

# 1. DWR

---

## 1.1 DWR 基础

### 1.1.1 概念

#### 1、什么是 DWR?

Direct Web Remote, 直接 Web 远程。是一个 Ajax 的框架。

#### 2、作用

使用 DWR,可以直接在 html 网页中调用 Java 对象的方法 (通过 JS 和 Ajax)。

#### 3、基本原理

主要技术基础是: AJAX+反射。

- 1) JS 通过 AJAX 发出请求, 目标地址为 “/dwr/\*”, 被 DWRServlet (中央控制器) 拦截到;
- 2) DWRServlet 再根据请求地址中的对象名, 和 dwr.xml 配置文件中的对象名相匹配, 根据其类名用反射机制来创建对象并调用其目标方法;
- 3) DWR 再将目标方法的返回值以响应的方式, 通过 AJAX 发送到客户端 JS 中的响应函数 (自定义), 响应函数再对返回值进行处理。

### 1.1.2 使用

使用 DWR 的步骤

#### 1、下载 DWR 的包。

Dwr.jar

#### 2、引用 DWR

复制 dwr.jar 到你的 web 工程的 WEB-INF/lib 目录下;

在你的 Web.xml 文件中添加一个 servlet 的配置:

```
<servlet>
    <servlet-name>dwr-invoker</servlet-name>

    <servlet-class>org.directwebremoting.servlet.DwrServlet</servlet-
class>
    <init-param>
        <param-name>debug</param-name>
        <param-value>>false</param-value>
    </init-param>
    <init-param>
        <param-name>logLevel</param-name>
        <param-value>WARN</param-value>
    </init-param>
```

```

</servlet>
<servlet-mapping>
    <servlet-name>dwr-invoker</servlet-name>
    <url-pattern>/dwr/*</url-pattern>
</servlet-mapping>

```

在 WEB-INF 目录下创建 dwr 的配置文件 (dwr.xml)

### 3、创建普通 Java 类和方法

```

package com.zhoujisiong.dwr.biz;

public class LoginBiz {
    public boolean login(String name,String pwd) {
        System.out.println("-----取得用户名:"+name+"; 和密
码:"+pwd);
        if ("zuxia".equals(name) && "123".equals(pwd) ) {
            return true;
        }else{
            return false;
        }
    }
}

```

4、在 web-inf 目录下，新建 dwr.xml 文件，内容如下：

```

<dwr>
    <allow>
        <create creator="new" javascript="LoginBiz" scope="application">
            <param name="class"
                value="com.zhoujisiong.dwr.biz.LoginBiz" />
        </create>
    </allow>
</dwr>

```

### 5、创建 html 文件

引用 3 个 js 文件 ( )，并写 JS 请求函数和响应函数

```

<script type="text/javascript" src="dwr/engine.js"> </script>
<script type="text/javascript" src="dwr/util.js"> </script>

<!-- 引用dwr 框架根据它的配置文件动态生成的js -->
<script type="text/javascript"
src="dwr/interface/LoginBiz.js"></script>
<script type="text/javascript" language="javascript">
    //请求函数
    function login() {

```

法

```
var uname=document.getElementsByName("uname")[0].value;
var upass=document.getElementsByName("upass")[0].value;
alert(uname+"\n"+upass);
LoginBiz.login(uname,upass,loginResponse); //调用指定对象的指定方法
}
//响应函数
function loginResponse(msg){
    alert(msg);
}
</script>
```

## 1.2 DWR 高级

### 1.2.1 在 JS 和 Java 之间传递实体对象。

Entity:JS→JAVA

Entity:JS←JAVA

### 1.2.2 在 JS 和 Java 之间传递 List 对象。

List:JS←-JAVA

List:JS-->JAVA

### 1.2.3 在 JS 和 Java 之间传递 Map 对象。

Map:JS←-JAVA

Map:JS-->JAVA

## 1.3 DWR 和 SSH 集成

### 1.3.1 引用 SSH 的包和 Dwr 的包

将 SSH 的包复制到 WEB-INF/lib 目录下;

将 dwr.jar 复制到 WEB-INF/lib 目录下;

### 1.3.2 Web.xml 文件

```
<!-- 配置struts2的核心控制器 -->
<filter>
    <filter-name>Struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
```

```

<filter-mapping>
    <filter-name>Struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<!-- 配置Struts2集成Spring -->
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderList
ener</listener-class>
</listener>

<context-param>
    <param-name>contextConfigLocation</param-name>

    <param-value>/WEB-INF/classes/applicationContext.xml</param-value
>
</context-param>

<!-- Dwr的控制器 -->
<!-- Dwr的控制器 -->
<servlet>
    <servlet-name>DwrServlet</servlet-name>
    <servlet-class>org.directwebremoting.servlet.DwrServlet</servlet-
class>
</servlet>
<servlet-mapping>
    <servlet-name>DwrServlet</servlet-name>
    <url-pattern>/dwr/*</url-pattern>
</servlet-mapping>

```

### 1.3.3 在 WEB-INF/lib 目录下新建 Dwr.xml 文件，内容如下:

```

<dwr>
    <allow>
        <create creator="spring" javascript="userBiz">
            <param name="beanName" value="userBizImpl"/>
        </create>
        <convert converter="bean"
            match="com.zjs.sshems.entity.Userinfo" />
    </allow>
    <signatures>
        <![CDATA[
            import com.zjs.sshems.entity.Userinfo;
        ]]>
    </signatures>

```

```
</dwr>
```

### 1.3.4 Login.jsp

```
<%@ page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%@ taglib uri="/struts-tags" prefix="s"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<base href="<%=basePath%>">

<title>My JSP 'index.jsp' starting page</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">
<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->
<script type="text/javascript" src="dwr/engine.js"></script>
<script type="text/javascript" src="dwr/util.js"></script>
<script type="text/javascript"
src="dwr/interface/userBiz.js"></script>
<script type="text/javascript">
    function login() {
        var userName = DWRUtil.getValue("user.userName");
        var userPwd = DWRUtil.getValue("user.userPwd");
        alert(userName + ";" + userPwd);
        userBiz.loginVerify(userName, userPwd, loginResponse);
    }

    function loginResponse(userInfo) {
        alert("取得业务方法的返回值: " + userInfo.userId + ";" +
userInfo.userName + ";" + userInfo.userPwd);
    }
</script>
```

```
</head>

<body>
  <form action="userLogin!login" method="post" >

    用户名: <input type="text" name="user.userName"/><br/>
    密码: <input type="password" name="user.userPwd"/><br/>
    <input type="submit" value="登录"/>
    <input type="button" value="通过Dwr访问登陆方法"
onclick="login()" />
  </form>
  ${error }
</body>
</html>
```