

05

## Mão a obra: Publicando o serviço no Wildfly

Com o projeto `estoque-web` criado e classes geradas podemos implementar e publicar o serviços. **Se você já implementou o serviço pode pular este exercício.**

1) No pacote `br.com.caelum.estoque.ws` que foi gerado pelo `wsdl2java`, crie uma nova classe com o nome `EstoqueWSImpl`. A classe `EstoqueWSImpl` deve implementar a interface `EstoqueWS`.

2) Na classe `EstoqueWSImpl` use a anotação `@WebService` para definir a interface, nome do serviço e port:

```
@WebService(endpointInterface="br.com.caelum.estoque.ws.EstoqueWS",
  serviceName="EstoqueWS",
  portName="EstoqueWSPort")
public class EstoqueWSImpl implements EstoqueWS {
```

3) Implemente os métodos exigidos pela interface:

```
@Override
public ListaItens todosOsItens(Filtros filtros) {
    System.out.println("Chamando todos os Itens");
    return new ListaItens();
}

@Override
public CadastrarItemResponse cadastrarItem(CadastrarItem parameters, TokenUsuario tokenUsuario)
    System.out.println("Chamando cadastrarItem");
    return new CadastrarItemResponse();
}
```

Observação: Em ambos os métodos estamos retornando apenas um objeto vazio para deixar o código compilar.

4) Reinicie o JBoss Wildfly no Eclipse e fique atento ao console.

5) Se o servidor subiu sem problemas acesse: <http://localhost:8080/estoque-web/EstoqueWS?wsdl>  
[\(http://localhost:8080/estoque-web/EstoqueWS?wsdl\)](http://localhost:8080/estoque-web/EstoqueWS?wsdl)

Você deve ver o WSDL publicado.