04

Transformando os itens em JSON

Transcrição

O próximo passo é implementar o método `toJson()`. Para isso precisamos pegar um *builder*:

```
public String toJson() {
    Json.createArrayBuilder();
    return "{}"
}
```

Vamos criar a classe Json. Mas para isso adicionamos uma nova dependência no `pom.xml`:

```
<dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-api</artifactId>
    <version>7.0</version>
    <scope>provided</scope>
</dependency>
```

No nosso `import` inicial já vieram diversas dependências que precisaremos mudar no decorrer do projeto.

No método `toJson()` apertamos CTRL + SHIFT + O para importar o código fonte, dentro do arquivo `Json.class`. O próximo passo é colocar o `ArrayBuilder` dentro de uma variável `builder` e ir populando-o:

```
public Atring toJson() {
    JsonArrayBuilder builder = Json.createArrayBuilder();
    for (CarrinhoItem item : itens) {
        builder.add(Json.createObjectBuilder()
                .add("titulo", item.getLivro().getTitulo())
                .add("preco", item.getLivro().getPreco())
                .add("quantidade", item.getQuantidade())
                .add("total", getTotal(item)));
    }
    return "{}"
}
```

Temos basicamente tudo que precisamos para o nosso *builder*. Falta retornarmos o Json como String:

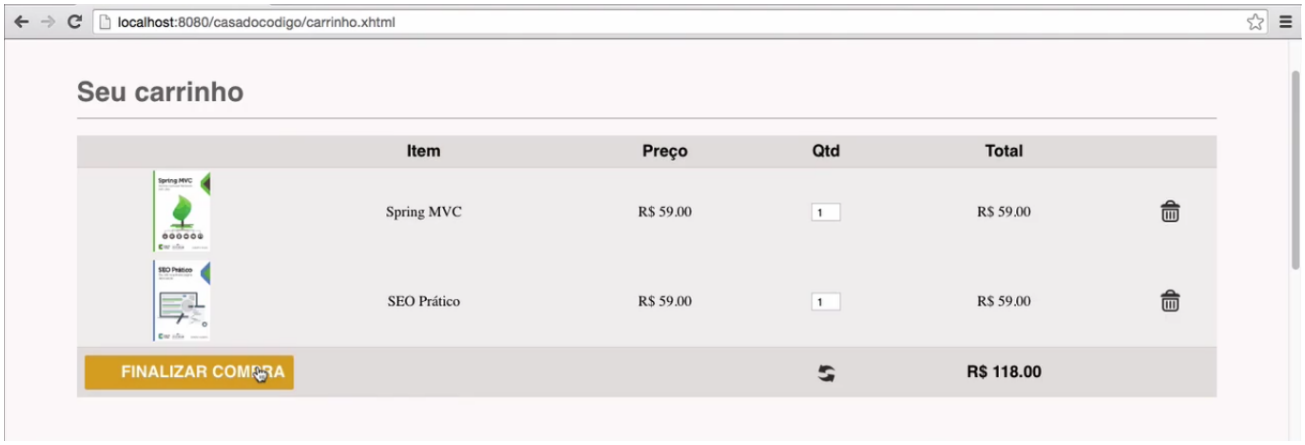
```
    return builder.build().toString();
```

Mas vamos observar este objeto Json antes, pedindo para imprimí-lo

```
    String json = builder.build().toString();
    System.out.println(json);

    return json;
```

Façamos novamente a simulação adicionando itens ao Carrinho:



Passamos por todo o processo de finalizar a comprae e vejamos o console:

```
[{"titulo":"Spring MVC","preco":59.00,"quantidade":1,"total":59.00},{"titulo":"SEO Prático","pre
```

De fato, o Json está funcionando e foi impresso para que possamos observá-lo.

A API do Json é muito manual. Uma vez que criamos o builder, implementamos o `.add` item a item dele. Isso acontece porque ainda não foi lançado um *autobiding* do Java EE para Json. Esta nova feature provavelmente virá com o Java EE 8. Mesmo assim, vale a pena aprender a JSR-353 e fazermos esse teste.