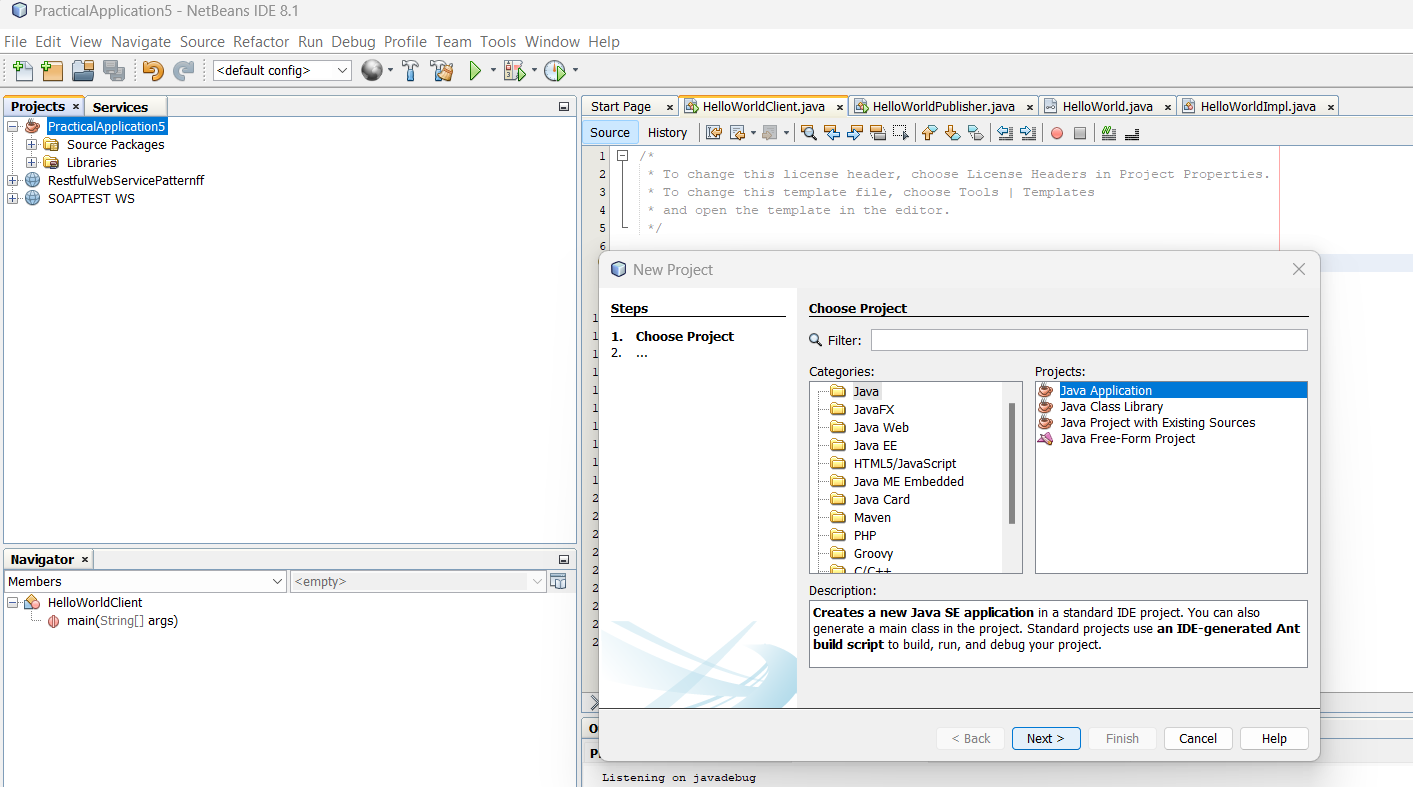
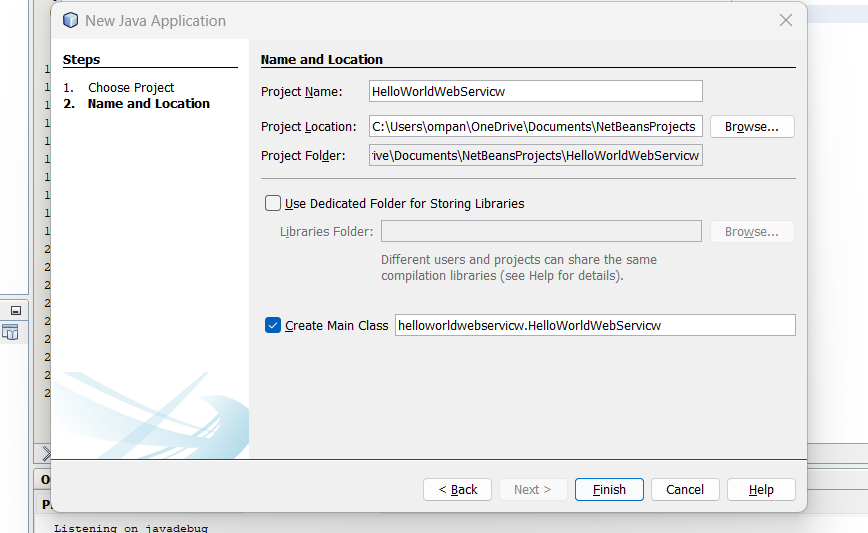
**Step 1: Open NetBeans and Create a New Project**

1. Go to File > New Project...
2. In the "New Project" dialog, select "Java" from the categories and "Java Application" as the project type.
3. Click "Next."

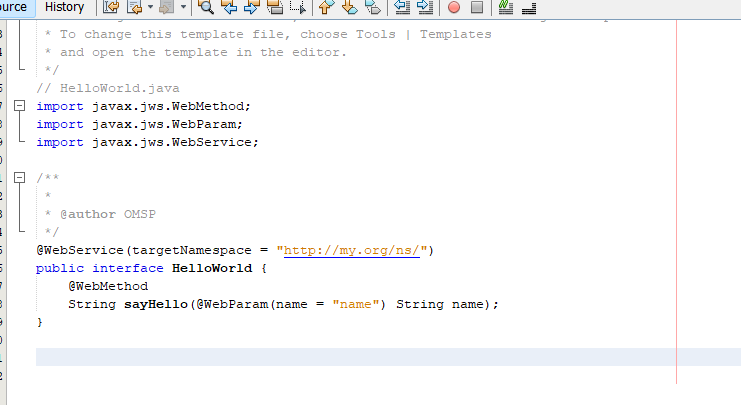


1. Enter a project name, for example, HelloWorldWebService, and specify the project location.
2. Click "Finish."



Step 2: Create HelloWorld Interface

1. Right-click on your project in the "Projects" window.
2. Select New > Java Interface.
3. Enter the interface name as HelloWorld and click "Finish



// HelloWorld.java

import javax.jws.WebMethod;

import javax.jws.WebParam;

import javax.jws.WebService;

@WebService

public interface HelloWorld {

@WebMethod

String sayHello(@WebParam(name = "name") String name);

}

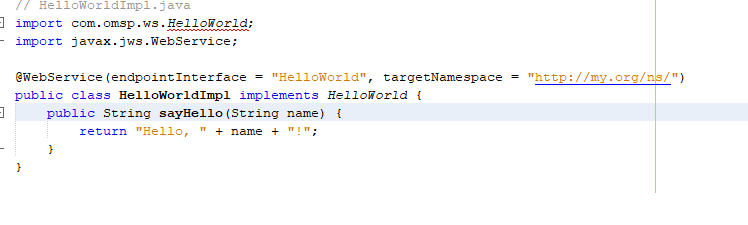
Step 3: Create HelloWorldImpl Class

Right-click on your project.

Select New > Java Class.

Enter the class name as HelloWorldImpl and click "Finish."

Replace the content of the HelloWorldImpl.java file with the content from the previous response.



// HelloWorldImpl.java

import javax.jws.WebService;

@WebService(endpointInterface = "HelloWorld")

public class HelloWorldImpl implements HelloWorld {

@Override

public String sayHello(String name) {

return "Hello, " + name + "!";

}

}

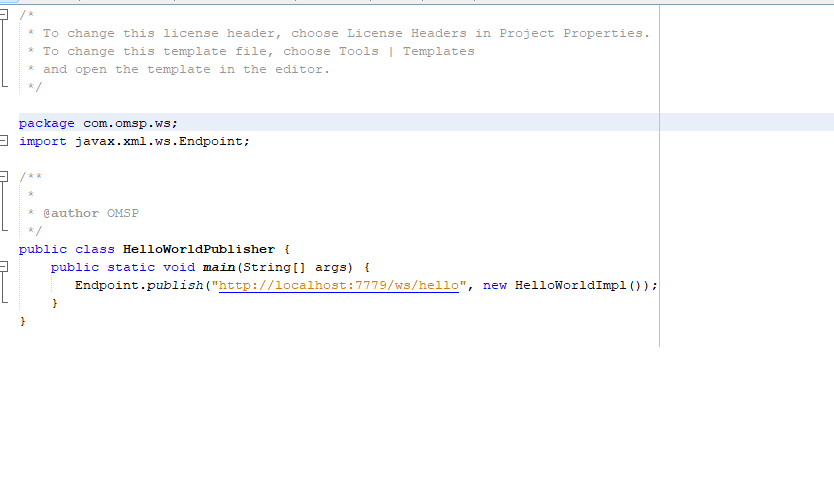
Step 4: Create HelloWorldPublisher Class

Right-click on your project.

Select New > Java Class.

Enter the class name as HelloWorldPublisher and click "Finish."

Replace the content of the HelloWorldPublisher.java file with the content from the previous response.



// HelloWorldPublisher.java

import javax.xml.ws.Endpoint;

public class HelloWorldPublisher {

public static void main(String[] args) {

// Replace "http://localhost:8080/HelloWorld" with your desired endpoint URL

String endpointUrl = "http://localhost:8080/HelloWorld";

Endpoint.publish(endpointUrl, new HelloWorldImpl());

System.out.println("Web service published at: " + endpointUrl);

}

}

Step 5: Compile and Run

Compile all four Java files.

Make sure the HelloWorldPublisher is running.

Run the HelloWorldClient to test the web service.

This example uses JAX-WS for creating a SOAP-based web service and client. The HelloWorld interface defines the web service contract, the HelloWorldImpl class implements the actual logic, the HelloWorldPublisher publishes the web service, and the HelloWorldClient is a simple client to test the service.