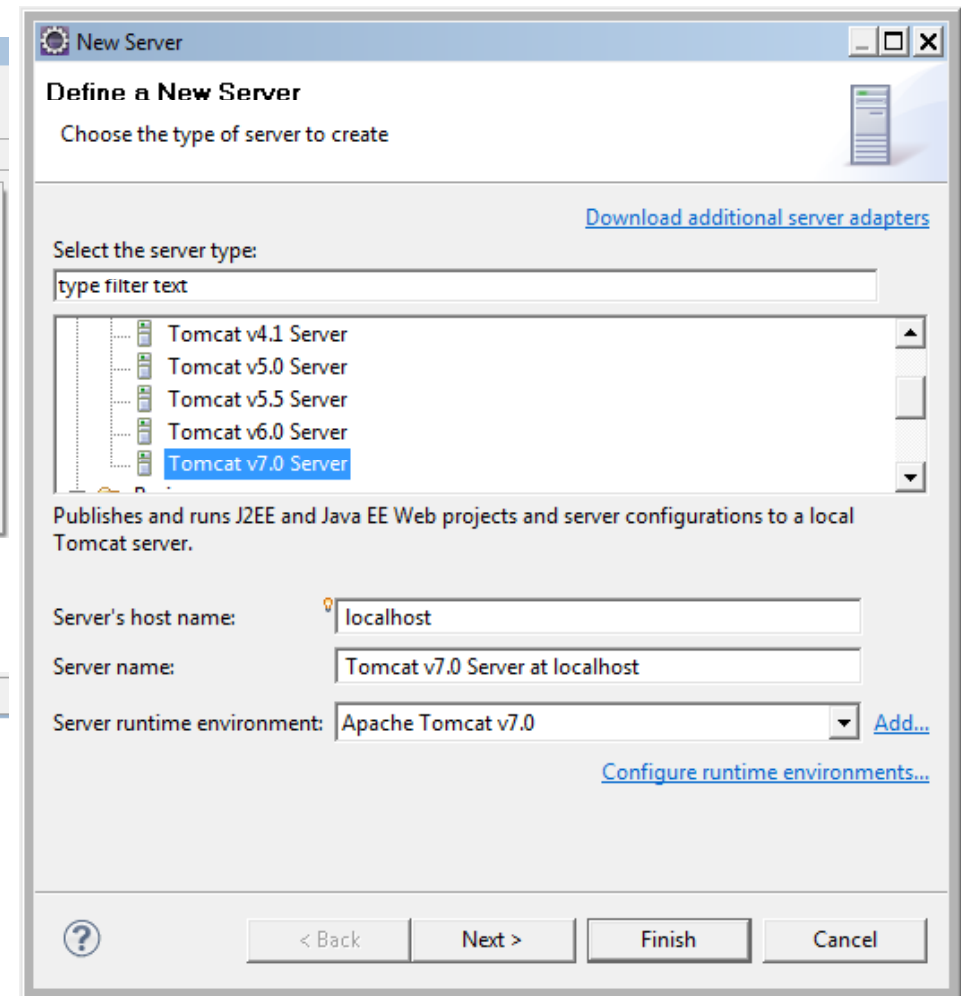
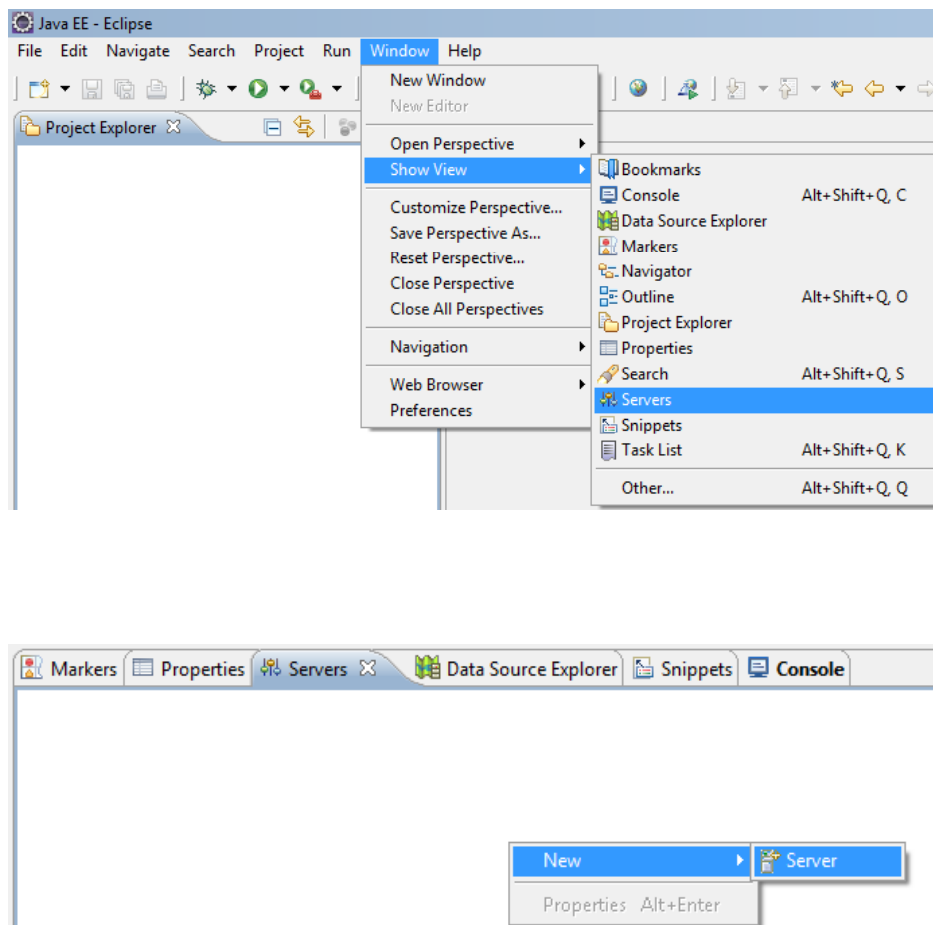
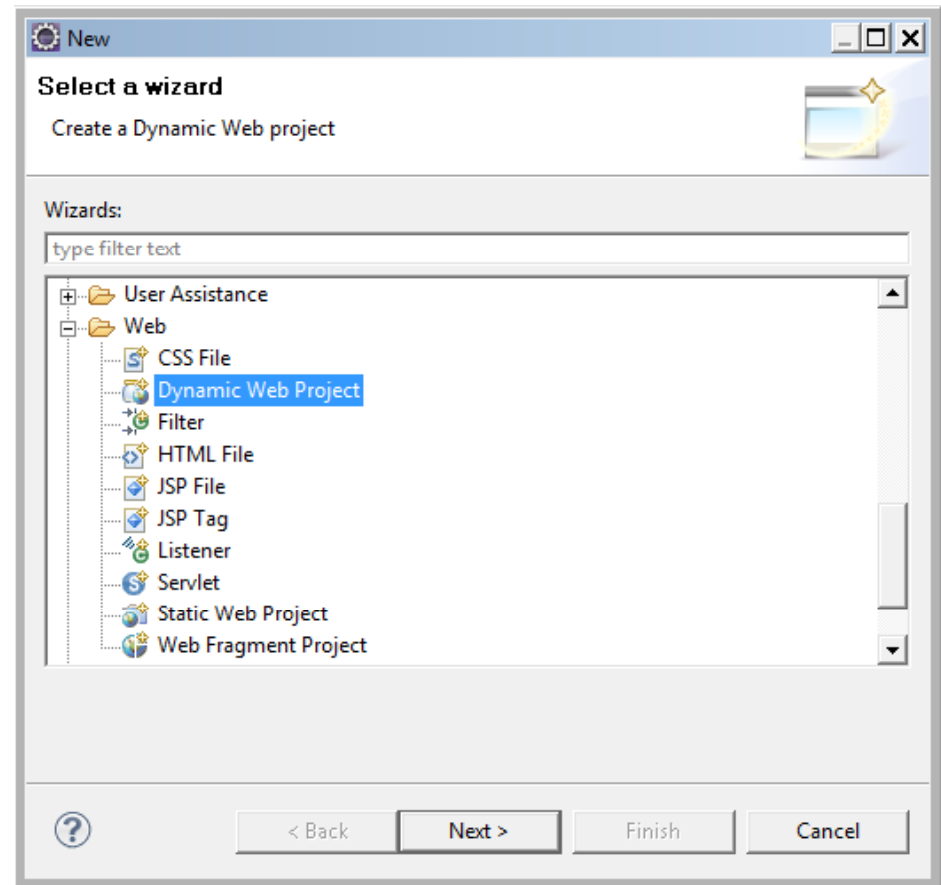
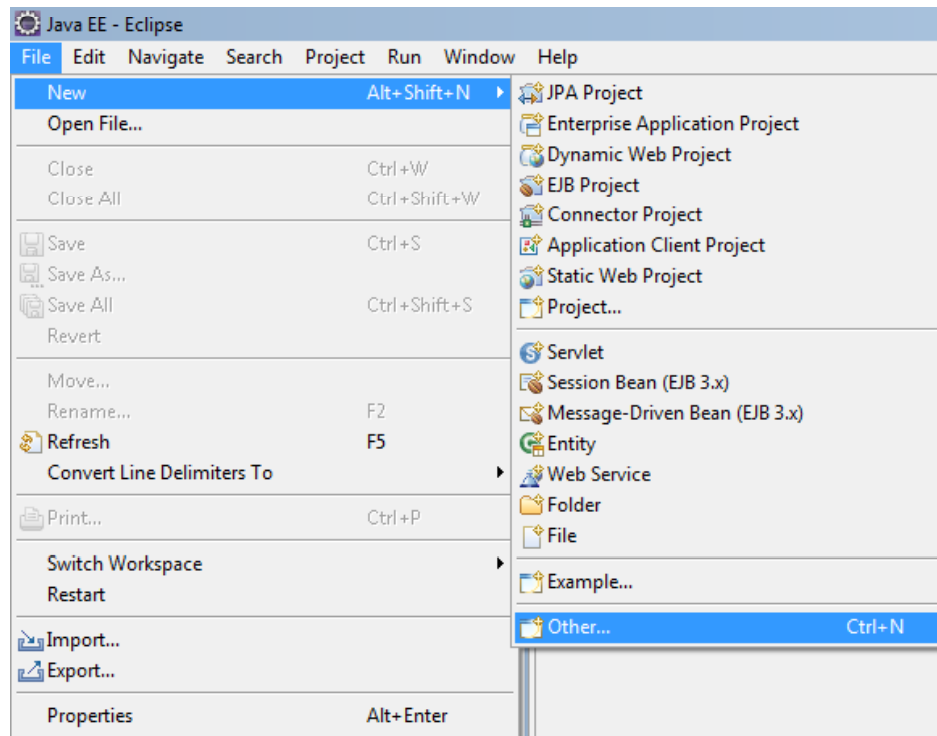


Web Services

**Les services Web : vers l'interopérabilité
des applications réparties sur internet**





New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Configuration

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership

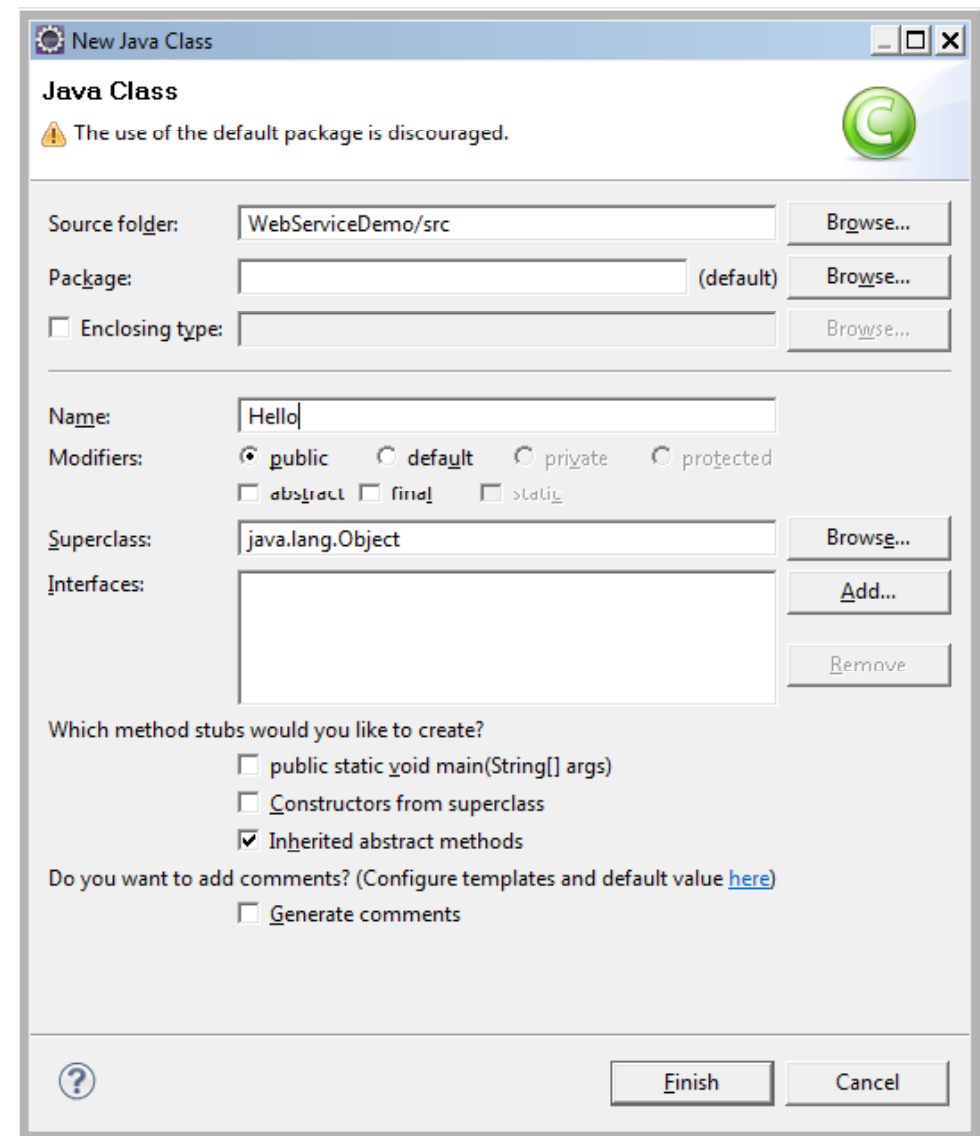
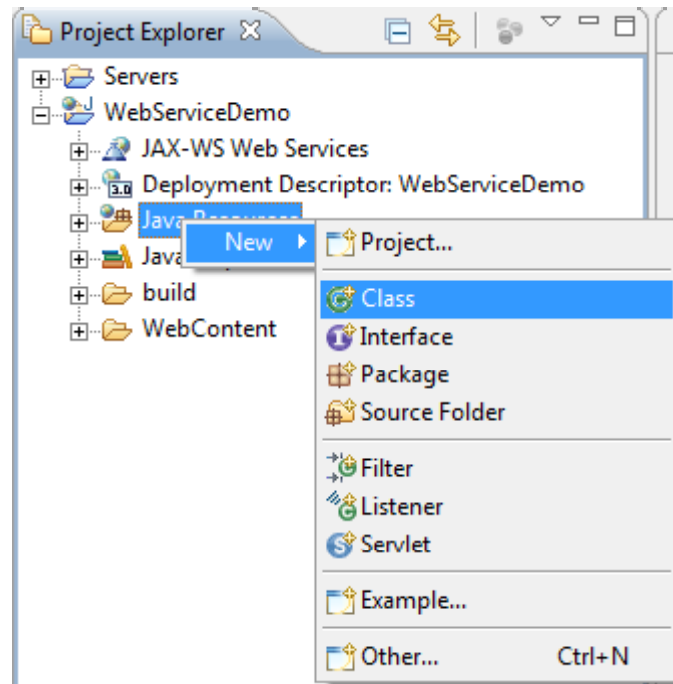
☐ Add project to an EAR

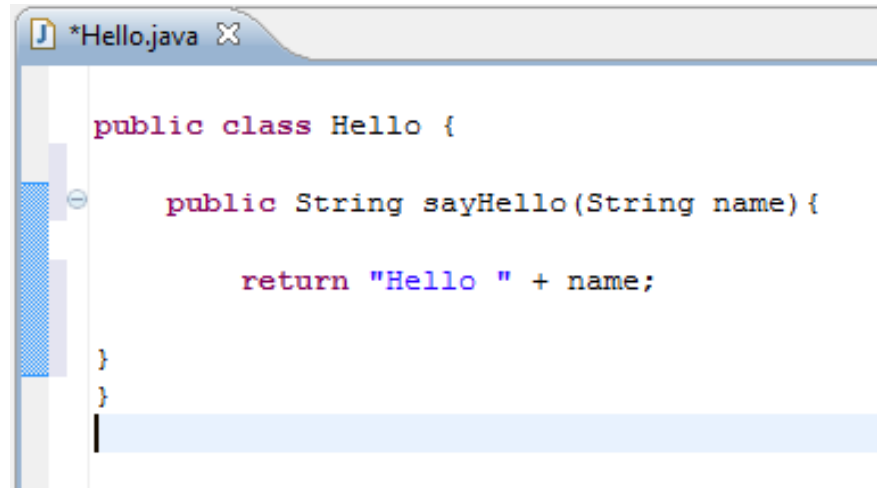
EAR project name:

Working sets

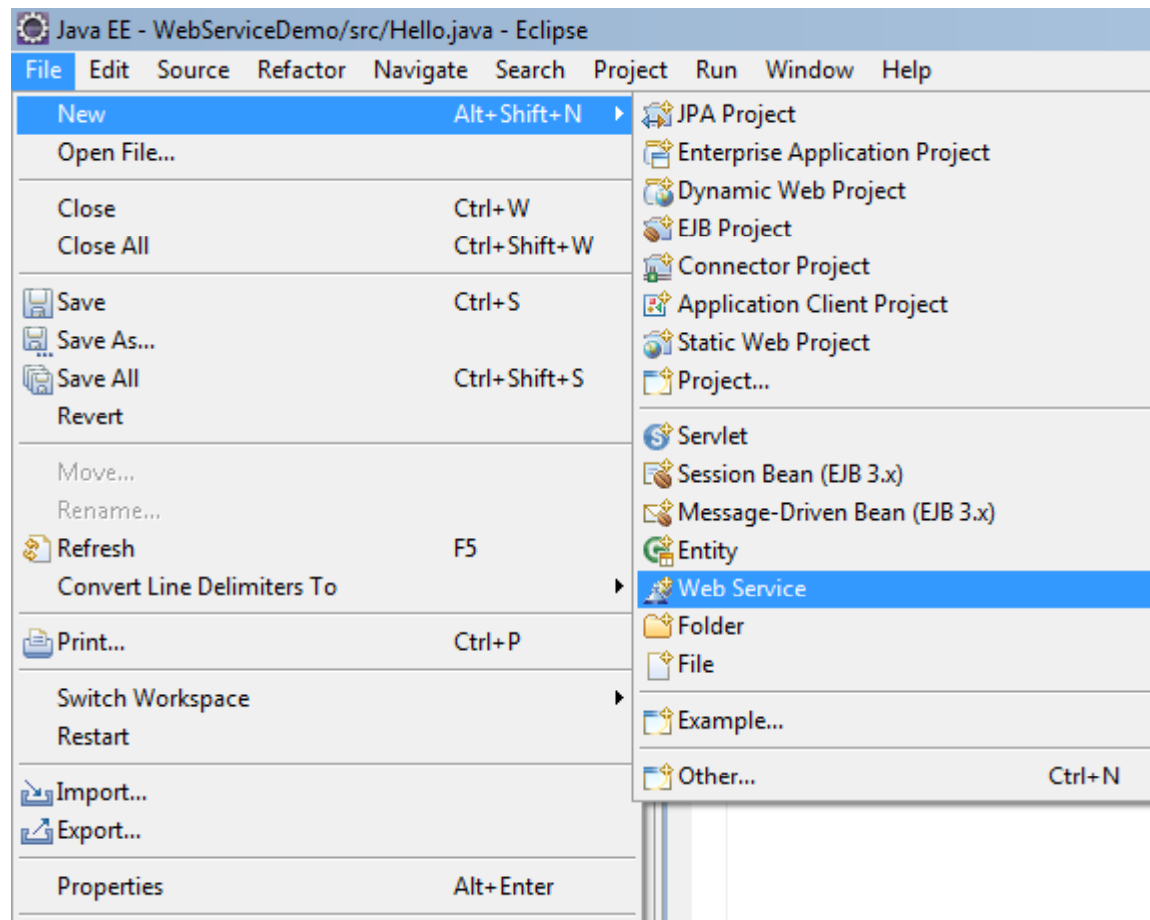
☐ Add project to working sets

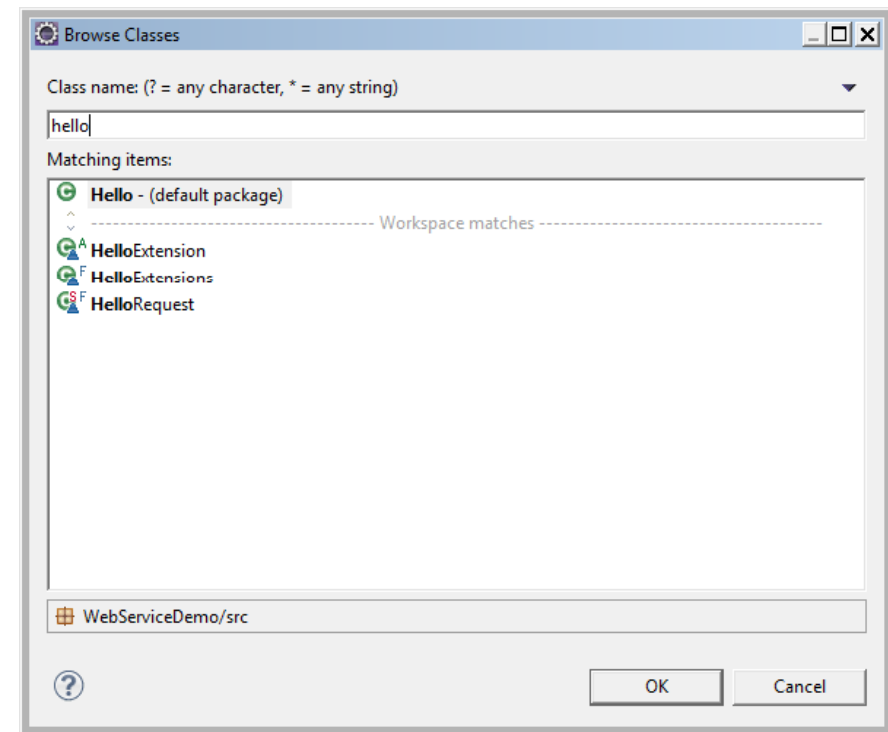
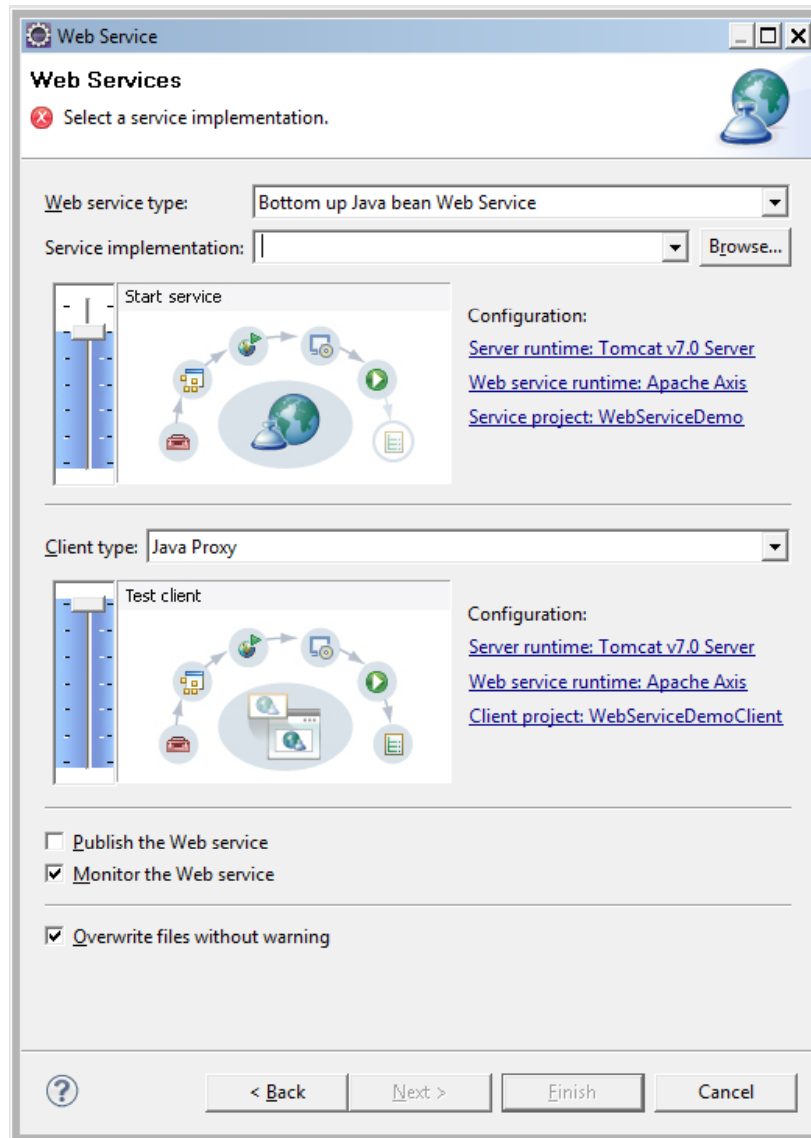
Working sets:





```
public class Hello {  
    public String sayHello(String name){  
        return "Hello " + name;  
    }  
}
```





Web Service

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: Hello Browse...

Start service

Configuration:

- [Server runtime: Tomcat v7.0 Server](#)
- [Web service runtime: Apache Axis](#)
- [Service project: WebServiceDemo](#)

Client type: Java Proxy

Test client

Configuration:

- [Server runtime: Tomcat v7.0 Server](#)
- [Web service runtime: Apache Axis](#)
- [Client project: WebServiceDemoClient](#)

☐ Publish the Web service

☒ Monitor the Web service

☒ Overwrite files without warning

? < Back Next > Finish Cancel

Web Service

Web Service Java Bean Identity

Configure the Java bean as a Web service.

WSDL file: Hello.wsdl

Methods

☒ sayHello(java.lang.String)

Java bean methods to include.

Select All Deselect All

Style and use

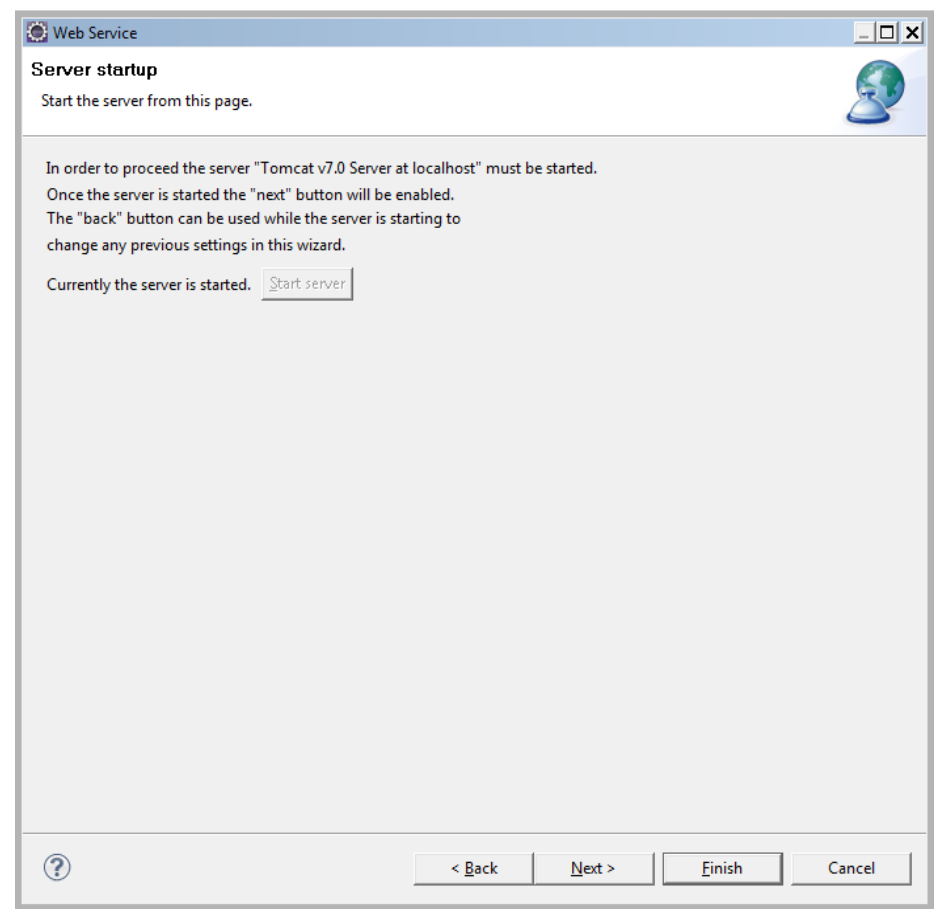
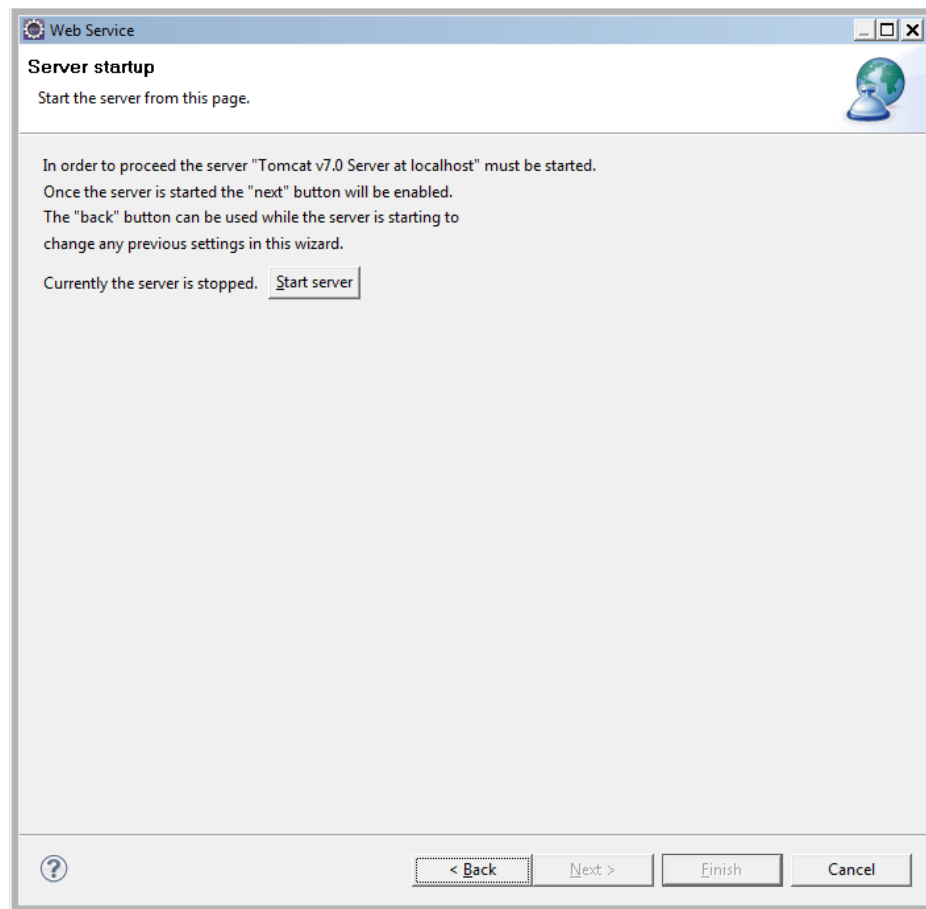
☒ document/literal (wrapped)

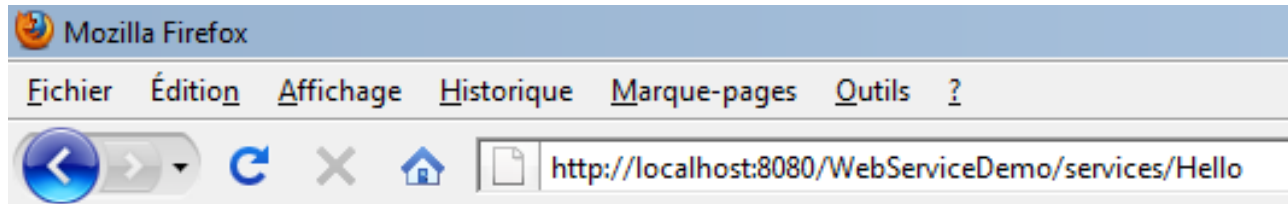
☐ document/literal

☐ RPC/encoded

☐ Define custom mapping for package to namespace.

? < Back Next > Finish Cancel

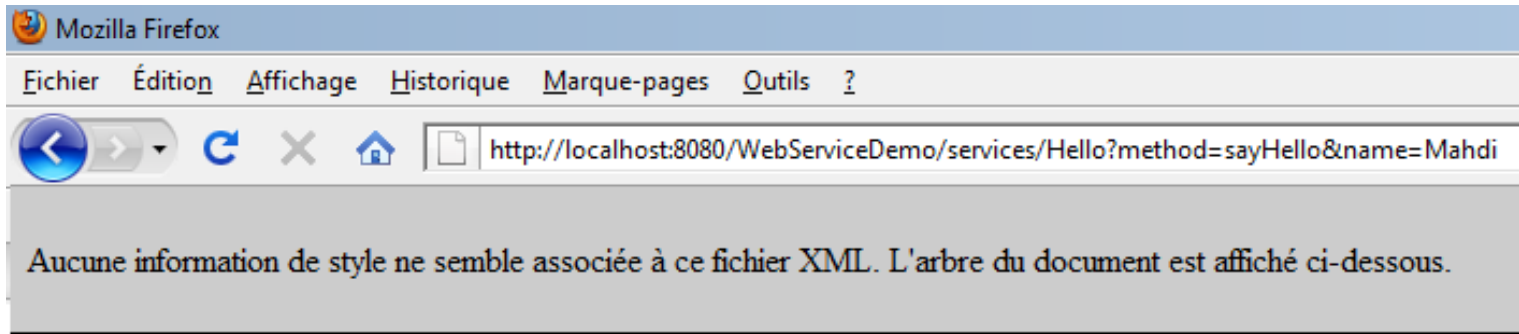




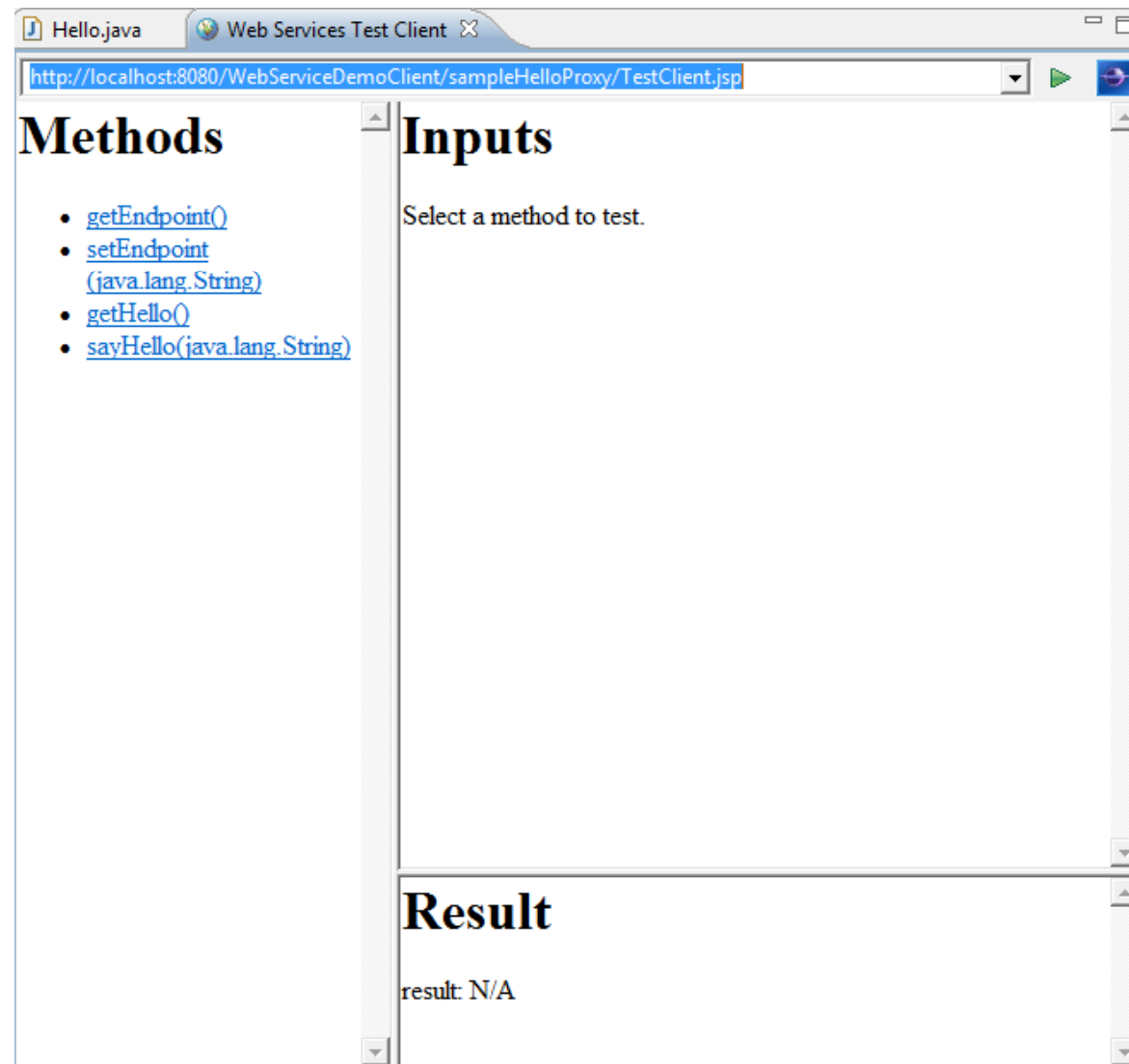
Hello

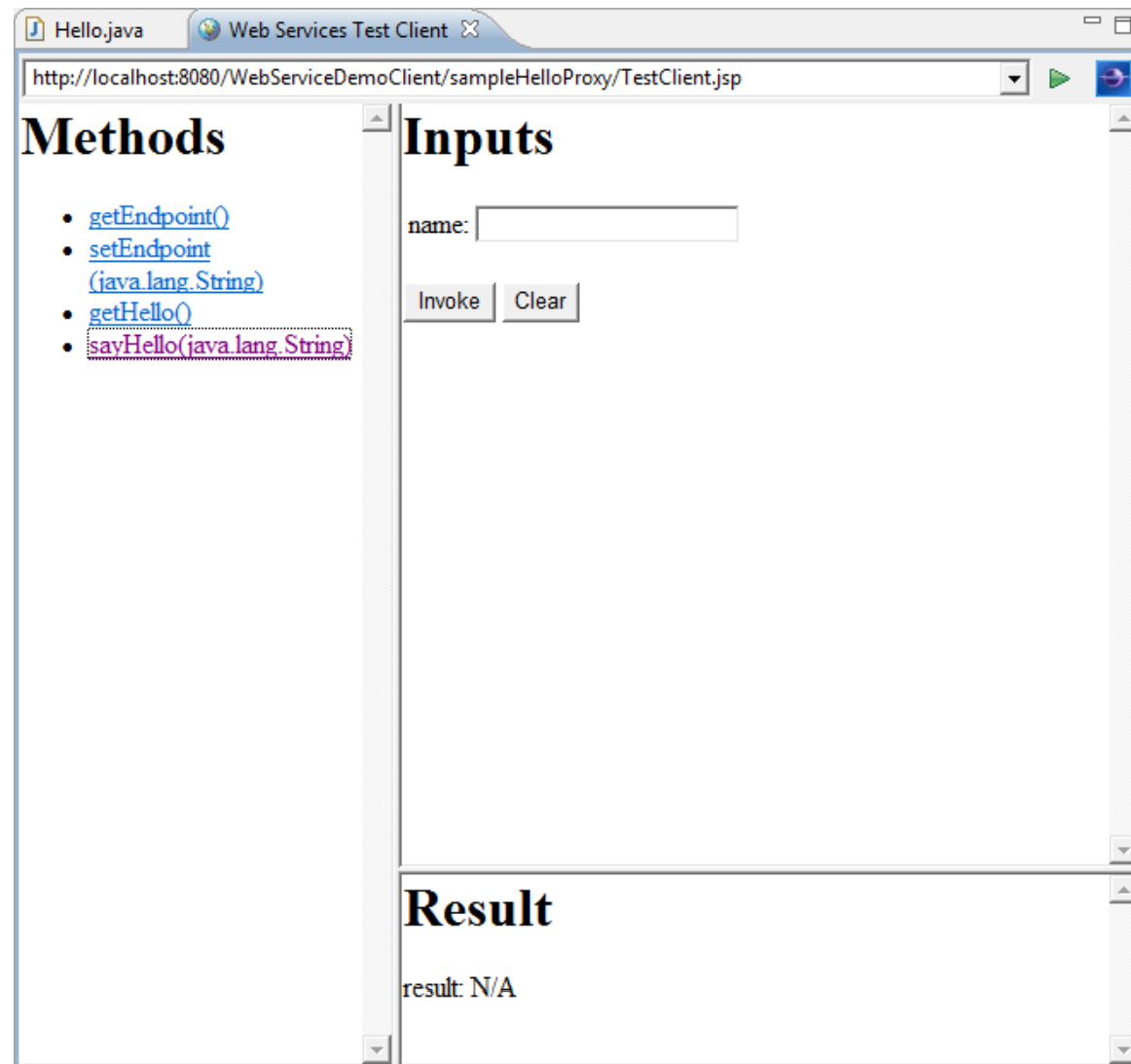
Hi there, this is an AXIS service!

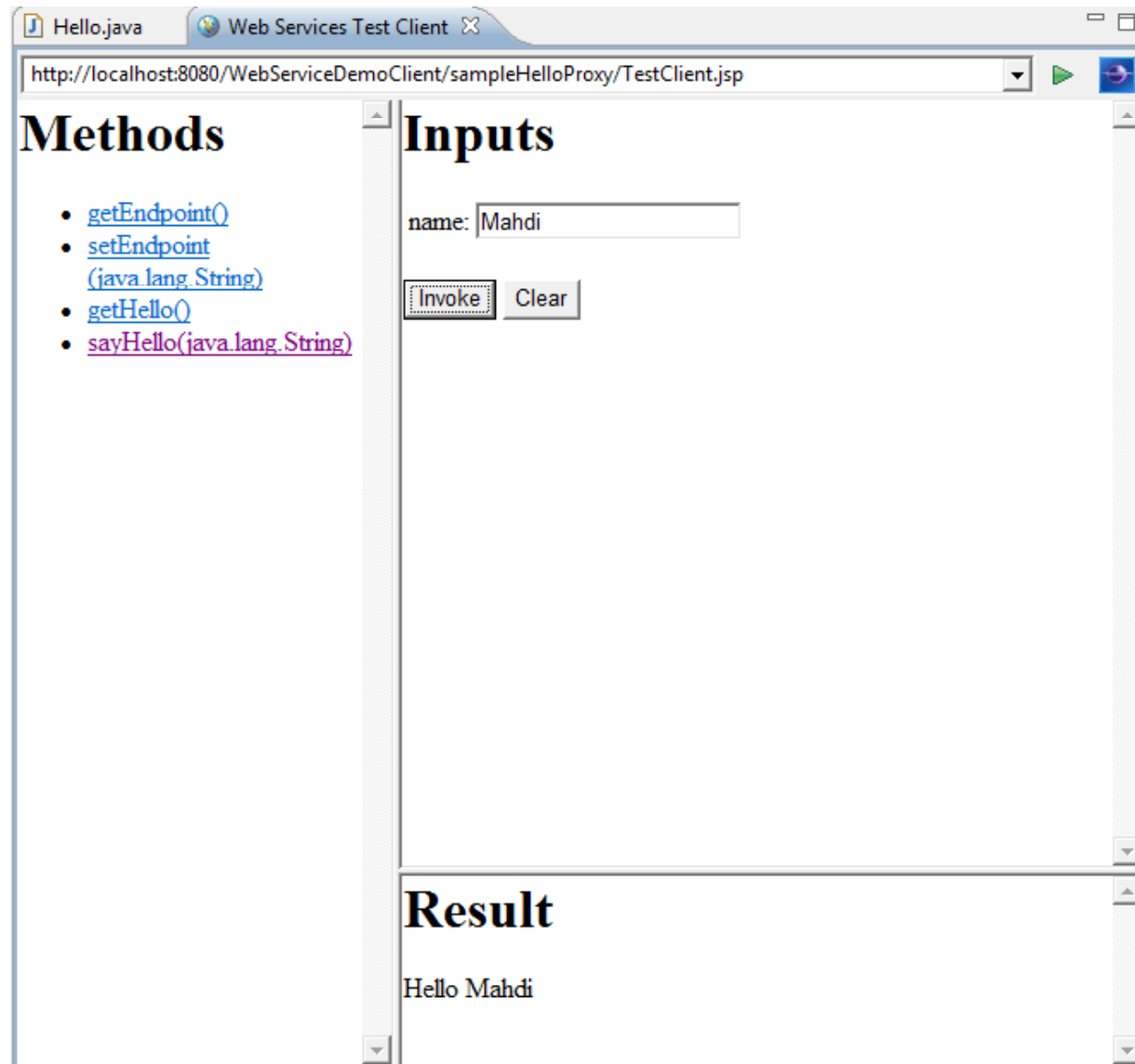
Perhaps there will be a form for invoking the service here...

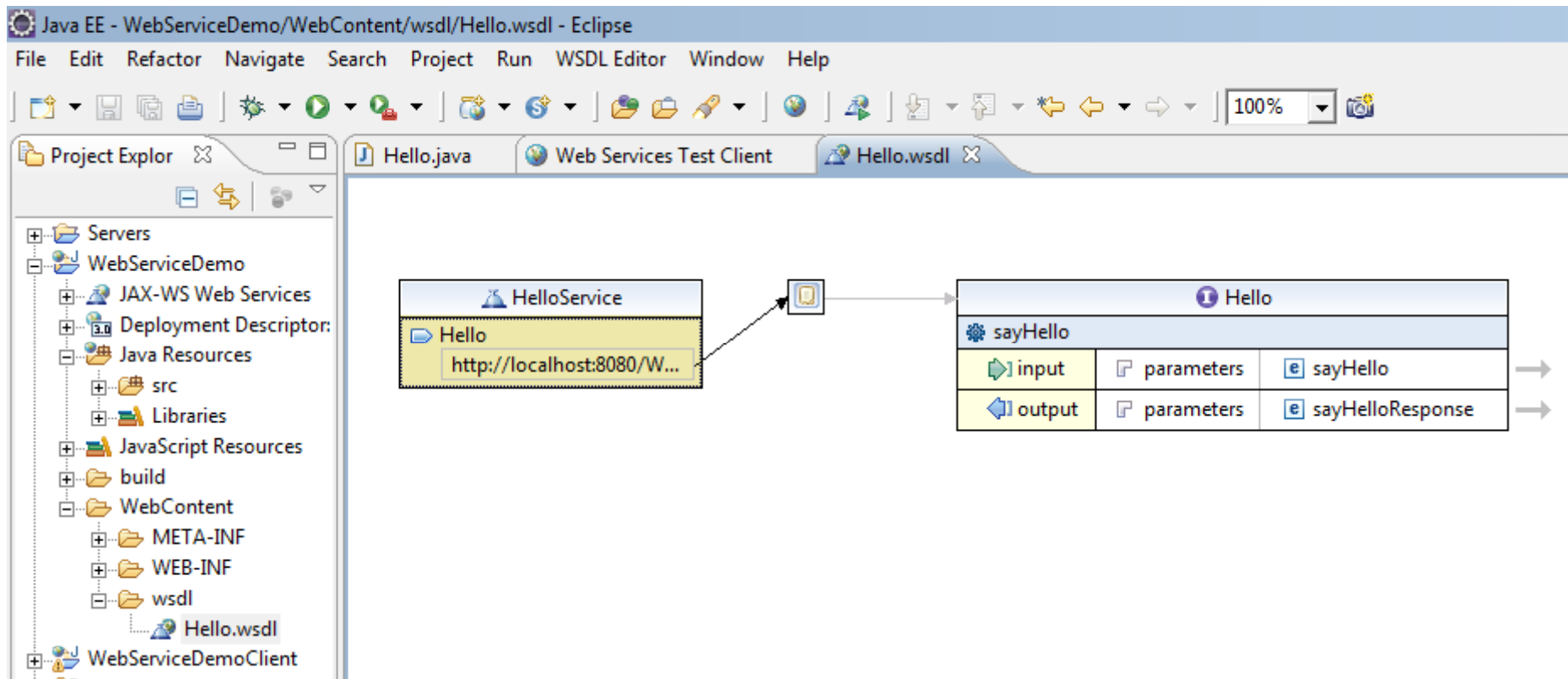


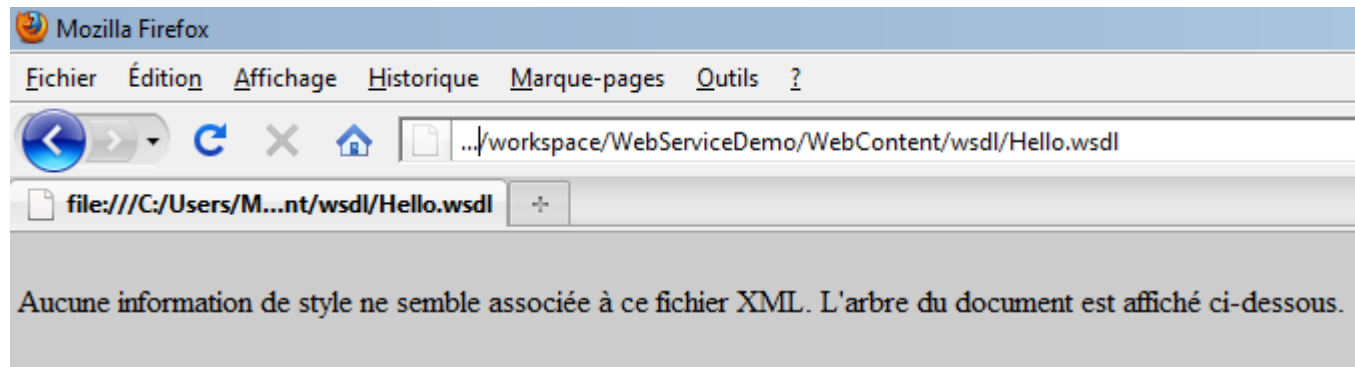
```
- <soapenv:Envelope>  
  - <soapenv:Body>  
    - <sayHelloResponse>  
      <ns1:sayHelloReturn>Hello Mahdi</ns1:sayHelloReturn>  
    </sayHelloResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```











```
-<wsdl:definitions targetNamespace="http://DefaultNamespace">
+<!-->
+<wsdl:types></wsdl:types>
+<wsdl:message name="sayHelloResponse"></wsdl:message>
+<wsdl:message name="sayHelloRequest"></wsdl:message>
- <wsdl:portType name="Hello">
-   -<wsdl:operation name="sayHello">
-       <wsdl:input message="impl:sayHelloRequest" name="sayHelloRequest"> </wsdl:input>
-       <wsdl:output message="impl:sayHelloResponse" name="sayHelloResponse"> </wsdl:output>
-   </wsdl:operation>
- </wsdl:portType>
+<wsdl:binding name="HelloSoapBinding" type="impl:Hello"></wsdl:binding>
- <wsdl:service name="HelloService">
-   -<wsdl:port binding="impl:HelloSoapBinding" name="Hello">
-       <wsdlsoap:address location="http://localhost:8080/WebServiceDemo/services/Hello"/>
-   </wsdl:port>
- </wsdl:service>
</wsdl:definitions>
```

Java EE - WebServiceDemo/WebContent/WEB-INF/HelloService/DefaultNamespace/deploy.wsdd - Eclipse

File Edit Navigate Search Project Run Design Window Help

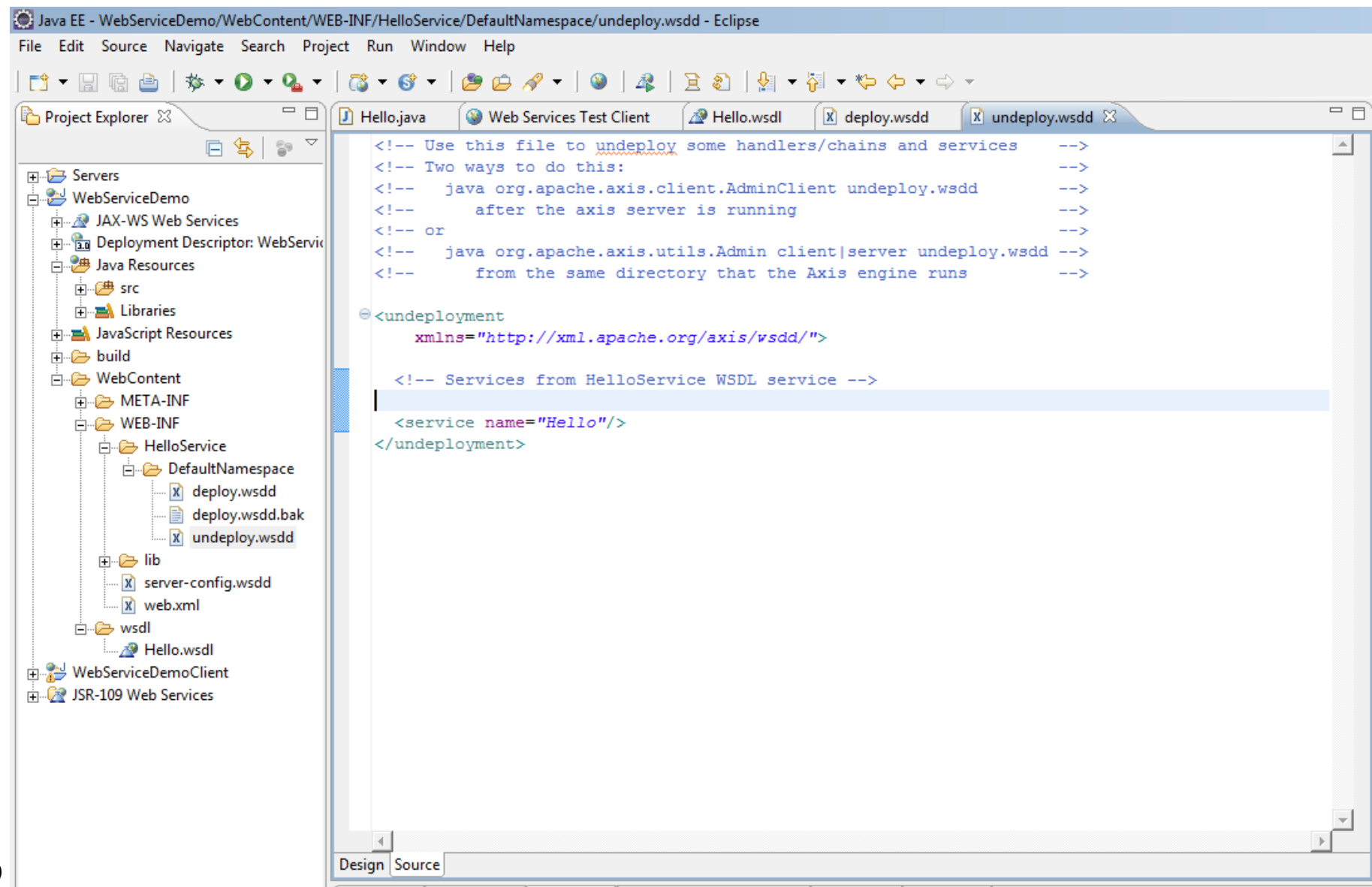
Project Explorer

- Servers
 - WebServiceDemo
 - JAX-WS Web Services
 - Deployment Descriptor: WebService
 - Java Resources
 - src
 - Libraries
 - JavaScript Resources
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - HelloService
 - DefaultNamespace
 - deploy.wsdd
 - deploy.wsdd.bak
 - undeploy.wsdd
 - lib
 - server-config.wsdd
 - web.xml
 - wsdl
 - Hello.wsdl
 - WebServiceDemoClient
 - JSR-109 Web Services

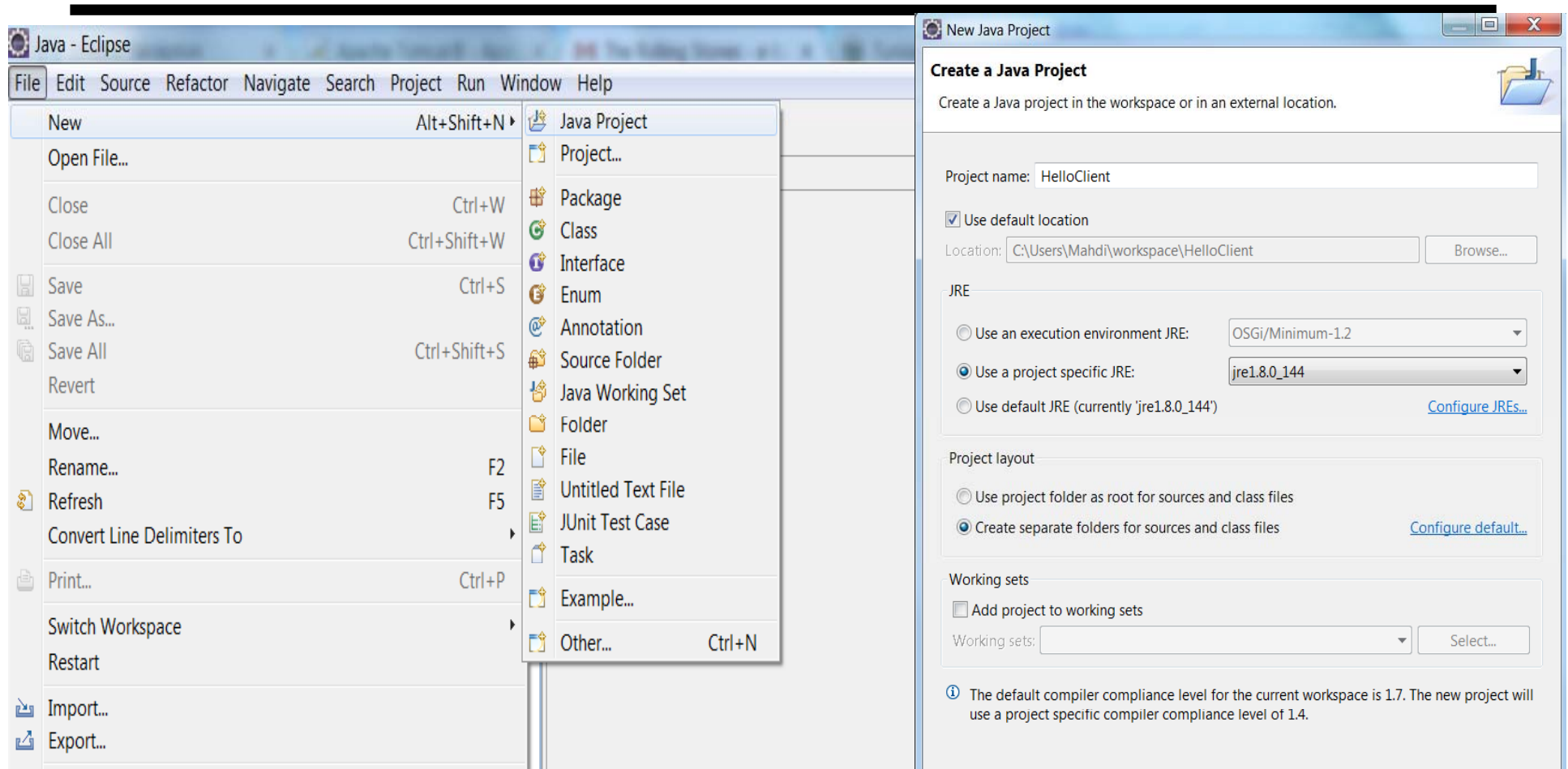
Editors

Hello.java Web Services Test Client Hello.wsdl deploy.wsdd undeploy.wsdd

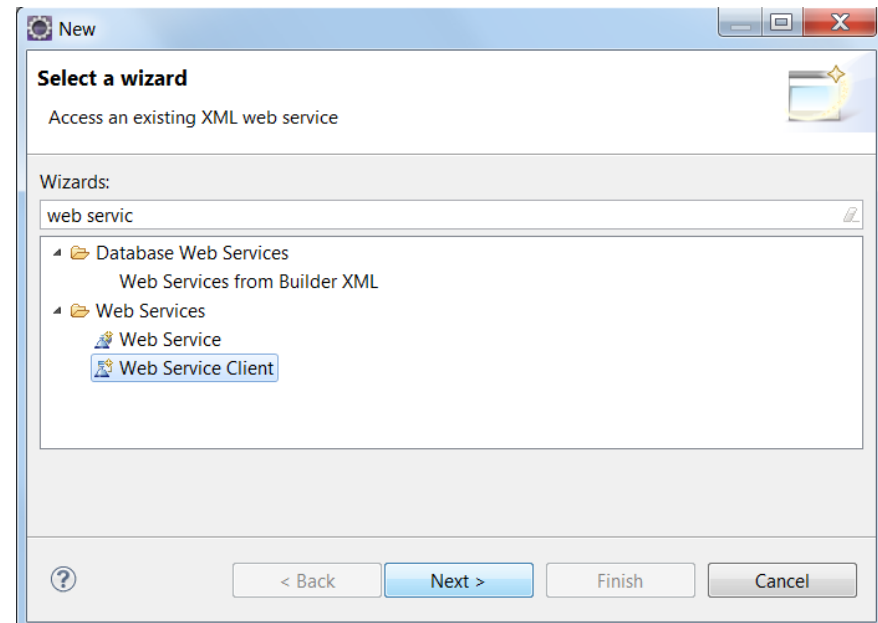
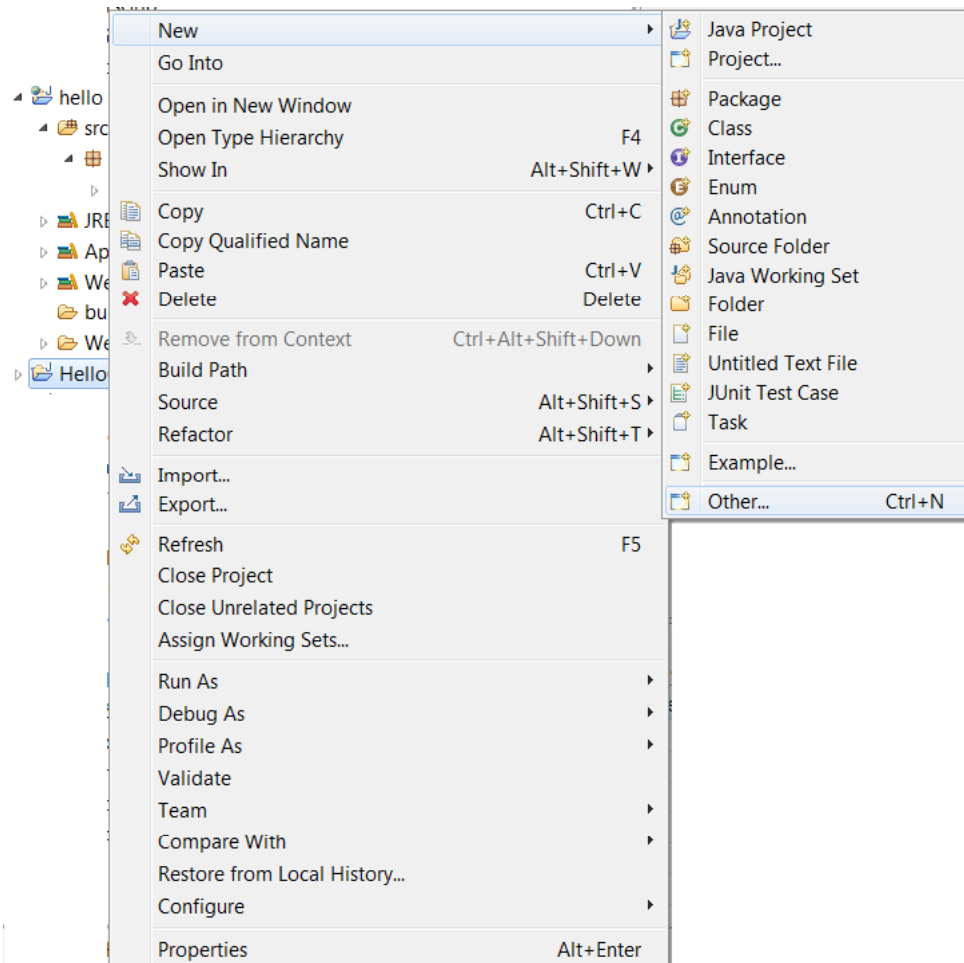
Node	Content
?? xml	version="1.0" encoding="UTF-8"
!--	Use this file to deploy some handlers/chains and services
!--	Two ways to do this:
!--	java org.apache.axis.client.AdminClient deploy.wsdd
!--	after the axis server is running
!--	or
!--	java org.apache.axis.utils.Admin client server deploy.wsdd
!--	from the same directory that the Axis engine runs
deployment	
xmlns	http://xml.apache.org/axis/wsdd/
xmlns:java	http://xml.apache.org/axis/wsdd/providers/java
--	Services from HelloService WSDL service
service	
name	Hello
provider	java:RPC
style	wrapped
use	literal
parameter	
parameter	
parameter	
parameter	
parameter	
parameter	
parameter	
parameter	
operation	
parameter	



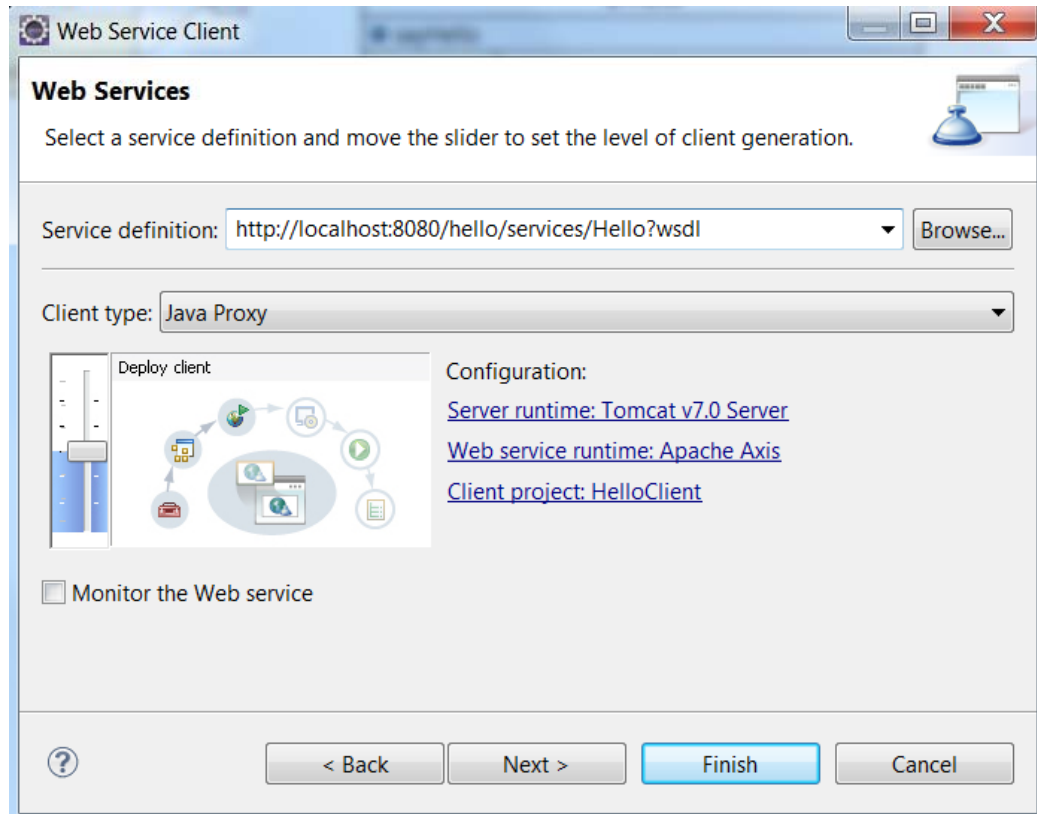
Créer un Client pour le service Web Hello













New Web Service Client

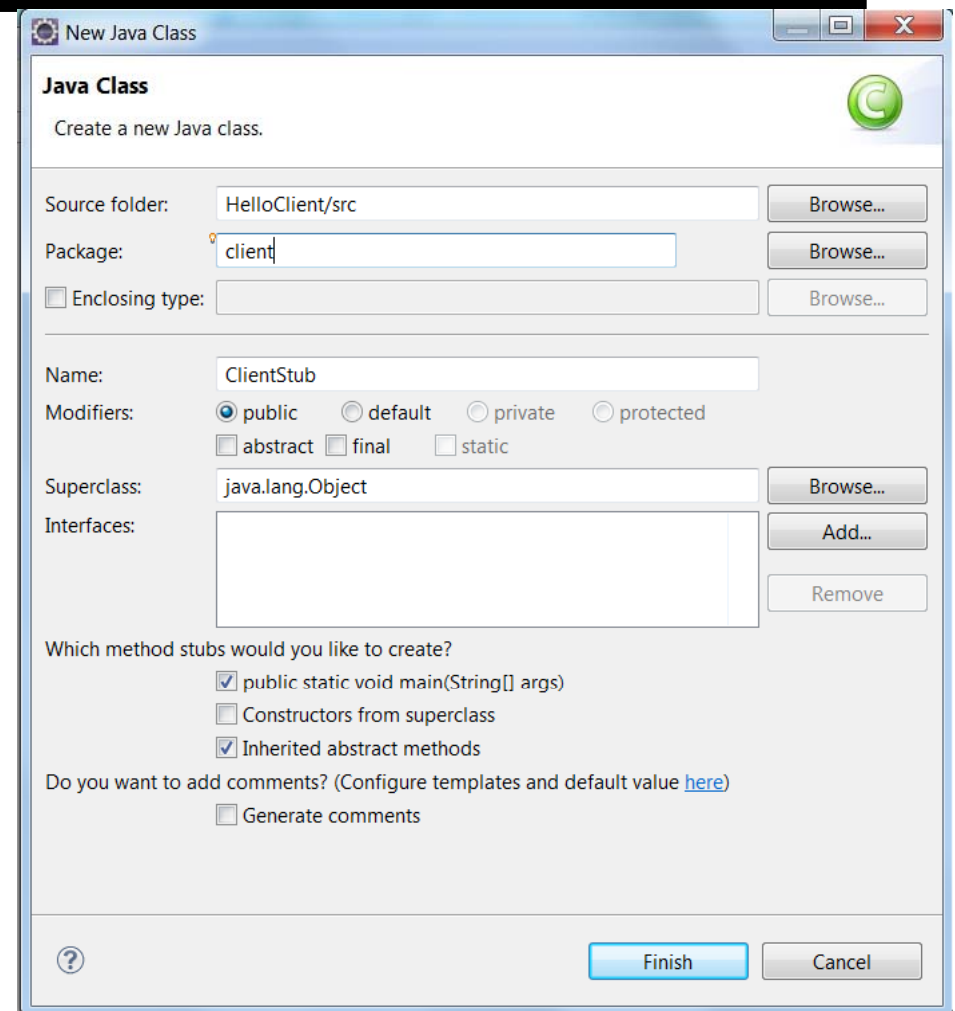
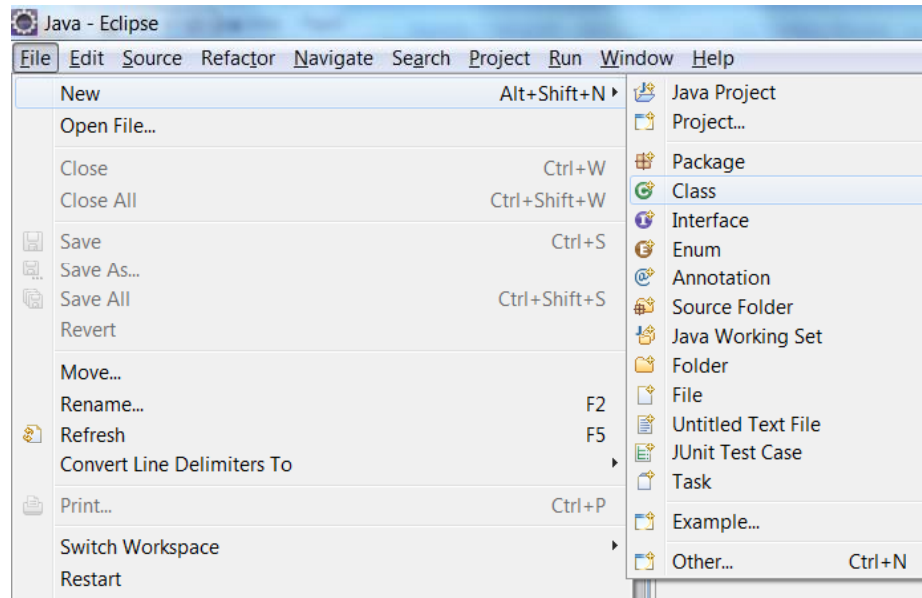


New Web Service Client



- ▲  HelloClient
 - ▲  src
 - ▲  hello
 - ▷  Hello.java
 - ▷  HelloProxy.java
 - ▷  HelloService.java
 - ▷  HelloServiceLocator.java
 - ▷  HelloSoapBindingStub.java
 - ▷  JRE System Library [jre1.8.0_144]
 - ▷  Referenced Libraries

Le main Client



ClientStub.java

```
package client;

import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;
import hello.Hello;
import hello.HelloServiceLocator;

public class ClientStub {

    public static void main(String[] args) throws ServiceException, RemoteException {
        Hello service = new HelloServiceLocator().getHello ();
        System.out.println(service.sayHello("Mahdi"));
    }

}
```