



# **Advanced Ajax with DWR**

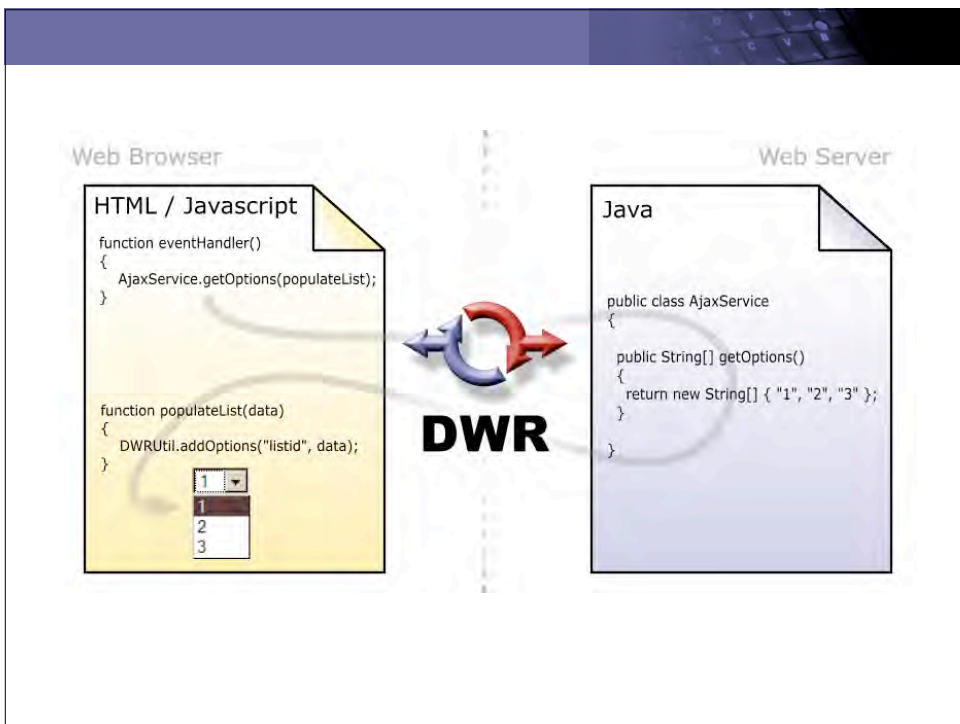
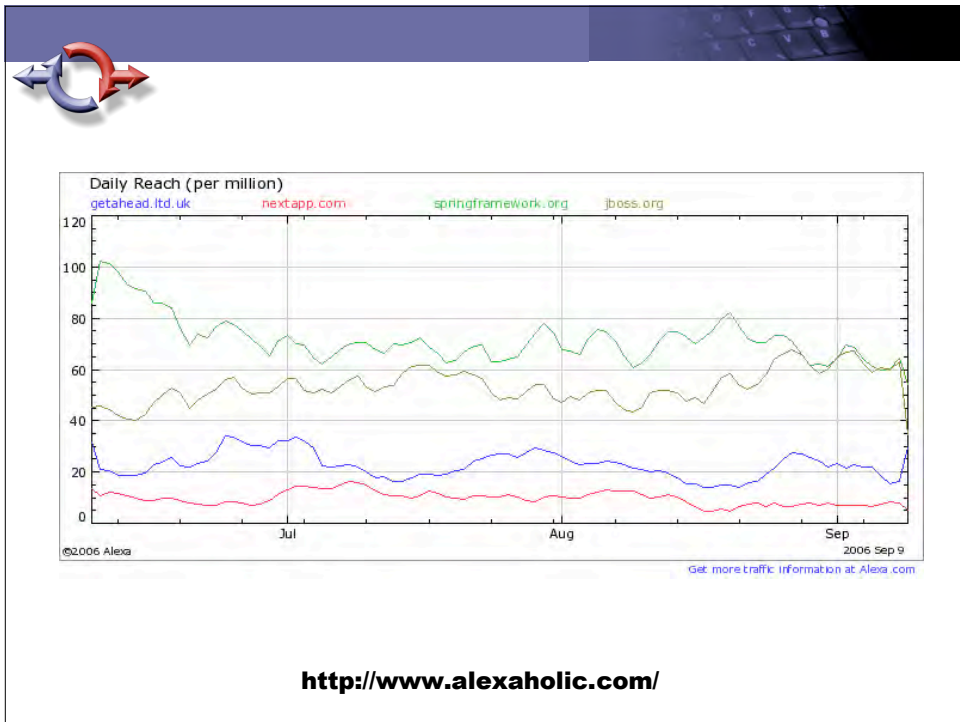
**Joe Walker**



# **DWR**

**Direct Web Remoting  
Easy Ajax for Java**

**Export Java to the browser  
and Javascript to the server**





## Step 1

### Copy dwr.jar file into WEB-INF/lib

<http://www.directwebremoting.org>



## Step 2

### Tweak WEB-INF/web.xml: (DWR 1.x)

```
<servlet>
  <servlet-name>dwr-invoker</servlet-name>
  <display-name>DWR Servlet</display-name>
  <servlet-class>uk.ltd.getahead.dwr.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>
```



## Step 2

### Tweak WEB-INF/web.xml: (DWR 2.x)

```
<servlet>
  <servlet-name>dwr-invoker</servlet-name>
  <display-name>DWR Servlet</display-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>
```



## Step 3

### dwr.xml gives DWR permission to call your code


```
<dwr>
  <allow>
    <create creator="new" javascript="JDate">
      <param name="class" value="java.util.Date"/>
    </create>
  </allow>
</dwr>
```



**Step 4**

**Browse the automagically generated test pages**

<http://localhost:8080/WEBAPP/dwr/>



**Demo**



**Reverse Ajax**



# Interactivity



**Reverse Ajax**



**Javascript: Turn Reverse Ajax on**

```
DWREngine.setReverseAjax(true);
```

**Java: Send scripts to the browser**

```
ScriptSession.addScript("alert('Hi')");
```



## Server Side DwrUtil

**Access all DWRUtil Javascript  
functions directly from Java**

```
import org.....DwrUtil;
```

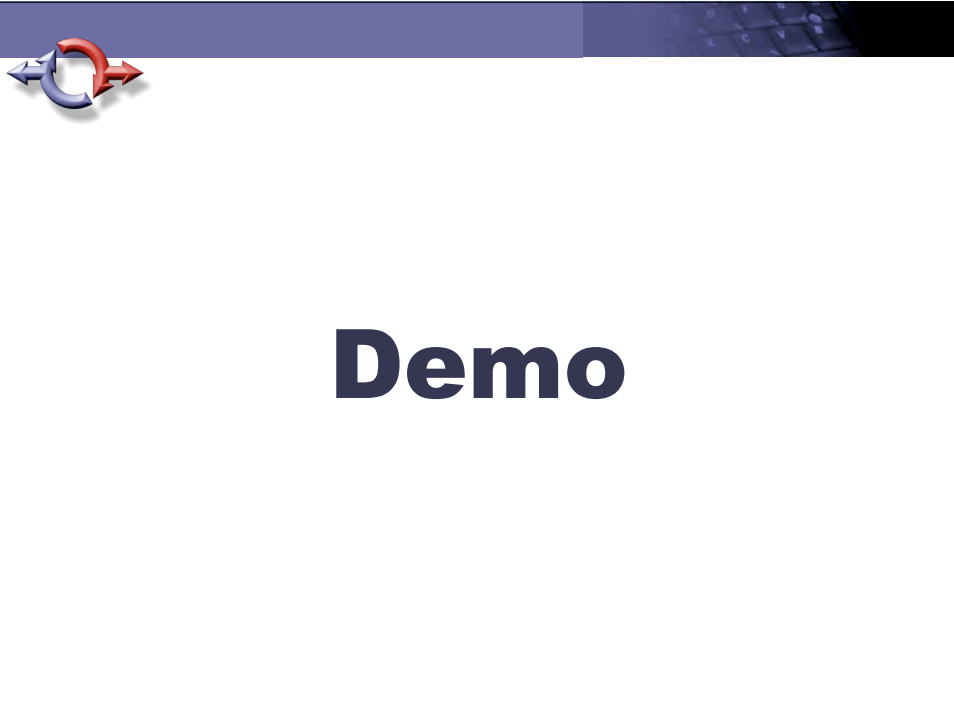
```
DwrUtil util = new DwrUtil(...);  
util.addOptions("list", array);
```



## Script.aculo.us

```
import org...scriptaculous.Effect;
```

```
effect.fade("elementId");
```





## Security

- **DWR does not call anything without permission**
- **Audit**
- **Multiple dwr.xml Files**
- **Role based security**
- **Method level access control**
- **Risks**



## Accessibility



**DWRUtil has pluggable notifiers**

- **.focus() for screen readers**
- **Yellow Fade for partially sighted**

**New notifiers can be added**

- **Example: ring bell on change**
- **Interact with advanced accessibility aids**



## Ajax Filters



**Ajax Filters give you:**

- **Logging**
- **Latency (more realistic testing)**
- **Security**



## Creators / Converters

- **Creators create objects that live on the server and have their methods remoted**
- **Converters marshal parameters and return types**

```
var r = Remote.method(param, callback);
```

Uses a  
Converter

Uses a  
Creator

- **Created object do things**
- **Converted objects carry data**



## Spring 2



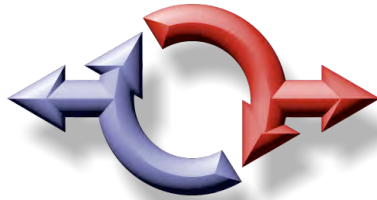
### Just use beans.xml

```
<beans>
  <bean id="Clock" class="com.eg.Clock">
    <property name="foo" ref="foo"/>
    <dwr:remote javascript="Clock"/>
  </bean>
  ...
</beans>
```



## Integration

- **Struts v1: StrutsCreator**
- **Struts v2: Comes with DWR**
- **Webwork: InfoQ Integration**
- **JSF: FacesCreator**
- **Rife: Comes with DWR**
- **DOM, XOM, JDOM, DOM4J**
- **Hibernate: Converters**

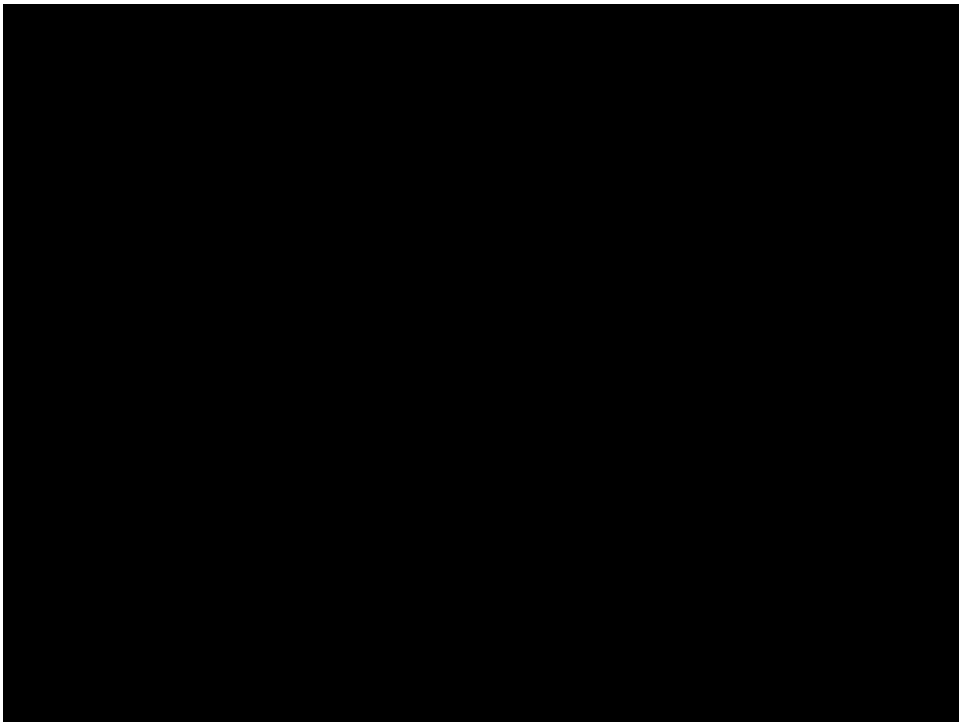


# DWR

**Easy Ajax for Java**

<http://www.directwebremoting.org/>

<http://getahead.ltd.uk/blog/joe/>





## Engine Options

### Robustness

- **timeout**
- **errorHandler**
- **warningHandler**

### UI Clues

- **preHook**
- **postHook**

### Remoting Options

- **method**
- **verb**
- **async**

### Call Sequencing

- **ordered**
- **callback**

### Future

- **skipBatch**
- **onBackButton**
- **onForwardButton**



## DWRUtil Functions

- **\$**
- **addOptions** and **removeAllOptions**
- **addRows** and **removeAllRows**
- **getValue**, **getValues** and **getText**
- **setValue** and **setValues**
- **toDescriptiveString**
- **useLoadingMessage**



## Signatures

- **Sometimes reflection is not enough**
- **The <signature> element fixes the hole**

```
<signatures>
  <![CDATA[
    import java.util.List;
    import com.example.Check;
    Check.setLotteryResults(List<Integer> nos);
  ]]>
</signatures>
```

- **Not required in DWR 2 + JDK 5**



## Servlet Objects

**How can I access a session, servlet context, or other servlet objects?**

- **Either:**

```
WebContextFactory.get().getHttpSession();
```

- **Or just use a method parameter!**

```
class Test {
    public int method(HttpSession s) { ... }
}
```

**Ignore the parameter when calling from Javascript - DWR will fill it in.**



## Ajax Includes

**What's the DWR equivalent of  
<jsp:include ...>?**

- **Java:**

```
public String getInclude() {  
    return WebContextFactory.get()  
        .forwardToString("forward.jsp");  
}
```

- **Javascript:**

```
function update() {  
    Demo.getInclude(function(html) {  
        DWRUtil.setValue("somediv", html);  
    });  
}
```



## Multiple dwr.xml Files

- **For separate J2EE security domains**
- **Or to separate components**
- **Configured in web.xml:**

```
<init-param>  
    <param-name>config*****</param-name>  
    <param-value>WEB-INF/dwr.xml</param-value>  
</init-param>
```



## Advanced Converters

- **Declare new converters in the <init> element in dwr.xml**
- **Use \$ for inner classes**
- **BeanConverter can restrict exported properties**
- **Convert anything syntax:**

```
<convert converter="bean" match="**"/>
```



## Advanced Creators

- **Scope options**
- **Javascript names for session pre-population**
- **NullCreator for static methods**
- **Reloading the ScriptedCreator**