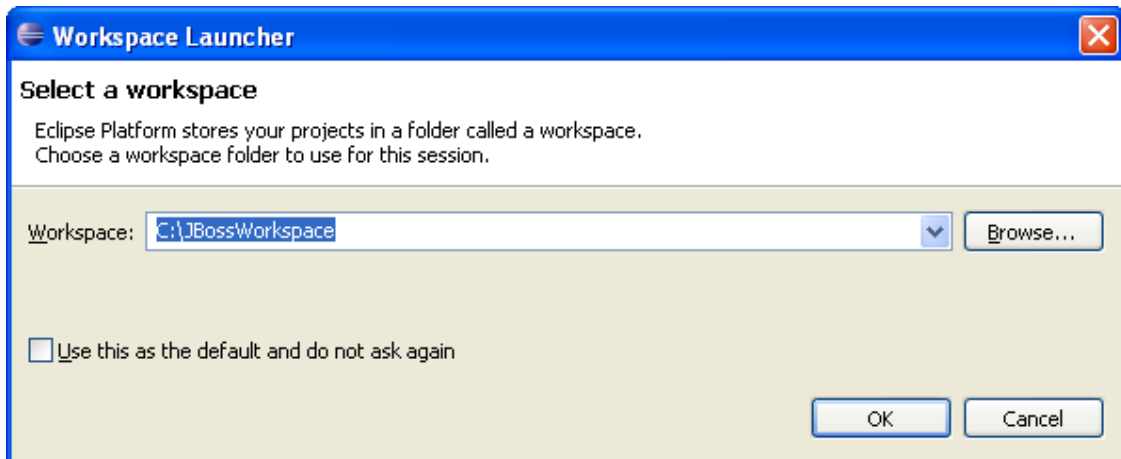


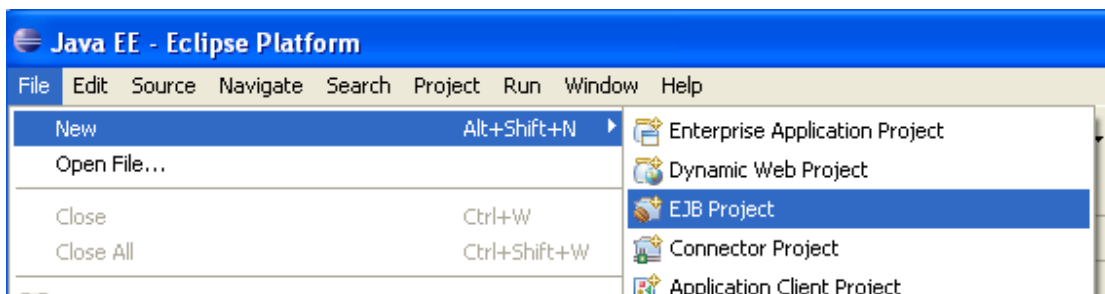
## Simple EJB 3 Application using JBoss and Eclipse.

Start Eclipse.

Open your Workspace or create a new one.



Create a new EJB Project.

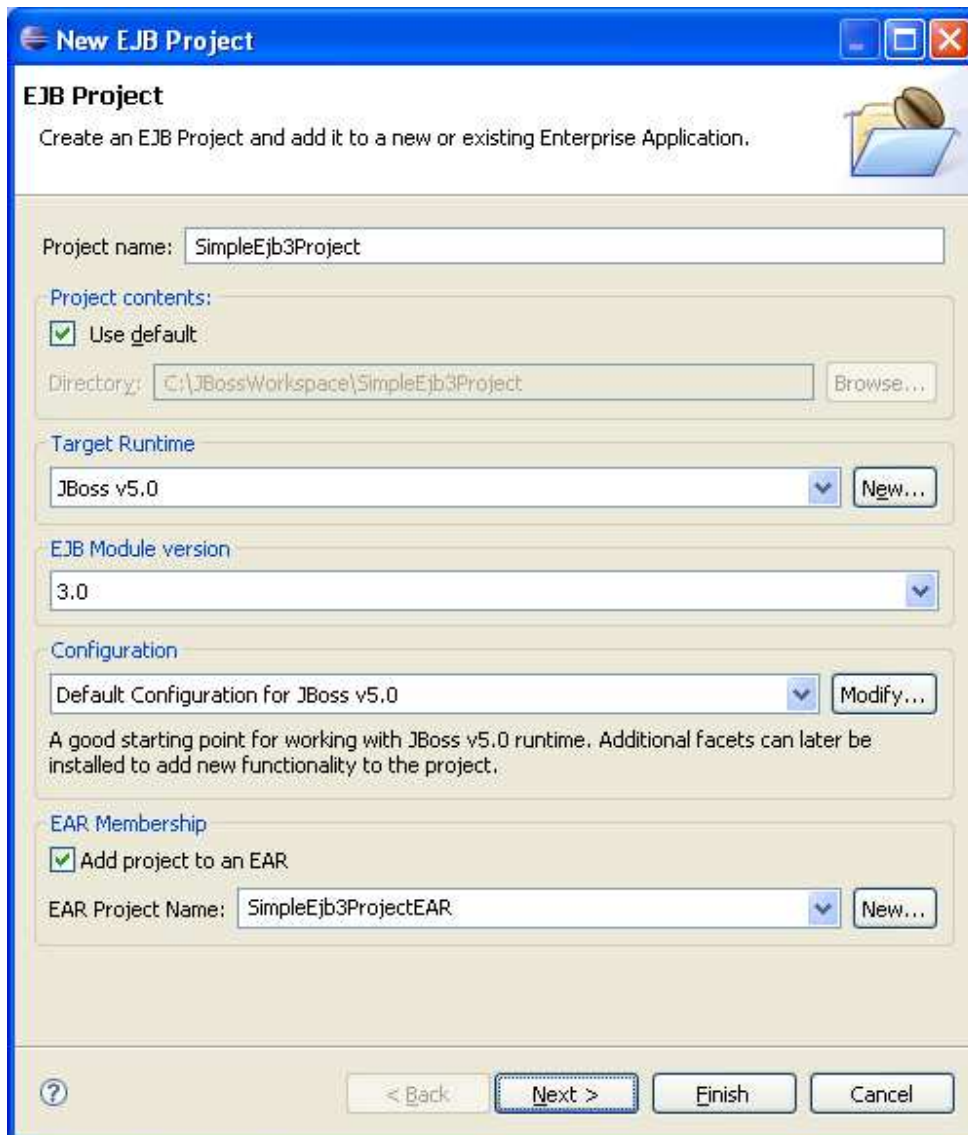


Name the Project **SimpleEjb3Project**.

Make sure the EJB Module version is 3.0.

Check Add project to an EAR. This will create a separate Project for deployment.

Click **Next**.



The image shows a 'New EJB Project' dialog box with a blue title bar. The main area is divided into several sections: 'Project name' with a text field containing 'SimpleEjb3Project'; 'Project contents' with a checked 'Use default' checkbox and a 'Directory' field showing 'C:\JBossWorkspace\SimpleEjb3Project'; 'Target Runtime' with a dropdown menu set to 'JBoss v5.0' and a 'New...' button; 'EJB Module version' with a dropdown menu set to '3.0'; 'Configuration' with a dropdown menu set to 'Default Configuration for JBoss v5.0' and a 'Modify...' button; and 'EAR Membership' with a checked 'Add project to an EAR' checkbox and an 'EAR Project Name' field containing 'SimpleEjb3ProjectEAR'. At the bottom, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is located in the bottom left corner.

**New EJB Project**

**EJB Project**  
Create an EJB Project and add it to a new or existing Enterprise Application.

Project name: SimpleEjb3Project

Project contents:  
☒ Use default  
Directory: C:\JBossWorkspace\SimpleEjb3Project Browse...

Target Runtime  
JBoss v5.0 New...

EJB Module version  
3.0

Configuration  
Default Configuration for JBoss v5.0 Modify...  
A good starting point for working with JBoss v5.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR Membership  
☒ Add project to an EAR  
EAR Project Name: SimpleEjb3ProjectEAR New...

< Back Next > Finish Cancel

We will not use a Client Project for this simple example.

We will not generate a Deployment Descriptor.

Click **Finish**.

**New EJB Project**

**EJB Module**  
Configure ejb module settings.

Source Folder:  
ejbModule

☒ Create an EJB Client JAR module to hold the client interfaces and classes.

Name:  
SimpleEjb3ProjectClient

Client JAR URI:  
SimpleEjb3ProjectClient.jar

☐ Generate deployment descriptor

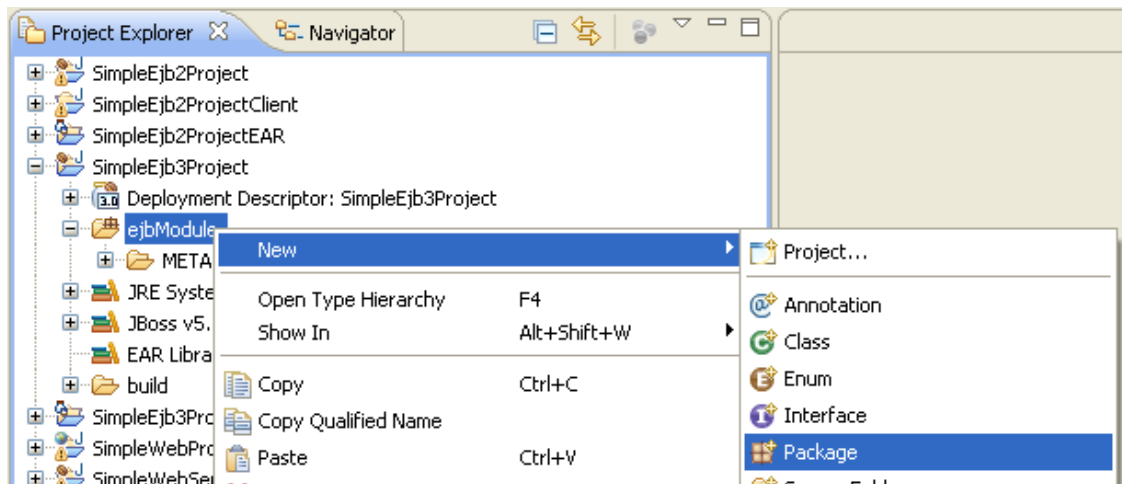
? < Back Next > Finish Cancel

We will create a very simple application with 1 EJB Stateless Session Bean and create a standalone client for a test.

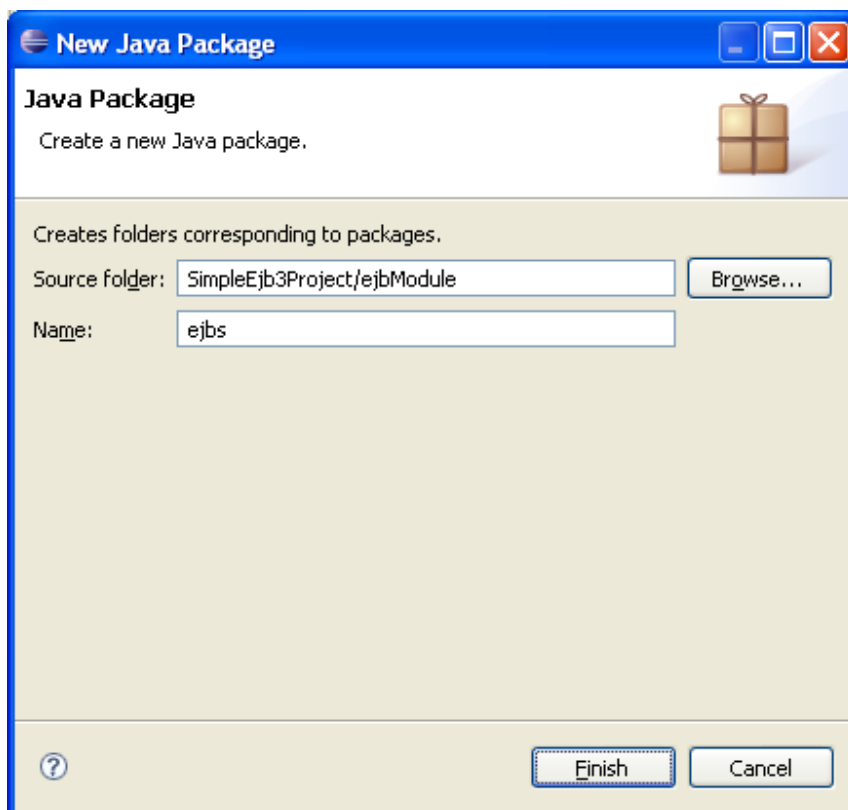
The basic steps are:

- In the EJB project
  - Create a new package named **ejbs**
  - Create the EJB Bean
  - Create the Test Client

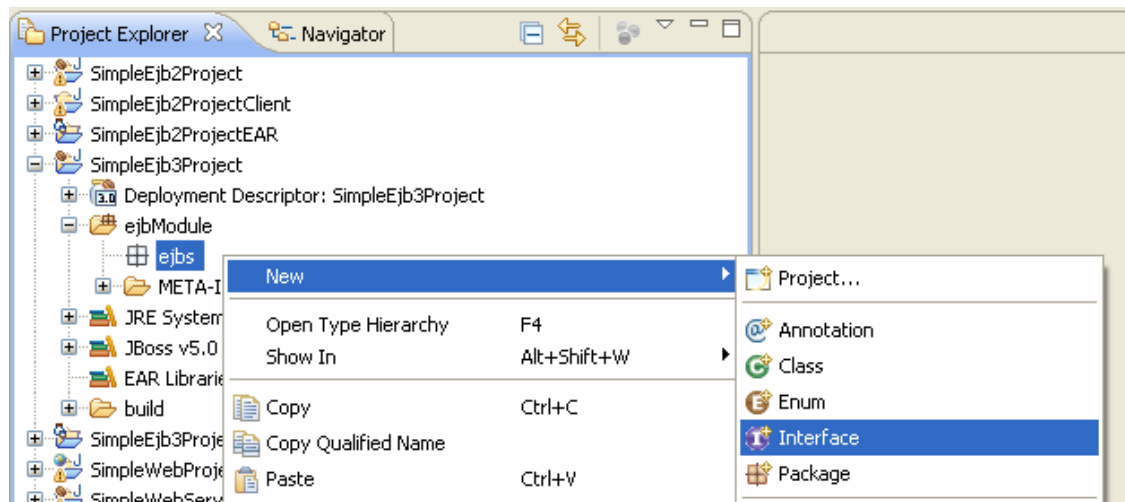
Create a new Package.



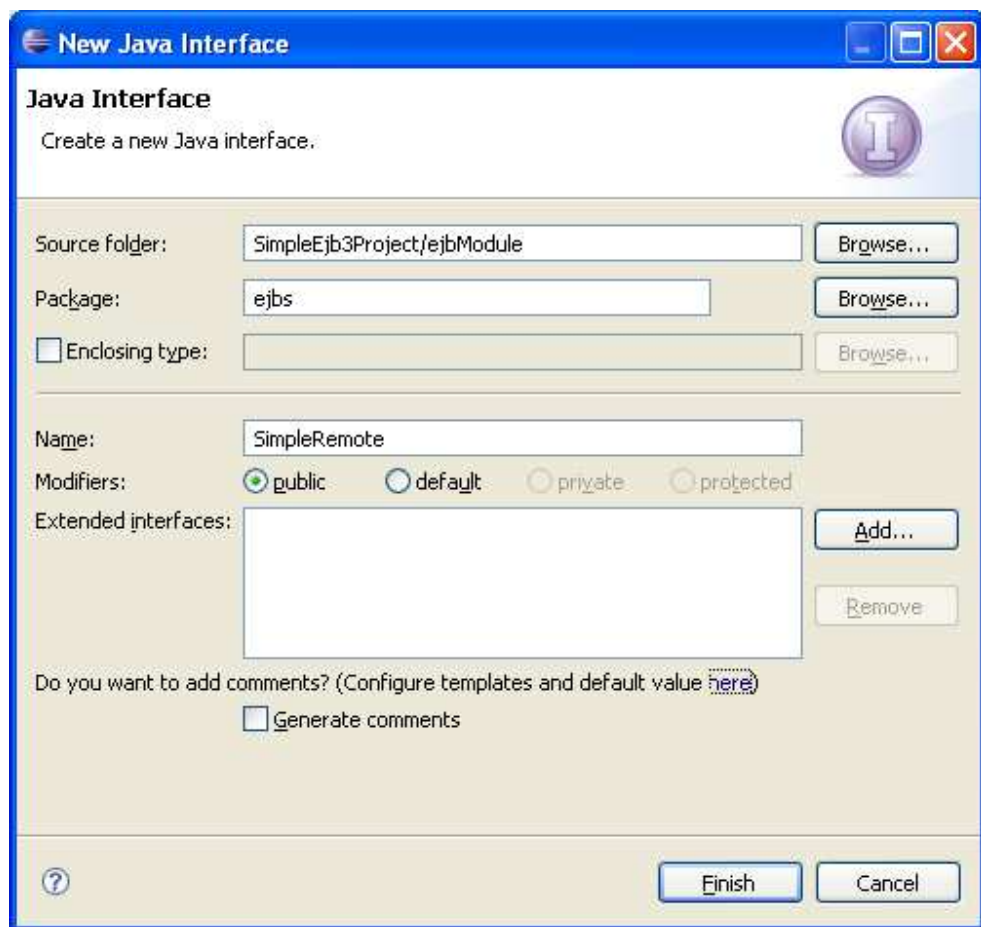
Name the Package **ejbs**.



Create a new Interface.



Name it **SimpleRemote**.



Add the following.

```
package ejbs;

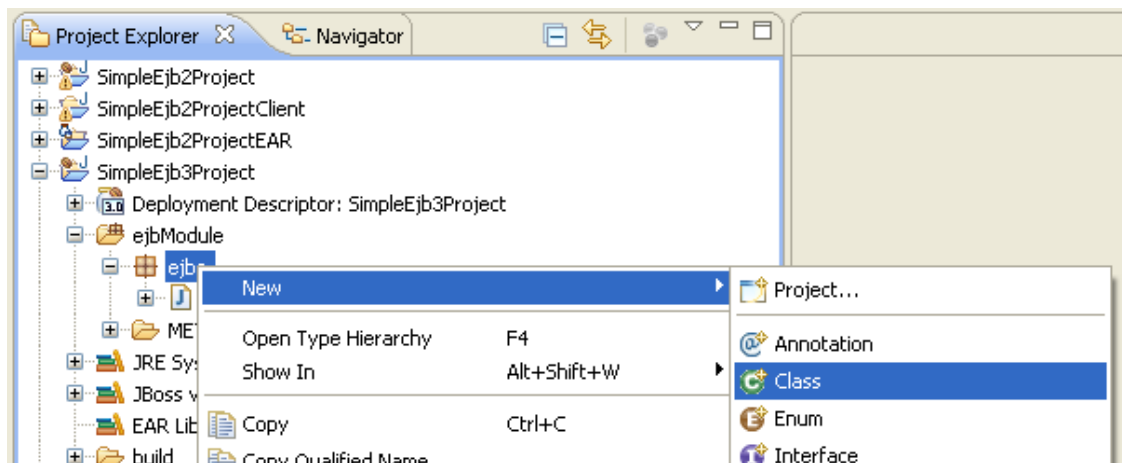
import javax.ejb.Remote;

@Remote
public interface SimpleRemote {

    public String getMessage();

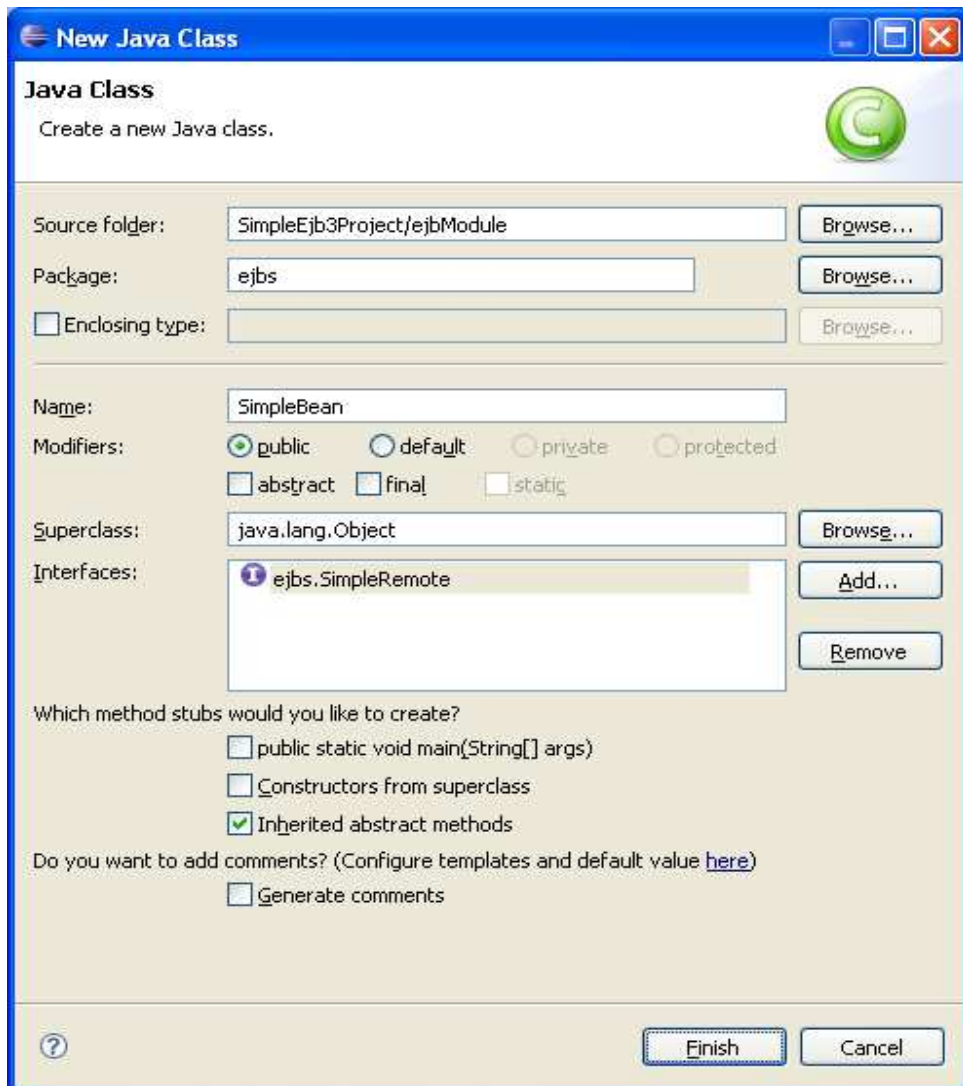
}
```

Create a new Class.



Name the Class **SimpleBean**.

Implement **SimpleRemote**.



Add the following.

```
package ejbs;

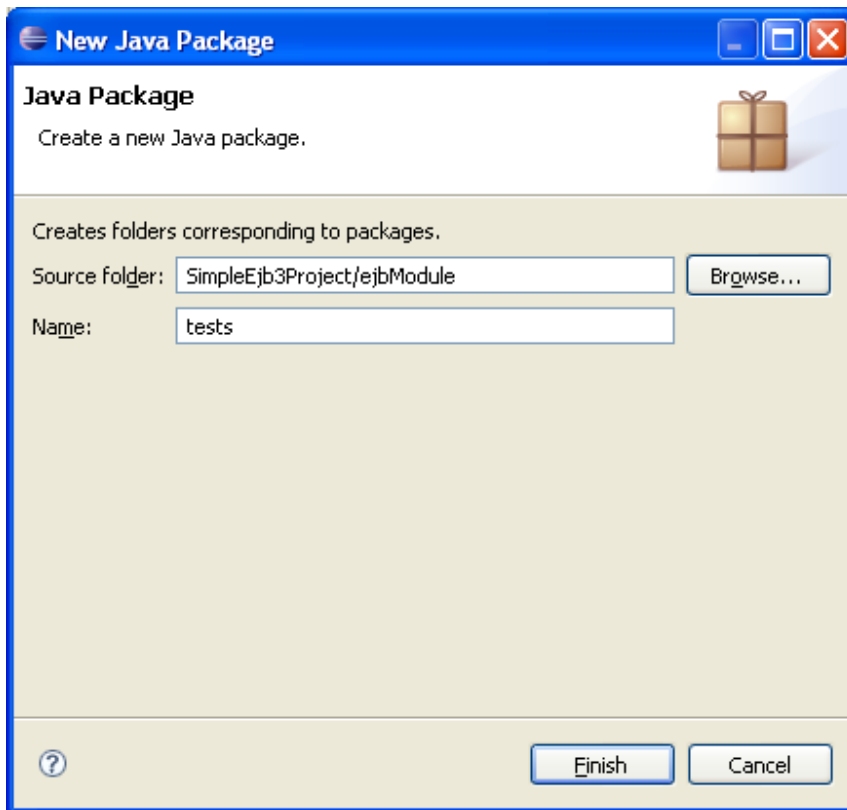
import javax.ejb.Stateless;

@Stateless(name="SimpleBean", mappedName="ejb/SimpleBean")
public class SimpleBean implements SimpleRemote {

    public String getMessage() {
        return "Hello from EJB 3.0";
    }

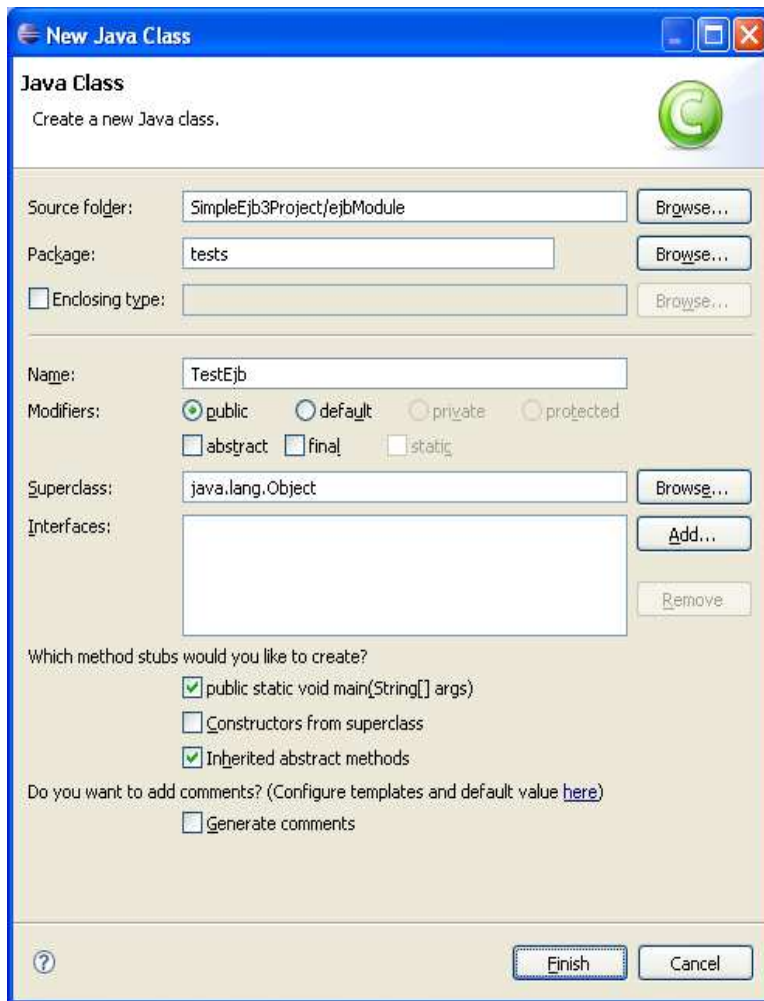
}
```

Create a new Package named **tests**.



Create a new Class to test the EJB named **TestEjb**.





Add the following.

```
package tests;

import javax.naming.Context;
import javax.naming.InitialContext;

import ejbs.SimpleRemote;

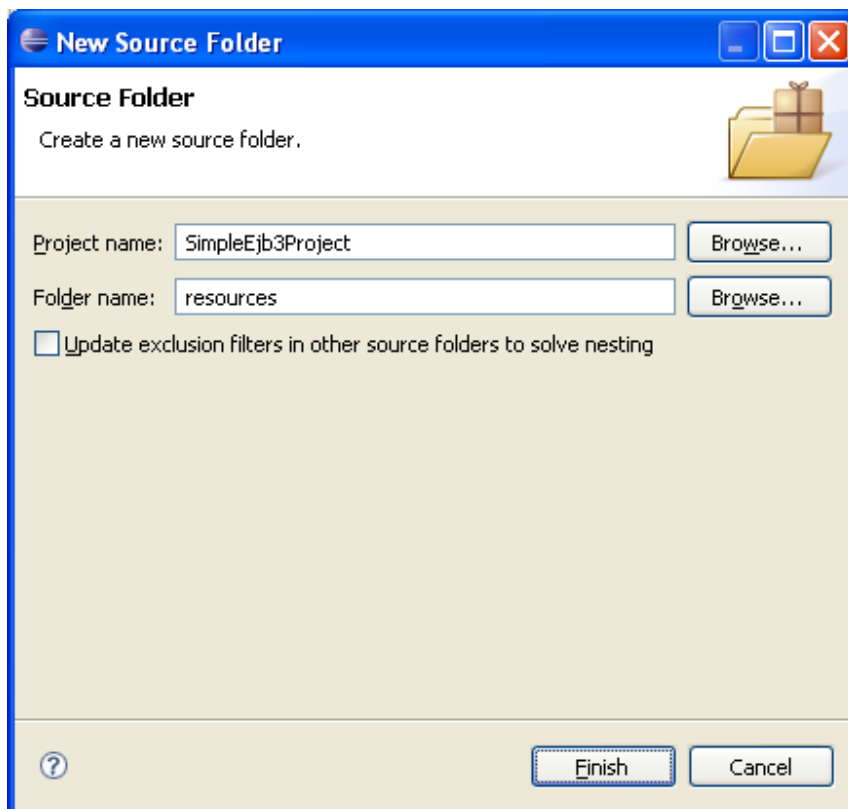
public class TestEjb {

    /**
     * @param args
     */
    public static void main(String[] args)
        throws Exception {

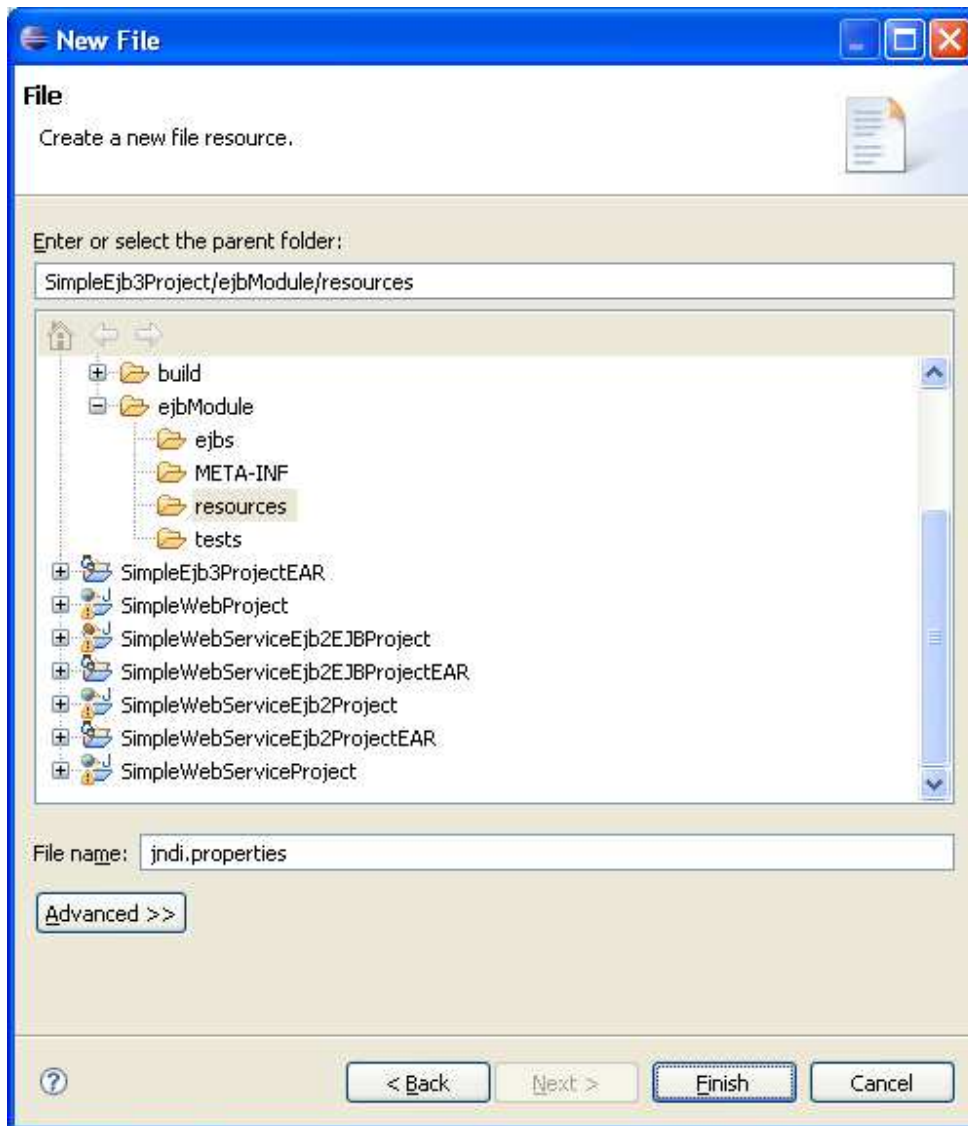
        String jndiName = "ejb/SimpleBean";
        Context context = new InitialContext();
        System.out.println("Looking up JNDI Name " + jndiName);
    }
}
```

```
        Object object = context.lookup(jndiName);  
        System.out.println("Lookup returned " + object);  
  
        SimpleRemote remote = (SimpleRemote) object;  
        System.out.println(remote.getMessage());  
    }  
}
```

Create a new Source Folder named **resources**.



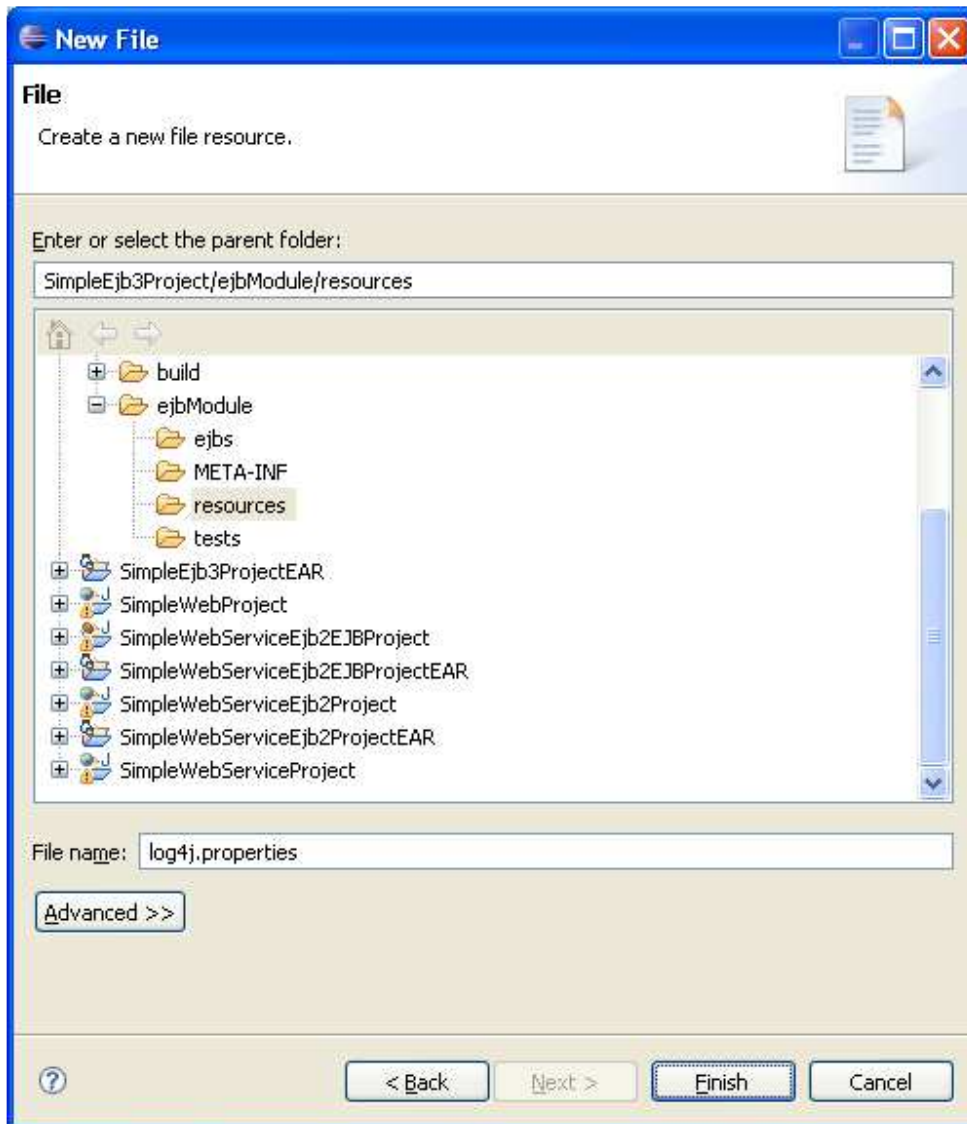
Create a new file named **jndi.properties**.



Add the following.

```
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
java.naming.provider.url=localhost:1099
```

Create another file named **log4j.properties**.

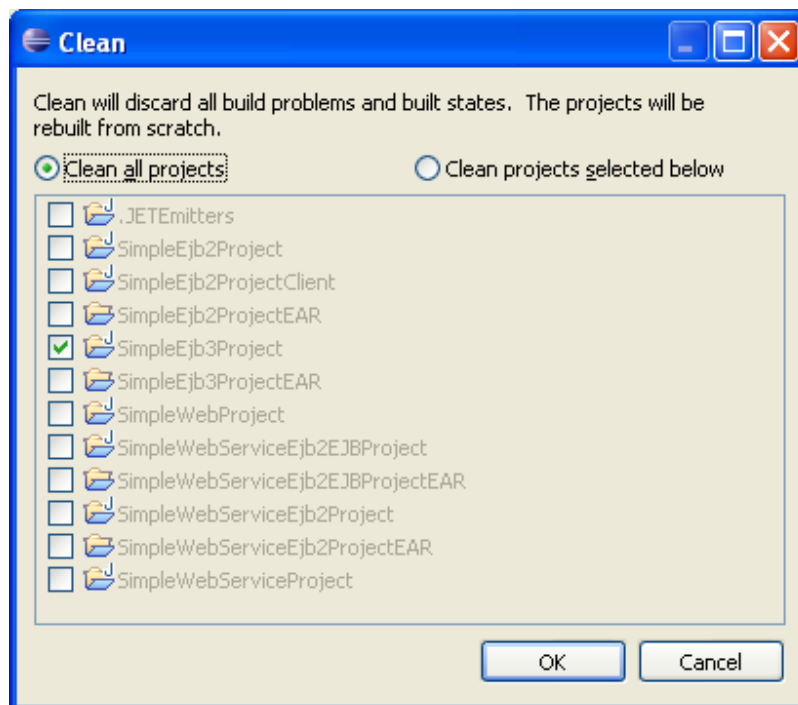
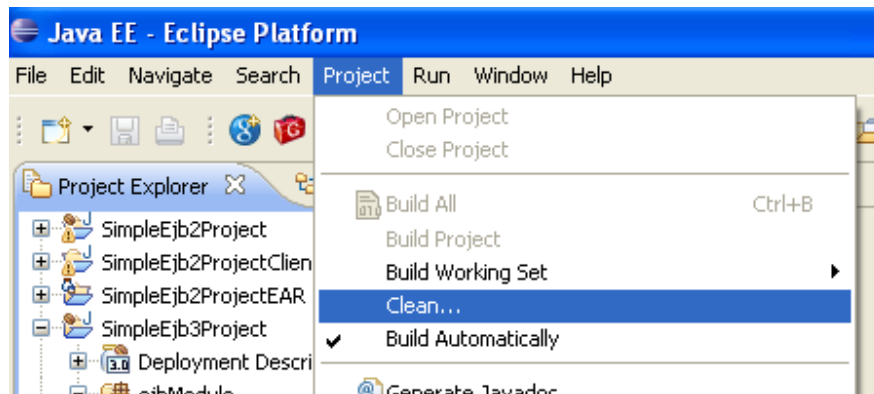


Add the following.

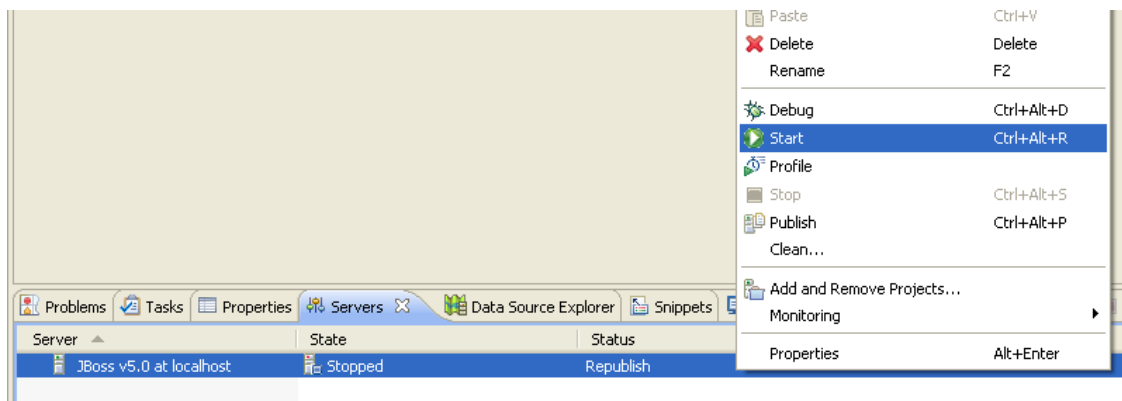
```
# Set root category priority to INFO and its only appender to CONSOLE.
log4j.rootCategory=INFO, CONSOLE

# CONSOLE is set to be a ConsoleAppender using a PatternLayout.
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.Threshold=INFO
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=- %m%n
```

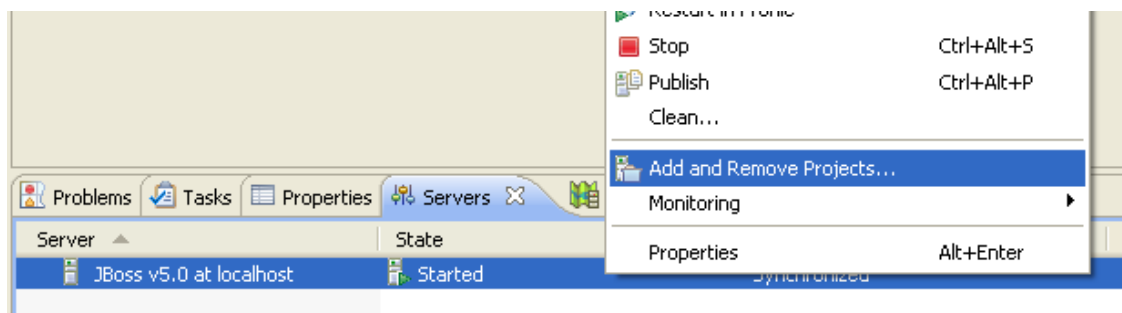
Clean and Build the Project.

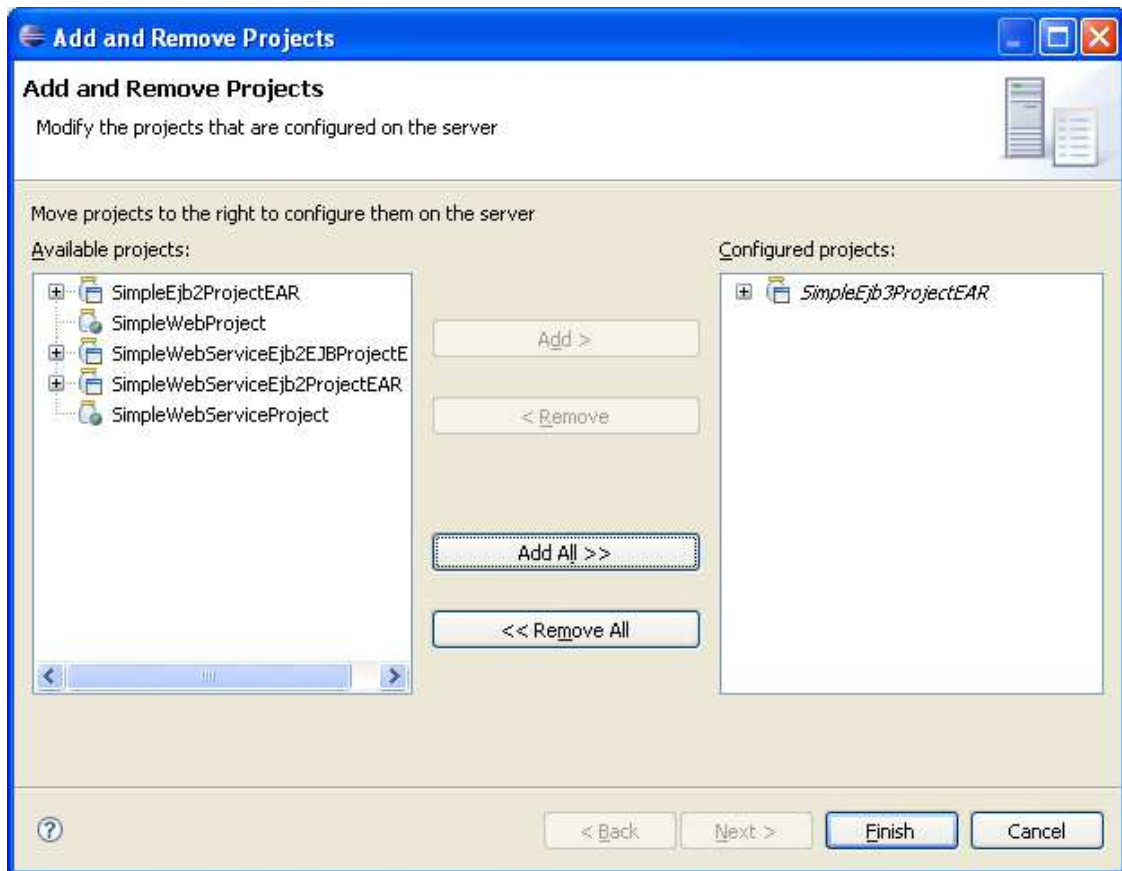


Start the Server.

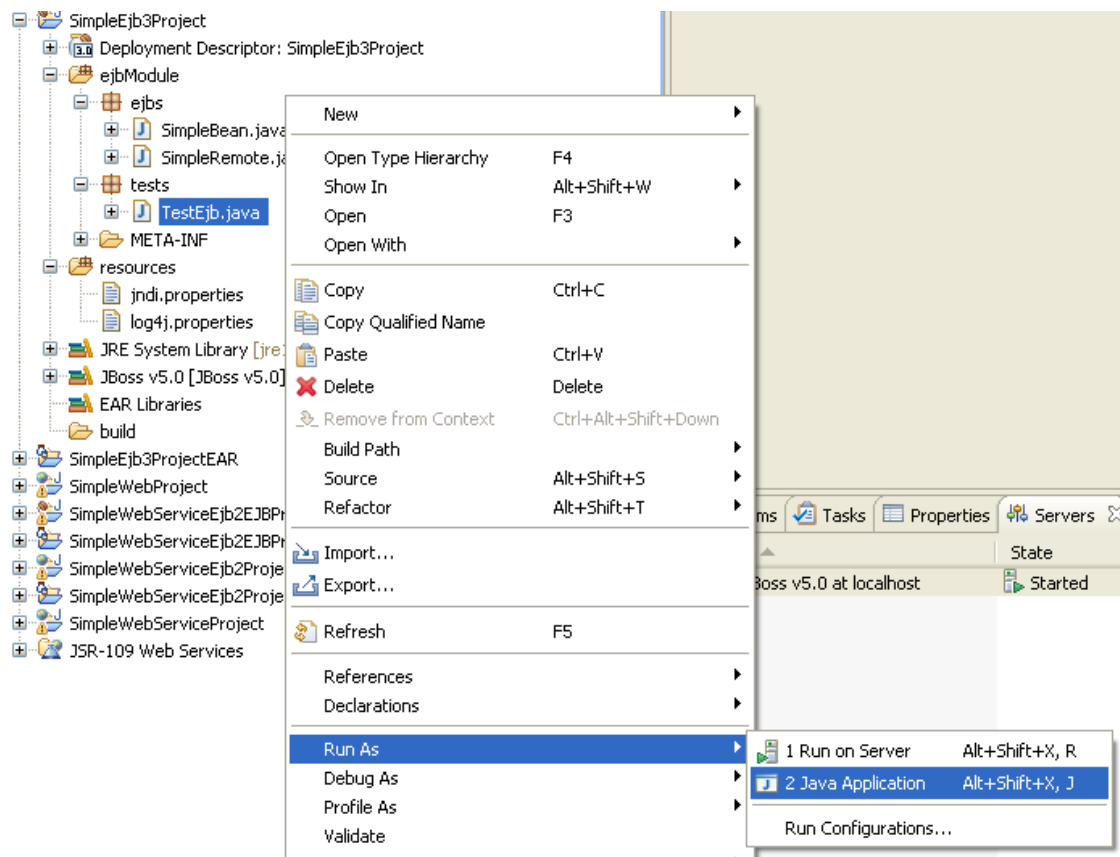


Deploy the application.





Run the Test.



You should see the following.

