02

Trabalhando com parâmetros

Nós temos um recurso que está disponível na WEB, podemos acessá-lo através da URL `http://localhost:8080/carrinhos`, mas no momento ele sempre trará o carrinho de `id 1`. Queremos poder escolher qual carrinho trazer.

Estamos usando o método `HTTP GET`, então podemos receber um parâmetro pela URL, algo do tipo `http://localhost:8080/carrinhos?id=1`. Para isso, também precisamos tratar no lado do servidor, vamos receber este parâmetro no `resource`.

```
@Path("carrinhos")
public class CarrinhoResource {

    @GET
    @Produces(MediaType.APPLICATION_XML)
    public String busca(long id) {
        Carrinho carrinho = new CarrinhoDAO().busca(1l);
        return carrinho.toXML();
    }
}
```

Precisamos também indicar para o JAX-RS que este id veio para nós através da URL. Fazemos isso através da anotação `@QueryParam` e passamos o parâmetro para o método `busca`, deixando nosso método do seguinte modo:

```
    @GET
    @Produces(MediaType.APPLICATION_XML)
    public String busca(@QueryParam("id") long id) {
        Carrinho carrinho = new CarrinhoDAO().busca(id);
        return carrinho.toXML();
    }
```

Levantamos nosso servidor e ao acessar a url `http://localhost:8080/carrinhos?id=1`, vemos que tudo continua funcionando, mas se tentamos passar outro id, que não existe, como por exemplo `http://localhost:8080/carrinhos?id=13`, recebemos uma `exception` em nosso console. Mais para frente veremos como tratar isto em nossa API Rest.

Mas ainda está estranho, passar o parâmetro diretamente na URL, pois iremos perder a capacidade de usar o `cache`, já que o navegador não utiliza `cache` em `urls` com parâmetro.

Seria melhor se fizéssemos com que a `uri` tivesse o `id` do carrinho diretamente nela, ficando do seguinte modo `http://localhost:8080/carrinhos/1`.

Para que isto ocorra, vamos alterar o nosso método `busca`, no `CarrinhoResource` e adicionar uma anotação, `@Path`. Ela irá indicar que iremos receber o `id` através da `uri`, ficando assim:

```
    @Path("{id}")
    @GET
    @Produces(MediaType.APPLICATION_XML)
    public String busca(@QueryParam("id") long id) {
```

```
        Carrinho carrinho = new CarrinhoDAO().busca(id);
        return carrinho.toXML();
    }
```

Agora ainda temos mais uma alteração, precisamos avisar que o `id` não é mais um `QueryParam`, agora ele é um `PathParam`:

```
    @Path("{id}")
    @GET
    @Produces(MediaType.APPLICATION_XML)
    public String busca(@PathParam("id") long id) {
        Carrinho carrinho = new CarrinhoDAO().busca(id);
        return carrinho.toXML();
    }
```

Vamos arrumar nossa `uri` no teste:

```
@Test
    public void testaQueBuscarUmCarrinhoTrasUmCarrinho() {
        Client client = ClientBuilder.newClient();
        WebTarget target = client.target("http://localhost:8080");
        String conteudo = target.path("/carrinhos/1").request().get(String.class);
        Carrinho fromXML = (Carrinho) new XStream().fromXML(conteudo);
        Assert.assertEquals("Rua Vergueiro 3185, 8 andar",fromXML.getRua());
    }
```