[**BlazeDS 整合 Flex HelloWorld 示例**](http://www.cnblogs.com/hoojo/archive/2011/09/16/2178802.html)

开发环境：

System：Windows

WebBrowser：IE6+、Firefox3+

JavaEE Server：tomcat5.0.2.8、tomcat6

IDE：eclipse、MyEclipse 8

Flex IDE：Flash Builder 4

BlazeDS：4.5

开发依赖库：

JavaEE5、blazeDS 4.5

Email：hoojo\_@126.com

Blog：<http://blog.csdn.net/IBM_hoojo>

<http://hoojo.cnblogs.com/>

**一、准备工作**

1、 首先要提供相关的jar包

Java服务器端需要提供BlazeDS相关的配置和jar包

下载地址：<http://opensource.adobe.com/wiki/display/blazeds/download+blazeds+trunk>

下载后，解压你可以看到这样的一个目录



Docs就是文档

Resource是源码

SampleDB是示例用的数据库，可以运行startdb.bat来启动数据库

Tomcat是内置的tomcat，如果你没有tomcat的话可以使用它，在tomcat的webapps目录中有samples示例

blazeds.war就是blazeDS的核心文件、库，你可以把这个war放到tomcat的webapps目录下，就会自动解压。当然你也可以自己手动解压。

Blazeds-spring.war是和spring整合的配置

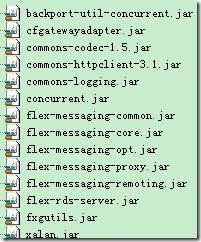
Ds-console.war是blazeDS的控制台程序

Samples.war是官方提供的示例

Samples-spring.war是spring和blazeDS的整合示例

**二、部署服务器端程序**

1、新建一个JavaWeb Project工程，然后在WEB-INF/lib目录中添加如下jar包



这些jar包可以在blazeds.war包中的lib目录中可以找到

2、 然后你需要将blazeds.war包中的WEB-INF目录下的flex目录复制到当前工程的WEB-INF下

3、 将blazeds.war包中的WEB-INF目录下的web.xml的配置，添加到当前工程的web.xml文件中

4、 最后基本的样式如下



5、 最后你发布当前工程，如果没有错误就表明你服务器端部署成功了。

6、 编写一个HelloWorld的java程序。代码如下

package com.hoo.flex;

/\*\*

\* <b>function:</b> HelloWorld Example

\* @author hoojo

\* @createDate 2011-8-31 下午06:11:27

\* @file HelloWorld.java

\* @package com.hoo.flex

\* @project BlazeDSServer

\* @blog http://blog.csdn.net/IBM\_hoojo

\* @email hoojo\_@126.com

\* @version 1.0

\*/

public class HelloWorld {

public HelloWorld() {

}

public String sayHello(String name) {

return "[" + name + "] say hello!";

}

}

就一个sayHello方法，接收一个参数。

**三、Flex客户端程序**

1、创建一个Flex工程，在选择服务器技术的时候，你需要选择J2EE。然后勾上使用J2EE技术，然后选择BlazeDS。点击Next下一步



2、配置根文件夹，也就是JavaEE服务器端发布程序在tomcat中的位置。我这里是在tomcat的webapps的BlazeDSServer中，BlazeDSServer是我的服务器端程序。根URL是访问服务器端程序的url；上下文目录对应工程名称；最后就是输出文件夹目录，这个是Flex的文件最后在tomcat中保存的目录。



3、最后你需要设置服务器端的services-config.xml的路径到编译参数中，这个很重要！如果你不设置的话，那么你在后面用RemoteObject调用BlazeDS的时候，就需要设置endpoint。设置如下：



-services是参数键，后面的字符串是值。我这里是设置BlazeDSServer发布到tomcat目录中的services-config.xml的路径。

4、编译Flex前端代码，代码如下：

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

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" viewSourceURL="BlazeDSHelloWorld.mxml" layout="absolute" minWidth="955" minHeight="600">

<mx:Script>

<![CDATA[

import mx.controls.Alert;

import mx.rpc.AsyncToken;

import mx.rpc.events.ResultEvent;

private function faultHandler(event: Event): void {

Alert.show(event.toString(), event.type);

}

private function resultHandler(event: ResultEvent): void {

//event.result是服务器端返回对象

result.text = "Message：" + event.result.toString();

}

private function sendHandler(): void {

helloRemoteObject.sayHello(userName.text);

}

]]>

</mx:Script>

<!-- 当工程没有设置编译器-service参数 或是-context-root等参数，就需要手动设置endpoint参数 -->

<mx:RemoteObject

id="helloRemoteObject"

destination="helloWorld"

fault="faultHandler(event)"

result="resultHandler(event)"

showBusyCursor="true"/>

<mx:Panel x="10" y="10" width="272" height="148" layout="absolute" title="BlazeDS Remote HelloWorld Sample">

<mx:Label x="10" y="22" text="请输入名称"/>

<mx:TextInput x="70" y="19" id="userName"/>

<mx:Button x="184" y="45" label="发送" click="sendHandler()"/>

<mx:Text x="10" y="79" id="result"/>

</mx:Panel>

</mx:Application>

首先你需要将Java服务器端的HelloWorld程序配置在flex的remoting-config.xml中，配置如下：

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

<service id="remoting-service"

class="flex.messaging.services.RemotingService">

<adapters>

<adapter-definition id="java-object" class="flex.messaging.services.remoting.adapters.JavaAdapter" default="true"/>

</adapters>

<default-channels>

<channel ref="my-amf"/>

</default-channels>

<destination id="helloWorld">

<properties>

<source>com.hoo.flex.HelloWorld</source>

</properties>

</destination>

</service>

上面mxml代码中的RemoteObject的destination对应的就是remoting-config.xml配置文件中的destination的id。这个是一一对应的，然后在sendHandler方法中，helloRemoteObject对应的就是RemoteObject的id，而sayHello方法对应的就是配置在remoting-config.xml中的destination的source的Java服务器端代码的公有方法。添加完配置后，需要重启tomcat。

运行上面的flex程序后，如果输入参数后，点击发送，可以看到服务器端返回的消息就说明BlazeDS整合Flex成功了。