Environment Setup for Java Web Services

3 components are required for the complete environment setup:

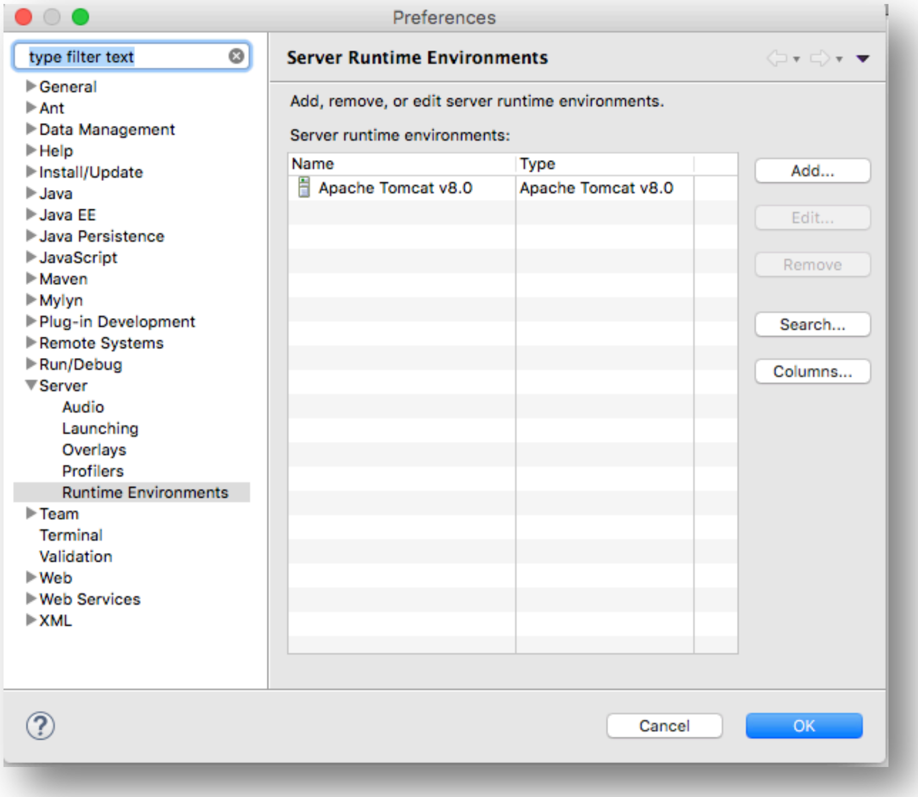
1. Eclipse JEE
2. JDK
3. Apache Tomcat

1. Download the Eclipse IDE for Java EE Developers from the link below
    a. http://www.eclipse.org/downloads/packages/
2. Ensure you have Java Development Kit installed on your machines. If you don't have JDK installed, download and install from the link below:
    a. http://www.oracle.com/technetwork/java/javase/downloads/index.html
3. Download Apache Tomcat 8 from the link:
    a. http://tomcat.apache.org/download-80.cgi

This will setup you machine for creating Java Web Services

The next step will be to configure Eclipse JEE with Apache Tomcat:

1. Go to "Preferences -> Server -> Runtime Environments" in Eclipse



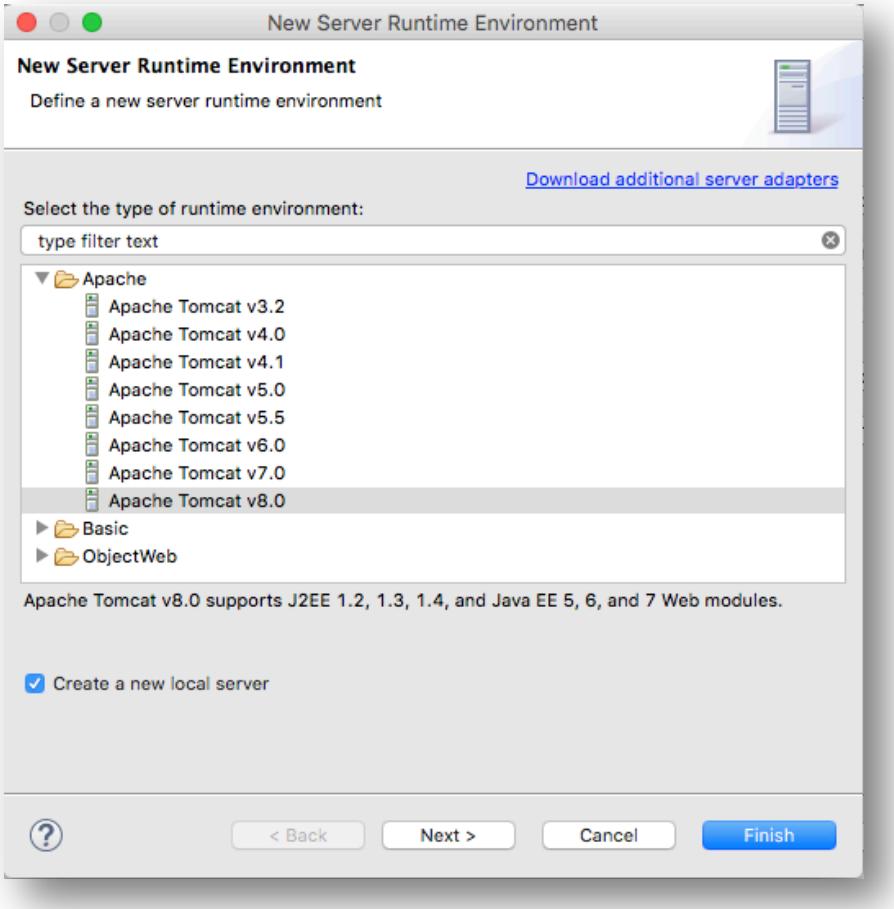
Server Runtime Environments

Add, remove, or edit server runtime environments.

Server runtime environments:

Name	Type
Apache Tomcat v8.0	Apache Tomcat v8.0

2. Click “Add” to a new Server runtime. Select Apache Tomcat v8.0 server. Click next



New Server Runtime Environment

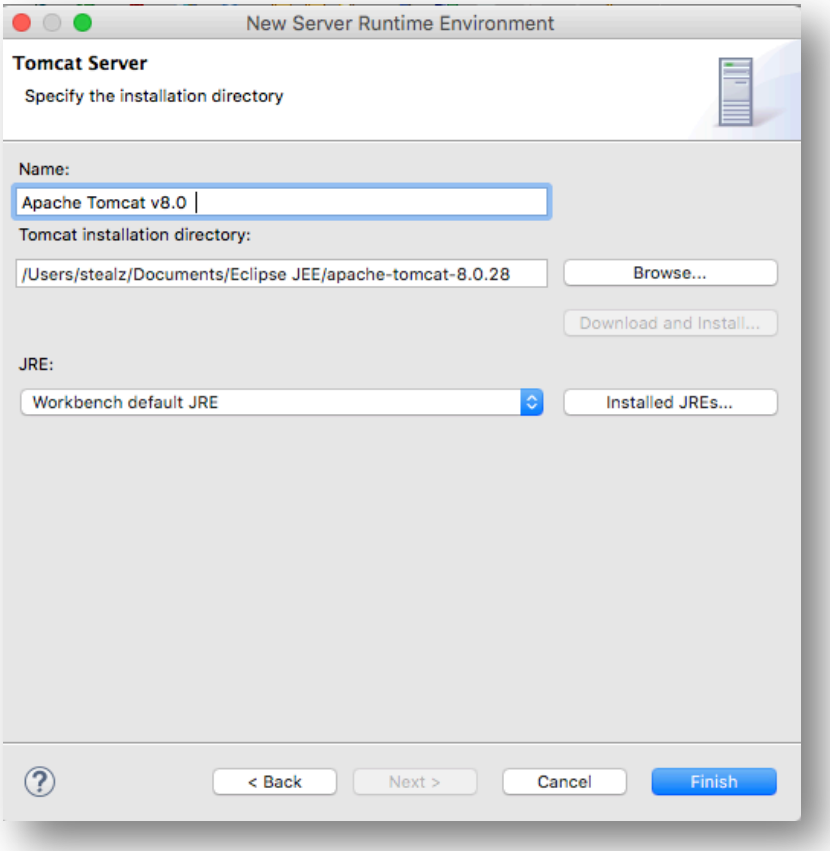
Define a new server runtime environment

Select the type of runtime environment:

Apache Tomcat v8.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, and 7 Web modules.

☑ Create a new local server

3. On the next screen select the tomcat directory, where tomcat is installed



New Server Runtime Environment

Tomcat Server

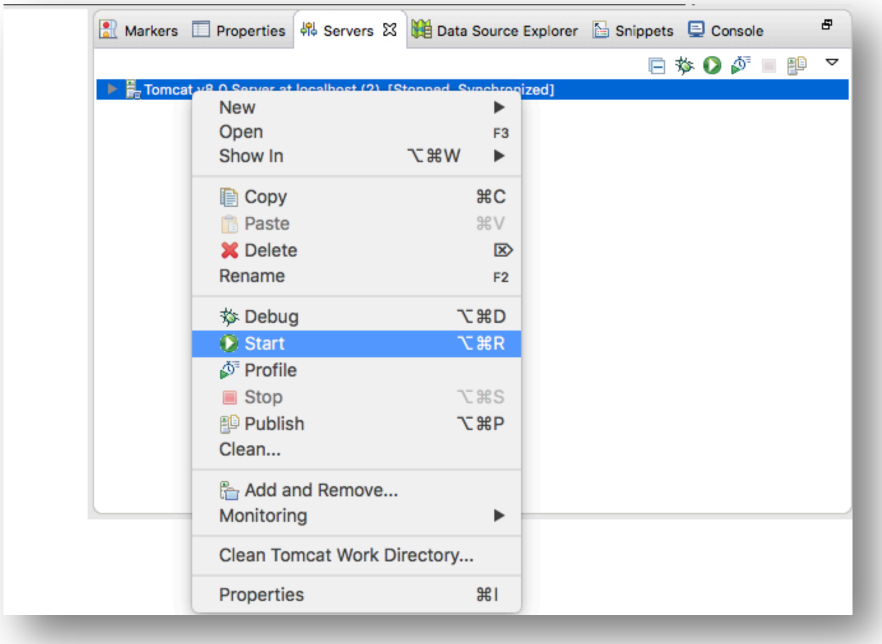
Specify the installation directory

Name: Apache Tomcat v8.0

Tomcat installation directory: /Users/stealz/Documents/Eclipse JEE/apache-tomcat-8.0.28

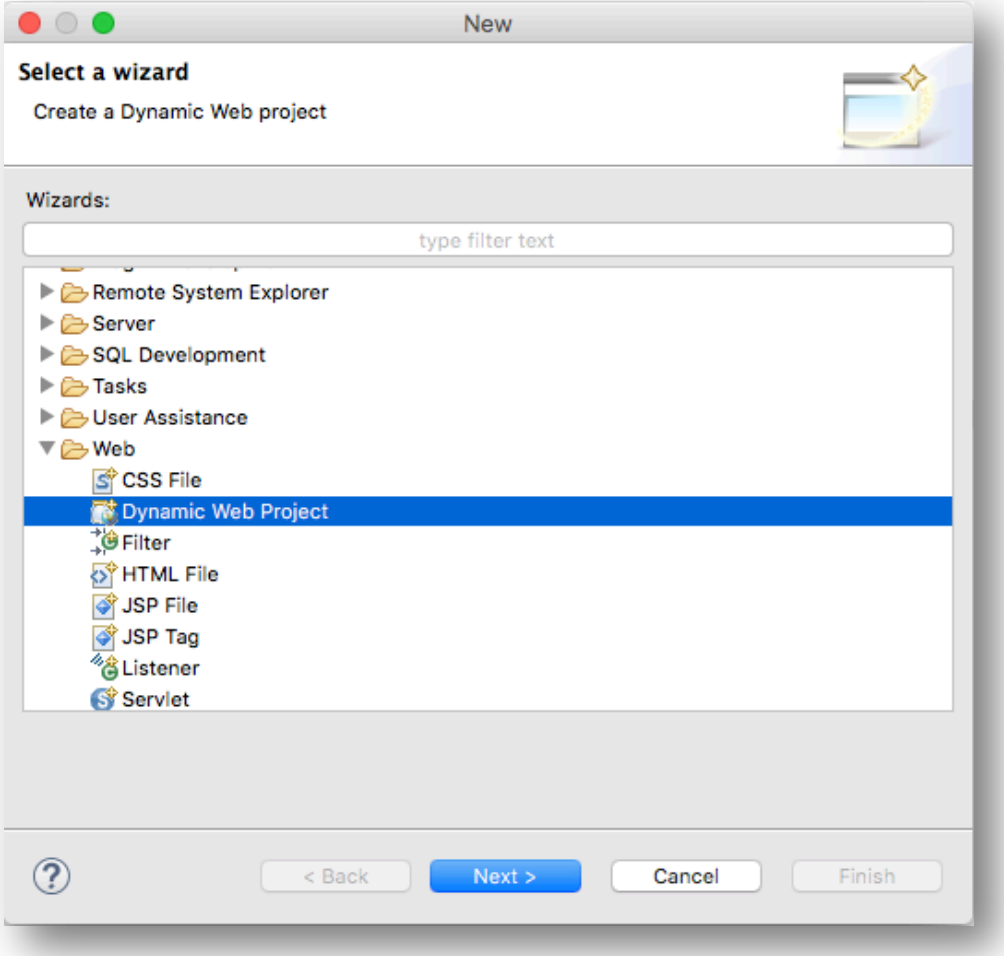
JRE: Workbench default JRE

4. Click “Finish” to save the settings
5. Go to the “Server” tab in Eclipse and start the server. If, it starts then you have successfully configured, tomcat in eclipse



Creating Java Web Services

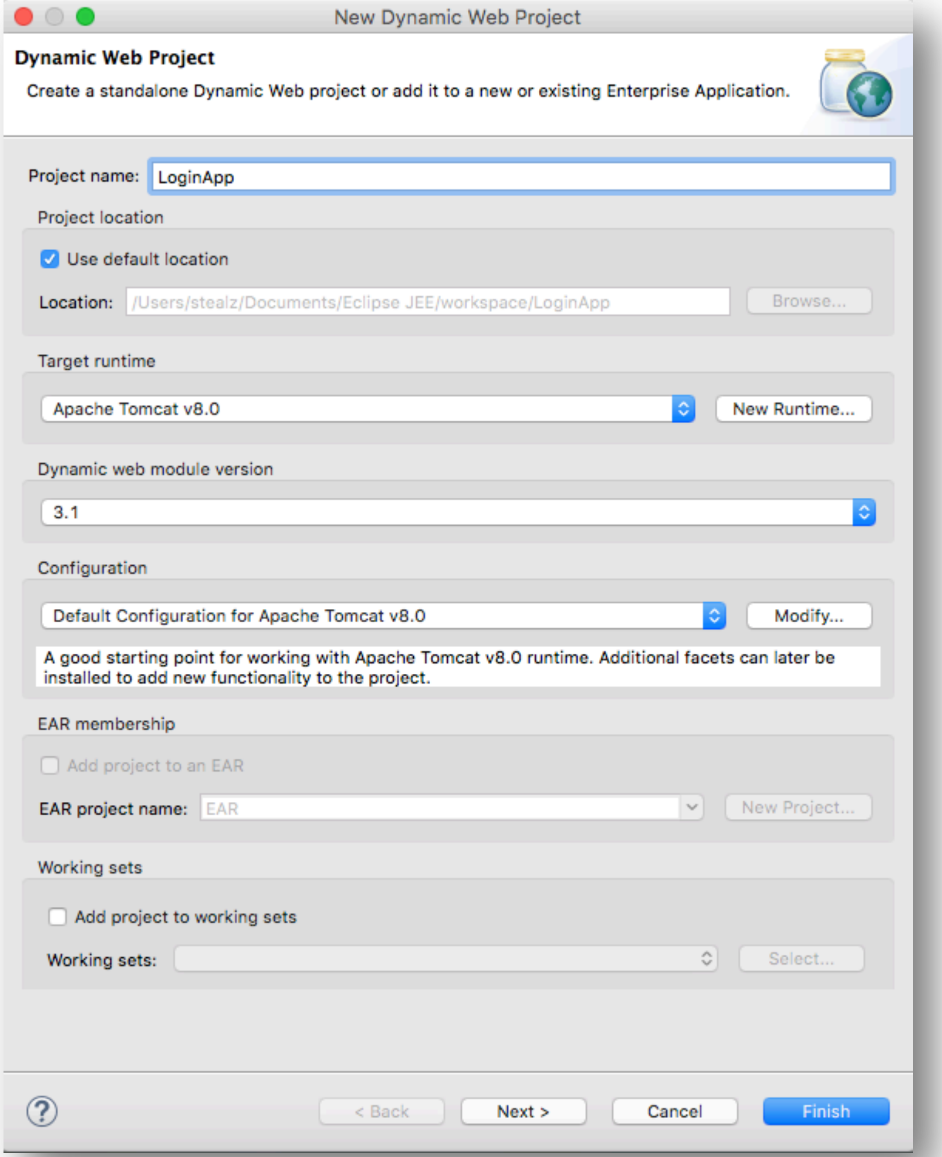
1. Create a Dynamic Web Project



Select a wizard

Create a Dynamic Web project

2. Give the project a Name – here LoginApp and click Finish



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: LoginApp

Project location

☑ Use default location

Location: /Users/stealz/Documents/Eclipse JEE/workspace/LoginApp

Target runtime: Apache Tomcat v8.0

Dynamic web module version: 3.1

Configuration: Default Configuration for Apache Tomcat v8.0

A good starting point for working with Apache Tomcat v8.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

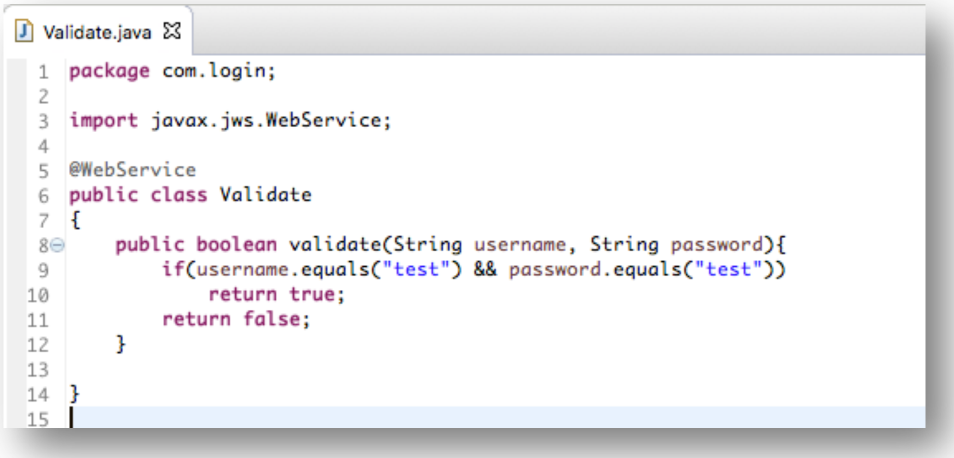
EAR project name: EAR

Working sets

☐ Add project to working sets

Working sets:

3. Create a package inside ‘Java Resources -> src’ and create a class in that package. This class will be the Web Service. @WebService defines it as a Web Service



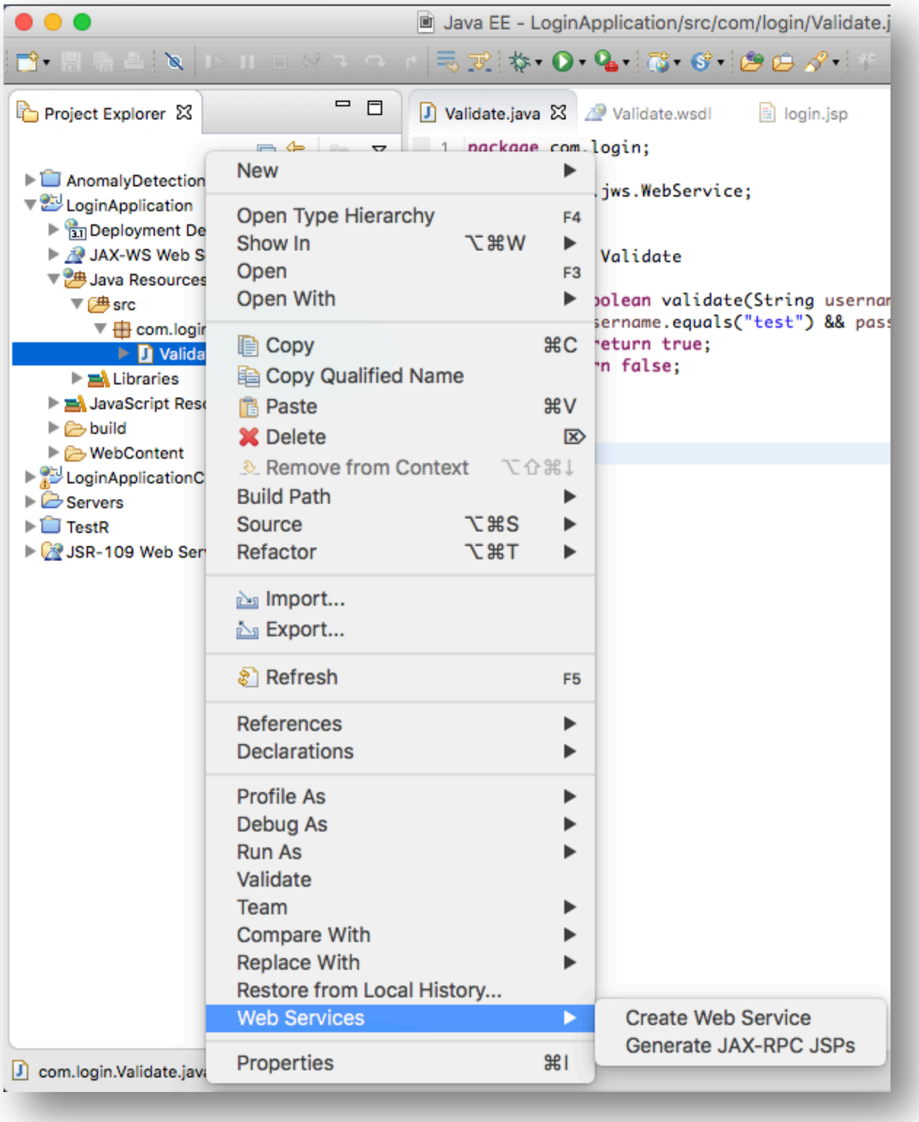
```
package com.login;

import javax.jws.WebService;

@WebService
public class Validate
{
    public boolean validate(String username, String password){
        if(username.equals("test") && password.equals("test"))
            return true;
        return false;
    }

}
```

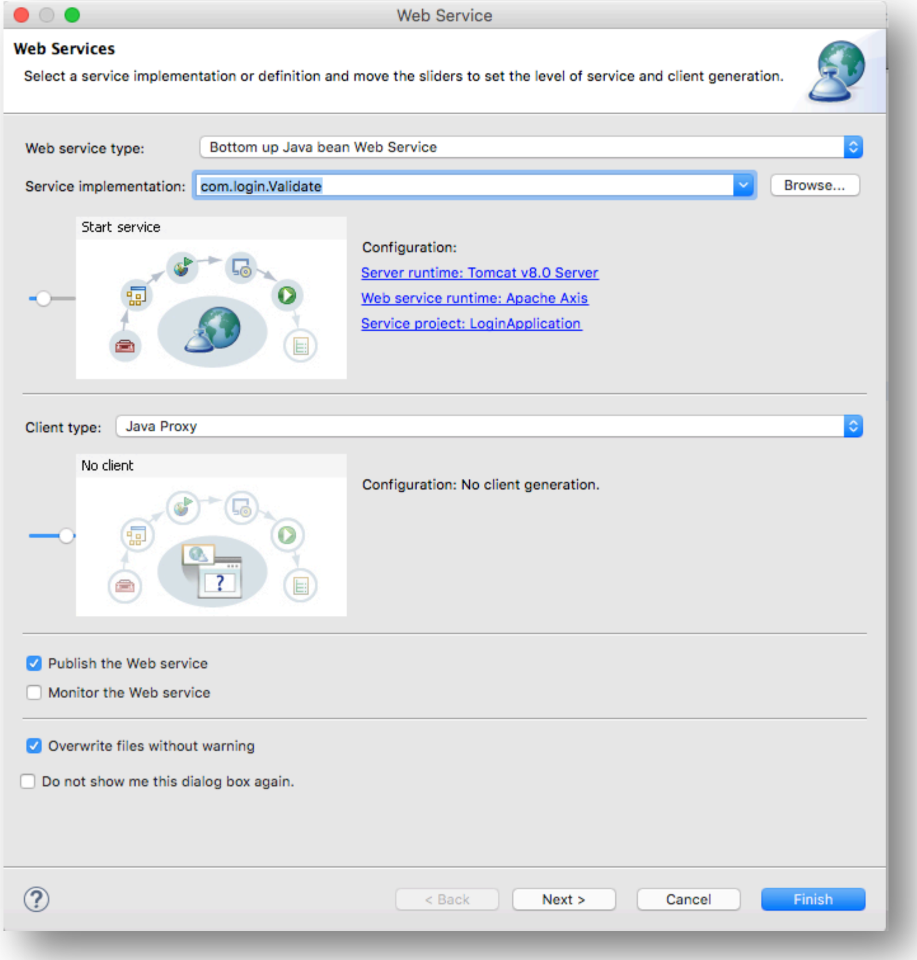
4. Right Click on the file and go to Web Services -> Create web service



```
package com.login;
.jws.WebService;
Validate
oolean validate(String usernar
sername.equals("test") && pass
eturn true;
n false;
```

5. Check the publish button. If you want to test your application, you can drag the slider in Client type to 'test Client'

   a. With no client option



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: com.login.Validate

Start service

Configuration:

Server runtime: Tomcat v8.0 Server

Web service runtime: Apache Axis

Service project: LoginApplication

Client type: Java Proxy

No client

Configuration: No client generation.

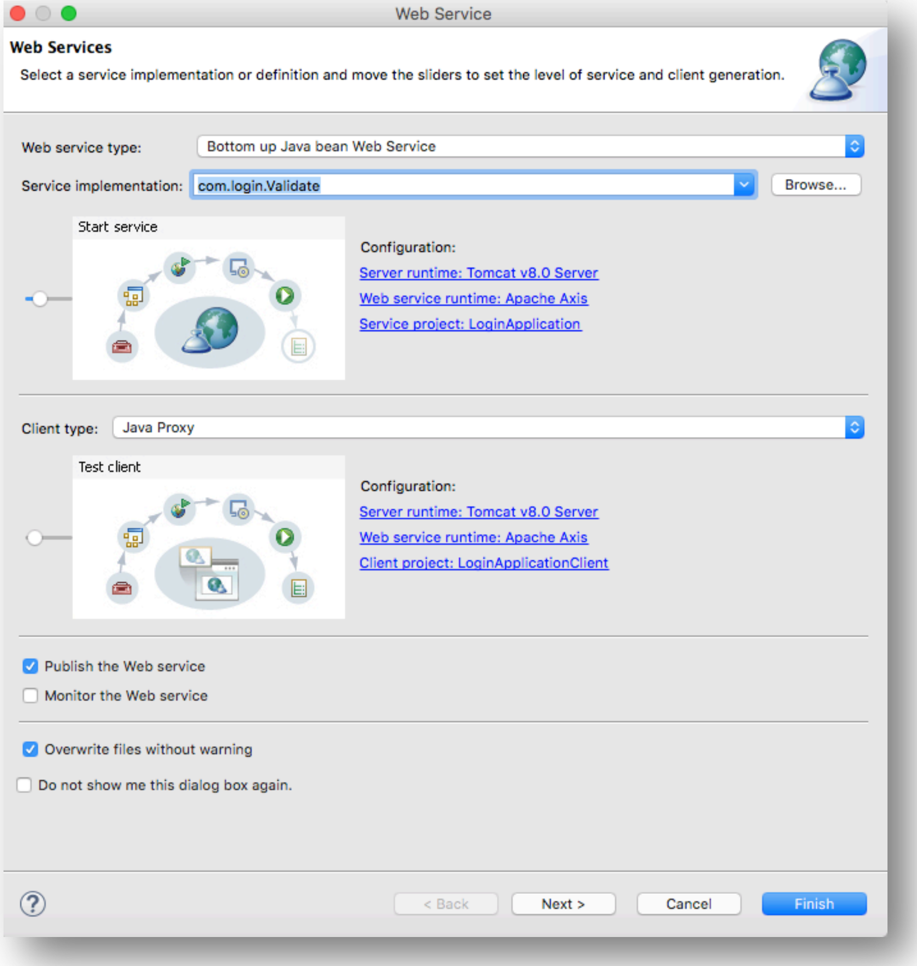
☑ Publish the Web service

□ Monitor the Web service

☑ Overwrite files without warning

□ Do not show me this dialog box again.

b. With Test client option



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: com.login.Validate

Configuration:
Server runtime: Tomcat v8.0 Server
Web service runtime: Apache Axis
Service project: LoginApplication

Client type: Java Proxy

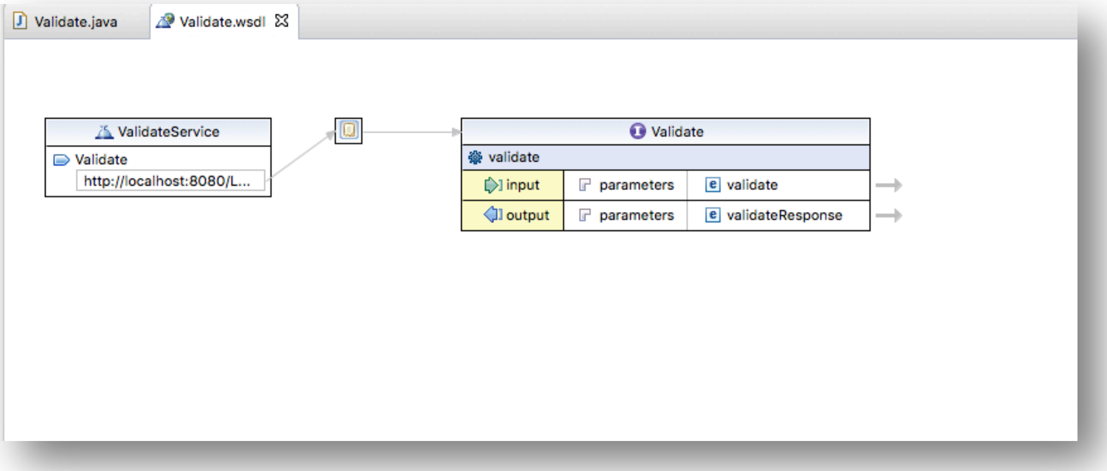
Configuration:
Server runtime: Tomcat v8.0 Server
Web service runtime: Apache Axis
Client project: LoginApplicationClient

☑ Publish the Web service
□ Monitor the Web service

☑ Overwrite files without warning

□ Do not show me this dialog box again.

6. Now click Finish.
7. This will create a wsdl file in ‘WebContent -> wsdl’ folder. Click on the wsdl file to see the link used to connect to the webservice.



The above process will create a Java Web Service.