

## DWR para iniciantes

Este tutorial assume que você tenha conhecimentos básicos de uma aplicação web em java. No entanto um conhecimento em C, garante o bom entendimento dos códigos.

- Instalar o JAR DWR e colocá-lo no *WEB-INF/lib* de sua aplicação.
- Editar o seu *WEB-INF/web.xml* adicionando as seguintes linhas :

```
<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>
```

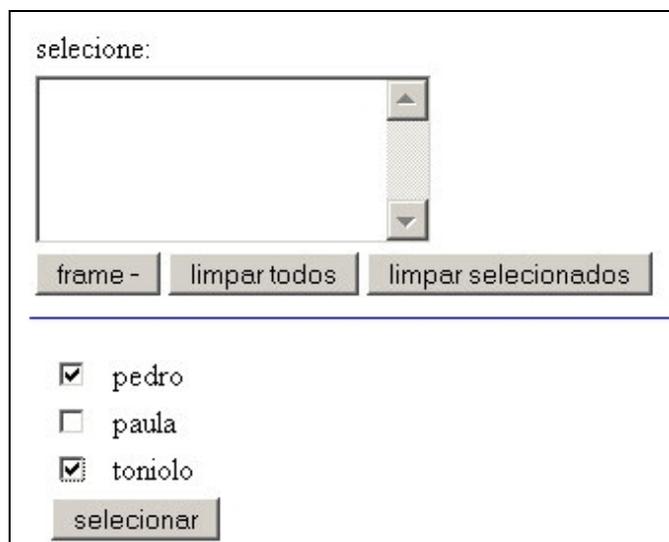
- Criar um arquivo chamado *dwr.xml* no mesmo diretório de seu *web.xml* . Este arquivo será o responsável por dizer ao DWR quais os métodos poderão ser acessados. Segue exemplo abaixo :

```
<!DOCTYPE dwr PUBLIC
  "-//GetAhead Limited//DTD Direct Web Remoting 1.0//EN"
  "http://www.getahead.ltd.uk/dwr/dwr10.dtd">

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

- Agora acesse <http://localhost:8080/suaaplicacao/dwr/> . Você deverá ver uma lista das classes que o DWR liberou acesso, entrando em cada uma delas você poderá ver os métodos liberados de cada uma delas.
- Se tudo correu bem, sua aplicação está pronta para o uso do DWR.

Vamos começar com um exemplo de aplicação prática de AJAX com um formulário. Desejamos popular o nosso select de acordo com as opções selecionadas em checkboxes dentro de um iframe. Ver figura 1.



The image shows a web form with the following elements:

- A label "selecione:" above a select dropdown menu.
- Three buttons: "frame -", "limpar todos", and "limpar selecionados".
- A list of checkboxes with names: "pedro" (checked), "paula" (unchecked), and "toniolo" (checked).
- A "selecionar" button at the bottom.

figura 1

Devemos incluir no cabeçalho de nossas páginas os seguintes código Javascript.

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

Não podemos esquecer de incluir o fonte em javascript da classe que liberamos acesso ao DWR, que no nosso exemplo chama-se AjaxAction.

```
<script type='text/javascript' src='dwr/interface/AjaxAction.js'></script>
```

Feito isso, podemos nos concentrar em implementar funcionalidades ajax com total transparência, pois o DWR possui suporte a todos os tipos primitivos. Para tipos próprios verifique a documentação online do DWR.

Agora, vamos fazer com que ao clicar em “selecionar”, o nosso elemento select seja populado sem a necessidade de um submit de nosso form dentro do iframe. Para isso, criaremos uma função javascript no evento onClick de nosso botão “selecionar”.

```
<input type="button"  
value="selecionar"  
onclick="selectPeople(this); parent.getPessoas(); ">
```

A função `selectPeople(this)` varre o nosso form em busca de elementos selecionados e através da função `selecionaPessoas(pessoas)`, inclui na nossa instância de classe, as pessoas selecionadas em nosso form, e vale repetir, sem a necessidade de dar submit. Note a forma com quem os nomes foram passados por parâmetro. Um string com cada nome separado por “~”, poderíamos passar sem problema um array javascript e ele seria convertido automaticamente para um array em java.

A função `parent.getPessoas()` está definida no pai de nosso frame. Ela é a responsável por buscar na instância da classe através da função `getPessoasArray()`, as pessoas que foram selecionadas no form.

```

<!--page.jsp-->
function selectPeople(form) {
    for(i=0;i<document.form2.people.length;i++) {
        if(document.form2.people[i].checked) {
            var pessoas = document.form2.people[i].value+'~~'+pessoas;
        }
    }
    if(pessoas !=null) {
        AjaxAction.selecionaPessoas (pessoas);
    }
}

-<!--index.jsp-->
function getPessoas() {
    AjaxAction.getPessoasArray (popula);
}
function popula(dados) {
    var sel = DWRUtil.getValue('sel');
    DWRUtil.removeAllOptions('sel');
    DWRUtil.addOptions('sel', dados);
    DWRUtil.setValue('sel', sel);
}

```

Vamos examinar agora as nossas classes em Java.

Na nossa classe AjaxAction teremos os nossos métodos exportados pelo DWR

```

public void selecionaPessoas(String pessoaLista) {
    String[] pessoas = pessoaLista.split("~~");
    this.pessoasArray = new String[pessoas.length];
    for(int i=0; i<pessoas.length; i++){
        if(pessoas[i].equalsIgnoreCase("undefined")) {
        }else{
            this.pessoasArray[i] = pessoas[i];
        }
    }
}

public String[] getPessoasArray() {
    return this.pessoasArray;
}

```

Com isso basta abrirmos nossa aplicação em <http://localhost:8080/suaaplicacao/> e ver resultado. As demais funcionalidades que estão previstas na figura 1 são sugeridas como exercício. Caso deseje o código fonte completo basta enviar-me um email com o assunto “DWR source” .

Pedro Pimentel  
 zukunft at gmail dot com

This work is licensed under the Creative Commons Attribution-ShareAlike 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Links:

- DWR (<http://getahead.ltd.uk/dwr/getstarted>)