06

Disponibilizando livros XML

Transcrição

Uma vez que nosso retorno JSON está funcionando, podemos evoluir nossa aplicação. Muitos sistemas preferem XML ao JSON (sem fazermos julgamento de valores aqui). O ponto importante é que, se nosso sistema que se integrar com a maior parte dos sistemas possíveis, é importante que ele se comunique também via XML.

Voltando para nosso código, nós fizemos toda a configuração para que o sistema retornasse JSON. Façamos o mesmo para XML:

- `ultimosLancamentosJson()` vira `ultimosLancamentosXml()`
- `APPLICATION_JSON` vira `APPLICATION_XML`

E a url não pode ficar igual, então mudemos as duas, tanto para JSON quanto para XML:

```
@GET
@Path("json")
@Produces({MediaType.APPLICATION_JSON})
public List<Livro> ultimosLancamentosJson() {
    return dao.ultimosLancamentos();
}

@GET
@Path("xml")
@Produces({MediaType.APPLICATION_XML})
public List<Livro> ultimosLancamentosXml() {
    return dao.ultimosLancamentos();
}
```

Trocamos a url inteira porque se formos em `/livros/json` já temos todos os lançamentos como padrão. O mesmo acontece para XML. Isso foi uma escolha, mas você pode utilizar outros nomes para as urls em JSON e XML, apenas lembre de sempre utilizar o mesmo em outros lugares.

Reiniciemos o Wildfly e no Postman podemos repetir a tarefa anteiror olhando para

- http://localhost8080/casadocodigo/services/livros/json (http://localhost8080/casadocodigo/services/livros/json) (já tínhamos testado)
- http://localhost8080/casadocodigo/services/livros/xml (http://localhost8080/casadocodigo/services/livros/xml) (não irá funcionar)

O status que retorna ao tentarmos com o XML é de *406 Not Acceptable*. Se voltarmos no console da aplicação podemos perceber uma Exception:

```
ERROR [org.jboss.resteasy.resteasy_jaxrs.i18n] (default task-3) RESTEASY002010: Failed to execut
```

Este erro é referente ao cabeçalho (header). Então voltemos ao Postman e no cabeçalho colocamos para aceitar XML e content-type também:

Vai dar o erro *500 Internal Server Error*:

Isso acontece porque não limpamos o *header* com o JSON. O JAX-RS, que é a especificação que estamos utilizando, tenta fazer uma transformação automática do objeto. Com o JSON funcionou, com o XML não, pois existe outra especificação que trata da transformação para XML, a JAXB. Essa especificação diz que um objeto precisa estar mapeado como XML Root Element. Esta é a *anotation* que precisamos colocar em cima do objeto.

Então, em `Livro.java` precisamos anotar a classe:

```
@Entity
@Cacheable
@XmlRootElement
public class Livro {

    //...

}
```

O *import* do Xml Root Element é o ´javax.xml.bind.annotation.XMLRootElement`. Essa anotação do JAXB diz que o livro será um elemento raíz do XML, ou seja, iremos começar e fechá-lo assim:

```
<livro>
...
</livro>
```

E tudo dentro serão as informações. O único problema é que a aplicação não está pedindo que o livro seja raíz, mas sim `java.util.ArrayList` , como vimos na Exception. Não dá para abrir a classe `ArrayList` e colocar o Xml Root Element nela.

Existem duas soluções. Se você está utilizando o mesmo framework que nós, assim como a implementação do JAX-RS do Wildfly default, que é a RESTEasy, então não precisará fazer nada. Por si só o RESTEasey faz a transformação do livro em uma lista de livros, mas ainda sim a `@XmlRootElement` deve ser colocada em cima do objeto "Livro". Enquanto que algumas implementações das especificações de JAX-RS não fazem isso. Esta é uma solução.

A outra possibilidade é criar uma nova Classe `Livros` na qual colocariamos uma lista de livros, com a anotação `@XmlRootElement` :

```
package br.com.casadocodigo.loja.models;

import javax.xml.bind.annotation.XmlRootElement;
```

```
@XmlRootElement
public class Livros {

    private List<Livro> livros = new ArrayList<>();

    public List<Livro> getLivros() {
        return this.livros;
    }
}
```

Implementamos essa Classe só para que você tenha uma ideia de como funcionaria. Mas criar uma Classe para fazer essa transformação seria bem verboso, este caso é apenas se você não estiver utilizando o RESTEasy.

Então vamos reiniciar o servidor e testar no Postman com as mesmas configurações anteriores:

O retorno será o XML esperado. Usando como Root Element o collection da lista de livros.