



```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
    version="3.0">
4    <display-name>CSC4710 Demo</display-name>
5    <servlet>
6      <description></description>
7      <display-name>initserver</display-name>
8      <servlet-name>initserver</servlet-name>
9      <servlet-class>user.web.servlet.UserServletRegister</servlet-class>
10    </servlet>
11    <servlet-mapping>
12      <servlet-name>initserver</servlet-name>
13      <url-pattern>/initserver</url-pattern>
14    </servlet-mapping>
15    <servlet>
16      <description></description>
17      <display-name>UserServletRegister</display-name>
18      <servlet-name>UserServletRegister</servlet-name>
19      <servlet-class>user.web.servlet.UserServletRegister</servlet-class>
20    </servlet>
21    <servlet-mapping>
22      <servlet-name>UserServletRegister</servlet-name>
23      <url-pattern>/UserServletRegister</url-pattern>
24    </servlet-mapping>
25    <servlet>
26      <description></description>
27      <display-name>UserServletLogin</display-name>
28      <servlet-name>UserServletLogin</servlet-name>
29      <servlet-class>user.web.servlet.UserServletLogin</servlet-class>
30    </servlet>
31    <servlet-mapping>
32      <servlet-name>UserServletLogin</servlet-name>
33      <url-pattern>/UserServletLogin</url-pattern>
34    </servlet-mapping>
35    <servlet>
36      <description></description>
37      <display-name>findAll</display-name>
38      <servlet-name>findAll</servlet-name>
39      <servlet-class>user.web.servlet.findAll</servlet-class>
40    </servlet>
```

```
1  <%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
2  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3
4
5  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
6  <html>
7      <head>
8
9          <title>initialize DB</title>
10
11          <meta http-equiv="pragma" content="no-cache">
12          <meta http-equiv="cache-control" content="no-cache">
13          <meta http-equiv="expires" content="0">
14          <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
15          <meta http-equiv="description" content="This is my page">
16          <meta http-equiv="content-type" content="text/html; charset=utf-8">
17          <!--
18          <link rel="stylesheet" type="text/css" href="styles.css">
19          -->
20
21      </head>
22
23      <body>
24          <h1>Inititalize database</h1>
25
26
27          <p style="color: red; font-weight: 900"> ${msg}</p>
28          <form action="<c:url value='/initserver'/"> method="post">
29
30              <input type="submit" value="initialize DB"/>
31          </form>
32      </body>
33  </html>
34
```


Users > ebugarin > Downloads > CS480Demo > WEB-INF > classes > myuser > dao > UserDao.java

```
69  * insert User
70  * @param user
71  * @throws ClassNotFoundException
72  * @throws IllegalAccessException
73  * @throws InstantiationException
74  */
75  public void add(User user) throws ClassNotFoundException, InstantiationException, IllegalAccessException {
76      try {
77          Class.forName("com.mysql.jdbc.Driver").newInstance();
78          Connection connect = DriverManager
79              .getConnection("jdbc:mysql://141.217.48.128:3306/song_chart?"
80                  + "user=root&password=3l0nMuk!");
81
82          String sql = "insert into tb_user values(?,?,?)";
83          PreparedStatement preparestatement = connect.prepareStatement(sql);
84          preparestatement.setString(1,user.getUsername());
85          preparestatement.setString(2,user.getPassword());
86          preparestatement.setString(3,user.getEmail());
87          preparestatement.executeUpdate();
88      } catch(SQLException e) {
89          throw new RuntimeException(e);
90      }
91  }
92  }
93
94
95  public List<Object> findall() throws InstantiationException, IllegalAccessException, ClassNo
96      List<Object> list = new ArrayList<>();
97      try {
98          Class.forName("com.mysql.jdbc.Driver").newInstance();
99          Connection connect = DriverManager
100              .getConnection("jdbc:mysql://141.217.48.128:3306/song_chart?"
101                  + "user=root&password=3l0nMuk!");
102
103          String sql = "select * from tb_user";
104          PreparedStatement preparestatement = connect.prepareStatement(sql);
105          ResultSet resultSet = preparestatement.executeQuery();
106
107          // while(resultSet.next()){
108          //     User user = new User();
109          //     user.setUsername(resultSet.getString("username"));
110          //     user.setPassword(resultSet.getString("password"));
111          //     user.setEmail(resultSet.getString("email"));
112          //     list.add(user);
113          // }
114
115      } catch(SQLException e) {
116          throw new RuntimeException(e);
117      }
118  }
```




Skip the meeting and record a quick video instead.

Sign in

Get Loom for Free

Libraries

 Personal

 Shared

 Team

☆ Starred

[Learn more](#)

💡 Ways to use Loom

\$ Pricing

 About us

👁 315 views

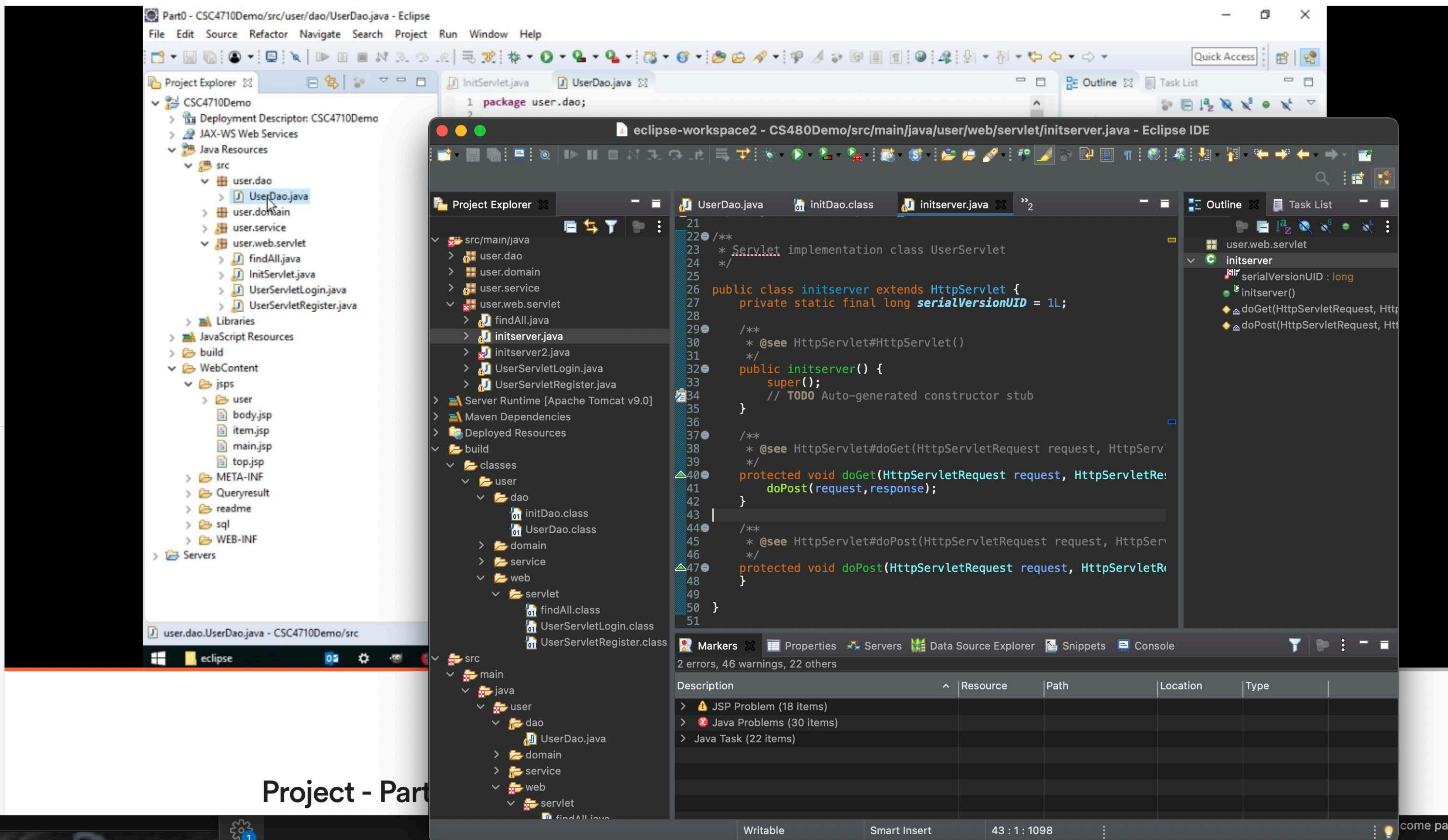


Share



Sign in

Get Loom for Free



Project Explorer

CS480Demo

Deployment Descriptor: CS480Demo

JAX-WS Web Services

JRE System Library [JavaSE-16]

src/main/java

user.dao

initdao.java

UserDao.java

user.domain

user.service

user.web.servlet

findAll.java

initserver.java

initserver2.java

UserServletLogin.java

UserServletRegister.java

Server Runtime [Apache Tomcat v9.0]

Maven Dependencies

Deployed Resources

build

classes

user

dao

initDao.class

UserDao.class

domain

service

web

servlet

findAll.class

UserServletLogin.class

UserServletRegister.class

src

main

java

user

dao

UserDao.java

domain

service

web

servlet

findAll.java

initserver.java

initserver2.java

UserServletLogin.java

UserServletRegister.java

webapp

jsp

META-INF

Queryresult

readme

sql

WEB-INF

target

pom.xml

Servers

CS480Demo/pom.xml

user.sql

UserDao.java

initDao.class

initserver.java

initdao.java

```
15 import user.domain.User;
16 import java.sql.Statement;
17
18
19 /**
20  * DDL functions performed in database
21  * @author eric bucarin
22  *
23  */
24 public class initdao {
25
26
27     public void initDB() {
28         Statement statement;
29
30         try {
31
32             Class.forName("com.mysql.jdbc.Driver").newInstance();
33             Connection connect = DriverManager
34                 .getConnection("jdbc:mysql://141.217.48.128:3306/song_chart?"
35                     + "user=root&password=3!0nMuk!");
36
37             statement = connect.createStatement();
38             statement.executeUpdate("DROP TABLE IF EXISTS song");
39             String sqlstmt = " create database song_chart;\r\n"
40                 + "CREATE TABLE song (\r\n"
41                 + "    songID INTEGER AUTO_INCREMENT,\r\n"
42                 + "    genre VARCHAR(30),\r\n"
43                 + "    streams BIGINT,\r\n"
44                 + "    song_name VARCHAR(30),\r\n"
45                 + "    bpm SMALLINT,\r\n"
46                 + "    album_name VARCHAR(30),\r\n"
47                 + "    artist_name VARCHAR(30),\r\n"
48                 + "    PRIMARY KEY (songID)\r\n"
49                 + ");\r\n"
50             + "alter table song add(artistID INTEGER);\r\n"
51             + "alter table song add FOREIGN KEY (artistID) REFERENCES artist(artistID);\r\n"
52             + "\r\n"
53             + "CREATE TABLE artist (\r\n"
54             + "    artistID INTEGER AUTO_INCREMENT,\r\n"
55             + "    nationality VARCHAR(30),\r\n"
56             + "    total_albums SMALLINT,\r\n"
57             + "    awards SMALLINT,\r\n"
58             + "    artist_name VARCHAR(30),\r\n"
59             + "    PRIMARY KEY (artistID)\r\n"
60             + ");\r\n"
61             + "CREATE TABLE album (\r\n"
62             + "    albumID INTEGER AUTO_INCREMENT,\r\n"
63             + "    total_streams BIGINT,\r\n"
64             + "    album_name VARCHAR(30),\r\n"
65             + "    rating SMALLINT,\r\n"
66             + "    length TIME,\r\n"
67             + "    artist_name VARCHAR(30),\r\n"
68             + "    artistID INTEGER,\r\n"
69             + "    PRIMARY KEY (albumID),\r\n"
70             + "    FOREIGN KEY (artistID) REFERENCES artist(artistID)\r\n"
71             + ");\r\n"
72             + "CREATE TABLE top_chart (\r\n"
73             + "    week_of DATE,\r\n"
74             + "    genre VARCHAR(30),\r\n"
75             + "    streams BIGINT,\r\n"
76             + "    song_name VARCHAR(30),\r\n"
77             + "    bpm SMALLINT,\r\n"
78             + "    album_name VARCHAR(30),\r\n"
79             + "    songID INTEGER,\r\n"
80             + "    artist_name VARCHAR(30),\r\n"
81             + "    PRIMARY KEY (week_of),\r\n"
82             + "    FOREIGN KEY (songID) REFERENCES song(songID)\r\n"
83             + ");\r\n"
84             + "CREATE TABLE tb_user \r\n"
85             + "(\r\n"
86             + "    \r\n"
87             + "    username VARCHAR(50) primary key,\r\n"
88             + "    password VARCHAR(50) NOT NULL,\r\n"
89             + "    email VARCHAR(50) NOT NULL\r\n"
90             + ");\r\n"
91             + "show tables;\r\n"
92             + "INSERT INTO tb_user VALUES\r\n"
93             + "('eric', 'pass', 'e@uic.edu'),\r\n"
94             + "('john', 'pass1', 'j@uic.edu'),\r\n"
95             + "('dom', 'pass2', 'd@uic.edu');\r\n"
96             + "\r\n"
97             + "INSERT INTO top_chart VALUES #genre VARCHAR(30), streams BIGINT, song_name VARCHAR(30),\r\n"
98             + "    #bpm SMALLINT, album_name VARCHAR(30), artist_name VARCHAR(30)\r\n"
99             + "    ('2021-07-16', 'pop', 1600000000, 'blinding lights', 86, 'after hours', 4, 'the weeknd'),\r\n"
100             + "    ('2021-07-17', 'pop', 1600000000, 'save your tears', 118, 'after hours', 5, 'the weeknd'),\r\n"
101             + "    ('2019-07-17', 'rap', 1500000000, 'sicko mode', 78, 'astro world', 6, 'travis scott');\r\n"
102             + "\r\n"
103             + "\r\n"
104             + "INSERT INTO song VALUES #genre VARCHAR(30), streams BIGINT, song_name VARCHAR(30),\r\n"
105             + "    #bpm SMALLINT, album_name VARCHAR(30), artist_name VARCHAR(30)\r\n"
106             + "    ('1, 'psychodelic', 829000000, 'the less i know the better', 117, 'currents', 'tame impala',1),\r\n"
107             + "    ('2, 'psychodelic', 144000000, 'new person, same old mistakes', 152, 'currents', 'tame impala',1),\r\n"
108             + "    ('3, 'psychodelic', 265000000, 'let it happen', 123, 'currents', 'tame impala',1);\r\n"
109             + "    INSERT INTO song VALUES (4, 'pop', 1600000000, 'blinding lights', 86, 'after hours', 'theweeknd',2);\r\n"
110             + "    INSERT INTO song VALUES (5, 'pop', 600000000, 'save your tears', 118, 'after hours', 'theweeknd',2);\r\n"
111             + "    INSERT INTO song VALUES (6, 'rap', 1500000000, 'sicko mode', 78, 'astro world', 'travis scott',3);\r\n"
112             + "\r\n"
113             + "\r\n"
114             + "INSERT INTO artist VALUES #genre VARCHAR(30), streams BIGINT, song_name VARCHAR(30),\r\n"
115             + "    #bpm SMALLINT, album_name VARCHAR(30), artist_name VARCHAR(30)\r\n"
116             + "    ('1, 'australian', 4, 16, 'tame impala'),\r\n"
117             + "    ('2, 'canadian', 5, 59, 'the weeknd'),\r\n"
118             + "    ('3, 'american', 3, 0, 'travis scott');\r\n"
119             + "\r\n"
120             + "INSERT INTO album VALUES \r\n"
121             + "    ('1, 1600000000, 'currents', 84, '00:51:12', 'tame impala',1),\r\n"
122             + "    ('2, 5600000000, 'after hours', 80, '00:56:19', 'the weeknd',2),\r\n"
123             + "    ('3, 6100000000, 'astroworld', 85, '00:58:33', 'travis scott',3);\r\n"
124             + "    ";
125
126             statement.executeUpdate(sqlstmt);
127
128         } catch (SQLException e) {
129             throw new RuntimeException(e);
130         } catch (InstantiationException e) {
131             // TODO Auto-generated catch block
132         }
133     }
134 }
```

Outline

Task List

user.dao

initdao

initDB() : void

Markers

Properties

Servers

Data Source Explorer

Snippets

Console

2 errors, 52 warnings, 25 others

| Description | Resource | Path | Location | Type |
|------------------------------|----------|------|----------|------|
| > ⚠ JSP Problem (18 items) | | | | |
| > ✖ Java Problems (36 items) | | | | |

Writable

Smart Insert

93 : 58 : 3233



Skip the meeting and record a quick video instead.

Sign in Get Loom for Free

Libraries

- Personal
- Shared
- Team
- Starred

Learn more

- Ways to use Loom
- Pricing
- About us

315 views Share Sign in Get Loom for Free

eclipse-workspace2 - CS480Demo/src/main/java/user/web/servlet/initserver.java - Eclipse IDE

Project Explorer

- CS480Demo
 - Deployment Descriptor: CS480Demo
 - JAX-WS Web Services
 - JRE System Library [JavaSE-16]
 - src/main/java
 - user.dao
 - initdao.java
 - UserDao.java
 - user.domain
 - user.service
 - user.web.servlet
 - findAll.java
 - initserver.java
 - initserver2.java
 - UserServletLogin.java
 - UserServletRegister.java
 - Server Runtime [Apache Tomcat v9.0]
 - Maven Dependencies
 - Deployed Resources
 - build
 - classes
 - user
 - dao
 - initDao.class
 - UserDao.class
 - domain
 - service
 - web
 - servlet
 - findAll.class
 - UserServletLogin.class
 - UserServletRegister.class
 - src
 - main
 - java
 - user

user.sql initDao.class initserver.java initdao.java

```
1 package user.web.servlet;
2
3 import java.io.IOException;
21
22 /**
23  * Servlet implementation class UserServlet
24  */
25
26 public class initserver extends HttpServlet {
27     private static final long serialVersionUID = 1L;
28
29     /**
30      * @see HttpServlet#HttpServlet()
31      */
32     public initserver() {
33         super();
34         // TODO Auto-generated constructor stub
35     }
36
37     /**
38      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
39      */
40     protected void doGet(HttpServletRequest request, HttpServletResponse response) {
41         doPost(request, response);
42     }
43
44     /**
45      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
46      */
47     protected void doPost(HttpServletRequest request, HttpServletResponse response) {
48         initdao newObj = new initdao();
49         newObj.initDB();
50     }
51 }
52
53
```

user.web.servlet

 - initserver
 - serialVersionUID
 - initserver()
 - doGet(HttpServletRequest request, HttpServletResponse response)
 - doPost(HttpServletRequest request, HttpServletResponse response)

Markers Properties Servers Data Source Explorer Snippets Console

4 errors, 41 warnings, 25 others

| Description | Resource | Path | Location | Type |
|-------------|--------------|---------------|----------|------|
| Writable | Smart Insert | 43 : 1 : 1098 | | |

Quick Access

Outline Task List

- user.dao
 - InitializeDao
 - initDB() : void

nger for a complete list of JARs that were

5.5 Comman...

Comment



Skip the meeting and record a quick video instead.

Sign in Get Loom for Free

- Libraries
- Personal
 - Shared
 - Team
 - Starred
- Learn more
- Ways to use Loom
 - Pricing
 - About us

eclipse-workspace2 - CS480Demo/src/main/java/user/web/servlet/initserver.java - Eclipse IDE

Project Explorer

- CS480Demo
 - Deployment Descriptor: CS480Demo
 - JAX-WS Web Services
 - JRE System Library [JavaSE-16]
 - src/main/java
 - user.dao
 - initdao.java
 - UserDao.java
 - user.domain
 - user.service
 - user.web.servlet
 - findAll.java
 - initserver.java
 - initserver2.java
 - UserServletLogin.java
 - UserServletRegister.java
 - Server Runtime [Apache Tomcat v9.0]
 - Maven Dependencies
 - Deployed Resources
 - build
 - classes
 - user
 - dao
 - initDao.class
 - UserDao.class
 - domain
 - service
 - web
 - servlet
 - findAll.class
 - UserServletLogin.class
 - UserServletRegister.cl

user.sql initDao.class *initserver... initdao.java

```
1 package user.web.servlet;
2
3 import java.io.IOException;
22
23 /**
24  * Servlet implementation class UserServlet
25  */
26
27 public class initserver extends HttpServlet {
28     private static final long serialVersionUID = 1L;
29
30     /**
31      * @see HttpServlet#HttpServlet()
32      */
33     public initserver() {
34         super();
35         // TODO Auto-generated constructor stub
36     }
37
38     /**
39      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
40      */
41     protected void doGet(HttpServletRequest request, HttpServletResponse response) {
42         doPost(request, response);
43     }
44
45     /**
46      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
47      */
48     protected void doPost(HttpServletRequest request, HttpServletResponse response) {
49         initdao newObj = new initdao();
50         newObj.initDB();
51     }
52 }
53
54
```

user.web.servlet

- initserver
 - serialVersionUID :
 - initserver()
 - doGet(HttpServletRequest request, HttpServletResponse response)
 - doPost(HttpServletRequest request, HttpServletResponse response)

Markers Properties Servers Data Source Explorer Snippets Console

4 errors, 41 warnings, 25 others

| Description | Resource | Path | Location | Type |
|-------------|--------------|---------------|----------|------|
| Writable | Smart Insert | 47 : 8 : 1219 | | |

Comment