

Todo Web Application

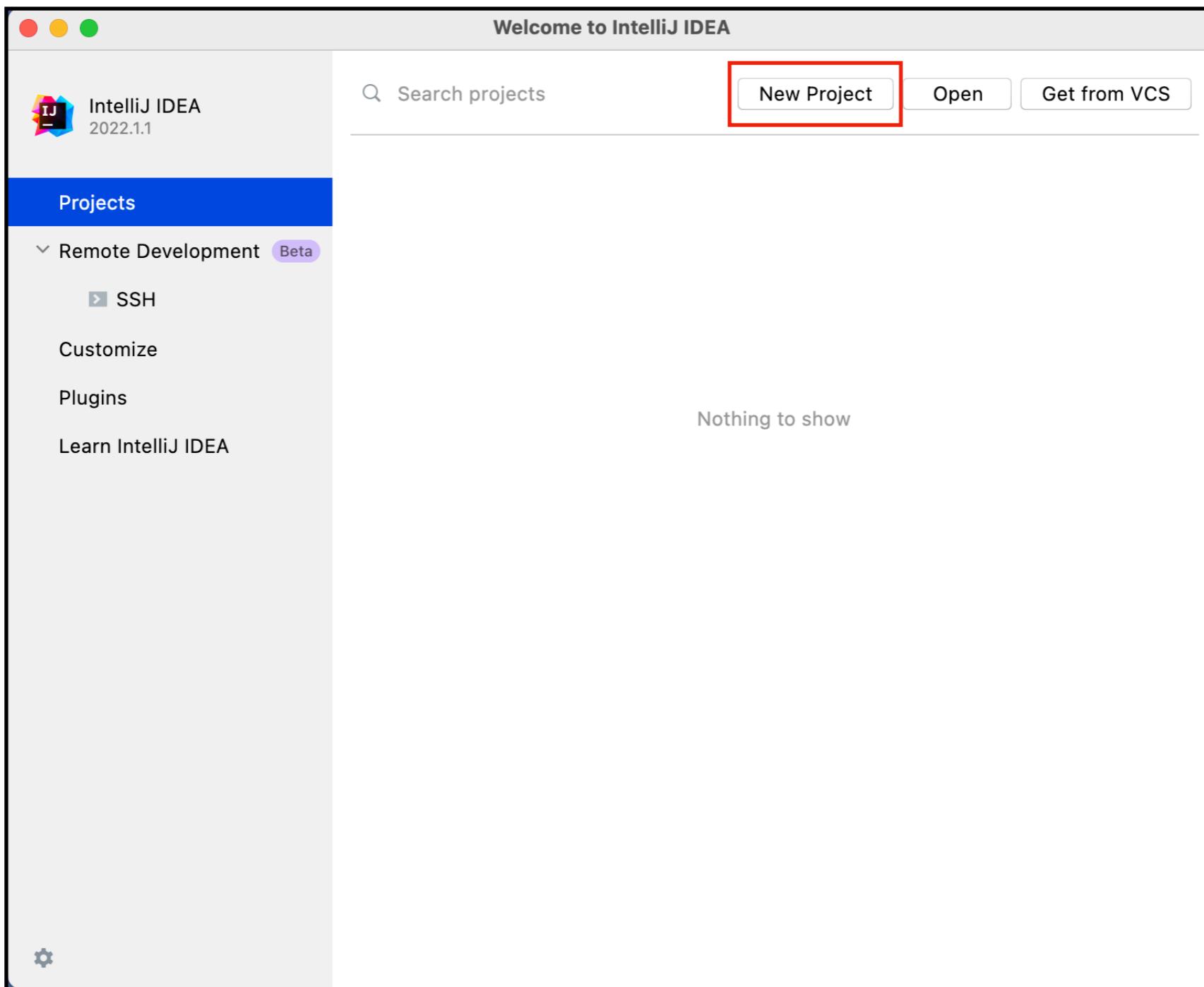
Lab 5

INT202

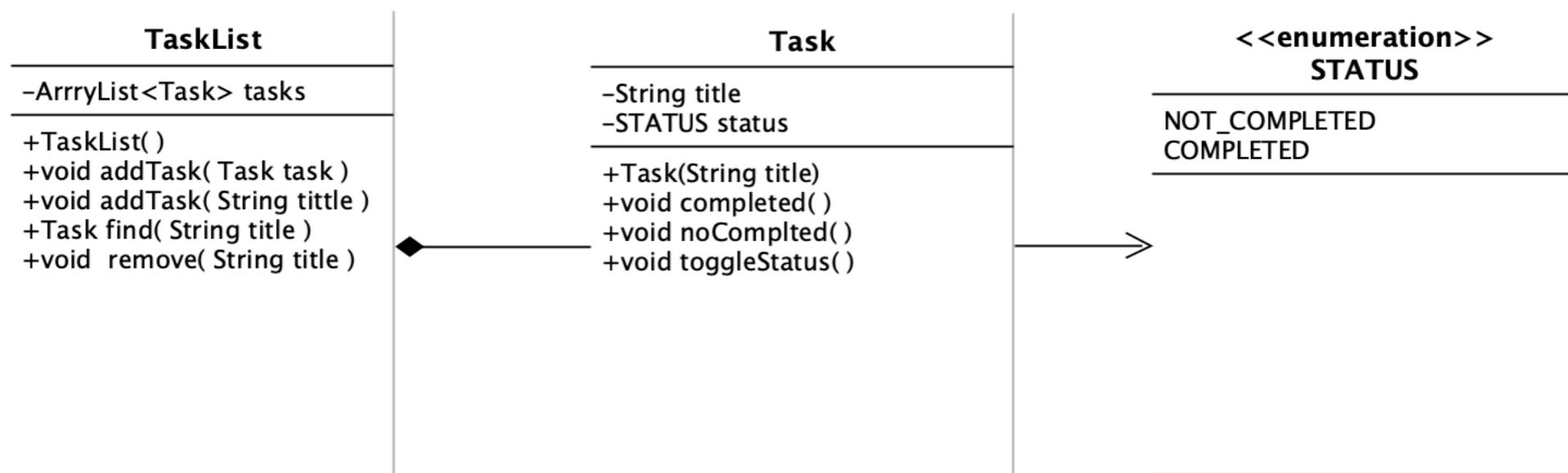
2/2023

Sanit Sirisawatvatana

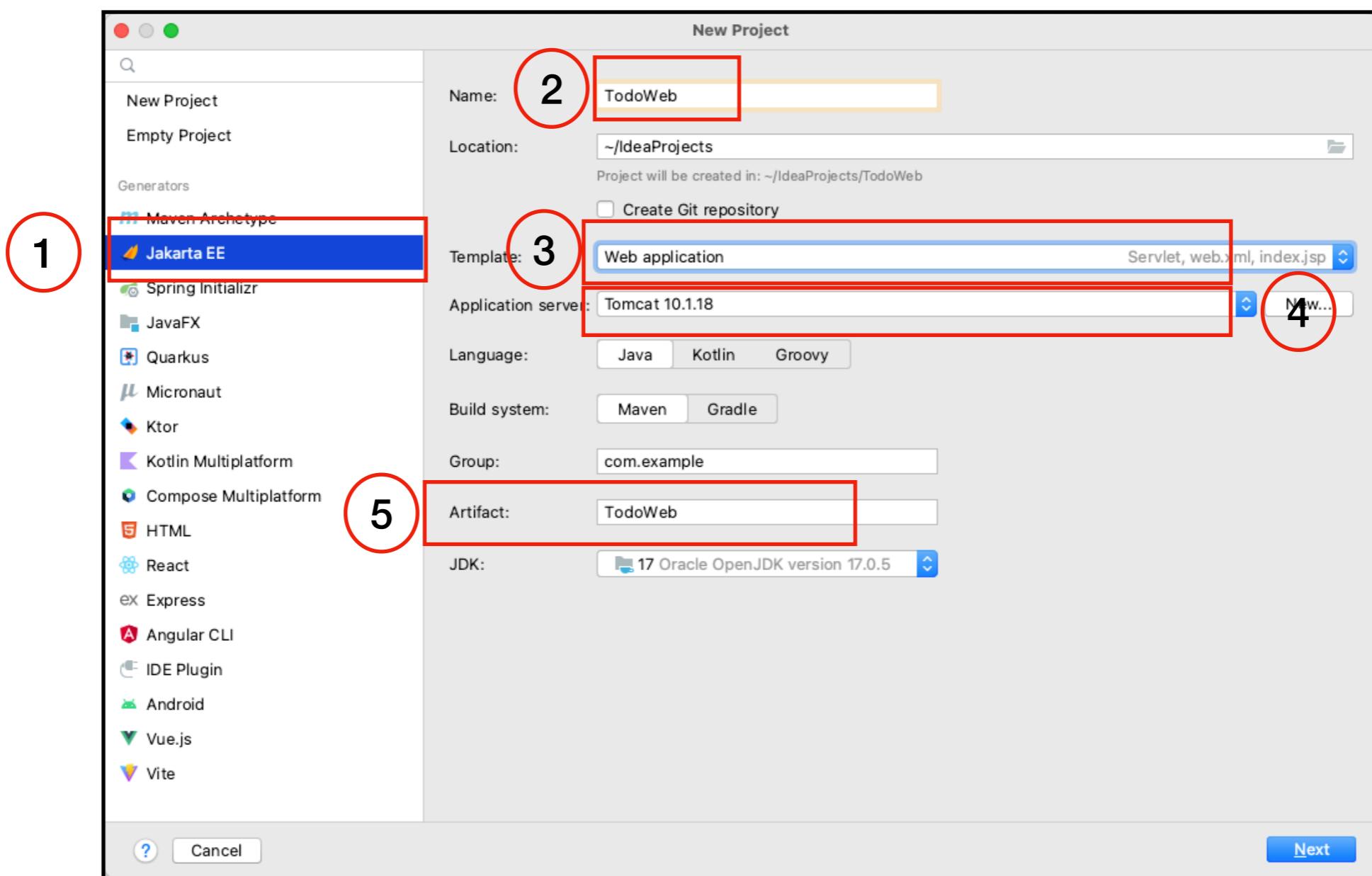
Project > New Project



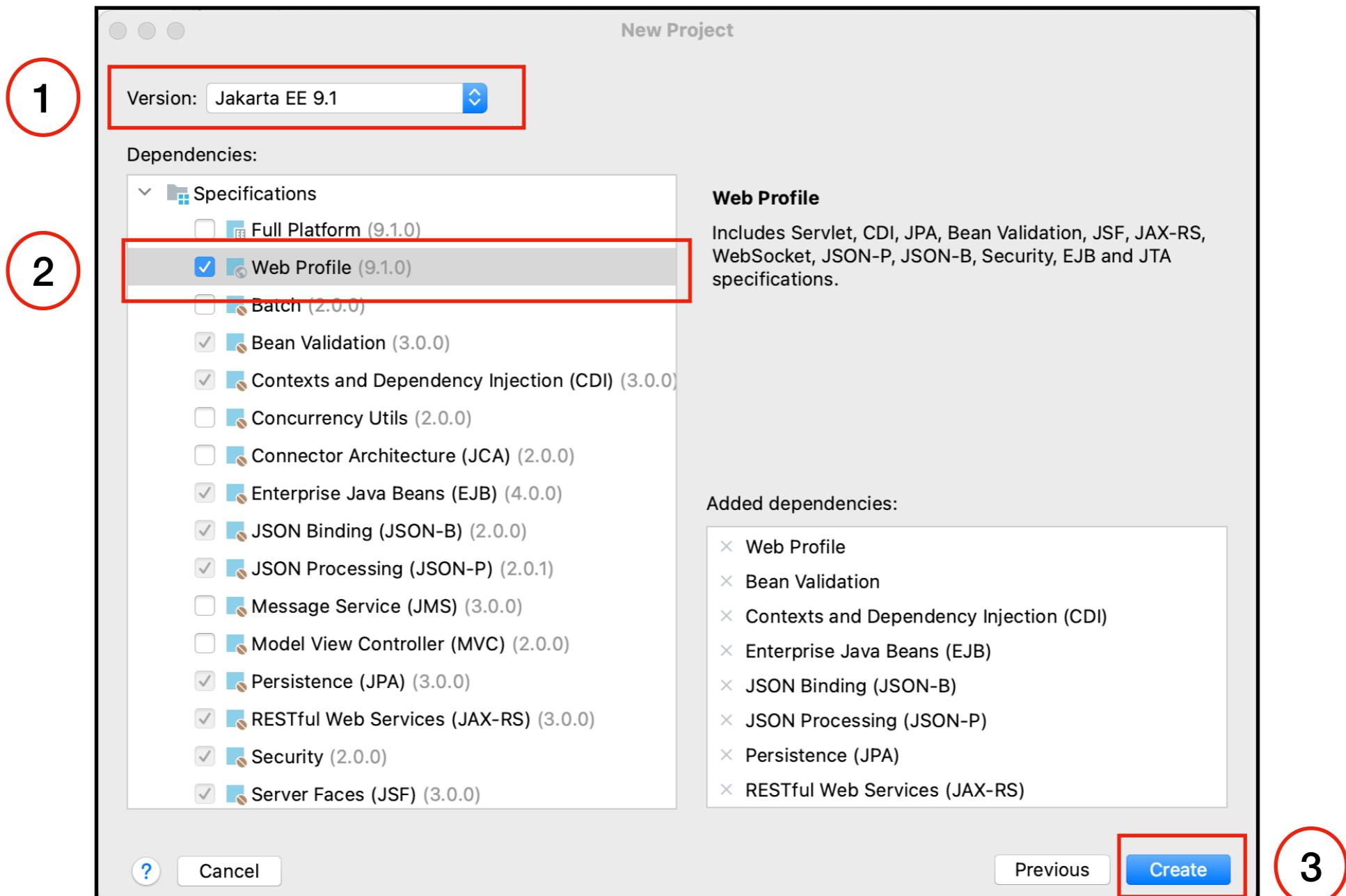
TodoWeb



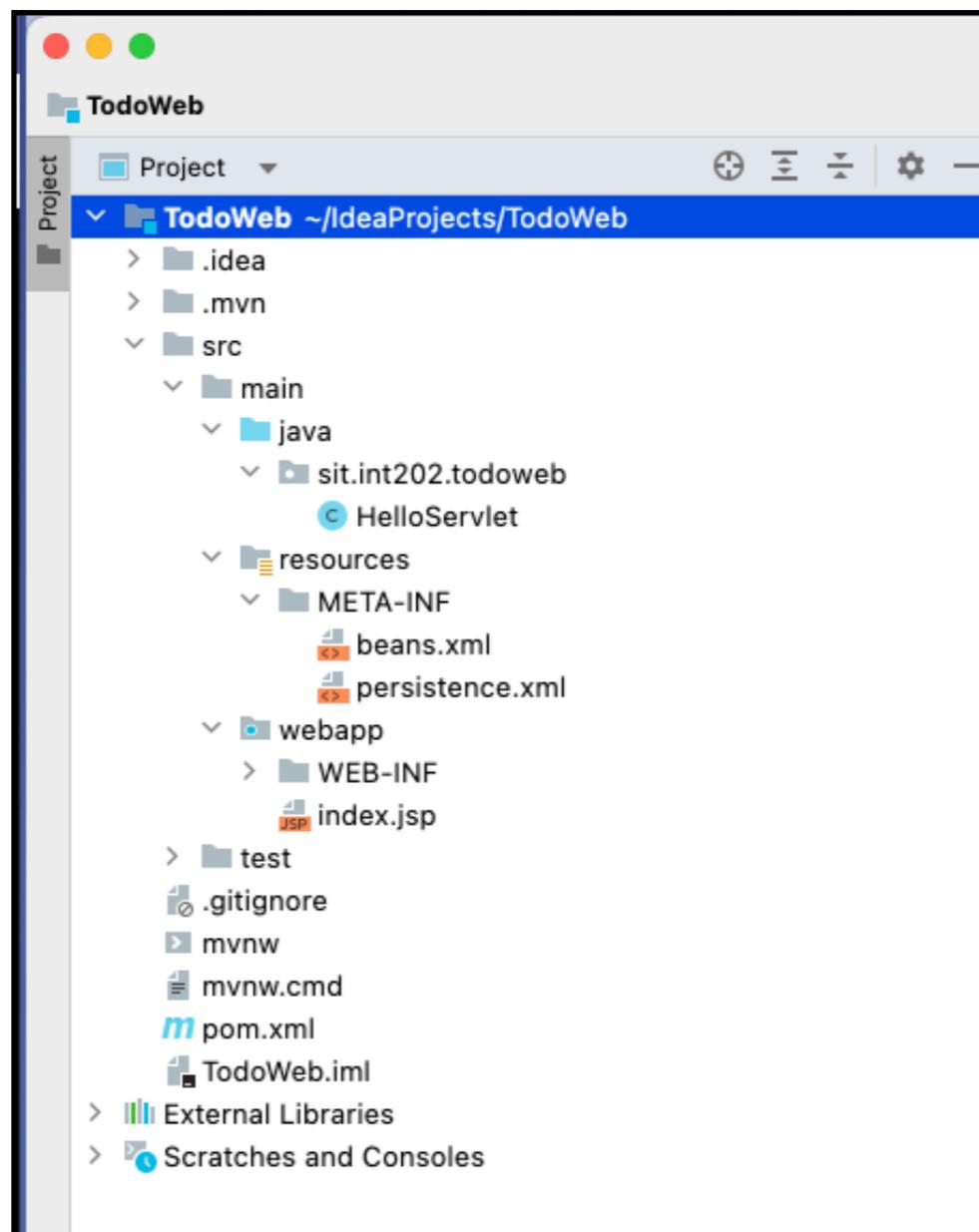
New Project



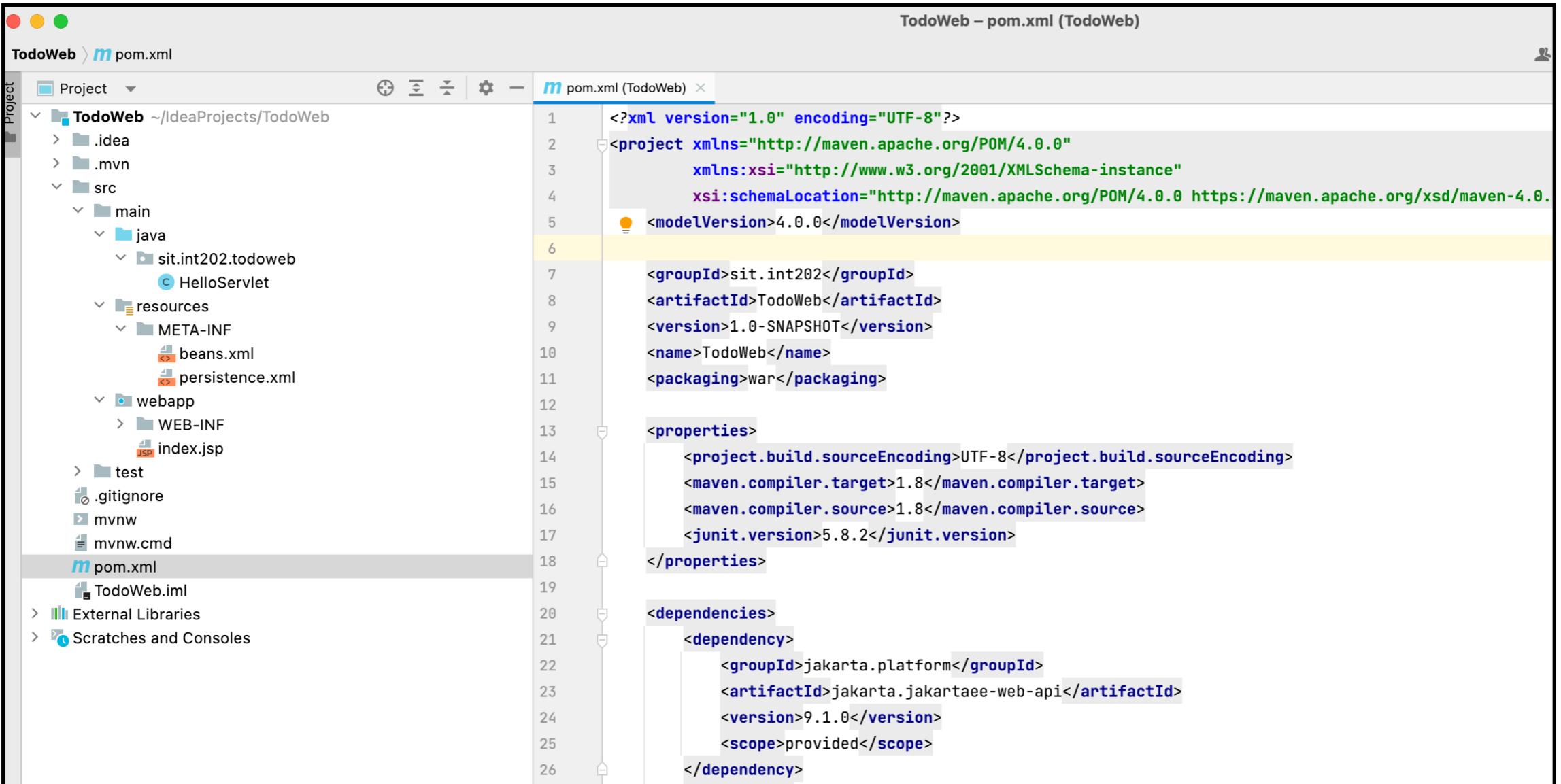
Dependencies



Project Structure



pom.xml

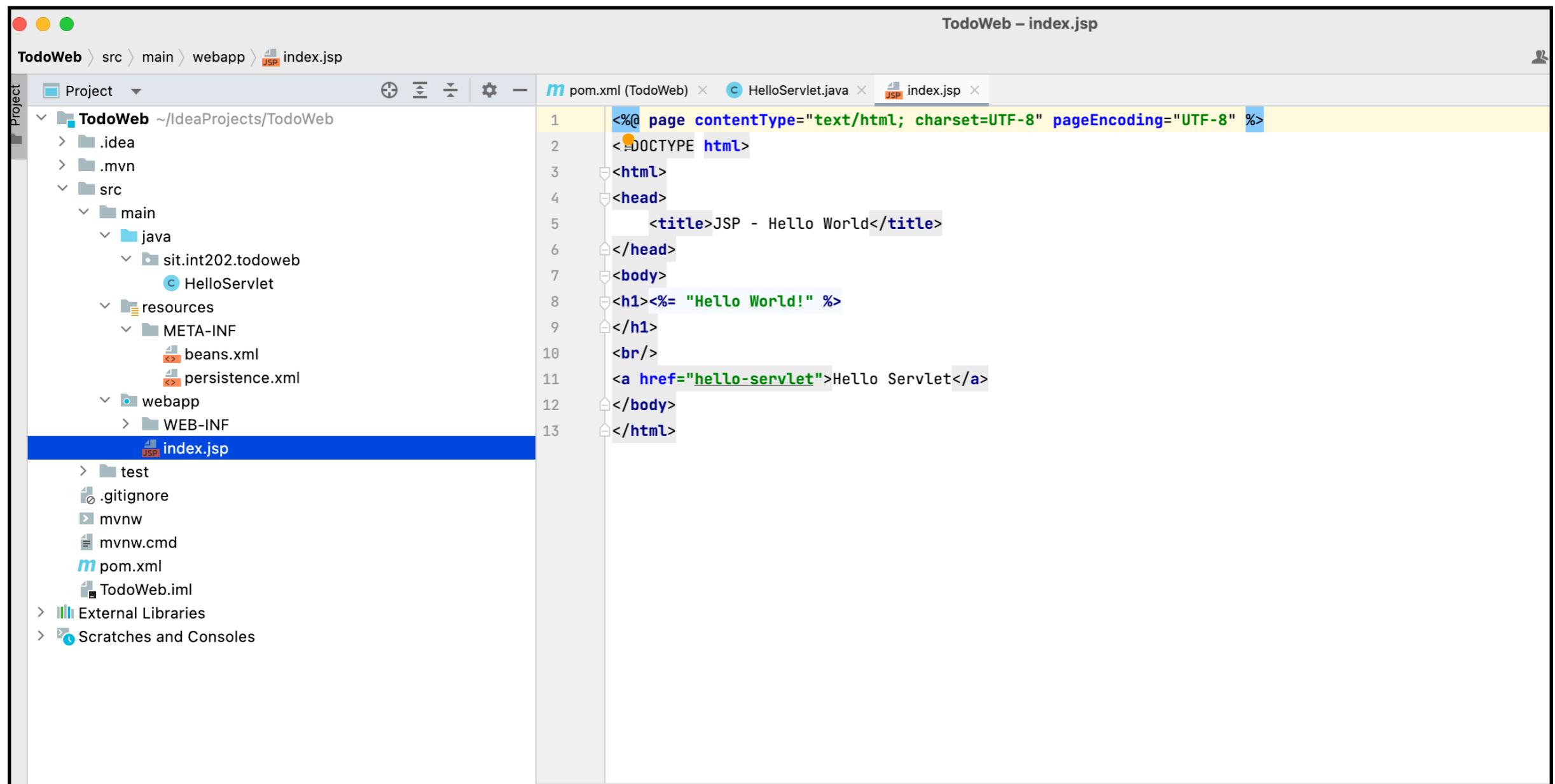


The screenshot shows the Maven Project Structure in an IDE. The left pane displays the project tree for 'TodoWeb' under 'IdeaProjects'. The right pane shows the content of the 'pom.xml' file.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>sit.int202</groupId>
    <artifactId>TodoWeb</artifactId>
    <version>1.0-SNAPSHOT</version>
    <name>TodoWeb</name>
    <packaging>war</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.target>1.8</maven.compiler.target>
        <maven.compiler.source>1.8</maven.compiler.source>
        <junit.version>5.8.2</junit.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>jakarta.platform</groupId>
            <artifactId>jakarta.jakartaee-web-api</artifactId>
            <version>9.1.0</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>

```

index.jsp



The screenshot shows the IntelliJ IDEA interface with the project 'TodoWeb' open. The left sidebar displays the project structure, including the 'src' directory with 'main' and 'java' sub-directories, which contain 'HelloServlet.java'. The 'resources' directory includes 'META-INF' with 'beans.xml' and 'persistence.xml'. The 'webapp' directory contains 'WEB-INF' and the file 'index.jsp' (highlighted with a blue bar). The top navigation bar shows tabs for 'pom.xml (TodoWeb)', 'HelloServlet.java', and 'index.jsp'. The code editor on the right shows the JSP code:

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>JSP - Hello World</title>
</head>
<body>
<h1><%= "Hello World!" %>
</h1>
<br/>
<a href="hello-servlet">Hello Servlet</a>
</body>
</html>
```

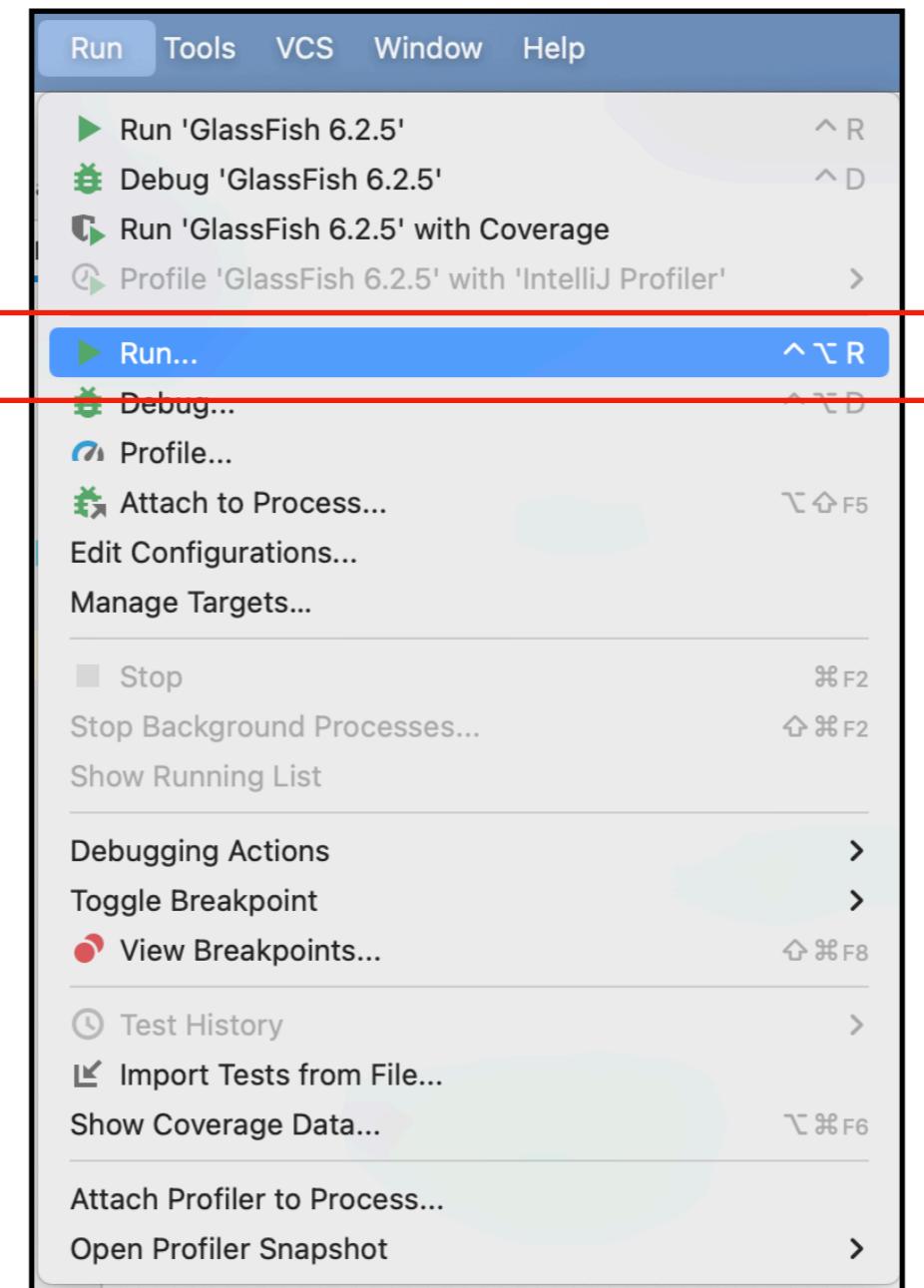
HelloServlet

The screenshot shows the IntelliJ IDEA interface with the following details:

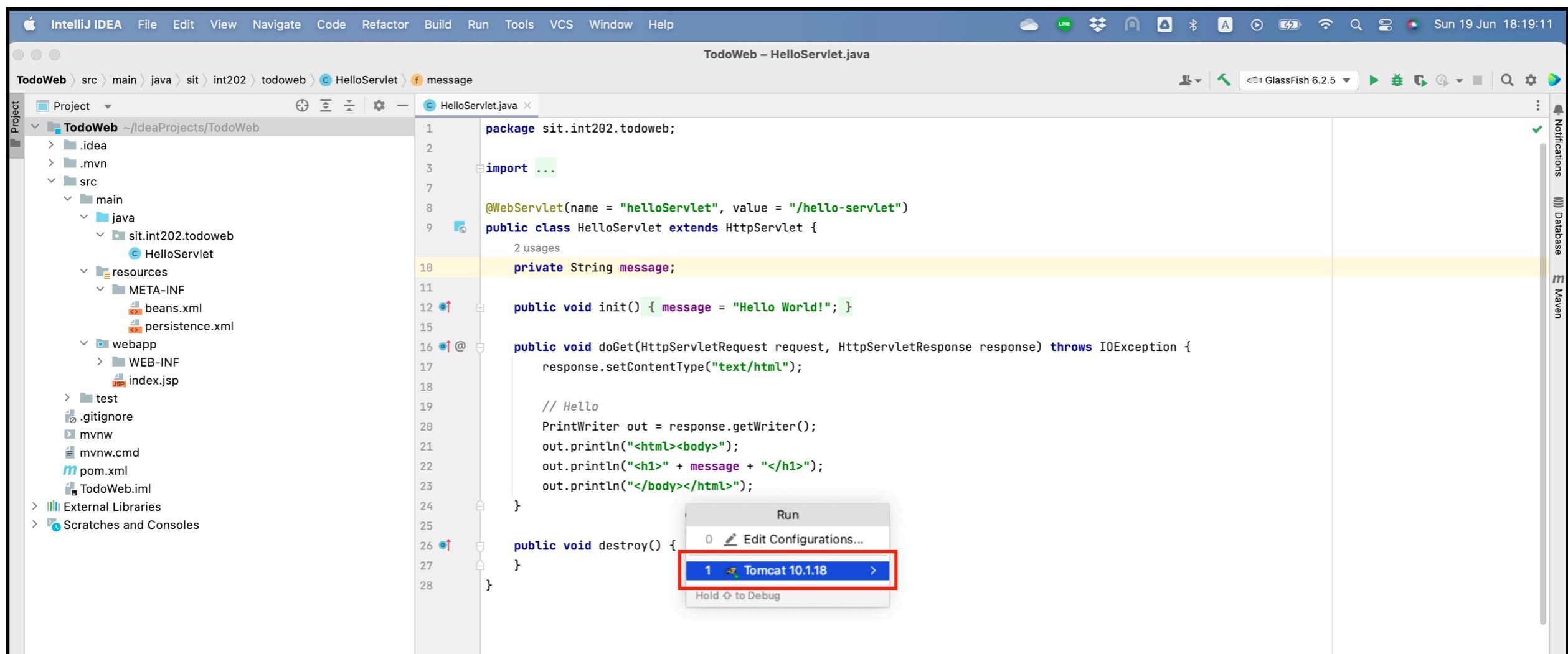
- Title Bar:** TodoWeb – HelloServlet.java
- Project Tool Window:** Shows the project structure under "TodoWeb". The "HelloServlet.java" file is selected in the "src/main/java/sit/int202/todoweb" package.
- Code Editor:** Displays the Java code for `HelloServlet.java`. The code defines a servlet that prints "Hello World!" to the response.

```
package sit.int202.todoweb;
import ...
@WebServlet(name = "helloServlet", value = "/hello-servlet")
public class HelloServlet extends HttpServlet {
    private String message;
    public void init() { message = "Hello World!"; }
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        response.setContentType("text/html");
        // Hello
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");
        out.println("</body></html>");
    }
    public void destroy() {
}
```

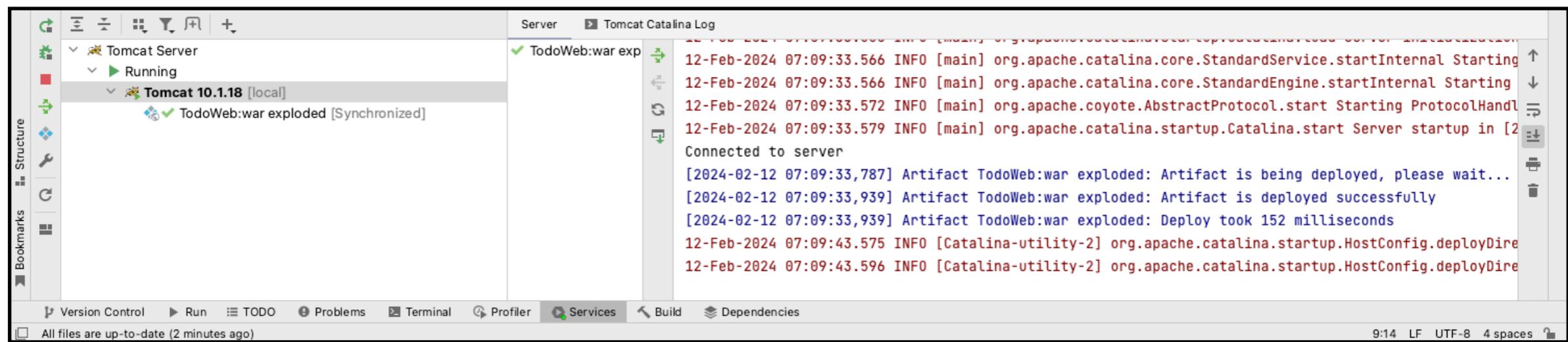
Deploy and Run



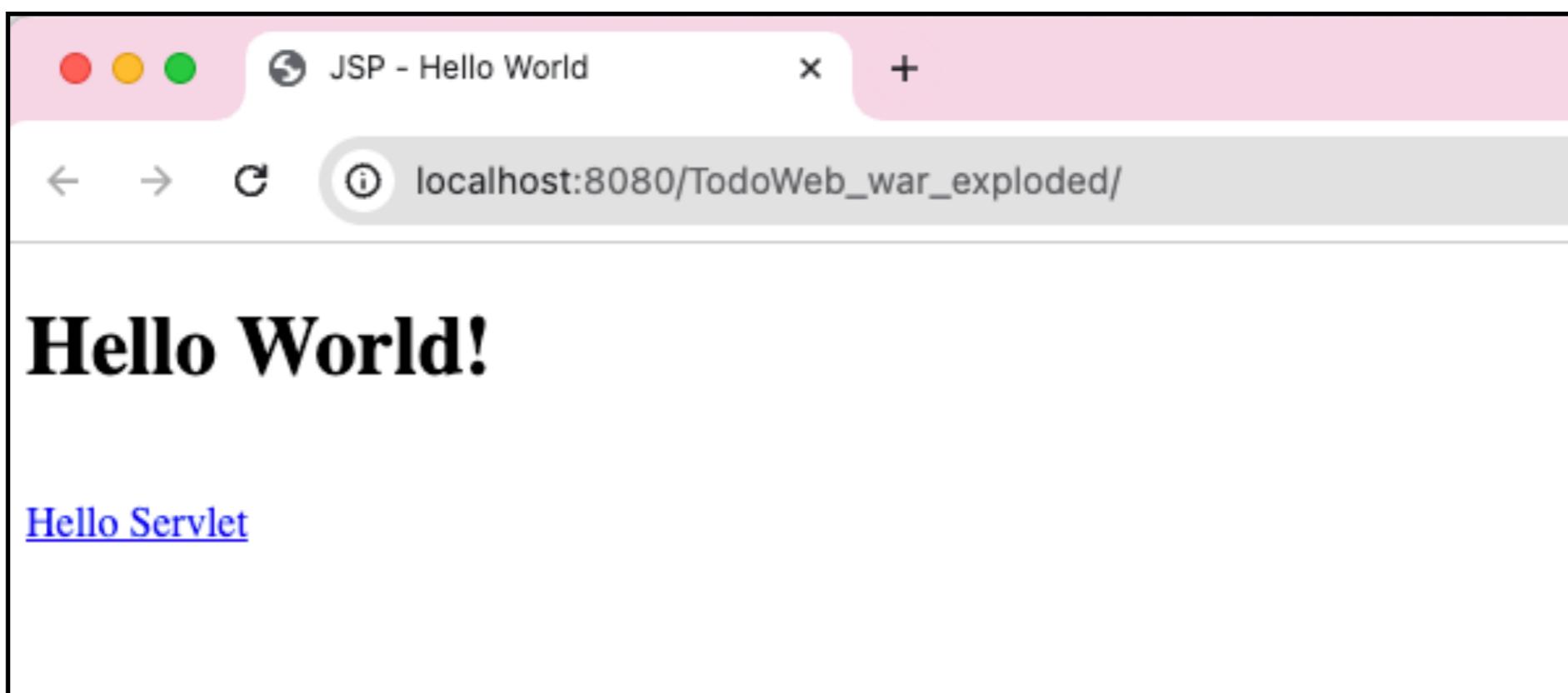
Choose Application Server: Tomcat



Tomcat Server



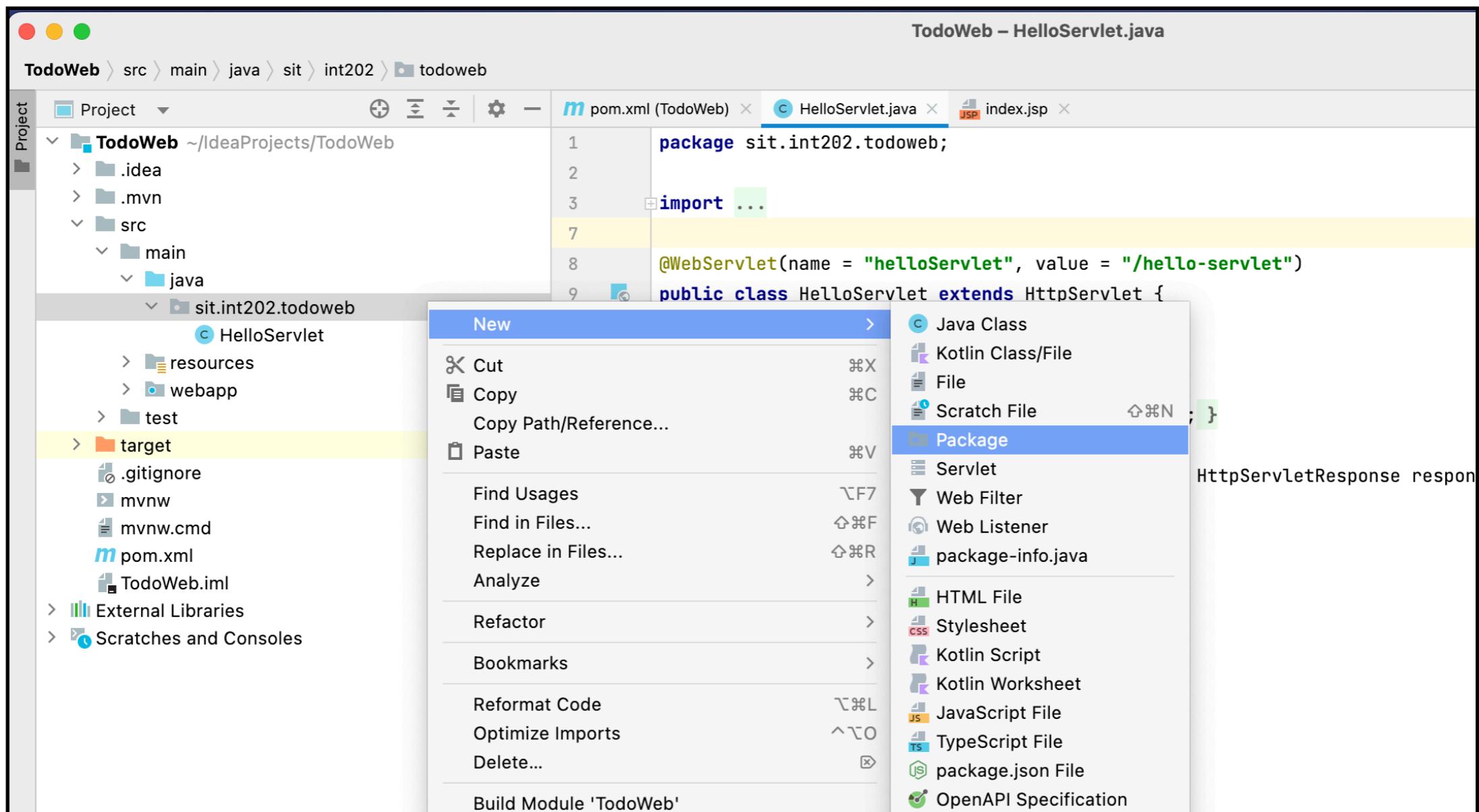
Welcome page



hello-servlet



New > Package



New > Package > models

The screenshot shows the IntelliJ IDEA interface with the following details:

- Title Bar:** TodoWeb – HelloServlet.java
- Project Bar:** TodoWeb > src > main > java > sit > int202 > todoweb
- Toolbars:** Project, pom.xml (TodoWeb), HelloServlet.java (selected), index.jsp
- Code Editor:** The code for `HelloServlet.java` is displayed. The cursor is at the end of the line `out.println("sit.int202.todoweb.models");`. A red box highlights the completion dropdown menu, which shows "New Package".
- Project Explorer:** Shows the project structure with the package `sit.int202.todoweb` selected.
- Bottom Status Bar:** Shows line numbers 1 through 27.

```
TodoWeb – HelloServlet.java

TodoWeb > src > main > java > sit > int202 > todoweb
Project pom.xml (TodoWeb) HelloServlet.java index.jsp

package sit.int202.todoweb;
import ...
@WebServlet(name = "helloServlet", value = "/hello-servlet")
public class HelloServlet extends HttpServlet {
    private String message;
    public void init() { message = "Hello World!"; }
    public void doGet(HttpServletRequest request, HttpServletResponse response) {
        response.setContentType("text/html");
        // Hello
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("New Package");
        out.println("sit.int202.todoweb.models");
    }
    public void destroy() {
}
```

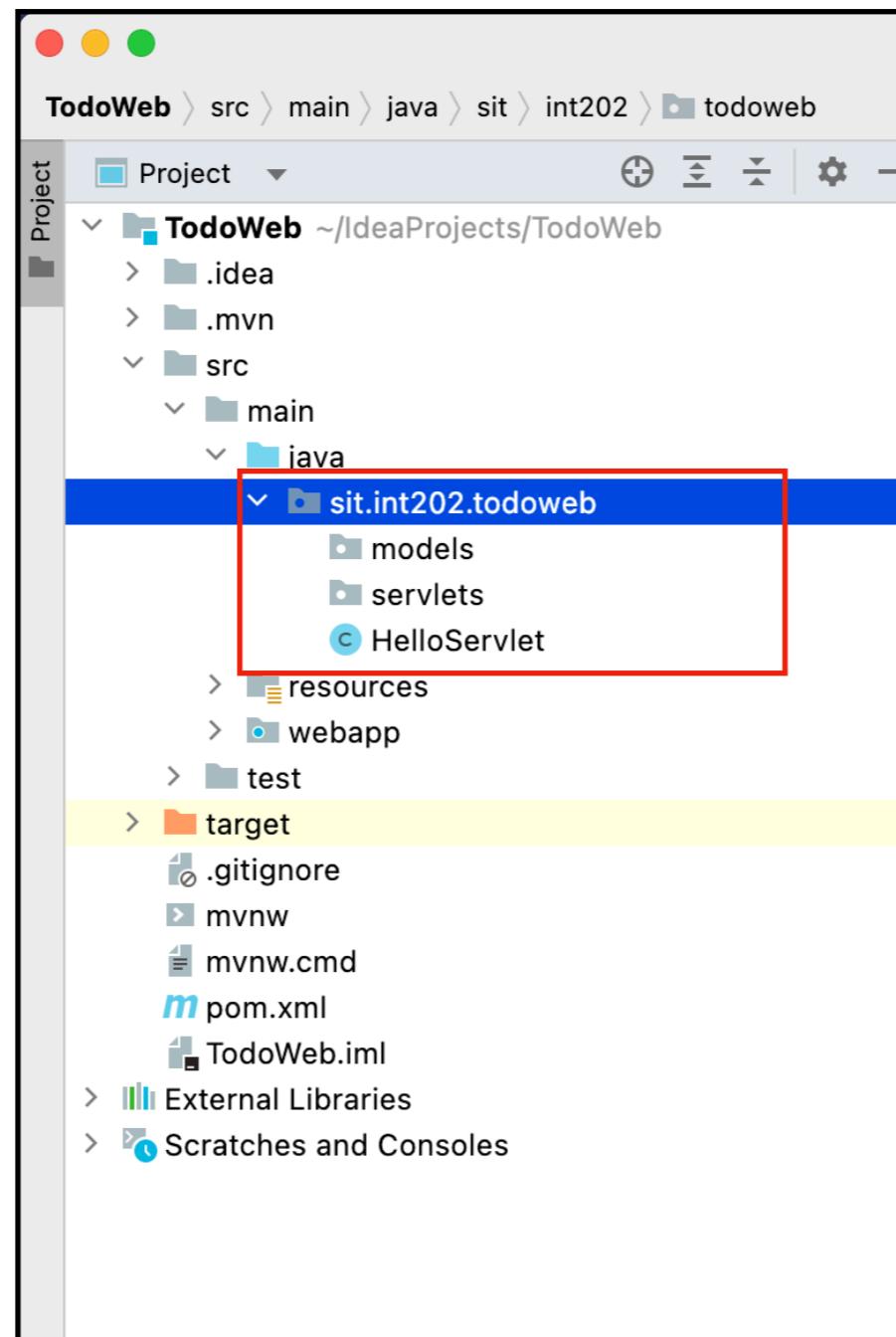
New > Package > servlets

The screenshot shows the IntelliJ IDEA interface with the following details:

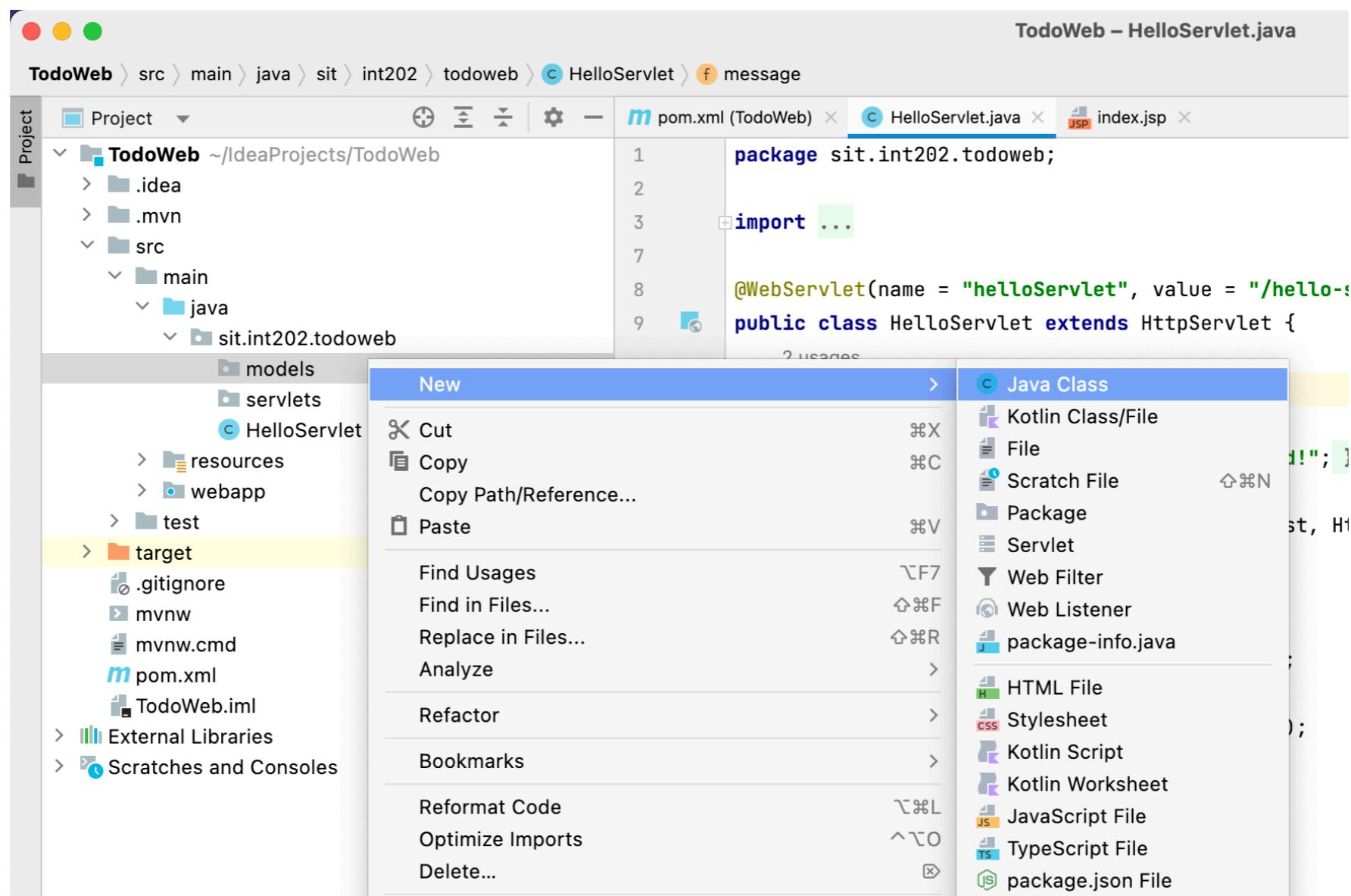
- Project Bar:** TodoWeb > src > main > java > sit > int202 > todoweb
- Toolbars:** Project, pom.xml (TodoWeb), HelloServlet.java, index.jsp
- Code Editor:** TodoWeb – HelloServlet.java
- Code Content:**

```
1 package sit.int202.todoweb;
2
3 import ...
4
5 @WebServlet(name = "helloServlet", value = "/hello-servlet")
6 public class HelloServlet extends HttpServlet {
7     private String message;
8
9     public void init() { message = "Hello World!"; }
10
11    public void doGet(HttpServletRequest request, HttpServletResponse response) {
12        response.setContentType("text/html");
13
14        // Hello
15        PrintWriter out = response.getWriter();
16        out.println("<html><body>");
17        out.print("Hello World!");
18        out.println("</body></html>");
19    }
20
21    public void destroy() {
22    }
23}
```
- Context Menu:** A context menu is open at the bottom right of the code editor, listing "New Package" and "sit.int202.todoweb.servlets". The "sit.int202.todoweb.servlets" option is highlighted with a red border.
- File Explorer:** Shows the project structure with TodoWeb, src, target, and various files like pom.xml and TodoWeb.iml.

models & servlets



New > Java Class > Task



Task.java

The screenshot shows an IDE interface with a code editor displaying a Java file named `HelloServlet.java`. The code defines a servlet named `HelloServlet` that prints "Hello World!" to the response. A context menu is open over the word `Task` in the code, listing options: `New Java Class`, `Task`, `Class` (which is selected), `Interface`, `Enum`, and `@Annotation`.

```
TodoWeb – HelloServlet.java
TodoWeb > src > main > java > sit > int202 > todoweb > models
HelloServlet.java

1 package sit.int202.todoweb;
2
3 import ...
4
5
6 @.WebServlet(name = "helloServlet", value = "/hello-servlet")
7
8 public class HelloServlet extends HttpServlet {
9     private String message;
10
11     public void init() { message = "Hello World!"; }
12
13     public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
14         response.setContentType("text/html");
15
16         // Help
17         PrintWriter out = response.getWriter();
18         out.println(message);
19         out.println("
```

Task.java

The screenshot shows the IntelliJ IDEA interface with the following details:

- Title Bar:** TodoWeb – Task.java
- Toolbar:** Standard IntelliJ toolbar with icons for file operations.
- Project Bar:** Shows the current project path: TodoWeb > src > main > java > sit > int202 > todoweb > models > Task.
- Toolbars:** Standard IntelliJ toolbars for navigation and search.
- Code Editor:** The editor displays the `Task.java` file content:

```
package sit.int202.todoweb.models;

public class Task {
```

The code editor has a yellow background for the entire code area.
- Project Explorer:** On the left, the Project Explorer shows the project structure:
 - TodoWeb** (Project root)
 - src**
 - main**
 - java**
 - sit.int202.todoweb**
 - models**
 - Task** (Selected)
 - HelloServlet**
 - resources**
 - test**
 - target**
 - External Libraries**
 - Scratches and Consoles**
 - Services**: A small section at the bottom.

Task.java

```
1 package sit.int202.todoweb.models;  
2  
3 public class Task {  
4     10 usages  
5     public enum Status {  
6         5 usages  
7             NOT_COMPLETED,  
8             2 usages  
9                 COMPLETED  
10            }  
11            4 usages  
12            private String title ;  
13            8 usages  
14            private Status status = Status.NOT_COMPLETED;  
15  
16            public Task(String title) {  
17                this.title = title;  
18            }  
19  
20            public String getTitle() {  
21                return title;  
22            }  
23  
24            public void setTitle(String title) {  
25                this.title = title;  
26            }  
27  
28            public Status getStatus() {  
29                return status;  
30            }  
31  
32            public void setStatus(Status status) {  
33                this.status = status;  
34            }  
35        }
```

Task.java

```
31
32     public void completed(){
33         this.status = Status.COMPLETED ;
34     }
35
36     public void noCompleted() {
37         this.status = Status.NOT_COMPLETED;
38     }
39
40     public void toggleStatus(){
41         if(this.status==Status.NOT_COMPLETED){
42             this.status = Status.COMPLETED ;
43         } else {
44             this.status = Status.NOT_COMPLETED ;
45         }
46     }
47
48     @Override
49     public String toString() {
50         return "Task : "+title+"\nStatus : "+(status==Status.NOT_COMPLETED ?"Not completed":"Completed");
51     }
52 }
```

TaskList.java

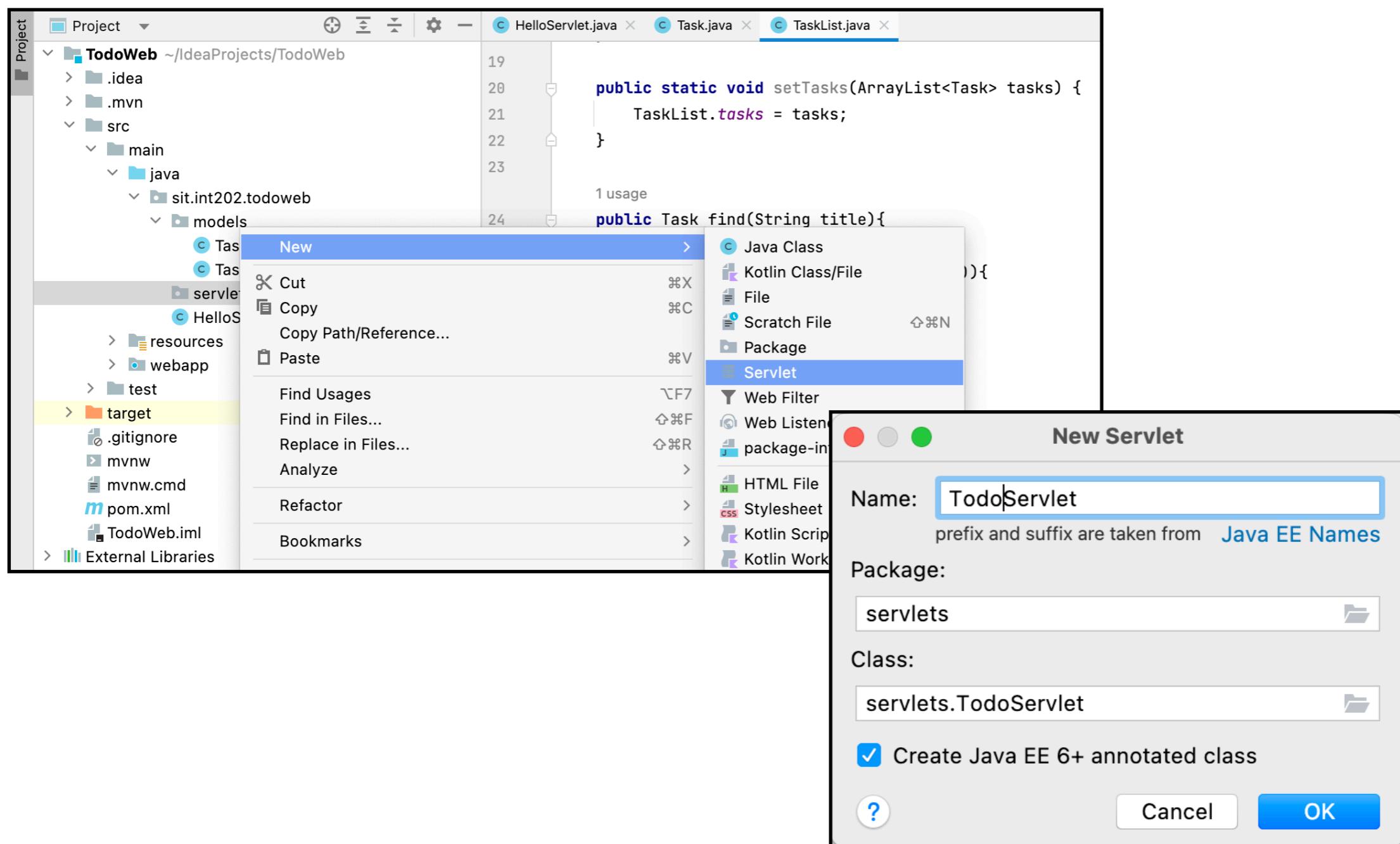
```
1 package sit.int202.todoweb.models;
2
3 import java.util.ArrayList;
4
5 4 usages
6 public class TaskList {
7     9 usages
8     private static ArrayList<Task> tasks = new ArrayList<>();
9
10    static {
11        tasks.add(new Task( title: "Review Java") );
12        tasks.add(new Task( title: "Intro Web App") );
13    }
14
15    public void addTask(Task newTask){
16        tasks.add(newTask) ;
17    }
18
19    public void addTask(String title){
20        tasks.add(new Task(title)) ;
21    }
22
23    1 usage
24    public ArrayList<Task> getTasks() {
25        return tasks;
26    }
27
28    public static void setTasks(ArrayList<Task> tasks) {
29        TaskList.tasks = tasks;
30    }
31
```

Static block

TaskList.java

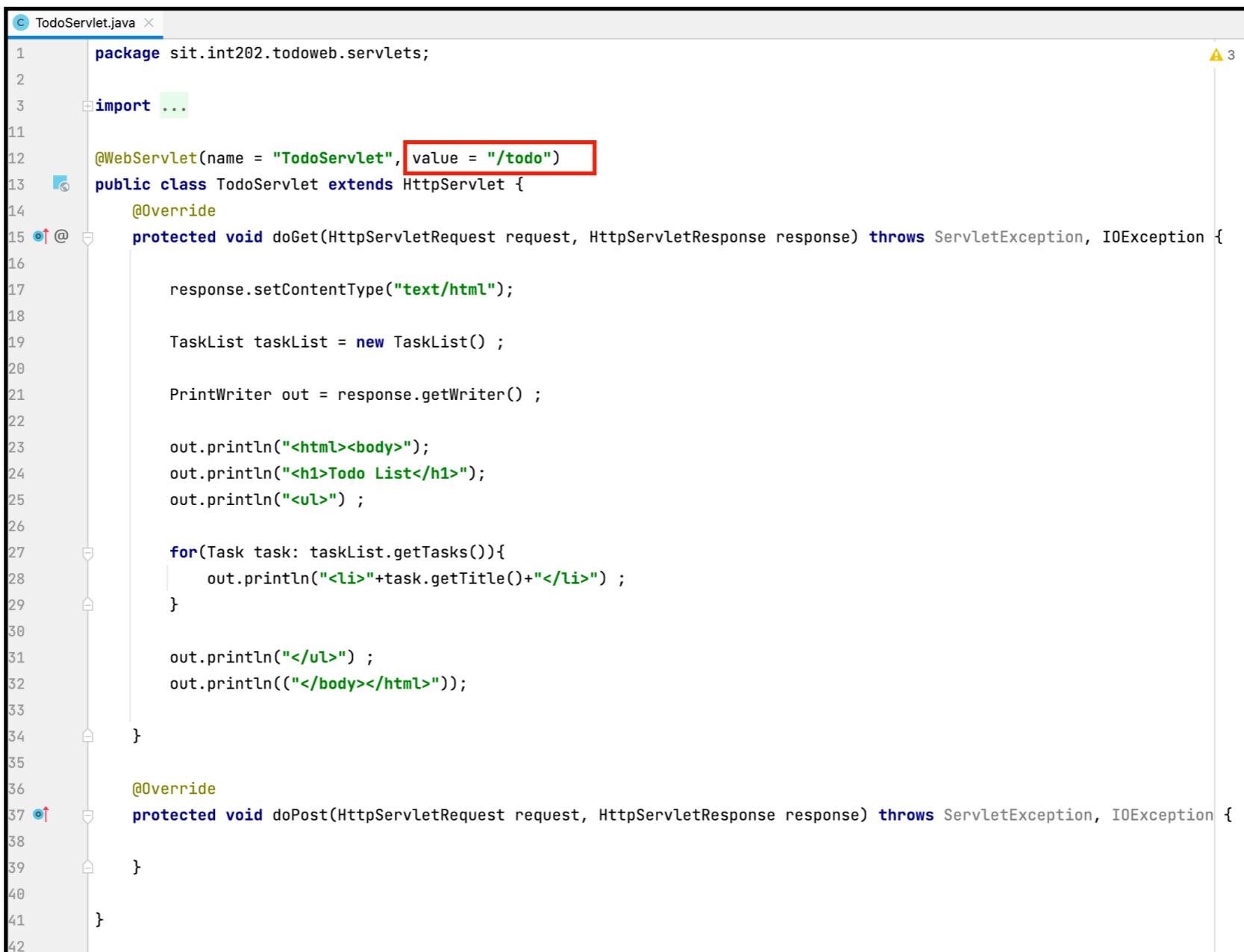
```
1 usage
29  public Task find(String title){
30      for(Task task: tasks){
31          if(task.getTitle().equals(title)){
32              return task ;
33          }
34      }
35      return null ;
36  }
37
38  public void remove(String title){
39      Task task = this.find(title);
40      tasks.remove(task) ;
41  }
42
43  @Override
44  public String toString() {
45      StringBuilder task_format = new StringBuilder("-----\n");
46      task_format.append("List tasks: \n");
47      task_format.append("-----\n") ;
48      for(Task task: tasks){
49          task_format.append(task.toString()).append('\n');
50      }
51      return task_format.toString();
52  }
53
54
55 }
```

New > Servlet > TodoServlet



TodoServlet

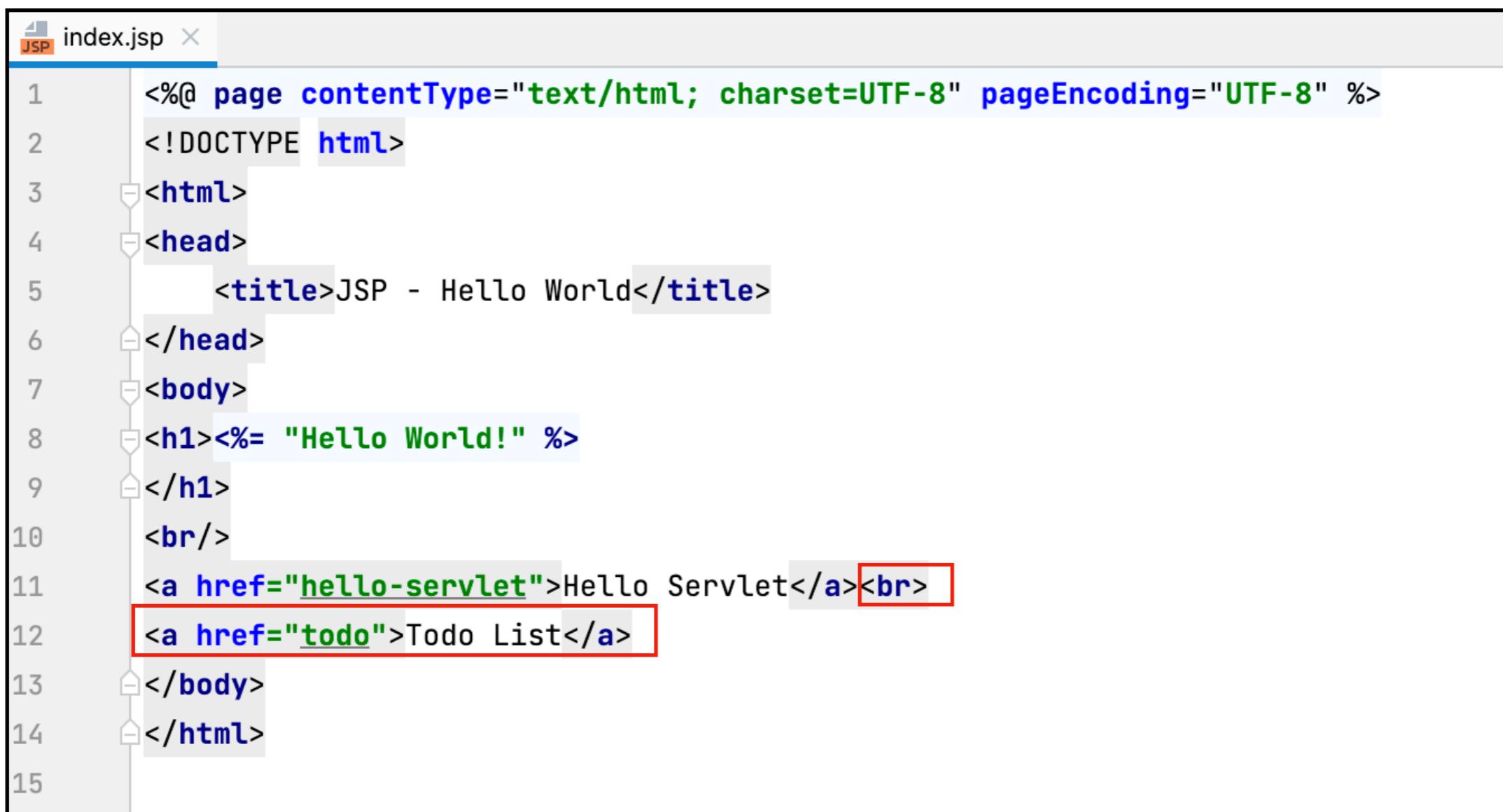
- Set the value (urlPattern) to "/todo"



```
TodoServlet.java
1 package sit.int202.todoweb.servlets;
2
3 import ...
4
5 @WebServlet(name = "TodoServlet", value = "/todo")
6 public class TodoServlet extends HttpServlet {
7     @Override
8     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
9
10         response.setContentType("text/html");
11
12         TaskList taskList = new TaskList();
13
14         PrintWriter out = response.getWriter();
15
16         out.println("<html><body>");
17         out.println("<h1>Todo List</h1>");
18         out.println("<ul>");
19
20         for(Task task: taskList.getTasks()){
21             out.println("<li>" + task.getTitle() + "</li>");}
22
23         out.println("</ul>");out.println("</body></html>");}
24
25     }
26
27     @Override
28     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}}
```

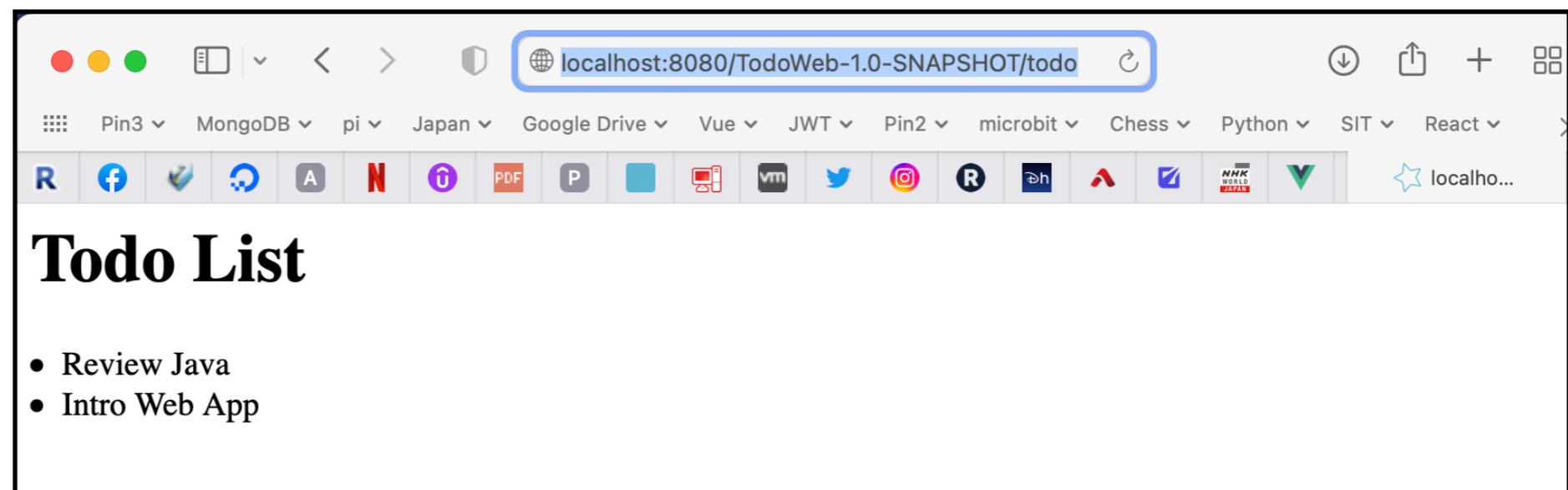
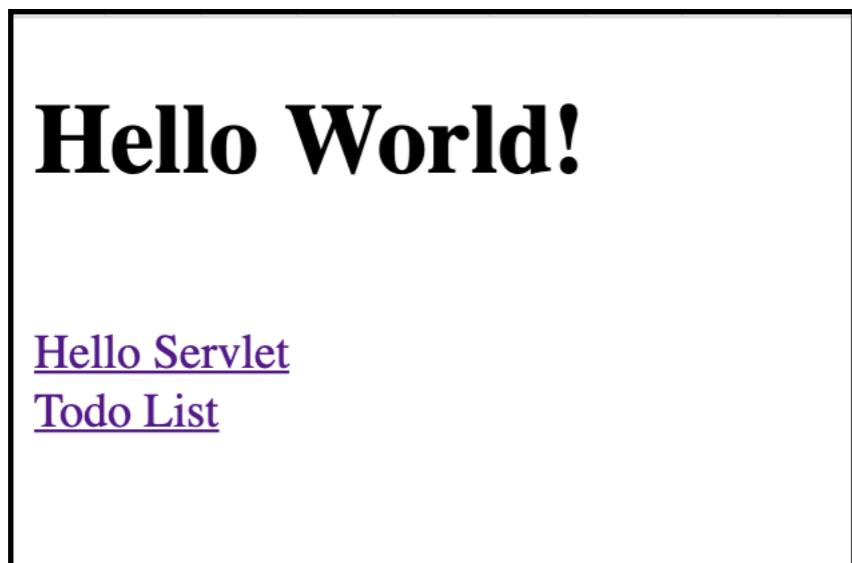
index.jsp

- Add HTML link to link to the servlet ("todo")



```
index.jsp
1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>JSP - Hello World</title>
6 </head>
7 <body>
8   <h1><%= "Hello World!" %>
9   </h1>
10  <br/>
11  <a href="hello-servlet">Hello Servlet</a><br>
12  <a href="todo">Todo List</a>
13 </body>
14 </html>
```

Run the Todo Web App



TodoServlet2

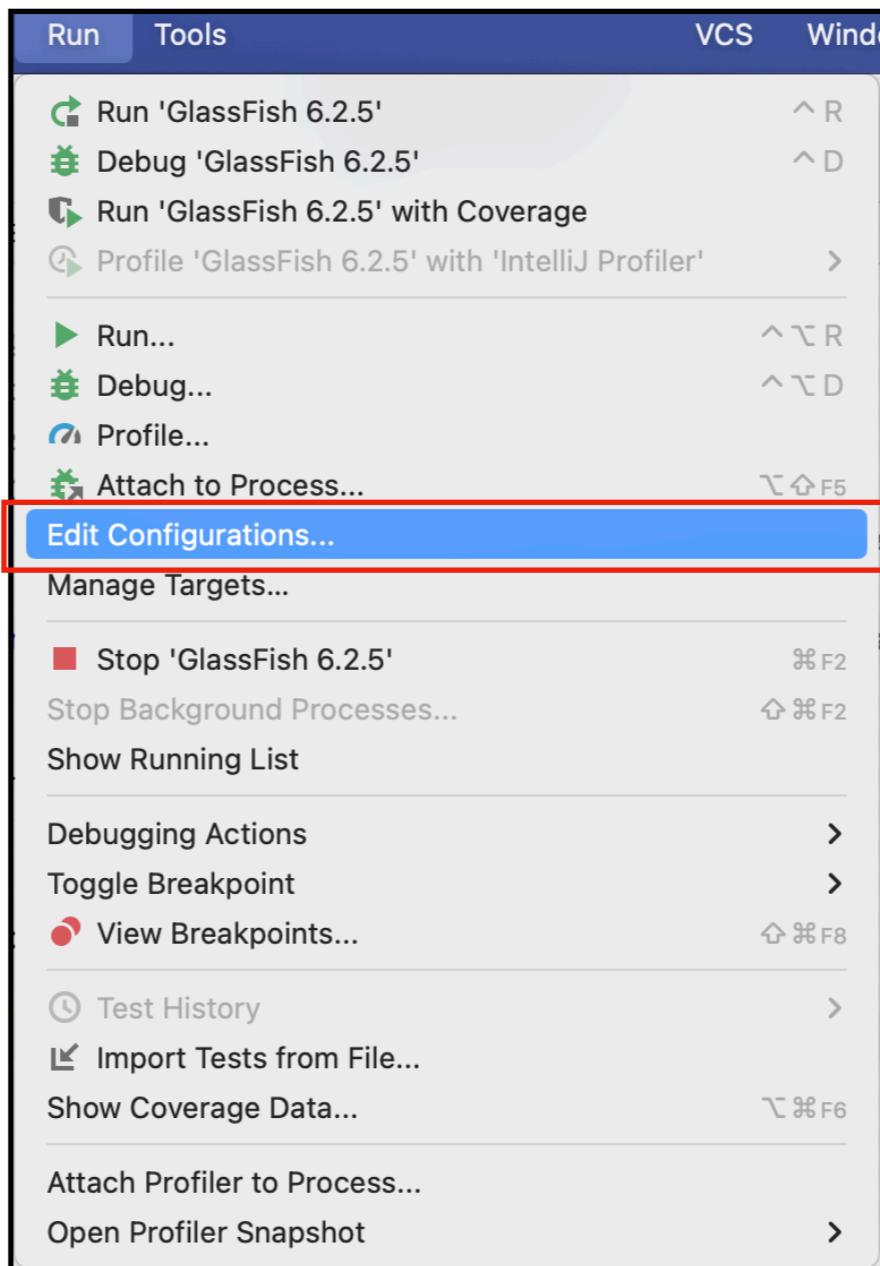
- Use the **RequestDispatcher** to dispatch the request to a JSP called "/todo.jsp".



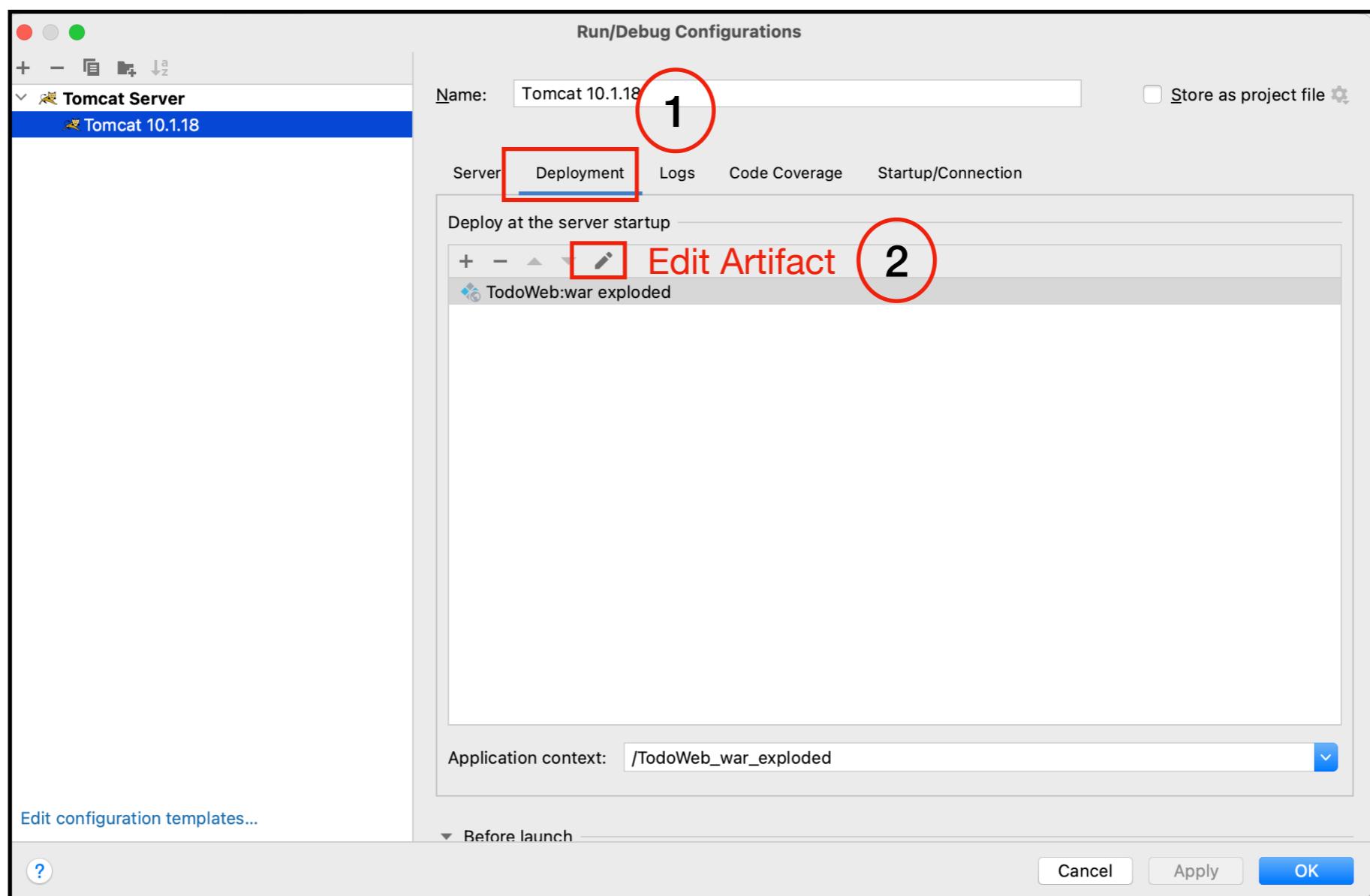
```
TodoServlet2.java
1 package sit.int202.todoweb.servlets;
2
3 import ...
4
5
6 @WebServlet(name = "TodoServlet2", value = "/todo2")
7 public class TodoServlet2 extends HttpServlet {
8     @Override
9     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10         TaskList taskList = new TaskList();
11
12         request.setAttribute("tasks", taskList.getTasks());
13
14         RequestDispatcher dispatcher = request.getRequestDispatcher("/todo.jsp");
15         dispatcher.forward(request, response);
16     }
17
18     @Override
19     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
20     }
21
22
23 }
```

Add maven library

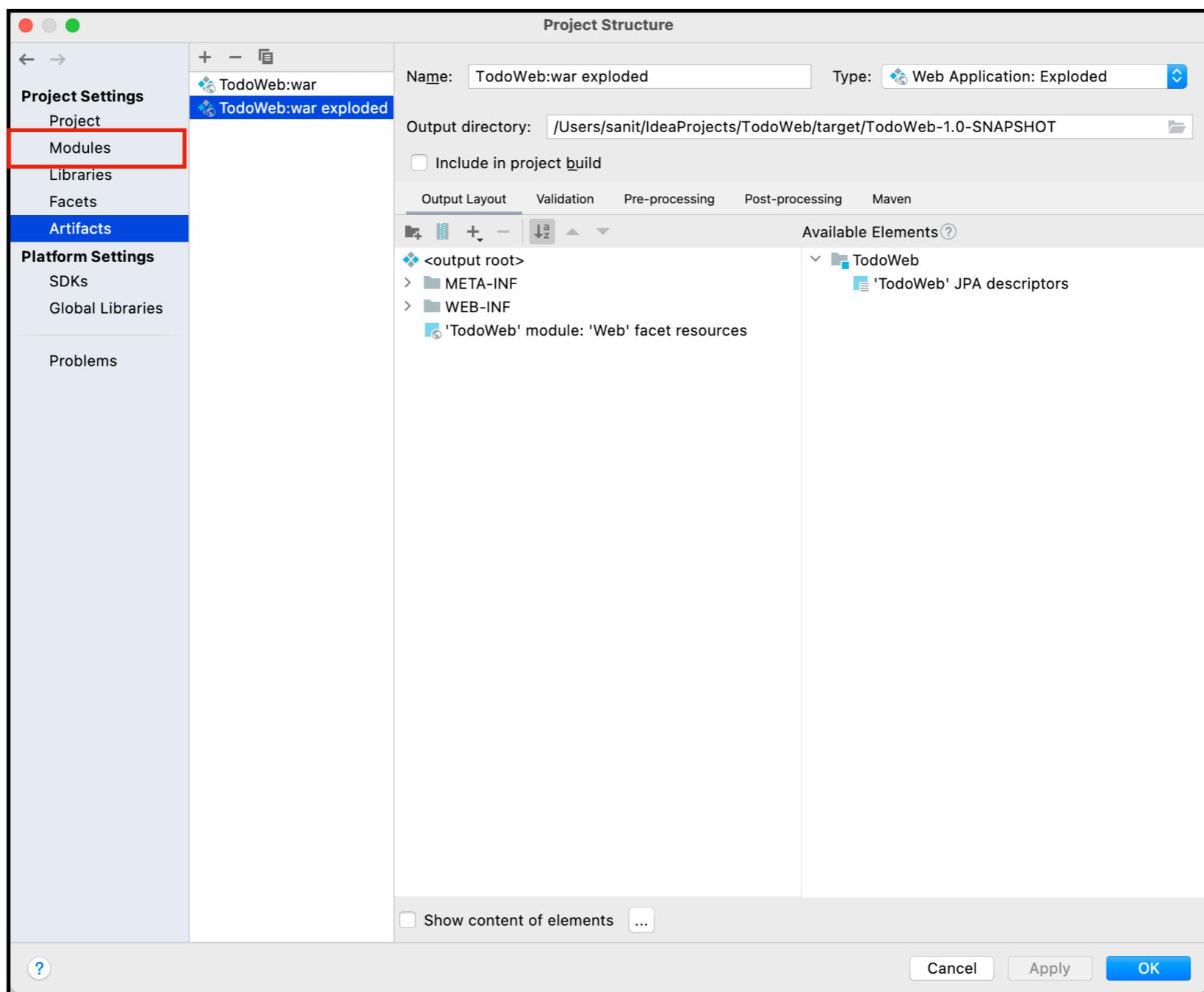
- Add JSP Standard Tag Library (JSTL) to IntelliJ



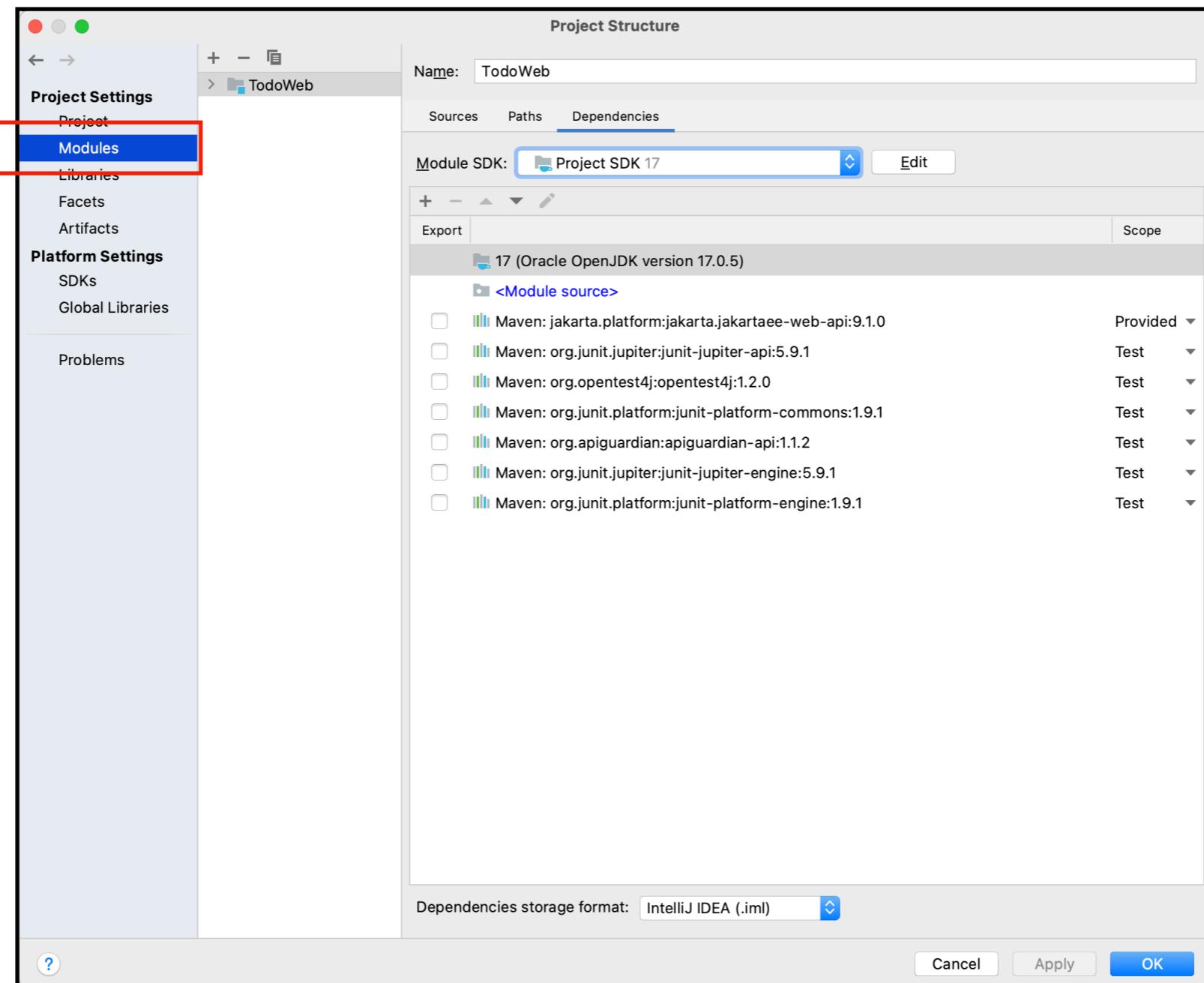
Glassfish > Deployment > Edit Artifact



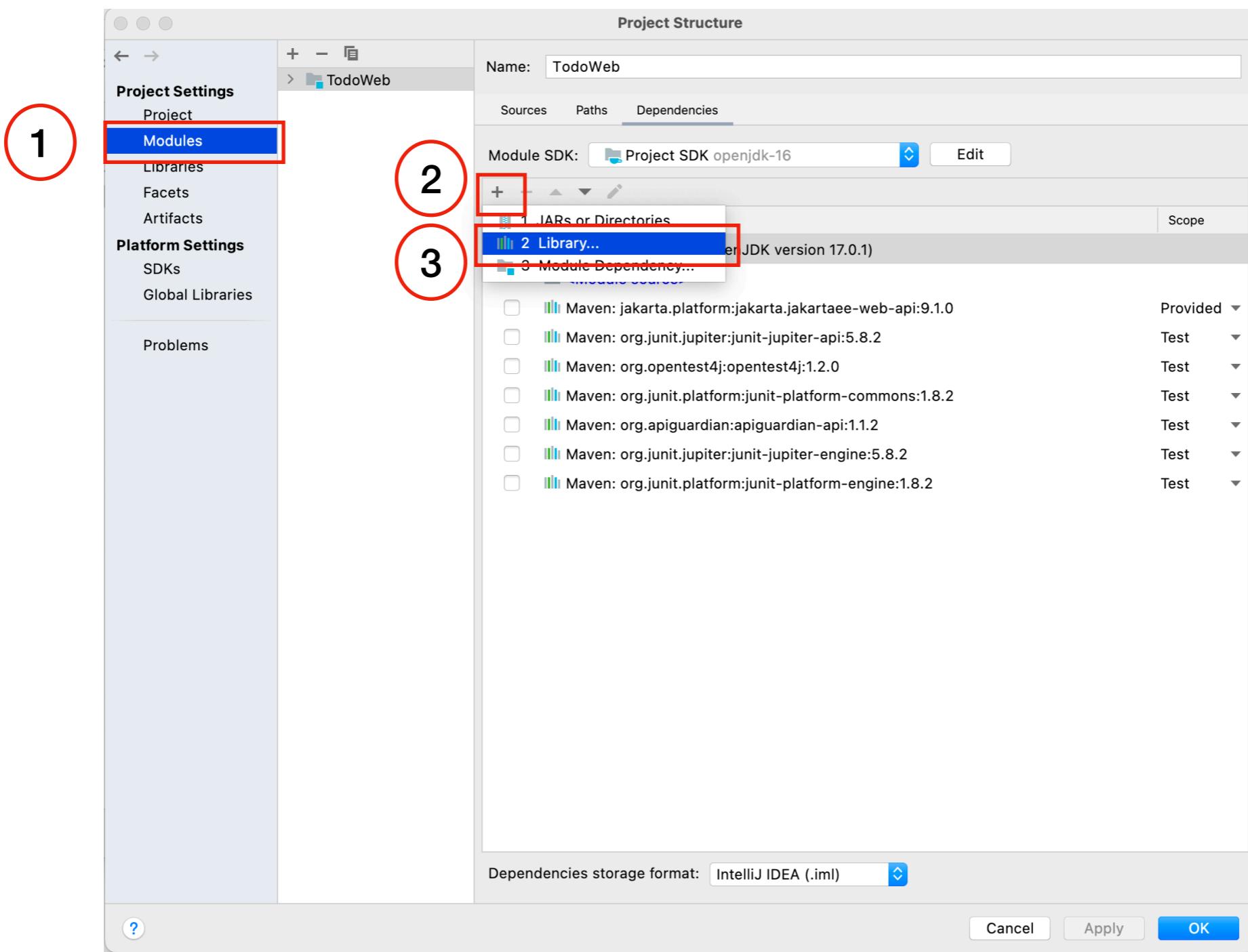
Project Settings > Modules



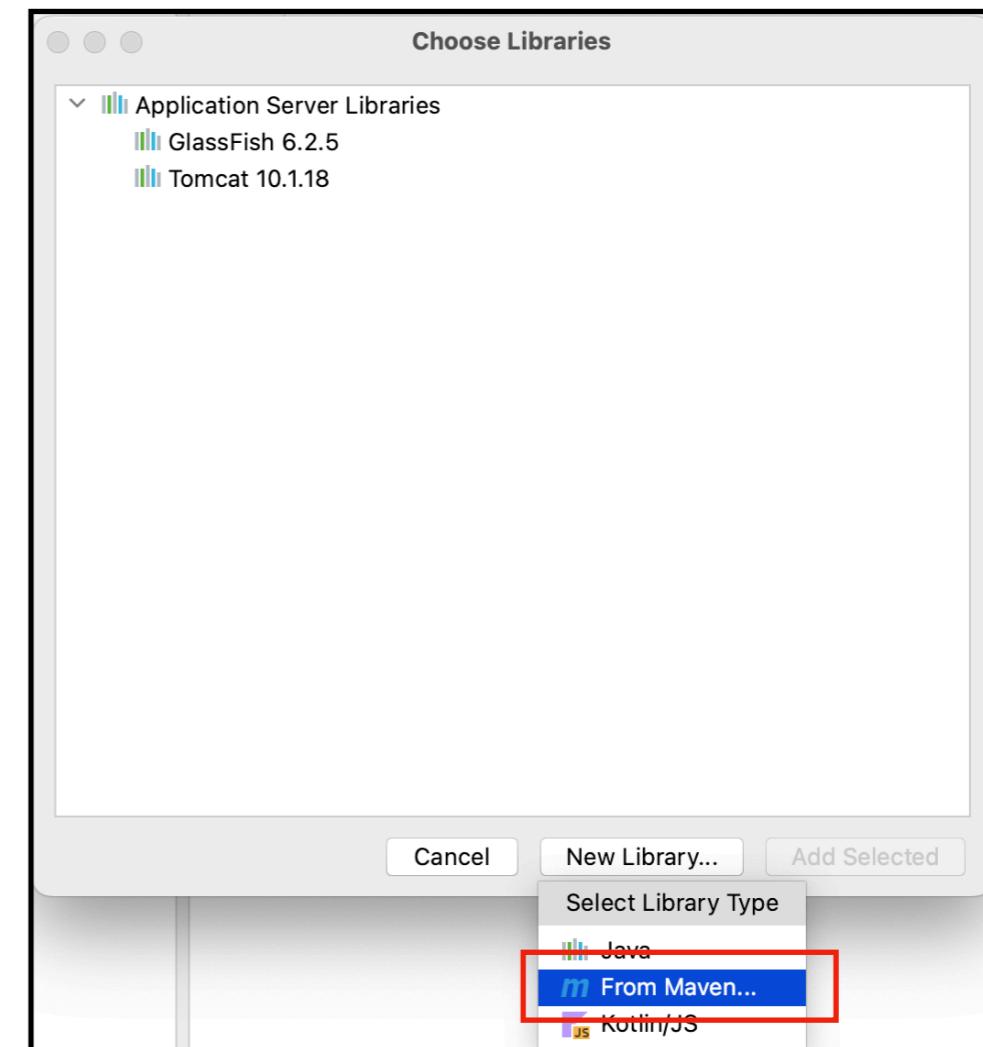
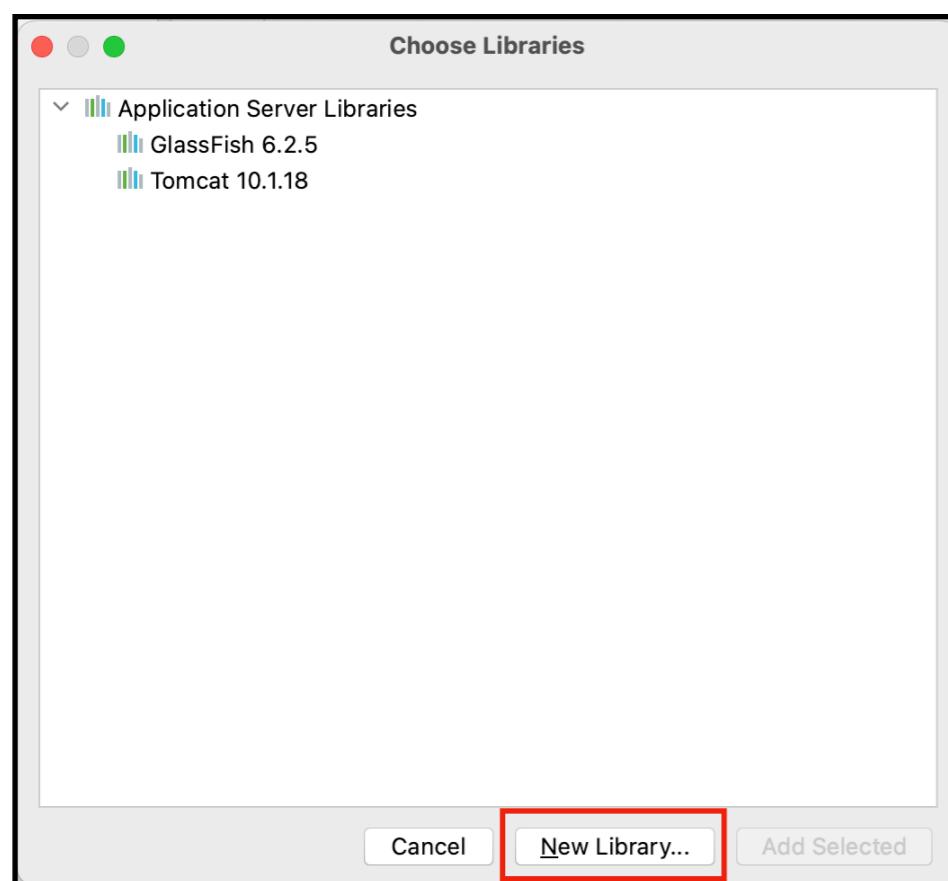
Modules

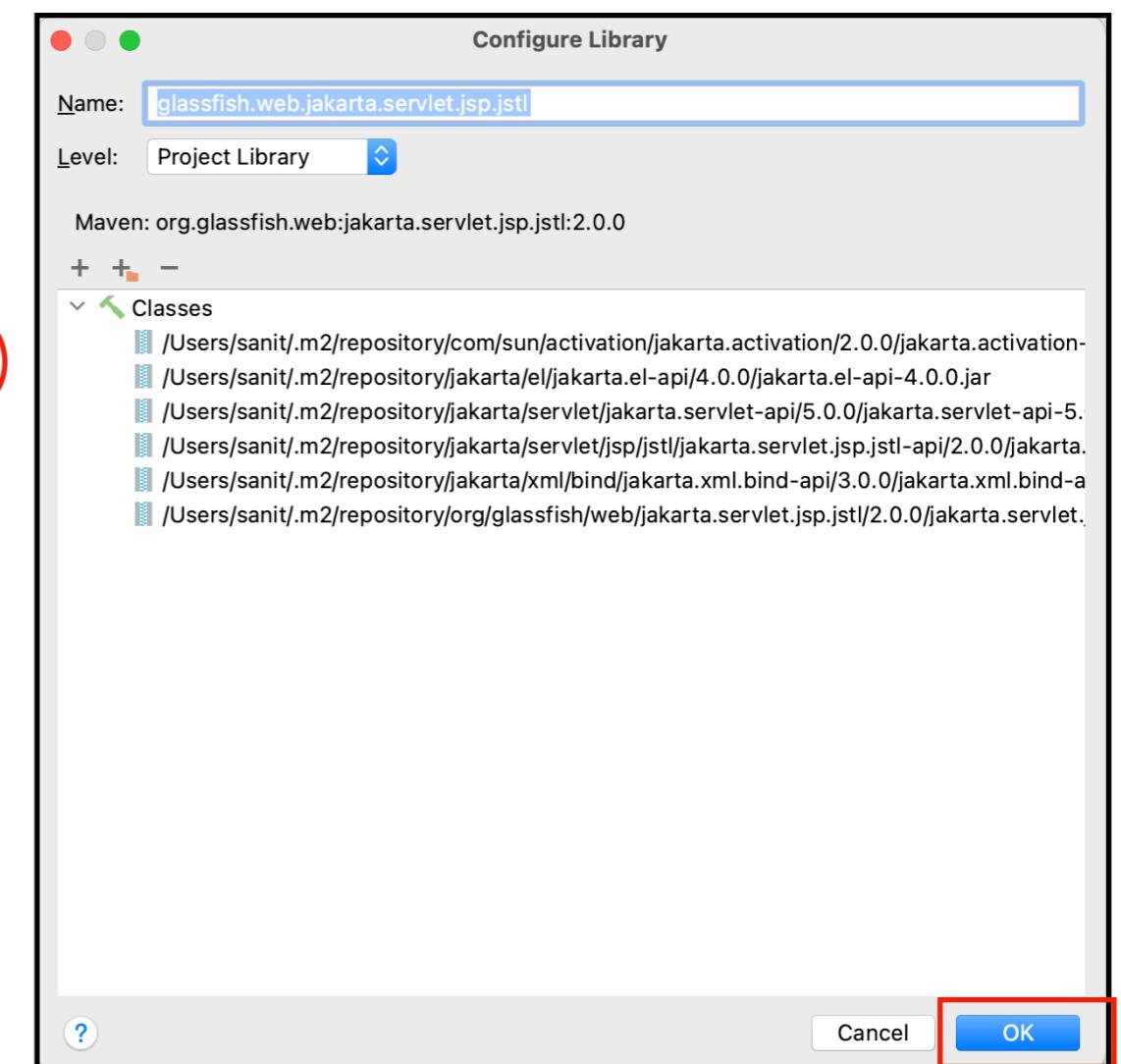
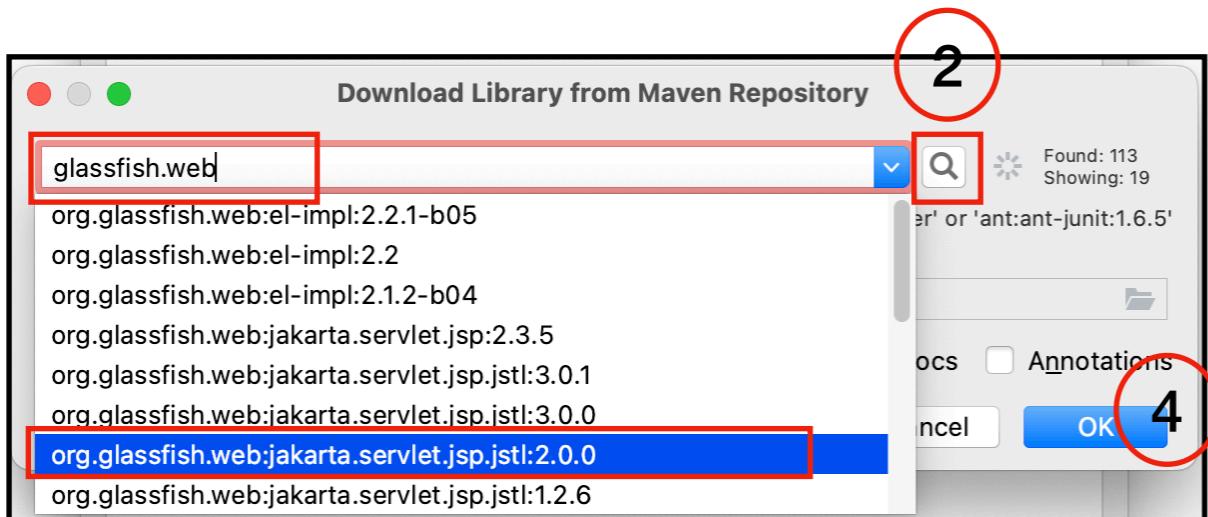


Modules > Add > Library



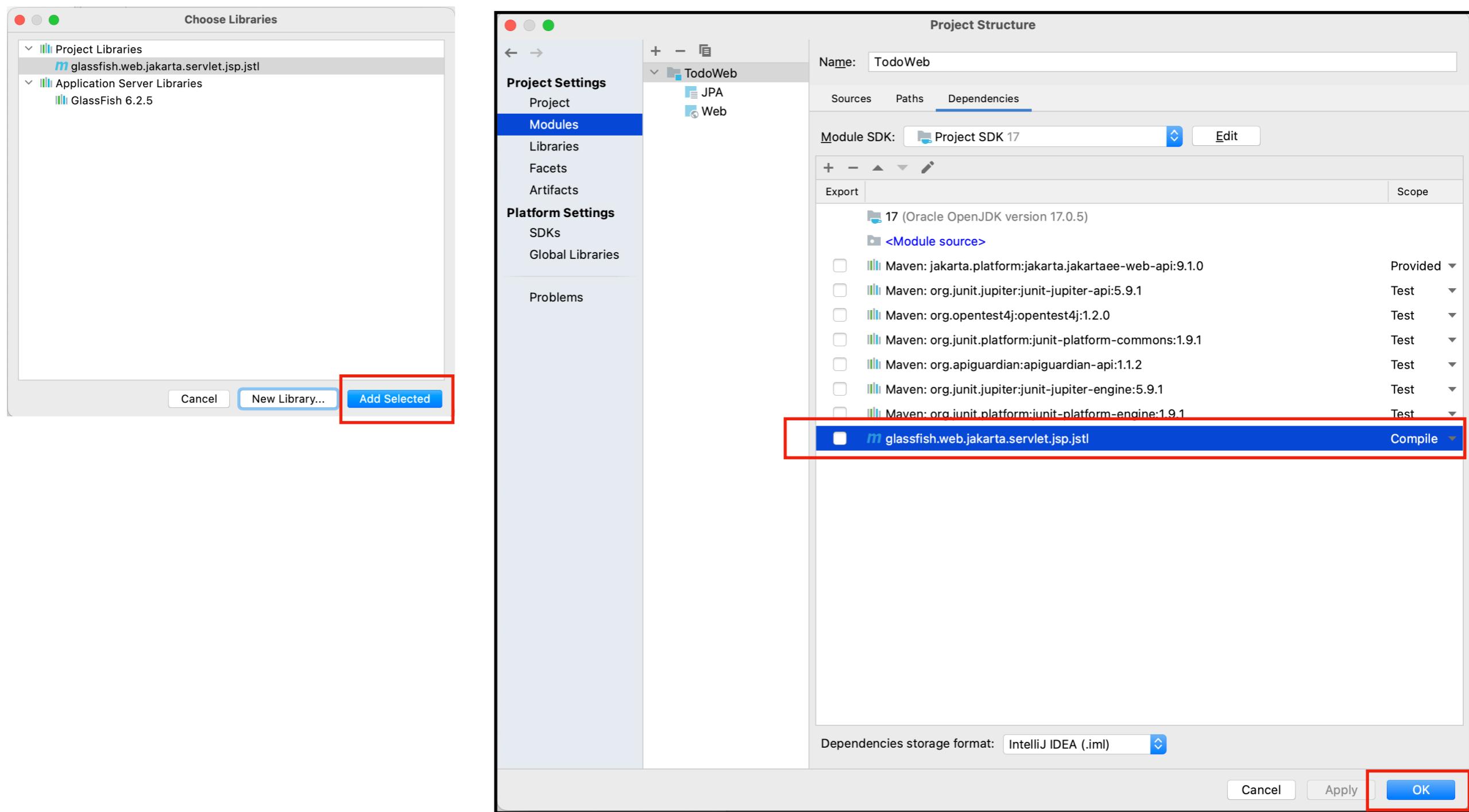
New Library > From Maven...





1. Type "glassfish.web"
2. Click "Search" button
3. Select "org.glassfish.web.jakarta.servlet.jsp.jstl:2.0.0"
4. Click "OK" button
5. Click "Ok" button

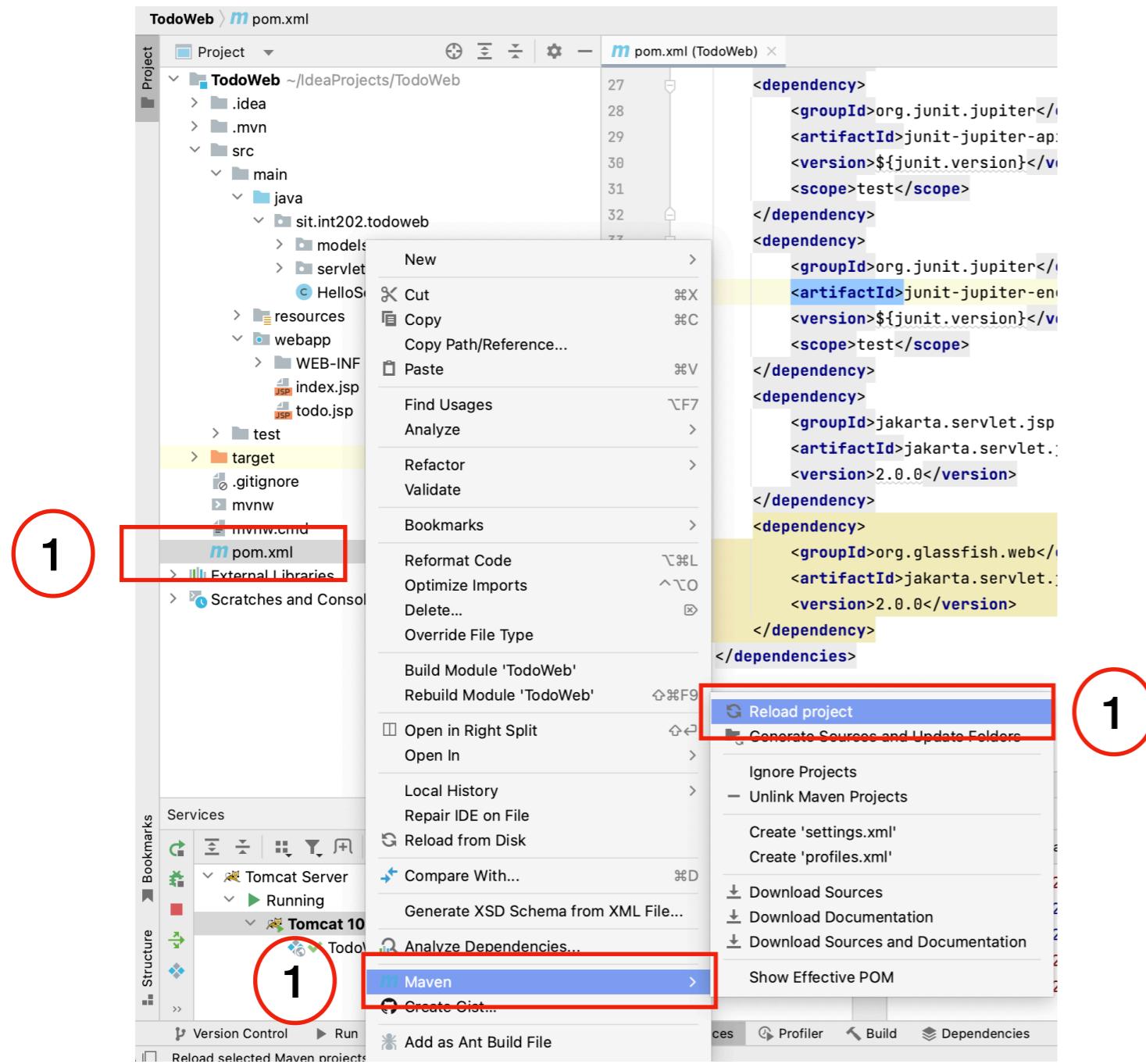
Add Selected > Ok



pom.xml

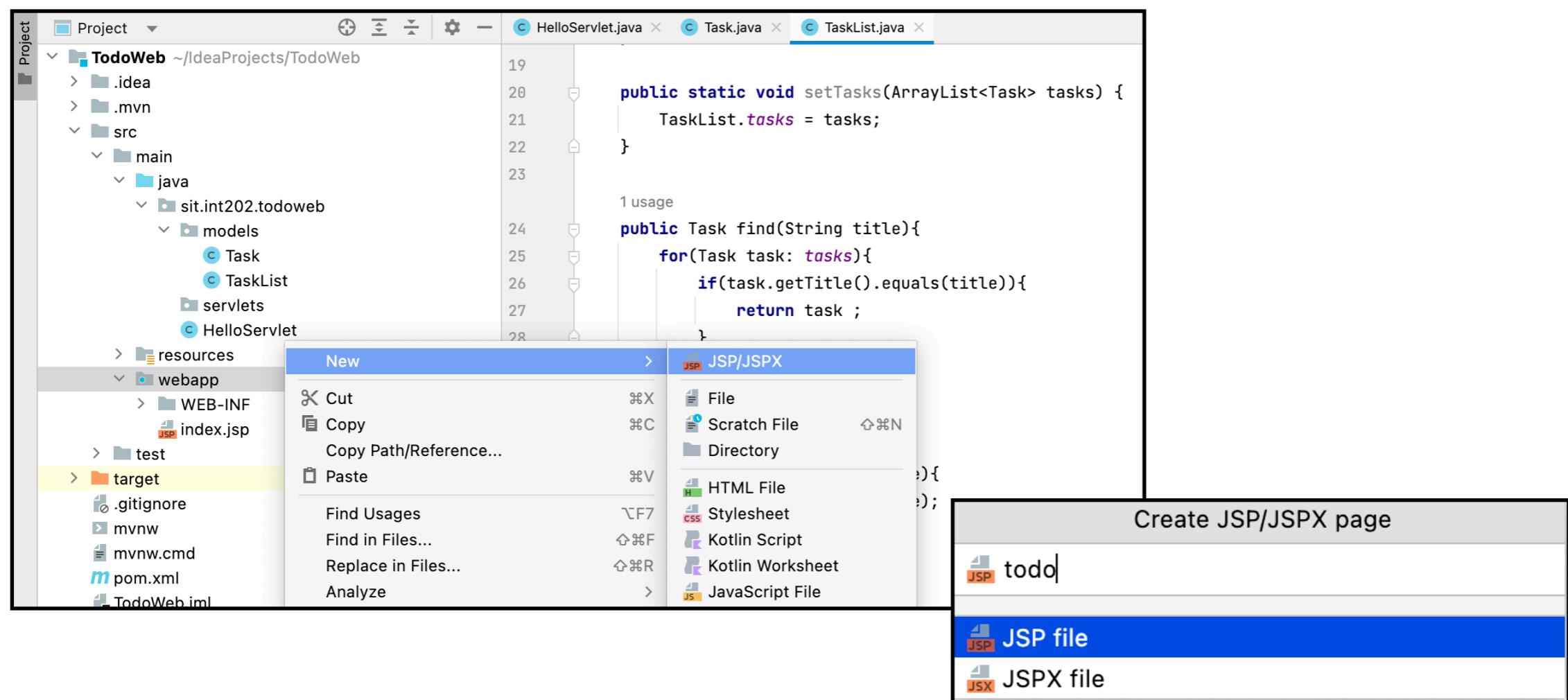
```
34         <groupId>org.junit.jupiter</groupId>
35         <artifactId>junit-jupiter-engine</artifactId>
36         <version>${junit.version}</version>
37         <scope>test</scope>
38     </dependency>
39     <dependency>
40         <groupId>jakarta.servlet.jsp.jstl</groupId>
41         <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
42         <version>2.0.0</version>
43     </dependency>
44     <dependency>
45         <groupId>org.glassfish.web</groupId>
46         <artifactId>jakarta.servlet.jsp.jstl</artifactId>
47         <version>2.0.0</version>
48     </dependency>
49   </dependencies>
```

Reload project



1. pom.xml
2. Click right mouse, Choose 'Maven'
3. Select 'Reload project'

New > JSP/JSPX > todo.jsp



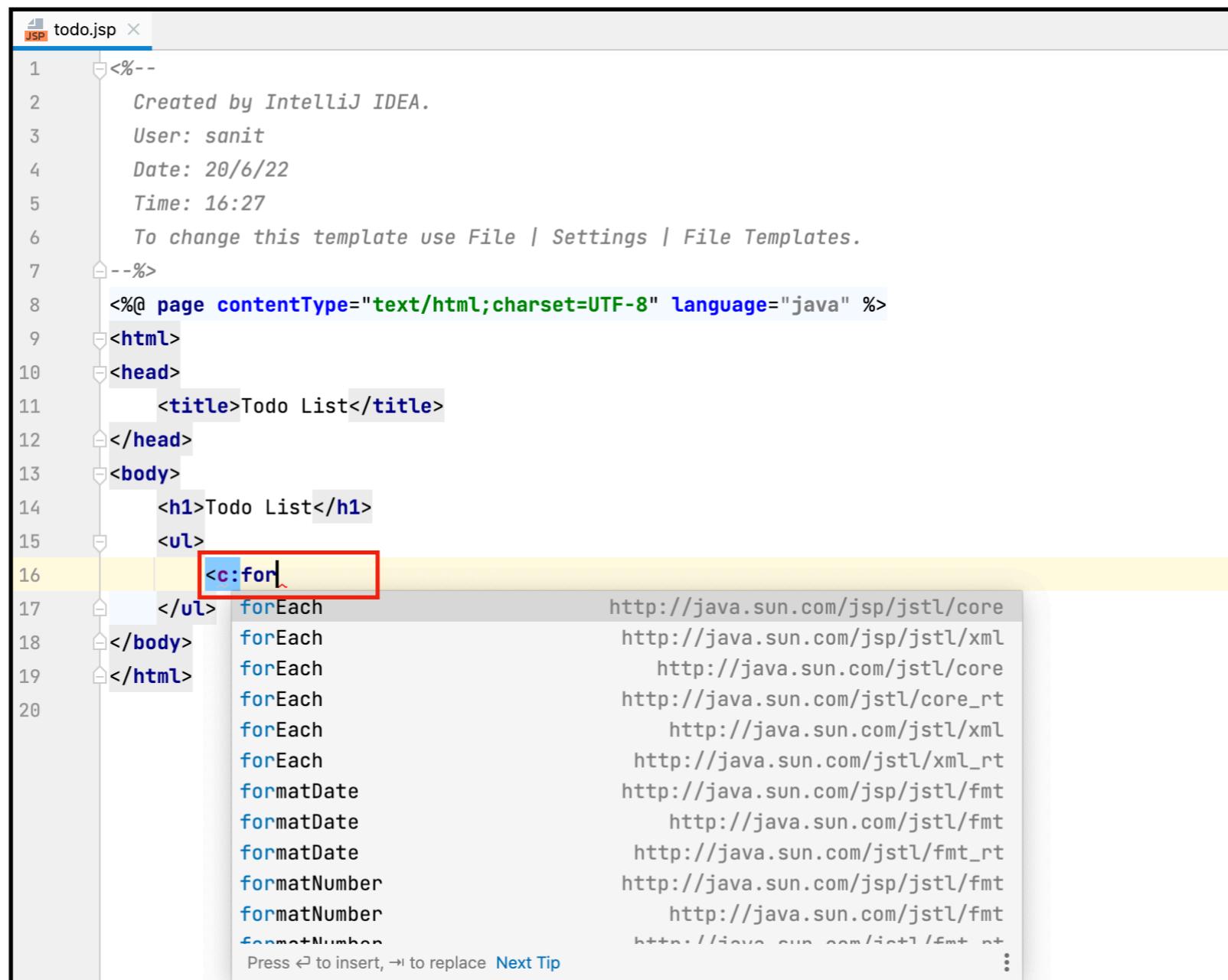
todo.jsp



The screenshot shows the code editor of IntelliJ IDEA displaying a JSP file named 'todo.jsp'. The code includes a header with creation details, a page directive, and a simple HTML structure with a title placeholder.

```
JSP todo.jsp ×  
1 <%--  
2     Created by IntelliJ IDEA.  
3     User: sanit  
4     Date: 20/6/22  
5     Time: 16:27  
6     To change this template use File | Settings | File Templates.  
7 --%>  
8 <%@ page contentType="text/html;charset=UTF-8" language="java" %>  
9 <html>  
10 <head>  
11     <title>Title</title>  
12 </head>  
13 <body>  
14 </body>  
15 </html>  
16  
17
```

todo.jsp



The screenshot shows the IntelliJ IDEA interface with a file named "todo.jsp" open. The code editor displays the following JSP code:

```
1 <%--  
2     Created by IntelliJ IDEA.  
3     User: sanit  
4     Date: 20/6/22  
5     Time: 16:27  
6     To change this template use File | Settings | File Templates.  
7 --%>  
8 <%@ page contentType="text/html;charset=UTF-8" language="java" %>  
9 <html>  
10 <head>  
11     <title>Todo List</title>  
12 </head>  
13 <body>  
14     <h1>Todo List</h1>  
15     <ul>  
16         <c:for|  
17             </ul> forEach http://java.sun.com/jsp/jstl/core  
18         </body> forEach http://java.sun.com/jsp/jstl/xml  
19     </html> forEach http://java.sun.com/jstl/core  
20
```

A code completion dropdown is open at line 16, showing suggestions for the JSTL `c:for` tag. The suggestion `forEach` is highlighted. The dropdown also lists other JSTL tags and their URLs:

- forEach http://java.sun.com/jsp/jstl/core
- for http://java.sun.com/jsp/jstl/xml
- foreach http://java.sun.com/jstl/core
- foreach http://java.sun.com/jstl/core_rt
- for http://java.sun.com/jstl/xml
- foreach http://java.sun.com/jstl/xml_rt
- formatDate http://java.sun.com/jsp/jstl/fmt
- formatDate http://java.sun.com/jstl/fmt
- formatDate http://java.sun.com/jstl/fmt_rt
- formatNumber http://java.sun.com/jsp/jstl/fmt
- formatNumber http://java.sun.com/jstl/fmt
- formatNumber http://java.sun.com/jstl/fmt_rt

At the bottom of the dropdown, it says "Press ⌘ to insert, ⌘ to replace Next Tip".

todo.jsp

- Use `<c:forEach>` tag for iterating over Java Collections.

The screenshot shows a Java Server Page (JSP) file named "todo.jsp" in an IDE. The code is as follows:

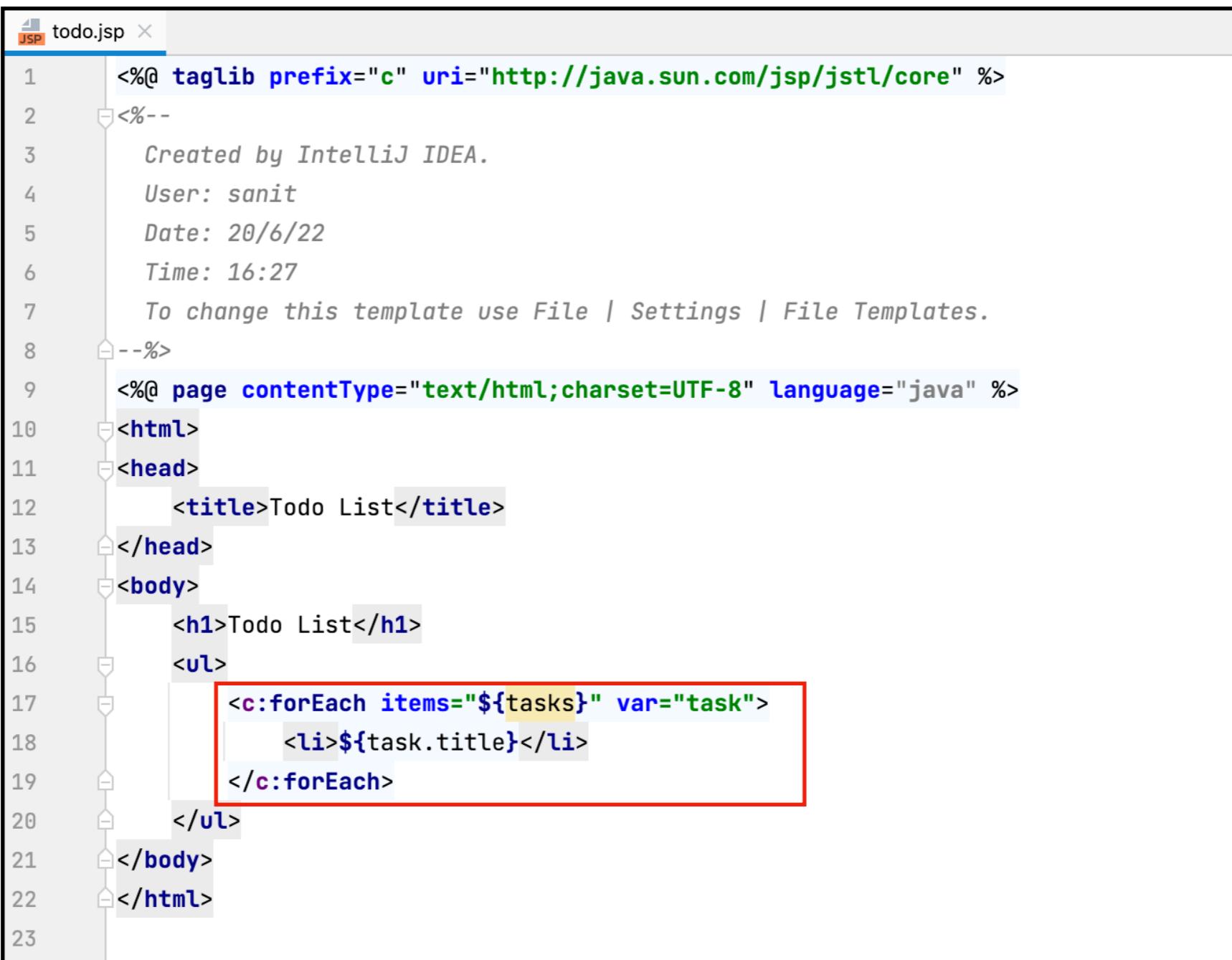
```
1  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2  <%-- 
3      Created by IntelliJ IDEA.
4      User: sanit
5      Date: 20/6/22
6      Time: 16:27
7      To change this template use File | Settings | File Templates.
8  --%>
9  <%@ page contentType="text/html;charset=UTF-8" language="java" %>
10 <html>
11   <head>
12     <title>Todo List</title>
13   </head>
14   <body>
15     <h1>Todo List</h1>
16     <ul>
17       <c:forEach
18         </ul>
19       </body>
20     </html>
21
```

The cursor is positioned at the start of the `<c:forEach` tag on line 17. A code completion dropdown menu is open, listing the following attributes:

- var
- items
- begin
- end
- step
- varStatus

A red box highlights the first two items in the list: "var" and "items".

todo.jsp

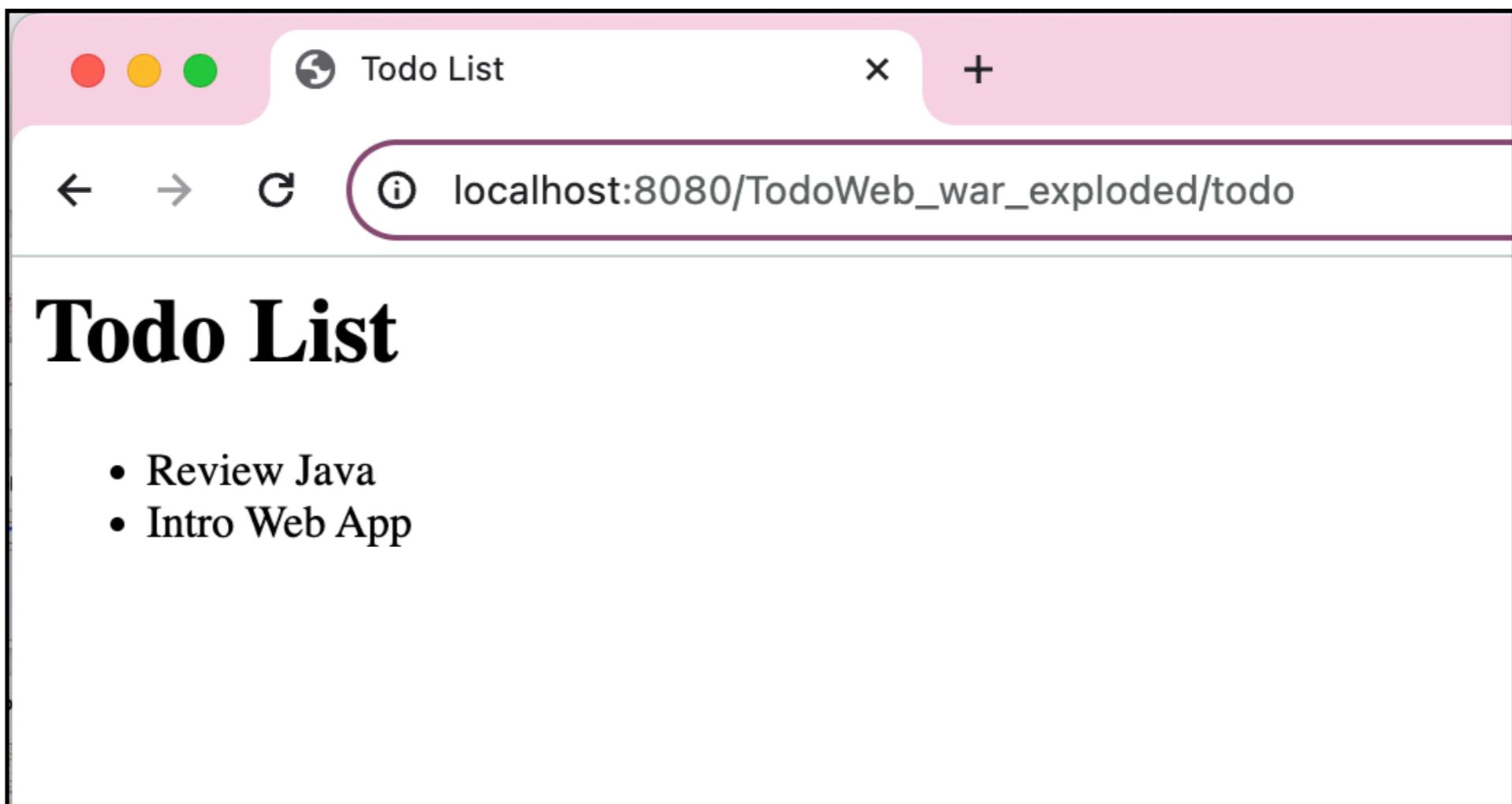


```
todo.jsp x
1  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2  <%-- 
3      Created by IntelliJ IDEA.
4      User: sanit
5      Date: 20/6/22
6      Time: 16:27
7      To change this template use File | Settings | File Templates.
8  --%>
9  <%@ page contentType="text/html;charset=UTF-8" language="java" %>
10 <html>
11 <head>
12     <title>Todo List</title>
13 </head>
14 <body>
15     <h1>Todo List</h1>
16     <ul>
17         <c:forEach items="${tasks}" var="task">
18             <li>${task.title}</li>
19         </c:forEach>
20     </ul>
21 </body>
22 </html>
23
```

index.jsp



```
index.jsp
1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2 <!DOCTYPE html>
3 <html>
4 <head>
5     <title>JSP - Hello World</title>
6 </head>
7 <body>
8     <h1><%= "Hello World!" %>
9     </h1>
10    <br/>
11    <a href="hello-servlet">Hello Servlet</a><br>
12    <a href="todo">Todo List</a>
13 </body>
14 </html>
```



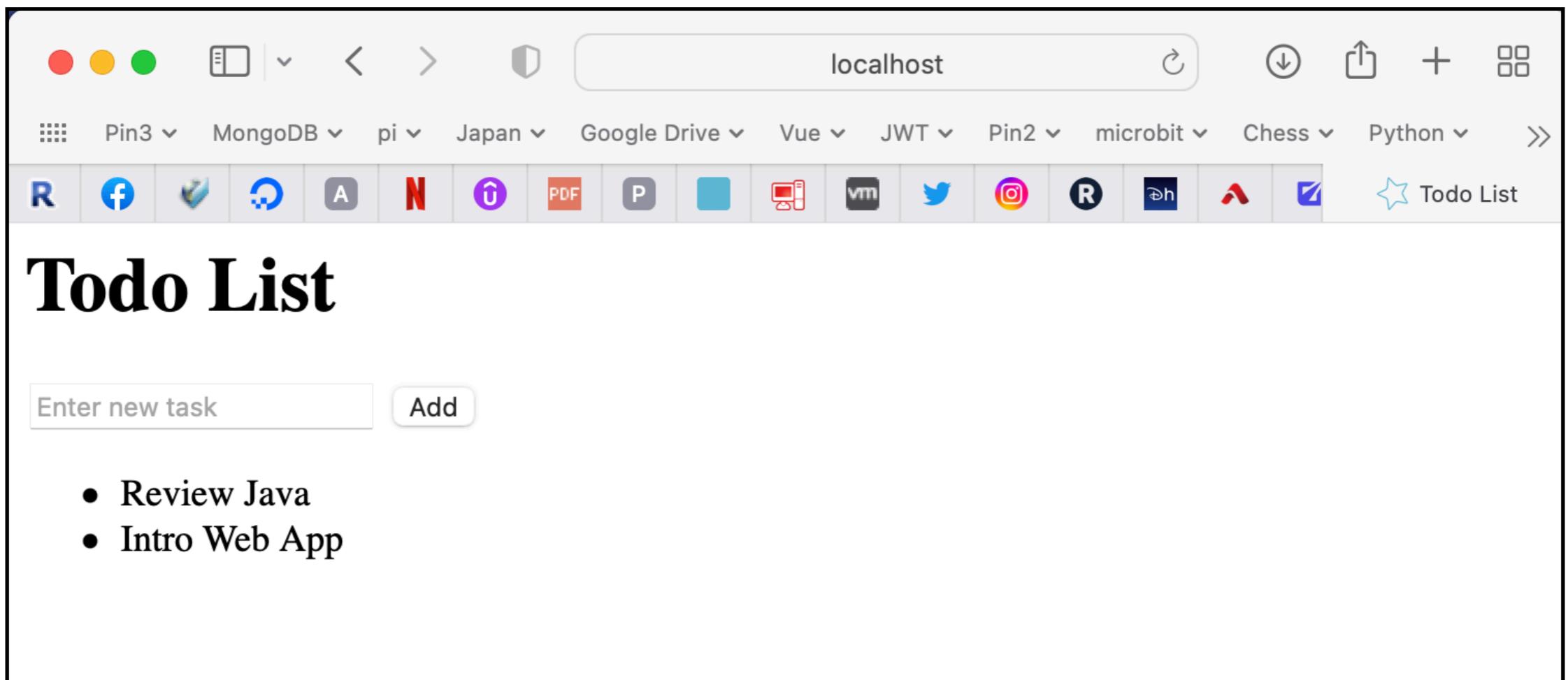
todo.jsp

- Add HTML Form to get data from users.



```
JSP todo.jsp ×
1  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2  <%--%
3      Created by IntelliJ IDEA.
4      User: sanit
5      Date: 20/6/22
6      Time: 16:27
7      To change this template use File | Settings | File Templates.
8  --%>
9  <%@ page contentType="text/html;charset=UTF-8" language="java" %>
10 <html>
11 <head>
12     <title>Todo List</title>
13 </head>
14 <body>
15     <h1>Todo List</h1>
16     <form action="add-todo" method="post">
17         <input type="text" name="title" placeholder="Enter new task">
18         <input type="submit" value="Add">
19     </form>
20
21     <ul>
22         <c:forEach items="${tasks}" var="task">
23             <li>${task.title}</li>
24         </c:forEach>
25     </ul>
26
27 </body>
28 </html>
```

todo.jsp



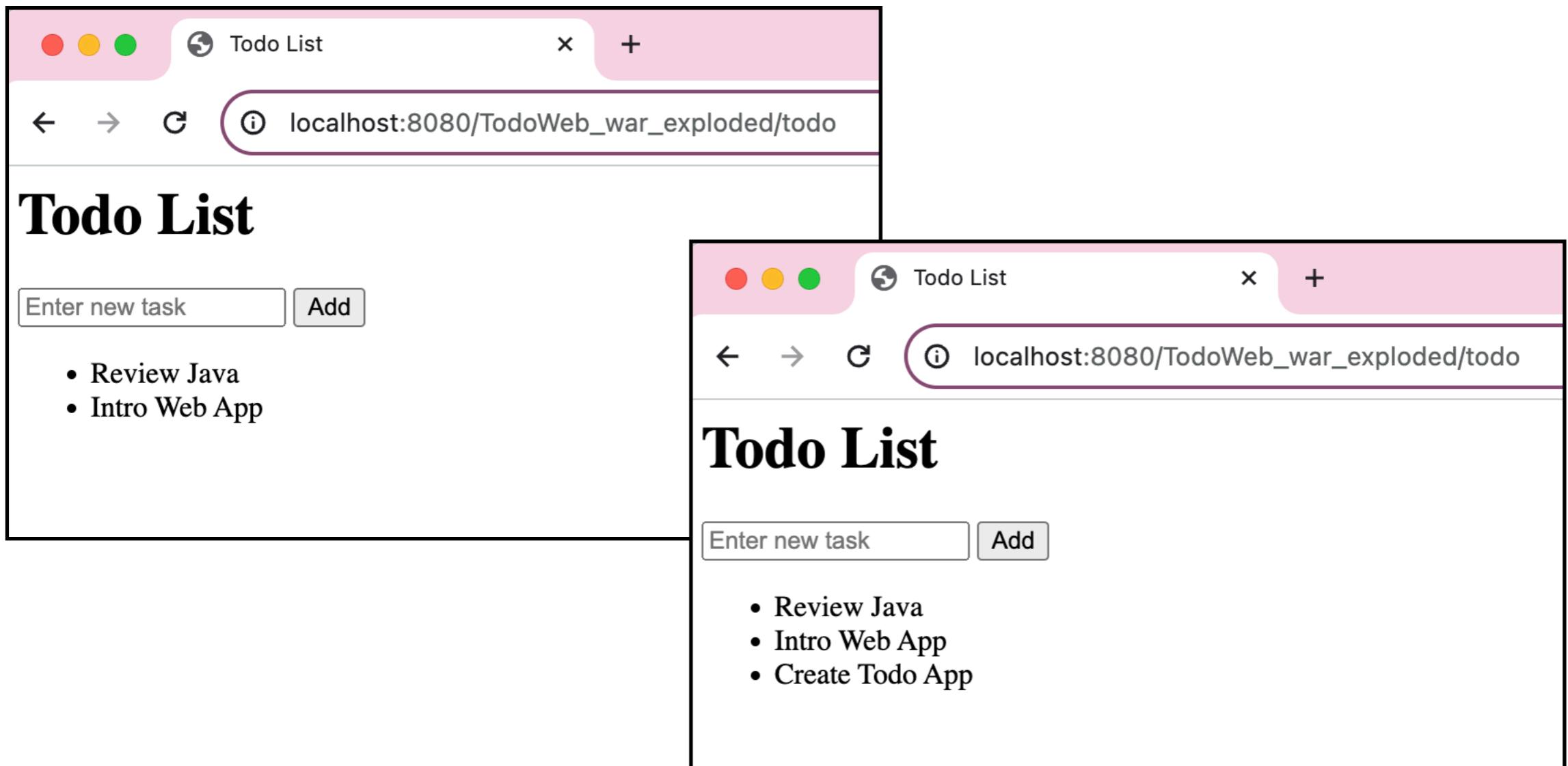
AddTodoServlet

```
1 package sit.int202.todoweb.servlets;
2
3 import ...
4
5
6 @WebServlet(name = "AddTodoServlet", value = "/add-todo")
7 public class AddTodoServlet extends HttpServlet {
8     @Override
9     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10
11     }
12
13     @Override
14     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15         String title = request.getParameter("title");
16
17         TaskList taskList = new TaskList();
18         taskList.addTask(title);
19
20         request.setAttribute("tasks", taskList.getTasks());
21         getServletContext().getRequestDispatcher("/todo.jsp").forward(request, response);
22     }
23
24
25
26
27
28 }
```

response.sendRedirect

```
c AddTodoServlet.java ×
1 package sit.int202.todoweb.servlets;
2
3 import ...
4
5 no usages
6
7 @WebServlet(name = "AddTodoServlet", value = "/add-todo")
8 public class AddTodoServlet extends HttpServlet {
9     no usages
10    @Override
11    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12        }
13    }
14
15    no usages
16    @Override
17    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
18        String title = request.getParameter("title");
19
20        TaskList taskList = new TaskList();
21        taskList.addTask(title);
22
23        // request.setAttribute("tasks", taskList.getTasks());
24        // getServletContext().getRequestDispatcher("/todo.jsp").forward(request, response);
25        response.sendRedirect("todo");
26    }
27}
28}
```

Add a task

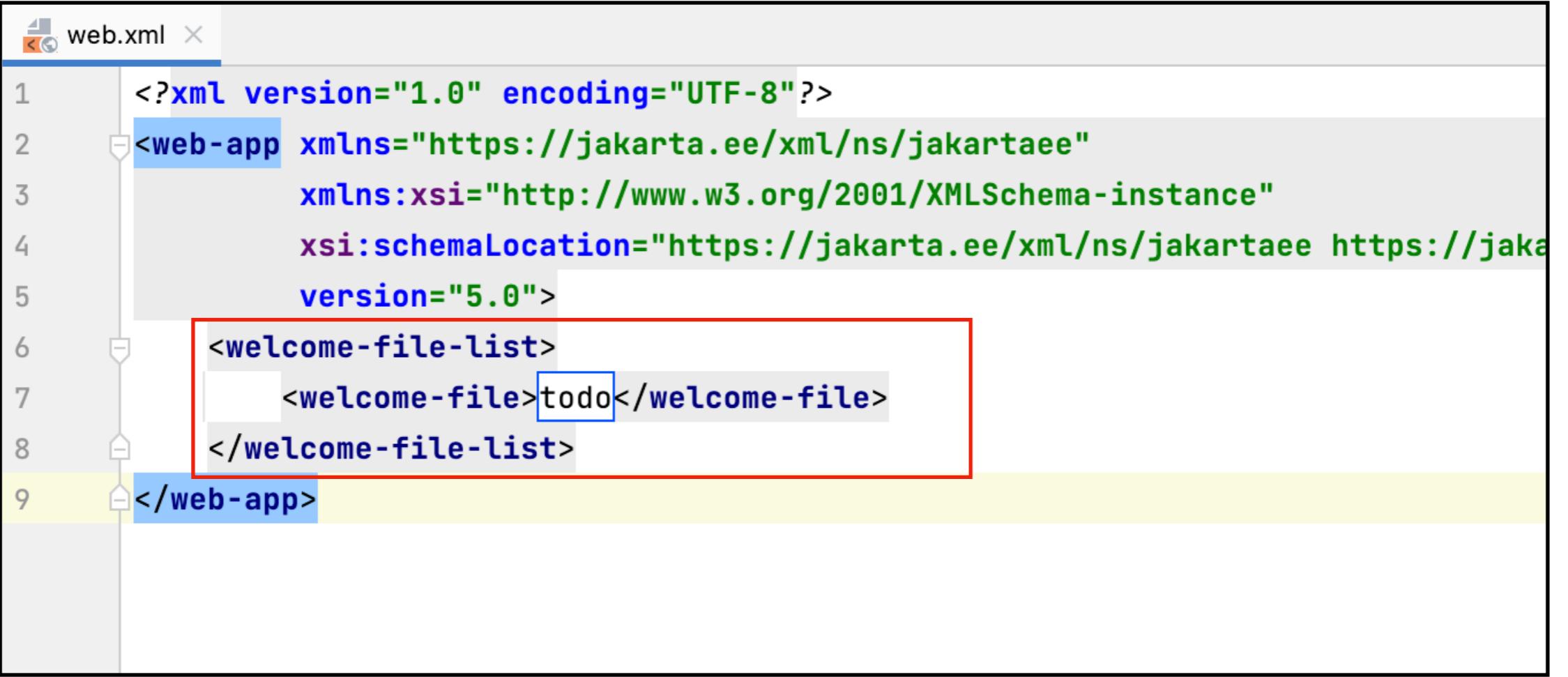


Handling Null Parameter and Empty

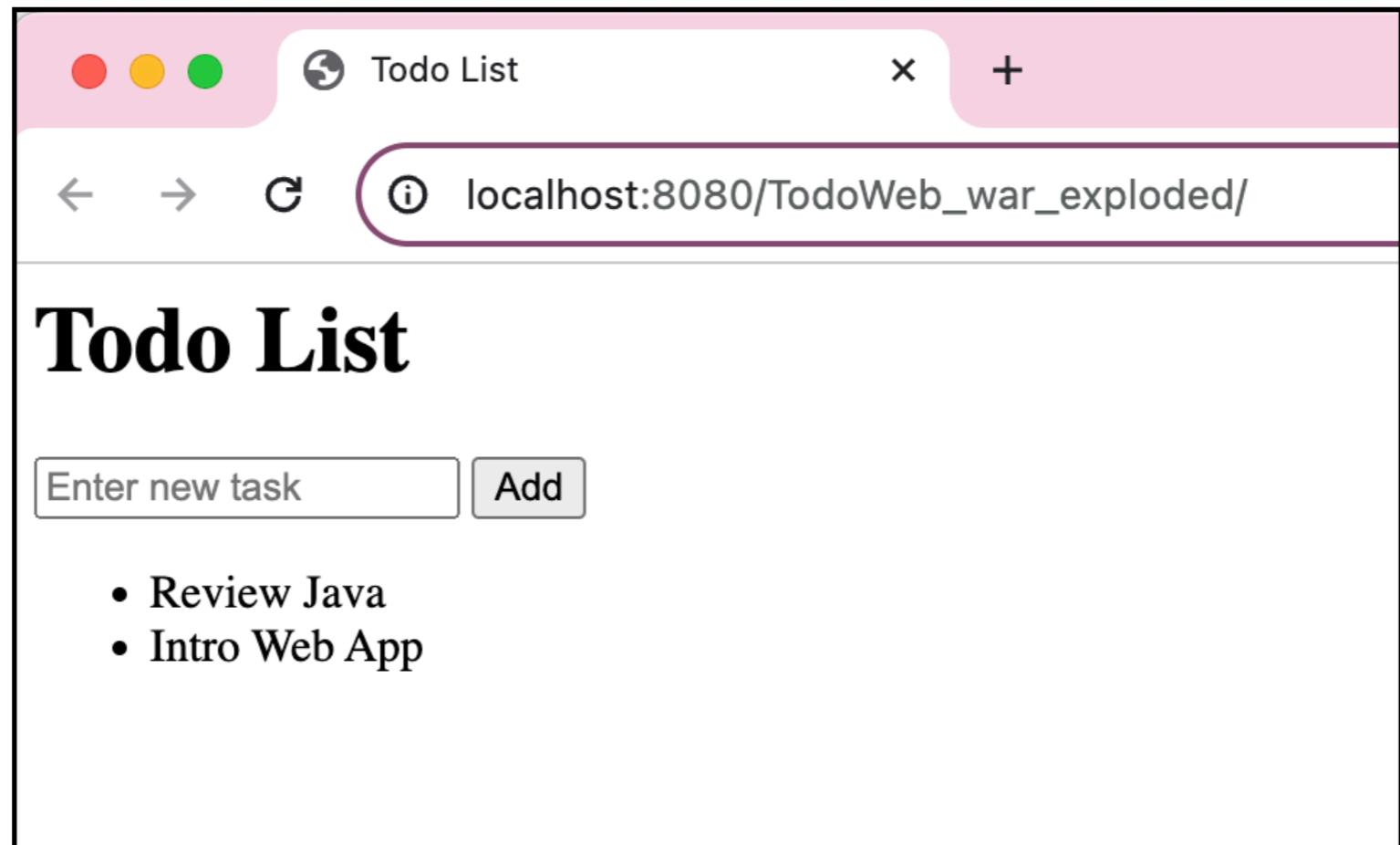


```
1 package sit.int202.todoweb.servlets;
2
3 import ...
4
5 no usages
6
7 @WebServlet(name = "AddTodoServlet", value = "/add-todo")
8 public class AddTodoServlet extends HttpServlet {
9     no usages
10    @Override
11    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
12        }
13    }
14
15    no usages
16
17    @Override
18    @Override
19    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
20        String title = request.getParameter( s: "title" );
21
22        if(title != null && title.length()>0){
23            TaskList taskList = new TaskList();
24            taskList.addTask(title);
25        }
26
27        //request.setAttribute("tasks", taskList.getTasks());
28        //getServletContext().getRequestDispatcher("/todo.jsp").forward(request, response);
29        response.sendRedirect( s: "todo" );
30    }
31}
```

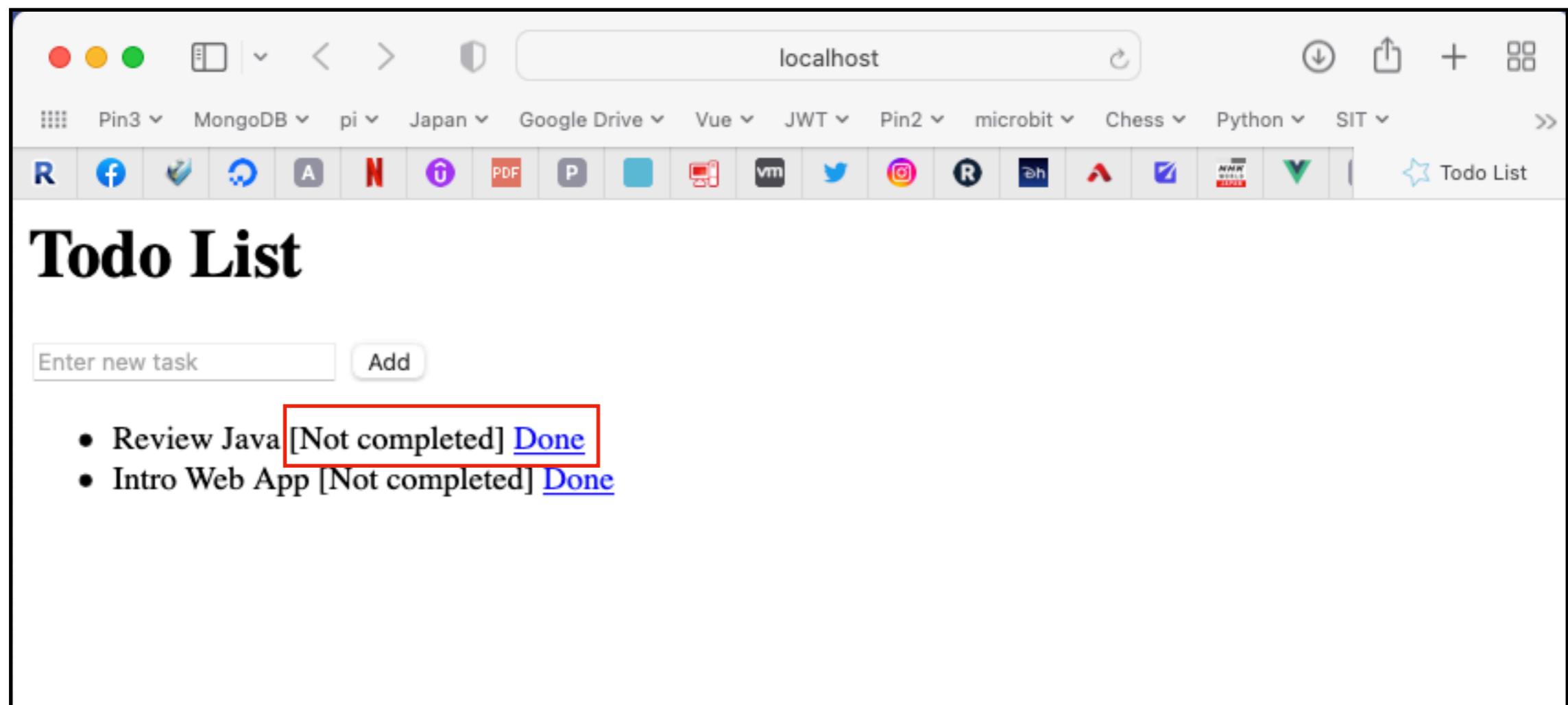
Set TodoServlet as Welcome file



```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jaka
5          version="5.0">
6      <welcome-file-list>
7          <welcome-file>todo</welcome-file>
8      </welcome-file-list>
9  </web-app>
```



Add Task Status and HTML Link



todo.jsp

```
todo.jsp x
1 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2 <%-- 
3     Created by IntelliJ IDEA.
4     User: sanit
5     Date: 20/6/22
6     Time: 16:27
7     To change this template use File | Settings | File Templates.
8 --%>
9 <%@ page contentType="text/html;charset=UTF-8" language="java" %>
10 <html>
11 <head>
12     <title>Todo List</title>
13 </head>
14 <body>
15     <h1>Todo List</h1>
16     <form action="add-todo" method="post">
17         <input type="text" name="title" placeholder="Enter new task">
18         <input type="submit" value="Add">
19     </form>
20
21     <ul>
22         <c:forEach items="${tasks}" var="task">
23             <li>
24                 ${task.title} ${ task.status=="NOT_COMPLETED"? "[Not completed]": "[Completed]"}  

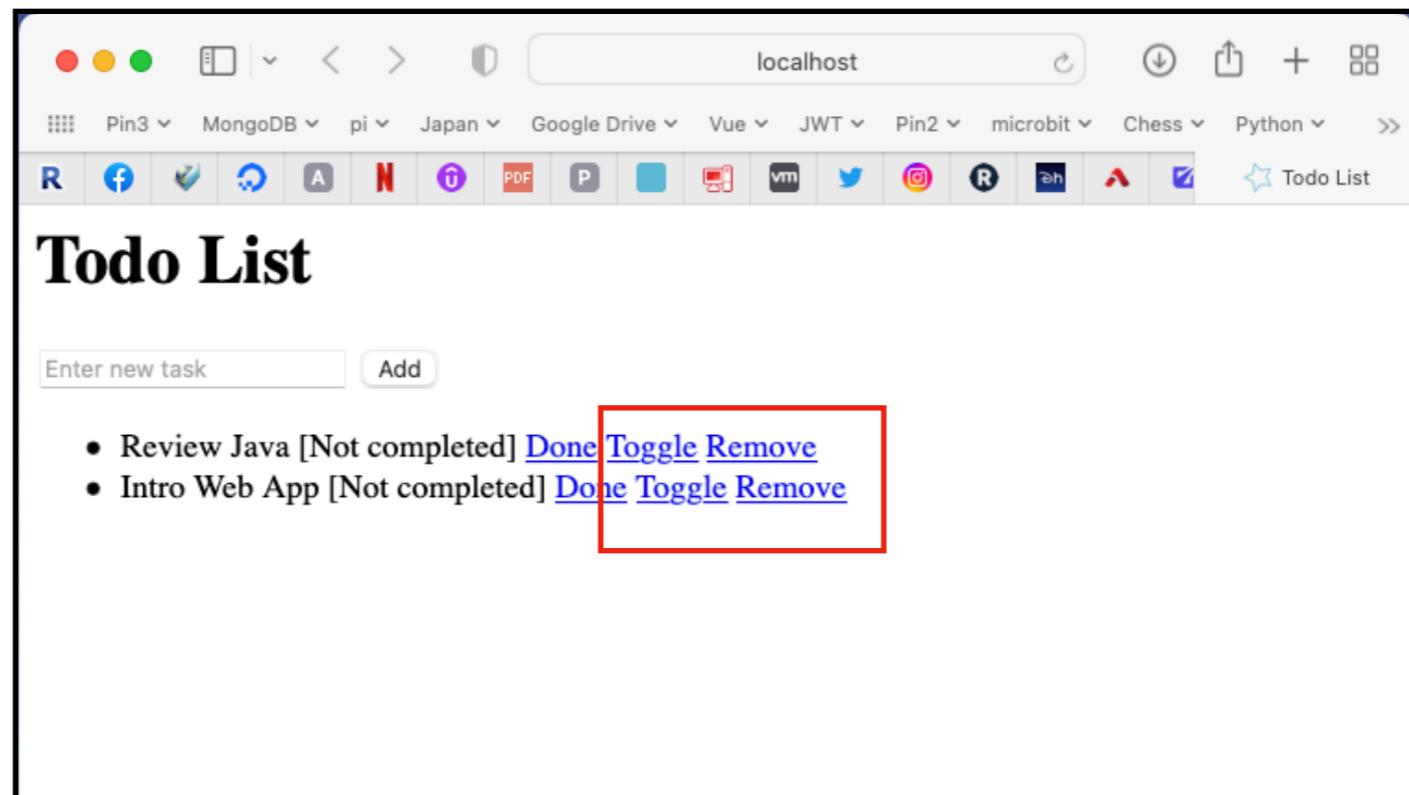
25                 <a href="completed-todo?title=${task.title}">Done</a>
26             </li>
27         </c:forEach>
28     </ul>
29 </body>
30 </html>
```

CompletedTodoServlet

- Add a servlet named `CompletedTodoServlet`

```
c CompletedTodoServlet.java x
1 package sit.int202.todoweb.servlets;
2
3 import ...
10
11 @WebServlet(name = "CompleteTodoServlet", value = "/completed-todo")
12 public class CompletedTodoServlet extends HttpServlet {
13     @Override
14     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15
16         String title = request.getParameter("title");
17
18         if(title != null && title.length()>0){
19             TaskList taskList = new TaskList();
20             Task task = taskList.find(title);
21             if(task != null){
22                 task.completed();
23             }
24         }
25
26         response.sendRedirect("todo");
27     }
28
29     @Override
30     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
31
32     }
33
34 }
```

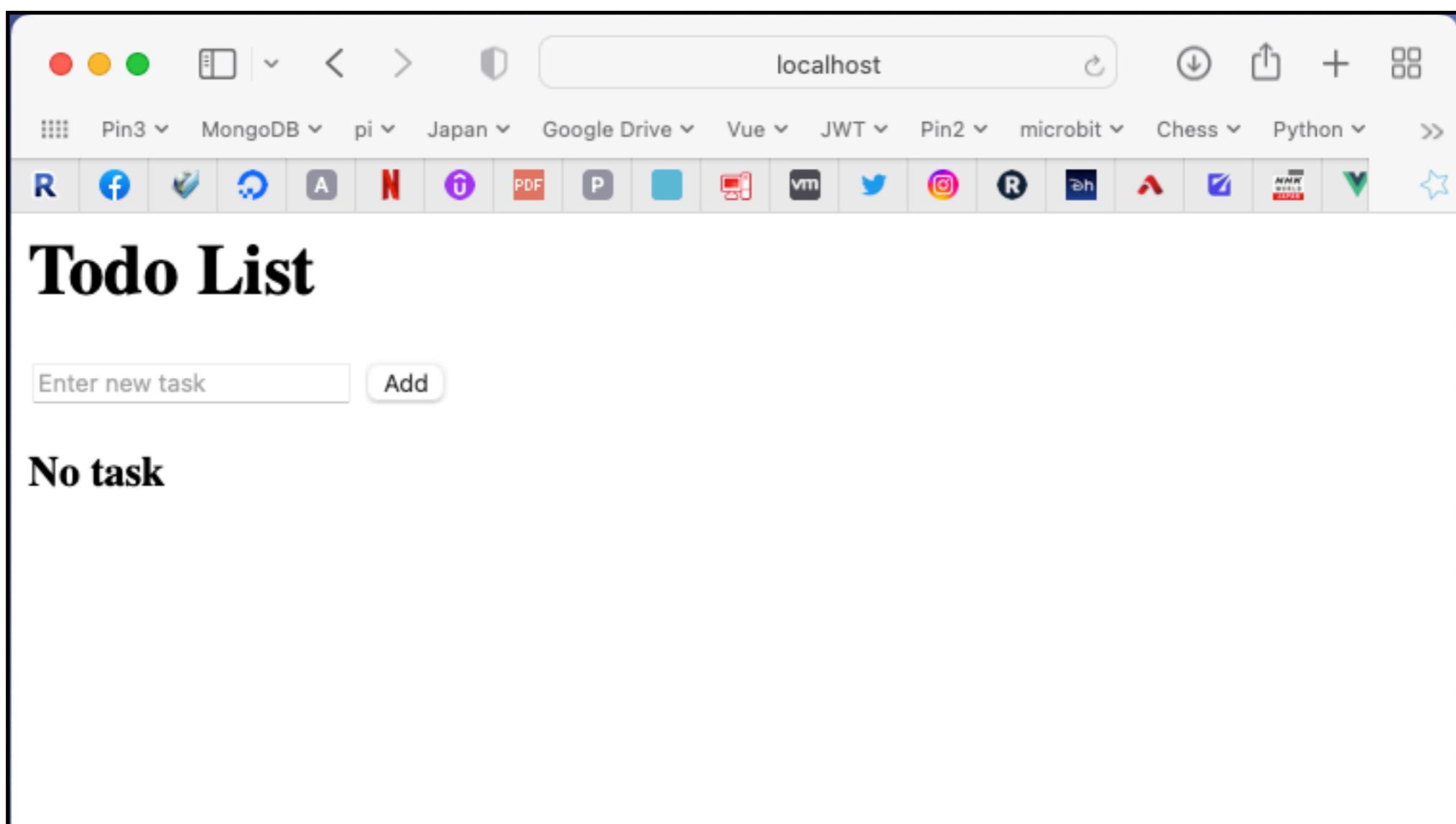
Lab 4: Task 1



- Add two HTML Links : [Toggle](#) , [Remove](#)
- Add two Servlets ([ToggleTodoServlet](#) and [RemoveTodoServlet](#)) with following URL patterns:
 - [/toggle-todo](#) will call `task.toggleStatus()`
 - [/remove-todo](#) will call `taskList.remove(title)`

No task

- When the task list is empty, show "No task".



todo.jsp

```

1  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2  <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
3  <%-- ... --%>
10 <%@ page contentType="text/html; charset=UTF-8" language="java" %>
11 <html>
12 <head>
13   <title>Todo List</title>
14 </head>
15 <body>
16   <h1>Todo List</h1>
17   <form action="add-todo" method="post">
18     <input type="text" name="title" placeholder="Enter new task">
19     <input type="submit" value="Add">
20   </form>
21
22   <c:choose>
23     <c:when test="${fn:length(tasks)==0}">
24       <h3>No task</h3>
25     </c:when>
26     <c:otherwise>
27       <ul>
28         <c:forEach items="${tasks}" var="task">
29           <li>
30             ${task.title} ${task.status=="NOT_COMPLETED"? "[Not completed]": "[Completed] "}
31             <a href="completed-todo?title=${task.title}">Done</a>
32           </li>
33         </c:forEach>
34       </ul>
35     </c:otherwise>
36   </c:choose>
37 </body>
38 </html>

```

https://www.tutorialspoint.com/jsp/jstl_function_length.htm

Lab 4 : Task 2

- Create a new servlet named `ActionTodoServlet`
 - `/action-todo?title=Test&action=completed`
 - `/action-todo?title=Test&action=toggle`
 - `/action-todo?title=Test&action=remove`
- Modify "todo.jsp" and replace `/completed-todo`, `/toggle-todo` and `/remove-todo` with `/action-todo`