

Transcrição

Criamos um novo Servlet para cadastramento de empresas. Neste ponto, não realizamos o cadastro de fato, por enquanto apenas lemos o parâmetro enviado pelo servidor. Vamos melhorar tudo isso e avançar na construção no projeto, por enquanto o nosso foco é protocolo HTTP e o Servlet.

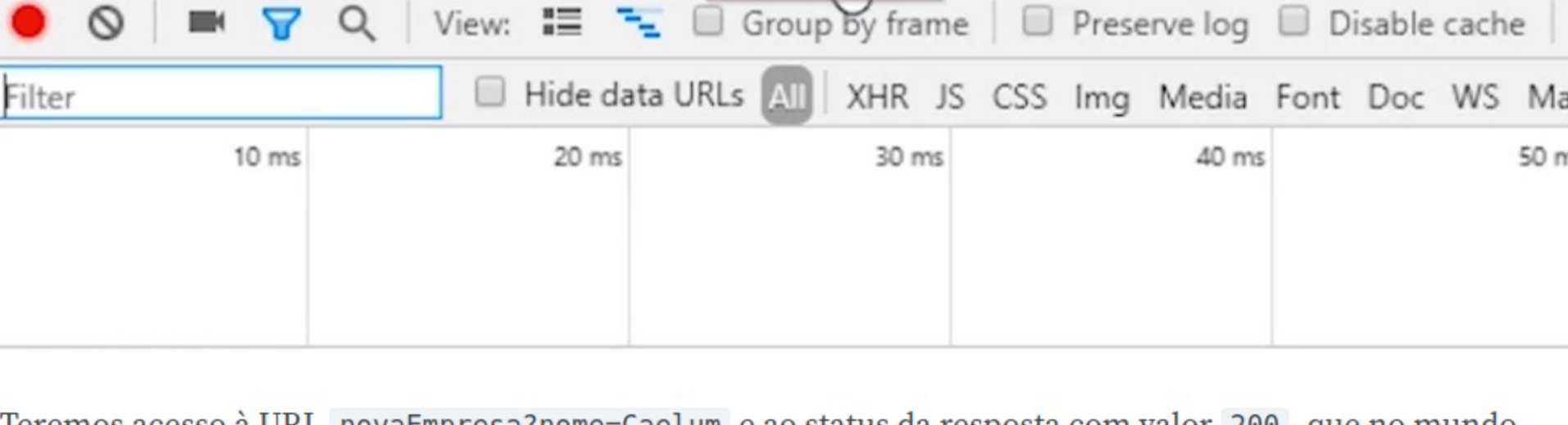
Nos atentaremos para uma questão importante: o parâmetro está visível na URL.

```
localhost:8080/gerenciador/novaEmpresa?nome=Caelum
```

Se tivermos muitos parâmetros precisaremos enviar um texto enorme na URL? O que nos leva a pensar que, talvez a melhor forma de enviar um parâmetro não seja via URL. Em alguns casos, precisaremos efetuar o login em algum site, como na **Alura** para acessar a área de alunos. Neste momento, é enviado pelo servidor um e-mail e senha, e esses parâmetros não devem ser passados via URL, afinal eles estarão visíveis.

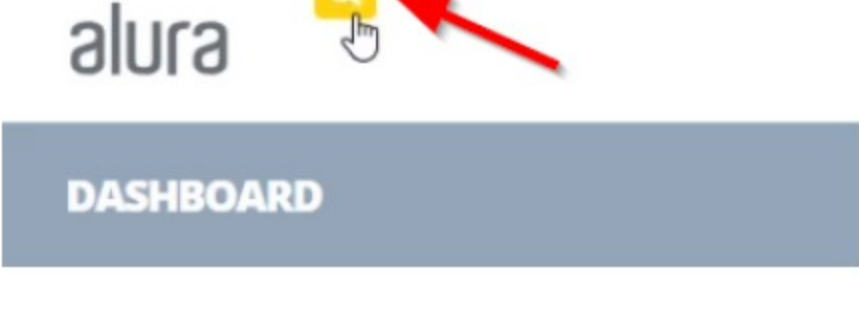
No Chrome, ao clicarmos no menu de ferramentas, encontraremos a opção "Mais ferramentas > Ferramentas do desenvolvedor", ou simplesmente usar o atalho "F12". Teremos acesso a algumas abas que nos revelam como o navegador trabalha internamente, são informações importantes para desenvolvedores front-end que lidam com HTML, CSS e JavaScript,por exemplo.

Essas informações são também valiosas para nós que estamos desenvolvendo um aplicação web. Clicaremos sobre a aba "Network" dessa nova área do desenvolvedor.



Teremos acesso à URL `novaEmpresa?nome=Caelum` e ao status da resposta com valor `200` , que no mundo HTTP significa que tudo ocorreu bem, e algumas outras informações que não comentaremos agora. Clicaremos sobre a a URL e acessaremos mais dados sobre o protocolo HTTP, e veremos que ao enviar uma requisição, na verdade estamos enviando uma série de conteúdos sobre o navegador, o endereço específico e assim por diante. Teremos uma informação importante no campo "Request Method", que apresenta o método "GET". *Get*, significa "receber", em inglês, a ideia é de que enviamos a requisição para o servidor para recebermos informações.

Para ficar mais clara a ideia, abriremos a página da **Alura**, já na área do aluno, isto é, pós-login. Teremos um mecanismo de busca na parte superior da tela, simbolizado pelo ícone de lupa.



Ao clicarmos neste ícone, poderemos pesquisar pelo termo "Java" e encontraremos os cursos desse campo. Raparem que na URL depois de submetermos o formulário com o item buscado, teremos o termo `search` e o parâmetro `query` e valor `java` .

```
https://cursos.alura.com.br/search?query=java
```

Poderíamos realizar uma busca digitando diretamente um valor na URL, como `javascript` , e o resultado seria os cursos desta área. A ideia do método é justamente essa: enviar informações e receber resultados. No caso, o resultado está relacionado à pesquisa de cursos da linguagem Java, mas poderia ser um documento, gráfico ou qualquer outro item.

O que estamos idealizando para nosso Servlet é algo diferente: estamos enviando informações para serem cadastradas, e altarmos o dado no servidor. Para isso, não deveríamos usar o método "GET", existe outro método mais adequado para as intenções que temos no projeto, e este método é chamado de "POST".

"POST" posta informações, enviar dados que alterem algo no servidor. Como podemos enviar uma requisição dessa natureza? Primeiramente, precisamos criar um formulário dentro da pasta `WebContent` do projeto `gerenciador` . Clicaremos com o botão direito sobre a pasta `WebContent` e criaremos um "HTML File", que chamaremos de `formNovaEmpresa.html` . O Eclipse criará automaticamente uma estrutura padrão

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
</body>
</html>
```

Para criarmos o formulário usaremos uma tag `<form>` , que possui uma *action* que define na URL para onde será enviada a requisição ao submeter o formulário. A URL, neste projeto, será sempre definida a partir da porta `localhost:8080` , portanto escreveremos na tag `<form>` : `/gerenciador/novaEmpresa` .

Em seguida, criaremos os campos do formulário. O primeiro campo será `Nome:` , e o faremos por meio do elemento `<input>` , e o `type` nos ajudará a definir o tipo de input que queremos utilizar, no caso, usaremos `text` . Além disso, o input deve definir qual o nome do parâmetro utilizando o atributo `name` . Na URL o parâmetro possui um nome, portanto precisamos defini-lo de alguma maneira. No caso, o atributo será `nome` .

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <form action="/gerenciador/novaEmpresa">

        Nome: <input type="text" name="nome">

    </form>

</body>
</html>
```

Podemos escrever a URL `localhost:8080/gerenciador/formNovaEmpresa.html` no navegador, teremos acesso ao campo e podemos escrever nele, contudo ainda resta criarmos um botão que nos permite submeter o formulário. Usaremos novamente um `<input>` do tipo `submit` .

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <form action="/gerenciador/novaEmpresa">

        Nome: <input type="text" name="nome" />

        <input type="submit" />
    </form>

</body>
</html>
```

Ao atualizarmos o navegador, veremos disponível o formulário com um único campo "Nome" e o botão "Enviar". No navegador escreveremos no campo a empresa "Caelum", e essa informação será enviada para o Servlet. No entanto, veremos na URL o nome da empresa:

```
localhost:8080/gerenciador/novaEmpresa?nome=Caelum
```

Isso significa que foi enviada uma requisição "GET", e devemos modificar essa característica e enviar uma requisição "POST" inserindo essa especificidade na declaração inicial do formulário utilizando `method="post"` .

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <form action="/gerenciador/novaEmpresa" method="post">

        Nome: <input type="text" name="nome" />

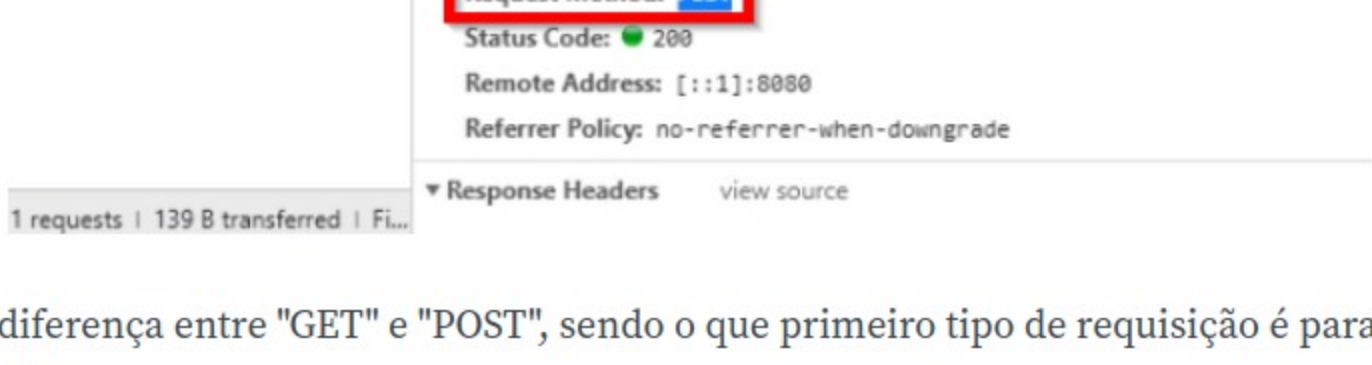
        <input type="submit" />
    </form>

</body>
</html>
```

Recarregaremos o navegador e escreveremos "Alura" no campo "Nome". O Servlet foi chamado, mas **não** teremos na URL o parâmetro `Alura` .

```
localhost:8080/gerencidor/novaEmpresa
```

O servidor conseguiu ler a informação enviada, pois teremos a mensagem `Empresa Alura` cadastrada com sucesso . Podemos confirmar a implementação da requisição "POST" observando a aba "Network" em "Área do Desenvolvedor", campo "Request Method".



Aprendemos a diferença entre "GET" e "POST", sendo o que primeiro tipo de requisição é para enviar parâmetros na URL, mas o que pode gerar problemas quando possuímos uma quantidade grande de parâmetros. O "POST" esconde os parâmetros dentro do corpo da requisição.