

JAVA RPC (Remote Procedure Call)

<https://sivakumar-prasanth.medium.com/java-rpc-remote-procedure-call-99cfaca34c36>

A remote procedure call is an inter-process communication technique that is used for client-server based applications. It is also known as a subroutine call or a function call.

A client has a request message that the RPC translates and sends to the server. This request may be a procedure or a function call to a remote server. When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.

The sequence of events in a remote procedure call are given as follows –

- The client stub is called by the client.
- The client stub makes a system call to send the message to the server and puts the parameters in the message.
- The message is sent from the client to the server by the client's operating system.
- The message is passed to the server stub by the server operating system.
- The parameters are removed from the message by the server stub.
- Then, the server procedure is called by the server stub.

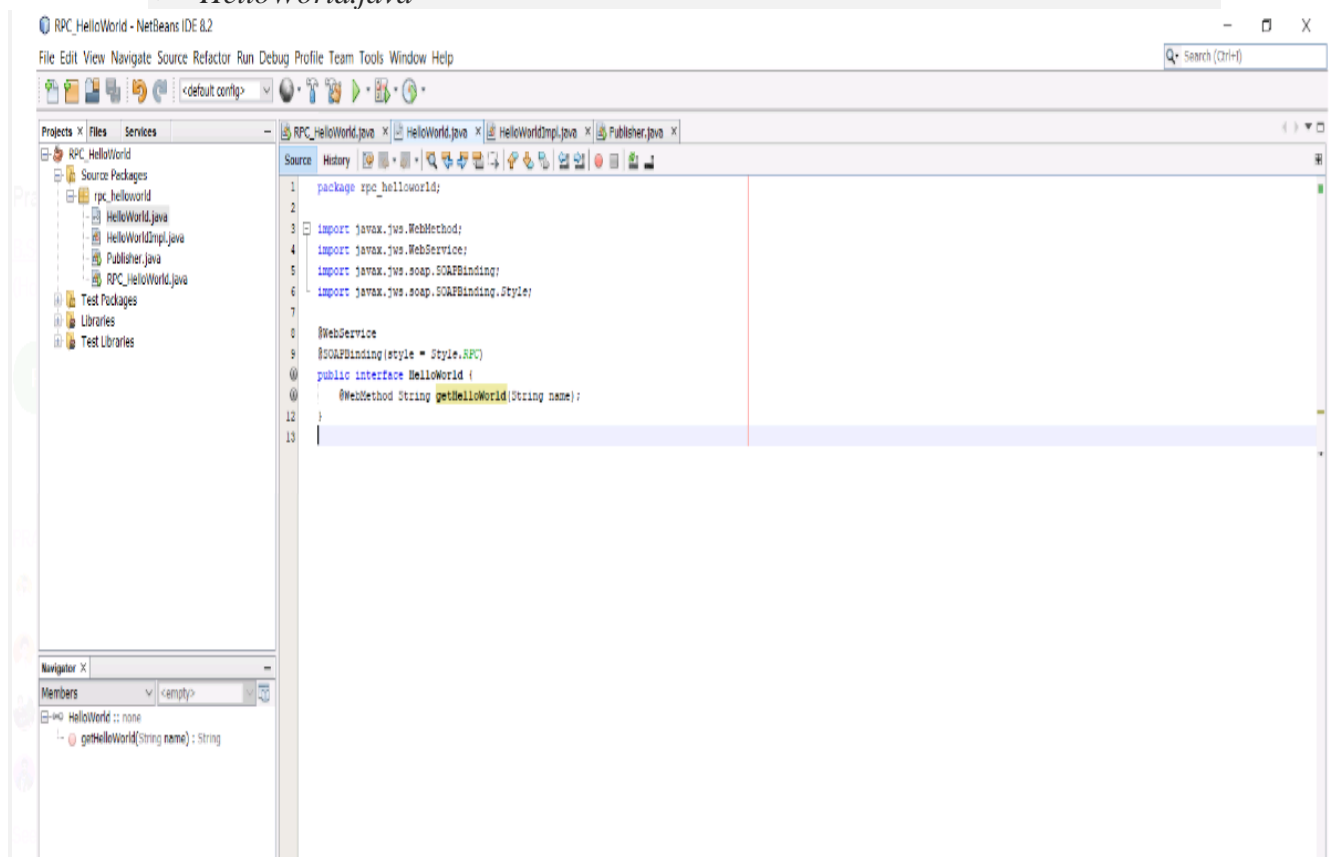
Now let's see how we can implement a simple hello world java RPC program. You can get the following sample project in my GitHub account. visit the following link: https://github.com/prasanthse/RPC_HelloWorld.git

First we want to create 4 files. Three of them are for server side and the other one is for client side.

Luckily it requires no extra configuration settings. So you don't need to load any extra jar file for it. Just import some libraries which already inbuilt in JDK.

Server side:

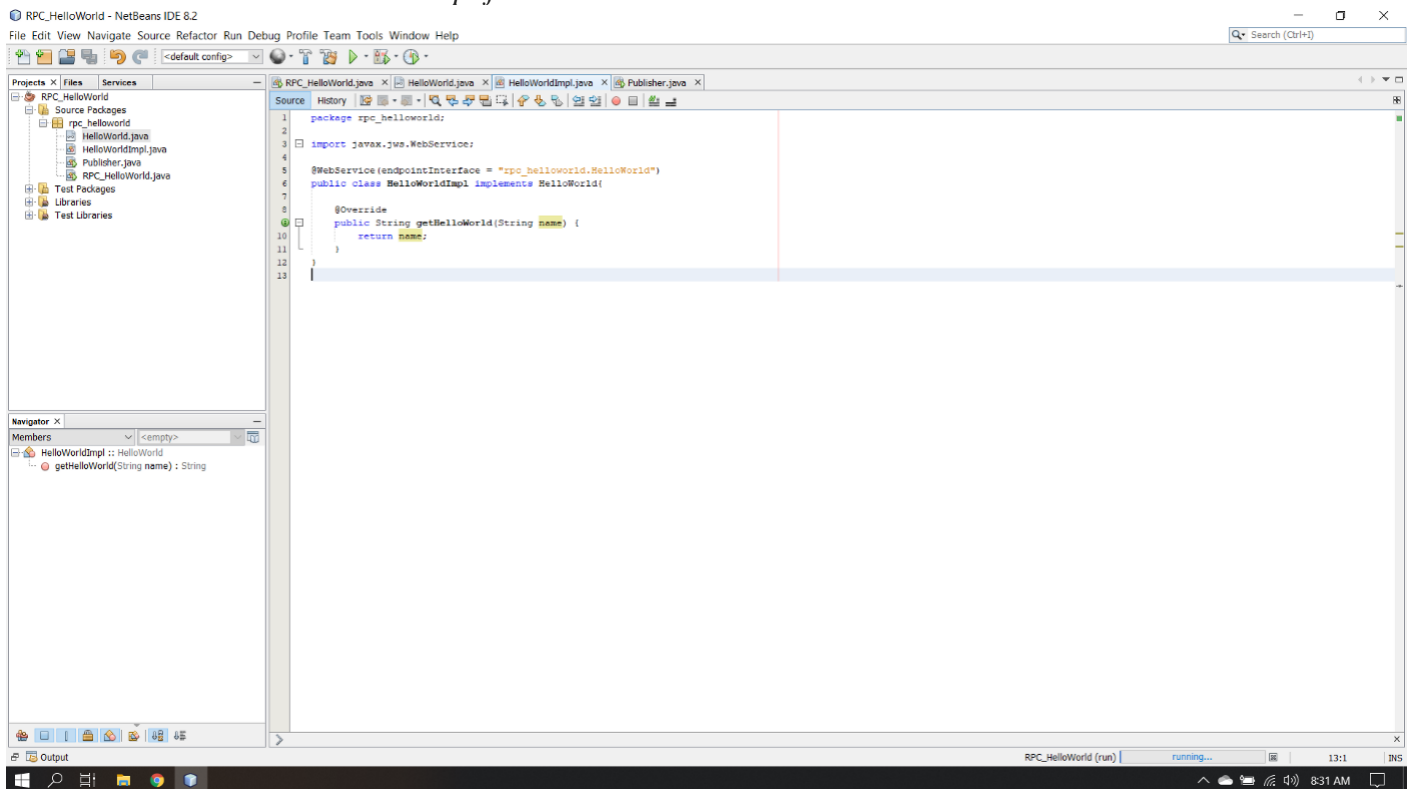
- *HelloWorld.java*



```
import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;
```

This is a Web Service Endpoint Interface. This interface will contain the declarations of all the methods you want to include in the Web Service.

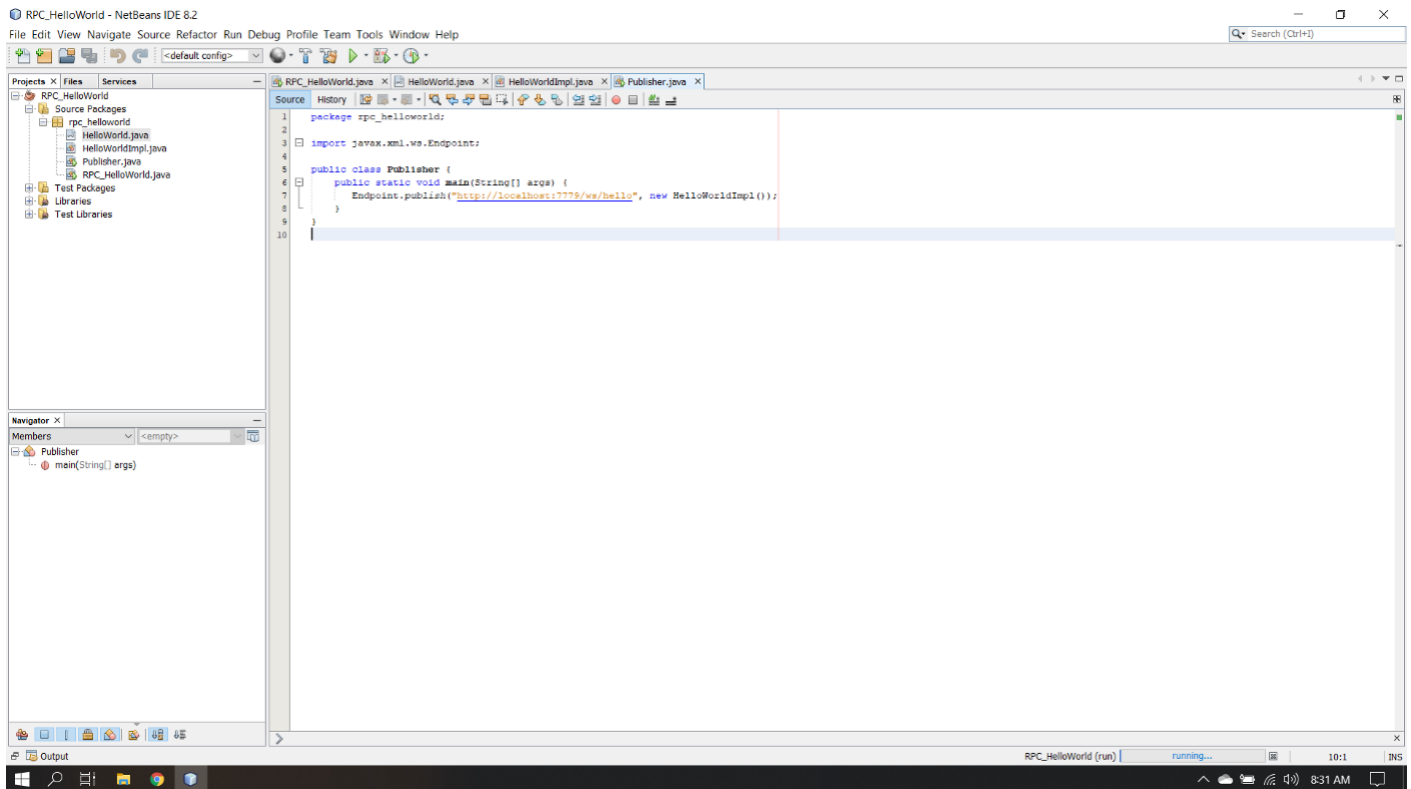
- *HelloWorldImpl.java*



```
import javax.jws.WebService;
```

Then you have to create a class that actually implements the above interface, which will be your Endpoint implementation.

Publisher.java

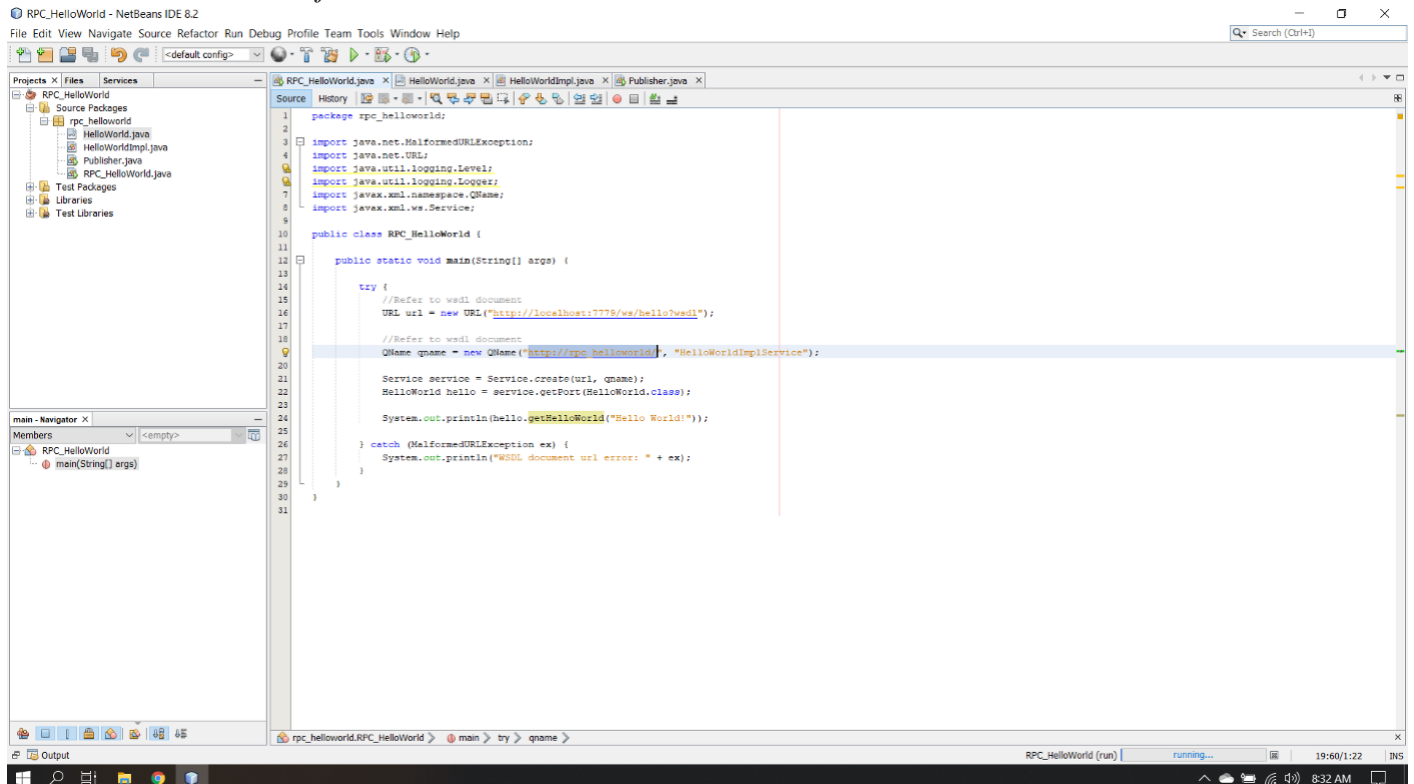


import javax.xml.ws.Endpoint;

Finally you create your Endpoint publisher which actually deploys the web service and creates and publishes the endpoint for the specified implementer object at a given address. The necessary server infrastructure will be created and configured by the JAX-WS implementation. You have to run the publisher to make your Web Service available to clients.

Client Side

HelloWorldClient.java



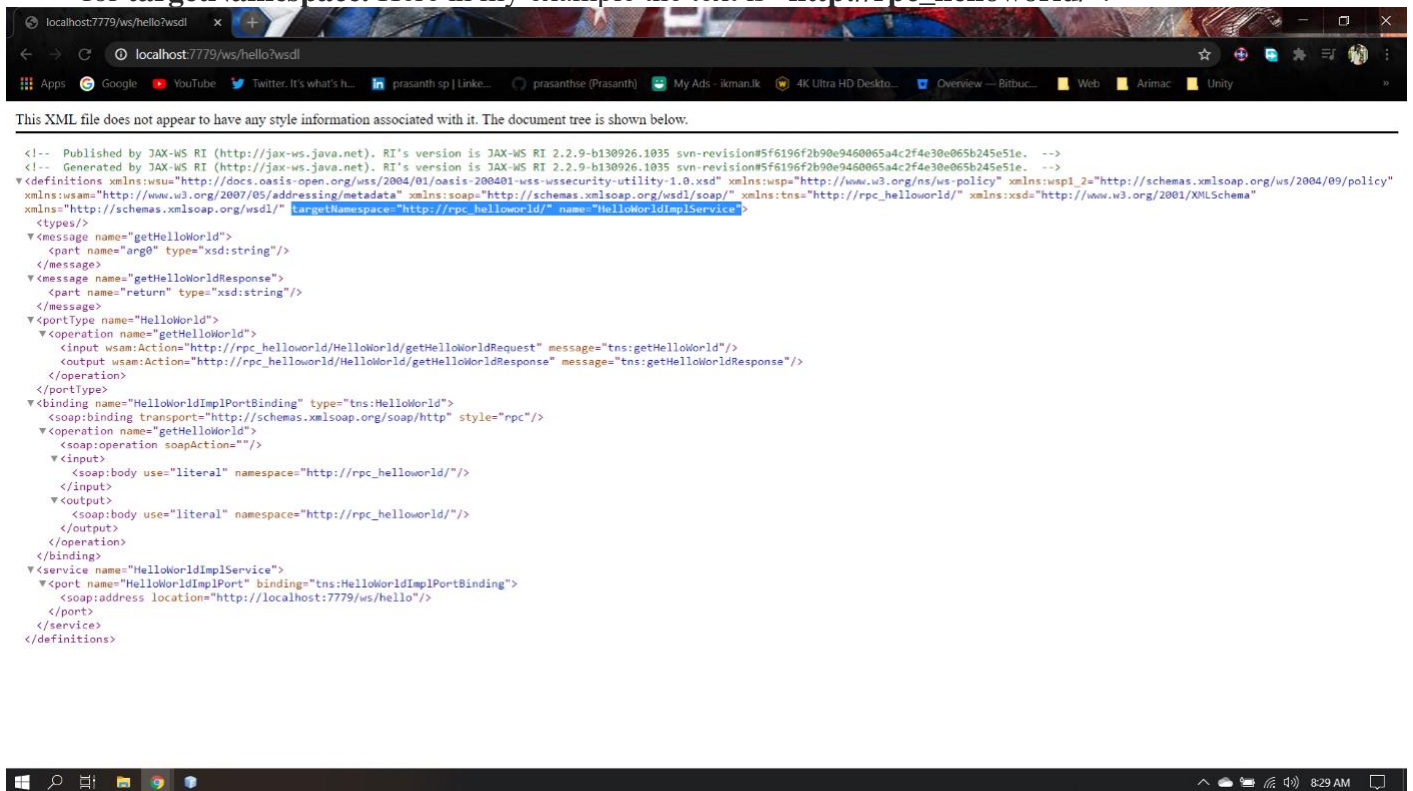
```
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
```

This is client that consumes the above Web Service.

After create all these files, you need to run your Publisher.Java and then go to your browser and type the following:

<http://localhost:7779/ws/hello?wsdl>

Then you will get the response in XML format. After that you need to copy the text that assign for **targetNamespace**. Here in my example the text is “**http://rpc_helloworld/**”.



Then paste the text in your Client side file, as the QName first parameter (The image is shown above). Now run your program and you will get the output.

