02

Rotas e views: criando um formulário

Uma grande facilidade trazida pelo conjunto Activator-Play! é o comando "~", que escuta por modificações nos arquivos do projeto. Em conjunto com subir o servidor (`~run`), temos um sistema que ao identificar uma alteração em um arquivo, recompila e atualiza o servidor, economizando bastante tempo que passaríamos fazendo o mesmo manualmente.

Agora mão na massa! O fundamento de uma loja são os produtos vendidos nela, então vamos criar um formulário de cadastro de produtos. Faremos isso substituindo o conteúdo da página principal, então abra o arquivo `app/views/index.scala.html` . Repare que existe um bloco de código chamado **main**. Substitua o conteúdo desse bloco pelo formulário, com o arquivo final assim:

```
@(message: String)
@main("Cadastro de produto") {
    <form action="/produto/novo" method="post">
        <h1>Cadastrar novo produto</h1>
        <label for="titulo">Título</label>
        <input type="text" name="titulo" id="titulo">
        <label for="codigo">Código</label>
        <input type="text" name="codigo" id="codigo">
        <label for="tipo">Tipo</label>
        <input type="text" name="tipo" id="tipo">
        <label for="descricao">Descrição</label>
        <textarea name="descricao" id="descricao"></textarea>
        <label for="preco">Preço</label>
        <input type="number" name="preco" id="preco">
        <input type="submit" value="Cadastrar">
    </form>
}
```

Agora já temos um formulário na página principal! Mas ao enviá-lo, recebemos um erro de *rota não encontrada*, juntamente com uma listagem das rotas existentes. O que precisamos fazer é criar uma nova rota, atrelando a ação do formulário a uma lógica dentro do nosso aplicativo. Para isso, abra o arquivo `conf/routes` , remova os comentários e rotas que não utilizaremos, e adicione uma nova rota, terminando com o seguinte conteúdo:

```
GET  /             controllers.HomeController.index
POST /produto/novo controllers.ProdutoController.salvaNovoProduto
GET  /assets/*file controllers.Assets.versioned(path="/public", file: Asset)
```

Isso porém gera um erro de compilação, já que o `ProdutoController` não existe. Vamos criá-lo então!

```
package controllers;
import play.mvc.*;
public class ProdutoController extends Controller {
    public Result salvaNovoProduto() {
        return ok("Formulário recebido!");
    }
}
```

Ao reenviar o formulário, vemos a mensagem escrita acima! Agora vamos aproveitar e mover a lógica do formulário para lá também. Altere o nome do arquivo `app/views/index.scala.html` para `app/views/formularioDeNovoProduto.scala.html`. Depois, remova a primeira entrada do arquivo `conf/routes`, substituindo por nossa nova rota para o formulário do **Produto**. Esse é o resulado final:

```
GET  /produto/novo controllers.ProdutoController.formularioDeNovoProduto
POST /produto/novo controllers.ProdutoController.salvaNovoProduto
GET  /assets/*file controllers.Assets.versioned(path="/public", file: Asset)
```

Já que não usamos mais o **HomeController**, vamos apagá-lo. Se quiser, remova também as outras classes listadas a seguir pois também não as usaremos no projeto.

```
app/controllers/HomeController.java
                AsyncController.java
                CountController.java

test/ApplicationTest.java
     IntegrationTest.java
```

No **ProdutoController**, crie o método que renderiza o formulário. A mensagem passada se deve a um parâmetro que veremos em breve.

```
public Result formularioDeNovoProduto() {
    return ok(formularioDeNovoProduto.render("Cadastro de produto"));
}
```

Como último passo, ainda no **ProdutoController** precisamos fazer a importação das nossas views. Adicione o import no início do arquivo, após a importação de play.mvc.*.

```
import views.html.*;
```

Agora, se você acessar a página localhost:9000/produto/novo (http://0.0.0.0:9000/produto/novo), já consegue visualizar e enviar o formulário!