EJB - WEB SERVICES

EJB 3.0 provides option to expose session ejb as a webservice. @WebService annotation is used to mark a class as a web service end point and @WebMethod is used to expose a method as web method to client.

```
@Stateless
@WebService(serviceName="LibraryService")
public class LibraryPersistentBean implements LibraryPersistentBeanRemote {

   ...
   @WebMethod(operationName="getBooks")
   public List getBooks()  {
      return entityManager.createQuery("From Books").getResultList();
   }
   ...
}
```

Example Application

Let us create a test EJB application to test blob/clob support in EJB 3.0.

Step	Description
1	Create a project with a name EjbComponent under a package com.tutorialspoint.entity as explained in the EJB - Create Application chapter. Please use the project created in EJB - Persistence chapter as such for this chapter to understand clob/blob objects in ejb concepts.
2	Create LibraryPersistentBean.java under package com.tutorialspoint.stateless. Use EJB - Persistence chapter as reference. Keep rest of the files unchanged.
3	Clean and Build the application to make sure business logic is working as per the requirements.
4	Finally, deploy the application in the form of jar file on JBoss Application Server. JBoss Application server will get started automatically if it is not started yet.

LibraryPersistentBean.java

```
package com.tutorialspoint.stateless;

import com.tutorialspoint.entity.Book;
import java.util.List;
import javax.ejb.Stateless;
import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

@Stateless
@WebService(serviceName="LibraryService")
public class LibraryPersistentBean implements LibraryPersistentBeanRemote {

   public LibraryPersistentBean(){
   }

   @PersistenceContext(unitName="EjbComponentPU")
   private EntityManager entityManager;

   public void addBook(Book book) {
```

```
        entityManager.persist(book);
    }

    @WebMethod(operationName="getBooks")
    public List <Book> getBooks() {
        return entityManager.createQuery("From Book").getResultList();
    }
}
```

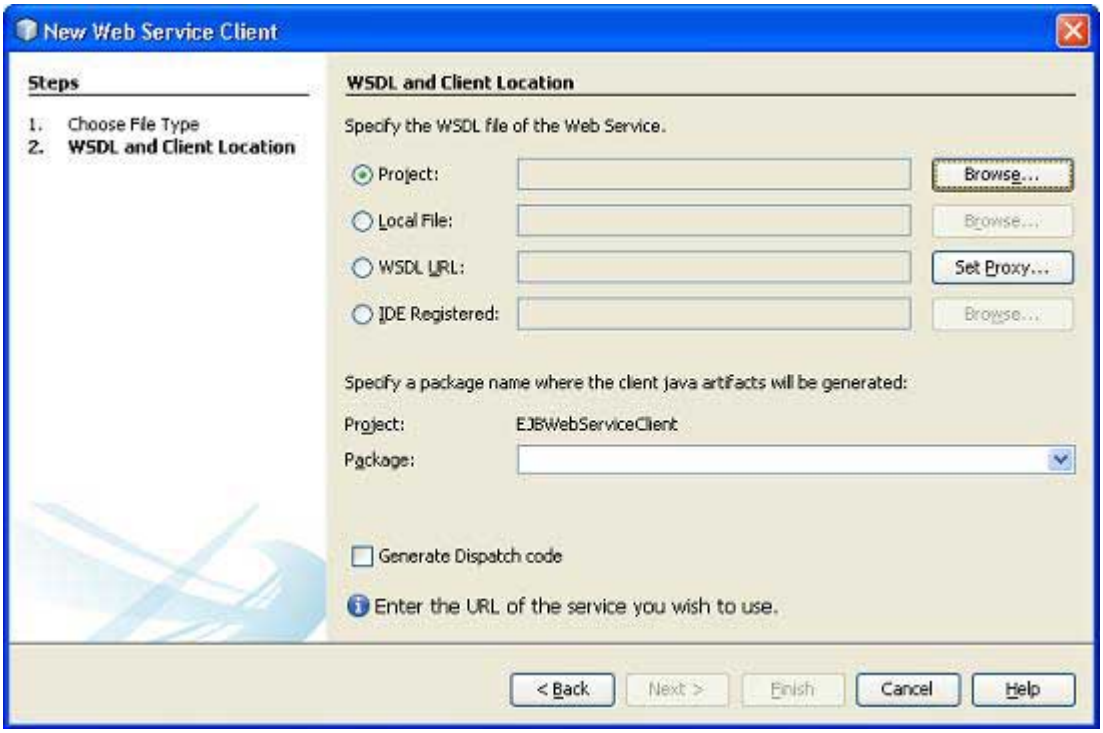
JBoss Application server log output

```
10:51:37,271 INFO  [EJBContainer] STARTED EJB:
com.tutorialspoint.stateless.LibraryPersistentBean ejbName: LibraryPersistentBean
10:51:37,287 INFO  [JndiSessionRegistrarBase] Binding the following Entries in Global
JNDI:

 LibraryPersistentBean/remote - EJB3.x Default Remote Business Interface
 LibraryPersistentBean/remote-com.tutorialspoint.stateless.LibraryPersistentBeanRemote -
EJB3.x Remote Business Interface

10:51:37,349 INFO  [EJBContainer] STARTED EJB:
com.tuturialspoint.messagebean.LibraryMessageBean ejbName: BookMessageHandler
10:51:37,443 INFO  [DefaultEndpointRegistry] register:
jboss.ws:context=EjbComponent,endpoint=LibraryPersistentBean
10:51:38,191 INFO  [WSDLFilePublisher] WSDL published to: file:/D:/Jboss-
5.0.1/server/default/data/wsdl/EjbComponent.jar/
LibraryService3853081455302946642.wsdl
```

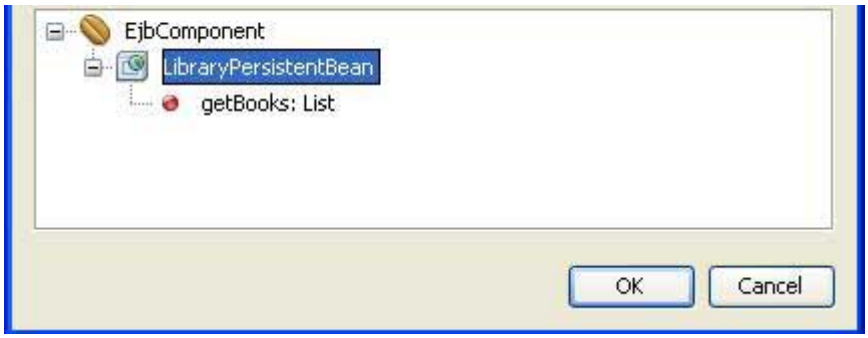
Create Client to access EJB as Web Service

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've chosen name as EJBWebServiceClient.

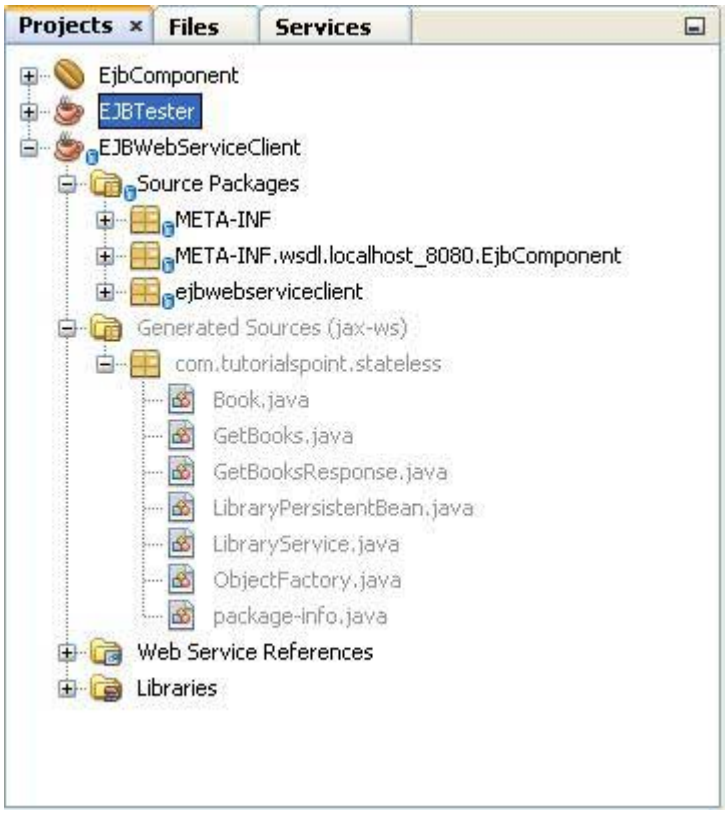
Right click on project name in Project exporer window. Select **New > WebService Client** .



Add ejb component project's LibraryPersistentBean created earlier under WSDL and Client Location using **Add Project** button in **compile** tab.



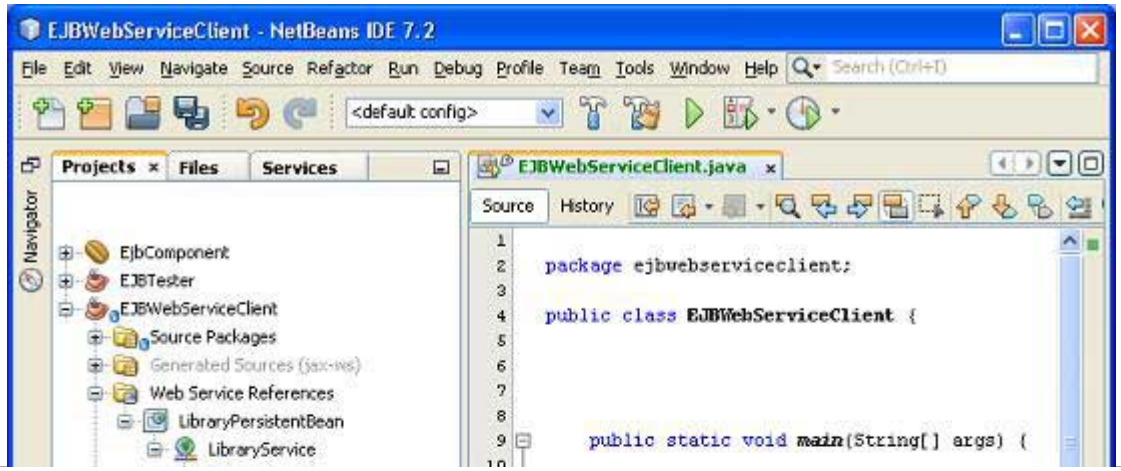
Click Finish Button. Verify the following structure in project explorer.

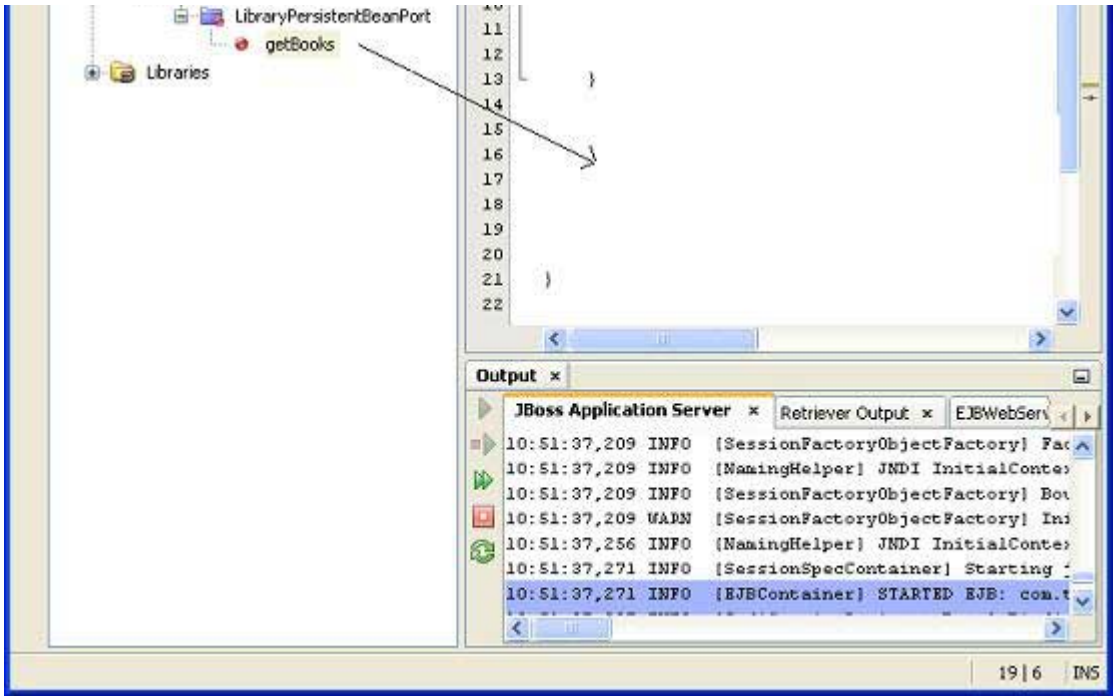


Create EJBWebServiceClient.java

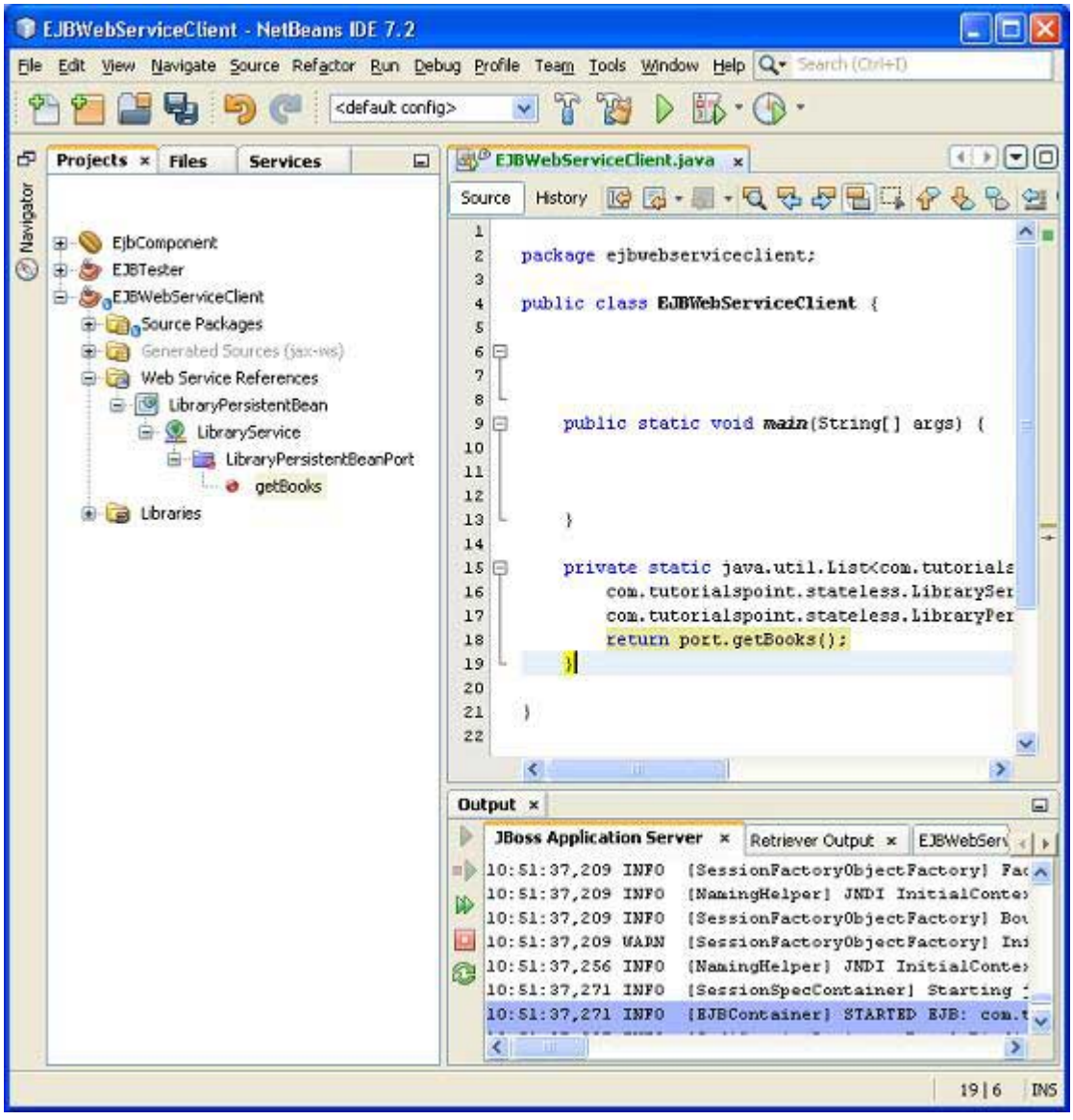
```
package ejbwebserviceclient;

public class EJBWebServiceClient {
   public static void main(String[] args) {
   }
}
```

Select Web Service getBooks web method as shown in figure below and drag it to code window of EJBWebServiceClient.





You'll see the output similar to as shown below.



Update the EJBWebServiceClient code to use this method.

```
package ejbwebserviceclient;
```

```
public class EJBWebServiceClient {

   public static void main(String[] args) {
      for(com.tutorialspoint.stateless.Book book:getBooks()){
         System.out.println(book.getName());
      }
   }

   private static java.util.List
   <com.tutorialspoint.stateless.Book> getBooks() {
      com.tutorialspoint.stateless.LibraryService service =
         new com.tutorialspoint.stateless.LibraryService();
      com.tutorialspoint.stateless.LibraryPersistentBean port =
         service.getLibraryPersistentBeanPort();
      return port.getBooks();
   }
}
```

Run the Client

Right click on project name in Project explorer window. Select **Run**. Netbeans will build the client and run it. Verify the following output.

```
ant -f D:\\SVN\\EJBWebServiceClient run
init:
Deleting: D:\SVN\EJBWebServiceClient\build\built-jar.properties
deps-jar:
Updating property file: D:\SVN\EJBWebServiceClient\build\built-jar.properties
wsimport-init:
wsimport-client-LibraryPersistentBean:
files are up to date
classLoader = java.net.URLClassLoader@4ce46c
SharedSecrets.getJavaNetAccess()=java.net.URLClassLoader$7@182cdac
wsimport-client-generate:
Compiling 1 source file to D:\SVN\EJBWebServiceClient\build\classes
compile:
run:
learn java
Learn Spring
learn JSF
Learn HTML
Learn JBoss
Learn EJB
Learn Hibernate
Learn IBatis
Times Now
learn html5
Learn images
Learn Testing
Forbes
test1
BUILD SUCCESSFUL (total time: 1 second)
```