**DA EVITARE SaajSoapMessageFactory**

Legata a version JDK per via di librerie xerces, xalan, saaj che possono cambiare da una versione all’altra della JDK.

**4.1.3.2. AxiomSoapMessageFactory**

The AxiomSoapMessageFactory uses the AXis 2 Object Model to create SoapMessage implementations. AXIOM is based on StAX, the Streaming API for XML. StAX provides a pull-based mechanism for reading XML messages, which can be more efficient for larger messages.

To increase reading performance on the AxiomSoapMessageFactory, you can set the payloadCaching property to false (default is true). This will read the contents of the SOAP body directly from the socket stream. When this setting is enabled, the payload can only be read once. This means that you have to make sure that any pre-processing (logging etc.) of the message does not consume it.

You use the AxiomSoapMessageFactory as follows:

<bean id="messageFactory" class="org.springframework.ws.soap.axiom.AxiomSoapMessageFactory">

<property name="payloadCaching" value="true"/>

</bean>

In addition to payload caching, AXIOM also supports full streaming messages, as defined in the StreamingWebServiceMessage. This means that the payload can be directly set on the response message, rather than being written to a DOM tree or buffer.

Full streaming for AXIOM is used when a handler method returns a JAXB2-supported object. It will automatically set this marshalled object into the response message, and write it out to the outgoing socket stream when the response is going out.

For more information about full streaming, refer to the class-level Javadoc for StreamingWebServiceMessage and StreamingPayload.

**4.1.3.3. SOAP 1.1 or 1.2**

Both the SaajSoapMessageFactory and the AxiomSoapMessageFactory have a soapVersion property, where you can inject a SoapVersion constant. By default, the version is 1.1, but you can set it to 1.2 like so:

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:util="http://www.springframework.org/schema/util"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.0.xsd

http://www.springframework.org/schema/util

http://www.springframework.org/schema/util/spring-util-2.0.xsd">

<bean id="messageFactory" class="org.springframework.ws.soap.saaj.SaajSoapMessageFactory">

<property name="soapVersion">

<util:constant static-field="org.springframework.ws.soap.SoapVersion.SOAP\_12"/>

</property>

</bean>

</beans>

In the example above, we define a SaajSoapMessageFactory that only accepts SOAP 1.2 messages.

**Caution**

Even though both versions of SOAP are quite similar in format, the 1.2 version is not backwards compatible with 1.1 because it uses a different XML namespace. Other major differences between SOAP 1.1 and 1.2 include the different structure of a Fault, and the fact that SOAPAction HTTP headers are effectively deprecated, thought they still work.

One important thing to note with SOAP version numbers, or WS-\* specification version numbers in general, is that the latest version of a specification is generally not the most popular version. For SOAP, this means that currently, the best version to use is 1.1. Version 1.2 might become more popular in the future, but currently 1.1 is the safest bet.