**SWS-105-Exceptions:**

**Pratica:**In caso di eccezione.

* Se non viene configurato nulla, di default viene usato il **SimpleSoapExceptionResolver**.
* Se l’eccezione e’ annotata con @SoapFault(code, string)> **SoapFaultAnnotationExceptionResolver**
* Se si vuole creare un custom SoapExceptionResolver bisogna:
  + Estendere classe SoapFaultMappingExceptionResolver ( o sue subclassi)
  + Override metodo customizeFault

**Scelte:**

1. SimpleSoapExceptionResolver usa il messaggio dell eccezione.
2. Se I messaggi sono standard 1 a 1 per eccezione.  
   Puo’ essere comodo definirli con le annotations nelle eccezioni specifiche.
3. Usare implementazione custom per aggiungere dettagli. Meglio estendere una delle due precedent.

**TC:  
TC1: SimpleSoapExceptionResolver (Endpoint modifica)**Verificare che l’eccezione lanciata non sia annotata.   
  
**TC2: SoapFaultAnnotationExceptionResolver: (Endpoint modifica)**Lanciare eccezione annotata.

**TC3: CustomSoapFaultAnnotationExceptionResolver: (SWSContext modifica)**Decommentare @Bean in SWSContext

**TEORIA:**

Necessario meccanismo per mappare eccezioni in <soap:Fault>  
  
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<SOAP-ENV:Fault>

<faultcode xmlns:ns0="http://com.example">ns0:4</faultcode>

<faultstring xml:lang="en">Ops !!!</faultstring>

</SOAP-ENV:Fault>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

**Default:**

|  |  |
| --- | --- |
|  | MessageDispatcher raise two Resolvers by default:   1. SoapFaultAnnotationExceptionResolver order =0 2. SimpleSoapExceptionResolver and order= 2147483647 (Max Integer) ultimo   **Mappatura eccezioni Criteri:**   1. SimpleSoapExceptionResolver  FaultCode: Server FaultString: ex.getMessage() 2. SoapFaultAnnotationExceptionResolver Risolve annotation su <Exception> (contenente I due parametri) FaultCode: FaultString: 3. SoapFaultMappingExceptionResolver Mapping sulla base della propeita mapping  (criterio definito in SoapFaultDefinitionEditor)   <property name="exceptionMappings"> <value>  nome.qualificato.exception=Code, String, Locale  ESEMPIO:  <property name="exceptionMappings"> <value> corso.spring.ee.demo.ws.basic.service.exceptions.ServiceException= CLIENT,Invalid request </value>  **TUTORIAL STEPS**:   1. Aggiungere l’ExceptionResolver nel contesto di SWS 2. Definire un order inferiore a 0 (default di SoapFaultAnnotationExceptionResolver)   **Soluzioni per ogni tipo di ExceptionResolver:**  1) Abilitare SoapFaultAnnotationExceptionResolver  Basta mappare la classe di eccezione con l’annotation @SoapFault  2)SimpleSoapExceptionResolver (DEFAULT se l eccezione non e’ annotata) Server, ex.getMessage()  3) SoapFaultMappingExceptionResolver Impostare order<0 per sovrascrivere SoapFaultAnnotationExceptionResolver  <bean id="exceptionResolver"  class="myCustomSoapExceptionResolver"/> |

Estesioni Progetto base:

1. SERVER:
   1. Aggiungere classe implementazione di EndpointExceptionResolver nel contesto.
   2. In metodo di @Endpoint aggiungere throws <SpecificaException>
2. In CLIENT:
   1. Aggiungere handler fault nel WsTemplate

**TEST:**

**SERVER**:

1. Nessun EndpointExceptionResolver ->
   1. Default SoapFaultAnnotationExceptionResolver se annotations su Ex
   2. Default SimpleSoapExceptionResolver (default msg)
2. Test con SoapFaultMappingExceptionResolver: spring-ws-servlet.xml

**CLIENT**:

1. Default FaultHandler in WsTemplate:  
   Commentare collegamento con classe faultHandler