

Hi !

I have a fault message generated by SoapUI :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:app="XXXX/application">
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value>soap:Sender</soap:Value>
      </soap:Code>
      <soap:Reason>
        <!--1 or more repetitions:-->
        <soap:Text xml:lang="fr">blabla</soap:Text>
        <soap:Text xml:lang="fr">blibli</soap:Text>
      </soap:Reason>
      <!--Optional:-->
      <soap:Node>node</soap:Node>
      <!--Optional:-->
      <soap:Role>role</soap:Role>
      <!--Optional:-->
      <soap:Detail>
        <!--You may enter ANY elements at this point-->
        <app:DownloadedApplicationException>
<code>code</code>
<message>message</message>
<severity>severity</severity>
        </app:DownloadedApplicationException>
      </soap:Detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

In my job my tESBConsumer return the fault message but my tXMLMap throw an exception :

```
application}ApplicationWebService#{ }notifyDownloadApplication has thrown exception, unwinding now
org.apache.cxf.interceptor.Fault: Could not generate the XML stream caused by: com.ctc.wstx.exc.WstxEOFException: Unexpected EOF in prolog
at : .
at org.apache.cxf.databinding.source.XMLStreamDataWriter.write(XMLStreamDataWriter.java:94)
at org.apache.cxf.databinding.source.XMLStreamDataWriter.write(XMLStreamDataWriter.java:54)
at org.apache.cxf.databinding.source.XMLStreamDataWriter.write(XMLStreamDataWriter.java:48)
at org.apache.cxf.interceptor.AbstractOutDatabindingInterceptor.writeParts(AbstractOutDatabindingInterceptor.java:136)
at org.apache.cxf.wsdl.interceptors.BareOutInterceptor.handleMessage(BareOutInterceptor.java:68)
at org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:308)
at org.apache.cxf.interceptor.OutgoingChainInterceptor.handleMessage(OutgoingChainInterceptor.java:83)
at org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:308)
at org.apache.cxf.transport.ChainInitiationObserver.onMessage(ChainInitiationObserver.java:121)
at org.apache.cxf.transport.http.AbstractHTTPDestination.invoke(AbstractHTTPDestination.java:251)
at org.apache.cxf.transport.http_jetty.JettyHTTPDestination.doService(JettyHTTPDestination.java:234)
at org.apache.cxf.transport.http_jetty.JettyHTTPHandler.handle(JettyHTTPHandler.java:70)
at org.eclipse.jetty.server.handler.ContextHandler.doHandle(ContextHandler.java:1088)
at org.eclipse.jetty.server.handler.ContextHandler.doScope(ContextHandler.java:1024)
at org.eclipse.jetty.server.handler.ScopedHandler.handle(ScopedHandler.java:135)
at org.eclipse.jetty.server.handler.ContextHandlerCollection.handle(ContextHandlerCollection.java:255)
at org.eclipse.jetty.server.handler.HandlerWrapper.handle(HandlerWrapper.java:116)
at org.eclipse.jetty.server.Server.handle(Server.java:366)
at org.eclipse.jetty.server.AbstractHttpConnection.handleRequest(AbstractHttpConnection.java:494)
```

```

at org.eclipse.jetty.server.AbstractHttpConnection.content(AbstractHttpConnection.java:982)
at org.eclipse.jetty.server.AbstractHttpConnection$RequestHandler.content(AbstractHttpConnection.java:1043)
at org.eclipse.jetty.http.HttpParser.parseNext(HttpParser.java:865)
at org.eclipse.jetty.http.HttpParser.parseAvailable(HttpParser.java:240)
at org.eclipse.jetty.server.AsyncHttpConnection.handle(AsyncHttpConnection.java:82)
at org.eclipse.jetty.io.nio.SelectChannelEndPoint.handle(SelectChannelEndPoint.java:667)
at org.eclipse.jetty.io.nio.SelectChannelEndPoint$1.run(SelectChannelEndPoint.java:52)
at org.eclipse.jetty.util.thread.QueuedThreadPool.runJob(QueuedThreadPool.java:608)
at org.eclipse.jetty.util.thread.QueuedThreadPool$3.run(QueuedThreadPool.java:543)
at java.lang.Thread.run(Thread.java:745)
Caused by: com.ctc.wstx.exc.WstxEOFException: Unexpected EOF in prolog
at :
at com.ctc.wstx.sr.StreamScanner.throwUnexpectedEOF(StreamScanner.java:685)
at com.ctc.wstx.sr.BasicStreamReader.handleEOF(BasicStreamReader.java:2141)
at com.ctc.wstx.sr.BasicStreamReader.nextFromProlog(BasicStreamReader.java:2047)
at com.ctc.wstx.sr.BasicStreamReader.next(BasicStreamReader.java:1131)
at org.apache.cxf.staxutils.StaxUtils.copy(StaxUtils.java:771)
at org.apache.cxf.staxutils.StaxUtils.copy(StaxUtils.java:701)
at org.apache.cxf.staxutils.StaxUtils.copy(StaxUtils.java:625)
at org.apache.cxf.databinding.source.XMLStreamDataWriter.write(XMLStreamDataWriter.java:91)
... 28 more

```

This is the fault of tESBConsumer in the input of my tXMLMap3 (mapping on the screenshot) :

```

: XXXX.test_0_1.test - row3 - {http://www.w3.org/2003/05/soap-envelope}Sender|blabla|<null>|node|role|<?xml version="1.0" encoding="UTF-8"?>
<app:DownloadedApplicationException xmlns:app="XXXX/application">
  <code>code</code>
  <message>message</message>
  <severity>severity</severity>
  </app:DownloadedApplicationException>|

```

There is a <null> value in this message corresponding at faultActor ...
 faultActor is not valid element in my fault message according to soapui validation.

According to javadoc :

```

/**
 * Gets the fault actor for this <code>SOAPFault</code> object.
 * <P>
 * If this <code>SOAPFault</code> supports SOAP 1.2 then this call is
 * equivalent to {@link #getFaultRole()}
 *
 * @return a <code>String</code> giving the actor in the message path
 *         that caused this <code>SOAPFault</code> object
 * @see #setFaultActor
 */
public String getFaultActor();

```

My role is well returned but actor is always at null ...
 I don't understand !

If I see java code there is a method getFaultInfo:

```
private static java.util.Map<String, Object> collectFaultInfo(
    Exception error) {
    String faultString = null;
    String faultActor = null;
    String faultCode = null;
    String faultRole = null;
    String faultNode = null;
    org.dom4j.Document faultDetail = null;
    org.w3c.dom.Element faultDomDetail = null;
    if (error instanceof org.apache.cxf.binding.soap.SoapFault) {
        org.apache.cxf.binding.soap.SoapFault cxfSoapFault = (org.apache.cxf.binding.soap.SoapFault) error;
        faultString = cxfSoapFault.getReason();
        faultCode = cxfSoapFault.getFaultCode().toString();
        faultRole = cxfSoapFault.getRole();
        faultNode = cxfSoapFault.getNode();
        faultDomDetail = cxfSoapFault.getDetail();
    } else if (error instanceof GenericFaultInfoException) {
        GenericFaultInfoException genericFault = (GenericFaultInfoException) error;
        faultString = genericFault.getMessage();
        faultDetail = genericFault.getFaultInfo();
    } else if (error instanceof javax.xml.ws.soap.SOAPFaultException) {
        javax.xml.ws.soap.SOAPFault wsSoapFault = ((javax.xml.ws.soap.SOAPFaultException) error)
            .getFault();
        faultString = wsSoapFault.getFaultString();
        faultActor = wsSoapFault.getFaultActor();
        faultCode = wsSoapFault.getFaultCodeAsQName().toString();
        if (javax.xml.ws.soap.SOAPConstants.URI_NS_SOAP_1_2_ENVELOPE
            .equals(wsSoapFault.getNamespaceURI())) {
            // SOAP 1.2
            try {
                faultRole = wsSoapFault.getFaultRole();
                faultNode = wsSoapFault.getFaultNode();
            } catch (java.lang.UnsupportedOperationException uoe) {
            }
        }
        faultDomDetail = wsSoapFault.getDetail();
    } else {
        return null;
    }
    java.util.Map<String, Object> faultInfo = new java.util.HashMap<String, Object>();
    faultInfo.put("faultString", faultString);
    faultInfo.put("faultActor", faultActor);
    faultInfo.put("faultCode", faultCode);
    faultInfo.put("faultRole", faultRole);
    faultInfo.put("faultNode", faultNode);
    if (null != faultDomDetail) {
        org.w3c.dom.NodeList nl = faultDomDetail.getElementsByTagName("*");
        if (nl.getLength() > 0) {
            try {
                javax.xml.transform.Source faultSource = new javax.xml.transform.dom.DOMSource(
                    nl.item(0));
                faultDetail = GenericConsumer.sourceToDocument(faultSource);
            } catch (Exception e) {
            }
        }
    }
}
```

```
        e.printStackTrace();
    }
}
}
faultInfo.put("faultDetail", faultDetail);
return faultInfo;
}
```

So I think this is a cxf soap fault I have but my tesbconsumer want a faultActor element that is not treated by cxf soapfault case ... I try to delete faultActor but then there is an error on my tesbconsumer ("faultActor cannot be resolved or is not a field") so it's needed absolutely to have this faultActor ...???