



## Metodos GET e POST

### Transcrição

Anteriormente, nós criamos um Servlet para cadastramento de empresas. Neste ponto, não realizamos o cadastro de fato, apenas lemos o parâmetro enviado pelo servidor - ou seja, se reiniciamos o servidor, esses dados são perdidos, já que não são persistentes. Futuramente melhoraremos esses pontos, mas nosso foco por enquanto é o protocolo HTTP e o nosso Servlet.

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

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

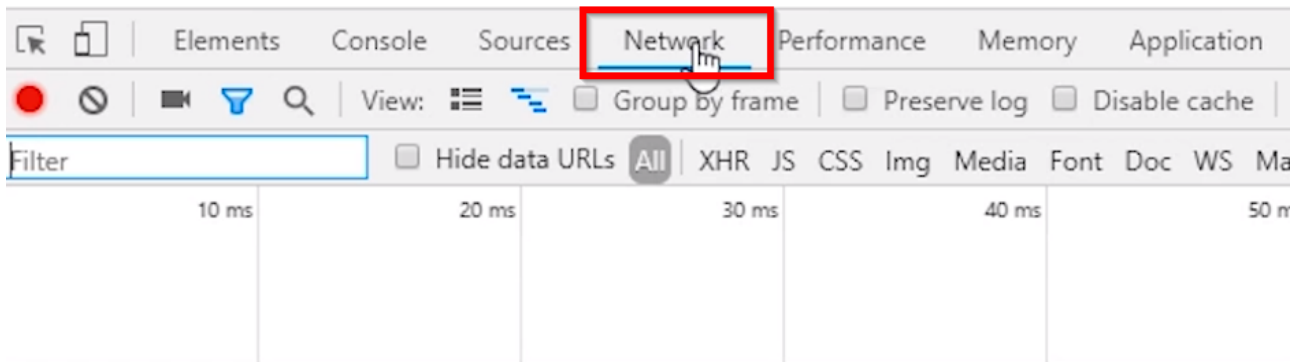
Mas e se tivermos muitos parâmetros ou precisarmos enviar um texto enorme para o servidor? Isso certamente atrapalharia a construção da nossa URL. Por isso, talvez a melhor forma de enviar um parâmetro não seja via URL.

Em alguns casos, precisaremos efetuar o login em algum site, como na **Alura**. Neste momento, normalmente é enviado para o servidor, na requisição, um e-mail e uma senha - dados que não queremos que fiquem visíveis na URL. E existe outra forma de enviar parâmetros!

No **Chrome**, ao clicarmos no menu de ferramentas, encontramos a opção "Mais ferramentas > Ferramentas do desenvolvedor" (que também pode ser acessada pelo atalho "F12" ou "Ctrl + Shift + L"). Outros navegadores, como o **Firefox** e o **Edge**, também possuem essa opção.

Nas "Ferramentas do desenvolvedor", temos acesso a algumas abas que nos revelam como o navegador trabalha internamente, informações importantes para desenvolvedores front-end que lidam com HTML, CSS e JavaScript, por exemplo, e para desenvolvedores de aplicações web.

Como queremos ver informações sobre a requisição HTTP e a comunicação com a rede, acessaremos a aba "*Network*".

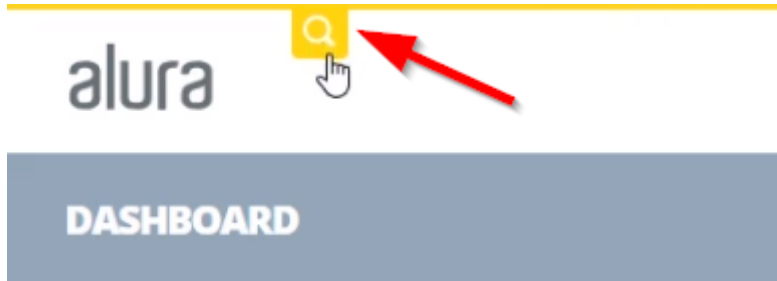


Acessando novamente a url <http://localhost:8080/gerenciador/novaEmpresa?nome=Caelum> (<http://localhost:8080/gerenciador/novaEmpresa?nome=Caelum>), receberemos nessa aba a requisição `novaEmpresa?nome=Caelum` e o status da resposta com valor `200`, que no mundo HTTP significa que tudo ocorreu bem, além de outras informações que não comentaremos agora.

Clicando sobre a requisição, teremos acesso a ainda mais dados sobre os cabeçalhos do protocolo HTTP. Na verdade, em uma requisição, estamos enviando uma série de informações sobre o navegador, o endereço específico e assim por diante.

Por exemplo, temos uma informação importante no campo "Request Method", que apresenta o método "GET". *Get*, significa "receber", em inglês, e a ideia é que enviamos a requisição para que o servidor receba informações.

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



Ao clicarmos neste ícone, poderemos pesquisar pelo termo "Java", recebendo resultados que correspondem a esse termo. Raparem que na URL, depois de submetermos o formulário com o item buscado, teremos o termo `search`, o parâmetro `query` e valor `java`.

<https://cursos.alura.com.br/search?query=java>  
(<https://cursos.alura.com.br/search?query=java>).

Podemos até realizar uma nova busca digitando diretamente um valor na URL, como `javascript`, e a plataforma retornará os itens correspondentes a esse termo.

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, um gráfico ou qualquer outro item.

O que estamos idealizando para nosso Servlet é algo diferente: enviaremos informações para serem cadastradas, alterando os dados no servidor. Para isso, não deveríamos usar o método "GET", já que existe outro método mais adequado para essa situação: o POST.

O método "POST" serve para enviar informações que alterem os dados no servidor. E como podemos enviar uma requisição dessa natureza?

Primeiramente, precisaremos criar um formulário dentro da pasta "WebContent" do projeto **gerenciador**. Clicaremos com o botão direito sobre a pasta "WebContent", 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>
```

[COPIAR CÓDIGO](#)

Criaremos o formulário usando a tag `<form></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 `/gerenciador/novaEmpresa` (o endereço que receberá os dados).

Em seguida, criaremos os campos do formulário. O primeiro campo será Nome: , e o faremos por meio do elemento `<input>` . Esse elemento possui um `type` que ajudará a definir o tipo de *input* que queremos utilizar, que nesse caso é `text` . Além disso, o `input` deve definir qual o nome do parâmetro, o que é feito utilizando o atributo `name` . Como na URL o parâmetro possui um `nome` , o atributo `name` receberá esse mesmo valor.

```
<!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>
```

[COPIAR CÓDIGO](#)

Podemos acessar a URL

<http://localhost:8080/gerenciador/formNovaEmpresa.html>  
(<http://localhost:8080/gerenciador/formNovaEmpresa.html>) no navegador para  
testar, mas ainda precisamos de um botão para submeter o formulário. Para  
criarmos um, usaremos novamente um `<input>` , dessa vez 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>
```

[COPIAR CÓDIGO](#)

Ao atualizarmos o navegador, teremos um formulário com um único campo "Nome" e o botão "Enviar". Na caixa de texto, escreveremos o nome da empresa "Caelum" e clicaremos em "Enviar" para enviá-la ao Servlet. Receberemos a mensagem "Empresa Caelum cadastrada com sucesso!", mas o nome da empresa continuará visível na URL:

localhost:8080/gerenciador/novaEmpresa?nome=Caelum

Isso significa que foi enviada uma requisição "GET", e devemos modificar essa característica para enviar uma requisição "POST", inserindo essa especificidade na declaração inicial do formulário com `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>

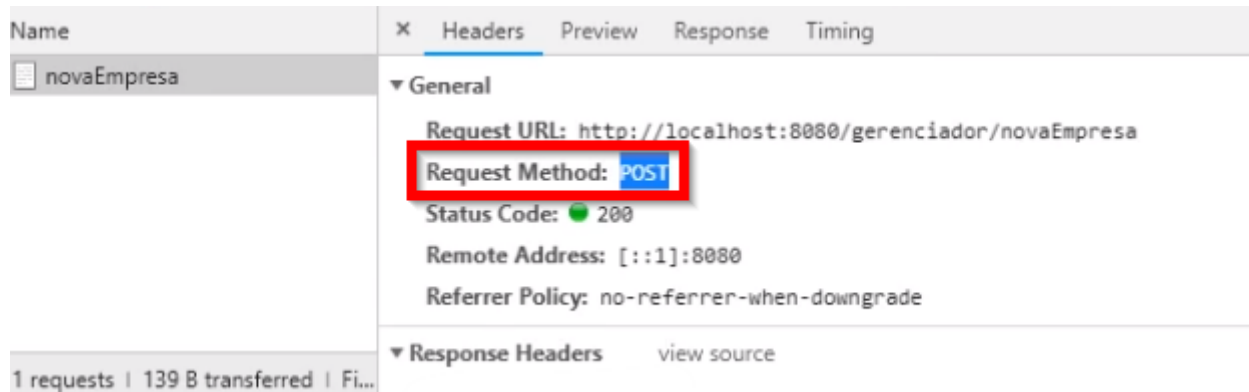
</body>
</html>
```

COPIAR CÓDIGO

Recarregando a página no navegador, podemos fazer o teste novamente, dessa vez com o nome "Alura". Ao enviarmos a requisição, nosso Servlet será executado corretamente, mas o parâmetro não será exibido na URL.

localhost:8080/gerenciador/novaEmpresa

Podemos confirmar a implementação da requisição "POST" observando a aba "Network" em "Área do Desenvolvedor", no campo "Request Method".



Deve ter ficado claro que o método "GET" é utilizado para enviar parâmetros na URL, e que ele é problemático quando estamos trabalhando com parâmetros muito longos. Além disso, nem sempre queremos mostrar esses parâmetros.

Já o método "POST" esconde os parâmetros no corpo da requisição, sem mostrá-los na URL. Confira os exercícios e até o próximo vídeo!