10

Mãos a obra: Dados obrigatórios

Ao atualizar e criar um novo request no SoapUI, o XML SOAP mostra alguns comentários indicando que os elementos do `Item` e `TokenUsuario` são opcionais! Veja só:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws="http://ws.est
   <soapenv:Header>
      <ws:tokenUsuario>
         <!--Optional:-->
         <token>?</token>
         <!--Optional:-->
         <dataValidade>?</dataValidade>
      </ws:tokenUsuario>
   </soapenv:Header>
   <soapenv:Body>
      <ws:CadastrarItem>
         <!--Optional:-->
         <item>
            <!--Optional:-->
            <codigo>?</codigo>
            <!--Optional:-->
            <nome>?</nome>
            <!--Optional:-->
            <tipo>?</tipo>
            <quantidade>?</quantidade>
         </item>
      </ws:CadastrarItem>
   </soapenv:Body>
</soapenv:Envelope>
```

Entretanto, para nossa aplicação funcionar, o cliente *deve enviar todos os dados sobre o token e item*. Essa confusão com certeza vai atrapalhar os clientes do nosso serviço web! Não podemos sinalizar os dados como opcionais que na verdade são obrigatórios.

Para resolver essa confusão é preciso mexer nas classes `TokenUsuario` e `Item` . Por padrão, qualquer dado do nosso modelo é opcional a não ser quando configurado como obrigatório.

Abra a classe `TokenUsuario` e faça que o `token` e `data` se tornem obrigatórios. Use a anotação `@XmlElement` em cada atributo da classe. Além disso, para simplificar, use a anotação `@XmlAccessorType` para definir o acesso aos atributo invés de usar os *Getter/Setter*:

```
@XmlAccessorType(XmlAccessType.FIELD)
public class TokenUsuario {

    @XmlElement(required=true)
    private String token;
    @XmlElement(required=true)
    private Date dataValidade;

    //JAX-B precisa desse construtor
```

```
    TokenUsuario() {
    }

    public TokenUsuario(String token, Date dataValidade) {
        this.token = token;
        this.dataValidade = dataValidade;
    }

    //outros métodos omitidos
}
```

Essas anotações são da especificação JAX-B que é utilizado pelo JAX-WS para gerar e ler o XSD/XML.

Agora faça o mesmo na classe `Item` :

```
@XmlAccessorType(XmlAccessType.FIELD)
@XmlRootElement
public class Item {

    @XmlElement(required=true)
    private String codigo;

    @XmlElement(required=true)
    private String nome;

    @XmlElement(required=true)
    private String tipo;

    @XmlElement(required=true)
    private int quantidade;

    //construtores e métodos omitidos
}
```

Republique o serviço web no Eclipse e depois atualize o SoapUI e gere um novo request. O SoapUI ainda mostra um elemento *opcional*?