MIIC SOAP/Web Services Specifications for Data Exchange

Contents

MIIC SOAP/Web Services Specifications for Data Exchange 1
Introduction to SOAP/Web Services in MIIC 1
Getting the WSDL 2
Method 1: Retrieve the WSDL from the web service server 2
Method 2: Extract the version from this document 3
Getting credentials 3
HL7 file testing and verification 3
Appendix A: Test WSDL and XSD File Contents 3

Introduction to SOAP/Web Services in MIIC

A Web service is a standards-based method of allowing one computer to access functions in another computer through the Internet. The functions are accessed through the same ports used by Internet browsers, allowing MIIC and other applications to share messages using existing Web application protocols.

Security is implemented in a couple different ways. First, MIIC uses the HTTPS protocol to provide bidirectional encryption of communications between a client and server. Second, we use a username, password, and facility ID to ensure authentication and authorization.

MIIC supports the following functions through Web services:

1. connectivityTest, used to check for connectivity to the service and echo back message sent.
2. submitSingleMessage, used to send either VXU vaccine updates or QBP query messages.

Web Service functions are called using Simple Object Access Protocol (SOAP) requests, which are formatted as XML (eXtensible Markup Language) messages.

Each SOAP request is made up of the following elements:

1. The envelope, which identifies the message as a SOAP request
2. The function name
3. The parameters of the function call [In most cases, the parameters are login details and HL7 message]
4. Function end
5. Soap envelope end

Here is a simple VXU message sent through the submitSingleMessage Web Service function. The pieces are numbered from the list above.

Item	Code
1. SOAP Envelope Start	<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cdc:iisb:2011"> <soap: Header/> <soap:Body>
2. Function tag	<urn:submitSingleMessage>
3. Parameter, the HL7 message, segments vary depending on the function	<urn:username>insert_your_username</urn:username> <urn:password>insert_your_password</urn:password> <urn:facilityID>insert_your_facility_ID</urn:facilityID> <urn:hl7Message>insert_your_HL7_message </urn:hl7Message>
4. Function End	</urn:submitSingleMessage>
5. SOAP Envelope End	</soap:Body> </soap:Envelope>

If successful, the SOAP response will contain an embedded HL7 ACK message. The message will look similar to this:

<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope">

<S:Body>

<submitSingleMessageResponse xmlns="urn:cdc:iisb:2011">

<return>MSH|^~\&|MIIC 5.1.17^^|MIIC^^|||20130117||ACK^|20100127GM901676|P^|2.4^^|||ER
MSA|AA|20100127GM901676||0||0^Message Accepted^HL70357^^^</return>

</submitSingleMessageResponse>

</S:Body>

</S:Envelope>

Getting the WSDL

WSDL (Web Services Definition Language, pronounced whiz-dul) is a method used to describe Web Services functions that can be accessed through the Internet from other computers.

The contents of the Test version of our WSDL are included in Appendix A, for reference. Using the WSDL file, modern development environments can turn the WSDL file into a programming interface to simplify implementation.

The WSDL can be obtained in two ways:

1. Retrieve it from the Web Service Server.
2. Extract the version from this document.

Method 1: Retrieve the WSDL from the web service server

This address can be used to retrieve the WSDL into a Web browser or a development tool to generate a programming interface. The steps for working with the programming interface are beyond the scope of this documentation.

1. Enter the address into the browser address bar or the development tool.
   - Production: https://miic.health.state.mn.us/miic-ws/client_Service?wsdl
   - Test: https://miic.health.state.mn.us/miic-ws-test/client_Service?wsdl
2. The WSDL appears in the Web browser and can be printed or saved for reference.

Method 2: Extract the version from this document

The full content for the Test environment WSDL can be found on pages 7-10 of this document. The only difference in the Test and Production WSDL files is the address location field at the end of the file.

1. Highlight the lines in Appendix A (the WSDL section only) from <?xml version="1.0" encoding="UTF-8" ?> to </definitions>
2. Copy the highlighted text to the clipboard with Control-C.
3. Open up Notepad and paste the contents of the clipboard into the empty Notepad file.
4. Verify the contents include all of the lines you highlighted in Step 1.

Getting credentials

Contact the MIIC Help Desk at health.miichelp@state.mn.us to request credentials. MIIC staff will provide them to you as you work through the process found for Onboarding for Electronic Data Exchange with MIIC (www.health.state.mn.us/people/immunize/miic/provider/dxprocess.html).

HL7 file testing and verification

HL7 file testing and verification will happen after setup of the web service but is a necessary check before moving from the test environment to the production environment. MIIC staff perform a number of tests and checks to ensure completeness and accuracy of the data. HL7 specifications can be found at [link to new Electronic Data Exchange with MIIC webpage].

Appendix A: Test WSDL and XSD File Contents

The following are the contents of the Test version of the WSDL.

1. <?xml version="1.0" encoding="UTF-8" ?>
   - <!--
2. Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.1.1-b09 (branches/2.1-6834; 2011-07-16T17:14:48+0000) JAXWS-RI/2.2.5-promoted-b04 JAXWS/2.2.
   -->
3. - <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsaw="http://www.w3.org/2005/08/addressing" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:tns="urn:cdc:iisb:2011" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"

targetNamespace="urn:cdc:iisb:2011" name="IISServiceNew">
- <!--

4. schema for types
   -->
5. - <types>
6. - <xsd:schema elementFormDefault="qualified" targetNamespace="urn:cdc:iisb:2011">
7. - <xsd:complexType name="connectivityTestRequestType">
8. - <xsd:sequence>
9. <xsd:element name="echoBack" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true" />
10. </xsd:sequence>
11. </xsd:complexType>
12. - <xsd:complexType name="connectivityTestResponseType">
13. - <xsd:sequence>
14. <xsd:element name="return" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true" />
15. </xsd:sequence>
16. </xsd:complexType>
17. - <xsd:complexType name="submitSingleMessageRequestType">
18. - <xsd:sequence>
19. <xsd:element name="username" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true" />
20. <xsd:element name="password" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true" />
21. <xsd:element name="facilityID" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true" />
22. <xsd:element name="hl7Message" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true" />
23. </xsd:sequence>
24. </xsd:complexType>
25. - <xsd:complexType name="submitSingleMessageResponseType">
26. - <xsd:sequence>
27. <xsd:element name="return" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true" />
28. </xsd:sequence>
29. </xsd:complexType>
30. - <xsd:complexType name="soapFaultType">
31. - <xsd:sequence>
32. <xsd:element name="Code" type="xsd:integer" minOccurs="1" />
33. <xsd:element name="Reason" type="xsd:string" minOccurs="1" />
34. <xsd:element name="Detail" type="xsd:string" minOccurs="1" />

```
</xsd:sequence>
</xsd:complexType>
- <xsd:complexType name="UnsupportedOperationFaultType">
- <xsd:sequence>
<xsd:element name="Code" type="xsd:integer" minOccurs="1" />
<xsd:element name="Reason" fixed="UnsupportedOperation" />
<xsd:element name="Detail" type="xsd:string" minOccurs="1" />
</xsd:sequence>
</xsd:complexType>
- <xsd:complexType name="SecurityFaultType">
- <xsd:sequence>
<xsd:element name="Code" type="xsd:integer" minOccurs="1" />
<xsd:element name="Reason" fixed="Security" />
<xsd:element name="Detail" type="xsd:string" minOccurs="1" />
</xsd:sequence>
</xsd:complexType>
- <xsd:complexType name="MessageTooLargeFaultType">
- <xsd:sequence>
<xsd:element name="Code" type="xsd:integer" minOccurs="1" />
<xsd:element name="Reason" fixed="MessageTooLarge" />
<xsd:element name="Detail" type="xsd:string" minOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:element name="connectivityTest" type="tns:connectivityTestRequestType" />
<xsd:element name="connectivityTestResponse" type="tns:connectivityTestResponseType" />
<xsd:element name="submitSingleMessage" type="tns:submitSingleMessageRequestType" />
<xsd:element name="submitSingleMessageResponse" type="tns:submitSingleMessageResponseType" />
<xsd:element name="fault" type="tns:soapFaultType" />
<xsd:element name="UnsupportedOperationFault" type="tns:UnsupportedOperationFaultType" />
<xsd:element name="SecurityFault" type="tns:SecurityFaultType" />
<xsd:element name="MessageTooLargeFault" type="tns:MessageTooLargeFaultType" />
</xsd:schema>
```

```
67. </types>
    - <!--
68. Message definitions
    -->
69. - <message name="connectivityTest_Message">
70. <documentation>connectivity test request</documentation>
71. <part name="parameters" element="tns:connectivityTest" />
72. </message>
73. - <message name="connectivityTestResponse_Message">
74. <documentation>connectivity test response</documentation>
75. <part name="parameters" element="tns:connectivityTestResponse" />
76. </message>
77. - <message name="submitSingleMessage_Message">
78. <documentation>submit single message request.</documentation>
79. <part name="parameters" element="tns:submitSingleMessage" />
80. </message>
81. - <message name="submitSingleMessageResponse_Message">
82. <documentation>submit single message response</documentation>
83. <part name="parameters" element="tns:submitSingleMessageResponse" />
84. </message>
85. - <message name="UnknownFault_Message">
86. <part name="fault" element="tns:fault" />
87. </message>
88. - <message name="UnsupportedOperationFault_Message">
89. <part name="fault" element="tns:UnsupportedOperationFault" />
90. </message>
91. - <message name="SecurityFault_Message">
92. <part name="fault" element="tns:SecurityFault" />
93. </message>
94. - <message name="MessageTooLargeFault_Message">
95. <part name="fault" element="tns:MessageTooLargeFault" />
96. </message>
    - <!--
```

```
97. Operation/transaction declarations
    -->
98. - <portType name="IIS_PortType">
99. - <operation name="connectivityTest">
100.    <documentation>the connectivity test</documentation>
101.    <input message="tns:connectivityTest_Message" wsaw:Action="urn:cdc:iisb:2011:connectivityTest" />
102.    <output message="tns:connectivityTestResponse_Message"
    wsaw:Action="urn:cdc:iisb:2011:connectivityTestResponse" />
103.    <fault name="UnknownFault" message="tns:UnknownFault_Message" />
    - <!—
104.    a general soap fault
    -->
105.    <fault name="UnsupportedOperationFault" message="tns:UnsupportedOperationFault_Message" />
    - <!--
106.    The UnsupportedOperation soap fault
    -->
107.    </operation>
108.    - <operation name="submitSingleMessage">
109.    <documentation>submit single message</documentation>
110.    <input message="tns:submitSingleMessage_Message"
    wsaw:Action="urn:cdc:iisb:2011:submitSingleMessage" />
111.    <output message="tns:submitSingleMessageResponse_Message"
    wsaw:Action="urn:cdc:iisb:2011:submitSingleMessageResponse" />
112.    <fault name="UnknownFault" message="tns:UnknownFault_Message" />
    - <!--
113.    a general soap fault
    -->
114.    <fault name="SecurityFault" message="tns:SecurityFault_Message" />
115.    <fault name="MessageTooLargeFault" message="tns:MessageTooLargeFault_Message" />
116.    </operation>
117.    </portType>
    - <!--
118.    SOAP 1.2 Binding
    -->
119.    - <binding name="client_Binding_Soap12" type="tns:IIS_PortType">
120.    <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
121.    - <operation name="connectivityTest">
```

```
<soap12:operation soapAction="urn:cdc:iisb:2011:connectivityTest" />
- <input>
<soap12:body use="literal" />
</input>
- <output>
<soap12:body use="literal" />
</output>
- <fault name="UnknownFault">
<soap12:fault use="literal" name="UnknownFault" />
</fault>
- <fault name="UnsupportedOperationFault">
<soap12:fault use="literal" name="UnsupportedOperationFault" />
</fault>
</operation>
- <operation name="submitSingleMessage">
<soap12:operation soapAction="urn:cdc:iisb:2011:submitSingleMessage" />
- <input>
<soap12:body use="literal" />
</input>
- <output>
<soap12:body use="literal" />
</output>
- <fault name="UnknownFault">
<soap12:fault use="literal" name="UnknownFault" />
</fault>
- <fault name="SecurityFault">
<soap12:fault use="literal" name="SecurityFault" />
</fault>
- <fault name="MessageTooLargeFault">
<soap12:fault use="literal" name="MessageTooLargeFault" />
</fault>
</operation>
```

```
154.    </binding>
    - <!--
155.    Service definition
    -->
156.    - <service name="client_Service">
157.    - <port binding="tns:client_Binding_Soap12" name="client_Port_Soap12">
158.    <soap12:address location="https://miic.health.state.mn.us/miic-ws-test/client_Service"/>
159.
160.    </port>
161.    </service>
162.    </definitions>
```

Minnesota Department of Health
Minnesota Immunization Information Connection (MIIC)
PO Box 64975, St. Paul, MN 55164-0975
health.miichelp@state.mn.us | www.health.state.mn.us/miic

07/16/2026

To obtain this information in a different format, call: 651-201-5207.