

## Mãos a obra: Usando o FaultInfo da exceção

Cada `Fault` deve ter no mínimo um `<faultcode>` e `<faultstring>` mas vimos no vídeo que existe a possibilidade de definir mais detalhes:

```
<S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
  <faultcode>S:Server</faultcode>
  <faultstring>Autorizacao falhou</faultstring>
  <detail>
    <ns2:AuthorizationFault xmlns:ns2="http://ws.estoque.caelum.com.br/">Token invalido</ns2:AuthorizationFault>
  </detail>
</S:Fault>
```

Para alterar o conteúdo do elemento `<detail>` é preciso mexer na exceção. Abre a classe `AutorizacaoException` e crie um novo método `getFaultInfo()`:

```
@WebFault(name="AutorizacaoFault", messageName="AutorizacaoFault")
public class AutorizacaoException extends Exception {

    private static final long serialVersionUID = 1L;

    public AutorizacaoException(String msg) {
        super(msg);
    }

    //novo
    public String getFaultInfo() {
        return "Token invalido";
    }
}
```

Publique o serviço e faça o teste através do SoapUI! Envie um requisição SOAP com um token inválido. Você deve receber um `Fault` parecido com o acima.