

EJB enterprise java beans

simply easylearning

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

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of the contents of this e-book in any manner without written consent of the publisher.

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 <u>contact@tutorialspoint.com</u>



Table of Contents

	About the Tutoriali
	Audiencei
	Prerequisitesi
	Copyright & Disclaimeri
	Table of Contentsii
1.	EJB – OVERVIEW1
	Турез1
	Benefits1
2.	EJB – ENVIRONMENT SETUP2
	System Requirement2
3.	EJB – CREATE APPLICATION
	Create Project7
	Create a Sample EJB9
	Build the Project11
	Start the Application Server
	Deploy the Project
	Create Client to Access EJB15
	Run Client to Access EJB
4.	EJB – STATELESS BEAN
	Steps to Create a Stateless EJB
	Example Application21
	EJBComponent (EJB Module)21
	EJBTester (EJB Client)23
	Run Client to Access EJB



	Run Client Again to Access EJB	27
5.	EJB – STATEFUL BEAN	29
	Steps to Create Stateful EJB	29
	Example Application	30
	EJBComponent (EJB Module)	30
	EJBTester (EJB Client)	32
	Run Client to Access EJB	36
	Run Client Again to Access EJB	36
6.	EJB – PERSISTENCE	
	Create Table	
	Create Entity Class	38
	Create DataSource and Persistence Unit	39
	Create Stateless EJB having EntityManager Instance	40
	Example Application	41
	EJBComponent (EJB Module)	41
	EJBTester (EJB Client)	45
	Run Client to Access EJB	48
	Run Client Again to Access EJB	49
7.	EJB – MESSAGE DRIVEN BEANS	50
	Create Queue	50
	Create Message Driven Bean	51
	Example Application	52
	EJBComponent (EJB Module)	52
	EJBTester (EJB Client)	54
	Run Client to Access EJB	57
8.	EJB – ANNOTATIONS	59 iii



9.	EJB – CALLBACKS
	Example Application63
	EJBComponent (EJB Module)63
	EJBTester (EJB Client)70
	Run Client to Access EJB
10.	EJB – TIMER SERVICE
	Steps to Create Timer76
	Steps to Use Timer76
	Example Application77
	EJBComponent (EJB Module)77
	EJBTester (EJB Client)79
	Run Client to Access EJB81
11.	EJB – DEPENDENCY INJECTION
	Steps to Use @EJB
	Steps to use @Resource
	Example Application83
	EJBComponent (EJB Module)83
	EJBTester (EJB Client)85
	Run Client to Access EJB
12.	EJB – INTERCEPTORS
	Example Application91
	EJBComponent (EJB Module)92
	JBoss Application Server Log Output93
	EJBTester (EJB Client)93
	Run Client to Access EJB



13.	EJB – EMBEDDABLE OBJECTS	99
	Example Application	.100
	EJBComponent (EJB Module)	.100
	EJBTester (EJB Client)	.105
	Run Client to Access EJB	. 109
14.	EJB – BLOBS/CLOBS	111
	Example Application	.111
	EJBComponent (EJB Module)	.112
	EJBTester (EJB Client)	.116
	Run Client to Access EJB	.120
15.	EJB – TRANSACTIONS	121
	Container Managed Transactions	.121
	Bean Managed Transactions	.123
16.	EJB – SECURITY	125
	Important Terms of Security	.125
	Container Managed Security	.125
	Security Configuration	.126
17.	EJB – JNDI BINDINGS	128
18.	EJB – ENTITY RELATIONSHIPS	131
	Create Tables	.131
	Create Entity Classes	.132
	Example Application	.133
	EJBComponent (EJB Module)	.133
	EJBTester (EJB Client)	.138
	Run Client to Access EJB	.142



v

19.	EJB – ACCESS DATABASE	144
	Create Table	144
	Create a Model Class	144
	Create Stateless EJB	145
	Example Application	145
	EJBComponent (EJB Module)	146
	EJBTester (EJB Client)	150
	Run Client to Access EJB	153
20.	EJB – QUERY LANGUAGE	155
	Create Table	155
	Create a Model Class	155
	Create Stateless EJB	156
	Example Application	156
	EJBComponent (EJB Module)	157
	EJBTester (EJB Client)	160
	Run Client to Access EJB	163
21.	EJB – EXCEPTION HANDLING	165
	How Does EJB Container Handle Exceptions?	165
	Handling Application Exception	165
	Handling System Exception	166
22.	EJB – WEB SERVICES	168
	Example Application	168
	Create Client to Access EJB as Web Service	170
	Run the Client	174
23.	EJB – PACKAGING APPLICATIONS	176



vi

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:

Туре	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	c:\> java -version
WINDOWS	Console	
Linux	Open Command	\$ java -version
LIIIUX	Terminal	
Мас	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)
Мас	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
Мас	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/
Мас	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 <u>https://netbeans.org/downloads/index.html</u>. At the time of writing this tutorial, I downloaded *Netbeans 7.3,* which comes bundled with JDK 1.7 using the following link <u>http://www.oracle.com/technetwork/java/javase/downloads/index.html</u>

OS	Installer name
Windows	Netbeans 7.3
Linux	Netbeans 7.3
Мас	Netbeans 7.3

Step 4 – Set up JBoss Application Server

You can download the latest version of JBoss Server from <u>http://www.jboss.org/jbossas/downloads/</u>. 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
Мас	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).

pdate	s Available Plugins (121)	Downloaded 1	Installed	(42)	Settings	
					Search:	
Se	Name	Category ₹	Ac	:	Java EE Bace	
同	Spring Beans	Java SE	Ö		Java LL Dase	
E	Ant	Java SE	Ø		Version: 1, 18, 1	
100	Java Profiler	Java SE	Ø		Source: NetBeans Distribution	
两	Hibernate	Java SE	Ø			
100	Maven	Java SE	Ø		mt 1 m 1 t	
m	Java Debugger	Java SE	Ø		Plugin Description	
F	GUI Builder	Java SE	Ø		Provides baselevel support for Java EE, i.e. web application	
V	Java EE Base	Java Web an	. 😨	project and basic wizards, plus support for o	project and basic wizards, plus support for deployment,	
同	RichFaces Java Web an Image: Construction of the second s	debugging and profiling web applications on GlassFish,				
1		Java Web an	n O		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.	
		Java Web an				
		Java Web an	. 0	5		
回		Java Web an	0			
們	RESTful Web Services	Java Web an	. 0			
1271	SOAP Web Services	Java Web an	. 0	100		
E	PrimeFaces	Java Web an	. 0			
-	100	.terre titele en	-	-11		
Act	vate Deactivate	Uginstal	2 plug	ins se	slected	

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.

Steps	Choose Server	
1. Choose Server 2	Server: Apache Tomcat GlassFish Server 3+ JBoss Application Server Orade WebLogic Server	
	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 <u>http://www.postgresql.org/download/</u>. At the time of writing this tutorial, I downloaded *PostGreSql 9.2*

OS	OS Installer name		
Windows	PostGreSql 9.2		
Linux	PostGreSql 9.2		
Мас	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

	Choose Project				
Categories: ↓ Java ↓ JavaFX ↓ JavaFX ↓ Java Web ↓ Java EE ↓ HTML/JavaScript ↓ Maven ↓ NetBeans Modules ⊕ ↓ Samples	Projects: Enterprise Application Enterprise Application with Existing Sources EIB Module EIB Module with Existing Sources Enterprise Application Client Enterprise Application Client with Existing Sources				
Description:	1010				
Creates a new Enterprise Java projects use an IDE-generated A	Bean (EJB) module in a standard IDE project. Standard ant build script to build and run your project.				
	Quegones. Java JavaFX Java Web Java EE Maven Maven NetBeans Modules Gescription: Creates a new Enterprise Java projects use an IDE-generated A				

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

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



ceps	Name and Loca	tion	
. Choose Project	Project Name:	EjbComponent	
Server and Settings	Project Location:	C: Wsers\Acer\Documents\NetBeansProjects	Browse
	Project Eolder:	C:\Users\Acer\Documents\NetBeansProjects\EjbComponent	
	Use Dedicate	d Folder for Storing Libraries	
	Libraries Folder:		Browse
1			

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

Step		Server and Set	tings	
1. 0	Thoose Project Iame and Location	Add to Enterprise	Application: <	
3. 9	Server and Settings	Server:	JBoss Application Server	← Add
		Java EE Version:	Java EE 5 💌	
			V Set Source Level to 1.5	
			Recommendation: Source Level 1.5 should be used in Java EE 5 projects.	
	X			



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

EjbCompo	nent	All the second sec
 □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	r rackages K 1.6 oss Application Se rise Beans uration Files Resources	ver

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.



teps	Name and Location
. Choose File Type . Name and Location	EJB Name: NewSessionBean
	Project: EjbComponent
	Location: Source Packages 🗸 🗸
	Padgage: com.tutorialspoint.stateless
	© Stateful Create Interface: ☑ Local Ⅲ 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.



End of ebook preview If you liked what you saw... Buy it from our store @ **https://store.tutorialspoint.com**