

1、架构：maven+spring4+struts2+hibernate5

mysql+c3p0+fastjson+[shiro](#)加密框架

2、html+css+js+EasyUI+ajax

3、导出导入POI类库

4、fastjson对数据的处理

@JSONField(serialize=false) //请求时在转json格式数据不对该字段不处理

private String pwd;//密码

主要应用方向：

一个密码类 不可见的~~数据不解析不处理~~

两张表双向关联引用，查询时要去其中某个表进行~~不处理~~

// 多对一 该注解在此的作用：由于销售表和详情表 是双向关联

// 没有这个注解会导致对象循环引用 使内存益出

@JSONField(serialize=false)

private Salesmanagement sales; // 多个销售详情对应一个销售订单

5、ajax传递数据的封装处理类型

第一种：解析form表单提交的数据

```
var formData = $("#editForm").serializeJSON();
```

```
data:formData, //ajax发送解析的数据到服务器
```

第二种：组装数据与表单一起发送给服务器

```
data:$.extend(formdata,{"add":"adduser"}),
```

第三种：通过参数传递组装

```
var add = "addcommod"; 参数
```

```
data:$.extend(formdata,{"add":add}),
```

第四种：通过ID获取对应的值组装数据传给后端

```
<input id="oldPass"/>
```

```
var $oldpass = $('#oldPass');//获取input的值
```

```
data:{"newPwd":$newpass.val(),"oldPwd":$oldpass.val()},
```

第五种：解析表单拼接后一起传递

```
var formData = $("#editForm").serializeJSON(); 处理解析from表单
```

```
var $sum = $('#sum');//通过ID获取输入的值
```

```
//将form表单的数据与id获取的数据组装备到formData里
```

```
formData['t.totalAmount']=$sum.val(); formData =  
{“nameid”:1, “name”:“xxx”,“t.totalAmount”: “254元”
```

```
var rowsData = $("#addOrdersGrid").datagrid('getRows');//获取解析  
datagrid的所有数据
```

```
var orderdetailsJSON = JSON.stringify(rowsData); //将获取到的数据转为  
JSON
```

```
formData['orderdetailsJSON']=orderdetailsJSON; //再次组装在formData  
里去
```

```
data:formData,
```

6、datagrid的动态添加行，可编辑行

动态添加行 appendRow

```
$("#addOrdersGrid").datagrid('appendRow',  
{ 'number':0, 'money':0.00, 'totalMon':0.00});
```

获取index 是datagrid的最后一行 getRows获取datagrid表格中的所有行数据
[{}, {}]

```
isEditingRowIndex = $("#addOrdersGrid").datagrid('getRows').length-  
1;
```

7、多表查询分组求合

```
select gt.name, sum(od.money) from Goodstype gt,  
goods g,  
orderdetail od,  
orders o  
where gt.uuid=g.goodstypeuuid  
and g.uuid=od.goodsuuid and od.ordersuuid=o.uuid  
and o.type='2' group by gt.name
```

结果 多表查询 按分组 求合

```
public class ReportDao extends HibernateDaoSupport implements IReportDao {  
  
    @Override  
    public List orderReport() {  
  
        String hql="select gt.name,sum(od.money) from"  
            + " Goodstype gt,Goods g,Orderdetail od,Orders o"  
            + " where gt=g.goodstype"  
            + " and g.uuid=od.goodsuuid and od.orders=o"  
            + " and o.type='2' group by gt.name";  
        return this.getHibernateTemplate().find(hql);  
    }  
}
```

```
@Override  
public List orderReport() {  
    String hql="select new Map( gt.name as name, sum(od.money) as y ) from"  
        + " Goodstype gt,Goods g,Orderdetail od,Orders o"  
        + " where gt=g.goodstype"  
        + " and g.uuid=od.goodsuuid and od.orders=o"  
        + " and o.type='2' group by gt.name";  
    return this.getHibernateTemplate().find(hql);  
}
```

将查询结果保存到一个Map对象中

这个查询返回本来就是一个list

```

@Override
public List orderReport(Date date1, Date date2) {
    String hql="select new Map( gt.name as name, sum(od.money) as y ) from"
        + " Goodstype gt, Goods g, Orderdetail od, Orders o"
        + " where gt=g.goodstype"
        + " and g.uuid=od.goodsuuid and od.orders=o"
        + " and o.type='2' and o.createtime>=? and o.createtime<=? "
        + " group by gt.name";
    return this.getHibernateTemplate().find(hql, date1, date2);
}

```

按时间范围查询

8、统计图表

HighChart **js** **css** 纯前端

JFreeChart 生成为图片模式的图表

9、发送邮件技术

javamail 手动发送邮件

org.springframework.mail.javamail.JavaMailSender;

org.springframework.mail.javamail.MimeMessageHelper;

Quartz定时发送邮件的框架

Cron 表达式是一个字符串,字符串以 5 或 6 个空格隔开,分为 6 或 7 个域,每一个域代表一个含义。

1. 域

Seconds (秒):可出现", - * /"四个字符,有效范围为 0-59 的整数

Minutes (分钟):可出现", - * /"四个字符,有效范围为 0-59 的整数

Hours (小时):可出现", - * /"四个字符,有效范围为 0-23 的整数

DayofMonth (日 of 月):可出现", - * / ? L W C"八个字符,有效范围为 1-31 的整数

Month (月):可出现", - * /"四个字符,有效范围为 1-12 的整数

DayofWeek (日 of 星期):可出现", - * / ? L C #"八个字符,有效范围为 1-7 的整数

1 表示星期天, 2 表示星期一, 依次类推

Year (年):可出现", - * /"四个字符,有效范围为 1970-2099 年

10、远程调用技术 CXF框架

webservice技术: 是跨语言跨平台 内部采用xml方式交互数据

一、WebService 与 CXF 框架简介

(一) 什么是 WebService

WebService 也称为 web 服务。是通过 SOAP 协议实现的远程调用技术。

常见的远程调用技术：

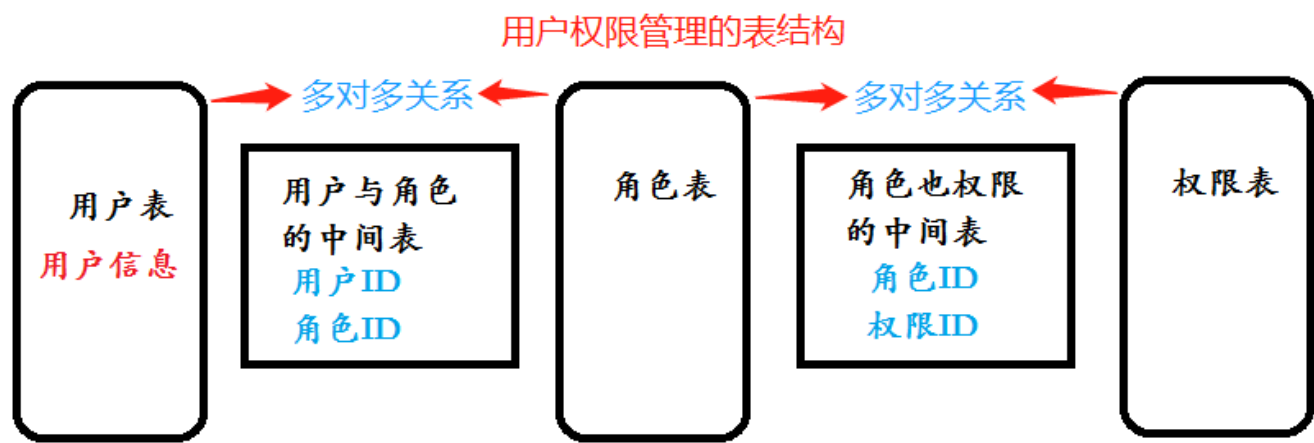
- 1. Socket 套接字 TCP/IP UDP
- 2. webservice :
- 3. http 调用 。
- 4. RMI(远程方法调用) Hessian 框架

webService 的特点：

- 1、可以跨平台，跨语言。
- 2、W3C(万维网联盟)制定的标准。
- 3、可以穿透防火墙（因为 soap 协议是基于 HTTP 协议）。

SOAP 协议(简单对象访问协议)： HTTP + XML

11、用户与权限管理及根据当前登录人显示不同内容



shiro框架：认证，授权，会话，加密

使用：

- 1、引入JAR
- 2、在web.xml里添加过滤器（过滤器的filter-name要和配置文件ID保持一致，这个过滤器一定要在struts2过滤器的前面配置）

```
<filter>

<filter-name>shiroFilter</filter-name>
```

```
<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-  
class>  
  
</filter>  
  
<filter-mapping>  
  
  <filter-name>shiroFilter</filter-name>  
  
  <url-pattern>*.action</url-pattern>  
  
  <url-pattern>*.html</url-pattern>  
  
  <url-pattern>*</url-pattern>  
  
</filter-mapping>
```

3、配置文件的导入 见附件

4、认证的方法：

授权的方法：

