



80%

ATIVIDADES
8 de 12FÓRUM DO
CURSOVOLTAR
PARA
DASHBOARD

Vamos trabalhar com *Web Services*?

1) Crie uma nova classe chamada `LivrariaWS` no pacote `br.com.caelum.livraria.webservice`.

2) Ela também será um *session bean stateless*, devemos então anotá-la com `@Stateless` e um *web service*, anotando-a com `@WebService`.

```
@WebService
@Stateless
public class LivrariaWS {

}
```

COPIAR CÓDIGO

3) Agora, crie um método chamado `getLivrosPorTitulo`, que deve retornar uma lista de livros e receber um `String` como parâmetro. Por enquanto, faça seu método retornar apenas `null` e imprimir uma mensagem no console que foi acionado.

```
public List<Livro>
getLivrosPorTitulo(String titulo) {
    System.out.println("[LivrariaWS]
    Procurando pelo titulo " + titulo);
    return null;
}
```

COPIAR CÓDIGO

4) Faça o *Full Publish* da aplicação e procure no console o endereço do novo serviço.

```
11:18:16,009 INFO
[org.jboss.wsf.stack.cxf.metadata.MetadataBuilder
(MSC service thread 1-7) Add Service
id=LivrariaWS
address=http://localhost:8080/livraria
/LivrariaWS

implementor=br.com.caelum.livraria.webservice.Li

invoker=org.jboss.wsf.stack.cxf.JBossWSInvoker
serviceName=
{http://webservice.livraria.caelum.com.br
/}LivrariaWSService
```



119.0k xp





80%

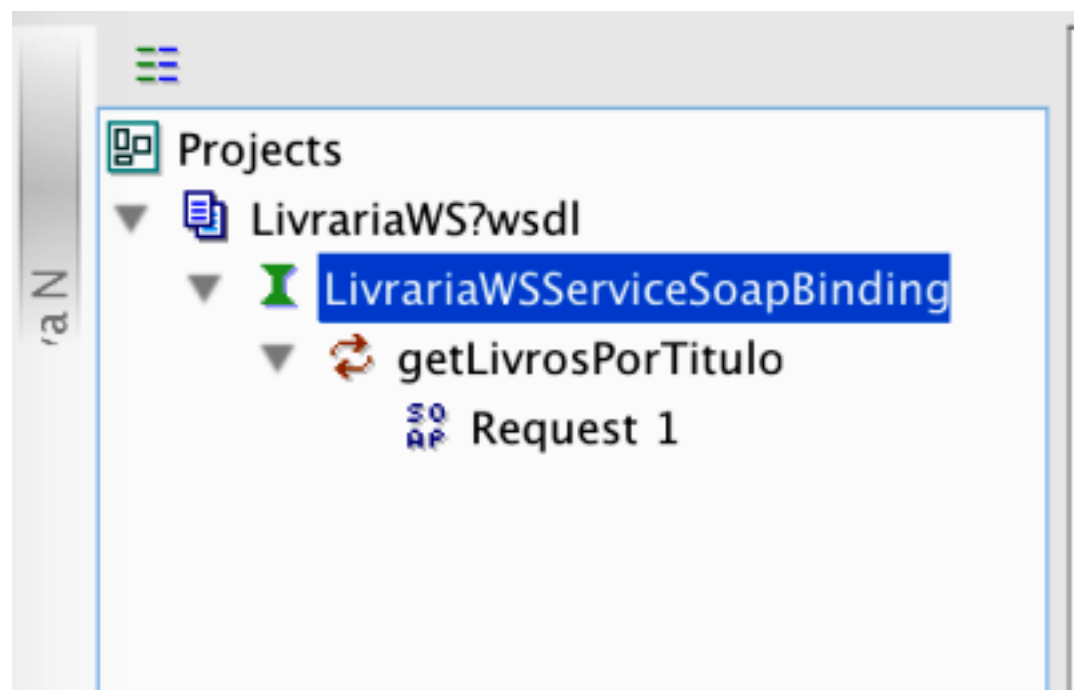
ATIVIDADES
8 de 12FÓRUM DO
CURSOVOLTAR
PARA
DASHBOARD

```
portName=
{http://webservice.livraria.caelum.com.br
/}LivrariaWSPort
wsdlLocation=null
...
```

COPIAR CÓDIGO

5) Use esse *address* para testar o serviço no navegador. Não se assuste com um erro `<soap:Fault>`. Ele aconteceu porque utilizamos o padrão **HTTP GET** ao invés do **HTTP POST** exigido pelo serviço. Você pode ver a interface do serviço incluindo no final da URI *address*: `?wsdl`.

6) Agora, abra a ferramenta **SoapUI** e crie um novo **SOAP Project** (*File > New SOAP Project*). Coloque a URL do serviço, incluindo o WSDL, no campo **Initial WSDL** e clique no botão **OK** (ex.: <http://localhost:8080/livraria/LivrariaWS?wsdl> (<http://localhost:8080/livraria/LivrariaWS?wsdl>)). Ficaremos com algo parecido com a imagem abaixo.



Atenção: O aplicativo pode ser obtido no endereço <http://sourceforge.net/projects/soapui/files/> (<http://sourceforge.net/projects/soapui/files/>) e sua instalação e execução são mostrados no vídeo deste capítulo.

7) Dê um duplo clique no nó **Request 1** do método `getLivrosPorTitulo` e substitua a `?` no elemento `<arg0>` pelo nome do livro, por exemplo **Arquitetura**. Depois envie a requisição clicando no botão verde de *play* na janela do *request*. Você receberá, na janela ao lado, o retorno da requisição que será um **XML SOAP** vazio e no console do eclipse a mensagem que o método do web service foi acionado.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org
/soap/envelope/"
```



119.0k xp





80%

ATIVIDADES
8 de 12FÓRUM DO
CURSOVOLTAR
PARA
DASHBOARD

```
xmlns:web="http://webservice.livraria.caelum.com"
<soapenv:Header/>
<soapenv:Body>
  <web:getLivrosPorTitulo>
    <!--Optional:-->
    <arg0>Arquitetura</arg0>
  </web:getLivrosPorTitulo>
</soapenv:Body>
</soapenv:Envelope>
```

COPIAR CÓDIGO

Retorno:

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org
/soap/envelope/">
  <soap:Body>
    <ns2:getLivrosPorTituloResponse

xmlns:ns2="http://webservice.livraria.caelum.com"
  </soap:Body>
</soap:Envelope>
```

COPIAR CÓDIGO



Opinião do instrutor

Pronto, nosso *Web Service* já está funcionando. Vamos melhorar nosso método?



119.0k xp

