DWR – DIRECTED WEB REMOTING

Dando continuidade ao nosso artigo vamos iniciar a segunda parte. Neste vamos abordar algumas funcionalidades um pouco mais avançadas do DWR.
Vamos também utilizar alguns exemplos com acesso ao banco de dados(MySql).

Parte 2 – Combos e Grids(addOptions(), addRows()).

Parte 2.2 – Grids (addRows()).

Criando grids dinâmicas(DWRUtil.addRows())

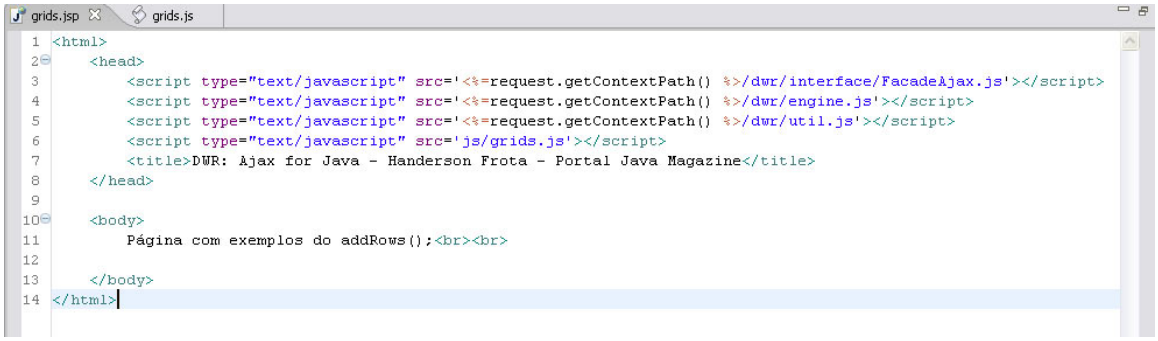
O DWR tem uma lib chamada *util.js,* que é nela que se localiza as principais funcionalidades do DWR. Vamos falar agora sobre o addRows() que tem como finalidade gerar grids(tabelas) dinâmicas.

Reutilizando a estrutura do nosso projeto anterior vamos criar um jsp chamado grids.jsp, e logo em seguida um link na nossa index.jsp que tenha acesso ao grids.jsp, criaremos também só por motivo didático um novo js, destinado apenas para as grids, vamos chama-lo de grids.js, veja abaixo:

Trecho de código na index.jsp.

```
<br>
<a href="grids.jsp">Exemplos de Grids</a>
```

Grids.jsp



Observe que os importes das bibliotecas .js são os mesmo, exceto pelo novo importe, que é o do nosso grids.js.

Explicando o addRows();

O addRows é uma função da biblioteca util.js que tem a finalidade de gerar tabelas e/ou linhas de uma tabela dinamicamente utilizando o conceito de Ájax e que rode em qualquer browser.

Para entendermos melhor.

Assim como as outras funções podemos também utilizar o mais por problemas de disposição e a finalidade do não é muito recomendada para essa função. Então qual utilizar ?
Vamos utilizar o a tag html <tbody>, que tem exatamente essa finalidade que necessitamos. Antes de mostrar algum exemplo vamos tentar entender a <tbody> .

<tbody>

Um <table> pode ser agrupado em “table head”(<thead>), “table foot”(<tfoot>>), e uma ou mais seções “table body”(<tbody>).
Uma tabela pode possuir apenas um “table head’ e “table foot”, mais pode conter dentro dela um ou mais “table body”(<tbody>).

O <thead> e <tfoot> devem conter informações acerca das colunas da tabela, e o <tbody> deve conter as linhas de dados da tabela , quando esses estão presentes, cada THEAD, TFOOT e TDOBY contém um grupo de linhas e cada grupo deve conter pelo menos uma linha (<tr>).

Onde inserir um <tbody> ?

Veja abaixo um exemplo simples de onde o <tbody> pode ser inserido:

```
<table>
	<tr>
		<td>Coluna da linha 1 sem tbody</td>
	</tr>
	<tbody>
		<tr>
			<td>Coluna da linha 2 com tbody</td>
		</tr>
	</tbody>
</table>
```

O tbody é perfeito para usarmos com o addRows, pois ele pode facilmente adicionar linhas na tabela que você desejar sem destruir o layout da mesma.

Utilizando o addRows().

Para os nossos exemplos vamos utilizar a seguinte tabela:

```
<table border="0">
	<tr>
		<td>Nome</td>
		<td>Endereço</td>
		<td>Telefone</td>
		<td>Cidade</td>
	</tr>
	<tbody id="identificadordoTbody"></tbody>
</table>
```

Parâmetros: addRows(id, cellfunc, opções);

1. **id:** Identificação do elemento da tabela, um elemento <tbody> é o recomendado.

2. **Cellfunc:** Um array de funções(uma função por coluna), utilizada para extrair os valores do array passado a função(os dados do banco por exemplo).

3. **Opções:** Utilizado para definição de colunas e linhas.Utilizado em situações que você deseja alterar as cores das novas linhas inseridas por exemplo.

    a. **rowCreator:** Função que você cria o tr(linha) alterando as definições: *css*, *style, bgcolor* ou qualquer outra tag que esteja dentro da tag <tr>.
        i. **Exemplo:**
```
rowCreator:function(options) {
		row = document.createElement("tr");
		row.bgColor = "#CCCC99";

		return row;
	}
```
    b. **cellCreator:** Função que você cria o td(coluna) alterando as definições: align, color ou qualquer outra tag que esteja dentro da tag **<td>.**
        i. **Exemplo:**
```
cellCreator:function(options) {
		var td = document.createElement("td");
		td.align = "center";
```

```
        return td;
    }
```

1. **Grid Simples:** Criar uma grid adicionando valores a uma tabela.
    a. **DWRUtil.addRows(idTbody, list, cellfunc);**
    b. Exemplo da grid gerada:

```
var cellfunc = [
                function(data) { return data.nome;  },
                function(data) { return data.endereco;  },
                function(data) { return data.telefone;  },
                function(data) { return data.cidade;  }
                ]
DWRUtil.addRows("grid1", list, cellfunc);
<table border="0">
 <tbody><tr bgcolor="#cccccc">
            <td>Nome</td>
            <td>Endereço</td>
            <td>Telefone</td>
            <td>Cidade</td>
            </tr>
    </tbody>
    <tbody id="Grid1">
            <tr>
                    <td>Christiano Milfont</td>
                     <td>Rua Rosas</td>
                     <td>32222222</td>
                    <td>Fortaleza</td>
            </tr>
    </tbody>
 </table>.
```

2. **Grid Avançado:** Criar uma grid adicionando valores a uma tabela e alterando as linhas e colunas.
    a. **DWRUtil.addRows(idTbody, list, cellfunc, options);**
    b. Exemplo da grid gerada:

```
var cellfunc = [
                function(data) { return data.nome;  },
                function(data) { return data.endereco;  },
                function(data) { return data.telefone;  },
                function(data) { return data.cidade;  }
]
DWRUtil.addRows("grid1", list, cellfunc, {
        rowCreator: function(options){
                row = document.createElement("tr");
                var row.bgColor = "#CCCC99";
                 return row;
```

```
                    } ,
                cellCreator: function(options) {
                        var td = document.createElement("td");
                        td.align = "center";
                        return td;
                        });
        <table border="0">
          <tbody><tr bgcolor="#cccccc">
                        <td>Nome</td>
                        <td>Endereço</td>
                        <td>Telefone</td>
                        <td>Cidade</td>
                        </tr>
            </tbody>
            <tbody id="Grid2">
              <tr bgcolor="#cccc99">
                        <td align="center">Christiano Milfont</td>
                        <td align="center">Rua Rosas</td>
                        <td align="center">32222222</td>
                        <td align="center">Fortaleza</td>
             </tr>
            </tbody>
          </table>.
```

Agora para entendermos melhor, vamos visualizar como ficou na nossa camada do JAVA, que é o nosso FacadeAjax e DAO's, para facilitar a compreensão de como serão utilizado os acessos e chamadas dos nossos métodos, e saberemos também o tipo de retorno e como esta sendo criado para o envio ao nosso JS. Vamos utilizar os seguintes métodos para acessar o banco de dados.

Criamos uma função JS para limpar as grids, sendo a mesma genérica, recebendo apenas o nome do id do tdoby:

```
function cleanAddRows(local){
      DWRUtil.removeAllRows(local);
}
```

Vou listar abaixo os exemplos equivalentes a explicação das assinaturas do addRows(), e neles o respectivo código JSP e JS.

Exemplo 1 (Grid Simples):

JSP

```
Exemplo grid 1:
<a href="javascript:carregaGrid1()" >Carregar</a> /
<a href="javascript:cleanAddRows('Grid1')" >Limpar</a>
<table border="0">
```

```
	<tr bgcolor="#CCCCCC">
		<td>Nome</td>
		<td>Endereço</td>
		<td>Telefone</td>
		<td>Cidade</td>
	</tr>
	<tbody id="Grid1"></tbody>
</table>
```

JS

```
/**
 * Função que acessa o método da classe java(FacadeAjax), e retorna
 * como callback para a função montaXXX.
 */
function carregaGrid1(){
	FacadeAjax.selectAllPessoa(montaGrid1);
}

/**
 * Função de callback que limpa a grid e depois a remonta.
 * O processo para remover a grid não é obrigatório, vai depender da
 * sua lógica.
 * Se você não remover a grid ela vai sempre adicionar novos valores
 * somando os já existentes.
 *
 * @param {Object} list - Objeto de retorno do
 * FacadeAjax.selectAllPessoa()
 */
function montaGrid1(list){
	cleanAddRows("Grid1");

	var cellFuncs = [
			function(data) { return data.nome; },
			function(data) { return data.endereco; },
			function(data) { return data.telefone; },
			function(data) { return data.cidade; }
		];

	DWRUtil.addRows("Grid1", list, cellFuncs);
}
```

HTML Gerado:

Nome	Endereço	Telefone	Cidade
Christiano Milfont	Rua Rosas	32222222	Fortaleza
Handerson Frota	Rua Maximo	34444444	Fortaleza
Rafael Ponte	Rua Senhor dos Anéis	24242424	Fortaleza

Exemplo 2(Grid Avançado):

JSP

Exemplo grid 2:

```
<a href="javascript:carregaGrid2()" >Carregar</a> /
<a href="javascript:cleanAddRows('Grid2')" >Limpar</a>
<table border="0">
      <tr bgcolor="#CCCCCC">
            <td>Nome</td>
            <td>Endereço</td>
            <td>Telefone</td>
            <td>Cidade</td>
      </tr>
      <tbody id="Grid2"></tbody>
</table>
```

JS

```
/**
 * Função que acessa o método da classe java(FacadeAjax), e retorna
 * como callback para a função montaXXX.
 */
function carregaGrid2(){
      FacadeAjax.selectAllPessoa(montaGrid2);
}

/**
 * Função de callback que limpa a grid e depois a remonta.
 * O processo para remover a grid não é obrigatório, vai depender da
 * sua lógica.
 * Se você não remover a grid ela vai sempre adicionar novos valores
 * somando os já existentes.
 *
 * Monta as linhas e colunas alterando o seu layout.
 *
 * @param {Object} list - Objeto de retorno do
 * FacadeAjax.selectAllPessoa()
 */
function montaGrid2(list){
      cleanAddRows("Grid2");

      var cellFuncs = [
                        function(data) { return data.nome; },
                        function(data) { return data.endereco; },
                        function(data) { return data.telefone; },
                        function(data) { return data.cidade; }
                       ];

      DWRUtil.addRows( "Grid2", list, cellFuncs, {
            rowCreator:function(options) {
            var row = document.createElement("tr");
            row.bgColor = "#CCCC99";

            return row;
            },
            cellCreator:function(options) {
            var td = document.createElement("td");
                  td.align = "center";
```

```
                return td;
                }
        });
}
```

HTML Gerado:

Nome	Endereço	Telefone	Cidade
Christiano Milfont	Rua Rosas	32222222	Fortaleza
Handerson Frota	Rua Maximo	34444444	Fortaleza
Rafael Ponte	Rua Senhor dos Anéis	24242424	Fortaleza

Estão listados acima as principais assinaturas do addRows(). Podemos ver o quão fácil é gerar uma grid com DWR.

Para baixar esse projeto para executar os exemplos basta acessar o link abaixo:
http://www.milfont.org/blog/wp-content/upload/artigoDWR/versao2.2/PalestraDWR.rar

Handerson Frota
(handersonbf@gmail.com)