

OperationInvocation



Archive Content

Please note this page is present for reference only. QMan has been removed and is no longer a released component.

Operation invocation on a WS-Resource

- [Description](#)
- [Return Type](#)
- [Request](#)
- [Response](#)
- [Faults](#)
- [Quick links](#)

Description

This interface allows a requestor to invoke an operation on a WS-Resource.

Let's say that there's a substantial difference with the other interfaces explained in this section. The name "Operation invocation" doesn't mean each WS-Resource

has an interface with this name. Remember, the WS-Resource interface definition is built at runtime, and therefore we don't know what will be the set of capabilities / operations / properties the resource will expose for management.

In this example we assume the resource has an operation called "echo" explained in detail below.

As part of that, we will provide information about

- How to invoke that operation;
- How to correctly specify / encode the input parameters;
- What is the expected return type on a basic scenario (without exceptions);
- What are possible faults that are returned as result of an exception / error scenario;

Generally speaking, an operation is mainly defined by its signature. That includes :

- Operation name : the name of the operation.
- Return Type : the return type of the operation.
- Parameters : The input parameters that will be used by operation in order to execute its task.
- Fault / Exceptions / Errors : thrown by the operation when an error occurs.

For QMan resources, consider these important points about operation signatures :

- Operations have always one and only one return type : Basically it is a value object containing an (optional) output parameters map. See section below for details.
- Operations have always the same set of faults in their signature. See section below for explanation of each fault.

Return Type

Return type is a simple "Value Object" that encapsulates the (optional) the output parameters map. Note that for example void methods won't contain that map.

The following is the schema of the mentioned return type. Obviously you can find it in each WS-Resource WSDL, too.

```

<!-- Invocation Result -->
    <xsd:complexType name="result">
        <xsd:sequence>
            <xsd:element name="outputParameters" type="qman:map"/>
        </xsd:sequence>
    </xsd:complexType>

    <!-- Output parameter map -->
    <xsd:complexType name="map">
        <xsd:sequence>
            <xsd:element name="entry" minOccurs="0" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:sequence>
                        <!-- Entry key is always a string. -->
                        <xsd:element name="key" type="xsd:string"/>
                        <!-- While entry value could be any arbitrary
object. -->
                        <xsd:element name="value" type="xsd:anyType"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>

```

Request

For this example we suppose the WS-Resource has the following operation :

```

01.      <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
02.          <soap:Header>
03.              <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
04.                  http://localhost:8080/qman/services/QManWsResource
05.              </wsa:To>
06.              <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
07.                  http://amqp.apache.org/qp/management/qman/echoWithSimpleTypes
08.              </wsa:Action>
09.              <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
10.                  uuid:0cdb5112-09e0-ac39-06ba-393843f06e42
11.              </wsa:MessageID>
12.              <wsa:From xmlns:wsa="http://www.w3.org/2005/08/addressing">
13.                  <wsa:Address>
14.                      http://www.w3.org/2005/08/addressing/role/anonymous
15.                  </wsa:Address>
16.              </wsa:From>
17.              <qman:ResourceId
18.                  xmlns:wsa="http://www.w3.org/2005/08/addressing"
19.                  wsa:IsReferenceParameter="true"
20.                  xmlns:qman="http://amqp.apache.org/qp/management/qman">
21.                      781f4ad7-4c96-4caa-b69d-291461cdb1fc
22.              </qman:ResourceId>
23.          </soap:Header>
24.          <soap:Body xmlns:qman="http://amqp.apache.org/qp/management/qman">
25.              <qman:echoWithSimpleTypesRequest>
26.                  <qman:p1>1373</qman:p1>
27.                  <qman:p2>true</qman:p2>
28.                  <qman:p3>12763.44</qman:p3>
29.                  <qman:p4>2727.233</qman:p4>
30.                  <qman:p5>28292</qman:p5>
31.                  <qman:p6>227</qman:p6>
32.                  <qman:p7>expectedStringResult</qman:p7>
33.                  <qman:p8>http://qp/management/qman/</qman:p8>
34.                  <qman:p9>1235061886761</qman:p9>
35.                  <qman:p9 xsi:type="xsd:long">
36.                      1235061886761
37.                  </qman:p9>
38.              </qman:echoWithSimpleTypesRequest>
39.          </soap:Body>
40.      </soap:Envelope>

```

Line (s)	Description
01	The SOAP <Envelope> is the root element in every SOAP message, and contains two child elements, <Header> and <Body>.
02	The SOAP Header will contain all metadata used for identifying the conversation participants (requestor and provider).
03 - 05	Convey the target endpoint also known (in the request phase) as service provider.
06 - 08	The action usually corresponds to the name of the operation. Genarally speaking it is composed by the QMan namespace followed by the name of the operation.
09 - 11	Convey a unique identifier associated with the current message. This will be used for request / response messages correlation.
12 - 15	Provide the address of the source endpoint also known (in the request phase) as service requestor.
17 - 19	This indicates the target resource of this request. Specifically the line 18 contains the resource identifier.
20 - 31	The operation invocation request. Note that parameters have no type specified (except for the line 30). This is because the corresponding element is explicitly declared on resource WSDL. Line 30 (qman:P9 parameter) has an inlined type declaration using the xsi:type attribute because in the WSDL its declared type is xsd:anyType so we can't know in advance what it really is.

Response

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
01.       <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
02.         http://www.w3.org/2005/08/addressing/role/anonymous
03.       </wsa:To>
04.       <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
05.         http://amqp.apache.org/qpidd/management/qman/echoWithSimpleTypesResponse
06.       </wsa:Action>
07.       <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
08.         uuid:980617c8-e3a0-ebf1-8f5a-2b43d3d6d416
09.       </wsa:MessageID>
10.       <wsa:RelatesTo RelationshipType="wsa:Reply" xmlns:wsa="http://www.w3.org/2005/08
/addressing">
11.         uuid:0cdb5112-09e0-ac39-06ba-393843f06e42
12.       </wsa:RelatesTo>
13.       <wsa:From xmlns:wsa="http://www.w3.org/2005/08/addressing">
14.         <wsa:Address>
15.           http://localhost:8080/qman/services/QManWsResource
16.         </wsa:Address>
17.         <wsa:ReferenceParameters>
           <qman:ResourceId
             xmlns:wsa="http://www.w3.org/2005/08/addressing"
             wsa:IsReferenceParameter="true"
             xmlns:qman="http://amqp.apache.org/qpidd/management/qman">
18.               781f4ad7-4c96-4caa-b69d-291461cdb1fc
19.             </qman:ResourceId>
20.           </wsa:ReferenceParameters>
         </wsa:From>
       </soap:Header>
    <soap:Body>
21.       <qman:echoWithSimpleTypesResponse xmlns:qman="http://amqp.apache.org/qpidd/management
/qman">
22.         <qman:echoWithSimpleTypesResponse>
23.           <outputParameters xmlns:xsi="http://www.w3.org/2001/XMLSchema-
24.             instance">
25.               <qman:p1>1373</qman:p1>
26.               <qman:p2>true</qman:p2>
27.               <qman:p3>12763.44</qman:p3>
28.               <qman:p4>2727.233</qman:p4>
29.               <qman:p5>28292</qman:p5>
30.               <qman:p6>227</qman:p6>
31.               <qman:p7>expectedStringResult</qman:p7>
32.               <qman:p8>http://qpidd.apache.org/</qman:p8>
33.               <qman:p9>1235061886761</qman:p9>
34.               <qman:p9 xsi:type="xsd:long">
35.                 1235061886761
36.               </qman:p9>
37.             </outputParameters>
           <qman:echoWithSimpleTypesResponse>
           </qman:echoWithSimpleTypesResponse>
         </soapBody>
    </soap:Envelope>

```

Line (s)	Description
01 - 03	Convey the recipient of the response message. Note that this time we are talking about the service requestor; The address matches the <wsa:From> previously found in the corresponding request.
04 - 06	Indicate this is a echoWithSimpleType response. This is done as usual using a wsa:Action that is part of WS-Addressing specification.
07 - 09	Convey a unique identifier associated with the current response message.
10 - 12	This element provides the identifier of the correlated (request) message.
13	The <wsa:From> element (part of WS-Addressing specs too) identifies the source endpoint, the originator of this response message.
14 - 16	This is the address of the source service endpoint. As said for lines 01-03 this time this is referred to service provider (the message originator).
17 - 20	As part of wsa:From element, this contains (specifically on line 18) additional information needed for identifying the originator of this message.

21 - 31	This is the GetMultipleResourceProperties response element which contains the requested property values as nested child.
22 - 37	This is the response body. Specifically the operation in the example is a simple "echo" operation, and therefore all input parameters are sent back as output section.

Faults

- ResourceUnknownFault : There's no resource on QMan associated with the given reference information (soap address and identifier).
- ResourceUnavailableFault : The requested resource is unavailable. This fault should indicate a transient condition. That means a requester might resend the message.
- InvalidResourcePropertyQNameFault : One or more of the names (QNames) in the request message doesn't correspond to a property element of the target WS-Resource.
- OperationInvocationFault : The operation invocation failed.

Quick links

[Web Services Resource](#)