DWR框架介绍及使用方法

—— giser062   2014-12-04

**1DWR介绍**

DWR是一个RPC类库，可以很容易的用Javascript调用Java方法，也可以用Java调用Javascript方法（也称为Reverse Ajax）。

主页：http://directwebremoting.org/dwr/index.html

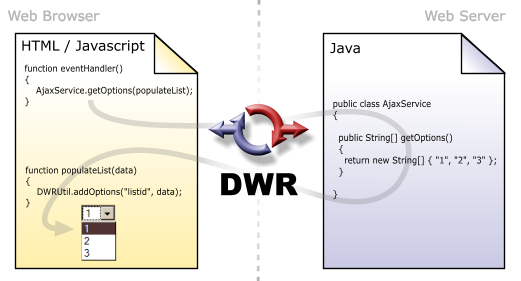
下载链接：

http://java.net/projects/dwr/downloads/download/DWR\_3.x/3.0\_RC3/dwr.jar

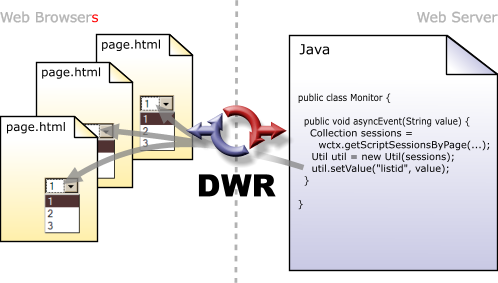
官方文档：

<http://directwebremoting.org/dwr/documentation/index.html>

**1.1Javascript调用Java**



**1.2Java调用Javascript（Reverse Ajax）**



**2快速使用指南**

**2.1下载安装dwr JAR 文件**

下载dwr.jar文件，并且复制到Web工程的WEB-INF/lib目录里

**2.2下载安装Commons Logging JAR文件**

dwr.jar依赖commons-logging.jar，下载该commons-logging.jar，并且复制到Web工程的WEB-INF/lib目录里

**2.3配置DWR servlet**

将下面的Servlet配置内容添加到Web应用程序的WEB-INF/web.xml里。其中<servlet>部分放在web.xml中最后一个<servlet>的后面，<servlet-mapping>同样放在web.xml中最后一个<servlet-mapping>的后面。

 XML Code

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 |  | <servlet> <display-name>DWR Servlet</display-name> <servlet-name>dwr-invoker</servlet-name> <servlet-class>org.directwebremoting.servlet.DwrServlet</servlet-class> <init-param> <param-name>debug</param-name> <param-value>true</param-value> </init-param> </servlet>  <servlet-mapping> <servlet-name>dwr-invoker</servlet-name> <url-pattern>/dwr/\*</url-pattern> </servlet-mapping> |

**2.4创建DWR的配置文件(dwr.xml)**

在WEB-INF目录里创建一个新的配置文件dwr.xml，详细的dwr.xml文档地址：

<http://directwebremoting.org/dwr/documentation/server/configuration/dwrxml/index.html>

简单示例：

 XML Code

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 |  | <!DOCTYPE dwr PUBLIC "-//GetAhead Limited//DTD Direct Web Remoting 3.0//EN" "http://getahead.org/dwr/dwr30.dtd"> <dwr> <allow> <create creator="new" javascript="JDate"> <param name="class" value="java.util.Date" /> </create> <create creator="new" javascript="Demo"> <param name="class" value="your.java.Bean" /> </create> </allow> </dwr> |

这个DWR配置文件定义了DWR能创建哪些类，并且能被JavaScript远程调用。上面的例子中我们定义了2个能被JavaScript远程调用的类，并且给出了在JavaScript中的类名。

其中上面用到的new creator，要求所有用到的Java Beans都必须有一个**无参的公共构造方法**。

DWR还有个限制：

**避免使用JavaScript的保留字**，因为用保留字命名的方法会自动被执行。大多数的JavaScript保留字和Java保留字相同，所以不能定义一个“try()”方法，然而有一个最常用的方法是“delete()”，这个方法在JavaScript中有特殊含义，但在Java中没有。

**2.5获取DWR的测试驱动**

启动Web服务器，访问下面的URL：

http://localhost:8080/[YOUR-WEBAPP-CONTEXT]/dwr/

你应该看到一个页面显示在上一步中配置的类。点击一个链接，应该能看到所有能被调用的方法列表，可以输入需要的参数（如果需要参数的话），然后点击“执行”按钮。如果设置了DWR的debug init-param为true的话，这个页面会触发后台Java的Debugging工具。

**2.6如何在Web应用程序中使用**

1、打开[http://localhost:8080/[YOUR-WEBAPP-CONTEXT]/dwr/](http://localhost:8080/%5bYOUR-WEBAPP-CONTEXT%5d/dwr/)，点击你的类

2、把页面上的javascript标签内容添加到需要的页面上

 XML Code

|  |  |  |
| --- | --- | --- |
| 1 2 3 |  | <script type='text/javascript' src='/[YOUR-WEBAPP-CONTEXT]/dwr/engine.js'></script> <script type='text/javascript' src='/[YOUR-WEBAPP-CONTEXT]/dwr/util.js'></script> <script type='text/javascript' src='/[YOUR-WEBAPP-CONTEXT]/dwr/interface/[YOUR-SCRIPT].js'></script> |

**3JavaScript调用Java方法**

**3.1使用回调函数处理异步请求**

有两种建议的方式

**3.1.1Call Options Object**

建议的语法是使用Call Options Object方式，明确指定callback函数和其它选项。

假设我们有一个Java方法，如下：

public class Remote {

    public String getData(int index) { return “data is ” + index }

}

可以使用下面的javascript调用方式：

<script type="text/javascript"

    src="*[WEBAPP]*/dwr/interface/Remote.js"> </script>

<script type="text/javascript"

    src="*[WEBAPP]*/dwr/engine.js"> </script>

...

Remote.getData(42, {

  callback:function(str) {

    alert(str);

  }

});

‘42’是传递给Java方法getData()的参数。str是getData()方法的返回值。

**指定timeout和错误处理**

Remote.getData(42, {

  callback:function(str) { alert(str); },

  timeout:5000,

errorHandler:function(message) { alert("Oops: " + message); }

});

**3.1.2Simple Callback Functions**

另一个方法是简单地把一个回调函数参数列表：

function handleGetData(str) {

  alert(str);

}

Remote.getData(42, handleGetData);

或者你可以都写在一行：

Remote.getData(42, function(str) { alert(str); });

**3.2创建Javascript Object和Java Object映射**

假设有下面的Java方法，有一个Person类型的Java Bean参数：

public class Remote {

  public void doSomethingWithPerson(Person p) {

    // Some Java code that does something with Person p.

  }

}

Person类如下：

public Person {

  private String name;

  private int age;

  private Date[] appointments;

  // getters and setters ...

}

你可以像下面一样调用Javascript：

var myJSPerson = {

  name:"Fred Bloggs",

  age:42,

  appointments:[ new Date(), new Date("1 Jan 2008") ]

};

Remote.doSomethingWithPerson(myJSPerson);

具体不同情况和不同参数类型的调用方式可参考附件《DWR框架的使用说明》