EJB

enterprise java beans

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

Enterprise **J**ava **B**eans (EJB) is a development architecture for building highly scalable and robust enterprise level applications to be deployed on J2EE compliant Application Server such as JBOSS, Web Logic etc.

EJB 3.0 is being a great shift from EJB 2.0 and makes development of EJB based applications quite easy.

This tutorial is developed to provide a comprehensive understanding about the EJB concepts helpful to create and deploy an enterprise level application up and running.

Audience

This tutorial is designed for Software Professionals as well as for all those who are willing to learn EJB Programming. This tutorial will give you a great understanding about EJB Programming concepts in simple and easy steps.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of Java programming language, text editor, and execution of programs etc. Because we are going to develop enterprise-based applications using EJB, it will be good, if you have understanding on other technologies like Database Servers, Application Servers.

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness, or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

1. EJB – OVERVIEW

EJB stands for **E**nterprise **J**ava **B**eans. EJB is an essential part of a J2EE platform. J2EE platform has component based architecture to provide multi-tiered, distributed and highly transactional features to enterprise level applications.

EJB provides an architecture to develop and deploy component based enterprise applications considering robustness, high scalability, and high performance. An EJB application can be deployed on any of the application server compliant with the J2EE 1.3 standard specification.

We'll be discussing EJB 3.0 in detail in this tutorial.

Types

EJB is primarily divided into three categories; following table lists their names with brief descriptions:

Type	Description
Session Bean	Session bean stores data of a particular user for a single session. It can be **stateful** or **stateless**. It is less resource intensive as compared to entity bean. Session bean gets destroyed as soon as user session terminates.
Entity Bean	**Entity beans** represent persistent data storage. User data can be saved to database via entity beans and later on can be retrieved from the database in the entity bean.
Message Driven Bean	**Message driven beans** are used in context of JMS (Java Messaging Service). Message Driven Beans can consumes JMS messages from external entities and act accordingly.

Benefits

Following are the important benefits of EJB:

- Simplified development of large-scale enterprise level application.
- Application Server/EJB container provides most of the system level services like transaction handling, logging, load balancing, persistence mechanism, exception handling, and so on. Developer has to focus only on business logic of the application.

- EJB container manages life cycle of EJB instances, thus developer needs not to worry about when to create/delete EJB objects.

2. EJB – ENVIRONMENT SETUP

EJB is a framework for Java, so the very first requirement is to have a **J**ava **D**evelopment **Kit** (JDK) installed in your machine.

System Requirement

JDK	1.5 or above.
Memory	no minimum requirement.
Disk Space	no minimum requirement.
Operating System	no minimum requirement.

Step 1 - Verify Java Installation in Your System

Now open console and execute the following **java** command.

OS	Task	Command
Windows	Open Command Console	c:\> java -version
Linux	Open Command Terminal	$ java -version
Mac	Open Terminal	machine:~ joseph$ java -version

Let's verify the output for all the operating systems:

OS	Output
Windows	java version "1.6.0_21" Java(TM) SE Runtime Environment (build 1.6.0_21-b11) Java HotSpot(TM) 64-Bit Server VM (build 23.21-b01, mixed mode)
Linux	java version "1.6.0_21" Java(TM) SE Runtime Environment (build 1.6.0_21-b11) Java HotSpot(TM) 64-Bit Server VM (build 23.21-b01, mixed mode)
Mac	java version "1.6.0_21" Java(TM) SE Runtime Environment (build 1.6.0_21-b11) Java HotSpot(TM) 64-Bit Server VM (build 23.21-b01, mixed mode)

If you do not have Java installed, install the Java Software Development Kit (SDK) from http://www.oracle.com/technetwork/java/javase/downloads/index.html.

We are assuming that Java 1.6.0_21 as installed version for this tutorial.

Step 2 – Set JAVA Environment

Set the **JAVA_HOME** environment variable to point the base directory location where Java is installed on your system. For example,

OS	Output
Windows	Set the environment variable JAVA_HOME to C:\Program Files\Java\jdk1.6.0_21
Linux	export JAVA_HOME=/usr/local/java-current
Mac	export JAVA_HOME=/Library/Java/Home

Append Java compiler location to System Path.

OS	Output
Windows	Append the string ;C:\Program Files\Java\jdk1.6.0_21\bin to the end of the system variable, Path.
Linux	export PATH=$PATH:$JAVA_HOME/bin/
Mac	not required

Verify Java Installation using **java -version** command explained above.

Step 3 – Download and Install NetBeans IDE

Download the latest version of NetBeans IDE from https://netbeans.org/downloads/index.html. At the time of writing this tutorial, I downloaded *Netbeans 7.3,* which comes bundled with JDK 1.7 using the following link http://www.oracle.com/technetwork/java/javase/downloads/index.html

OS	Installer name
Windows	Netbeans 7.3
Linux	Netbeans 7.3
Mac	Netbeans 7.3

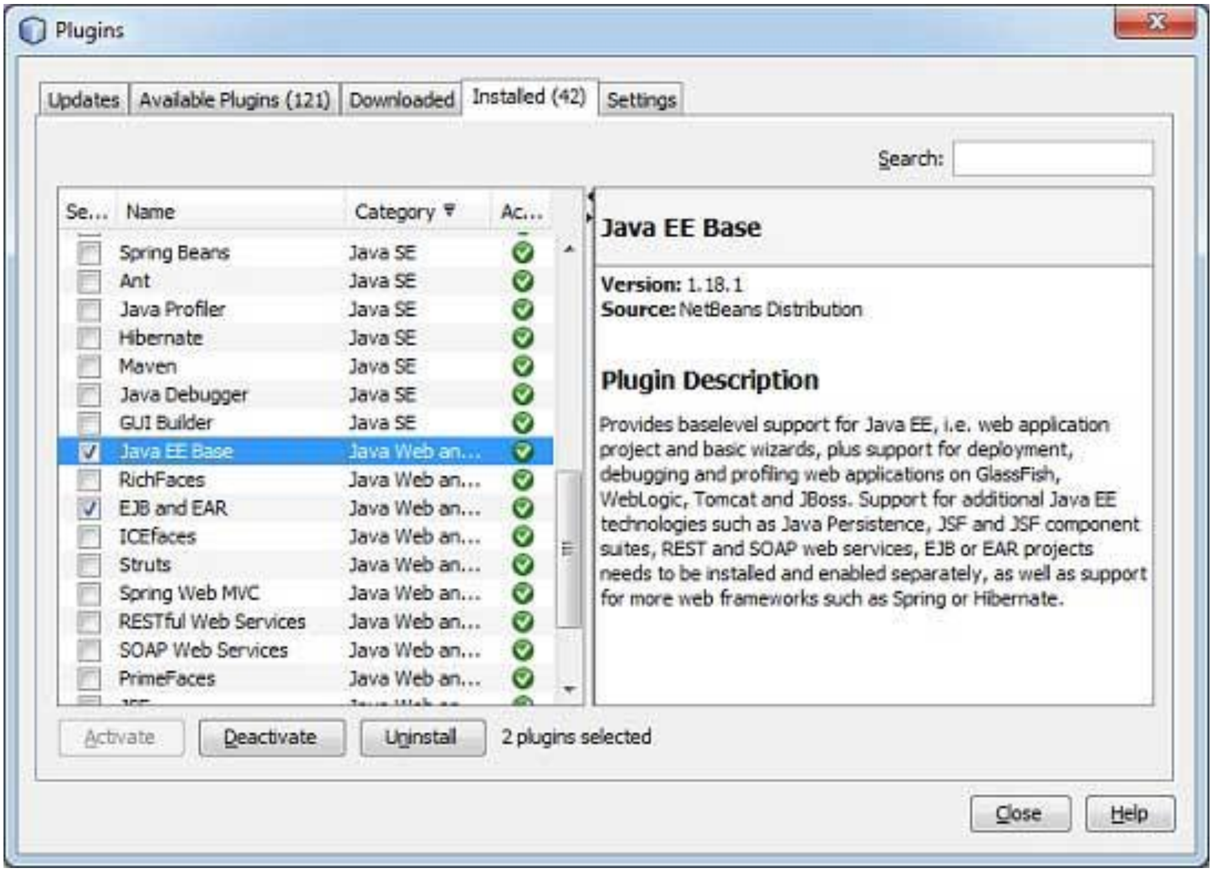
Step 4 – Set up JBoss Application Server

You can download the latest version of JBoss Server from http://www.jboss.org/jbossas/downloads/. Download the archive as per the platform. Extract the Jboss to any location on your machine.

OS	File name
Windows	jboss-5.1.0.GA-jdk6.zip
Linux	jboss-5.1.0.GA-src.tar.gz
Mac	jboss-5.1.0.GA-src.tar.gz

Step 5 – Configure JEE Plugins to Netbeans

Open Plugin window using Tools > Plugins. Open "Available Plugin" tab and select "Java EE Base" and "EJB and EAR" under "Java Web and EE" category. Click install button. Netbeans will download and install the respective plugins. Verify plugins installation using "Installed" tab (as shown in the image given below).



Se...	Name	Category	Ac...
☐	Spring Beans	Java SE	✓
☐	Ant	Java SE	✓
☐	Java Profiler	Java SE	✓
☐	Hibernate	Java SE	✓
☐	Maven	Java SE	✓
☐	Java Debugger	Java SE	✓
☐	GUI Builder	Java SE	✓
☑	Java EE Base	Java Web an...	✓
☐	RichFaces	Java Web an...	✓
☑	EJB and EAR	Java Web an...	✓
☐	ICEfaces	Java Web an...	✓
☐	Struts	Java Web an...	✓
☐	Spring Web MVC	Java Web an...	✓
☐	RESTful Web Services	Java Web an...	✓
☐	SOAP Web Services	Java Web an...	✓
☐	PrimeFaces	Java Web an...	✓

Java EE Base

Version: 1.18.1
Source: NetBeans Distribution

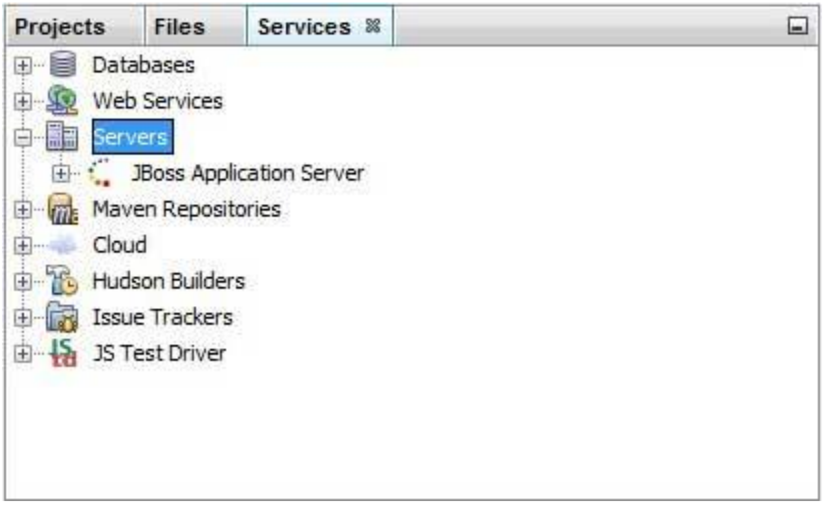
Plugin Description

Provides baselevel support for Java EE, i.e. web application project and basic wizards, plus support for deployment, debugging and profiling web applications on GlassFish, WebLogic, Tomcat and JBoss. Support for additional Java EE technologies such as Java Persistence, JSF and JSF component suites, REST and SOAP web services, EJB or EAR projects needs to be installed and enabled separately, as well as support for more web frameworks such as Spring or Hibernate.

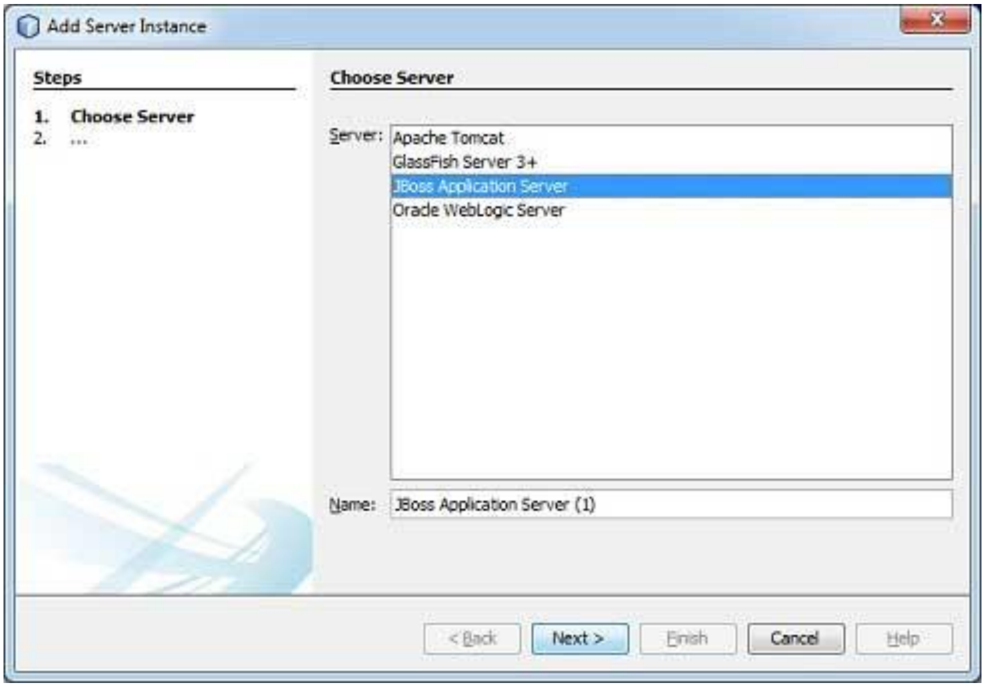
2 plugins selected

Step 6 – Configure JBoss Server in Netbeans

Go to Services tab and right click on servers to add a new server.

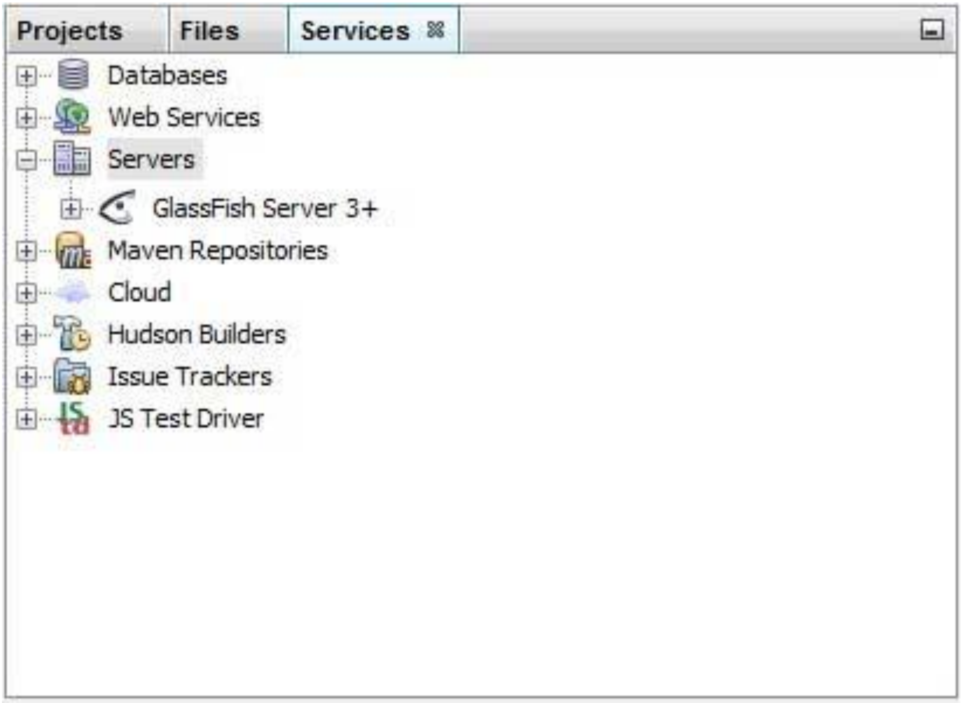


Add Server Instance wizard will open. Select JBoss and in next step enter the relevant details to configure server in netbeans.



Name: JBoss Application Server (1)

Once everything is configured, you will see the following screen.



Step 7 – Install Database Server (PostGreSql)

Download latest version of PostGreSql database server from http://www.postgresql.org/download/. At the time of writing this tutorial, I downloaded *PostGreSql 9.2*

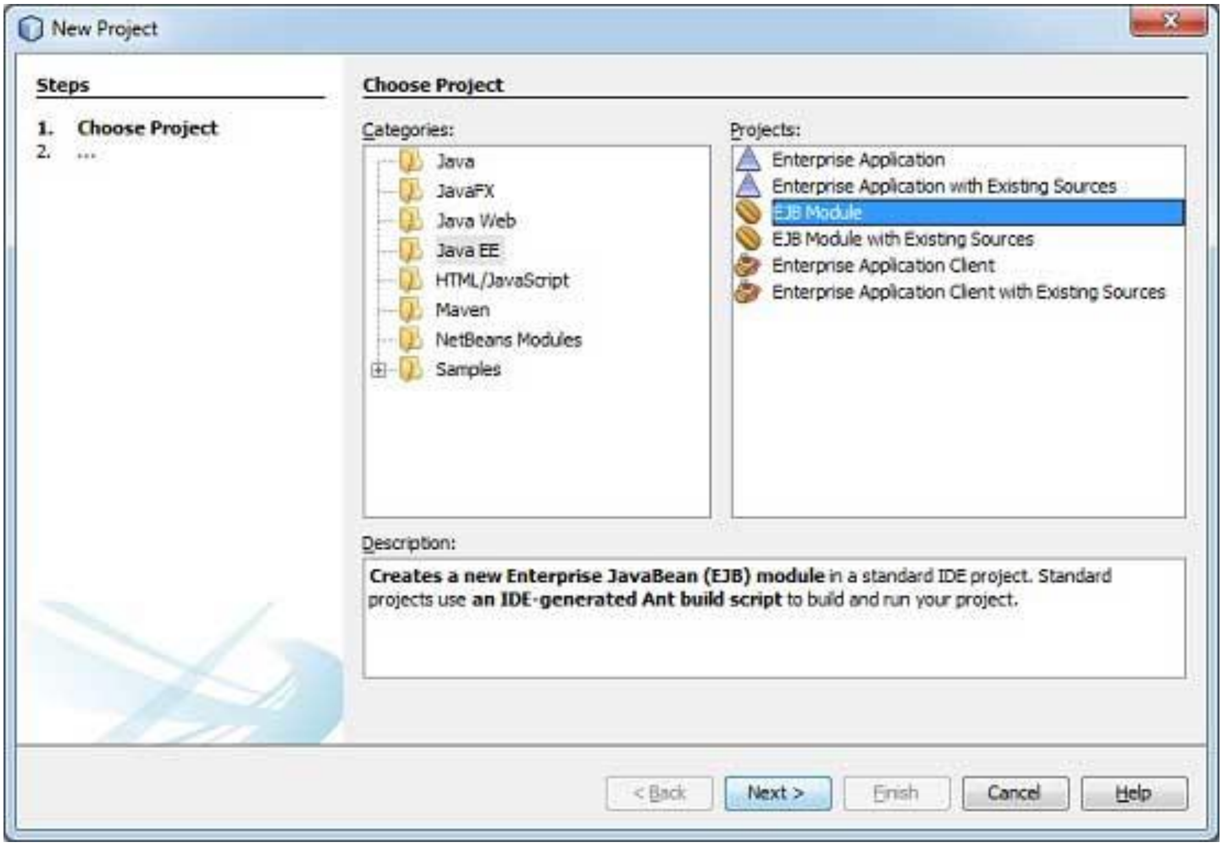
OS	Installer name
Windows	PostGreSql 9.2
Linux	PostGreSql 9.2
Mac	PostGreSql 9.2

3. EJB – CREATE APPLICATION

To create a simple EJB module, we will use NetBeans, "New project" wizard. In the example given below, We will create an EJB module project named Component.

Create Project

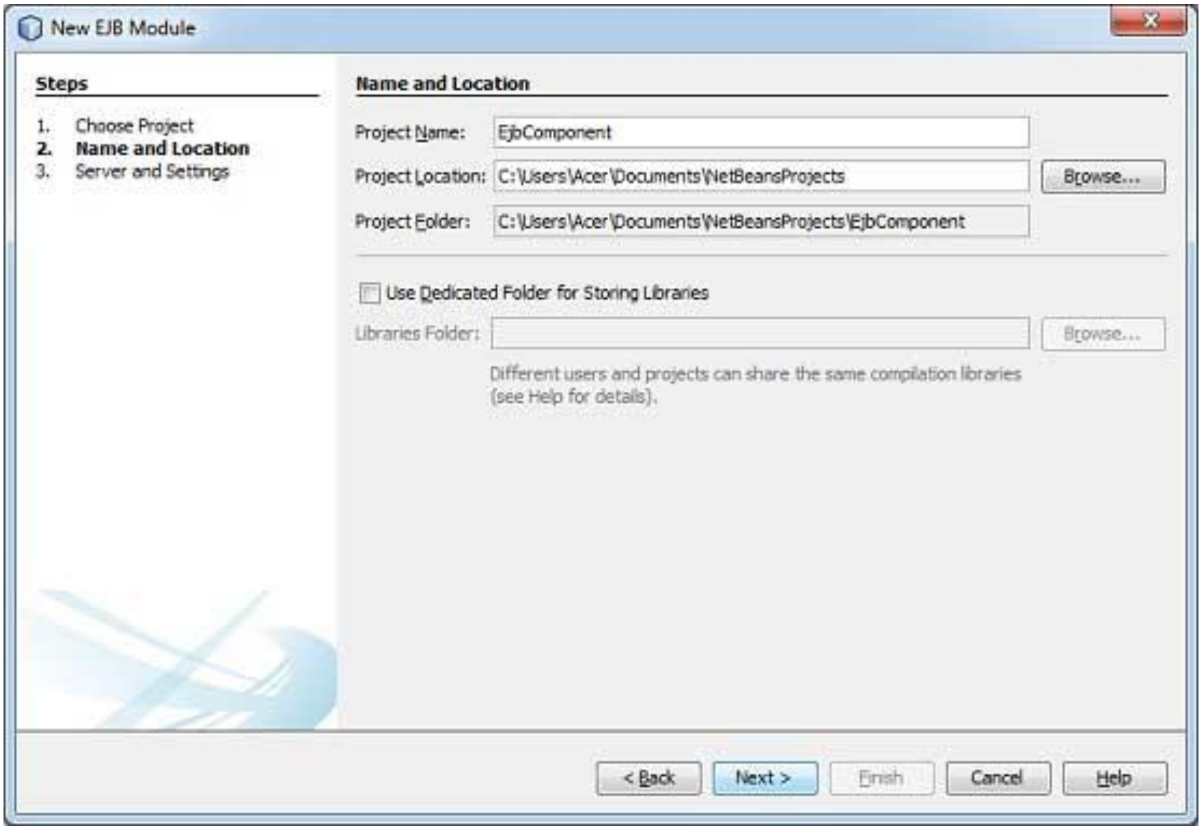
In NetBeans IDE, select **File > New Project >**. You will see the following screen.



Description:

Creates a new Enterprise JavaBean (EJB) module in a standard IDE project. Standard projects use **an IDE-generated Ant build script** to build and run your project.

Select project type under category **Java EE**, Project type as **EJB Module**. Click **Next >** button. You will see the following screen.



New EJB Module

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings

Name and Location

Project Name: EjbComponent

Project Location: C:\Users\Acer\Documents\NetBeansProjects

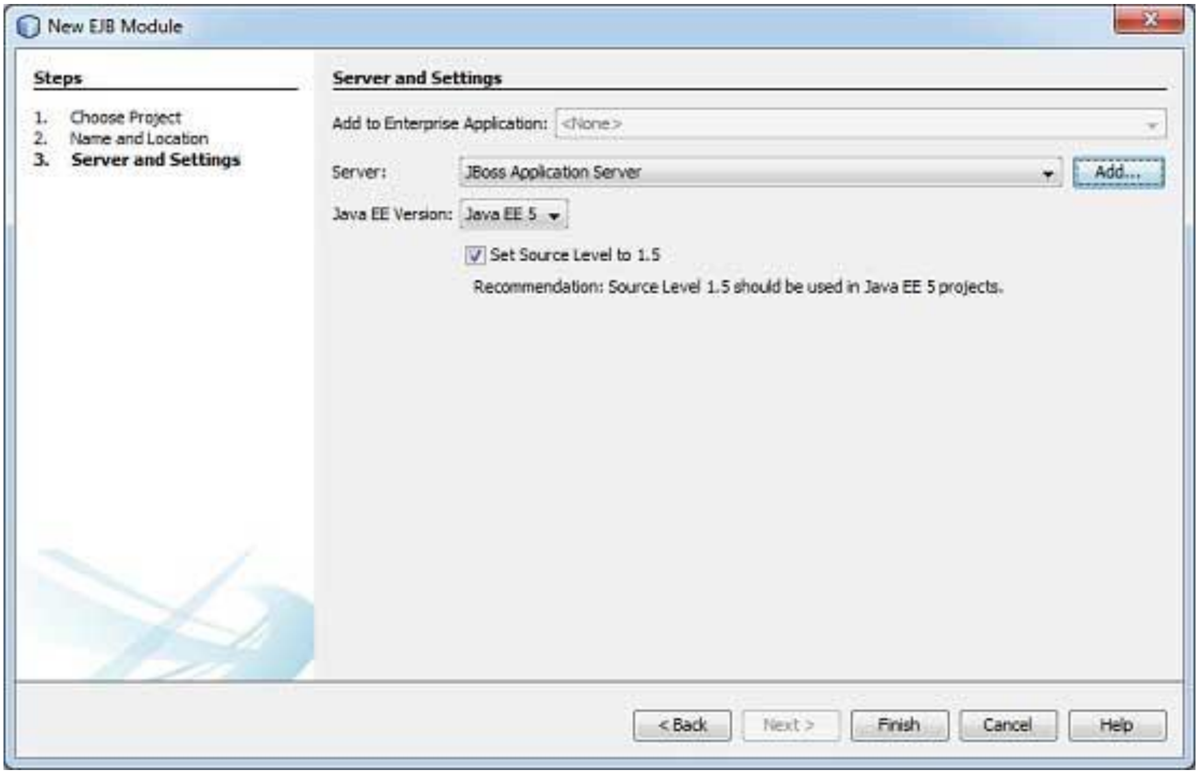
Project Folder: C:\Users\Acer\Documents\NetBeansProjects\EjbComponent

□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Enter project name and location. Click **Next >** button. You will see the following screen.



New EJB Module

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**

Server and Settings

Add to Enterprise Application: <None>

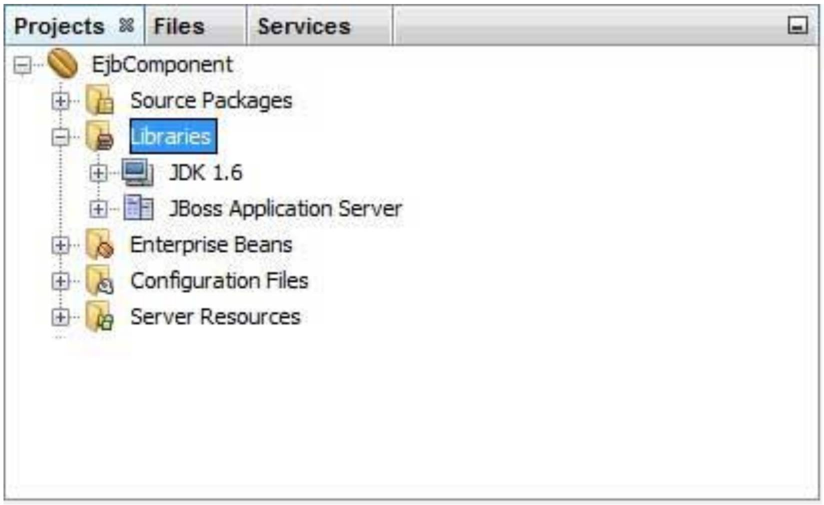
Server: JBoss Application Server

Java EE Version: Java EE 5

☑ Set Source Level to 1.5

Recommendation: Source Level 1.5 should be used in Java EE 5 projects.

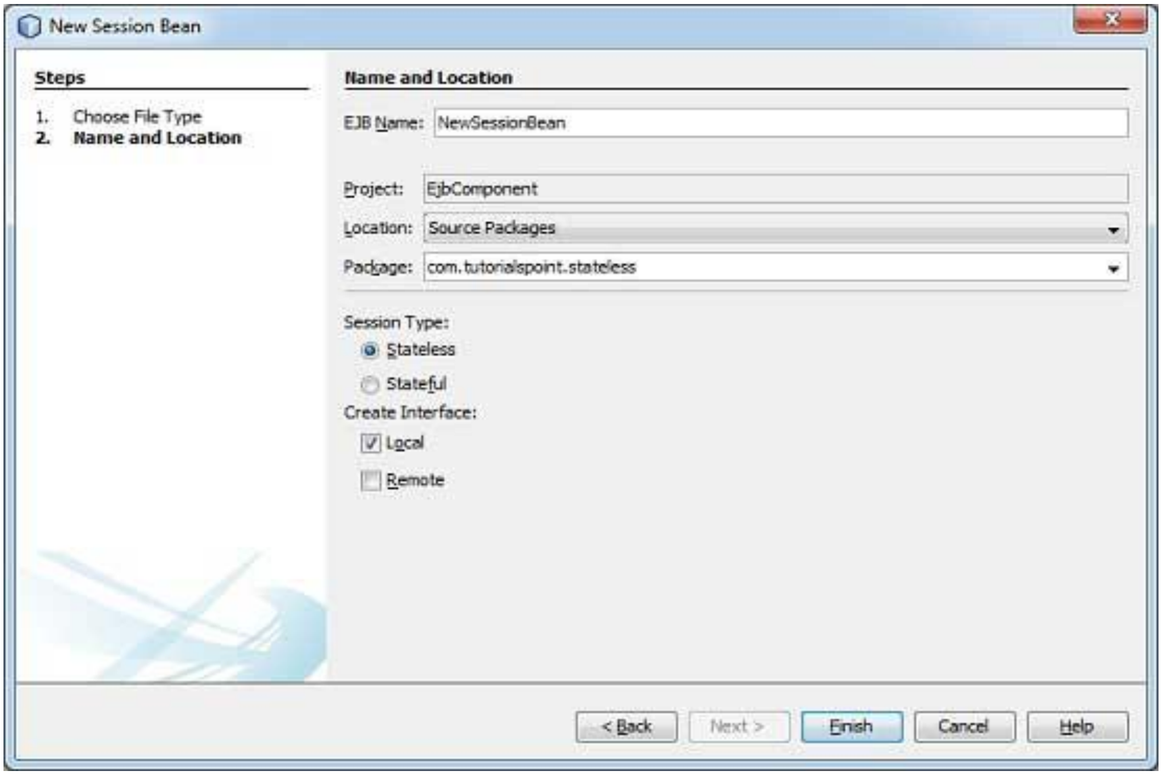
Select Server as **JBoss Application Server**. Click **Finish** button. You will see the following project created by NetBeans.



Create a Sample EJB

To create a simple EJB, we will use NetBeans "New" wizard. In the example given below, We will create a stateless EJB class named librarySessionBean under EjbComponent project.

Select project EjbComponent in project explorer window and right click on it. Select, **New > Session Bean**. You will see the **New Session Bean** wizard.



1. Choose File Type
2. **Name and Location**

EJB Name: NewSessionBean

Project: EjbComponent

Location: Source Packages

Package: com.tutorialspoint.stateless

Session Type:

● Stateless

○ Stateful

Create Interface:

☑ Local

☐ Remote

Enter session bean name and package name. Click **Finish** button. You will see the following EJB classes created by NetBeans.

- **LibrarySessionBean** - stateless session bean
- **LibrarySessionBeanLocal** - local interface for session bean

I am changing local interface to remote interface as we are going to access our EJB in a console based application. Remote/Local interface is used to expose business methods that an EJB has to implement.

LibrarySessionBeanLocal is renamed to LibrarySessionBeanRemote and LibrarySessionBean implements LibrarySessionBeanRemote interface.

LibrarySessionBeanRemote

```
package com.tutorialspoint.stateless;


import java.util.List;
import javax.ejb.Remote;


@Remote
```

```
public interface LibrarySessionBeanRemote {

    void addBook(String bookName);

    List getBooks();

}
```

LibrarySessionBean

```
package com.tutorialspoint.stateless;

import java.util.ArrayList;
import java.util.List;
import javax.ejb.Stateless;

@Stateless
public class LibrarySessionBean implements LibrarySessionBeanRemote {

    List<String> bookShelf;

    public LibrarySessionBean(){
       bookShelf = new ArrayList<String>();
    }

    public void addBook(String bookName) {
        bookShelf.add(bookName);
    }

    public List<String> getBooks() {
        return bookShelf;
    }
}
```

Build the Project

- Select EjbComponent project in Project Explorer window
- Right click on it to open context menu.
- Select clean and build.

You will see the following output in NetBeans console output.

```
ant -f C:\\EJB\\EjbComponent clean dist
init:
undeploy-clean:
deps-clean:
Deleting directory C:\EJB\EjbComponent\build
Deleting directory C:\EJB\EjbComponent\dist
clean:
init:
deps-jar:
Created dir: C:\EJB\EjbComponent\build\classes
Copying 3 files to C:\EJB\EjbComponent\build\classes\META-INF
Created dir: C:\EJB\EjbComponent\build\empty
Created dir: C:\EJB\EjbComponent\build\generated-sources\ap-source-output
Compiling 2 source files to C:\EJB\EjbComponent\build\classes
warning: [options] bootstrap class path not set in conjunction with -source 1.6
Note: C:\EJB\EjbComponent\src\java\com\tutorialspoint\stateless
\LibraryPersistentBean.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning
compile:
library-inclusion-in-archive:
Created dir: C:\EJB\EjbComponent\dist
Building jar: C:\EJB\EjbComponent\dist\EjbComponent.jar
dist:
BUILD SUCCESSFUL (total time: 3 seconds)
```

Start the Application Server

- Select JBoss application server under Servers in Services window.
- Right click on it to open context menu.
- Select start.

You will see the following output in NetBeans, output under JBoss Application Server.

```
Calling C:\jboss-5.1.0.GA\bin\run.conf.bat
=========================================================================


  JBoss Bootstrap Environment


  JBOSS_HOME: C:\jboss-5.1.0.GA


  JAVA: C:\Program Files (x86)\Java\jdk1.6.0_21\bin\java


  JAVA_OPTS: -Dprogram.name=run.bat -Xms128m -Xmx512m -server


  CLASSPATH: C:\jboss-5.1.0.GA\bin\run.jar


=========================================================================


16:25:50,062 INFO  [ServerImpl] Starting JBoss (Microcontainer)...
16:25:50,062 INFO  [ServerImpl] Release ID: JBoss [The Oracle] 5.1.0.GA (build:
SVNTag=JBoss_5_1_0_GA date=200905221634)
...


16:26:40,420 INFO  [TomcatDeployment] deploy, ctxPath=/admin-console
16:26:40,485 INFO  [config] Initializing Mojarra (1.2_12-b01-FCS) for context
'/admin-console'
16:26:42,362 INFO  [TomcatDeployment] deploy, ctxPath=/
16:26:42,406 INFO  [TomcatDeployment] deploy, ctxPath=/jmx-console
16:26:42,471 INFO  [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-
8080
16:26:42,487 INFO  [AjpProtocol] Starting Coyote AJP/1.3 on ajp-127.0.0.1-8009
```

```
16:26:42,493 INFO  [ServerImpl] JBoss (Microcontainer) [5.1.0.GA (build: 
SVNTag=JBoss_5_1_0_GA date=200905221634)] Started in 52s:427ms
```

Deploy the Project

- Select EjbComponent project in Project Explorer window.
- Right click on it to open context menu.
- Select Deploy.

You will see the following output in NetBeans console output.

```
ant -f C:\\EJB\\EjbComponent -DforceRedeploy=true -
Ddirectory.deployment.supported=false -Dnb.wait.for.caches=true run

init:

deps-jar:

compile:

library-inclusion-in-archive:

Building jar: C:\EJB\EjbComponent\dist\EjbComponent.jar

dist-directory-deploy:

pre-run-deploy:

Checking data source definitions for missing JDBC drivers...

Distributing C:\EJB\EjbComponent\dist\EjbComponent.jar to 
[org.jboss.deployment.spi.LocalhostTarget@1e4f84ee]

Deploying C:\EJB\EjbComponent\dist\EjbComponent.jar

Application Deployed

Operation start started

Operation start completed

post-run-deploy:

run-deploy:

run:

BUILD SUCCESSFUL (total time: 2 seconds)
```

JBoss Application Server Log Output

```
16:30:00,963 INFO  [DeployHandler] Begin start, [EjbComponent.jar]

...
```

```
16:30:01,233 INFO  [Ejb3DependenciesDeployer] Encountered deployment
AbstractVFSDeploymentContext@12038795{vfszip:/C:/jboss-
5.1.0.GA/server/default/deploy/EjbComponent.jar/}

...

16:30:01,281 INFO  [JBossASKernel]    jndi:LibrarySessionBean/remote-
com.tutorialspoint.stateless.LibrarySessionBeanRemote

16:30:01,281 INFO  [JBossASKernel]
Class:com.tutorialspoint.stateless.LibrarySessionBeanRemote

16:30:01,281 INFO  [JBossASKernel]    jndi:LibrarySessionBean/remote

16:30:01,281 INFO  [JBossASKernel]  Added
bean(jboss.j2ee:jar=EjbComponent.jar,name=

LibrarySessionBean,service=EJB3) to KernelDeployment of: EjbComponent.jar

16:30:01,282 INFO  [JBossASKernel] installing bean:
jboss.j2ee:jar=EjbComponent.jar,name=BookMessageHandler,service=EJB3

16:30:01,282 INFO  [JBossASKernel]   with dependencies:

16:30:01,282 INFO  [JBossASKernel]   and demands:

16:30:01,282 INFO  [JBossASKernel]    jboss.ejb:service=EJBTimerService

...

16:30:01,283 INFO  [EJB3EndpointDeployer] Deploy
AbstractBeanMetaData@5497cb{name=jboss.j2ee:jar=EjbComponent.jar,
name=LibrarySessionBean, service=EJB3_endpoint
bean=org.jboss.ejb3.endpoint.deployers.impl.EndpointImpl properties=[container]
constructor=null autowireCandidate=true}

...

16:30:01,394 INFO  [SessionSpecContainer] Starting
jboss.j2ee:jar=EjbComponent.jar,name=LibrarySessionBean,service=EJB3

16:30:01,395 INFO  [EJBContainer] STARTED EJB:
com.tutorialspoint.stateless.LibrarySessionBean ejbName: LibrarySessionBean

16:30:01,401 INFO  [JndiSessionRegistrarBase] Binding the following Entries in
Global JNDI:

   LibrarySessionBean/remote - EJB3.x Default Remote Business Interface

   LibrarySessionBean/remote-com.tutorialspoint.stateless.LibrarySessionBeanRemote
- EJB3.x Remote Business Interface

16:30:02,723 INFO  [SessionSpecContainer] Starting
jboss.j2ee:jar=EjbComponent.jar,name=LibrarySessionBean,service=EJB3

16:30:02,723 INFO  [EJBContainer] STARTED EJB:
com.tutorialspoint.stateless.LibrarySessionBean ejbName: LibrarySessionBean

16:30:02,731 INFO  [JndiSessionRegistrarBase] Binding the following Entries in
Global JNDI:
   LibrarySessionBean/remote - EJB3.x Default Remote Business Interface

   LibrarySessionBean/remote-com.tutorialspoint.stateless.LibrarySessionBeanRemote
- EJB3.x Remote Business Interface
```

Create Client to Access EJB

- In NetBeans IDE, select **File > New Project >**.
- Select project type under category **Java**, Project type as **Java Application**. Click **Next >** button.
- Enter project name and location. Click **Finish >** button. We have chosen name as EjbTester.
- Right click on project name in Project explorer window. Select **properties**.
- Add EJB component project created earlier under libraries using **Add Project** button in **compile** tab.
- Add jboss libraries using **Add jar/folder** button in **compile** tab. Jboss libraries can be located at <jboss installation folder>> client folder.