

01

## Montando URLs

### Transcrição

Outro grande problema do mundo de desenvolvimento web é gerar URLs. URLs são sempre relativas ao lugar atual que elas estão. Por exemplo, se a URL atual do site for `www.meusite.com.br/produto/lista`, e tivermos um link que aponta para `novo`, essa URL é relativa ao caminho atual, e ao clicar no link, a nova URL será `www.meusite.com.br/produto/novo`.

Para resolver esse problema, o correto é sempre gerarmos URLs completas para nossa aplicação. Mas isso também pode ser difícil, já que no mundo Java é bem comum o "contexto" mudar. Contexto é como chamamos o "começo da URL", que diz ao Tomcat qual projeto deve rodar; no nosso caso, é "produtos". Se colocarmos na URL, por exemplo, `/produtos/produto/novo`, ela funcionaria. O problema é que se "produtos" mudar, precisaremos mudar em todos os lugares.

Por esse motivo, a melhor maneira para se gerar uma URL é por meio da JSTL. Ela saberá sempre como montar a URL certa! Veja o HTML que temos, que gera a URL para a página de novo produto:

```
<a href="/produtos/produto/formulario">Adicionar um produto</a>
```

Vamos fazer agora com que esse link seja gerado de maneira inteligente. A tag que faz isso é a tag `c:url`. Ela recebe um link e o nome de uma variável que conterá o link gerado:

```
<c:url value="/produto/formulario" var="urlAdicionar" />
<a href="${urlAdicionar}">Adicionar um produto</a>
```

Repare que a URL que montamos começa com uma "", e é relativo a raiz do projeto. Se você olhar o HTML gerado, verá que a URL será `/produtos/produto/formulario`, pois a JSTL percebeu que existe um "context root" e automaticamente o colocou na URL!

Use sempre `c:url` para gerar seus links. Ele é extremamente útil quando esses links estão em componentes reutilizáveis. Aí, dependendo da página que você coloca o componente, a URL é diferente. Com a JSTL, isso deixa de ser um problema.

