

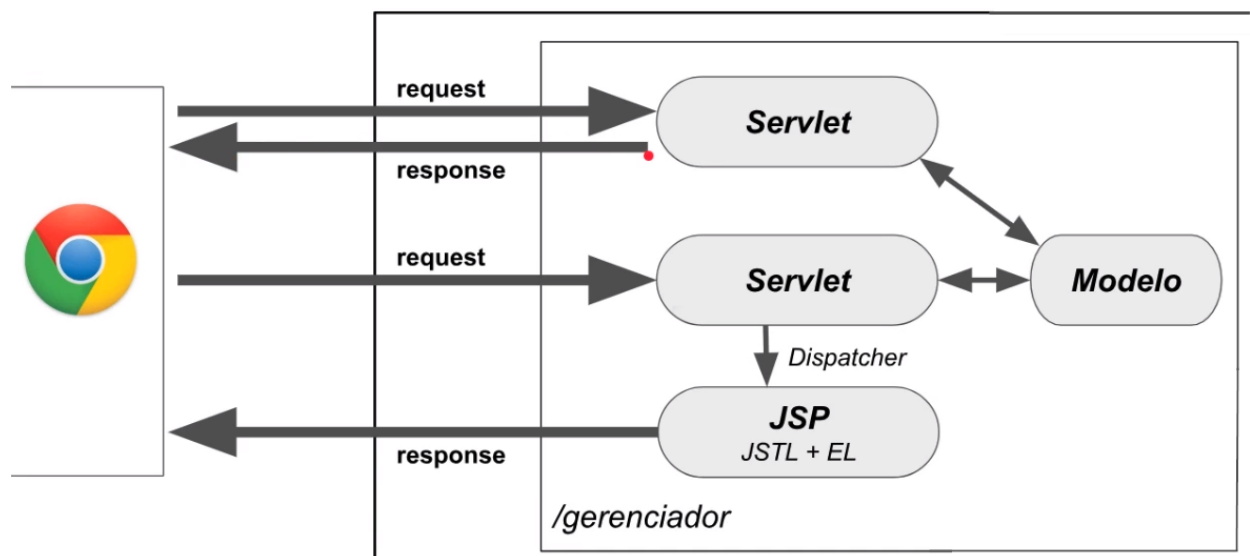


Redirecionamento pelo navegador

Transcrição

Temos um problema com o formulário: os Servlets foram chamados, o arquivo JSP e tudo isso em uma mesma requisição. Trata-se de uma má prática que gera problemas em nossa aplicação, como o cadastro de dados idênticos em nosso banco de dados.

Vamos entender melhor como isso ocorre. Teremos uma primeira requisição proveniente do formulário que chegará até o Servlet para realizar o cadastramento de uma empresa. Mas este Servlet, não mais chamará outro, e sim, devolverá uma resposta direta para o navegador. Contudo, essa resposta simples que indica um redirecionamento ao próprio navegador, indicando que este deve enviar uma segunda requisição para o Servlet que lista as empresas.



Portanto, quem realiza o redirecionamento não é o Servlet dentro do servidor, mas o navegador que enviará uma nova requisição a partir da resposta. O nome desse processo é **Redirecionamento Client Side**, isto é, "do lado do

cliente". Quando utilizamos despachadores de um Servlet para outro, esse processo é denominado **Redirecionamento Server Side**, "do lado do servidor".

O que precisamos fazer é modificar o Servlet e não mais utilizar o despachador, mas sim devolver uma resposta para o navegador, de forma que este envie uma requisição. Nosso objetivo é claro, mas precisamos estudar em como realizá-lo usando uma API Servlets, mas a mesma situação pode ocorrer em aplicações com Spring, V-Raptor e assim por diante.

Em `NovaEmpresaServlet`, nos atentaremos primeiramente para este trecho do código:

```
//chamar JPS
RequestDispatcher rd = request.getRequestDispatcher("/listaEmpres");
request.setAttribute("empresa", empresa.getNome());
rd.forward(request, response);
```

[COPIAR CÓDIGO](#)

Não poderemos mais utilizar o `RequestDispatcher`, portanto comentaremos todo o trecho destacado selecionando-o e utilizando o atalho "Ctrl + Shift + C". Precisamos devolver uma resposta, e para isso utilizaremos o termo `response`. Em seguida, teremos um método específico `sendRedirect()`. A ideia de que receber uma resposta é automaticamente enviar uma nova requisição já foi embutida no protocolo HTTP, por exemplo, ao tentarmos acessar um documento via protocolo HTTP, mas esse documento foi apagado por algum motivo e modificando em outro endereço, seremos notificados com o erro 404. Contudo, podemos avisar ao cliente desse novo endereço do arquivo devolvendo uma resposta a ele.

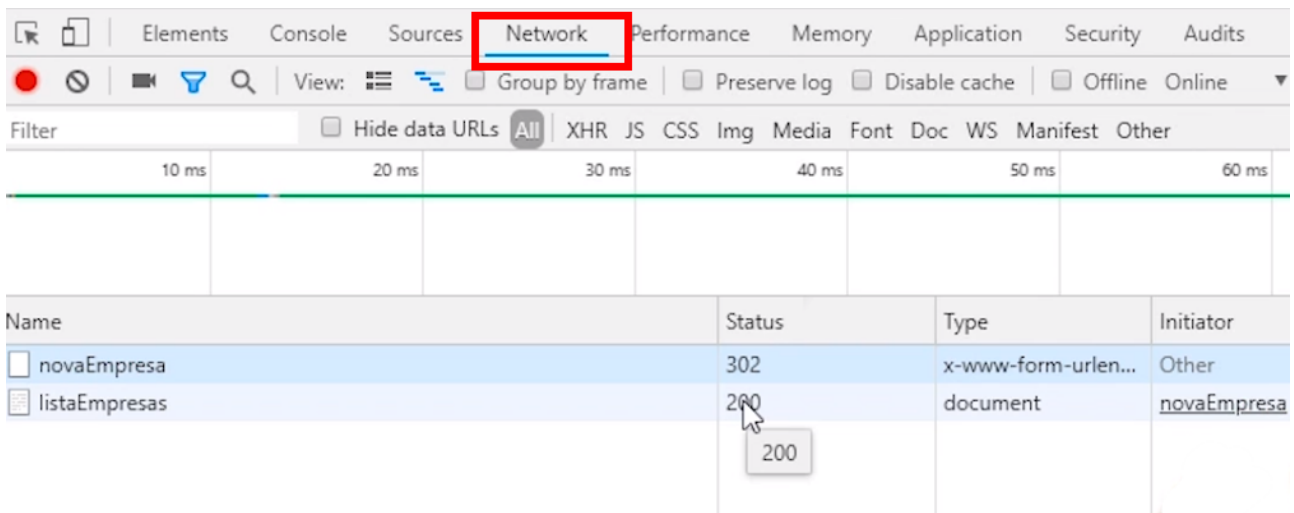
E exatamente esse movimento que faremos agora: devolveremos uma resposta com um novo endereço para o navegador, isto é, fazer um redirecionamento. O método `sendRedirect()` receberá `listaEmpresas`, sem o uso de `/`.

```
Banco banco = new Banco();  
banco.adiciona(empresa);  
  
response.sendRedirect("listaEmpresas");  
  
//chamar JPS  
//RequestDispatcher rd = request.getRequestDispatcher("/listaEr  
//request.setAttribute("empresa", empresa.getNome());  
//rd.forward(request, response);
```

[COPIAR CÓDIGO](#)

Assim feito, abriremos o navegador e recarregaremos o formulário com a URL `http://localhost:8080/gerenciador/formNovaEmpresa.jsp` . Em seguida, pressionaremos o atalho "F12" no Chrome para acessarmos a área de ferramentas do desenvolvedor. Na aba "Network" poderemos acessar todas as requisições realizadas, e assim, avaliar o que de fato foi enviado ou não. Vamos criar uma requisição: preencheremos o formulário com "Caelum" e data "01/01/2091" e enviaremos. Seremos redirecionados automaticamente para a lista de empresas, e podemos verificar a mudança da URL, que agora é `http://localhost:8080/gerenciador/listaEmpresas` , afinal o navegador enviou uma nova requisição e nos mostra para onde a última requisição foi enviada, no caso `listaEmpresas` .

Ao analisarmos a aba "Network" poderemos ver as duas requisições realizadas: `NovaEmpresa` e `ListaEmpresa` .



Podemos analisar cada uma delas com mais detalhes selecionando cada uma individualmente e clicando sobre a aba "Headers", em veremos o método a requisição, sua URL e outras informações. A requisição NovaEmpresa é do tipo POST, enquanto ListaEmpresa é do tipo GET.

Na área de ferramentas do desenvolvedor também podemos verificar o status de cada requisição. A requisição novaEmpresa apresenta o status 302, enquanto listaEmpresas apresenta 200. Aprendemos anteriormente que o status "200" significa que tudo ocorreu bem, mas existem vários outros códigos, como o "404", que significa que o recurso do servidor não foi encontrado. A categoria "302" significa redirecionamento, e poderemos inclusive verificar na aba "Headers" a localização desse redirecionamento na opção "Locations". Neste caso, teremos o redirecionamento para listaEmpresas .

A pressionarmos "F5" na lista de empresas, não teremos novos cadastros idênticos e será exibida a mensagem de reenvio do formulário. Contudo ainda temos um pequeno problema: ao cadastramos uma nova empresa utilizando o formulário, não conseguimos ver e mensagem de confirmação do cadastramento, apenas somos direcionados para a lista de empresas:

Lista de empresas:

.Alura - 24/08/2018

.Caelum - 24/08/2018

.Caelum - 01/01/2001

.Alura - 01/01/2001

[COPIAR CÓDIGO](#)

Quando cadastramos a empresa no primeiro Servlet, nós "penduramos" algo na requisição. Como havíamos comentado o trecho de código `request.setAttribute("empresa", empresa.getNome())`, em `NovaEmpresaServlet` a funcionalidade não foi executada. Iremos copiar esse trecho e torná-lo ativo novamente.

```
Banco banco = new Banco();
banco.adiciona(empresa);

/request.setAttribute("empresa", empresa.getNome());

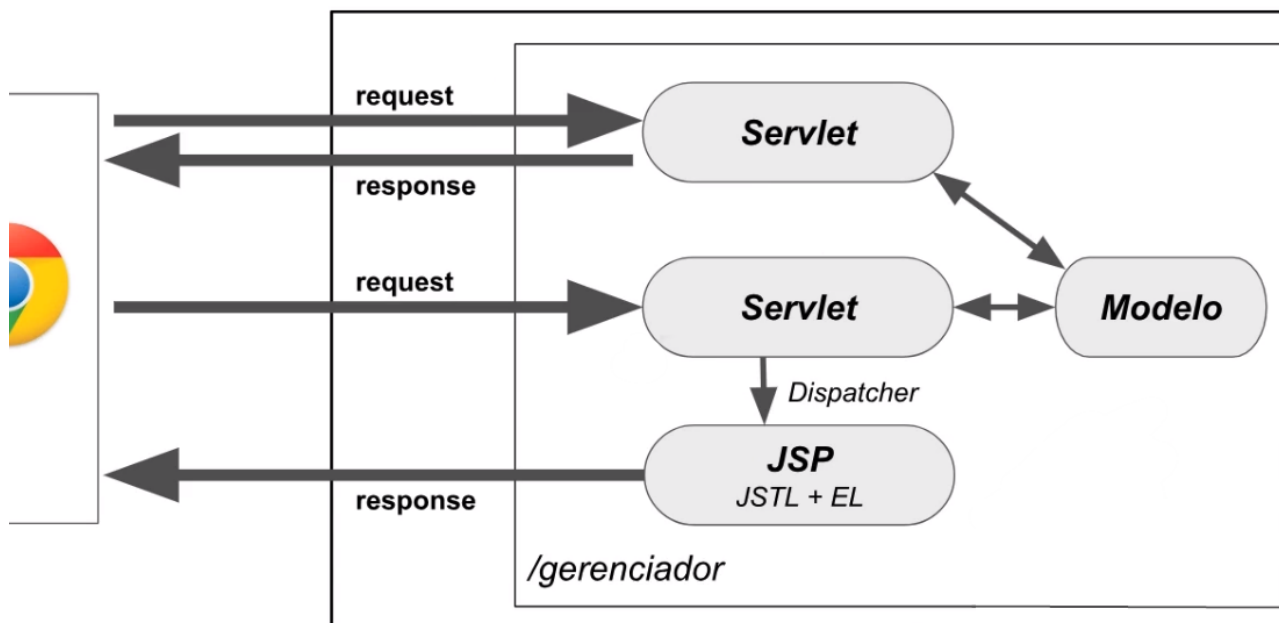
response.sendRedirect("listaEmpresas");

//chamar JPS
//RequestDispatcher rd = request.getRequestDispatcher("/listaEr
//request.setAttribute("empresa", empresa.getNome());
//rd.forward(request, response);
```

[COPIAR CÓDIGO](#)

Ao acessarmos o navegador, mesmo com as modificações realizadas e com a funcionalidade ativada, a mensagem continua não sendo exibida. Vamos entender o que está acontecendo.

Nós inserimos no Servlet o `request.setAttribute()`, contudo ele é válido para uma única requisição. Quando utilizamos o despachador, estamos trabalhando apenas com uma requisição que é transferida para outro Servlet até chegar o JSP e retornar uma resposta para o navegador. Contudo, agora, estamos utilizando duas requisições. Os objetos que "penduramos" dentro da requisição são mentidos apenas nela.



Veremos como resolver esse problema nas próximas aulas. Precisaremos de um escopo maior, isto é, apenas o escopo da requisição não servirá para este caso.