# **CC Lab 4:** Sending a request and handling a response from a SOAP web service – Part 2

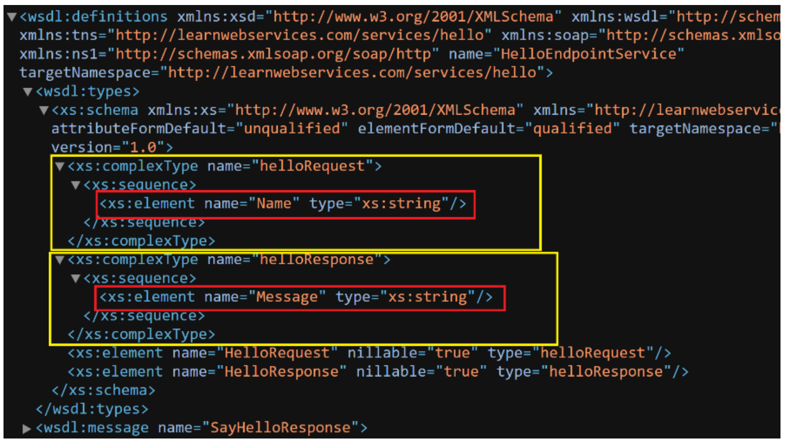
Prerequisites: **Java Environment.**

1. Following is the demo web service we will use for this experiment.

   [https://apps.learnwebservices.com/services/tempconverter?wsdl](https://apps.learnwebservices.com/services/tempconverter?wsdl)

   The webservice hosted on the above URL takes the temperature in either Celsius or Fahrenheit and returns the converted temperature in Fahrenheit or Celsius respectively.

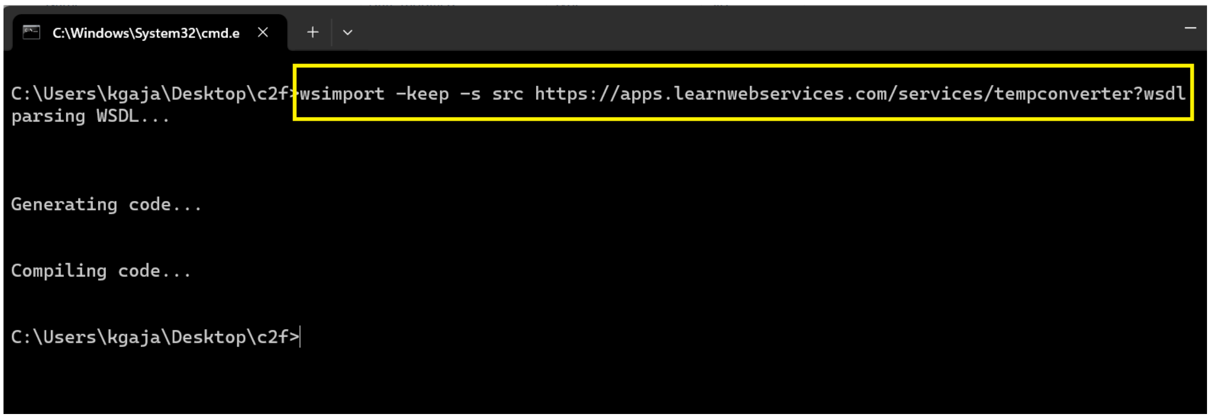
2. Now if you visit the URL you can see the WSDL in XML format, it tells us what input is the webservice expecting and of what datatype. Additionally, it also tells us the output type as highlighted in the image below:

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

3. Now execute the “wsimport” command as shown in the below image passing the above URL of webservice as an

input parameter. Also create a new folder named src in the directory you run this command. This will generate all the java files relevant to our webservice containing all the supported operations in "src" folder.



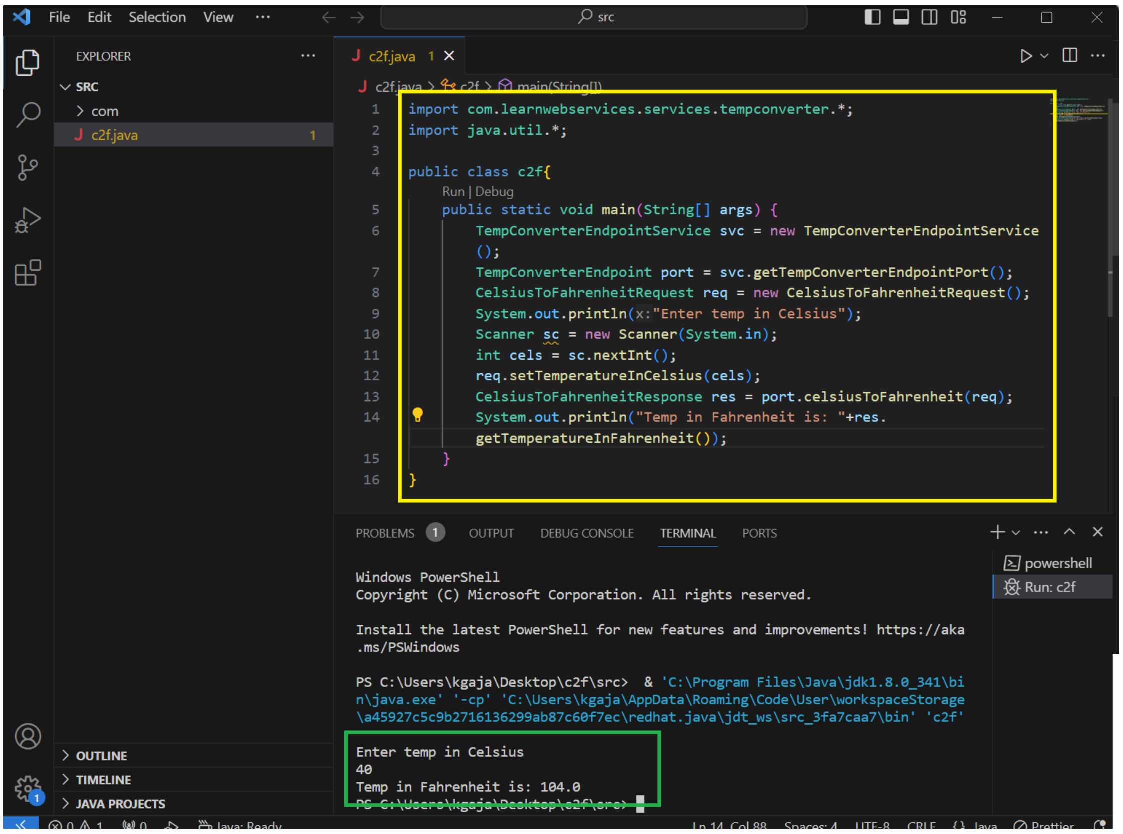
```
C:\Users\kgaja\Desktop\c2f>wsimport -keep -s src https://apps.learnwebservices.com/services/tempconverter?wsdl
parsing WSDL...


Generating code...


Compiling code...


C:\Users\kgaja\Desktop\c2f>
```

4. Now go into the "src" directory and create you Java project. The code will be as follows:



```java
import com.learnwebservices.services.tempconverter.*;
import java.util.*;

public class c2f{
    public static void main(String[] args) {
        TempConverterEndpointService svc = new TempConverterEndpointService();
        TempConverterEndpoint port = svc.getTempConverterEndpointPort();
        CelsiusToFahrenheitRequest req = new CelsiusToFahrenheitRequest();
        System.out.println(x:"Enter temp in Celsius");
        Scanner sc = new Scanner(System.in);
        int cels = sc.nextInt();
        req.setTemperatureInCelsius(cels);
        CelsiusToFahrenheitResponse res = port.celsiusToFahrenheit(req);
        System.out.println("Temp in Fahrenheit is: "+res.getTemperatureInFahrenheit());
    }
}
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka
.ms/PSWindows

PS C:\Users\kgaja\Desktop\c2f\src>  & 'C:\Program Files\Java\jdk1.8.0_341\bi
n\java.exe' '-cp' 'C:\Users\kgaja\AppData\Roaming\Code\User\workspaceStorage
\a45927c5c9b2716136299ab87c60f7ec\redhat.java\jdt_ws\src_3fa7caa7\bin' 'c2f'

Enter temp in Celsius
40
Temp in Fahrenheit is: 104.0
PS C:\Users\kgaja\Desktop\c2f\src>
```