

03

Enviando aluno para o Java

Transcrição

Com o formulário e as rotas funcionando corretamente, vamos pausar um pouco para entender o que está realmente acontecendo na aplicação, e como transformaremos a submissão do formulário para o Java.

Primeiro, o *controller* `AlunoController` recebe a requisição, criando um objeto `Aluno` e retornando-o juntamente com a *view* `Cadastrar.html` apelidando este objeto como `aluno`. Porém, o HTML em si não entende Java, então como o formulário populará o objeto `aluno`? É aqui que entra o **Thymeleaf**.

O *Thymeleaf* é o *template engine* do Spring, responsável por esta relação do HTML com os objetos Java, na página de cadastro. Descrevemos que estaremos usando tags do *Thymeleaf* quando usamos o atributo `xmlns:th="http://www.thymeleaf.org"` na tag `<html>`. A partir deste momento, poderemos usar atributos do *Thymeleaf* nas tags HTML.

O objeto `aluno` passa a ser utilizado quando abrimos o formulário, e nele existe o atributo `th:object="${aluno}"`. A partir deste ponto, nos inputs do formulário, utilizaremos o atributo `th:field` para nos referirmos a cada campo/atributo do objeto `aluno` (`th:field="*{nome}"`, por exemplo). Se prestarmos atenção, os campos para notas e habilidades inexistem. Estes são mais complexos, por serem listas, e serão feitos separadamente.

Ainda sobre o formulário, note que o atributo `action` não faz parte do HTML em si, é o próprio *Thymeleaf* que mapeia a submissão do formulário dessa vez, quando clicarmos em `/aluno/salvar` (`th:action="@{/aluno/salvar}"`), com o objeto `aluno` populado com os dados do formulário.

Para testarmos seu funcionamento, criaremos o método `salvar` na classe `AlunoController`, a receber o objeto `aluno`, imprimindo-o no console. Este método receberá requisições do tipo *post* no mapeamento `/aluno/salvar`, e fará um redirecionamento para a página inicial da aplicação.

```
@PostMapping("/aluno/salvar")
public String salvar(@ModelAttribute Aluno aluno){
    System.out.println(aluno);
    return "redirect: /";
}
```

Note que para recebermos o objeto `aluno` enviado pelo formulário, precisaremos utilizar a anotação `@ModelAttribute`.

Depois disto, poderemos testar o cadastro da Maria, por exemplo, nascida em 10 de Abril de 2017, com curso de Administração.

No console, devemos ver uma saída semelhante a que se encontra abaixo:

```
br.com.alura.escolalura.models.Aluno@574c958a
```

Isso indica que, com formulário preenchido, tivemos a criação do objeto `aluno` efetivada, o qual chegou até nosso método `salvar`. Nosso próximo passo é fazer com que esse objeto seja salvo no MongoDB.