19         $.ajax({
20             url: "${path}/emp/"+id,
21             type: "GET",
22             success: function(result){
23                 var emp=result.extend.emp;
24                 //alert(emp.gender)
25                 $("#empName_update_input").text(emp.empName);
26                 $("#email_update_input").attr("value", emp.email);
27                 //将查询出来的数据填充在id选择器中，实现radio被选中
28                 $("#emp_update_form input[name=gender]").val([emp.gender]);
29                 $("#emp_update_form select").val([emp.dId]);
30             }
31         })
32     }
33
34     //绑定更新按钮，也需要校验
35     $("#update_emp_btn").click(function(){
36
37         var email = $("#email_update_input").val();
38         var regEmail = /^[a-z0-9_\. -]+@([a-z\.-]+)\.([a-z\.-]{2,6})$/;
39         if (!regEmail.test(email)) {
40             showValidateMsg("#email_update_input", "error", "Email格式不正确");
41             return false;
42         } else {
43             showValidateMsg("#email_update_input", "success", "格式正确");
44         }
45
46         $.ajax({
47             url: "${path}/emp/"+$(this).attr("editid"),
48             data: $("#emp_update_form").serialize(), //+"&_method=PUT",
49             type: "PUT",
```

```
50         success:function(result){
51
52             $("#empUpdateModal").modal("hide");
53             toPage(currentPageNum);
54         }
55     })
56 });
```



后台获取部门数据展示在前端页面：

```
1  @ResponseBody
2  //获取部门数据
3  public Msg getDeptsWithJson(){
4      List<Department> list=departmentService.getAll();
5      //System.out.println(list.size());
6      return Msg.success().add("depts", list);
7  }
```

后台获取员工数据展示在前端页面

```
1  @RequestMapping(value="/emp/{id}",method=RequestMethod.GET)
2  @ResponseBody
3  public Msg getEmp(@PathVariable("id")Integer id){
4      Employee employee=employeeService.getEmp(id);
5      return Msg.success().add("emp", employee);
6  }
```

后台接收修改请求源码：

```
1  @ResponseBody
2  //empId是Employee的属性，否则无法绑定数据
3  @RequestMapping(value="/emp/{empId}",method=RequestMethod.PUT)
4  public Msg updateEmp(Employee employee){
5      employeeService.updateEmp(employee);
6      return Msg.success();
7  }
```

#### 4.删除单条员工数据记录及批量删除员工

效果如下：

### SSM演示

<input type="checkbox"/>	添加	删除					
<input type="checkbox"/>	编号	姓名	性别	邮箱	所属部门	操作	
<input type="checkbox"/>	201201	张三	男	323458734@qq.com	测试部	<input type="button" value="编辑"/>	<input type="button" value="删除"/>
<input checked="" type="checkbox"/>	201202	李思	女	2132123@qq.com	业务部	<input type="button" value="编辑"/>	<input type="button" value="删除"/>
<input type="checkbox"/>	201203	凯耐	男	23333333@sina.com	测试部	<input type="button" value="编辑"/>	<input type="button" value="删除"/>
<input type="checkbox"/>	201204	曹军	男	3262839@qq.com	后勤部	<input type="button" value="编辑"/>	<input type="button" value="删除"/>
<input checked="" type="checkbox"/>	201205	张潇	女	21878392@qq.com	业务部	<input type="button" value="编辑"/>	<input type="button" value="删除"/>

当前第1页，总202页，总1009条记录

首页 1 2 3 4 5 末页

前端源码

```
1  //-----4.实现删除某一条员工记录-----//
2      //绑定删除按钮
3      $(document).on("click", ".delbtn", function(){
4          //获取员工姓名
5          //var empName=$(this).parents().find("tr td:eq(1)").text();
6          var empId=$(this).attr("delid");
```



```

7         if(confirm("确认要删除吗? ")){
8             $.ajax({
9                 url:"${path}/emp/"+empId,
10                type:"DELETE",
11                success:function(result){
12                    alert("删除成功!");
13                    toPage(currentPageNum);
14                }
15            })
16        }
17    });
18
19    //全选按钮
20    $("#check_all").click(function(){
21        //attr()获取checked市undefined,全部选中显示true
22        //alert($("#this").prop("checked"));
23
24        //1.prop修改和读取dom原生属性值
25        $(".check_item").prop("checked",$("#this").prop("checked"));
26    });
27
28    //绑定事件
29    $(document).on("click", ".check_item",function(){
30        //判断选中的和所有要选的个数相等, 显示全部被选中
31        var flag=$("#check_item:checked").length==($(".check_item").length);
32        $("#check_all").prop("checked",flag);
33    });
34
35    //批量删除操作
36    $("#emp_del_btn").click(function(){
37        alert("确定要删除吗? ");
38        var ids="";
39        $.each($(".check_item:checked"),function(){
40            ids+=$(this).parents("tr").find("td:eq(1)").text()+"-";
41        });
42        //http://localhost:8083/ssm-crud/emp/201236-201237-201238-201239-201240
43        ids=ids.substring(0,ids.length-1);
44        $.ajax({
45            url:"${path}/emp/"+ids,
46            type:"DELETE",
47            success:function(result){
48                if(result.code==100){
49                    alert("删除成功!");
50                    toPage(currentPageNum);
51                }
52            }
53        });
54    });

```

后台源码:

```

1  @ResponseBody
2  @RequestMapping(value="/emp/{ids}",method=RequestMethod.DELETE)
3  public Msg deleteEmp(@PathVariable("ids")String ids){
4      //URL:http://localhost:8083/ssm-crud/emp/201236-201237-201238-201239-201240
5      //如果URL中包含"- "就执行批量删除
6      if(ids.contains("- ")){
7          List<Integer> list=new ArrayList<>();
8          String[]strs=ids.split("- ");
9          for (String id : strs) {
10              list.add(Integer.parseInt(id));
11          }
12          employeeService.bathDeleteEmps(list);
13      }else{
14          employeeService.deleteEmp(Integer.parseInt(ids));
15      }
16      return Msg.success();
17  }

```

## 5.Mabatis逆向工程生成POJO和Mapper.xml文件

参考: [MyBatis GeneratorXML Configuration File Reference](#)

mbg.xml文件

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE generatorConfiguration
3      PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4      "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5
6  <generatorConfiguration>
7
8      <context id="mysqlTables" targetRuntime="MyBatis3">
9          <commentGenerator>
10             <property name="suppressAllComments" value="true" />
11          </commentGenerator>
12          <!-- 1.配置数据库连接 -->
13          <jdbcConnection driverClass="com.mysql.jdbc.Driver"
14              connectionURL="jdbc:mysql://localhost:3306/ssm_crud" userId="root"
15              password="root">
16          </jdbcConnection>
17
18          <javaTypeResolver>
19              <property name="forceBigDecimals" value="false" />
20          </javaTypeResolver>
21          <!--2.生成的Bean存放的位置 -->
22          <javaModelGenerator targetPackage="com.wang.ssm.bean"
23              targetProject=".\\src\\main\\java">
24              <property name="enableSubPackages" value="true" />
25              <property name="trimStrings" value="true" />
26          </javaModelGenerator>
27          <!--3.自动生成sql映射文件存放的位置 -->
28          <sqlMapGenerator targetPackage="mapper" targetProject=".\\src\\main\\resources">
29              <property name="enableSubPackages" value="true" />
30          </sqlMapGenerator>
31          <!--4.生成dao接口存放的位置 -->
32          <javaClientGenerator type="XMLMAPPER"
33              targetPackage="com.wang.ssm.dao" targetProject=".\\src\\main\\java">
34              <property name="enableSubPackages" value="true" />
35          </javaClientGenerator>
36
37          <!-- 表对应的model -->
38          <table tableName="tbl_emp" domainObjectName="Employee"></table>
39          <table tableName="tbl_dept" domainObjectName="Department"></table>
40
41      </context>
42  </generatorConfiguration>
```

既生瑜何生亮！两位90天才股神2018操盘疯狂吸金！

大庆商贸 · 顶新

想对作者说点什么？

## bootstrapTable--4.删除和批量删除

一条记录可以看做一条数据的数组 1    Html 1.1  批量选择框  1.2  单个删除  2    bootStarp 2.1  批量获得  获得选择的数据 //批量删除 fu...



### Maven+SSM+Mysql+Bootstrap整合实现管理员登录、用户增删改查

2018年05月21日    995KB



### 你不知道区块链开发都是这样练成的？

区块链DApp开发学习路线图，月薪4万很轻松

## 使用bootstrap实现批量删除

&amp;lt;table class=&quot;table table-striped table-bordered table-hover&quot; id=&quot;users&quot;...&lt;/table>

## checkbox批量删除功能

今天来实现一个批量删除的功能：批量删除的功能呢，首先前台页面可以想到使用checkbox来做，这里后台数据库其实有两种删除的方式，如下：

## bootstrap实现删除和批量删除操作

bootstrap实现删除和批量删除操作原创2017-09-22 01374 一条记录可以看做一条数据的数组1 Html1.1 批量选择框 1.2 单个删除2 bootStar...

## jQuery 操作列表批量删除数据

方式一： //判断全选复选框是否选中(全选和取消全选) \$( "#allcbx").bind("click",function(){ var ck...

## 老中医说：男人多吃这个东西，时间延长5倍！

番当生物 · 顶新



## 数据批量删除/同时删除多条数据

2009年10月13日 1.88MB 下载

## BootstrapTable 实现toolbar删除操作

说明：操作前必须导入jquery.js、bootstrap.js和bootstrapTable.min.js、DIV数据： ...

## idea+maven+ssm+redis+bootstarp实现简单的高并发秒杀API

涉及到的知识点 数据库 1.表设计 2.SQL技巧 3.事务和行级锁 MyBatis 1.DAO层设计与开发 2.MyBatis合理应用 3.MyBatis与Spring整合 Spr...

## 使用ssm框架进行整合，实现一个小的项目，对员工信息的增删改查

一、项目简介 项目基于Maven+SpringMVC+Spring+MyBatis+Bootstrap的组合，快速开发一个完整的CRUD功能，对员工数据进行增删查改。项目效...

## <em>批量删除</em>文件.bat

下载 2018年08月

<em>批量删除</em>某文件下txt文件的前几行【请先备份原文件且和处理文件放在一起】 立即...spring mvc+mybatis+mysql+maven+<em>bootstrap</e

## 基于bootstrapTalbe查询，删除/批量删除，排序，搜索

1.导入css， js&amp;lt;link href=&quot;../css/bootstrap.min.css?v=3.3.6&quot; rel=&quot;stylesheet&quot;&lt;/link>...&lt;/p>
</div>
<div data-bbox="207 534 427 547" data-label="Section-Header">
<h2>文件BOM<em>批量删除</em>工具</h2>
</div>
<div data-bbox="821 536 919 548" data-label="Text">
<p>下载 2018年08月</p>
</div>
<div data-bbox="207 553 919 565" data-label="Text">
<p>文件BOM(byte order mark)<em>批量删除</em>工具支持utf-8 utf-16 utf-32... 文件BOM(byte order mark)<em>批量删除</em>工具支持utf...spring mvi</p>
</div>
<div data-bbox="281 579 552 594" data-label="Section-Header">
<h2>老中医说：多吃这个东西，时间延长5倍</h2>
</div>
<div data-bbox="281 601 337 612" data-label="Text">
<p>中科 · 顶新</p>
</div>
<div data-bbox="207 624 420 638" data-label="Section-Header">
<h2>BootstrapTable如何批量操作多行</h2>
</div>
<div data-bbox="207 641 919 652" data-label="Text">
<p>1、在BootstrapTable初始化表格时，在表格columns加上checkbox: true,这样第一列就是一个复选框。2、通过jquery操作//获得选中行var checkedbo</p>
</div>
<div data-bbox="207 665 473 679" data-label="Section-Header">
<h2>【bootstrap】--选中一行删除（页面操作）</h2>
</div>
<div data-bbox="207 682 919 693" data-label="Text">
<p>一开始做javaTOO，上来就给我一个我从来没有接触过的bootstrap页面，让它从不可用弄成既可以增删改表格，又有input框验证，然后将页面所有信息</p>
</div>
<div data-bbox="232 707 260 732" data-label="Image">
<img alt="RAR file icon" data-bbox="232 707 260 732" />
</div>
<div data-bbox="285 706 475 719" data-label="Section-Header">
<h2>单链表c语言实现增删改查操作</h2>
</div>
<div data-bbox="285 722 476 733" data-label="Text">
<p>2012年09月21日 1.97MB 下载</p>
</div>
<div data-bbox="207 750 417 764" data-label="Section-Header">
<h2>vs信息系统中完成增删改查的操作</h2>
</div>
<div data-bbox="207 767 632 779" data-label="Text">
<pre>namespace 学生信息 { public partial class Form1 : Form { public Form1() ...</pre>
</div>
<div data-bbox="207 792 504 806" data-label="Section-Header">
<h2>SSM 整合实现 增删改查、PageHelper 实现分页</h2>
</div>
<div data-bbox="207 809 919 820" data-label="Text">
<p>一、前言1、想法这个SSM（Spring+SpringMVC+MyBatis，基于Maven）实现的 CRUD 测试是之前写的，一直留着想等有空再使用点 BootStrap / Ea</p>
</div>
<div data-bbox="207 833 282 863" data-label="Image">
<img alt="Library Management System icon" data-bbox="207 833 282 863" />
</div>
<div data-bbox="285 833 374 847" data-label="Section-Header">
<h2>图书管理系统</h2>
</div>
<div data-bbox="285 853 328 865" data-label="Text">
<p>百度广告</p>
</div>
<div data-bbox="207 877 373 892" data-label="Section-Header">
<h2>MyBatis基本增删改查操作</h2>
</div>
<div data-bbox="207 894 919 906" data-label="Text">
<p>本文内容主要介绍单条记录的增删改查操作，MyBatis提供了很多完成单条记录的增删改查操作的API。本例主要讲述 UserMapper org.apache.ibatis.se</p>
</div>
</html>



## ssm-crud 用ssm+bootstrap+ajax实现增删改查

2017年04月29日 412KB 下载



0



### 个人资料



凯耐

关注

原创 178 粉丝 59 喜欢 19 评论 80

等级: 博客 5 访问: 7万+

积分: 2542 排名: 1万+

勋章: 恒



### 最新文章

第二篇: Spring Boot整合JPA、事务处理及AOP的使用

第一篇: 第一个Spring Boot应用程序

Git基本命令的使用

IntelliJ IDEA破解及入门学习快捷键

2018Java面试题 (Java基础部分)

### 个人分类

Java项目源码 2篇  
JDBC 10篇  
Hibernate 24篇  
Spring 5篇  
.java技术 11篇

展开

### 热门文章

Java 中的JButton按钮事件

阅读量: 6300

Python for循环的使用(三)

阅读量: 5885

Java实现简单的医院管理系统源码

阅读量: 3182

在Hibernate中使用@lob修饰大数据类型的属性

阅读量: 2495

Python 将JSON格式文件进行数据可视化制作世界人口地图(十一)

阅读量: 2149

### 最新评论

Java web项目01 停车管理...

weixin\_36279318: [reply]weixin\_42584977[/reply]  
链接: https://pan.bai...

Java web项目01 停车管理...

weixin\_42584977: [reply]weixin\_42584977[/reply]  
哦哦哦我看见了那个百度云分享 不麻烦...

Java web项目01 停车管理...

weixin\_42584977: 博主 可以求一份源码学习一下  
吗 感谢! 928578276@qq.com

Java web项目01 停车管理...

qq\_42391574: [reply]weixin\_36279318[/reply] 非  
常感谢!

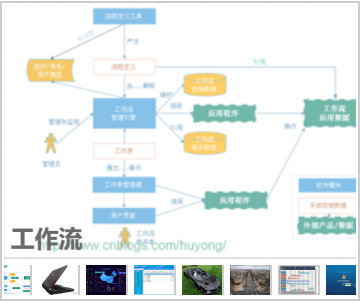
Java web项目01 停车管理...

weixin\_36279318: [reply]qq\_42391574[/reply] 链  
接: https://pan.baidu.c...

归档

2018年8月	2篇
2018年7月	2篇
2018年6月	3篇
2018年5月	3篇
2018年4月	13篇

展开



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 网站地图  
©2018 CSDN版权所有 京ICP证09002463号  
🔍 百度提供搜索支持

经营性网站备案信息  
网络110报警服务  
中国互联网举报中心  
北京互联网违法和不良信息举报中心

