

Introduction to Web Services (SOAP & REST)

Working with JAX-WS

Lesson Objectives

- Working with JAX-WS
 - WSDL
 - SOAP
- Creating JAX-WS service
- Consuming web service



Copyright © Capgemini 2015. All Rights Reserved 2

Following contents would be covered:

- 2.1: Working with JAX-WS
 - 2.1.1: What is WSDL
 - 2.1.2: Structure of WSDL
 - 2.1.3: Generating WSDL
 - 2.1.4: What is SOAP
 - 2.1.5: Structure of SOAP
- 2.2: Creating JAX-WS service
- 2.3: Consuming JAX-WS service

2.1: Overview

Working with JAX - WS

- Java API for XML web services (JAX-WS), is a set of APIs for creating web services in XML format
- In JAX-WS, a web service operation invocation is represented by an XML-based protocol, such as SOAP. The SOAP specification defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses. These calls and responses are transmitted as SOAP messages (XML files) over HTTP
- Although SOAP messages are complex, the JAX-WS API hides this complexity from the application developer. On the server side, the developer specifies the web service operations by defining methods in an interface written in the Java programming language.

2.1.1: WSDL

What is WSDL ?

- WSDL stands for Web Services Description Language
- WSDL is a document written in XML. This document describes a Web service. It specifies the location of the service and the operations (or the methods) the service exposes
- WSDL documents uses these major elements
 - <types>
 - Defines the datatypes (XML Schemas) used by the web service
 - <message>
 - Defines the data elements for each operation
 - <portType>
 - Describes the operations that can be performed and messages involved
 - <binding>
 - Defines the protocol and data format for each port type



Copyright © Capgemini 2015. All Rights Reserved 4

The main structure of WSDL document looks as below:

<definitions>

<types>

data type definitions.....

</types>

<message>

definition of the data being communicated....

</message>

<portType>

set of operations.....

</portType>

<binding>

protocol and data format specification....

</binding>

</definitions>

2.1.1: WSDL

WSDL Elements

- Below is the snap shot of the WSDL file:

```
<definitions name="CalculatorService" targetNamespace="http://webservice.learning.cg.com/"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://webservice.learning.cg.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:w3am="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <types/>
  - <message name="subtraction">
    <part name="arg0" type="xsd:int"/>
    <part name="arg1" type="xsd:int"/>
  </message>
  - <message name="subtractionResponse">
    <part name="return" type="xsd:int"/>
  </message>
```

Message -> Definition of data being communicated



Copyright © Capgemini 2015. All Rights Reserved 5

Add the notes here.

2.1.1: WSDL

WSDL Elements

- Below is the snap shot of the WSDL file:

```
<portType name="CalculatorServer">
- <operation name="subtraction" parameterOrder="arg0 arg1">
  <input message="tns:subtraction" wsam:Action="http://webservice.learning.cg.com/CalculatorServer/subtractionRequest"/>
  <output message="tns:subtractionResponse"
    wsam:Action="http://webservice.learning.cg.com/CalculatorServer/subtractionResponse"/>
</operation>
- <operation name="multiplication" parameterOrder="arg0 arg1">
  <input message="tns:multiplication"
    wsam:Action="http://webservice.learning.cg.com/CalculatorServer/multiplicationRequest"/>
  <output message="tns:multiplicationResponse"
    wsam:Action="http://webservice.learning.cg.com/CalculatorServer/multiplicationResponse"/>
</operation>
```

Port type-> contains set of operations

Add the notes here.

2.1.1: WSDL

WSDL Elements

- Below is the snap shot of the WSDL file:

```
<binding name="CalculatorPortBinding" type="tns:CalculatorServer">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  - <operation name="subtraction">
    <soap:operation soapAction=""/>
    - <input>
      <soap:body namespace="http://webservice.learning.cg.com/" use="literal"/>
    </input>
    - <output>
      <soap:body namespace="http://webservice.learning.cg.com/" use="literal"/>
    </output>
  </operation>
```

Binding->Defines Protocol and data format

Add the notes here.

2.1.1: WSDL

WSDL Elements

- Below is the snap shot of the WSDL file:

```
- <service name="CalculatorService">  
  - <port name="CalculatorPort" binding="tns:CalculatorPortBinding">  
    <soap:address location="http://localhost:9876/cs"/>  
  </port>  
</service>
```

Service: Describes the service name to be used , port name and the soap address, where the web service will be available for consumption

Add the notes here.

2.1.2: SOAP

What is SOAP

- SOAP is Simple Object Access Protocol
- Is a format for sending and receiving messages
- It is important for web applications to be able to communicate over the internet
- The best way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this
- SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages



Copyright © Capgemini 2015. All Rights Reserved 9

SOAP stands for **S**imple **O**bject **A**ccess **P**rotocol

SOAP is an application communication protocol

SOAP is a format for sending and receiving messages

SOAP is platform independent

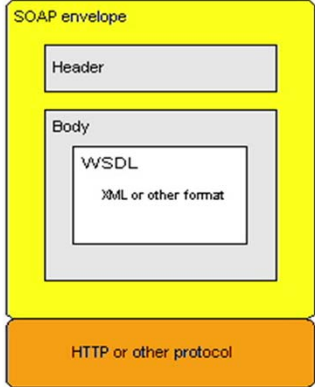
SOAP is based on XML

SOAP is a W3C recommendation

2.1.2: SOAP

SOAP Building Blocks

- Basic SOAP consists of Envelope, Header and Body:



```
<?xml version="1.0"?>

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

  <soap:Header>
    ...
  </soap:Header>

  <soap:Body>
    ...
  </soap:Body>

```

Capgemini
CONSULTING TECHNOLOGY ENTREPRENEURS

Copyright © Capgemini 2015. All Rights Reserved 10

A SOAP message is an ordinary XML document containing the following elements:

- An Envelope element that identifies the XML document as a SOAP message
- A Header element that contains header information
- A Body element that contains call and response information

A few syntax rules.....

- A SOAP message MUST be encoded using XML
- A SOAP message MUST use the SOAP Envelope namespace
- A SOAP message MUST use the SOAP Encoding namespace
- A SOAP message must NOT contain a DTD reference
- A SOAP message must NOT contain XML Processing Instructions

2.1.2: SOAP

SOAP Request and Response

- Following demonstrates a SOAP request

```
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: n

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

<soap:Body xmlns:m="http://webservice.learning.cg.com/">
<m:addition>
  <m:arg0>12</m:arg0>
  <m:arg1>10</m:arg1>
</m:addition>
</soap:Body>
</soap:Envelope>
```

- Following demonstrates a SOAP response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: n

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

<soap:Body xmlns:m="http://webservice.learning.cg.com/">
<m:additionResponse>
  <m:response>22</m:response>
</m:additionResponse>
</soap:Body>
</soap:Envelope>
```



Copyright © Capgemini 2015. All Rights Reserved 11

In the example above, an addition request is sent to a server. The request has 2 parameters as number1 and number2 and addition of 2 numbers is returned in the response. The namespace for the function is defined in "http://webservice.learning.cg.com/".

2.2: Creating Service

Creating JAX- WS Service

End Point Interface

```
@WebService
@SOAPBinding(style = Style.RPC)
public interface CalculatorServer {

    @WebMethod
    int addition(int param1, int param2);

    @WebMethod
    int subtraction(int param1, int param2);

    @WebMethod
    int multiplication(int param1, int param2);

    @WebMethod
    int division(int param1, int param2);

    @WebMethod
    int modulus(int param1, int param2);
}
```

Implementation class

```
@WebService(endpointInterface = "com.cg.learning.webservice.CalculatorServer")
public class Calculator {

    public int addition(int param1, int param2) {
        return param1 + param2;
    }

    public int subtraction(int param1, int param2) {
        return param1 - param2;
    }

    public int multiplication(int param1, int param2) {
        return param1 * param2;
    }

    public int division(int param1, int param2) {
        return param1 / param2;
    }

    public int modulus(int param1, int param2) {
        return param1 % param2;
    }
}
```

The starting point for developing a JAX-WS web service is a Java class annotated with the `javax.jws.WebService` annotation.

The `WebService` annotation defines the class as a web service endpoint. SOAP binding style is given as `RPC`.

A *service endpoint interface* (SEI) is a Java interface that declares the methods that a client can invoke on the service

The web service implementation class implicitly defines a SEI.

You may specify an explicit SEI by adding the `endpointInterface` element to the `WebService` annotation in the implementation class.

You must then provide a SEI that defines the public methods made available in the endpoint implementation class.

2.2: Creating Service

Creating JAX – WS Service

- The web service needs to be published so that it can await service requests
- Refer below screen shot for same

```
public class CalculatorPublisher {  
    public static void main(String[] args) {  
        // 1st argument is the publication URL  
        // 2nd argument is an SIB instance  
        Endpoint.publish("http://127.0.0.1:9876/cs", new Calculator());  
    }  
}
```



Copyright © Capgemini 2015. All Rights Reserved 13

Currently the service is published at network address 127.0.0.1 which is localhost, and at port number 9876,

The application path /cs is an arbitrary name.

The Endpoint class has an overloaded publish method. In this two-argument version, the first argument is the publication URL as a string and the second argument is an instance of the service SIB, in this case com.cg.learning.Calculator.

The application runs indefinitely, awaiting service requests.

It needs to be terminated at the command prompt with control-C or the equivalent.

You can check the following URL in browser:

<http://127.0.0.1:9876/cs?wsdl>

To view the service contract, the WSDL document.

2.3: Consuming a Service

Consuming JAX – WS Service

- Refer below screen shot to consume a service:

```
public static void main(String args[]) throws Exception {
    URL url = new URL("http://localhost:9876/cs?wsdl");

    QName qname = new QName("http://webservice.learning.cg.com/",
        "CalculatorService");

    // Create, in effect, a factory for the service.
    Service service = Service.create(url, qname);

    // Extract the endpoint interface, the service "port".
    CalculatorServer endPointIntf = service.getPort(CalculatorServer.class);

    System.out.println("Addition::\t" + endPointIntf.addition(12,10));
}
```



Copyright © Capgemini 2015. All Rights Reserved 14

Here basic XML API is being used to consume a service.

The URL defines the location of wsdl document.

The QName (qualified name) prints the service URI (as to where the service will be available). "CalculatorService" is the service name to be exposed in wsdl document.


Next we create a factory for the service with the URL and QName at a port number where SEI (Service Endpoint Interface) is available.


Lastly by using endPoint interface we can consume the service.

2.3: Consuming a Service

Demo

- JAX-WS-Calculator



 Capgemini
CONSULTING TECHNOLOGY ENTREPRENEUR

Copyright © Capgemini 2015. All Rights Reserved 15

Add the notes here.

Summary


- We have so far learnt
 - What are JAX-WS
 - What are wsdl documents and their representation
 - What are soap messages and their request and response structure
 - How to create a JAX – WS service
 - How to consume a JAX – WS service




Add the notes here.

Working with JAX-WS
Lab

- Lab 1



 Capgemini
CONSULTING TECHNOLOGY ENTREPRENEUR

Copyright © Capgemini 2015. All Rights Reserved 17

Add the notes here.

Review Question

- Question 1: What gives information about the web service location and operation?
 - SOAP message structure
 - WSDL document
 - Service End Point interface
- Question 2: @WebMethod signals that each method is a service operation (True or False)
 - True
 - False
- Question 3: _____ is used to define protocol and data format?
 - Message
 - Port type
 - Binding



Add the notes here.