What is a WSDL?

Web Service Description Language
An important extension of SOAP\(^1\) is the use of service description documents using Web Service Description Language (WSDL)\(^2\). As well as documenting the service interface, the WSDL for a service provides an interface contract, which service providers strive to maintain, thus maintaining compatibility with older clients. Another advantage of having a WSDL document describing a service is the availability of tools to generate client-side stub or server-side skeleton codes from the WSDL. This simplifies program development and also allow for the runtime discovery of services.

Given a WSDL service description tools such as soapUI provide a method for experimenting with the SOAP messages exchanges and the construction of test cases, by deriving the details of the SOAP messages from the information provided in the WSDL.

Anatomy
The WSDL document for a service defines the operations supported by a service and the formats for the messages required to invoke an operation and the message returned.

A WSDL 1.1 document has the following sections:

- "types": data structure types defined using XML Schema\(^3\)
- "message": the messages to be exchanged
- "portType": abstract definition of the operations provided by the service
- "bindings": binding for the operations that describe how to encode the messages
- "service": description of the endpoint (URL) to send messages to

Types
To allow for complex types to be exchanged data types are defined using XML Schema, either directly in the "types" section of the document or in external documents imported into the WSDL.

From the EBI Search service, the types required for the listDomains operation:

```
<wsdl:types>
  <xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    attributeFormDefault="qualified"
    elementFormDefault="qualified"
    targetNamespace="http://www.ebi.ac.uk/EBISearchService">
    <xsd:element
      name="listDomains">
      <xsd:complexType/>
    </xsd:element>
    <xsd:complexType name="ArrayOfString">
      <xsd:sequence>
        <xsd:element
          maxOccurs="unbounded"
          minOccurs="0"
          name="string"
          nillable="true"
          type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:element
      name="listDomainsResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element
            maxOccurs="1"
            minOccurs="1"
            name="arrayOfDomainNames"
            nillable="true"
            type="tns:ArrayOfString"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
```

Messages
Abstract definitions of the messages that are exchanged when operations are called. These use the types defined earlier.

From the EBI Search service, the messages for the listDomains operation:

```
<wsdl:message
  name="listDomainsResponse">
  <wsdl:part
    name="parameters" element="tns:listDomainsResponse"/>
</wsdl:message>
<wsdl:message
  name="listDomainsRequest">
  <wsdl:part
    name="parameters" element="tns:listDomains"/>
</wsdl:message>
```

Services which use the document/literal wrapped SOAP style (e.g. EBI Search) are required to have one, and only one, part per message, and the part must have the name parameters\(^4\). Some SOAP tool-kits treat document/literal and document/literal wrapped services slightly differently, thus being able to tell if a document/literal service is using the wrapped style can be useful when developing client software.

Port Type
Abstract definitions of the operations (methods) which are provided by the service. The inputs and outputs of the operations are defined using the messages defined above.

From the EBI Search service, the portType for the listDomains operation:

```xml
<wSDL:portType name="EBI_SearchService">
    <wSDL:operation name="listDomains">
        <wSDL:input name="listDomainsRequest" message="tns:listDomainsRequest"/>
        <wSDL:output name="listDomainsResponse" message="tns:listDomainsResponse"/>
    </wSDL:operation>
</wSDL:portType>
```

**Binding**

Unlike the operation definitions in the portType section the bindings define how the messages are exchanged (transport), formatted (style and use) and additional information about header information to be added to the message (e.g. SOAPAction). A WSDL may contain multiple bindings which use different “service” endpoints or alternative encodings.

The document/literal EBI Search service, has the following binding for the listDomains operation:

```xml
<wSDL:binding name="EBISearchServiceHttpBinding" type="tns:EBISearchService">
    <wSDLsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wSDL:operation name="listDomains">
        <wSDLsoap:operation soapAction="urn:ListDomains"/>
        <wSDL:input name="listDomainsRequest">
            <wSDLsoap:body use="literal"/>
        </wSDL:input>
        <wSDL:output name="listDomainsResponse">
            <wSDLsoap:body use="literal"/>
        </wSDL:output>
    </wSDL:operation>
</wSDL:binding>
```

Note the specification of the SOAP style using:

- style="document" in the wSDLsoap:binding element
- use="literal" in the wSDLsoap:body elements

The RPC/encoded WSDbfetch service, has the following binding for the getFormatStyles(db, format) operation:

```xml
<wSDL:binding name="WSDbfetchSoapBinding" type="impl:WSDBFetchServerLegacy">
    <wSDLsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wSDL:operation name="getFormatStyles">
        <wSDLsoap:operation soapAction="/"/>
        <wSDL:input name="getFormatStylesRequest">
            <wSDLsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="http://wdbfetch.ws.jdbfetch.ebi.ac.uk"
                use="encoded"/>
        </wSDL:input>
        <wSDL:output name="getFormatStylesResponse">
            <wSDLsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="http://www.ebi.ac.uk/ws/services/WSDbfetch"
                use="encoded"/>
        </wSDL:output>
        <wSDL:fault name="DbfParamsException">
            <wSDLsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                name="DbfParamsException"
                namespace="http://www.ebi.ac.uk/ws/services/WSDbfetch"
                use="encoded"/>
        </wSDL:fault>
    </wSDL:operation>
</wSDL:binding>
```

Note the specification of the SOAP style using:

- style="rpc" in the wSDLsoap:binding element
- use="encoded" in the wSDLsoap:body elements

**Service**

The “service” defines where the messages are sent. As with the bindings a WSDL may contain multiple “service” definitions.

From the EBI Search service, the service for the listDomains operation:
**WSDL Examples**

**Document/literal WSDL**

A listDomains() specific WSDL for the **EBI Search** service:
RPC/encoded WSDL

A getFormatStyles (db, format) specific WSDL for the WSDbfetch service:

```xml
<wsdl:definitions
    xmlns:apachesoap="http://xml.apache.org/xml-soap"
    xmlns:impl="http://www.ebi.ac.uk/ws/services/WSDbfetch"
    xmlns:intf="http://www.ebi.ac.uk/ws/services/WSDbfetch"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsdl="http://exceptions.jdbfetch.ebi.ac.uk"/>
```
<wsdl:types>
  <complexType name="ArrayOf_xsd_string">
    <complexContent>
      <restriction base="soapenc:Array">
        <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
      </restriction>
    </complexContent>
  </complexType>
  <complexType name="InputException">
    <sequence/>
  </complexType>
  <complexType name="DbfException">
    <sequence/>
  </complexType>
  <complexType name="DbfParamsException">
    <complexContent>
      <extension base="tns1:DbfException"/>
    </complexContent>
  </complexType>
</wsdl:types>

<wsdl:message name="DbfParamsException">
  <wsdl:part name="fault" type="tns1:DbfParamsException"/>
</wsdl:message>

<wsdl:message name="getFormatStylesResponse">
  <wsdl:part name="getFormatStylesReturn" type="impl:ArrayOf_xsd_string"/>
</wsdl:message>

<wsdl:message name="getFormatStylesRequest">
  <wsdl:part name="db" type="xsd:string"/>
  <wsdl:part name="format" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="getSupportedFormatsRequest">
</wsdl:message>

<wsdl:portType name="WSDBFetchServerLegacy">
  <wsdl:operation name="getFormatStyles" parameterOrder="db format">
    <wsdl:input message="impl:getFormatStylesRequest" name="getFormatStylesRequest"/>
    <wsdl:output message="impl:getFormatStylesResponse" name="getFormatStylesResponse"/>
    <wsdl:fault message="impl:DbfParamsException" name="DbfParamsException"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="WSDbfetchSoapBinding" type="impl:WSDBFetchServerLegacy">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getFormatStyles">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getFormatStylesRequest"/>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/
    namespace="http://wadbfetch.ws.jdbfetch.ebi.ac.uk" use="encoded"/>
  </wsdl:operation>
</wsdl:binding>

<wsdl:service name="WSDBFetchServerLegacyService">
  <wsdl:port binding="impl:WSDbfetchSoapBinding" name="WSDbf"/>
<wsdl:service name="WSDbfetch">
  <wsdl:port name="DBfetchPort">
    <wsdlsoap:address location="http://www.ebi.ac.uk/ws/services/WSDbfetch"/>
  </wsdl:port>
</wsdl:service>

1) SOAP 1.1: http://www.w3.org/TR/soap11/
2) SOAP 1.2: http://www.w3.org/TR/soap12/
3) WSDL 1.1: http://www.w3c.org/TR/wsd1
4) WSDL 2.0: http://www.w3.org/TR/wsdl20/