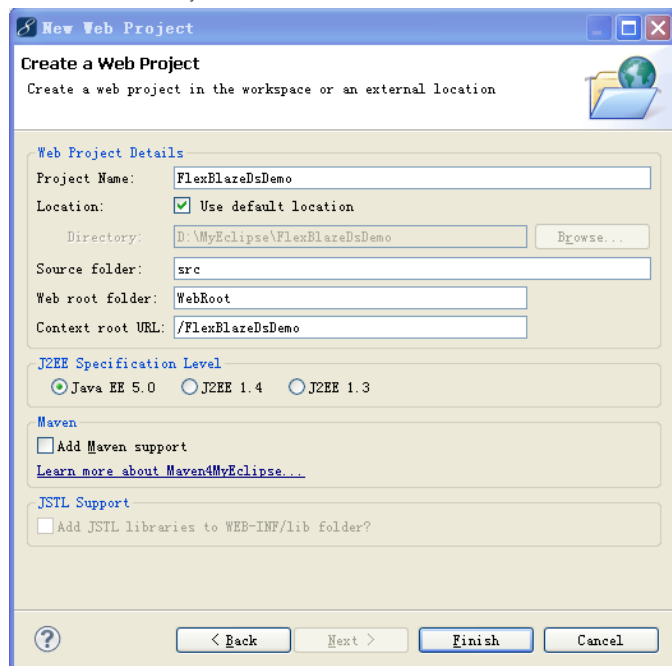


## 转 Flex+BlazeDs+Java的教程及Demo

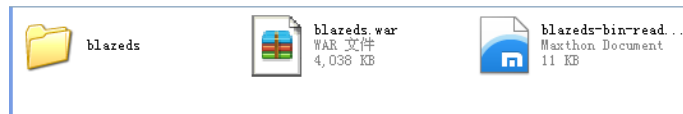
2013年10月06日 20:05:33 阅读数：3208

这个基础教程是 Flex4.1，Blazeds4.0 为基础写的。

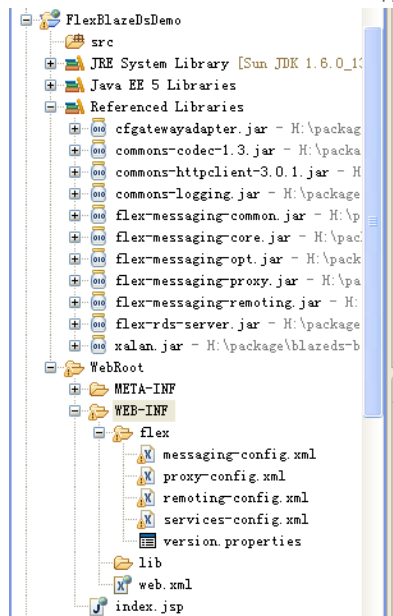
第一步：新建 Web Project，名字为：FlexBlazeDsDemo，点 Finish



第二步：去 Adobe 公司的网站上下载 Blazeds 的包，



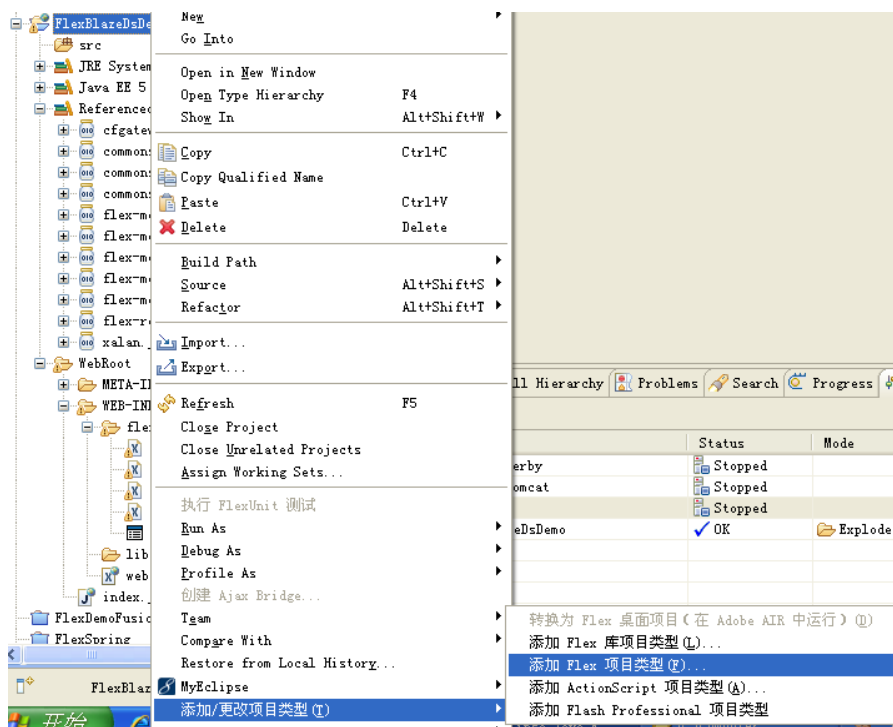
解压 blazeds.war，看到 blaeds 的文件夹，里面有一个 lib 的文件夹，将包引入到项目工作中去  
同时将 blazeds->WEB-INF 下的 flex 文件夹 Copy 到项目中的 WEB-INF 下，目录结构如下：



第三步：在 Tomcat 中发布项目

第四步：在 Web Project 中添加 Flex 项目

Step1：



Step2 : 服务器技术选择 J2EE 及 Blazeds



Step3 :



Step4 : 配置上述图中的信息, 根文件夹, 选择 FlexBlazeDsDemo 发布之后的位置, 配置之后的信息如下:



Step5 : 新建 html-template (打开 problem 窗口, 如果有提示这个错误就新建) 和 flex-src 文件夹, 目录结构如下: 建立完文件, 会提示: Description

Resource

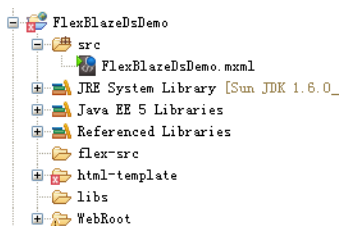
Path

Location

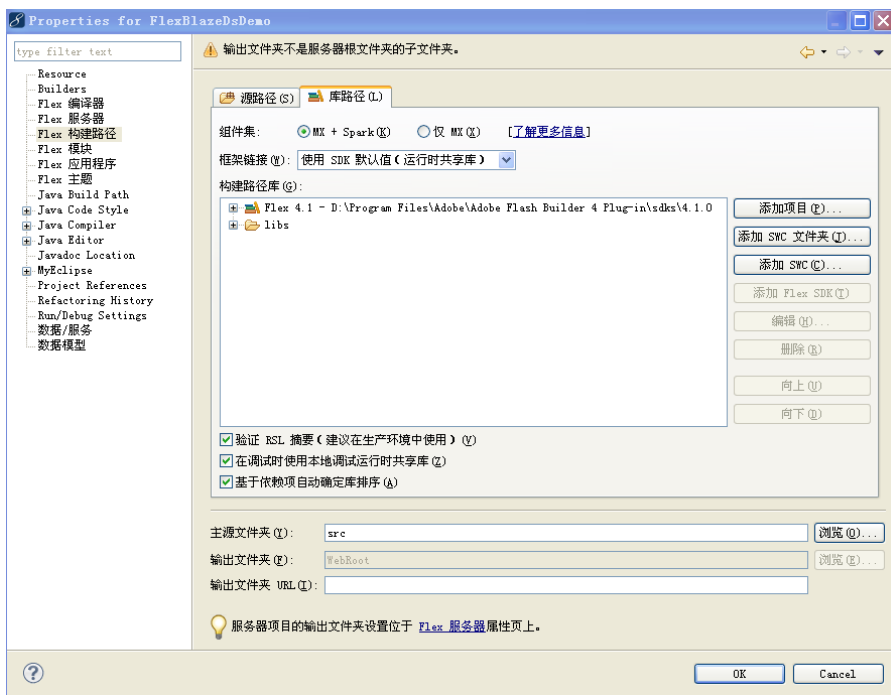
Type

missing ; before statement

index.template.htm, 这个错误不用处理, 接着往下走。



Step6 : 调整目录结构及参数配置, 如下图



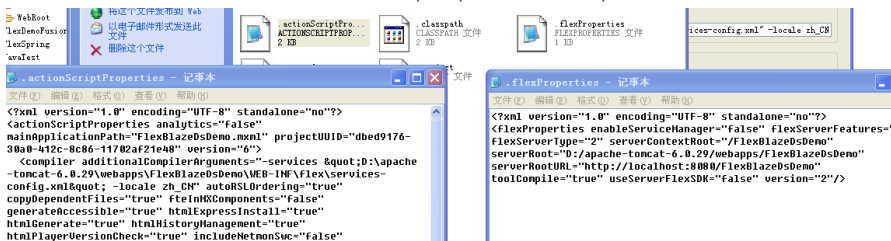
在上图中，将主源文件夹修改为：flex-src

**特别注意：**将FlexBlazeDsDemo.mxml文件移动到flex-src目录下，因为我们已经将主源文件夹修改，并且以后所建的Flex文件也都要放到flex-src下。

Step8：服务参数，也就是 Flex 编译器



我们也可以打开配置文件去查看参数的（.actionScriptProperties 及 .flexProperties），如下图：



第五步：在FlexBlazeDsDemo输入以下几行代码

```

<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600">
    <fx:Declarations>
        <!-- 将非可视元素（例如服务、值对象）放在此处 -->
    </fx:Declarations>
    <fx:Script>
        <![CDATA[
            import mx.controls.Alert;
            private function onClick():void{
                Alert.show("Hello, this is the first part of FlexBlazeDsDemo.");
            }
        ]]>
    </fx:Script>

    <s:Button label="click" click="onClick()" />
</s:Application>

```

第六步：在浏览器地址栏输入 <http://localhost:8080/FlexBlazeDsDemo/FlexBlazeDsDemo.html>，查看效果，端口号和项目名称改成自己即可。效果如下：



第七步：添加 BlazeDs，配置 web.xml 文件，将我们下载的 BlazeDs 包下在的 web.xml 文件覆盖我们项目的文件，web.xml 文件内容如下：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://
3 <web-app>
4
5 <!-- Http Flex Session attribute and binding listener support -->
6 <listener>
7     <listener-class>flex.messaging.HttpFlexSession</listener-class>
8 </listener>
9
10 <!-- MessageBroker Servlet -->
11 <servlet>
12     <servlet-name>MessageBrokerServlet</servlet-name>
13     <display-name>MessageBrokerServlet</display-name>
14     <servlet-class>flex.messaging.MessageBrokerServlet</servlet-class>
15     <init-param>
16         <param-name>services.configuration.file</param-name>
17         <param-value>/WEB-INF/flex/services-config.xml</param-value>
18     </init-param>
19     <load-on-startup>1</load-on-startup>
20 </servlet>
21
22 <servlet-mapping>
23     <servlet-name>MessageBrokerServlet</servlet-name>
24     <url-pattern>/messagebroker/*</url-pattern>
25 </servlet-mapping>
26 </web-app>
27

```

第八步：写一个 Java 类，HelloWorld，代码如下：

```

1 /**
2  * Author:Wangdongsong
3  * 2011-10-29 上午09:39:31
4  *
5  * Email:wangdongsong1229@163.com
6  * QQ:34655442
7  */
8 package com.wds.blazeds;
9
10 /**
11  * @author Administrator
12  * 2011-10-29 上午09:39:31
13  *
14  */
15 public class HelloWorld {
16
17     public String sayHi(String string){
18         System.out.println(string);
19         return "this is Java, welcome";
20     }
21
22 }
23

```

第九步：配置 remote-Object.xml 文件

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <service id="remoting-service"
3     class="flex.messaging.services.RemotingService">
4
5     <adapters>
6         <adapter-definition id="java-object" class="flex.messaging.services.remoting.ad
7     </adapters>
8
9     <default-channels>
10         <channel ref="my-amf"/>
11     </default-channels>
12
13     <destination id="helloService">
14         <properties>
15             <source>com.wds.blazeds.HelloWorld</source>
16             <scope>request</scope>
17         </properties>
18     </destination>
19
20 </service>

```

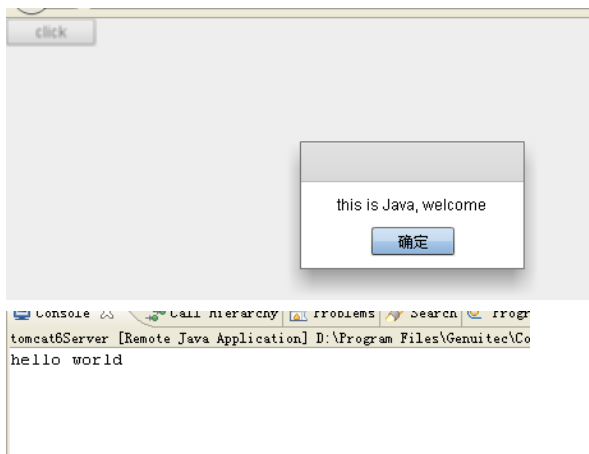
第十步：更改 FlexBlazeDsDemo 代码，内容如下： destination 一定要与上一步中的 remote-object.xml 的 destination 一样，result 是调用成功之后处理函数。

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
3     xmlns:s="library://ns.adobe.com/flex/spark"
4     xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600">
5     <fx:Declarations>
6         <s:RemoteObject id="helloService"
7             destination="helloService"
8             result="helloService_resultHandler(event)"/>
9     </fx:Declarations>
10    <fx:Script>
11        <![CDATA[
12            import mx.controls.Alert;
13            import mx.rpc.events.ResultEvent;
14            private function onClick():void{
15                helloService.sayHi("hello world");
16            }
17
18            protected function helloService_resultHandler(event:ResultEvent):void
19            {
20                Alert.show(event.result as String);
21            }
22        ]]>
23    </fx:Script>
24    <s:Button label="click" click="onClick()" />
25 </s:Application>

```

第十一步：测试



原文地址：<http://wangdongsong.iteye.com/blog/1222435>

文章标签：[java](#) [flex](#) [blazeds](#) [demo](#)

个人分类：[Flash相关](#)

此PDF由spyyg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com