

Mike's Software Development Blog

A computer blog with a focus on software development.

Sunday, May 8, 2011

WSDL SOAP bindings confusion - RPC vs document

What is the difference between RPC and document styles in SOAP web services?

I was asked this question at a job interview and was embarrassed to discover that I didn't know the answer (considering that I listed "SOAP web services" on my resume). I've heard the terms before, but couldn't remember what they meant. The main difference lies in what the body of the SOAP message looks like.

RPC vs document styles

The body of an **RPC** (remote procedure call) style SOAP message is constructed in a specific way, which is defined in the SOAP standard. It is built around the assumption that you want to call the web service just like you would call a normal function or method that is part of your application code. The message body contains an XML element for each "parameter" of the method. These parameter elements are wrapped in an XML element which contains the name of the method that is being called. The response returns a single value (encoded in XML), just like a programmatic method. The WSDL code for a RPC-style web service is less complex than that of a document-style web service, but this isn't a big deal since WSDLs aren't meant to be handled by humans.

A RPC-style request:

```
<soap:envelope>
  <soap:body>
    <multiply>      <!-- web method name -->
      <a>2.0</a>    <!-- first parameter -->
      <b>7</b>      <!-- second parameter -->
    </multiply>
  </soap:body>
</soap:envelope>
```

A **document** style web service, on the other hand, contains no restrictions for how the SOAP body must be constructed. It allows you to include whatever XML data you want and also to include a schema for this XML. This means that the client's and server's application code must do the marshalling and unmarshalling work. This contrasts with RPC in which the marshalling/unmarshalling process is part of the standard, so presumably should be handled by whatever SOAP library you are using. The WSDL code for a document-style web service is much more complex than that of a RPC-style web service, but this isn't a big deal since WSDLs aren't meant to be handled by humans.

A document-style request:

```
<soap:envelope>
  <soap:body>
    <!-- arbitrary XML -->
    <movies xmlns="http://www.myfavoritemovies.com">
```

Blog Archive

- ▶ 2011
- ▶ 2010
- ▶ 2009
- ▼ 2008
 - ▶ January
 - ▶ February
 - ▶ March
 - ▶ April
 - ▶ May
 - ▶ June
 - ▶ July
 - ▶ August
 - ▶ September
 - ▶ October
 - ▶ November
 - ▶ December
- ▶ 2007
- ▶ 2006
- ▶ 2005

About



Follow

Follow



Follow

```

<movie>
  <title>2001: A Space Odyssey</title>
  <released>1968</released>
</movie>
<movie>
  <title>Donnie Darko</title>
  <released>2001</released>
</movie>
</movies>
</soap:body>
</soap:envelope>

```

The main downside of the RPC style is that it is *tightly coupled* to the application code (that is, if you decide you want to call these web methods like normal methods—this is not a requirement, but this is what the RPC style was designed for). This means that if you want to change the order of the parameters or change the types of those parameters, this change will affect the definition of the web service itself (just as it would affect the definition of a normal function or method).

Document style services do not have this issue because they are *loosely coupled* with the application code—the application must handle the marshalling and unmarshalling of the XML data separately. For example, with a document style service, it doesn't matter if the programmer decides to use a "float" instead of an "int" to represent a particular parameter because it's all converted to XML text in the end.

The main downside of the document style is that there is no standard way of determining which method of the web service the request is for. It's easy to get around this limitation, but, however it's done, it must be done manually by the application code. (Note: The "document/literal wrapped" style removes this limitation; read on for more details.)

Another point to note about the document style is that there are no rules for how the SOAP body must be formatted. This can either be seen as a downside or a strength, depending on your perspective. It's a strength if you are looking for the freedom to handle the message the way you want, but a downside if you don't want to have to do the extra marshalling/unmarshalling work that it requires.

Encoded vs literal encodings

In addition to the RPC and document styles, there are two types of encodings: "encoded" and "literal".

Literal means that the SOAP body follows an XML schema, which is included in the web service's WSDL document. As long as the client has access to the WSDL, it knows exactly how each message is formatted.

Encoded, on the other hand, means that the SOAP body does not follow a schema, but still follows a specific format which the client is expected to already know. It is not endorsed by the WS-I standard because there can be slight differences in the way in which various programming languages and web service frameworks interpret these formatting rules, leading to incompatibilities.

This makes for 4 different style/encoding combinations:

RPC/encoded - RPC-style message that formats its body according to the rules defined in the SOAP standard (which are not always exact and can lead to incompatibilities).

RPC/literal - RPC-style message that formats its body according to a schema that reflects the rules defined in the SOAP standard. This schema is included in the WSDL.

document/encoded - Document-style message that does not include a schema

(nobody uses this in practice).

document/literal - Document-style message that formats its body according to a schema. This schema is included in the WSDL.

There's also a 5th type. It isn't an official standard but it is used a lot in practice. It came into being to compensate for document/literal's main shortcoming of not having a standard way of specifying the web method name:

document/literal wrapped - The same as document/literal, but wraps the contents of the body in an element with the same name as the web service method (just like RPC-style messages). This is what web services implemented in Java use by default.

Is my understanding of all this accurate? Which approach do you think is the best? Let me know in the comments.

References:

- <http://soa.sys-con.com/node/39901>
- <http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl>
- <http://www.coderanch.com/t/148478/java-Web-Services-SCDJWS/certification/document-Vs-rpc-message-styles>
- <http://www.coderanch.com/t/148480/java-Web-Services-SCDJWS/certification/encoding-style-literal-vs-encoded>
- <http://www.sdn.sap.com/irj/scn/index?nd=/library/uuid/c018da90-0201-0010-ed85-d714ff7b7019>
- <http://johnrangan.wordpress.com/2010/01/04/soap-messages-rpc-vs-document-vs-literal-vs-encoded-vs-wrapped-vs-unwrapped/>

Posted by **Michael Angstadt** at 10:03 AM

 +2 Recommend this on Google

Labels: [soap](#), [web-services](#), [wsdl](#)

41 comments:

Anonymous said...

Wow Mike. Great detail there.

I was asked this question in an interview about 4 years ago. Ever since, once in a while, I search the web, but never got a satisfactory answer...till now. Great work.

-Raghu M

July 18, 2011 at 9:53 PM



misha said...

Thank you, that is a great clear answer

July 20, 2011 at 6:29 AM

Anonymous said...

Mike, You are awesome..

I liked the way you presented. it is crystal clear.. I was going thru lots of tutorials and