

CC Lab 3: Sending a request and handling a response from a SOAP web service.

Prerequisites: **Java Environment.**

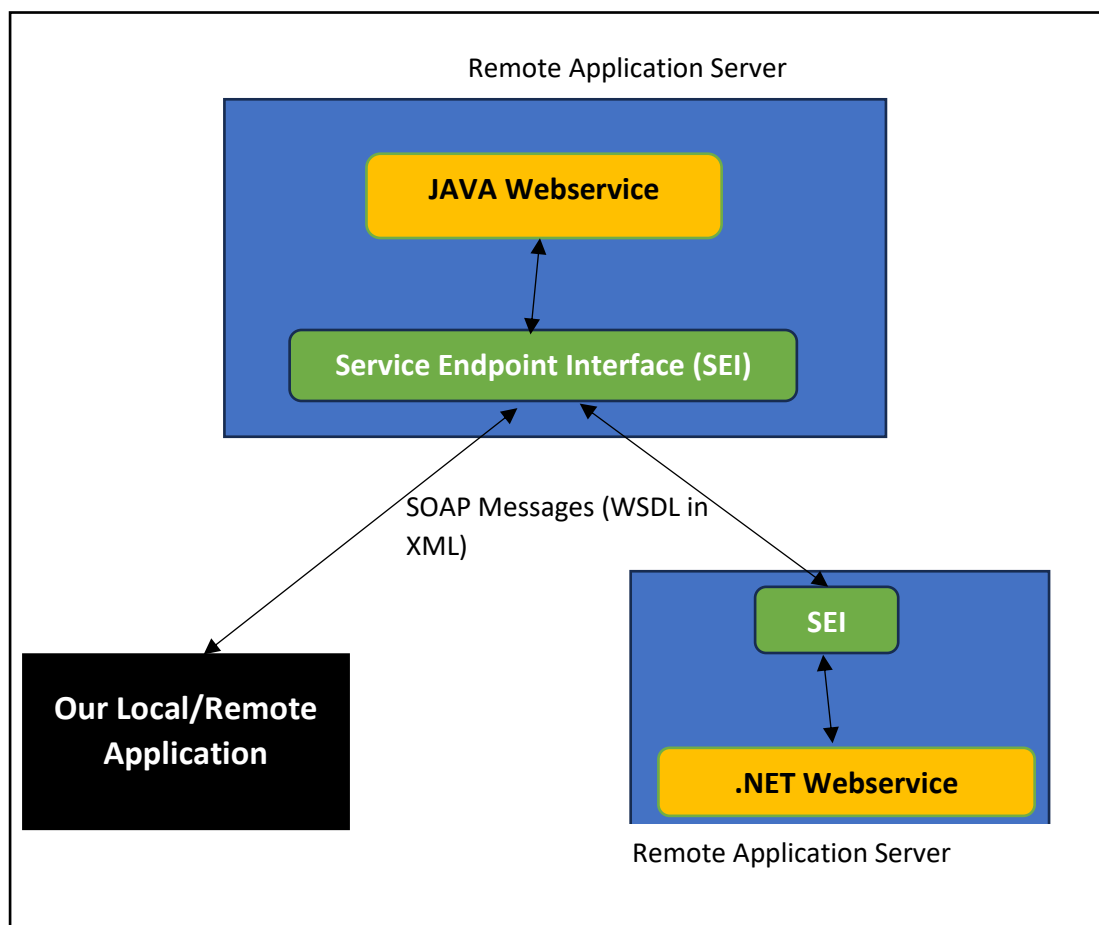
Web-Service: A service that is made available over the web.

Difference between a website and a webservice is that a website is meant and designed for human interaction/consumption whereas a webservice is meant for code/application-level consumption.

WSDL: Web Services Description Language is an XML notation for describing a web service.

A WSDL definition tells a client how to compose a web service request and describes the interface that is provided by the web service provider.

- Following is a high-level overview of the use of SOAP messages.




1. Now, we need a demo webservice to perform our experiment. And we need its WSDL in order to understand what operations does that webservice support. The WSDL can be acquired from follows:

<https://apps.learnwebservices.com/services/hello?WSDL>

The webservice hosted on the above URL takes your name as input and greets you with a Hello <Your-Name> message.

2. Now if you visit the URL you can see the WSDL in XML format, it tells us what input is the webservice expecting and of what datatype. Additionally, it also tells us the output type as highlighted in the image below:

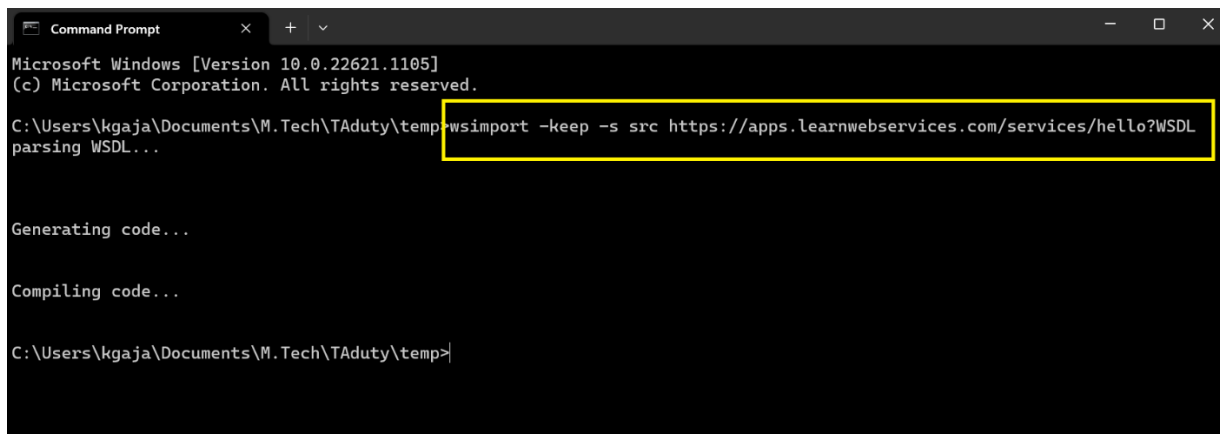


```
▼<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://learnwebservices.com/services/hello" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" name="HelloEndpointService" targetNamespace="http://learnwebservices.com/services/hello">
  ▼<wsdl:types>
    ▼<xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://learnwebservices.com/services/hello" attributeFormDefault="unqualified" elementFormDefault="qualified" targetNamespace="http://learnwebservices.com/services/hello" version="1.0">
      ▼<xs:complexType name="helloRequest">
        ▼<xs:sequence>
          <xs:element name="Name" type="xsd:string"/>
        </xs:sequence>
      </xs:complexType>
      ▼<xs:complexType name="helloResponse">
        ▼<xs:sequence>
          <xs:element name="Message" type="xsd:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="HelloRequest" nillable="true" type="helloRequest"/>
      <xs:element name="HelloResponse" nillable="true" type="helloResponse"/>
    </xs:schema>
  </wsdl:types>
  ►<wsdl:message name="SayHelloResponse">
```

The image shows a screenshot of a WSDL (Web Service Description Language) XML document. The document is displayed in a dark-themed editor. The XML code is color-coded. Two specific parts of the code are highlighted with yellow rectangular boxes. The first box highlights the input type definition: `<xs:complexType name="helloRequest">`, `<xs:sequence>`, and `<xs:element name="Name" type="xsd:string"/>`. The second box highlights the output type definition: `<xs:complexType name="helloResponse">`, `<xs:sequence>`, and `<xs:element name="Message" type="xsd:string"/>`. Both boxes also encompass the closing tags for the `sequence` and `complexType` elements. The rest of the WSDL code, including the `definitions` root, the `targetNamespace`, and the `message` element at the bottom, is visible but not highlighted.

3. Now execute the “wsimport” command as shown in the below image passing the above URL of webservice as an input parameter. This will generate all the java files

relevant to our webservice containing all the supported operations. After successful execution of this command you will have two folders, “com” and “src”. Here, “src” contains the source Java files(package) required to interact with the web service.



```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

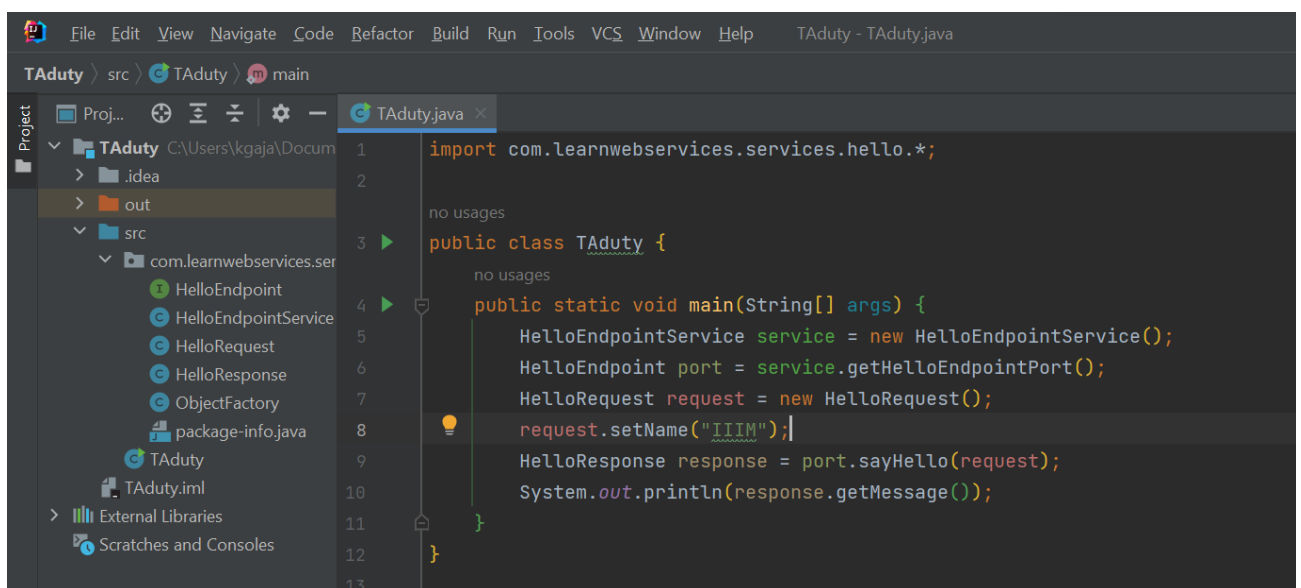
C:\Users\kgaja\Documents\M.Tech\TAduty\temp>wsimport -keep -s src https://apps.learnwebservices.com/services/hello?WSDL
parsing WSDL...

Generating code...

Compiling code...

C:\Users\kgaja\Documents\M.Tech\TAduty\temp>
```

4. Now we create a Java project, under which we import the source package we acquired from wsimport command and call the SOAP webservice as demonstrated below:



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help TAduty - TAduty.java

TAduty > src > TAduty > main

Project
  TAduty C:\Users\kgaja\Documents\M.Tech\TAduty
    .idea
    out
    src
      com.learnwebservices.services
        HelloEndpoint
        HelloEndpointService
        HelloRequest
        HelloResponse
        ObjectFactory
        package-info.java
        TAduty
        TAduty.iml
    External Libraries
    Scratches and Consoles

TAduty.java
1  import com.learnwebservices.services.hello.*;
2
3  public class TAduty {
4      public static void main(String[] args) {
5          HelloEndpointService service = new HelloEndpointService();
6          HelloEndpoint port = service.getHelloEndpointPort();
7          HelloRequest request = new HelloRequest();
8          request.setName("IIIM");
9          HelloResponse response = port.sayHello(request);
10         System.out.println(response.getMessage());
11     }
12 }
13
```

5. Output:



The screenshot shows the Run console of an IDE. At the top, there is a tab labeled "Run: TAduty x". Below the tab, the command executed is shown: `"C:\Program Files\Java\jdk1.8.0_341\bin\java.exe" ...`. The output of the program is `Hello IIIM!`, which is highlighted with a yellow rectangular box. Below the output, the status message `Process finished with exit code 0` is displayed. On the left side of the console, there are several icons: a green play button, a wrench, a square, a camera, and a list icon.

```
Run: TAduty x
"C:\Program Files\Java\jdk1.8.0_341\bin\java.exe" ...
Hello IIIM!
Process finished with exit code 0
```