

Introduction to Web Application - Java Servlet

Objectives:

1. How to create a java website project in netbeans.
2. How to create a jsp and a servlet in netbeans.
3. How to add servlet config in to descriptor file.

Introduction to Web Application

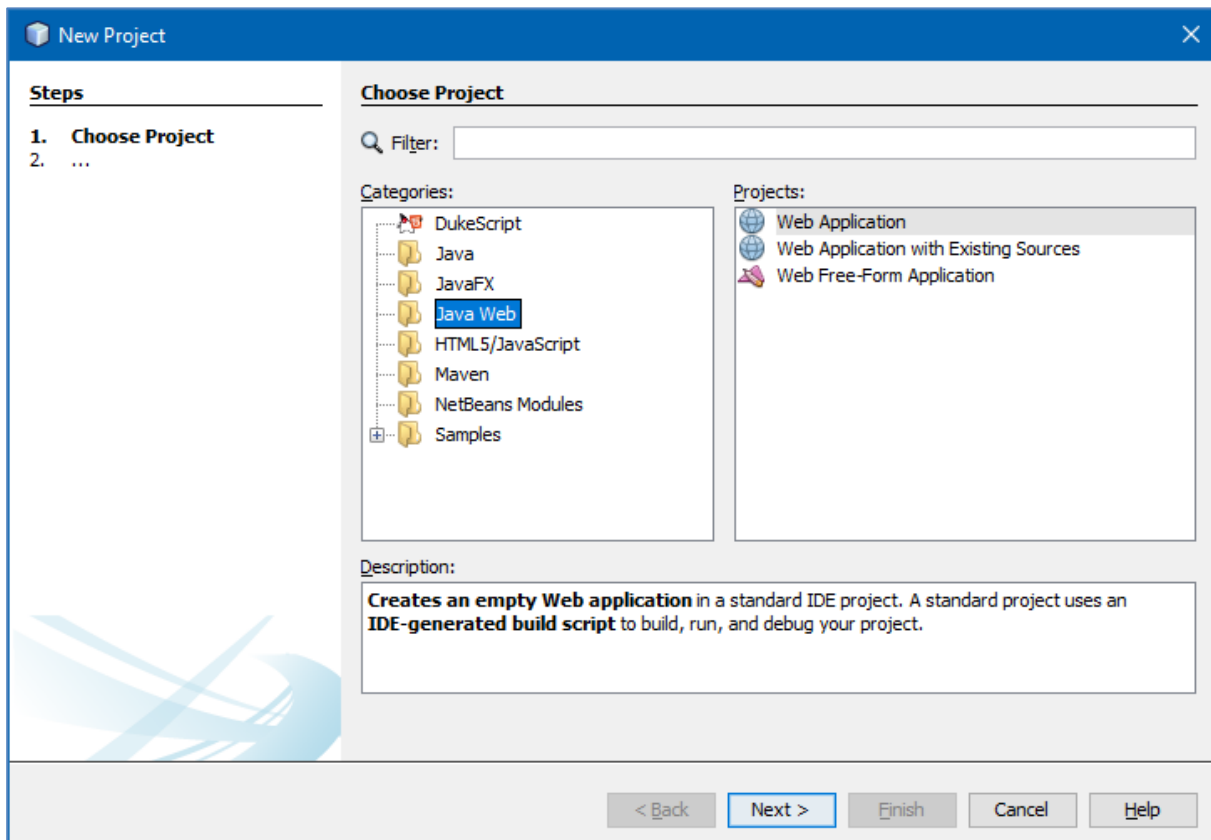
Java Servlet

1.1 Create a website project

In this exercise, we will create a java website by using Netbean IDE.

Step 1. In Netbeans, click on the New Project icon.

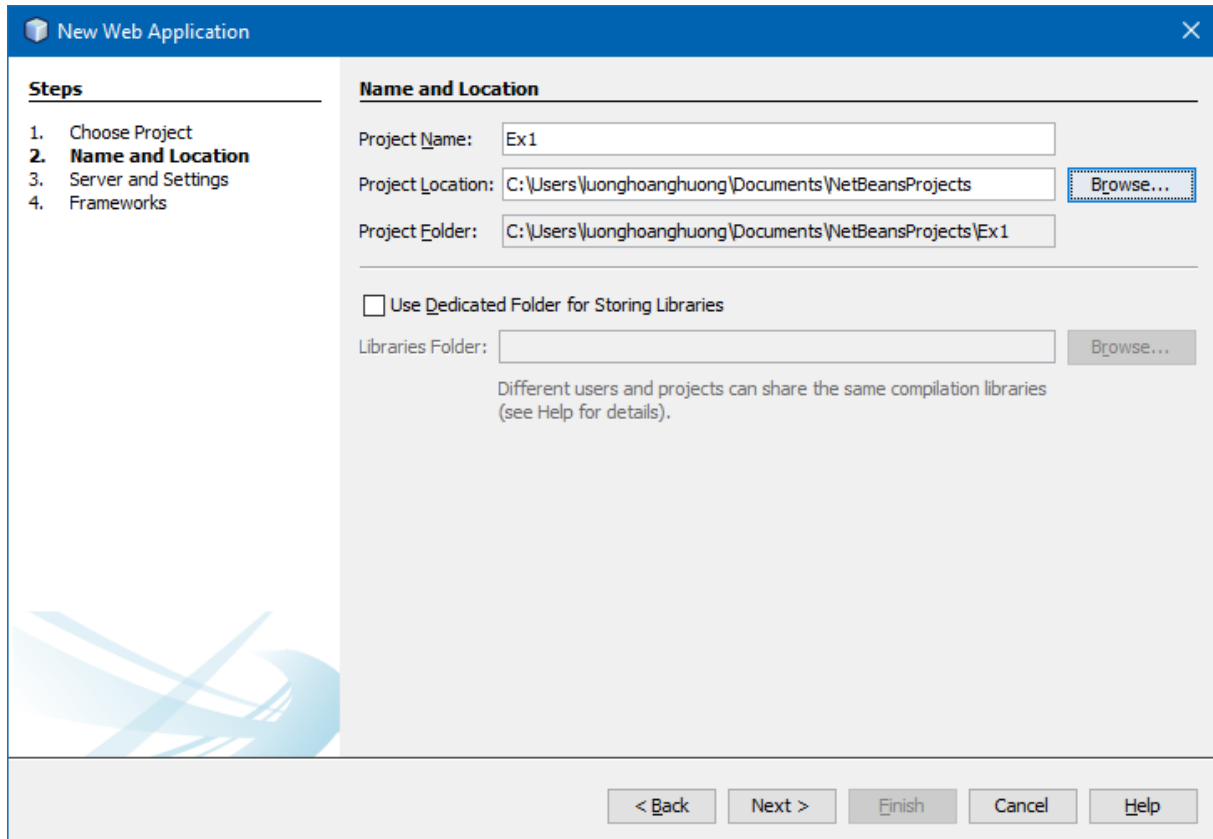
Step 2. New Project window will appear, Select Java Web → Web Application → Next.



Introduction to Web Application

Java Servlet

Step 3. In the next step, change the project name, project location and click Next button.



New Web Application

Steps

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

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

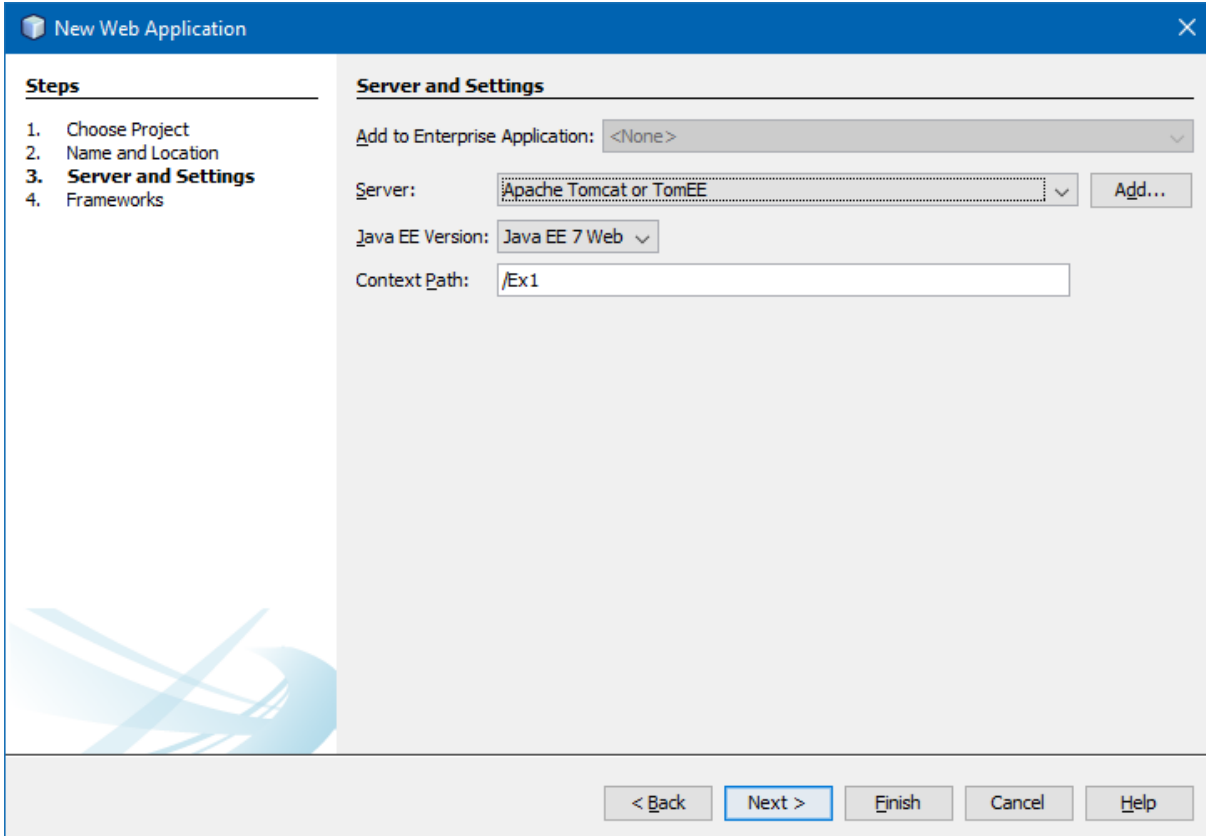
Different users and projects can share the same compilation libraries
(see Help for details).

< Back Next > Finish Cancel Help

Introduction to Web Application

Java Servlet

Step 4. Next step, select tomcat server at Server, select latest version for Java EE Version → Click Next button.



New Web Application

Steps

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

Server and Settings

Add to Enterprise Application: <None>

Server: Apache Tomcat or TomEE Add...

Java EE Version: Java EE 7 Web

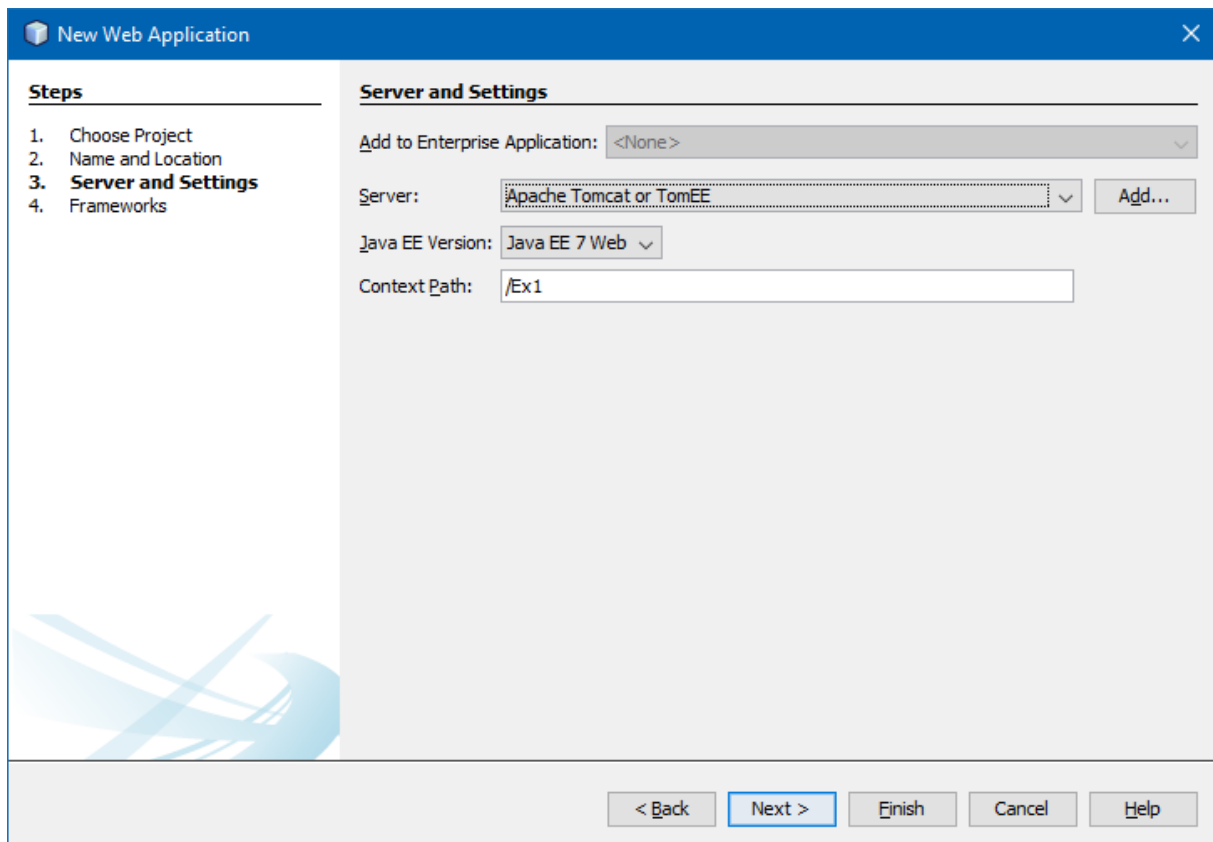
Context Path: /Ex1

< Back Next > Finish Cancel Help

Introduction to Web Application

Java Servlet

Step 5. The project is created.



1.2 Read all header of request

In this exercise, we will create a webpage to send a request to server. After that, server will read all header of request that is sent by client.

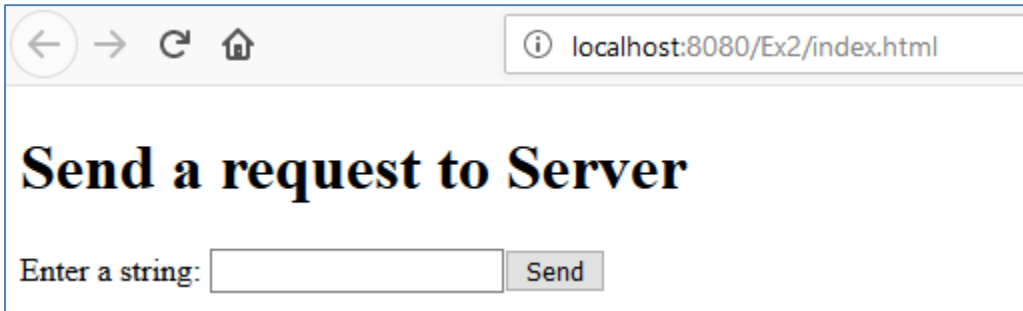
Step 1. Create a project named: Ex2.

Step 2. In Ex2 project, edit the code of index.html as below:

```
<html>
<head>
  <title>TODO supply a title</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <h1>Send a request to Server</h1>
  <form action="MyServlet">
    Enter a string: <input type="text" name="txtDemo" value="" /><input
type="submit" value="Send" />
  </form>
</body>
</html>
```

Introduction to Web Application

Java Servlet

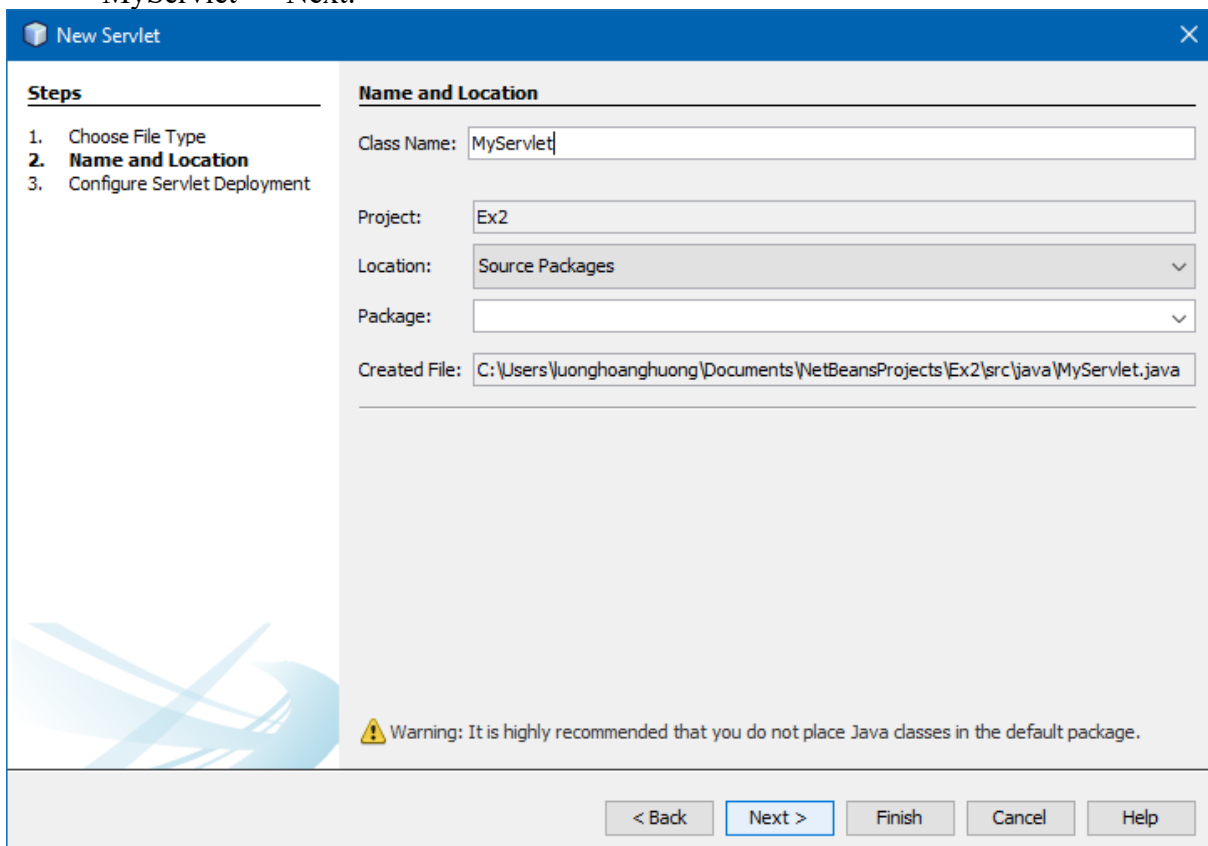


← → ↻ 🏠 localhost:8080/Ex2/index.html

Send a request to Server

Enter a string:

Step 3. Right-click on Ex2 project → New → Servlet. Set name of new servlet is MyServlet → Next.



New Servlet

Steps

1. Choose File Type
- 2. Name and Location**
3. Configure Servlet Deployment

Name and Location


Class Name:

Project:

Location:

Package:

Created File:

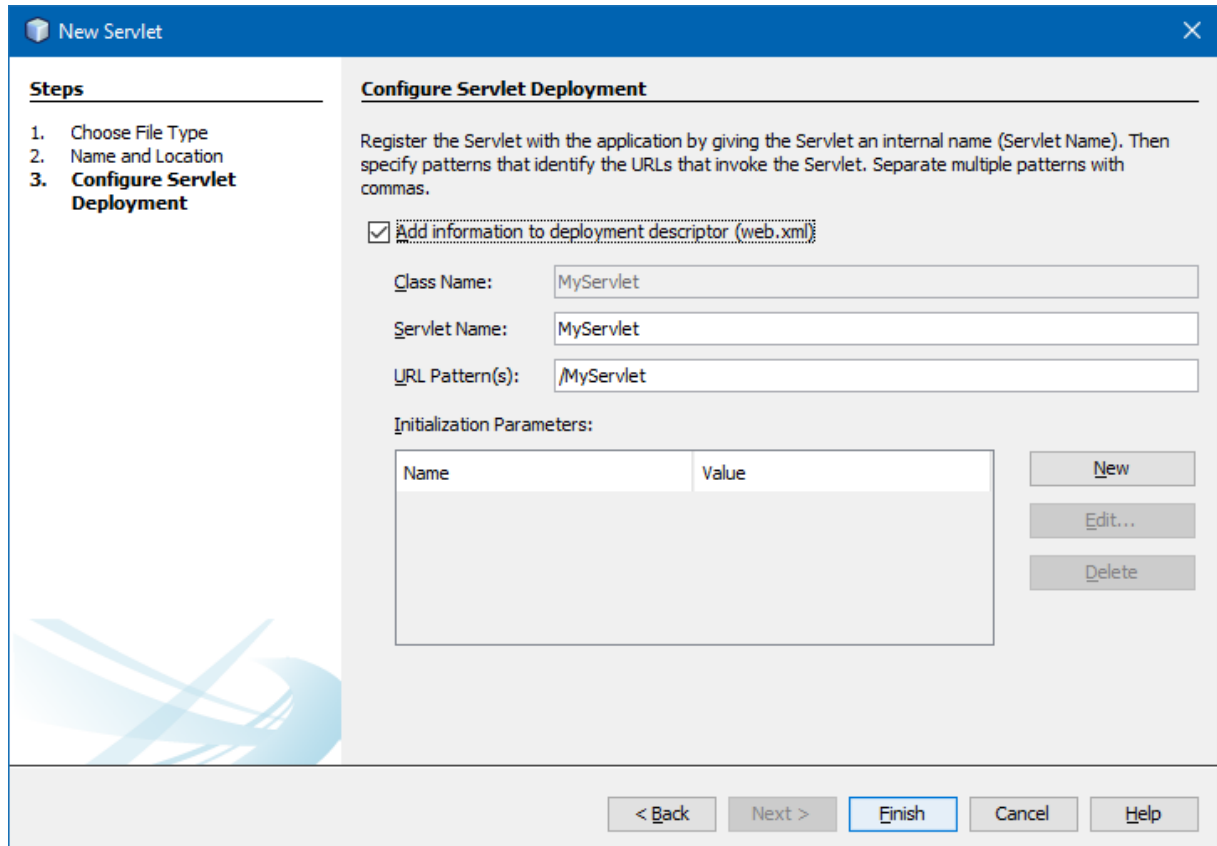
 Warning: It is highly recommended that you do not place Java classes in the default package.

< Back Next > Finish Cancel Help

Introduction to Web Application

Java Servlet

Step 4. Next step, check on "Add information to deployment descriptor (web.xml)" → Finish.



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:

Servlet Name:

URL Pattern(s):

Initialization Parameters:

| Name | Value |
|------|-------|
|------|-------|

New Edit... Delete

< Back Next > Finish Cancel Help

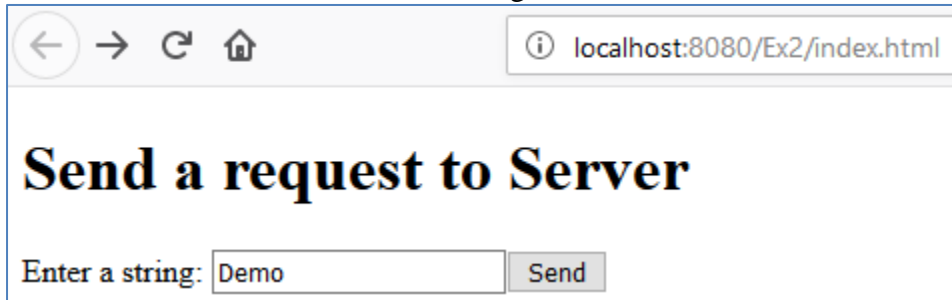
Step 5. Change the code in `processRequest()` method as below. In this code, we will use some methods `getParameter()`, `getHeaderNames()`, `getHeader()` to receive data and get the header of request that is sent from client.

```
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Read All Header Request</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Data sent from client is " + request.getParameter("txtDemo")+ "</h1>");
Enumeration list = request.getHeaderNames();
while (list.hasMoreElements()) {
    String s = list.nextElement().toString();
    out.println("<br>"+s+": <b>"+request.getHeader(s)+"</b>");
}
out.println("</body>");
out.println("</html>");
```

Introduction to Web Application

Java Servlet

Step 6. Run the index.html file, fill a string into a textbox and click on Send button.



Step 7. The request will send to server. This is a header of request that server received.

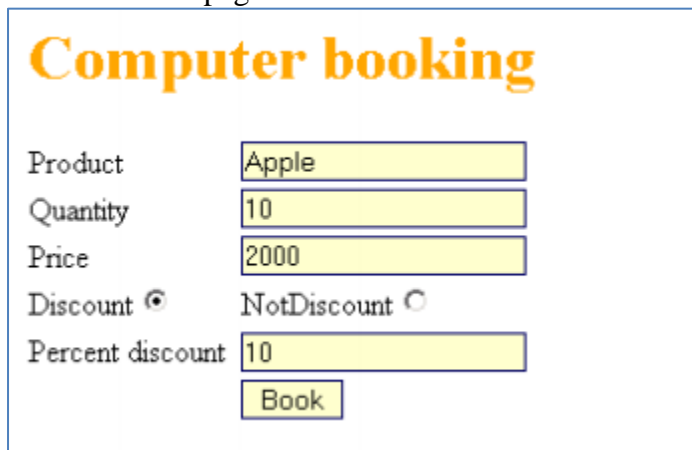


1.3 Do it Your self

In this exercise, you will create a project with booking product function. This project work as description as below:

- Client access to booking web page.
- Client will fill all fields in this page and submit it to server.
- Server will get all data that is sent from client and response all information to client.

Step 8. Create a index.html page with interface as below:



Introduction to Web Application

Java Servlet

Step 9. Create a Servlet to receive all data that is sent from client.

Step 10. Response all information to client.

Product

Name :Apple

Price :2000

Quantity :10

Total is :2000