cmc summary:

实现模块：

实时地采集服务器中业务情况，以XML格式文件进行网络传输，约定一个报文头，根据报文头双方知道请求和返回的都需要是什么，在客户端以JS来解析XML，用HighCharts来显示。

实时采集的实现是在J2EE工程的web.xml 中注册一个监听器，监听Session的启动和销毁，目的是当有用户在使用这个系统时，才使用扫描线程通过Socket周期性地去扫描服务器的状态，并将每个服务器的连接状态信息存储到数据库当中。

所用技术：

J2EE( Listener, Thread,Socket), Spring MVC, JS, JQuery, Highcharts ,XML

团队合作比一个人闷头搞效率高而且心情好。

大声说出你的想法的同时也是在促进自己思考，是否这个是严谨的，别人是否可以批驳，如果是，则又改进的空间了。

// js dispose xml

if (typeof xmlString == "string") { //先判定是否是xml

if (document.implementation.createDocument) {

var parser = new DOMParser();

xmlDoc = parser.parseFromString(xmlString, "text/xml");

} else if (window.ActiveXObject) { //支持低版本IE

xmlDoc = new ActiveXObject("Microsoft.XMLDOM");

xmlDoc.async = false;

xmlDoc.loadXML(xmlString);

}

}else{

xmlDoc=xmlString;

}

var elementList = xmlDoc.getElementsByTagName('body'); //解析xml文件

//将XML数据组装成js中需要的数据结构

if(elementList!=null && elementList[0].hasChildNodes){

//alert("开始解析");

var childs = elementList[0].childNodes;

for(var i=0;i<childs.length;i++){

if(childs[i].childNodes[0].nodeValue==null||childs[i].childNodes[0].nodeValue=="")

continue ;

var id=task\_num[childs[i].nodeName];

var info=childs[i].childNodes[0].nodeValue.split('v');

for(var j=0;j<3;j++){

task\_account[id][j]=parseInt(info[j]);

}

}

}

//JS auto run at a fixed period

int eventId=window.setInterval(“myFunction()”,1000);

window.clearInterval(eventId);

setTimeOut(function (data){…},1000);// only run one time after the specific time

jquery part:

$.ajax({

type:”post”,

[url:”/cmc/kms/maink/create\_mk](url:)”,

data:{type=1,code=2,format=”json”,rand=Math.random()},

dataType:”json”,

type:”POST”,

success: function(data){

data=eval(data);

}

});

4点注意：

1. $.ajax({}); 括号里面还有大括号！！
2. url 是相对于localhost:8080 而不是此工程
3. dataType: json代表的是预期返回的格式, format=”json”代表的是request的格式
4. eval(data) 不是必须的

HTML5 audio 用法：

<audio id=*"player"* controls=*"controls"* src=*"music/*

<s:property value=*'#request.currentMusic.path'*/>*"*

autoplay=*"autoplay"* style="width:*86%*;float:*left*;"

>

</audio>

//如果播放完成，则自动下一曲

**var** player=document.getElementById("player");

player.addEventListener('ended', **function** () {

//alert("播放完成");

$("#nextForm").submit();

}, **false**);

遍历data

$.each(data,function(){

alert(this.id)

}

$(“input[type=’checkbox’][name=’add’]”).serialize();//获得所有选中的checkbox

String values[]=request.getParameterValues("add");//可以获得所有同名的数组

1. 用request.getParameter(String str)  
   <input type="file" name="fileName">  
   后台用String fileName=request.getParameter("fileName")获取，试试  
   2.直接获取服务器上的内容  
   FileItemFactory factory = new DiskFileItemFactory();  
   ServletFileUpload upload = new ServletFileUpload(factory);  
   List<FileItem> list = upload.parseRequest(request);

获取Spring框架的ApplicationContext对象的方法：

**简单实用型： 实现ApplicationAware接口，**并在配置文件中注册一下这个bean,这样在加载完Spring框架之后，这个Application对象就会注入到这中间来。

**import** org.springframework.beans.BeansException;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.ApplicationContextAware;

**public** **class** ApplicationContextHelper **implements** ApplicationContextAware {

**private** **static** ApplicationContext *context*;

**public** **void** setApplicationContext(ApplicationContext arg0)

**throws** BeansException {//这个是一个覆盖函数

**this**.*context*=arg0;

}

**public** **static** ApplicationContext getContext(){

**return** *context*;

}

}

jQuery选择多个元素：

$("input[type='button'][name="+id+"]").attr("disabled","true");

$("#chatBox").hide(); $("#chatBox").show(); $("#chatBox").slideToggle(500); //切换

js confirm() //对话框

js 新建一个时间**new** Date()；没有格式化

jQuery 版

$.formatDate("yyyy-MM-dd hh:mm:ss",new Date());错误！！！

正确的JS写法：

var data = new Date();    
        var year = data.getFullYear();  //获取年  
        var month = data.getMonth() + 1;    //获取月  
        var day = data.getDate(); //获取日  
        var hours = data.getHours();   
        var minutes = data.getMinutes();  
        time = year + "年" + month + "月" + day + "日" + " " + hours + ":" + minutes;

$.trim(“asdf”);// 去掉空格

var pattern=new RegExp(“^[0-9]\*$”);//验证数字，不用在前后加反斜杠，不识别

pattern.test(“1233”);// true

验证数字的[正则表达式](http://zhidao.baidu.com/search?word=%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F&fr=qb_search_exp&ie=utf8" \t "_blank)集   
验证数字：^[0-9]\*$  
验证n位的数字：^\d{n}$  
验证至少n位数字：^\d{n,}$  
验证m-n位的数字：^\d{m,n}$  
验证零和非零开头的数字：^(0|[1-9][0-9]\*)$  
验证有[两位小数](http://zhidao.baidu.com/search?word=%E4%B8%A4%E4%BD%8D%E5%B0%8F%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank)的[正实数](http://zhidao.baidu.com/search?word=%E6%AD%A3%E5%AE%9E%E6%95%B0&fr=qb_search_exp&ie=utf8)：^[0-9]+(.[0-9]{2})?$  
验证有1-3位小数的[正实数](http://zhidao.baidu.com/search?word=%E6%AD%A3%E5%AE%9E%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank)：^[0-9]+(.[0-9]{1,3})?$  
验证非零的正整数：^\+?[1-9][0-9]\*$  
验证非零的[负整数](http://zhidao.baidu.com/search?word=%E8%B4%9F%E6%95%B4%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank)：^\-[1-9][0-9]\*$  
验证[非负整数](http://zhidao.baidu.com/search?word=%E9%9D%9E%E8%B4%9F%E6%95%B4%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank)（正整数 + 0） ^\d+$  
验证[非正整数](http://zhidao.baidu.com/search?word=%E9%9D%9E%E6%AD%A3%E6%95%B4%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank)（[负整数](http://zhidao.baidu.com/search?word=%E8%B4%9F%E6%95%B4%E6%95%B0&fr=qb_search_exp&ie=utf8) + 0） ^((-\d+)|(0+))$  
验证长度为3的字符：^.{3}$  
验证由26个英文字母组成的字符串：^[A-Za-z]+$  
验证由26个大写英文字母组成的字符串：^[A-Z]+$  
验证由26个小写英文字母组成的字符串：^[a-z]+$  
验证由数字和26个英文字母组成的字符串：^[A-Za-z0-9]+$  
验证由数字、26个英文字母或者[下划线](http://zhidao.baidu.com/search?word=%E4%B8%8B%E5%88%92%E7%BA%BF&fr=qb_search_exp&ie=utf8" \t "_blank)组成的字符串：^\w+$  
验证用户密码:^[a-zA-Z]\w{5,17}$ 正确格式为：以字母开头，长度在6-18之间，只能包含字符、数字和[下划线](http://zhidao.baidu.com/search?word=%E4%B8%8B%E5%88%92%E7%BA%BF&fr=qb_search_exp&ie=utf8" \t "_blank)。  
验证是否含有 ^%&',;=?$\" 等字符：[^%&',;=?$\x22]+  
验证汉字：^[\u4e00-\u9fa5],{0,}$  
验证Email地址：^\w+[-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$  
验证InternetURL：^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]\*)?$ ；^[a-zA-z]+://(w+(-w+)\*)(.(w+(-w+)\*))\*(?S\*)?$  
验证电话号码：^(\(\d{3,4}\)|\d{3,4}-)?\d{7,8}$：--正确格式为：XXXX-XXXXXXX，XXXX-XXXXXXXX，XXX-XXXXXXX，XXX-XXXXXXXX，XXXXXXX，XXXXXXXX。  
验证身份证号（15位或18位数字）：^\d{15}|\d{}18$  
验证一年的12个月：^(0?[1-9]|1[0-2])$ 正确格式为：“01”-“09”和“1”“12”  
验证一个月的31天：^((0?[1-9])|((1|2)[0-9])|30|31)$ 正确格式为：01、09和1、31。  
整数：^-?\d+$  
非负[浮点数](http://zhidao.baidu.com/search?word=%E6%B5%AE%E7%82%B9%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank)（正[浮点数](http://zhidao.baidu.com/search?word=%E6%B5%AE%E7%82%B9%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank) + 0）：^\d+(\.\d+)?$  
正[浮点数](http://zhidao.baidu.com/search?word=%E6%B5%AE%E7%82%B9%E6%95%B0&fr=qb_search_exp&ie=utf8" \t "_blank) ^(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*))$  
非正[浮点](http://zhidao.baidu.com/search?word=%E6%B5%AE%E7%82%B9&fr=qb_search_exp&ie=utf8" \t "_blank)数（负[浮点](http://zhidao.baidu.com/search?word=%E6%B5%AE%E7%82%B9&fr=qb_search_exp&ie=utf8" \t "_blank)数 + 0） ^((-\d+(\.\d+)?)|(0+(\.0+)?))$  
负[浮点](http://zhidao.baidu.com/search?word=%E6%B5%AE%E7%82%B9&fr=qb_search_exp&ie=utf8" \t "_blank)数 ^(-(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*)))$  
浮点数 ^(-?\d+)(\.\d+)?

JDBC中的ResultSet API没有直接获取记录条数的方法，现介绍几个：   
方法一：利用ResultSet的getRow方法来获得ResultSet的总行数   
Java代码

ResultSet rs;     
rs.last(); //移到最后一行     
int rowCount = rs.getRow(); //得到当前行号，也就是记录数     
rs.beforeFirst(); //如果还要用结果集，就把指针再移到初始化的位置

方法二：利用循环ResultSet的元素来获得ResultSet的总行数   
Java代码

ResultSet rs;     
int rowCount = 0;      
while(rset.next())      
{      
    rowCount++;      
}

ResultSet rs; int rowCount = 0; while(rset.next()) { rowCount++; }

方法三：利用sql语句中的count函数获得ResultSet的总行数   
Java代码

String sql = "select count(\*) record\_ from ( select \* from yourtable t where t.column\_ = 'value\_' )";     
ResultSet rs = ps.executeQuery(sql);      
int rowCount = 0;      
if(rs.next())      
{      
    rowCount=rs.getInt(1); //column 1 is the count     
}

Properties

Properties prop = **new** Properties();

InputStream in = DBHandler.**class**.getResourceAsStream("/db.properties");

**// 使用properties读数据，路径在WEB-INF/classes目录下**

**//因为类的部署路径在WEB-INF/classes，所以"/" 指的是WEB-INF/classes**

String URL="";

String Username="";

String Password="";

**try** {

prop.load(in);

Username=prop.getProperty("username");

Password=prop.getProperty("password");

URL=prop.getProperty("url");

System.*out*.print(Username+"\n"+Password+"\n"+URL);

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

 用惯了access mssql server的朋友,可能在用mysql查询前N条记录时,习惯的使用select top n 形式的语句,在这里说明一下,mysql没有此语法,mysql用limit来实现相关功能，而且功能更加强大，GOOD。以下是limit在mysql中的使用详解：   
语法：  
SELECT \* FROM table LIMIT [offset,] rows | rows OFFSET offset  
LIMIT 子句可以被用于强制 SELECT 语句返回指定的记录数。LIMIT 接受一个或两个数字参数。参数必须是一个整数常量。  
如果给定两个参数，第一个参数指定第一个返回记录行的偏移量，第二个参数指定返回记录行的最大数目。  
初始记录行的偏移量是 0(而不是 1)： 为了与 PostgreSQL 兼容，MySQL 也支持句法： LIMIT # OFFSET #。  
mysql> SELECT \* FROM table LIMIT 5,10; // 检索记录行 6-15 ,注意，10为偏移量   
//为了检索从某一个偏移量到记录集的结束所有的记录行，可以指定第二个参数为 -1：  
mysql> SELECT \* FROM table LIMIT 95,-1; // 检索记录行 96-last.  
//如果只给定一个参数，它表示返回最大的记录行数目：  
mysql> SELECT \* FROM table LIMIT 5; //检索前 5 个记录行 //也就是说，LIMIT n 等价于 LIMIT 0,n。  
如果你想得到最后几条数据可以多加个 order by id desc

mysql不支持select top n的语法，应该用这个替换：  
select \* from tablename order by orderfield desc/asc limit position， counter；  
position 指示从哪里开始查询，如果是0则是从头开始，counter 表示查询的个数  
取前15条记录：  
select \* from tablename order by orderfield desc/asc limit 0,15

**时间类型：**

resultSet.getTimestamp(“time”);//当数据库中定义的是Timestamp类型的时候，不能使用getString() 或者是 getDate() 来获取，这会造成精度丢失！！！

**Timestamp转化为String:**  
SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//定义格式，不显示毫秒  
Timestamp now = new Timestamp(System.currentTimeMillis());//获取系统当前时间  
String str = df.format(now);  
   
 **String转化为Timestamp:**  
SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
String time = df.format(new Date());  
Timestamp ts = Timestamp.valueOf(time);

java.lang.Exception: DEBUG -- CLOSE BY CLIENT STACK TRACE

at com.mchange.v2.c3p0.impl.NewPooledConnection.close(NewPooledConnection.java:566)

at com.mchange.v2.c3p0.impl.NewPooledConnection.close(NewPooledConnection.java:234)

at com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool$1PooledConnectionResourcePoolManager.destroyResource(C3P0PooledConnectionPool.java:470)

at com.mchange.v2.resourcepool.BasicResourcePool$1DestroyResourceTask.run(BasicResourcePool.java:964)

at com.mchange.v2.async.ThreadPoolAsynchronousRunner$PoolThread.run(ThreadPoolAsynchronousRunner.java:547)

C3P0的错误，可以更改日志等级或者去掉destroy-method=”close”

但如果是在配置文件中配的应该把preferredTestQuery设置为true

1. jetty 插件下载和整个服务器下载 是不一样的
2. jetty eclipse 适配器只有6.0的，只适合servlet <=2.5 的web 项目
3. sae 只支持jetty, css/image/js/fonts 访问路径有问题，得是http:// xxx
4. mysql 参数最好最好配置在配置文件中，否则改一次就得重新导出war，痛苦
5. jetty出现无法找到org/apache/tomcat/PeriodicEventListener 版本兼容性问题新建一个工程，复制代码
6. jetty 的工作目录不识别空格！！！！
7. sae 不能包含他已经给你的包特别是：servlet.jar，上传之前删掉它
8. 上传War包之前一定要把数据库的连接配置好！可以写死

## Eclipse下报Target runtime Apache Tomcat v6.0 is not defined.错误的解决方法

Eclipse导入工程后工程上显示一个小红叉，但工程里没有文件错误，也没有语法错误，百思不得其解啊，后来在网上找了一些资料说是项目引用的类库路径有问题。【项目】->【右键】->【build path】->【configure build path】->【libraries】，查看一下引用的类库路径。网上都说是因为这里引用错误引起的，但是我在项目导入的时候做的第一件事情就是修改这里的library，因此不是这个原因。

在problems中显示错误是：Target runtime Apache Tomcat 6.0 is not defined. 在网上查了一下终于找到解决方法。方法是：在工程目录下的.settings文件夹里，打开org.eclipse.wst.common.project.facet.core.xml文件，其内容是：

<?xmlversion="1.0"encoding="UTF-8"?>        
<faceted-project>        
<runtimename="Apache Tomcat v6.0"/>        
<fixedfacet="jst.web"/>        
<fixedfacet="jst.java"/>        
<installedfacet="jst.java"version="6.0"/>        
<installedfacet="jst.web"version="2.5"/>        
<installedfacet="wst.jsdt.web"version="1.0"/>        
</faceted-project>

将中间那堆东西去掉，修改为：

<?xml version="1.0" encoding="UTF-8" ?> <faceted-project></faceted-project>

即不指定运行环境

PS:进行上面的修改以后虽然没有错误了，但是想发布网站demo的时候发现没有【run as】->【run on server】这个选项了。这时候我们创建一个demo2的web project，然后查看.settings下的org.eclipse.wst.common.project.facet.core.xml，我们发现其内容是：

<?xml version="1.0" encoding="UTF-8" ?>

<faceted-project>

<runtime name="Apache Tomcat v6.0.29" />

<fixed facet="jst.web" />   <fixed facet="java" />

<fixed facet="wst.jsdt.web" />

<installed facet="java" version="1.6" />

<installed facet="jst.web" version="2.5" />

<installed facet="wst.jsdt.web" version="1.0" />

  </faceted-project>

那么我们只需要将该代码拷贝到网站demo下的.settings目录下，将原来的org.eclipse.wst.common.project.facet.core.xml覆盖即可。覆盖以后我们在Eclipse就能够使用【run as】->【run on server】来发布demo了。

这是设置运行环境时生成的文件，如果不修改，可能不兼容，最简单的办法就是不指定运行环境，那就没有那么多烦人的事了~~

        随着网络通信、多媒体及数据存储等技术的飞速发展,音乐网站作为一种大众的web服务,给喜爱音乐的网络用户提供了极大的便利。但是，仅仅提供音乐信息并不能满足用户的情感需要,市面上各种音乐网站，罗列繁杂的音乐信息，并不能提供惬意的欣赏音乐的环境，而且鲜有社交功能.     我们的Melody音乐社交平台以“以乐会友”为宗旨，旨在为音乐发烧友打造一片天地。首先，在社交方面侧重于让用户实时地表达自己的心境，并且和心境相同的人人实时交流沟通；其二、我们平台会对用户收藏和分享的音乐类型进行分析，为用户推荐他可能喜欢的音乐；其三、用户可以制作歌单并分享，吸引志同道合的小伙伴聆听；其四，我们的界面风格美观大方，提供或小清新或深沉或优雅或热烈的皮肤，给用户提供舒适的体验。

    我们已经将网站部署到新浪云上，想要立即体验此网站可以点击：[http://dlutmelody.sinaapp.com](http://dlutmelody.sinaapp.com/)  用户名：tom,密码123。由于新浪云会把代码和数据库分布到很多地方，所以程序中有些ajax访问无法到达出现302错误，因此此网站的体验效果不如本地，不如人意之处还望大家见谅。

    要想更好地体验此网站，此处提供了作品源文件、部署文件等 [点击下载](http://lnsjsjds.synu.edu.cn/forum/common/upload/2014/04/27/204710n8.zip)

Uncaught TypeError: Cannot use 'in' operator to search for '6721' in <html> <head> <title>Struts Problem Report</title> <style> pre { margin: 0; padding: 0; } </style> </head> <body> <h2>Struts Problem Report</h2> <p> Struts has detected an unhandled exception: </p> <div id="exception-info"> <table> <tr> <td><strong>Messages</strong>:</td> <td> <li>org.melody.controller.MessageAction.chat()</li> </td> </tr> <tr> <td><strong>File</strong>:</td> <td>java/lang/Class.java</td> </tr> <tr> <td><strong>Line number</strong>:</td> <td>1,655</td> </tr> </table> </div> <div id="stacktraces"> <hr /> <h3>Stacktraces</h3> <div class="stacktrace" style="padding-left: 0em"> <strong>java.lang.NoSuchMethodException: org.melody.controller.MessageAction.chat()</strong> <div> <pre> java.lang.Class.getMethod(Class.java:1655) org.apache.struts2.interceptor.validation.AnnotationValidationInterceptor.getActionMethod(AnnotationValidationInterceptor.java:75) org.apache.struts2.interceptor.validation.AnnotationValidationInterceptor.doIntercept(AnnotationValidationInterceptor.java:47) com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:98) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.ConversionErrorInterceptor.intercept(ConversionErrorInterceptor.java:138) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:249) com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:98) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:249) com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:98) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.StaticParametersInterceptor.intercept(StaticParametersInterceptor.java:191) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) org.apache.struts2.interceptor.MultiselectInterceptor.intercept(MultiselectInterceptor.java:73) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) org.apache.struts2.interceptor.CheckboxInterceptor.intercept(CheckboxInterceptor.java:91) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) org.apache.struts2.interceptor.FileUploadInterceptor.intercept(FileUploadInterceptor.java:252) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.ModelDrivenInterceptor.intercept(ModelDrivenInterceptor.java:100) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.ScopedModelDrivenInterceptor.intercept(ScopedModelDrivenInterceptor.java:141) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.ChainingInterceptor.intercept(ChainingInterceptor.java:145) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.PrepareInterceptor.doIntercept(PrepareInterceptor.java:171) com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:98) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.I18nInterceptor.intercept(I18nInterceptor.java:139) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) org.apache.struts2.interceptor.ServletConfigInterceptor.intercept(ServletConfigInterceptor.java:164) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.AliasInterceptor.intercept(AliasInterceptor.java:193) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) com.opensymphony.xwork2.interceptor.ExceptionMappingInterceptor.intercept(ExceptionMappingInterceptor.java:189) com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:246) org.apache.struts2.impl.StrutsActionProxy.execute(StrutsActionProxy.java:54) org.apache.struts2.dispatcher.Dispatcher.serviceAction(Dispatcher.java:562) org.apache.struts2.dispatcher.ng.ExecuteOperations.executeAction(ExecuteOperations.java:77) org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter.doFilter(StrutsPrepareAndExecuteFilter.java:99) org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:243) org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210) org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:222) org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:123) org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:502) org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:171) org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:100) org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:118) org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:408) org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Processor.java:1041) org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(AbstractProtocol.java:603) org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEndpoint.java:310) java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615) java.lang.Thread.run(Thread.java:724) </pre> </div> </div> </div> <div class="footer"> <hr /> <p> You are seeing this page because development mode is enabled. Development mode, or devMode, enables extra debugging behaviors and reports to assist developers. To disable this mode, set: <pre> struts.devMode=false </pre> in your <code>WEB-INF/classes/struts.properties</code> file. </p> </div> </body> </html

**部署到网上就需要相对地址正确**

对于静态资源不是放在WEB-INF目录下  
需要先获得上下文路径,然后将这个路径当做所有的资源访问的前缀  
**String path=request.getContextPath();**

<img src=“<%=path%>/css/abc.css”>

ajax(“<%=path%>/ajax/message\_chat”,{},function(){})

**BankLogin bugs list:**

Cannot find class: org/aopalliance/intercept/MethodInterceptor **缺少aopalliance.jar**

java.lang.NoClassDefFoundError: org/springframework/transaction/interceptor/TransactionInterceptor

**缺少org.springframework.transaction.jar**

java.lang.NoClassDefFoundError: org/springframework/aop/config/AopNamespaceUtils

**缺少org.springframework.aop.jar**

org.xml.sax.SAXParseException: The prefix "mvc" for element "mvc:resources" is not bound.

命名空间加上：xmlns:mvc=<http://www.springframework.org/schema/mvc>

在xsi:schemalocation加上<http://www.springframework.org/schema/mvc> <http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd>

java.lang.NoClassDefFoundError: org/aspectj/weaver/reflect/ReflectionWorld$ReflectionWorldException

**添加aspectj.jar**

java.lang.NoClassDefFoundError: org/apache/commons/collections/ExtendedProperties

缺少commons-collections.jar

**使用RequestMapping(“/”)**

**No mapping found for HTTP request with URI [/BankLogin/] 匹配不上，是没有加这句话**

**<mvc:annotation-driven/> 也有可能是<url-mapping>/</url-mapping>**