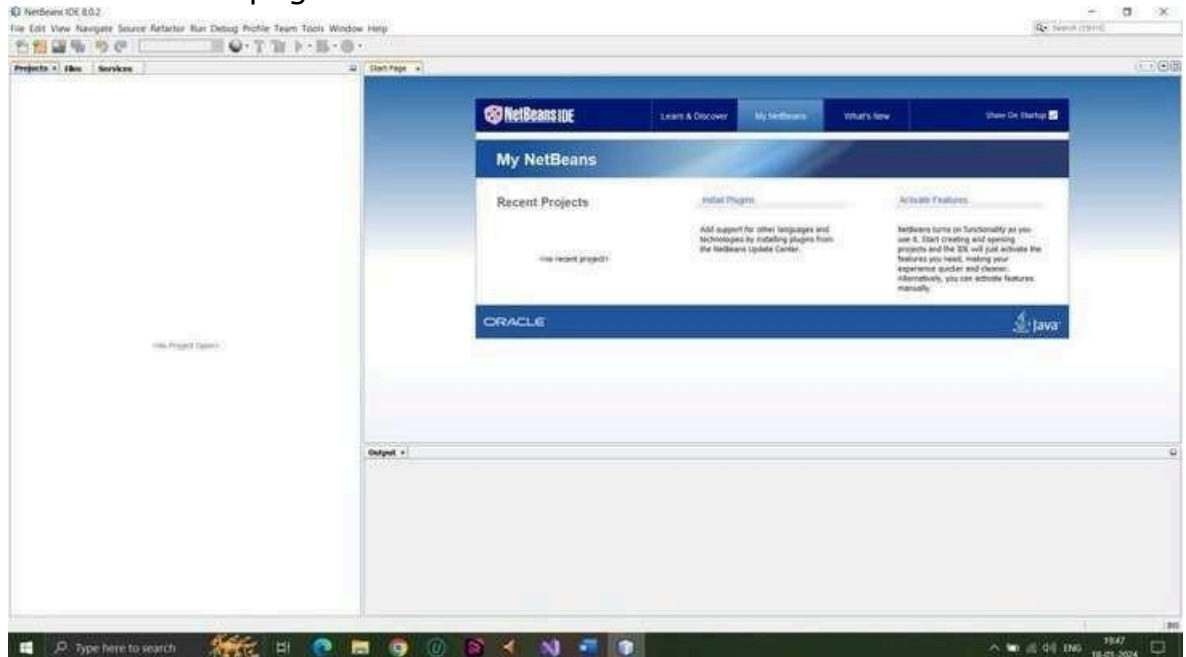


Practical 2

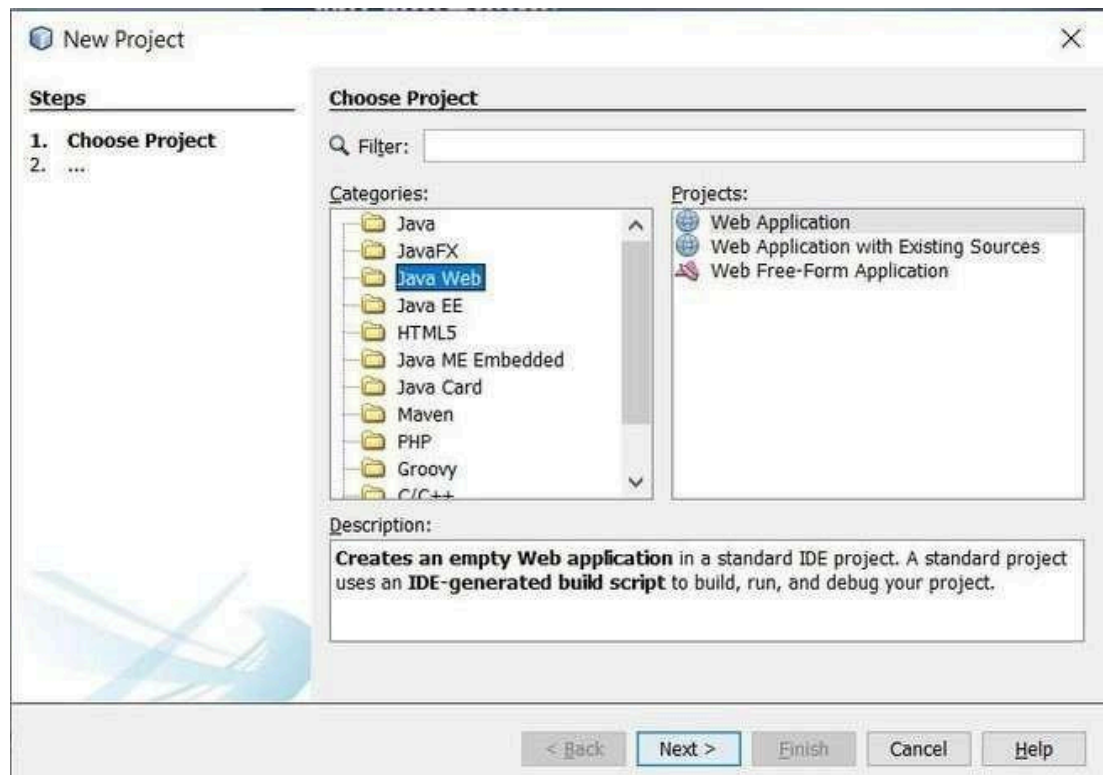
Aim: Create a Simple SOAP service for consuming java web services in Python.

Steps:

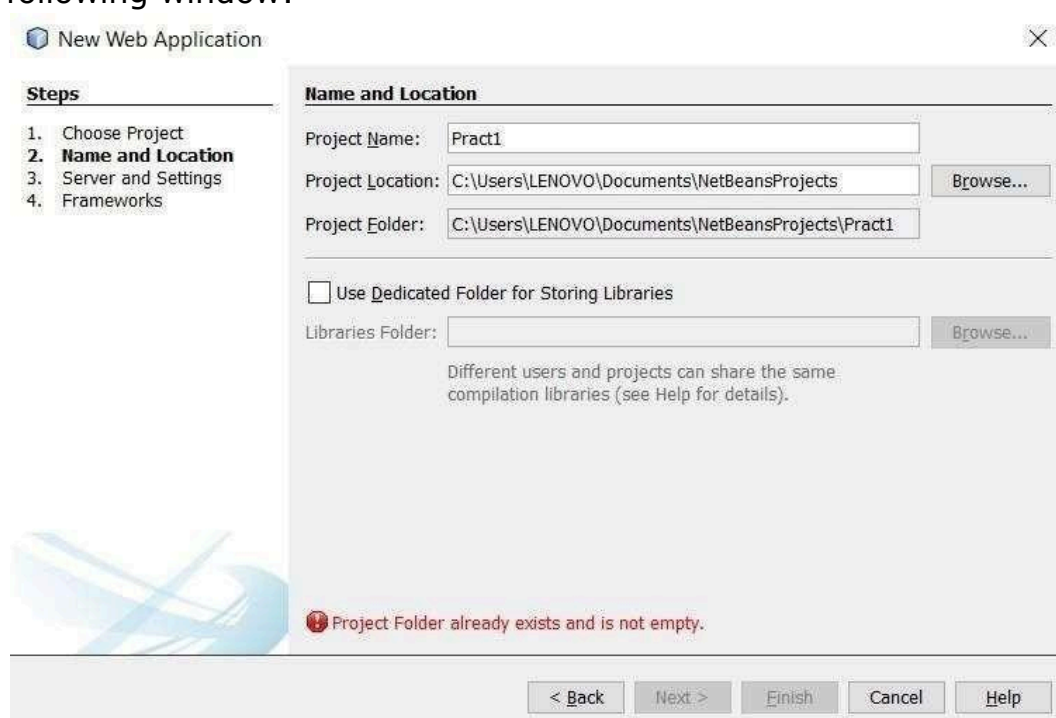
1. Open the NetBeans , and you will get the following screen.
Close the start page.



2. Now click on the file tab and click on new project you will get the following screen :



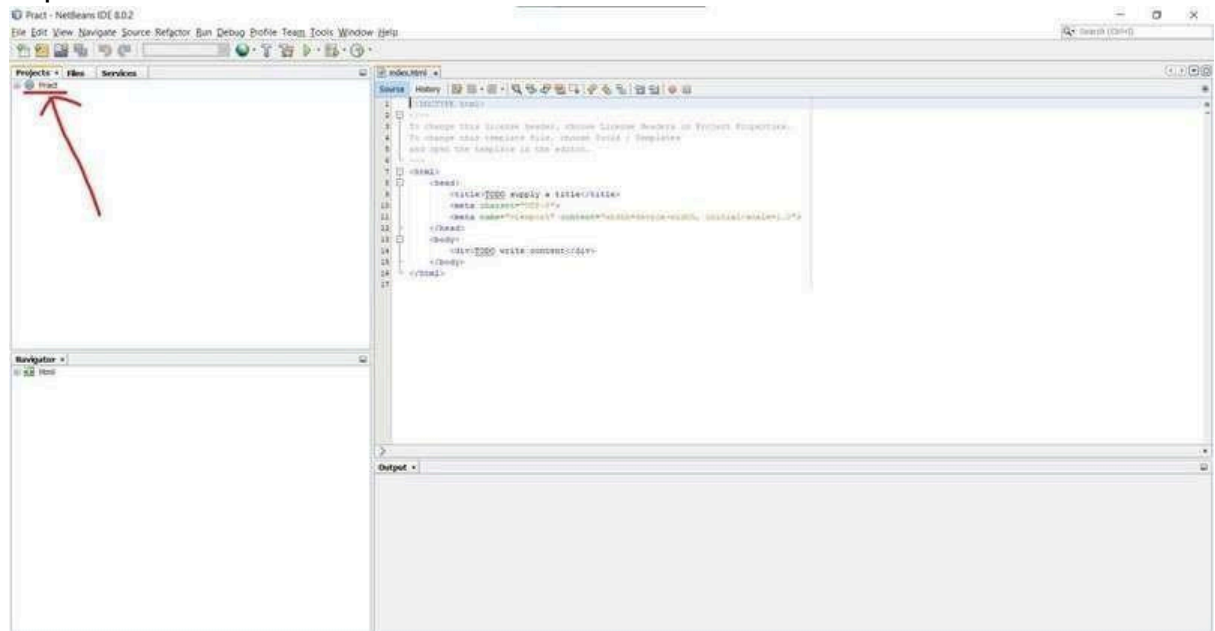
3. In Categories select Java Web and in Projects , Select Web Application .After selecting click on next You will get the following window:



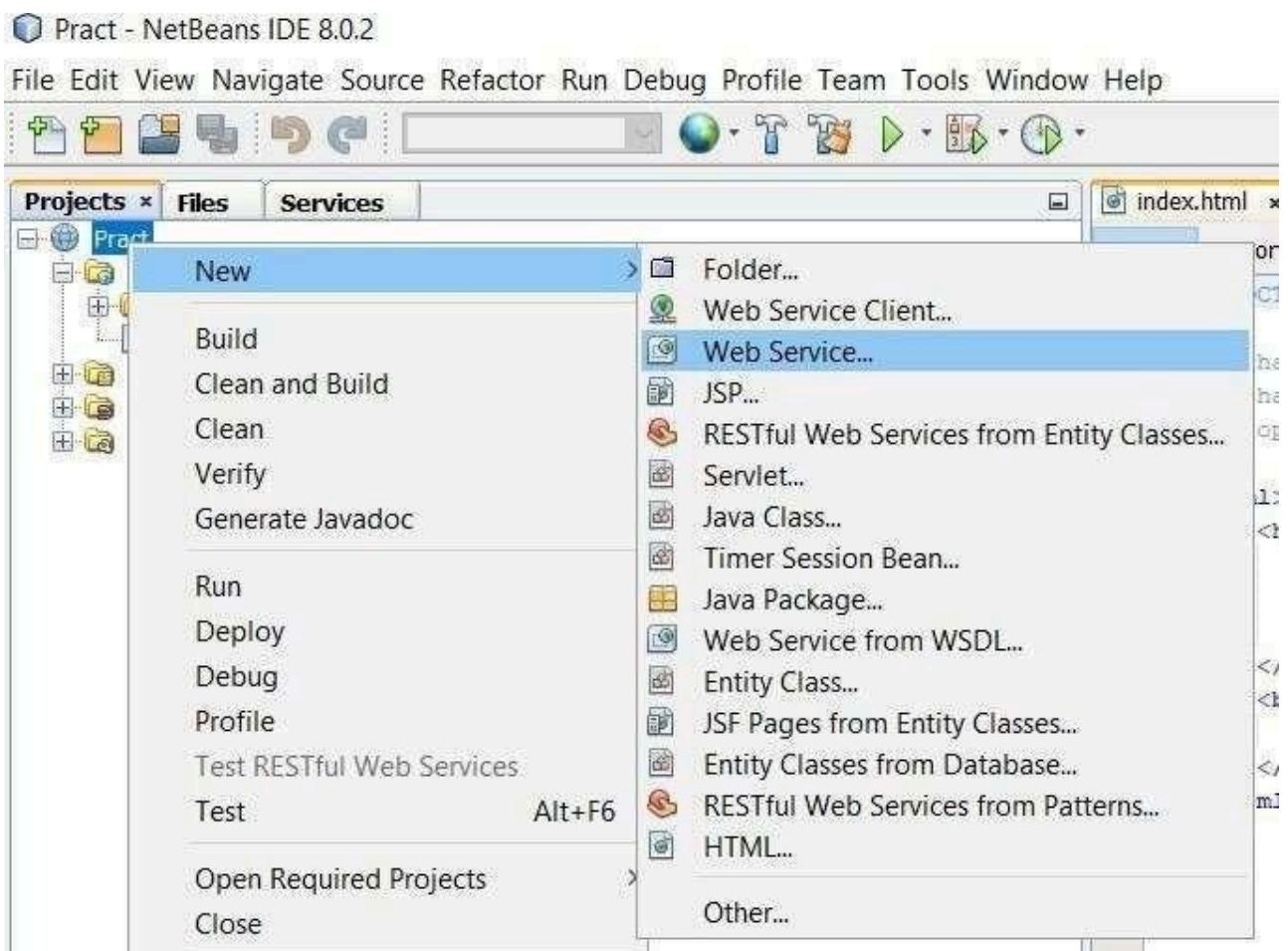
4. Now Give name to the Project Name: , and click on next. You will get the following window then again click on finish



5. You will get the following screen now carefully see in projects section your recently created project appears double click to expand it:



6. Now Right click on the project and select new and then select Web Service ,As shown Below



7. After clicking Web Service the following window should appear , now give name to web service and give package as "server". As shown in the image

New Web Service

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Web Service Name: Currency

Project: Pract

Location: Source Packages

Package: server

☒ Create Web Service from Scratch

☐ Create Web Service from Existing Session Bean

Enterprise Bean: Browse...

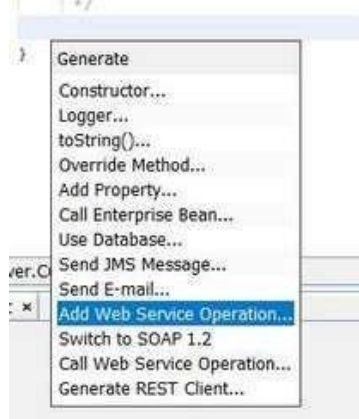
☐ Implement Web Service as Stateless Session Bean

< Back Next > Finish Cancel Help

After clicking finish you should get the following window erase the mentioned code

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package server;
7
8  import javax.jws.WebService;
9  import javax.jws.WebMethod;
10 import javax.jws.WebParam;
11
12 /**
13 *
14 * @author LENOVO
15 */
16 @WebService(serviceName = "Currency")
17 public class Currency {
18
19     /**
20     * This is a sample web service operation
21     */
22     @WebMethod(operationName = "hello")
23     public String hello(@WebParam(name = "name") String txt) {
24         return "Hello " + txt + " :";
25     }
26 }
27
```

8. Now right click anywhere and click on insert code and select Add Web Service Operation



9. The following window would appear :
Just give name to the method or operation and click on add button to add parameters to the method as here we are converting dollar to rupees we should need only one parameter

A screenshot of the 'Add Operation' dialog box. The dialog has a title bar with a close button. It contains a 'Name' field with the text 'operation' and a 'Return Type' field with the text 'java.lang.String'. There is a 'Browse...' button next to the 'Return Type' field. Below these fields are two tabs: 'Parameters' and 'Exceptions'. The 'Parameters' tab is active, showing a table with three columns: 'Name', 'Type', and 'Final'. To the right of the table are four buttons: 'Add', 'Remove', 'Up', and 'Down'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Name	Type	Final
------	------	-------

10. Give the name to the variable select data type as double and click on ok as shown below:

Dialog box titled "Add Operation" with a close button (X) in the top right corner.

Fields:

- Name: InrtoDollr
- Return Type: java.lang.String (with a "Browse..." button next to it)

Tabs: Parameters (selected), Exceptions

Parameters Table:

Name	Type	Final
a	double	<input type="checkbox"/>

Buttons on the right side of the table:

- Add
- Remove
- Up
- Down

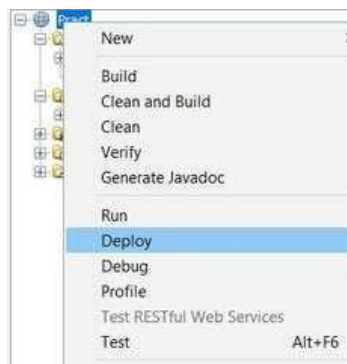
Buttons at the bottom right:

- OK
- Cancel

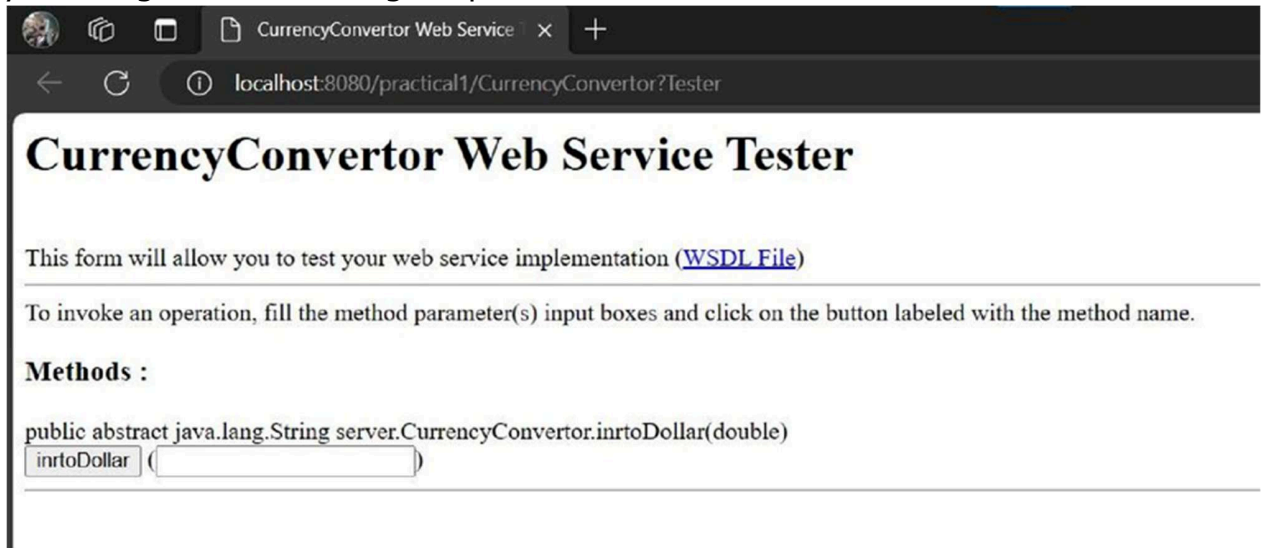
11. After clicking ok code will auto generated , make changes In that code as mentioned below:

```
@WebMethod(operationName = "InrtoDollar")
public String InrtoDollar(@WebParam(name = "a") double a) {
    //TODO write your implementation code here:
    return "The Indian rupees "+a+" in Dollars is "+(a/83.17);
}
```

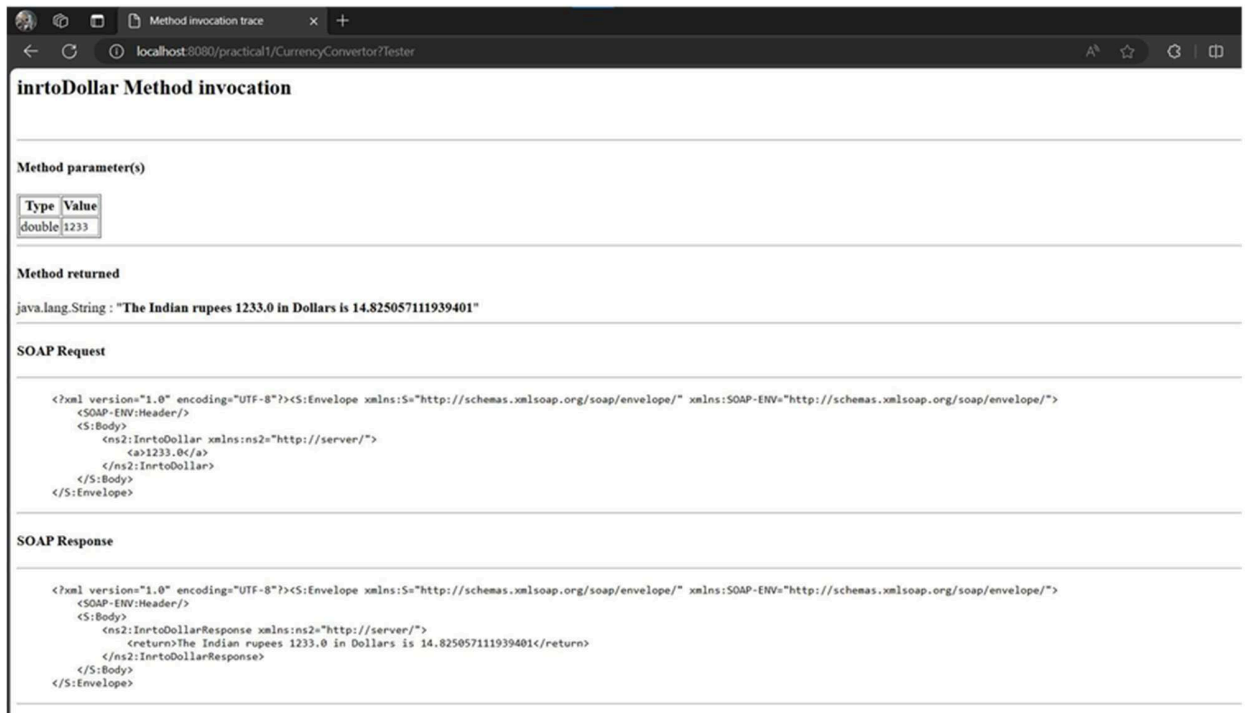
12. Now our web service is ready now right click on project and click on deploy



13. Now right click on web service and click on "test web service" you will get the following output:



The screenshot shows a web browser window with the title "CurrencyConverter Web Service". The address bar shows "localhost:8080/practical1/CurrencyConverter?Tester". The main heading is "CurrencyConverter Web Service Tester". Below the heading, there is a text box with the message "This form will allow you to test your web service implementation ([WSDL File](#))". Below this, there is a text box with the message "To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name." Below this, there is a section titled "Methods :". Below this, there is a text box with the message "public abstract java.lang.String server.CurrencyConverter.inrtoDollar(double)". Below this, there is a text box with the label "inrtoDollar" and a value of "1233".



The screenshot shows a web browser window with the title "Method invocation trace". The address bar shows "localhost:8080/practical1/CurrencyConverter?Tester". The main heading is "inrtoDollar Method invocation". Below the heading, there is a section titled "Method parameter(s)". Below this, there is a table with the following data:

Type	Value
double	1233

Below the table, there is a section titled "Method returned". Below this, there is a text box with the message "java.lang.String : "The Indian rupees 1233.0 in Dollars is 14.825057111939401"". Below this, there is a section titled "SOAP Request". Below this, there is a text box with the following XML code:

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:InrtoDollar xmlns:ns2="http://server/">
      <a>1233.0</a>
    </ns2:InrtoDollar>
  </S:Body>
</S:Envelope>
```

Below the XML code, there is a section titled "SOAP Response". Below this, there is a text box with the following XML code:

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:InrtoDollarResponse xmlns:ns2="http://server/">
      <return>The Indian rupees 1233.0 in Dollars is 14.825057111939401</return>
    </ns2:InrtoDollarResponse>
  </S:Body>
</S:Envelope>
```

So this is how we created our web service and deployed it.

Creating a Python Client

14. After doing all of the steps write the following code in python idle:

Code :

```
from zeep import Client
wsdl_url='http://localhost:8080/pract/Currency? wsdl'
client=Client(wsdl_url)
#call the method on the webservice
d=float(input("Enter Currency in Indian Rupees:"))
result = client.service.InrtoDollar(d)
print(result)
```

replace wsdl_url with the url which was copied at the time of web service client creation .

And replace the InrtoDollar with your method or operation name.and pass parameter to it final code should look like following:

A screenshot of a Python IDE window. The title bar reads '*2.py - C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\2.py (3.11.4)*'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following text:

```
from zeep import Client
wsdl_url='http://localhost:8080/pract/Currency?wsdl'
client=Client(wsdl_url)
#call the method on the webservice
d=float(input("Enter Currency in Indian Rupees:"))
result = client.service.InrtoDollar(d)
print(result)
```

15. After running the above code you should get the following output:

```
= RESTART: C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\2.py
Enter Currency in Indian Rupees:12000
The Indian rupees 12000.0 in Dollars is 144.28279427678248
|
```