

What is WSDL?

- Web Services Description Language
- An XML document
- Describe Web services
 - location
 - operations

WSDL Syntax

```

<definitions>
  <types>
  Define types and element using schema for later use in the part elements
  </types>

  <message>
  <part> – messages used by the web service – function call
  </message>

  <portType> ----- most important (operation and message)
  The operations performed by the web service
  </portType>

  <binding>
  communication protocols used by the web service
  </binding>

</definitions>
    
```

WSDL Example

```

<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms"> ----- function library, like Class in Java
  <operation name="getTerm"> ----- function
    <input message="getTermRequest"/> ----- input parameter
    <output message="getTermResponse"/> ----- output parameter
  </operation>
</portType>
    
```

part Element

<pre> <message name="PriceCheckRequest"> <part name="sku" element="avail:sku"/> </message> <types> <schema:element name="sku" type="schema:string"/> </types> </pre>	<pre> <soapenv:Body> <sku xmlns="http://www.example.com/ns/availability"> 123 </sku> </soapenv:Body> </pre> <p>document-oriented style</p>
<pre> <message name="PriceCheckRequest"> <part name="sku" type="xs:string"/> </message> </pre>	<pre> <soapenv:Body> <ns1:checkprice xmlns:ns1="http://www.example.com/services/PriceCheck" > <sku xsi:type="schema:string"> 123 </sku> </ns1:checkprice> </soapenv:Body> </pre> <p>RPC-style</p>

portType Element

Type	Description
One-way	receive a message, no response
Request-response	receive a request, return a response
Solicit-response	send a request, wait for a response
Notification	send a message, not wait for a response

One-way Operation Example

```

<message name="newTermValues">
  <part name="term" type="xs:string"/>
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="setTerm">
    <input name="newTerm" message="newTermValues"/>
  </operation>
</portType >
    
```

Request-Response Operation Example

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

Binding Element

- How to format the message to invoke operations (SOAP over HTTP, SOAP over SMTP, HTTP POST, or other networking and message protocol)
- Type attribute points to the port
- Name attribute can be anything

```
<binding type="glossaryTerms" name="b1">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<operation>
  <soap:operation
    soapAction="http://example.com/getTerm"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
```

Binding Example

```
<portType name="PriceCheckPortType">
  <operation name="checkPrice">
    <input message="pc:PriceCheckRequest"/>
    <output message="pc:PriceCheckResponse"/>
  </operation>
</portType>

<!-- Binding definitions -->
<binding name="PriceCheckSOAPBinding" type="pc:PriceCheckPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="checkPrice">
    <soap:operation soapAction="http://www.skatestown.com/services/PriceCheck/checkPrice" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
```

Binding Element

```
<operation name="checkPrice">
  <soap:operation
    soapAction="http://www.skatestown.com/services/PriceCheck/checkPrice" />
```

HTTP header:

```
...
Content-Type: text/xml; charset=utf-8
Host: ...
Cache-Control: no-cache
SOAPAction: "http://www.skatestown.com/services/PriceCheck/checkPrice"
Content-Length: 334
```

input Element

WSDL:

```
<input>
  <soap:body use="literal" />
</input>
```

SOAP:

```
<soap:body>
  <sku xmlns="http://www.example.com/ns/availability">123</sku>
</soap:body>
```

service Element

```
<service name="PriceCheck">
  <port name="Pricecheck" binding="pc:PriceCheckSOAPBinding">
    <documentation>
      <wsi:Claim
        conformsTo="http://ws-i.org/profiles/basic/1.0" />
    </documentation>
    <soap:address location="http://www.skatestown.com/services/PriceCheck"/>
  </port>
</service>
```

- soap:address – the URL to which the SOAP message should be sent to invoke the priceCheck operation

WSDL to Java

```
<portType name="PriceCheckPortType">
  <operation name="checkPrice">
    <input message="pc:PriceCheckRequest"/>
    <output message="pc:PriceCheckResponse"/>
  </operation>
</portType>
<definitions name="PriceCheck"
  targetNamespace="http://www.skatestown.com/services/PriceCheck"
  xmlns:pc="http://www.skatestown.com/services/PriceCheck"
  xmlns:avail="http://www.skatestown.com/ns/availability">
  -----
  package com.skatestown.www.service.priceCheck;
  public interface PriceCheckPortType extends java.rmi.Remote{

  public com.skatestown.www.ns.availability.AvailabilityType checkPrice (java.lang.String sku) throws
  java.rmi.RemoteException{

  }
  }
  Remote Method Invocation \(Java RMI\)
```

WSDL to Java

```
<portType name="PriceCheckPortType">
  <operation name="checkPrice">
    <input message="pc:PriceCheckRequest"/>
    <output message="pc:PriceCheckResponse"/>
  </operation>
</portType>
<definitions name="PriceCheck"
  targetNamespace="http://www.skatestown.com/services/PriceCheck"
  xmlns:pc="http://www.skatestown.com/services/PriceCheck"
  xmlns:avail="http://www.skatestown.com/ns/availability">
  -----
  package com.skatestown.www.service.priceCheck;
  public interface PriceCheckPortType extends java.rmi.Remote{

  public com.skatestown.www.ns.availability.AvailabilityType checkPrice (java.lang.String sku) throws
  java.rmi.RemoteException{

  }
  }
  }
```

binding -> implementation of the portType interface

```
<binding name="PriceCheckSOAPBinding" type="pc:PriceCheckPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="checkPrice">
    <soap:operation soapAction="http://www.skatestown.com/services/PriceCheck/checkPrice" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
-----
package com.skatestown.www.service.priceCheck;
public class PriceCheckSOAPBindingStub implements PriceCheckPortType{

public com.skatestown.www.ns.availability.AvailabilityType checkPrice (java.lang.String sku) throws
java.rmi.RemoteException{

  ....
}
}
```