Tutorial Creació Serveis Web
(Java, Eclipse Indigo, Tomcat i Axis2)

Arquitectura del Software

Cristina Gómez

Setembre 2012

L'objectiu d'aquest document és definir un servei web de prova i un client Java per a aquest servei web. Per definir i provar el servei web s'utilitzarà l'IDE Eclipse Indigo amb el servidor web Tomcat 7.0 i el motor de serveis Axis2.

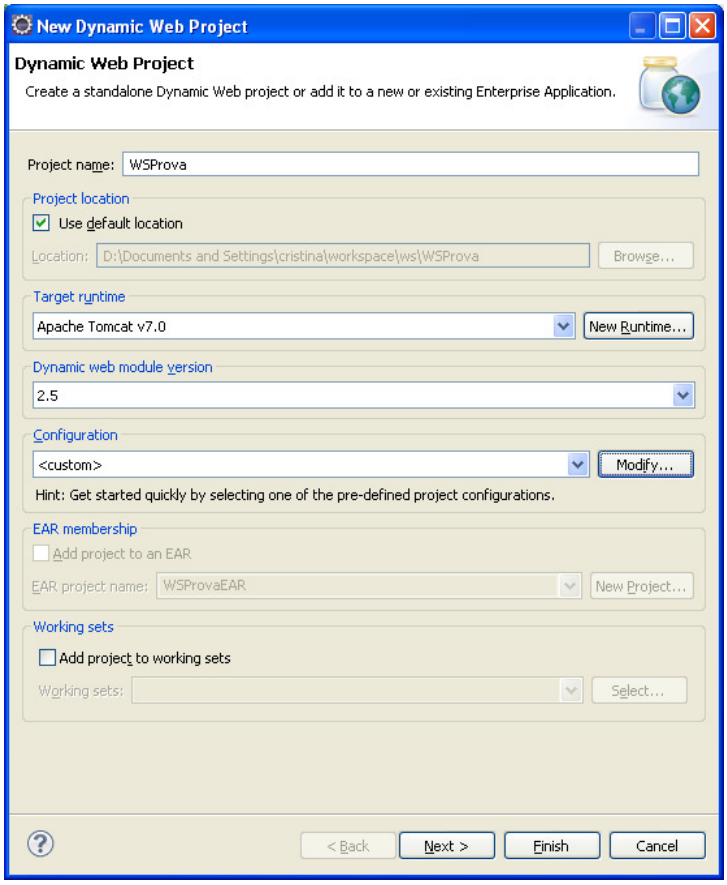
1. Configurar Tomcat 7 i Axis 2 a Eclipse Indigo

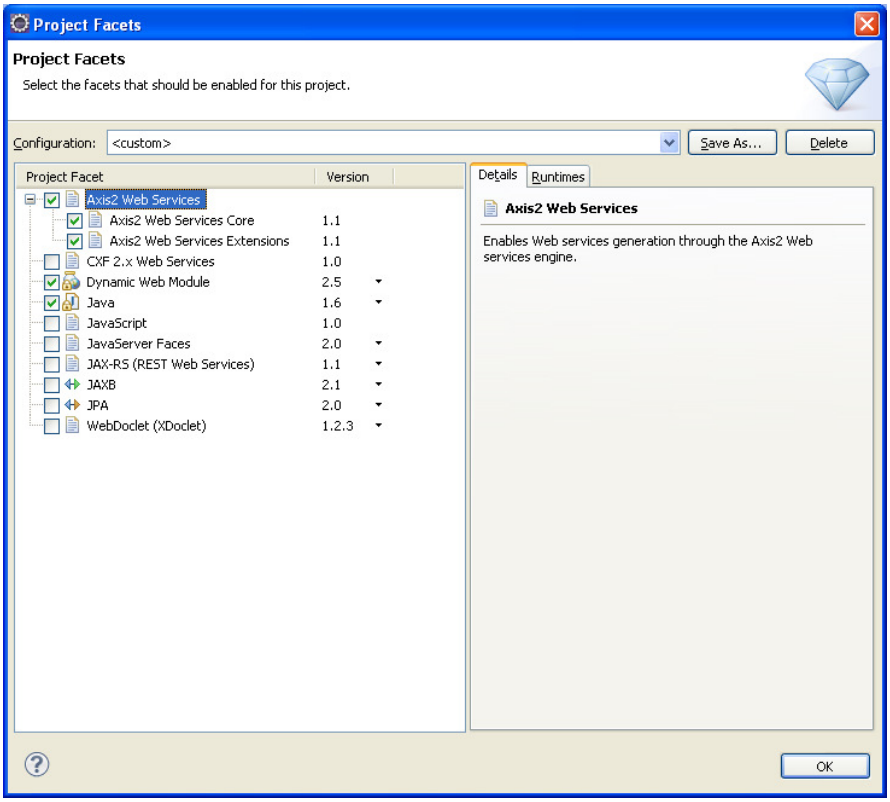
A les següents adreces podeu trobar com configurar Tomcat 7.0 i Axis2 a Eclipse Indigo.
http://efunctions.wordpress.com/2011/12/19/configurando-apache-tomcat-7-en-eclipse-indigo/

http://efunctions.wordpress.com/2011/12/19/configurando-apache-axis2-en-eclipse-indigo/
Després de fer la instal·lació i configuració ja estem preparats per crear un servei web.

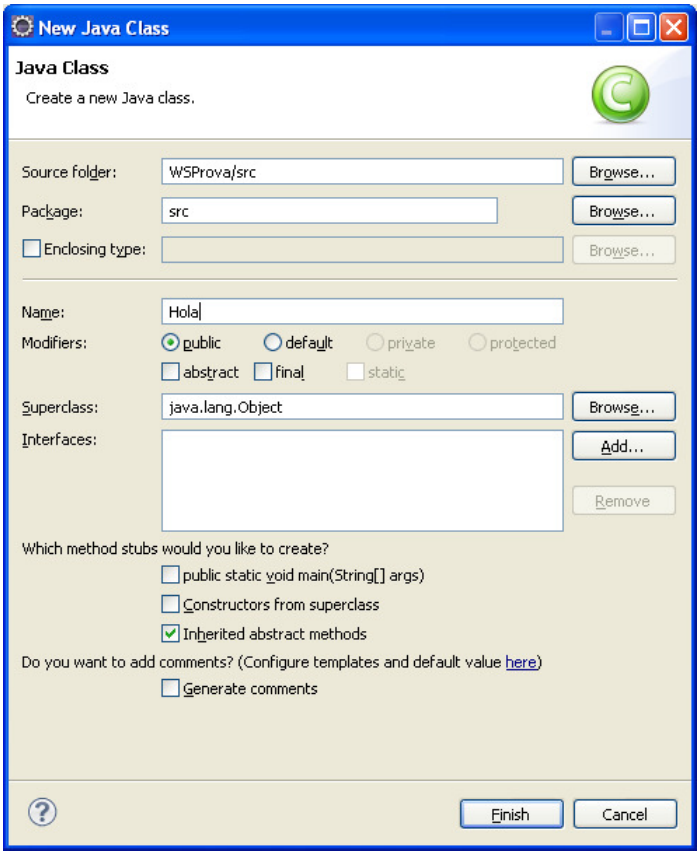
2. Creació Servei Web en Java

2.1. Obrir Eclipse i anar a **"File >> New >> Dinamic Web Project"**. S'obrirà una finestra on haurem d'indicar el nom del projecte (en el nostre cas el nom serà WSProva). Haurem d'especificar el **"Dynamic web module version"** en **2.5**, que funciona amb Axis2, i després a **"Configuration"** al clickar el botó **"Modify…"** s'obrirà una finestra on s'especifiquen les facets (mòduls que utilitzarà el nostre projecte). Per defecte es troben seleccionades les opcions **"Dinamic Web Module"** i **"Java"**. Haurem de seleccionar l'etiqueta que diu **"Axis2 Web Services"** (ens carregarà els mòduls de suport per a Axis2). Finalment farem un click a **"Finish".**





2.2. Sobre el nou projecte creem una nova classe **"Hola.java"** dins del **"Package"** **"src".**

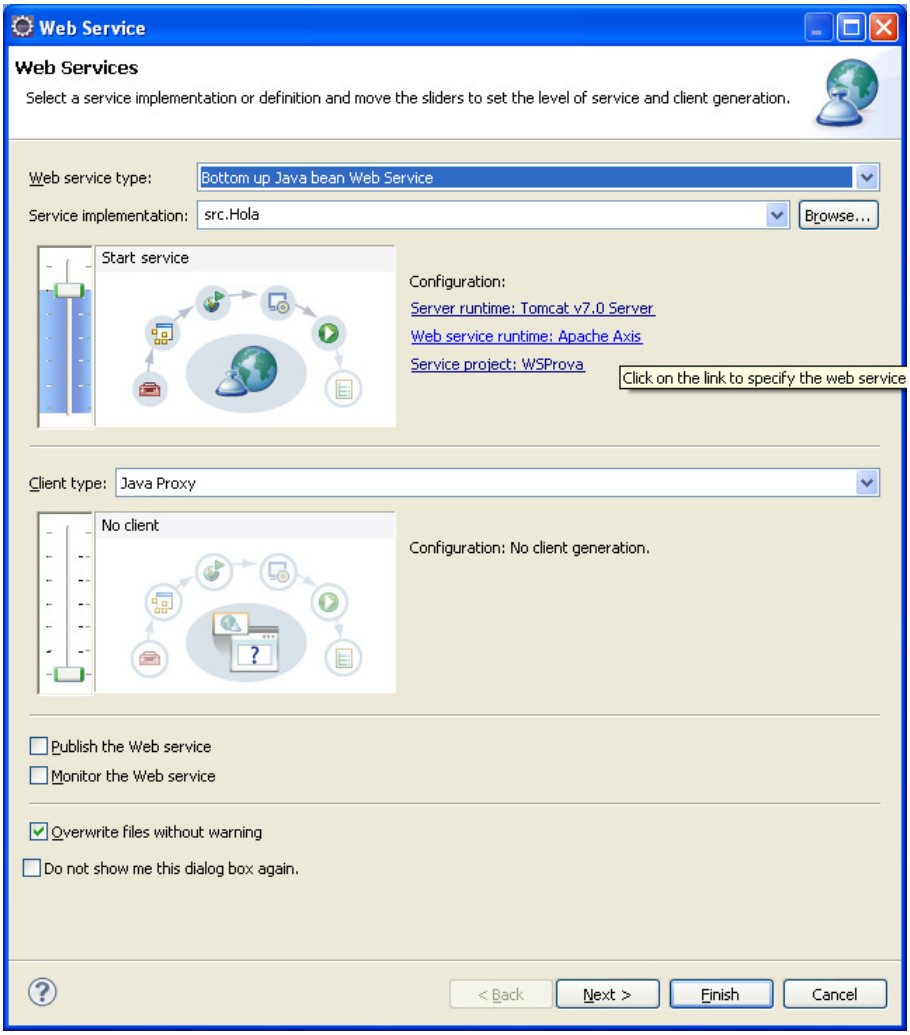


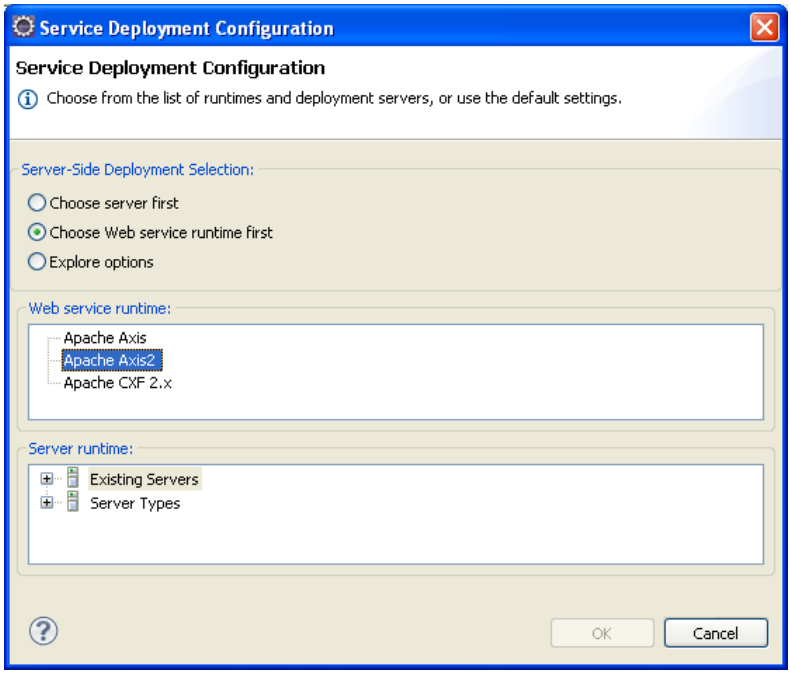
la classe ha de contenir el codi del servei (en el nostre cas serà un codi senzill):

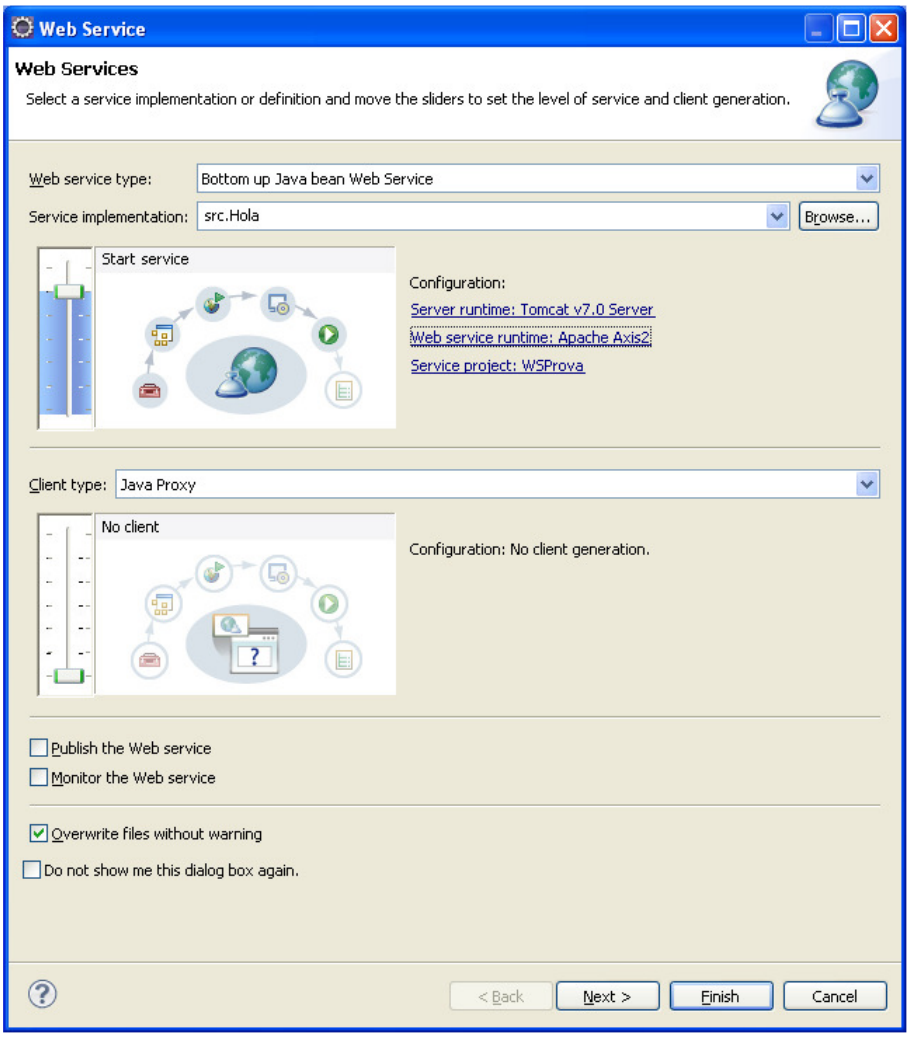
Hola.java

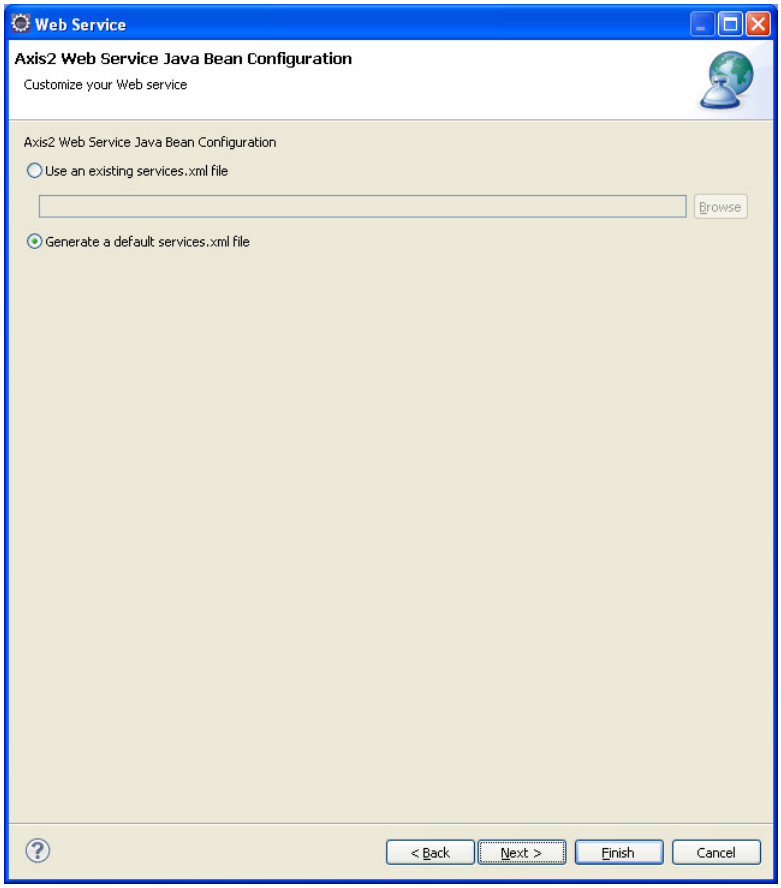
```
package src;

public class Hola {
    public String Saludar(String nom) {
        return("Hola " + nom);
    }
}
```

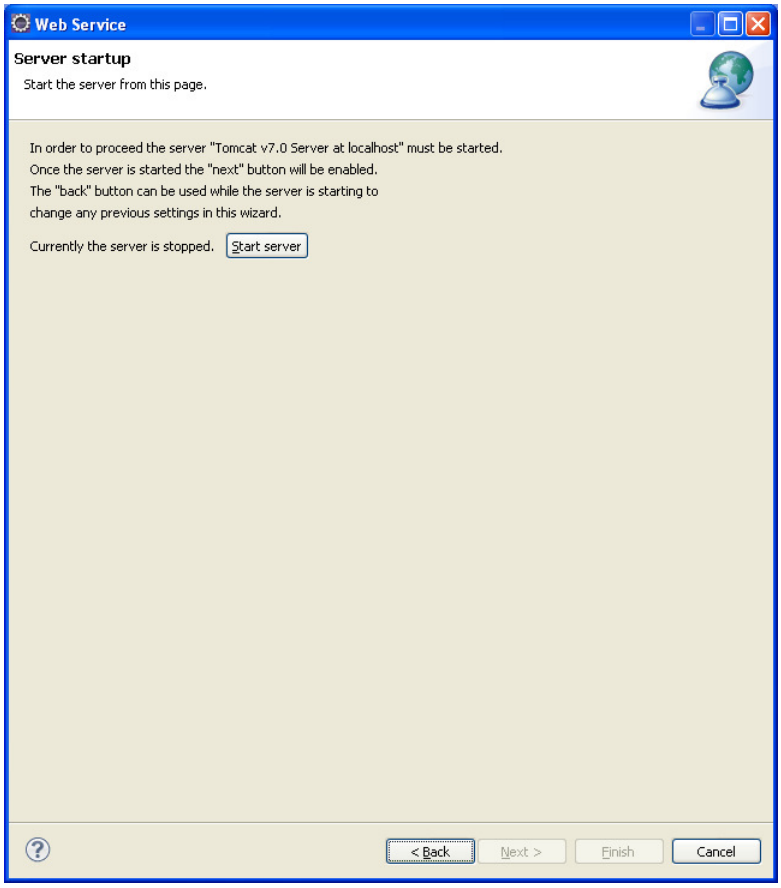
2.3. Sobre la classe *Hola* cliqueu botó dret i després **"Web Service >> Create Web Service"**. S'obrirà una finestra on només haurem de modificar el **"Web service runtime"**. Cliquem sobre l'enllaç i ens obrirà una nova finestra on hem de canviar el **"Web service runtime"**, que per defecte és **"Apache Axis"**, per **"Apache Axis2"**. Després cliquem **"Next"** i deixem les opcions per defecte. Cliquem de nou **"Next"** i el nostre servidor hauria d'estar aturat. Cliquem en **"Start Server"** i finalitzem (**"Finish"**).




Service Deployment Configuration
Service Deployment Configuration
Choose from the list of runtimes and deployment servers, or use the default settings.
Server-Side Deployment Selection:
Choose server first
Choose Web service runtime first
Explore options
Web service runtime:
Apache Axis
Apache Axis2
Apache CXF 2.x
Server runtime:
Existing Servers
Server Types
OK
Cancel


Web Service
Web Services
Select a service implementation or definition and move the sliders to set the level of service and client generation.
Web service type:
Bottom up Java bean Web Service
Service implementation:
src.Hola
Browse...
Start service
Configuration:
Server runtime: Tomcat v7.0 Server
Web service runtime: Apache Axis2
Service project: WSProva
Client type:
Java Proxy
No client
Configuration: No client generation.
Publish the Web service
Monitor the Web service
Overwrite files without warning
Do not show me this dialog box again.
< Back
Next >
Finish
Cancel


Web Service
Axis2 Web Service Java Bean Configuration
Customize your Web service
Axis2 Web Service Java Bean Configuration
Use an existing services.xml file
Browse
Generate a default services.xml file
< Back
Next >
Finish
Cancel

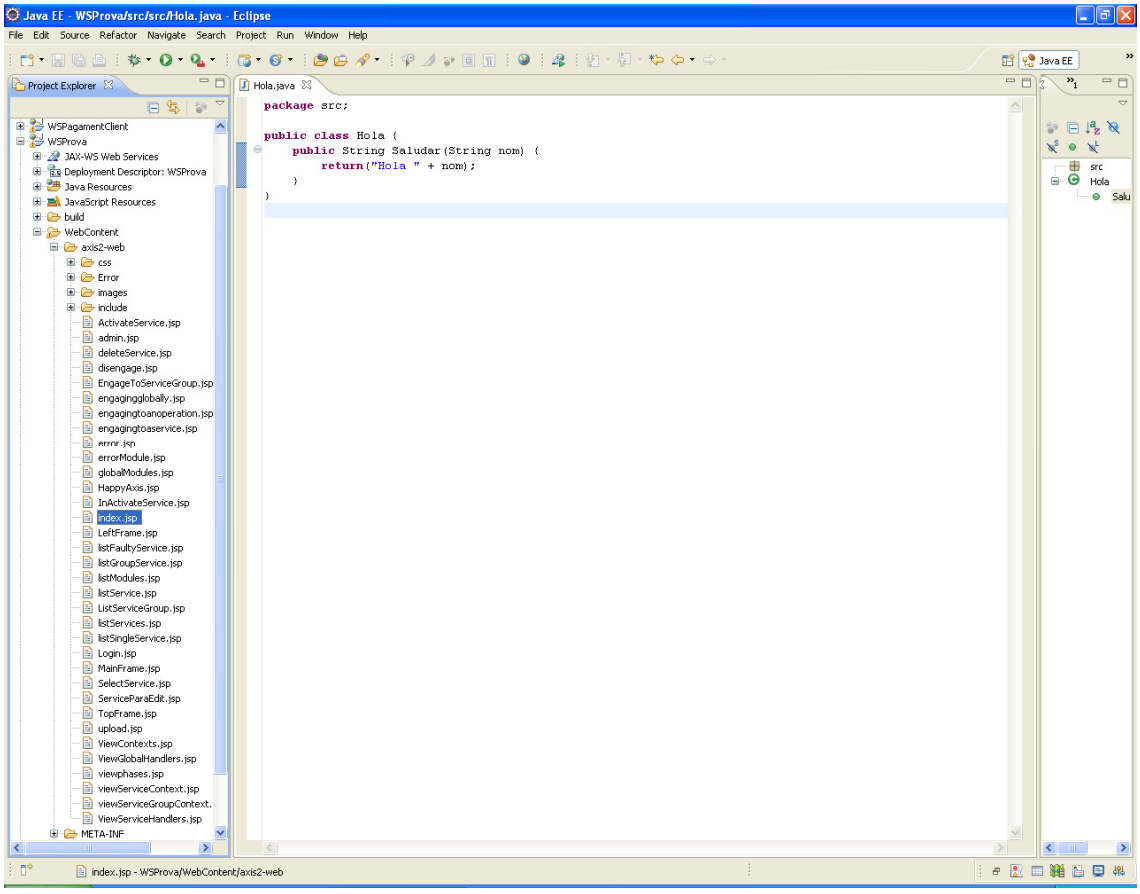

Web Service
Server startup
Start the server from this page.
In order to proceed the server "Tomcat v7.0 Server at localhost" must be started.
Once the server is started the "next" button will be enabled.
The "back" button can be used while the server is starting to
change any previous settings in this wizard.
Currently the server is stopped.
Start server
< Back
Next >
Finish
Cancel

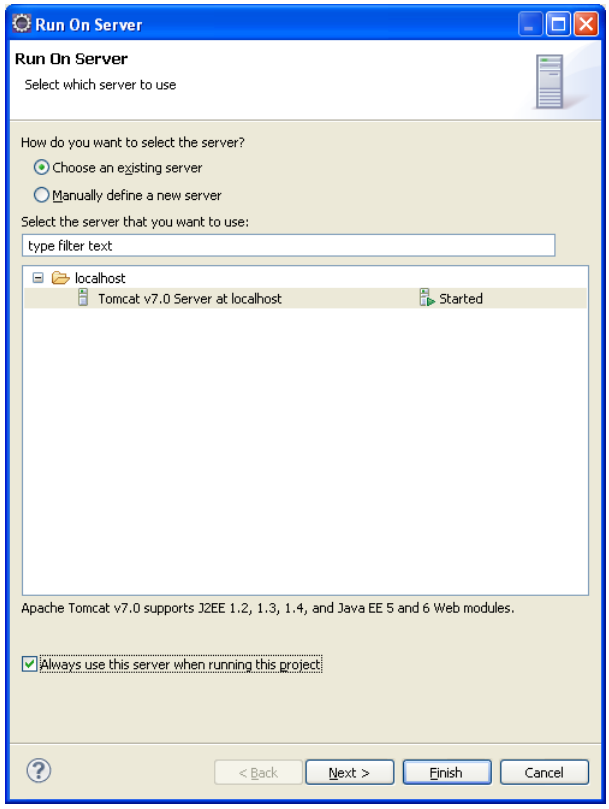
Ja hem creat el nostre servei. Ara només queda definir un client que invoqui al servei per poder provar-lo.

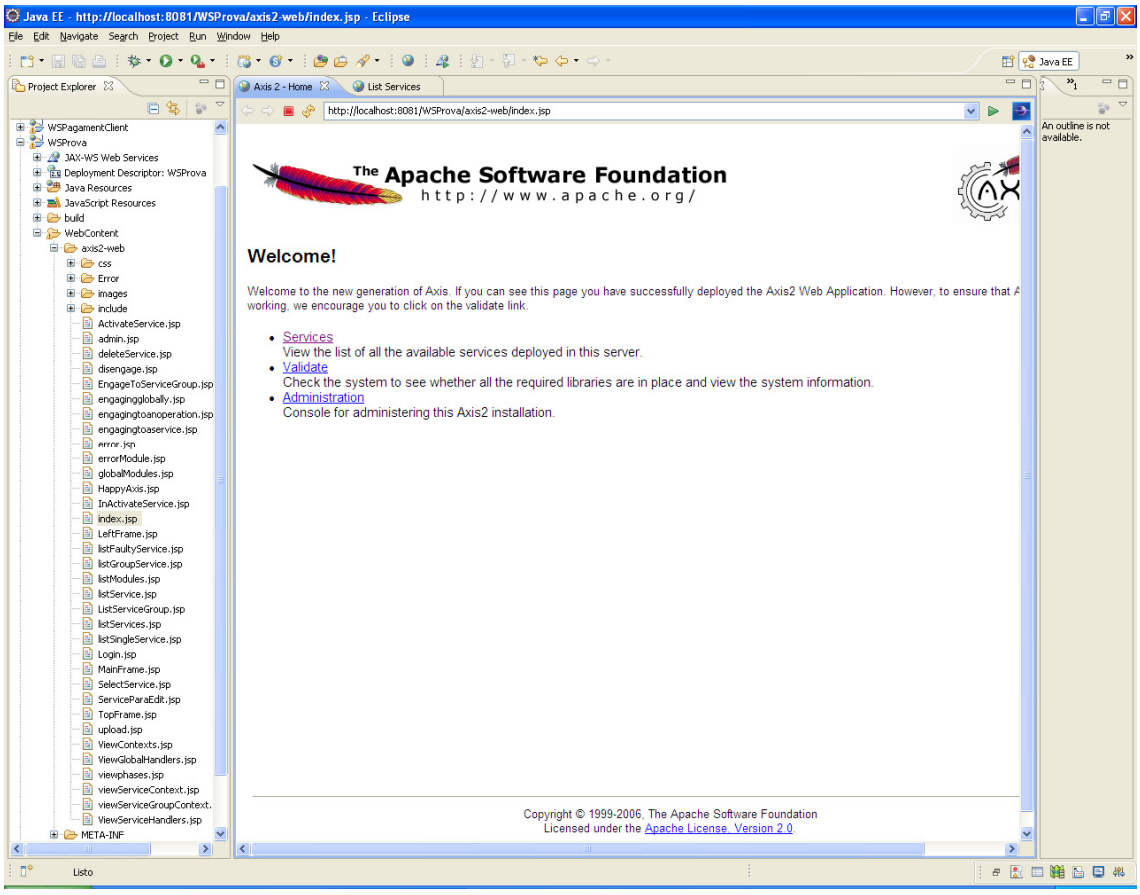
3. Creació Client d'un Servei Web

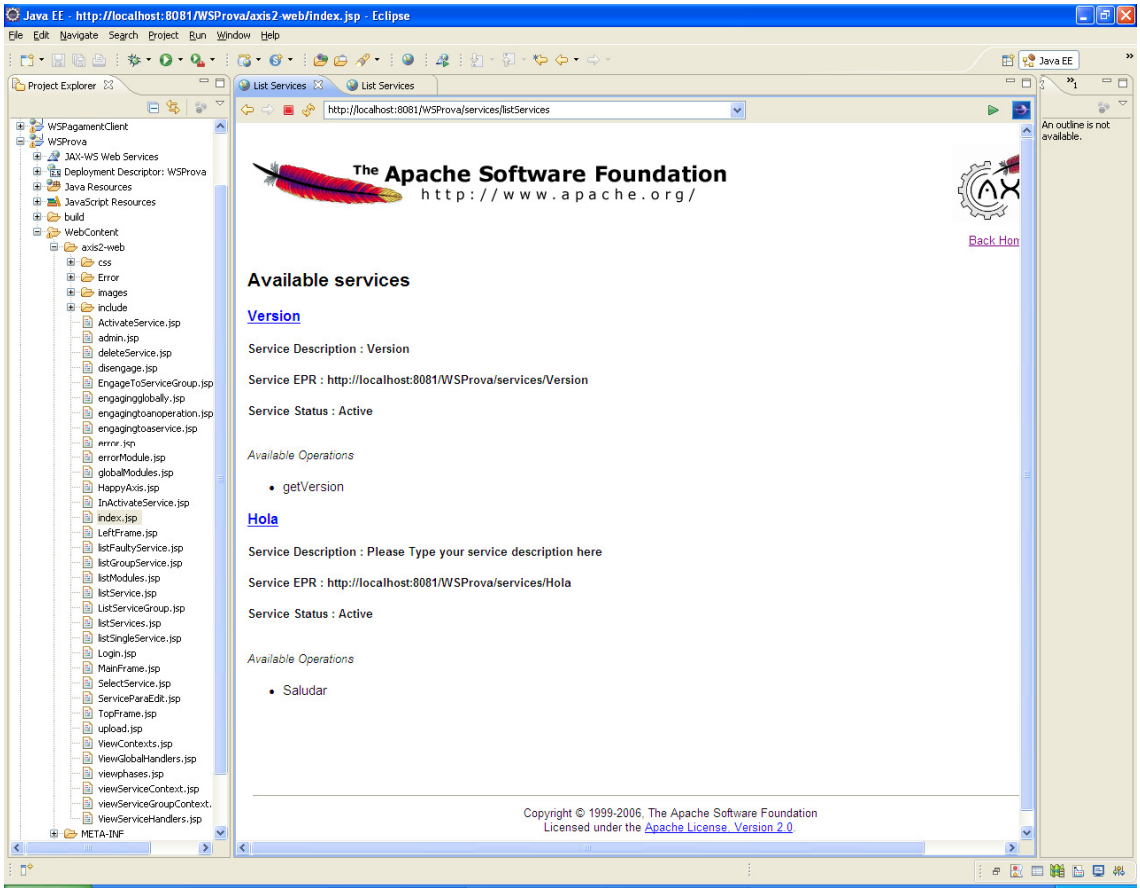
Per poder invocar a un servei web necessitem la descripció del servei en WSDL (Web Service Description Language). Aquesta descripció es fa en un document XML que descriu la mecànica per interaccionar amb el servei però amagant els detalls tècnics com, per exemple, el llenguatge de programació utilitzat per definir el servei. Afortunadament el motor de serveis Axis2 genera automàticament aquest fitxer i per tant no caldrà que el definim manualment. No obstant, hem de saber el nom del fitxer per poder crear el client.

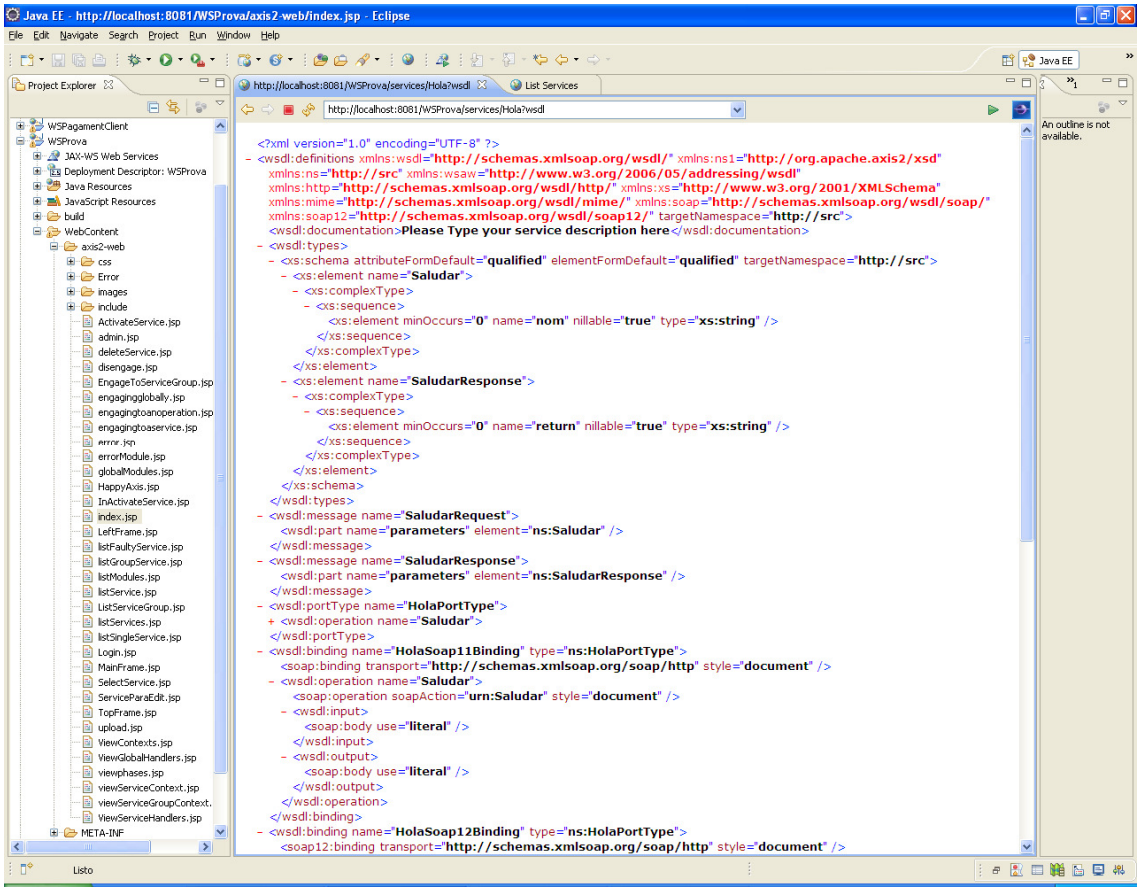
3.1. Sobre el fitxer index.jsp que podreu trobar a **"WSProva >> WebContent >> axis2-web >> include >> index.jsp"** cliqueu el botó dret i seleccioneu **"Run As >> Run On Server"**. S'obrirà la finestra Run On Server. Seleccionarem *Always use this server when ...* i després de clicar **"Next",** s'obrirà la pàgina d'Axis2. En aquesta pàgina seleccionarem "**Services"** i ens apareixerà el nostre servei *Hola* amb l'operació *Saludar* que hem definit. Si cliquem a **"Next"** obtindrem el fitxer el fitxer on es troba la descripció en WSDL del nostre servei. Hem de copiar l'adreça d'aquest fitxer ja que la necessitarem més endavant per crear el client del nostre servei (en el nostre exemple aquesta adreça és http://localhost:8081/WSProva/services/Hola?wsdl).



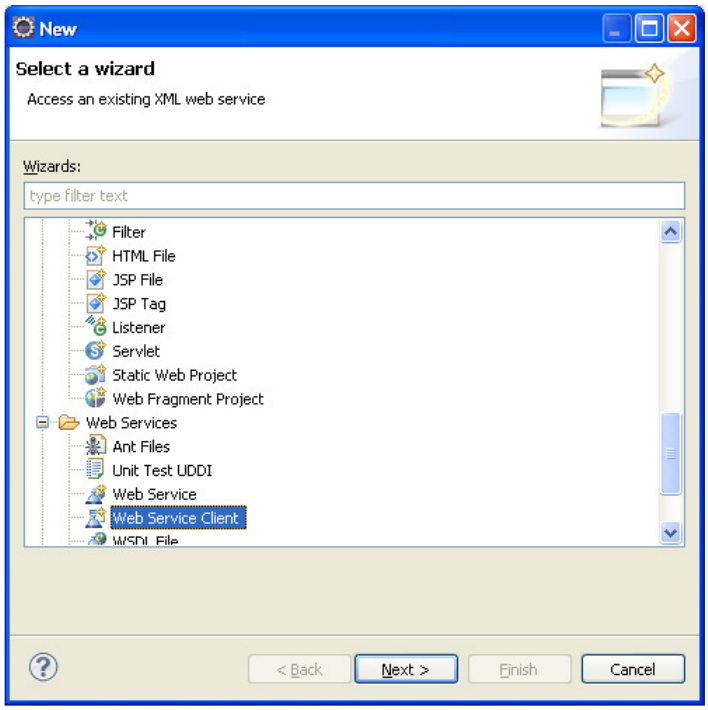

Run On Server
Run On Server
Select which server to use
How do you want to select the server?
Choose an existing server
Manually define a new server
Select the server that you want to use:
type filter text
localhost
Tomcat v7.0 Server at localhost
Started
Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.
Always use this server when running this project
< Back
Next >
Finish
Cancel

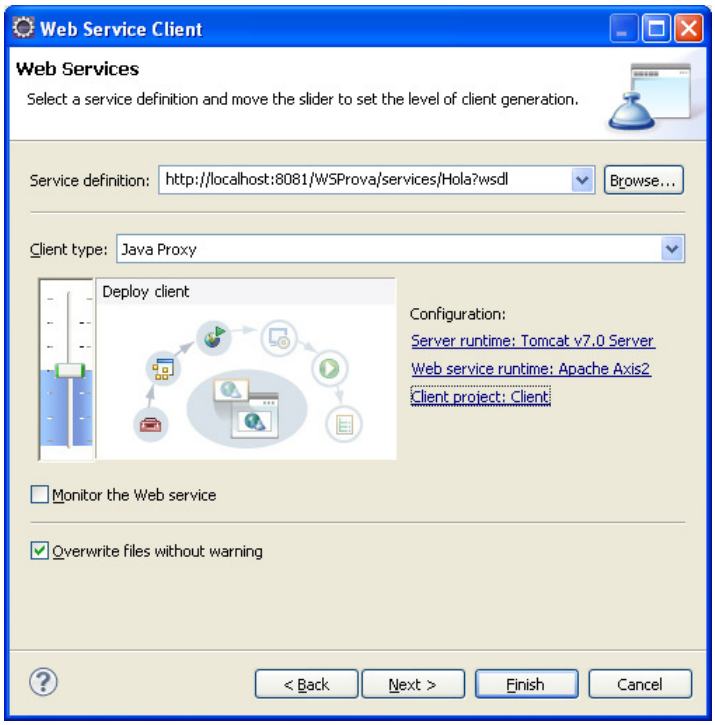

Java EE - http://localhost:8081/WSProva/axis2-web/index.jsp - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
Axis 2 - Home
List Services
http://localhost:8081/WSProva/axis2-web/index.jsp
The Apache Software Foundation
http://www.apache.org/
Welcome!
Welcome to the new generation of Axis. If you can see this page you have successfully deployed the Axis2 Web Application. However, to ensure that A
working, we encourage you to click on the validate link.
Services
View the list of all the available services deployed in this server.
Validate
Check the system to see whether all the required libraries are in place and view the system information.
Administration
Console for administering this Axis2 installation.
Copyright © 1999-2006, The Apache Software Foundation
Licensed under the Apache License, Version 2.0
An outline is not available.
Listo

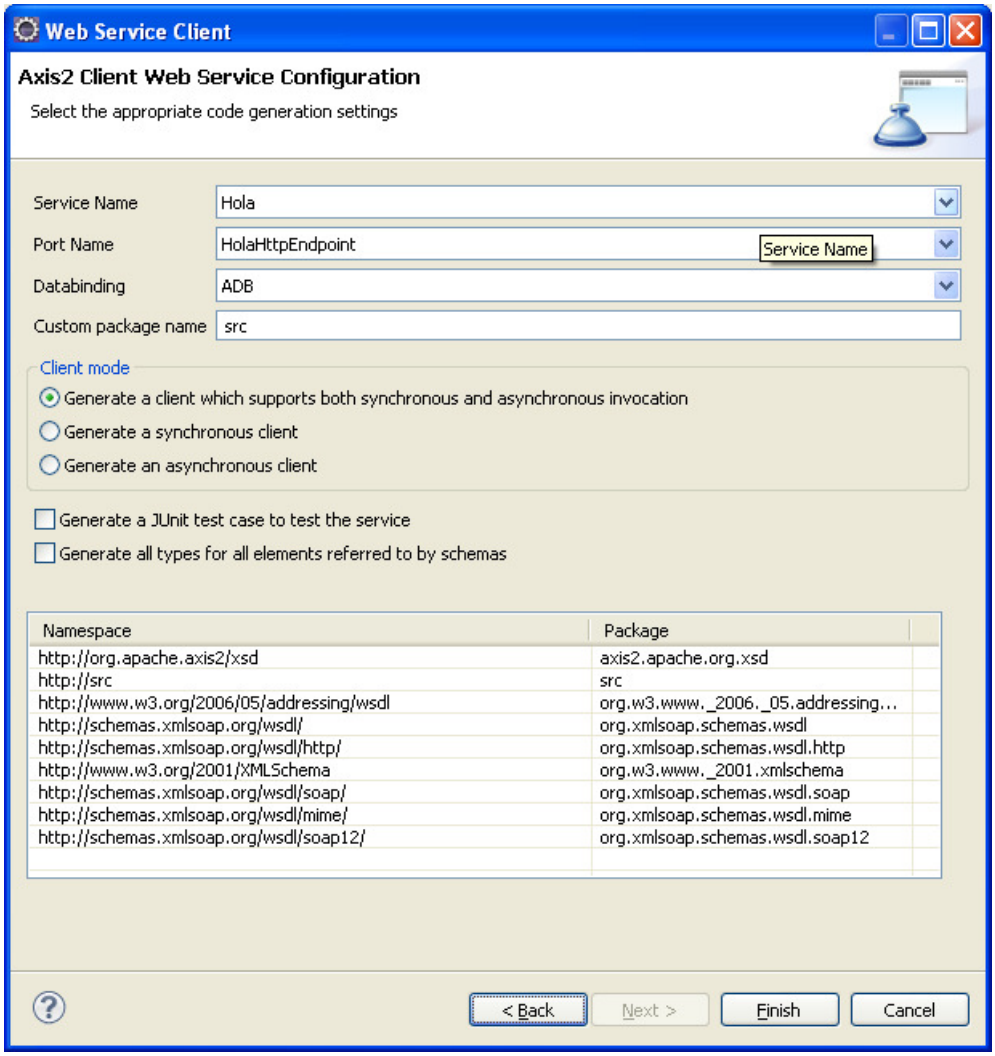

Java EE - http://localhost:8081/WSProva/axis2-web/index.jsp - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
List Services
http://localhost:8081/WSProva/services/listServices
The Apache Software Foundation
http://www.apache.org/
Back Hon
Available services
Version
Service Description : Version
Service EPR : http://localhost:8081/WSProva/services/Version
Service Status : Active
Available Operations
getVersion
Hola
Service Description : Please Type your service description here
Service EPR : http://localhost:8081/WSProva/services/Hola
Service Status : Active
Available Operations
Saludar
Copyright © 1999-2006, The Apache Software Foundation
Licensed under the Apache License, Version 2.0.

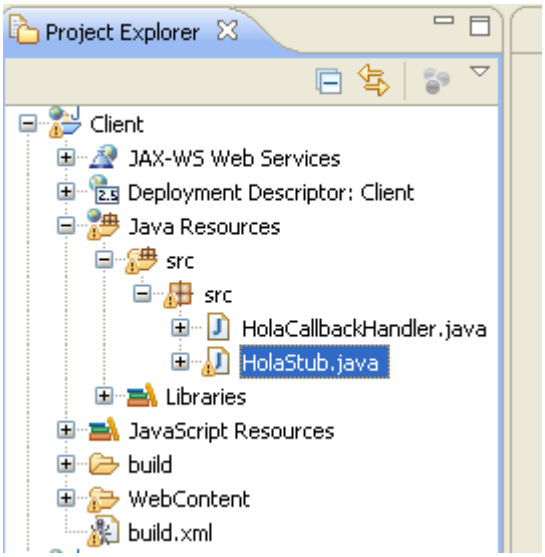

Java EE - http://localhost:8081/WSProva/axis2-web/index.jsp - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
http://localhost:8081/WSProva/services/Hola?wsdl
List Services
<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:ns="http://src" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://src">
<wsdl:documentation>Please Type your service description here</wsdl:documentation>
<wsdl:types>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://src">
<xs:element name="Saludar">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="nom" nillable="true" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SaludarResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="return" nillable="true" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="SaludarRequest">
<wsdl:part name="parameters" element="ns:Saludar" />
</wsdl:message>
<wsdl:message name="SaludarResponse">
<wsdl:part name="parameters" element="ns:SaludarResponse" />
</wsdl:message>
<wsdl:portType name="HolaPortType">
<wsdl:operation name="Saludar">
</wsdl:portType>
<wsdl:binding name="HolaSoap11Binding" type="ns:HolaPortType">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<wsdl:operation name="Saludar">
<soap:operation soapAction="urn:Saludar" style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="HolaSoap12Binding" type="ns:HolaPortType">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
Listo

3.2. Per generar el client anirem a **"File >> New >> Other >> WebServiceClient".** Clicarem **"Next"** i s'obrirà una finestra on haurem d'indicar: 1) l'adreça del fitxer wsdl (en el nostre exemple aquesta adreça és http://localhost:8081/WSProva/services/Hola?wsdl), 2) Web Service Runtime: Apache Axis2 i 3) Client Project (el nom del nostre client que en el nostre cas serà *Client*). Després clicarem **"Next"** i s'obrirà una nova finestra. Clicarem **"Finish".** Podeu comprovar que s'ha generat un nou projecte *Client* i la classe java HolaStub.java que és un representant del servei que hem definit i que utilitzarem per invocar a l'operació *Saludar* del servei.

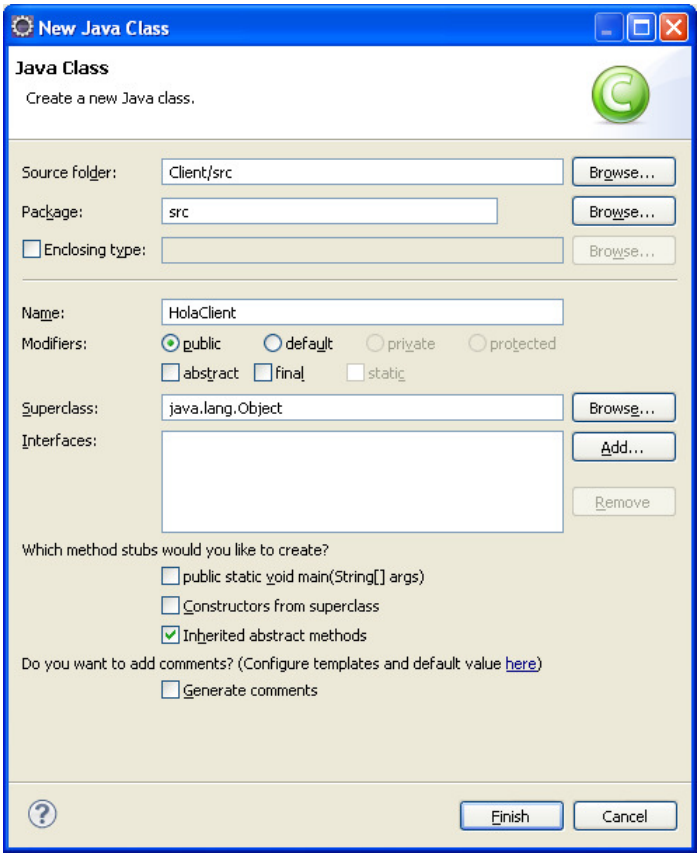








3.3. Ara escriurem el codi del client Java que invocarà al servei (mitjançant l'Stub).



HolaClient.java

```
package src;

import java.rmi.RemoteException;

import org.apache.axis2.AxisFault;

import src.HolaStub.Saludar;
import src.HolaStub.SaludarResponse;

public class HolaClient {
    public static void main(String[] args) {
        try {
            String nom = "Cristina";
            HolaStub stub= new HolaStub();
            Saludar salutació = new Saludar();
            salutació.setNom(nom);
            SaludarResponse res = stub.saludar(salutació);
            System.out.println(res.get_return());
        } catch (AxisFault e){
            e.printStackTrace();
        } catch (RemoteException e) {
            e.printStackTrace();
        }
        }
}
```

3.4. Seleccionarem el fitxer HolaClient i amb el botó dret clicarem **"Run As >> Java Application"**. Obtindrem el resultat a la consola.