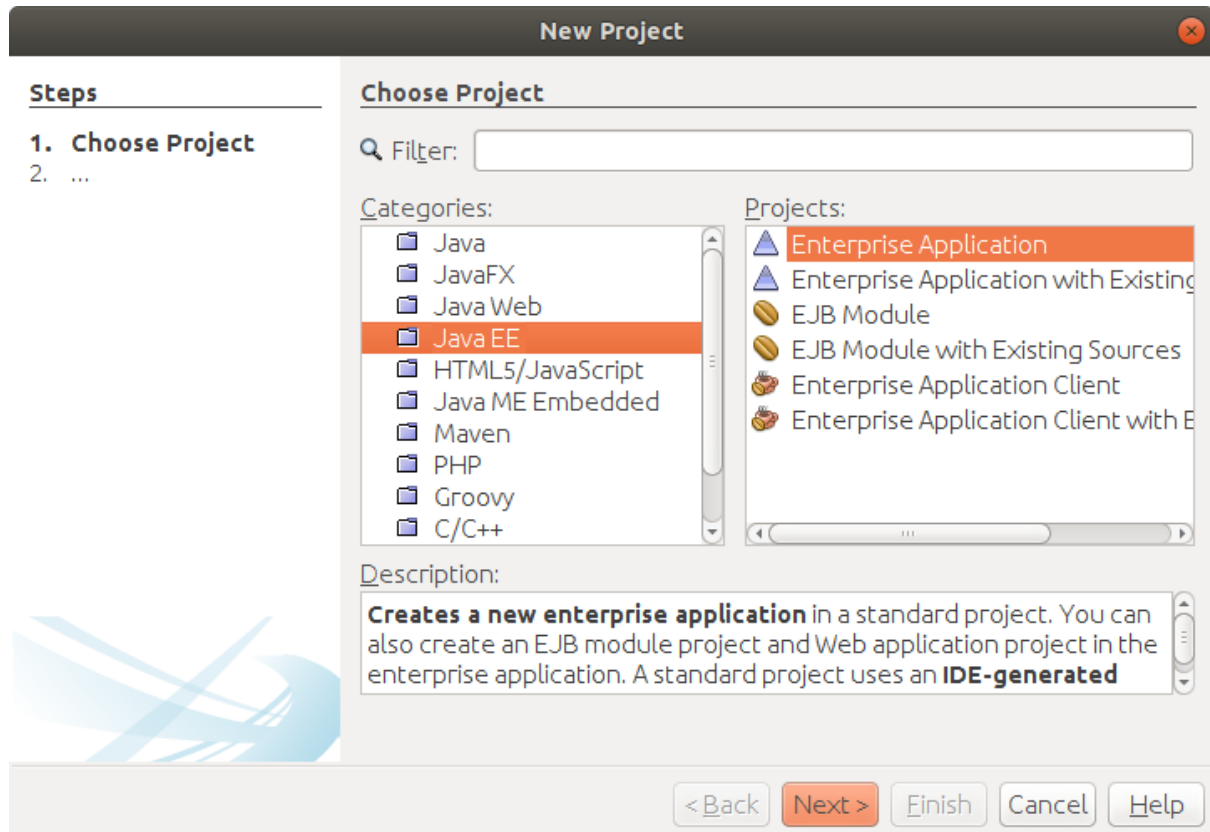


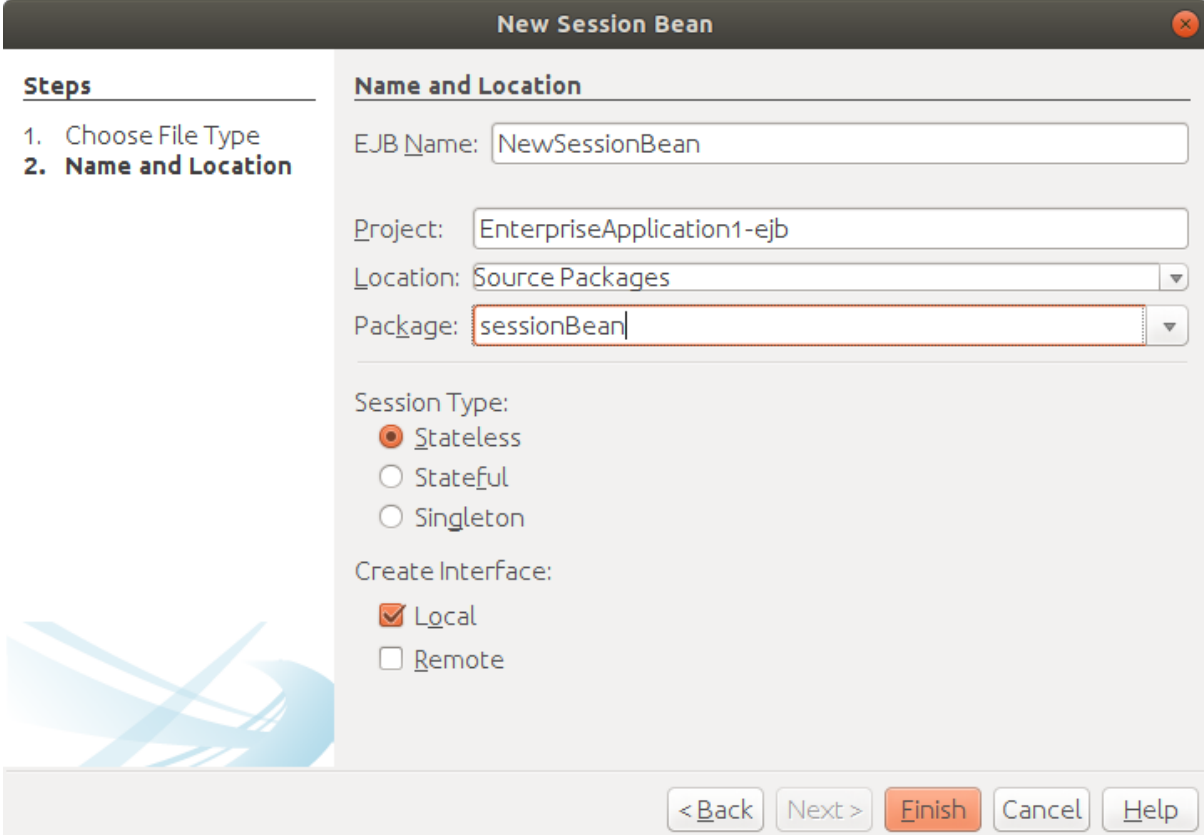
Open Netbeans

File >> New Project >> JAVA EE >> Enterprise Application



EnterpriseApplication1-ejb>>Right Click >>new >> session Beans

Put name for session bean



The image shows a 'New Session Bean' dialog box with a dark title bar. On the left, a 'Steps' panel lists '1. Choose File Type' and '2. Name and Location'. The main area is titled 'Name and Location' and contains several input fields: 'EJB Name' with the text 'NewSessionBean', 'Project' with 'EnterpriseApplication1-ejb', 'Location' with a dropdown set to 'Source Packages', and 'Package' with a dropdown set to 'sessionBean'. Below these are radio buttons for 'Session Type' (Stateless is selected) and checkboxes for 'Create Interface' (Local is checked, Remote is unchecked). At the bottom are buttons for '< Back', 'Next >', 'Finish' (highlighted in orange), 'Cancel', and 'Help'.

New Session Bean

Steps

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

Name and Location

EJB Name: NewSessionBean

Project: EnterpriseApplication1-ejb

Location: Source Packages

Package: sessionBean

Session Type:

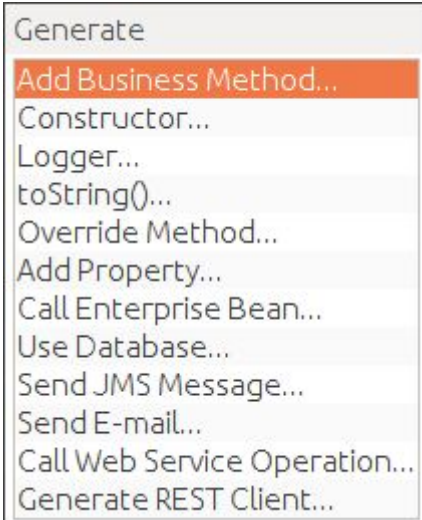
- ☒ Stateless
- ☐ Stateful
- ☐ Singleton

Create Interface:

- ☒ Local
- ☐ Remote

< Back Next > Finish Cancel Help

Right click on code editor and select insert code and select Add Business Method



The image shows a 'Generate' menu with a list of options. The first option, 'Add Business Method...', is highlighted with an orange background. The other options are Constructor..., Logger..., toString()..., Override Method..., Add Property..., Call Enterprise Bean..., Use Database..., Send JMS Message..., Send E-mail..., Call Web Service Operation..., and Generate REST Client....

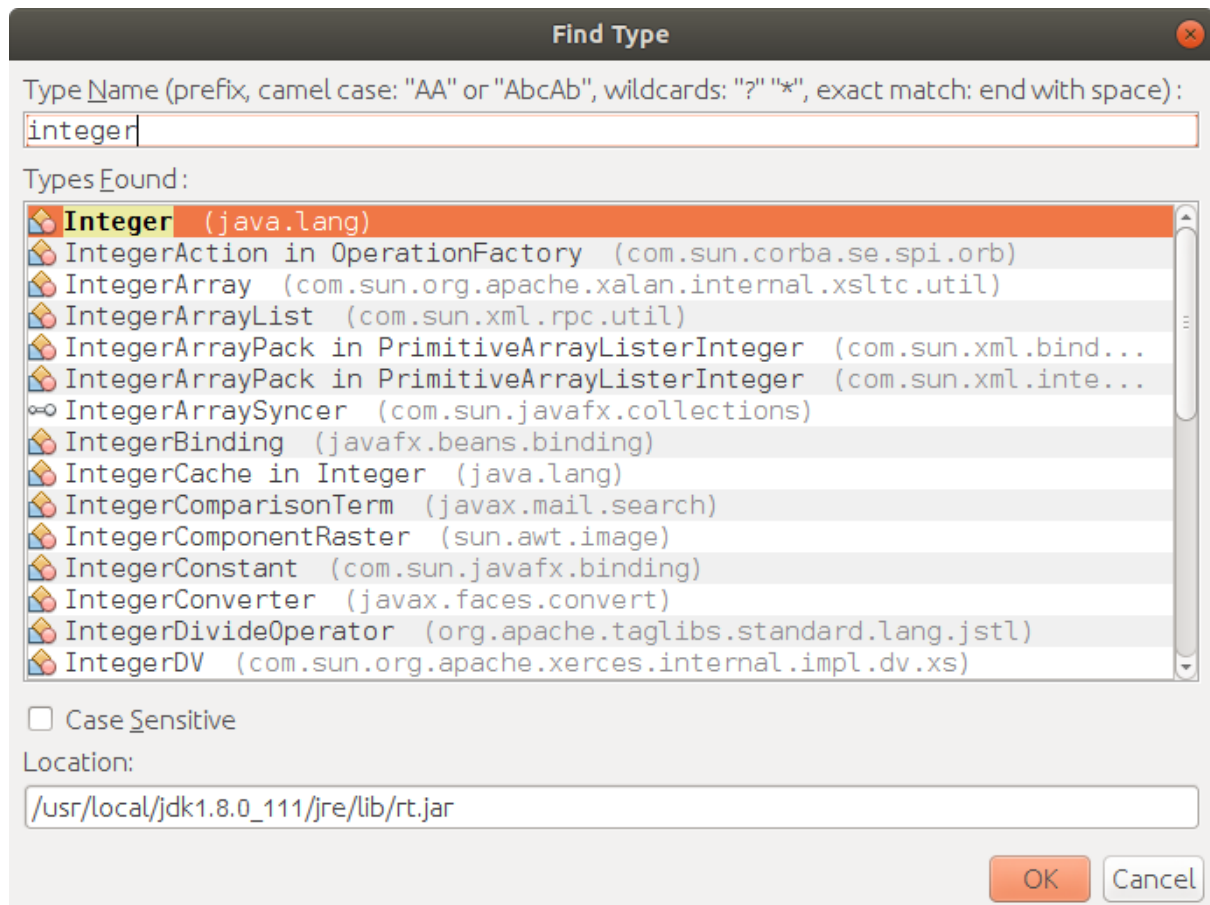
Generate

- Add Business Method...
- Constructor...
- Logger...
- toString()...
- Override Method...
- Add Property...
- Call Enterprise Bean...
- Use Database...
- Send JMS Message...
- Send E-mail...
- Call Web Service Operation...
- Generate REST Client...

Enter following details

Name :add

Return type : Click on browse and type integer , click Ok



Click on add

Enter name as **a** and type select **int** from list then click on add to enter details for parameter **b**

You will get automatically created code as

Add Business Method...

Name: add

Return Type: java.lang.Integer

Browse...

Parameters

Exceptions

Name	Type	Final
a	int	<input type="checkbox"/>
b	int	<input type="checkbox"/>

Add

Remove

Up

Down

Use in Interface:

☒ Local

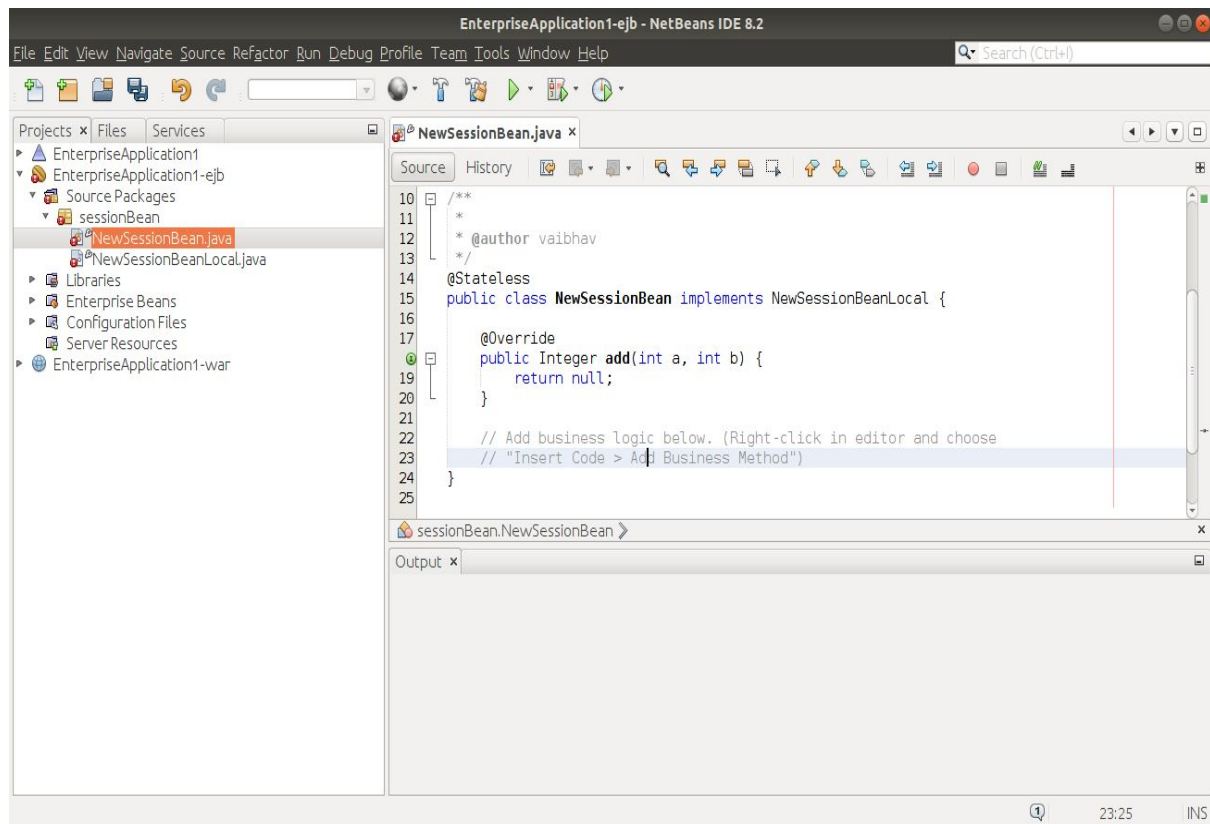
☐ Remote

☐ Both

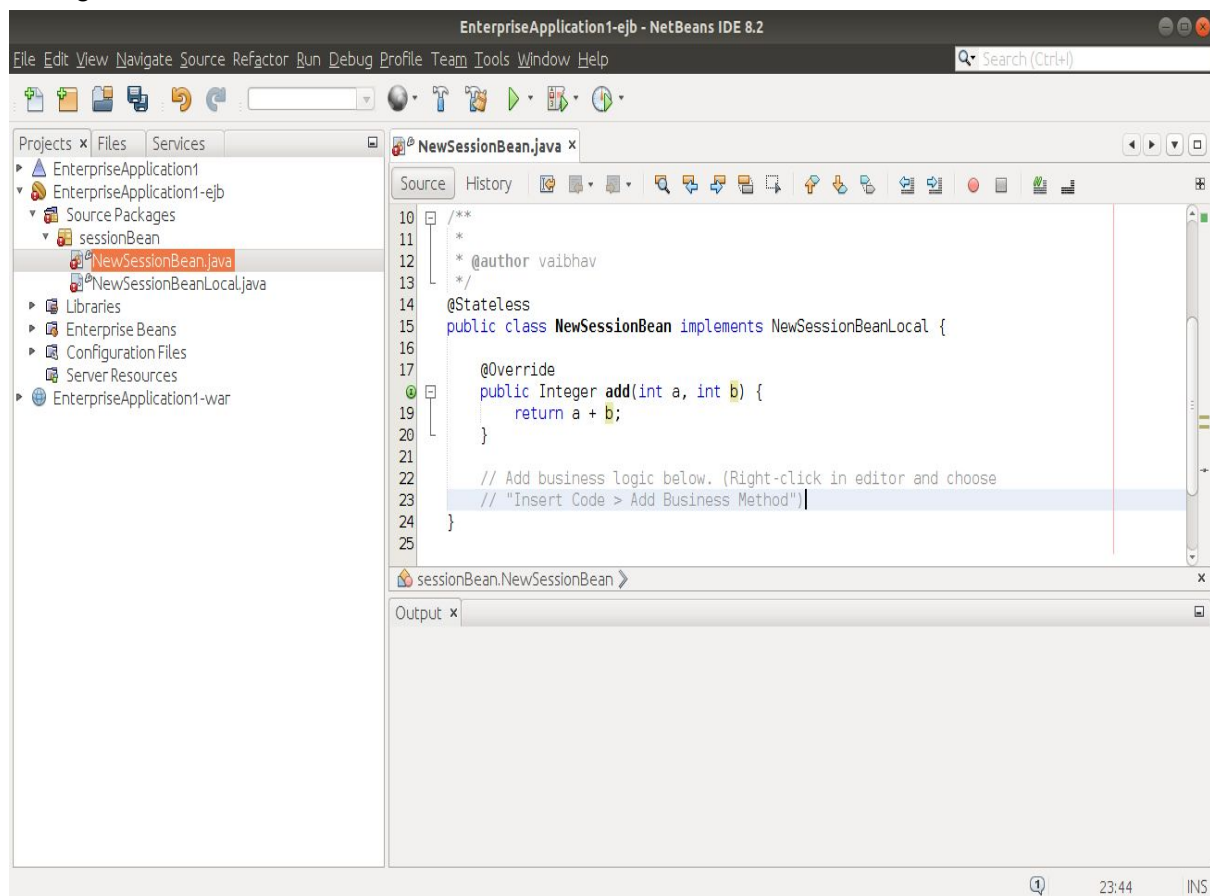
OK

Cancel

You will get automatically created code as



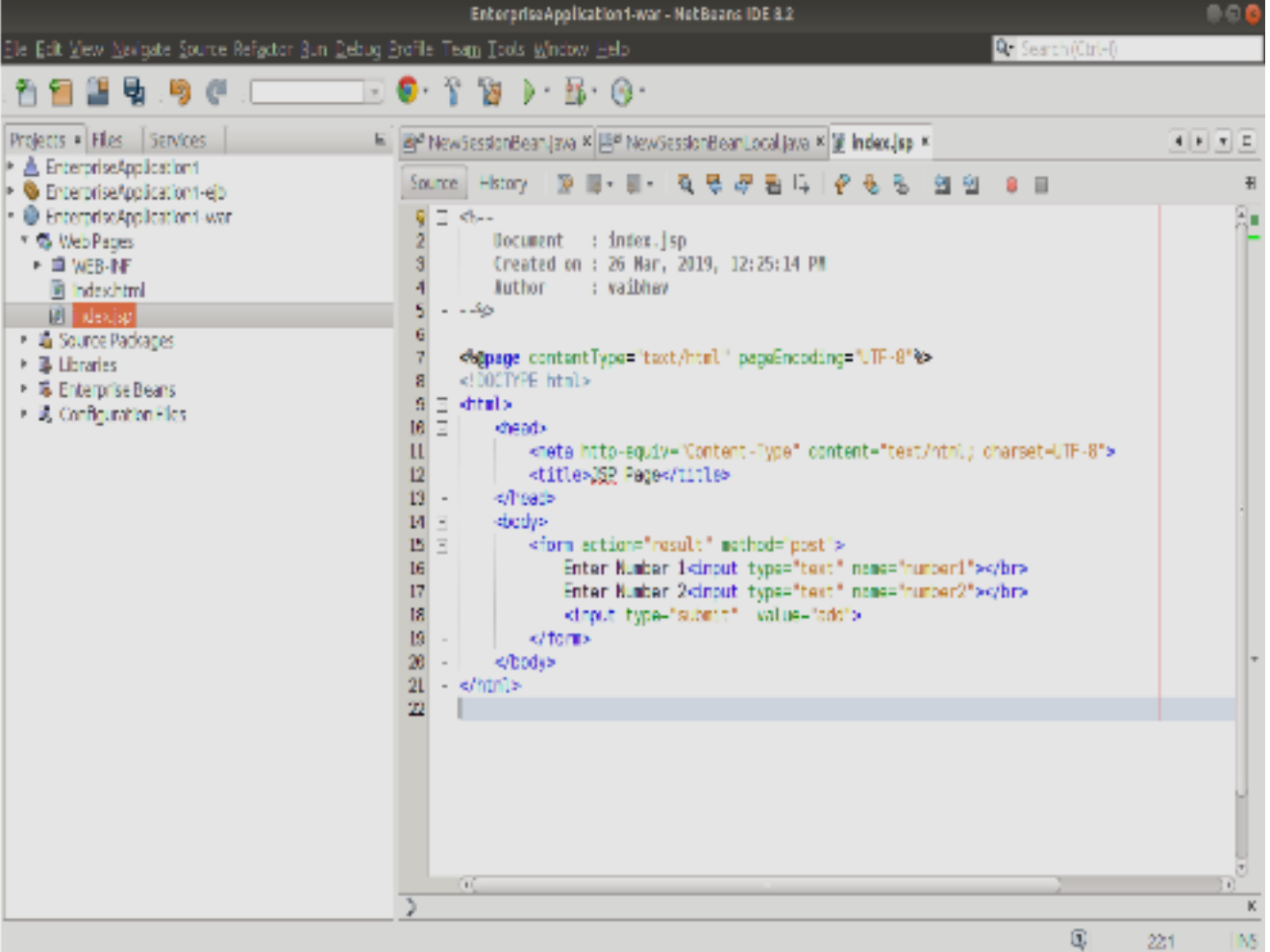
Change code as below



Right click on EnterpriseApplication1-war >> new >> jsp

Add name as index.jsp

Add following code in index.jsp



The screenshot shows the NetBeans IDE 8.2 interface. The title bar reads "EnterpriseApplication1-war - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains icons for file operations and development tools. The left sidebar shows the "Projects" view with the following structure:

- EnterpriseApplication1
- EnterpriseApplication1-eb
- EnterpriseApplication1-war
 - WebPages
 - WEB-INF
 - index.html
 - index.jsp**
 - Source Packages
 - Libraries
 - Enterprise Beans
 - Configuration Files

The main editor window displays the code for "index.jsp". The code is as follows:

```
1 <!--
2   Document   : index.jsp
3   Created on : 26 Mar, 2019, 12:25:14 PM
4   Author    : vaibhav
5 - -->
6
7 <%page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10
11     <head>
12         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13         <title>JSP Page</title>
14     </head>
15     <body>
16         <form action="result" method="post">
17             Enter Number 1<input type="text" name="number1"><br>
18             Enter Number 2<input type="text" name="number2"><br>
19             <input type="submit" value="add"/>
20         </form>
21     </body>
22 </html>
```

Right click on war >> new >> servlet

Put name : result

New Servlet

Steps

1. Choose File Type

2. **Name and Location**

3. Configure Servlet Deployment

Name and Location

Class Name: result

Project: EnterpriseApplication1-war

Location: Source Packages

Package: SessionBean

Created File: rpriseApplication1/EnterpriseApplication1-war/src/java/SessionBean/result.java

< Back

Next >

Finish

Cancel

Help

New Servlet

Steps

1. Choose File Type

2. Name and Location

3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name: SessionBean.result

Servlet Name: result

URL Pattern(s): /result

Initialization Parameters:

Name	Value

New

Edit...

Delete

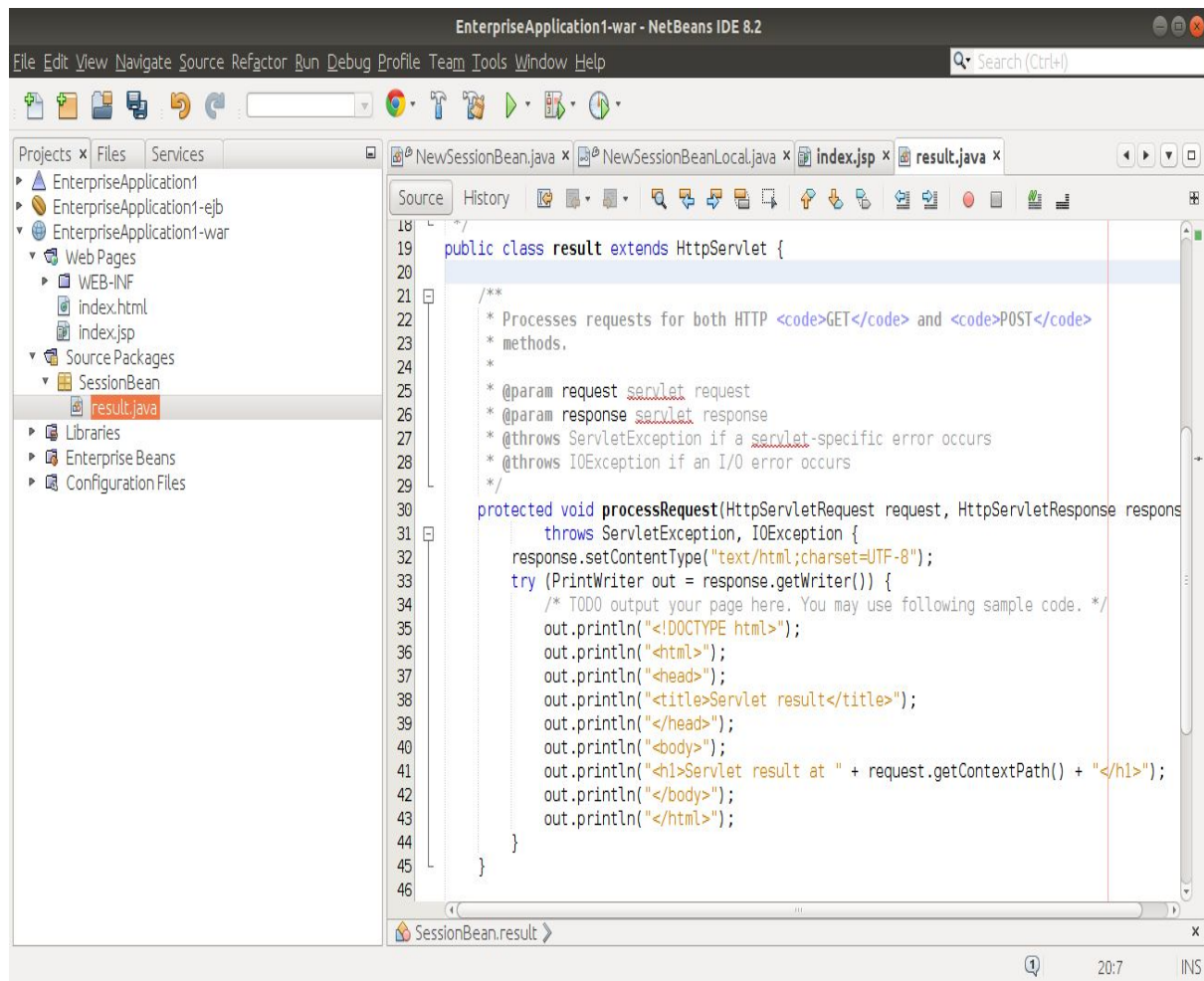
< Back

Next >

Finish

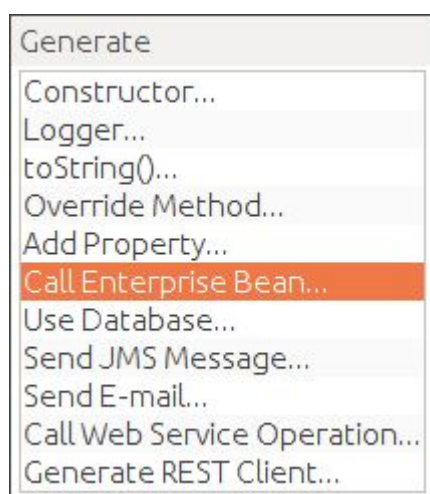
Cancel

Help

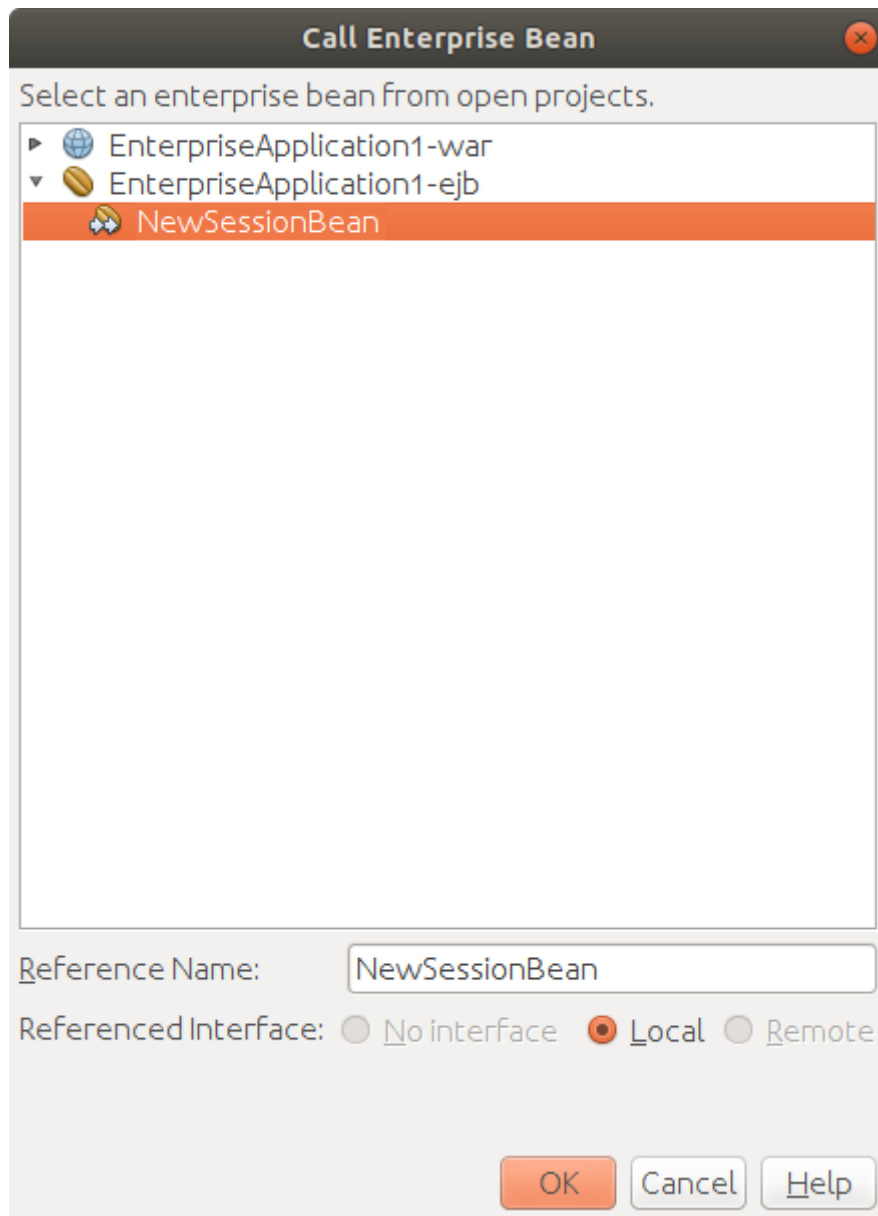


Right click on Line no 20 (after class declaration)

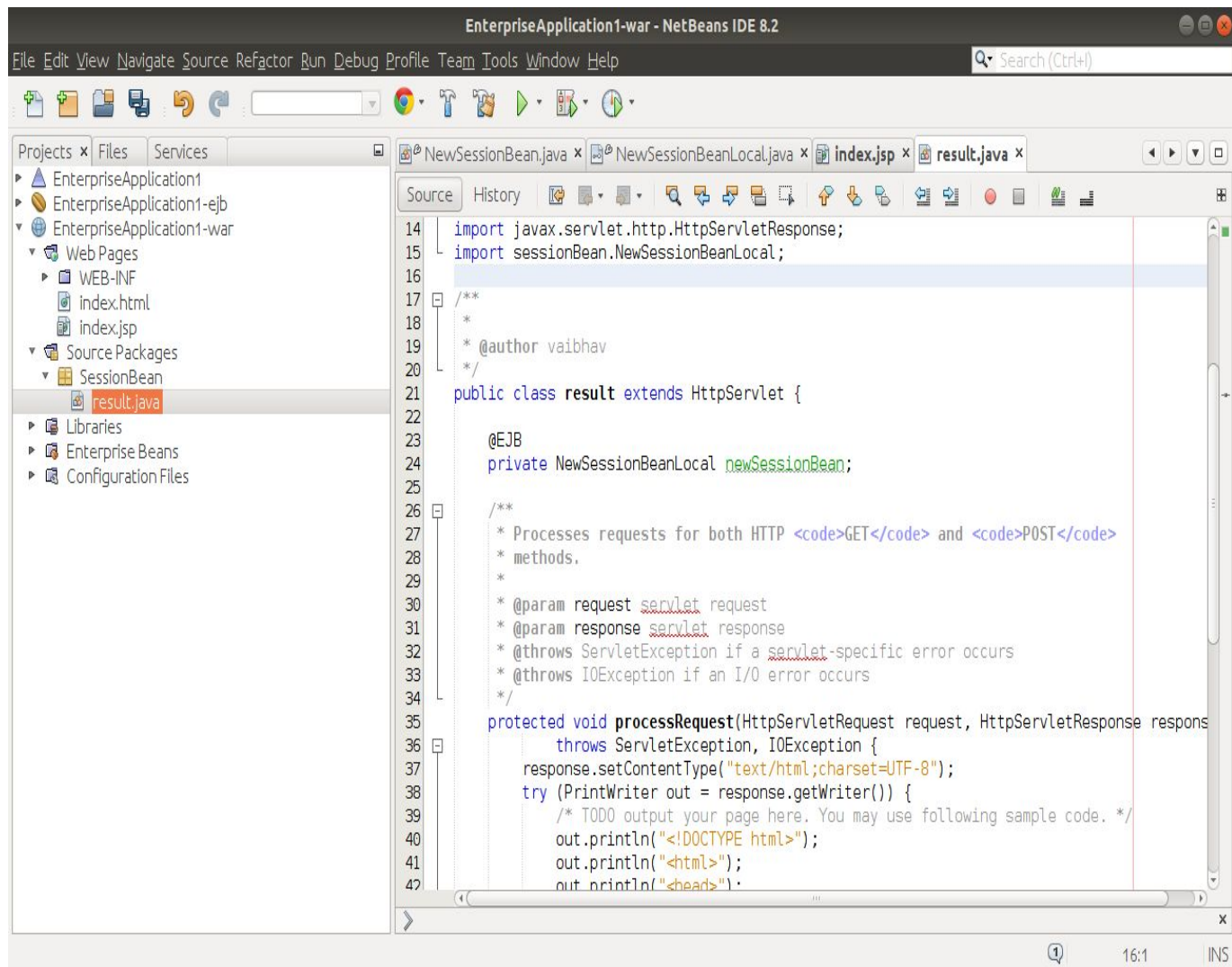
Select insert code >> call business method



Select ejb bean from list



You will get following code



Replace above try code with following code

```

try (PrintWriter out = response.getWriter()) {
    /* TODO output your page here. You may use following sample code. */
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Servlet result</title>");
    out.println("</head>");
    out.println("<body>");

    int a = Integer.parseInt(request.getParameter("number1"));
    int b = Integer.parseInt(request.getParameter("number2"));
    out.println("<h1>Addition is " + newSessionBean.add(a, b) + "</h1>");
    out.println("</body>");
    out.println("</html>");
}

```

Right click on EnterpriseApplication >> Deploy

Right click on EnterpriseApplication >> Run

<http://localhost:8080/EnterpriseApplication1-war/index.jsp>