

# **Homework #2**

## **Appendix**

### **Environment Setup**

---

KAIST

Prof. Myoung Ho Kim

# Docker Environment Setup

---

# Pre-installation

---

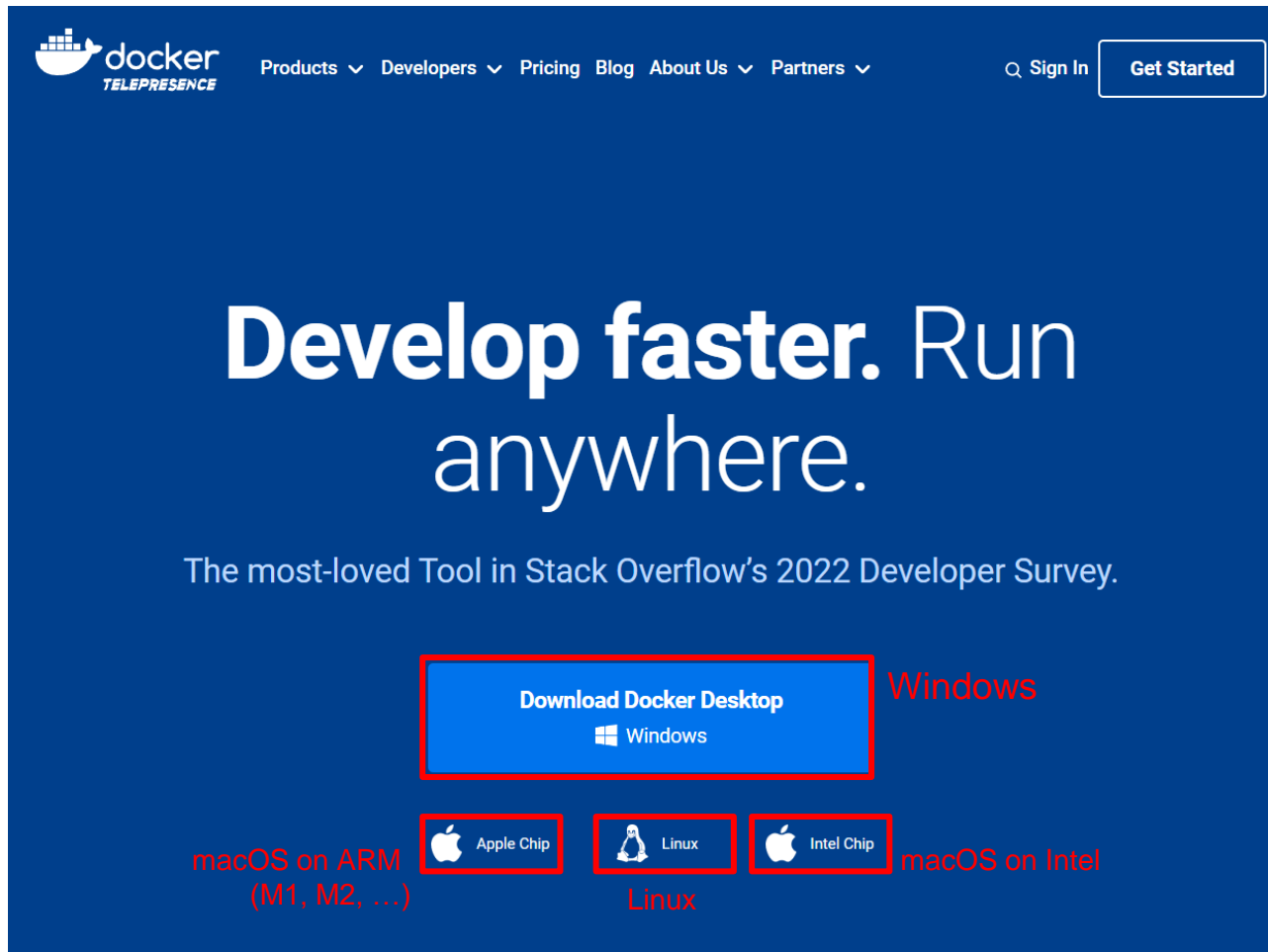
## ◆ Docker

- ◆ Requirement for Homework Environment
- ◆ We provide preset Docker configuration files for your convenience.
- ◆ <https://www.docker.com/>

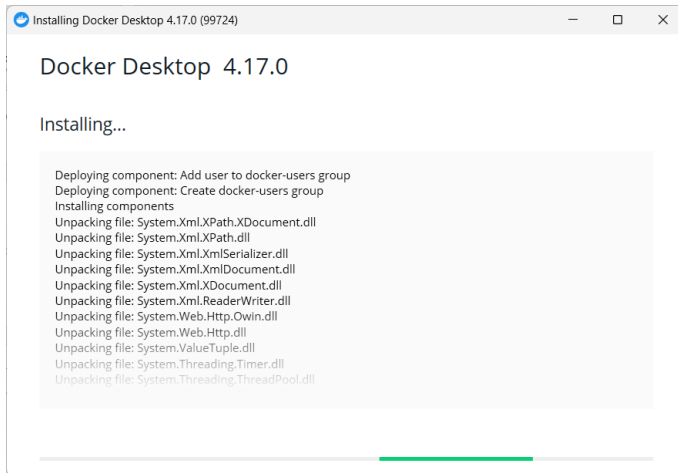
## ◆ Visual Studio Code

- ◆ Required for Development
- ◆ <https://code.visualstudio.com/>

# Install Docker



# Install Docker (References)



1. Install on Windows :

<https://docs.docker.com/desktop/install/windows-install/>

2. Install on Mac :

<https://docs.docker.com/desktop/install/mac-install/>

3. Install on Linux :

<https://docs.docker.com/desktop/install/linux-install/>

If you have errors installing Docker, refer to

<https://docs.docker.com/desktop/troubleshoot/topics/>

Or you can install a previous version of Docker :

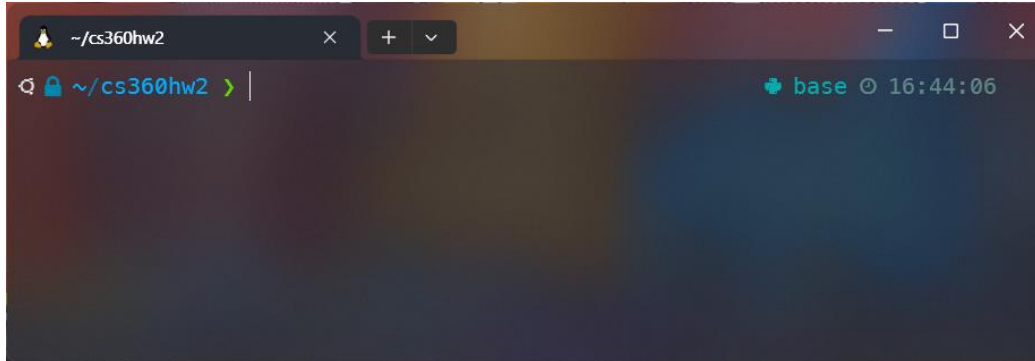
<https://docs.docker.com/desktop/release-notes/>

# HW File Directory

| 이름                   | 수정한 날짜              | 유형         | 크기  |
|----------------------|---------------------|------------|-----|
| config               | 2023-05-02 오전 12:22 | 파일 폴더      |     |
| HW                   | 2023-05-02 오전 12:22 | 파일 폴더      |     |
| ! docker-compose.yml | 2023-04-12 오전 9:19  | Yaml 원본 파일 | 1KB |
| Dockerfile           | 2023-04-12 오전 9:53  | 파일         | 2KB |
| whyyouwrong.txt      | 2023-04-05 오후 1:29  | 텍스트 문서     | 3KB |

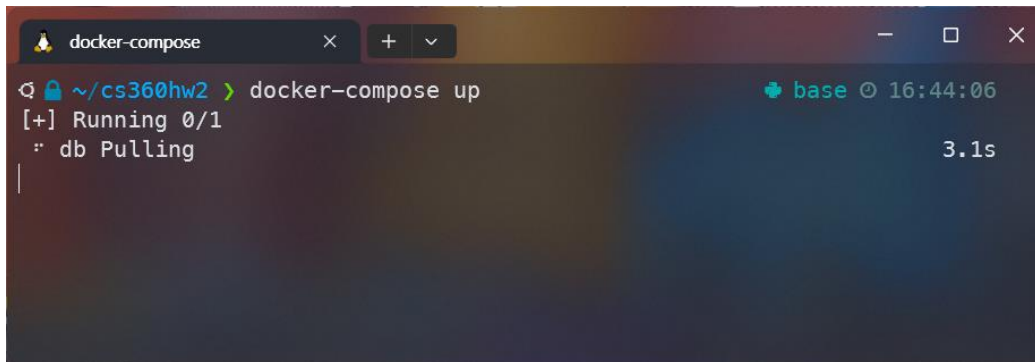
You should be given the above files in your HW directory

# Running Docker



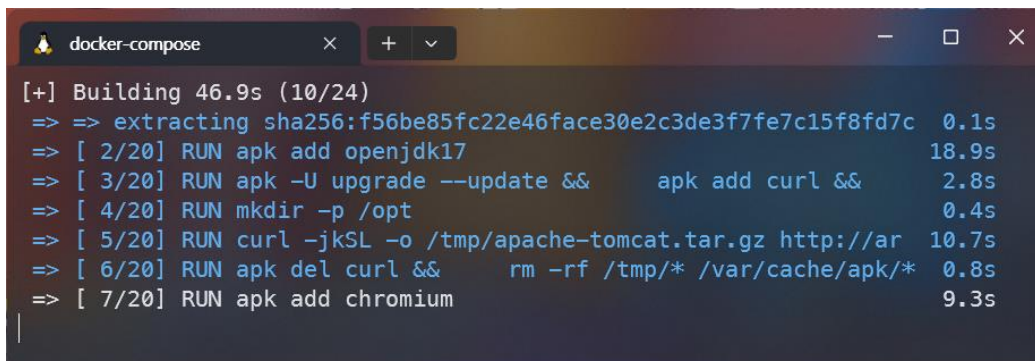
```
~/cs360hw2
~/cs360hw2 > |
```

Go to the HW directory.



```
docker-compose
~/cs360hw2 > docker-compose up
[+] Running 0/1
  * db Pulling                                3.1s
```

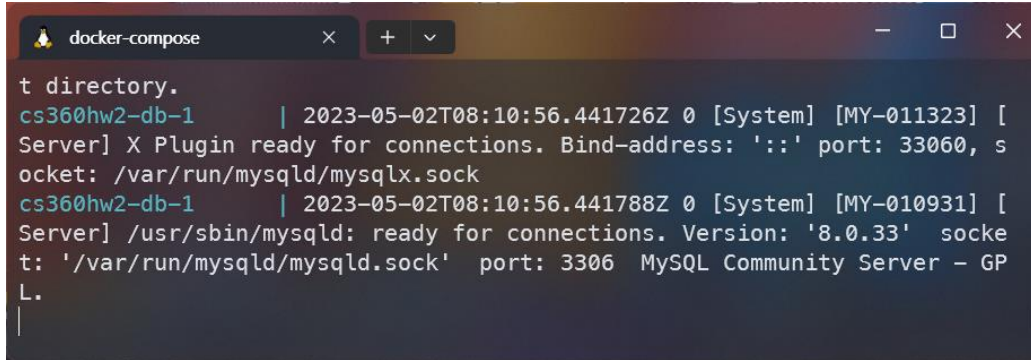
Run the command `docker-compose up`



```
docker-compose
[+] Building 46.9s (10/24)
=> => extracting sha256:f56be85fc22e46face30e2c3de3f7fe7c15f8fd7c 0.1s
=> [ 2/20] RUN apk add openjdk17 18.9s
=> [ 3/20] RUN apk -U upgrade --update && apk add curl && 2.8s
=> [ 4/20] RUN mkdir -p /opt 0.4s
=> [ 5/20] RUN curl -jksL -o /tmp/apache-tomcat.tar.gz http://ar 10.7s
=> [ 6/20] RUN apk del curl && rm -rf /tmp/* /var/cache/apk/* 0.8s
=> [ 7/20] RUN apk add chromium 9.3s
```

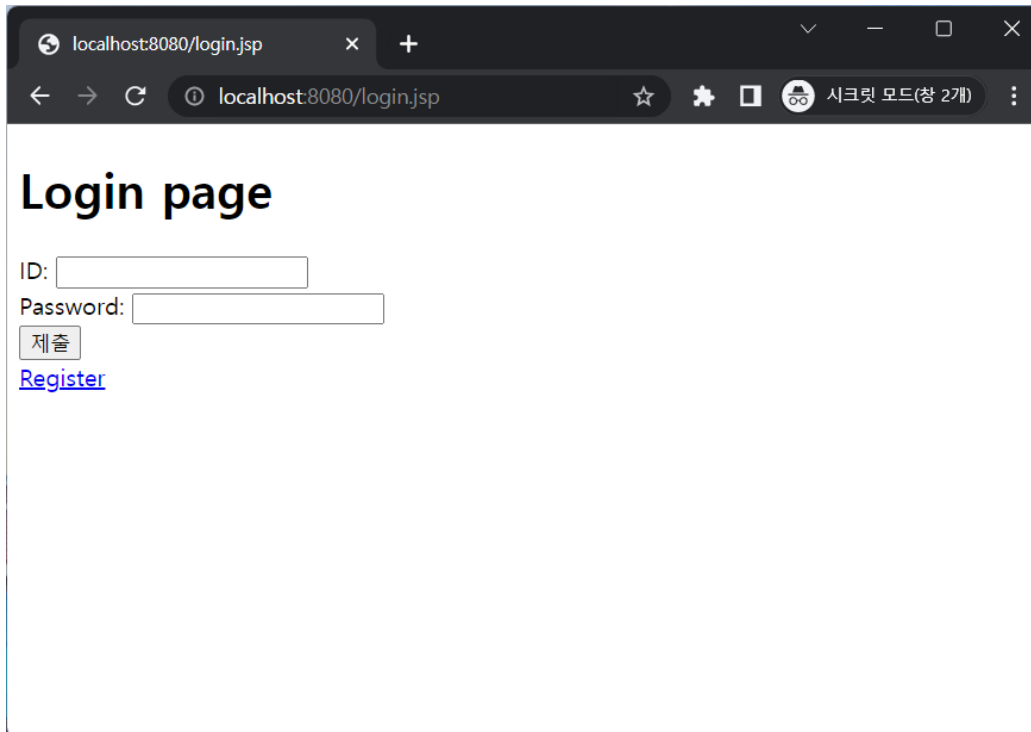
If it is your first time running,  
it will take a few minutes to configure.

# Running Docker

A terminal window titled 'docker-compose' showing the output of a Docker Compose command. The output indicates that two MySQL containers, 'cs360hw2-db-1' and 'cs360hw2-db-2', are ready for connections. The first container is on port 33060 and the second is on port 3306. The terminal text is as follows:

```
t directory.  
cs360hw2-db-1 | 2023-05-02T08:10:56.441726Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqld.sock  
cs360hw2-db-2 | 2023-05-02T08:10:56.441788Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.33' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.  
|
```

When the configuration is done, the terminal will pause at this screen.

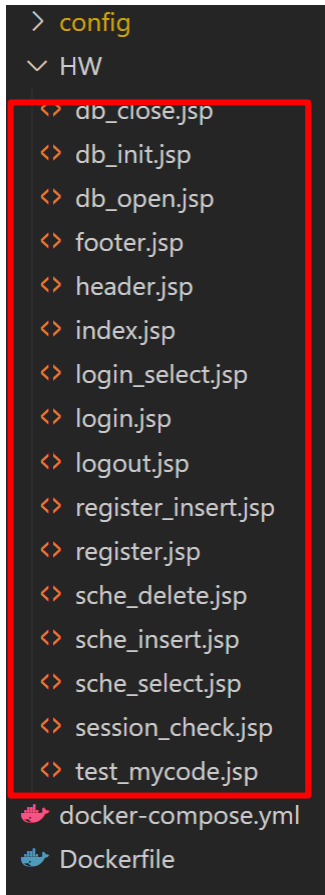


On your web browser, enter  
`localhost:8080/login.jsp`  
You will see the login page of this assignment.



# Writing Code

The code you have to modify is located inside the `HW` directory. Any changes made to the files will be applied directly to the web browser once you refresh the page.

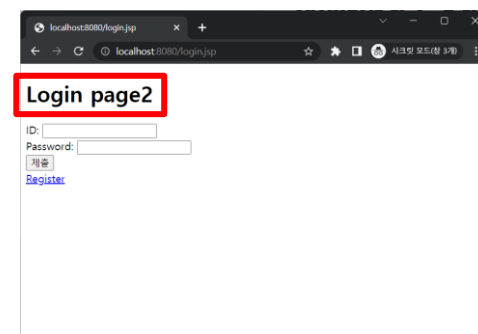
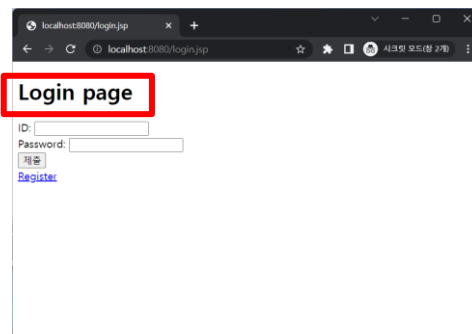


```
HW > login.jsp > ? > ?  
1 <%@ include file="header.jsp" %>  
2 <h1>Login page</h1>  
3 <form action="./login_select.jsp", method="post">  
4   <label for="id"> ID:  
5     <input type="text" name="id"> <br>  
6     <label for="id"> Password:  
7       <input type="password" name="password"> <br>  
8       <input type="submit">  
9 </form>  
10 <br>  
11 <a href="/register.jsp">Register</a>  
12 <%@ include file="footer.jsp" %>
```

save

```
HW > login.jsp > ? > h1  
1 <%@ include file="header.jsp" %>  
2 <h1>Login page2</h1>  
3 <form action= "./login_select.jsp", method="post">  
4   <label for="id"> ID:  
5     <input type="text" name="id"> <br>  
6     <label for="id"> Password:  
7       <input type="password" name="password"> <br>  
8       <input type="submit">  
9 </form>  
10 <br>  
11 <a href="/register.jsp">Register</a>  
12 <%@ include file="footer.jsp" %>
```

refresh



# Manual Environment Setup

---

When you cannot use the Docker environment  
(Not recommended)

# Pre-installation

---

## ◆ OpenJDK

- ◆ Required for Java development

- ◆ <https://docs.microsoft.com/en-us/java/openjdk/download>

## ◆ MySQL

- ◆ Required for RDBMS

- ◆ <https://dev.mysql.com/downloads/installer/>

## ◆ Visual Studio Code

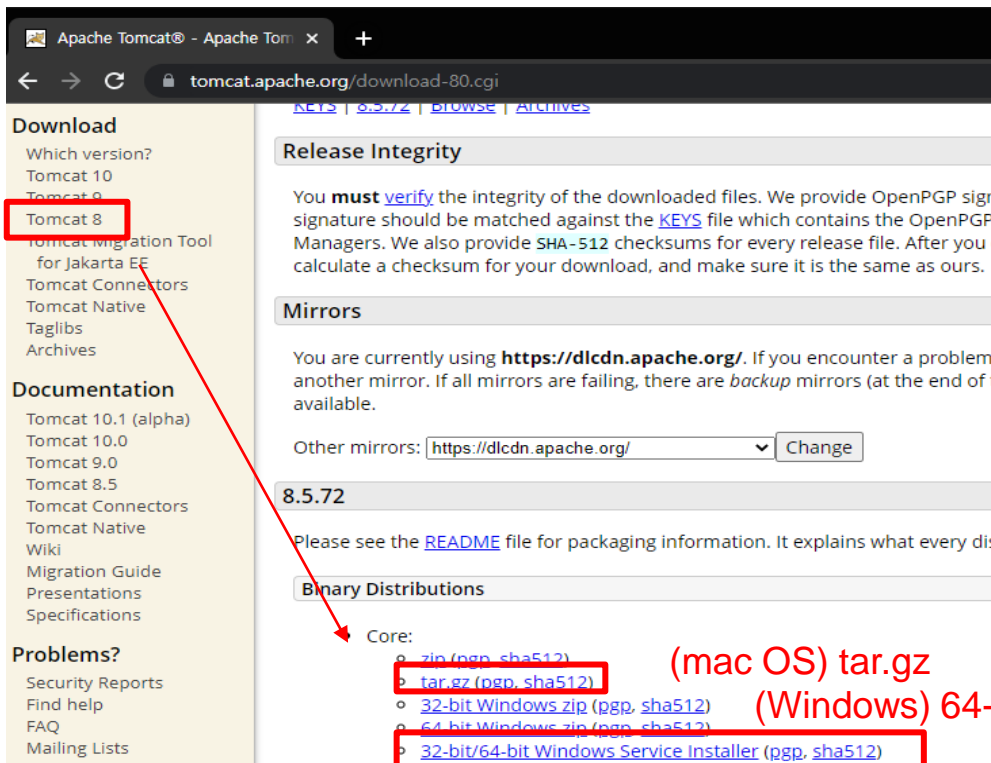
- ◆ Required for Development

- ◆ You need to install Java, MySQL extension

- ◆ <https://code.visualstudio.com/>

# Install Apache Tomcat

- ◆ Apache Tomcat 8 installation
  - Download the apache-tomcat file
  - <https://tomcat.apache.org/download-80.cgi>



If you use window OS,  
Click next button  
until the installation is over

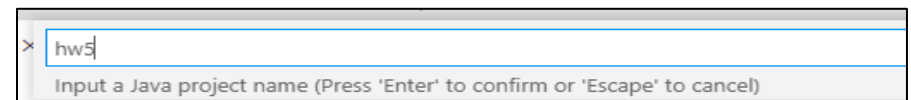
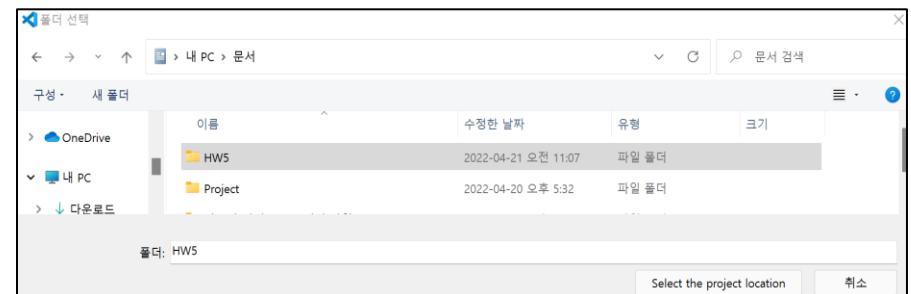
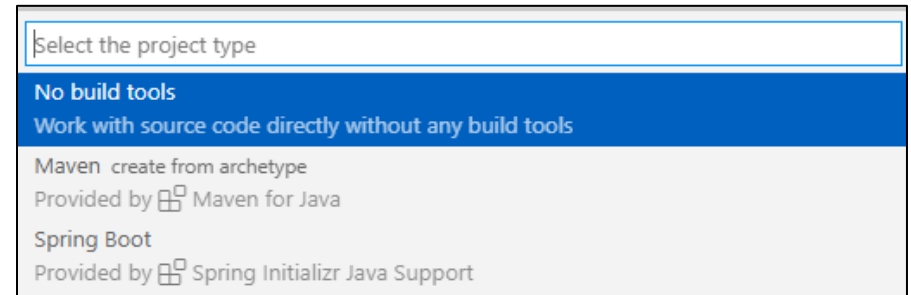
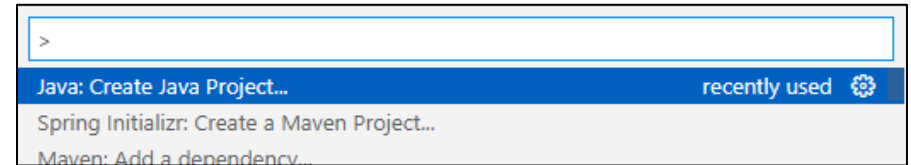
(mac OS) tar.gz

(Windows) 64-bit Window.zip

# Environment setup at VSCode

## ◆ Create a java project for hw2

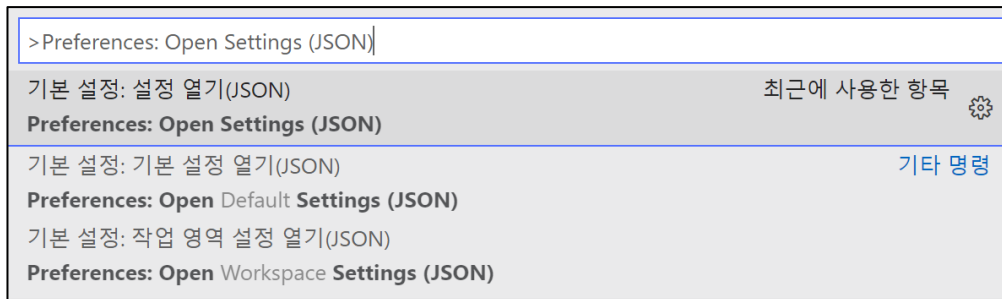
1. Open the command palette(Ctrl+Shift+P)
2. Select 'Java: Create Java Project'
3. Select 'No build tools'
4. Browse your folder to save your project
5. Input your java project name



# JDK Environment Setup at VSCode

## ◆ JDK Environment Setting

- ◆ Open the command palette (Ctrl+Shift+P)
- ◆ Add JDK path in settings.json



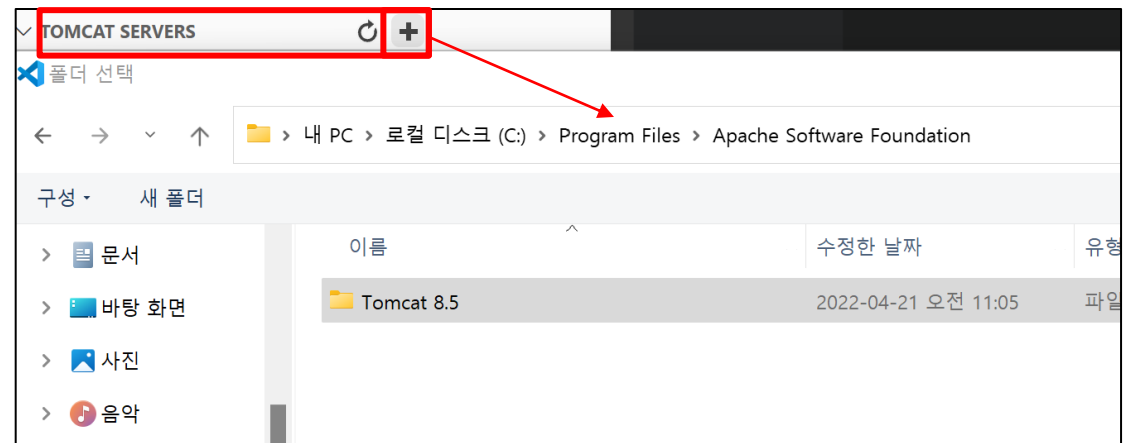
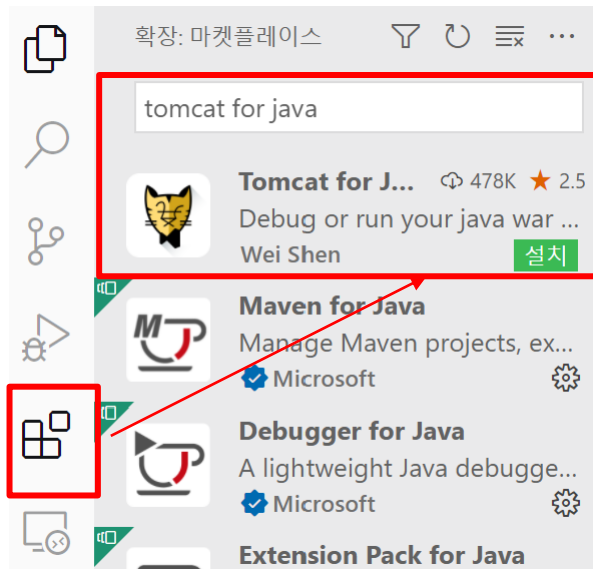
"java.jdt.ls.java.home": "[JDK Path]"

C: > Users > MJPark > AppData > Roaming > Code > User > {} settings.json > ...

```
1  {
2  |    "java.jdt.ls.java.home": "C:\\Program Files\\Microsoft\\jdk-17.0.2.8-hotspot"
3  }
```

# Using Tomcat with VSCode

- ◆ Download the 'Tomcat for Java' extension
  - Reload the vscode then you can see 'TOMCAT SERVERS'
- ◆ Open the Tomcat Server
  - Browse the 'Tomcat directory' that you downloaded
  - C:\Program Files\Apache Software Foundation\Tomcat 8.5



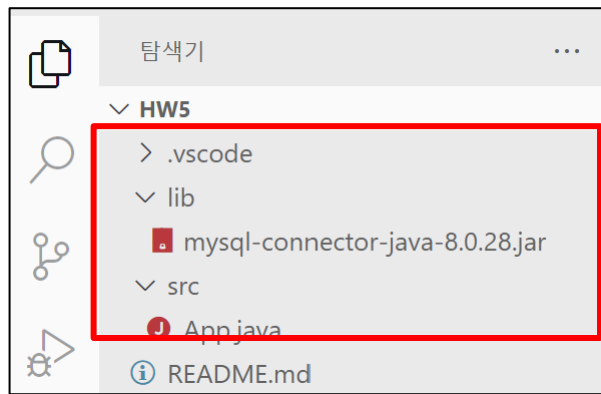
# Install JDBC

## ◆ Please download the 'mysql-connector-java-8.0.22.jar' file

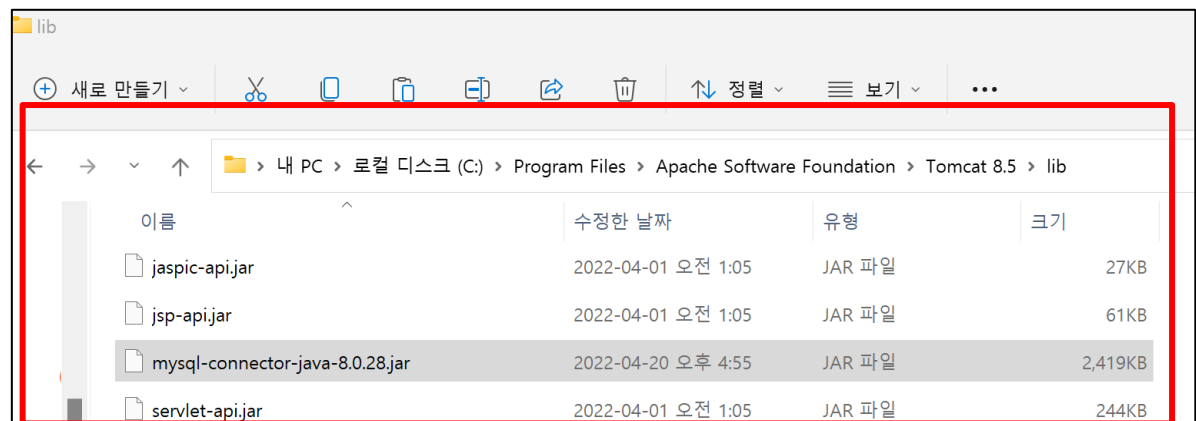
- <https://repo1.maven.org/maven2/mysql/mysql-connector-java/8.0.22/mysql-connector-java-8.0.22.jar>
- <https://repo1.maven.org/maven2/mysql/mysql-connector-java/8.0.28/mysql-connector-java-8.0.28.jar>

(For Example)

1) C:\Users\MJPark\Documents\HW\lib



2) (windows) C:\Program Files\Apache Software Foundation\Tomcat 8.5  
(macOS) \Downloads\apache-tomcat-8.5.72\lib



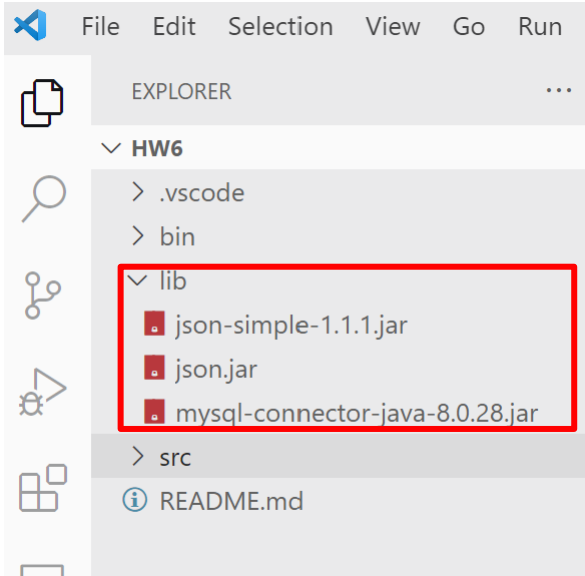


# Install JSON

- ◆ Download 'json' & 'json-simple-1.1.1.jar' in hw5.zip
- ◆ Copy 'json' & 'json-simple-1.1.1.jar', 'mysql-connector-java-8.0.22.jar' to  
**1) java project > lib directory & 2) tomcat > lib directory**

(For Example)

1) C:\Users\User\Desktop\hw5\hw5\lib



2) (windows)C:\Program Files\Apache Software Foundation\Tomcat 8.5\lib  
(macOS)\Downloads\apache-tomcat-8.5.72\lib

| « Program Files > Apache Software Foundation > Tomcat 8.5 > lib |                     |        |
|---|---------------------|--------|
| 이름  | 수정한 날짜              | 유형     |
| jasper-el   | 2022-04-01 오전 1:05  | JAR 파일 |
| jaspic-api  | 2022-04-01 오전 1:05  | JAR 파일 |
| json  | 2021-11-29 오전 12:55 | JAR 파일 |
| json-simple-1.1.1   | 2021-11-28 오후 11:52 | JAR 파일 |
| jsp-api   | 2022-04-01 오전 1:05  | JAR 파일 |
| mysql-connector-java-8.0.28                                     | 2022-05-12 오후 2:52  | JAR 파일 |

# Tomcat configuration

Step1) Click the right button of the mouse at Tomcat 8.5

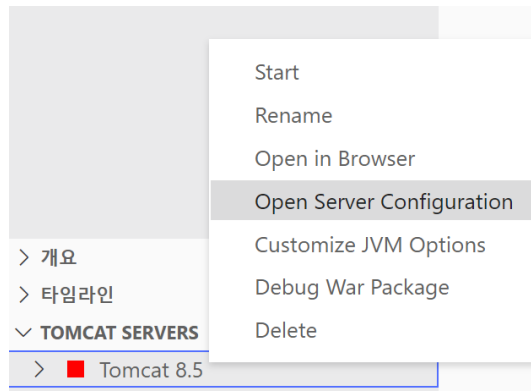
- Click the Open Server Configuration => then you can get 'server.xml' file

Step2) Modify the 'server.xml' file

- Connector port
- <Context />

2) Put `<Context docBase=" [Java project path]" path="" reloadable="true" />`

1) Change the port number



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Server port="8081" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener"/>
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on"/>
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container" type="org.apache.catalina.UserDatabase" description="User Database" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" URIEncoding="UTF-8" />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
        <Context docBase="C:\Users\MJPark\Documents\HW5\HW5\src\" path="" reloadable="true" />
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="logs" />
      </Host>
    </Engine>
  </Service>
</Server>
```

# Tomcat configuration

Step3) Open the 'Tomcat 8.5(apache-tomcat-8.5.72)/conf/context.xml' file

- Put the <Resource / > tag

C: > Program Files > Apache Software Foundation > Tomcat 8.5 > conf > context.xml

```
12 Unless required by applicable law or agreed to in writing, software
13 distributed under the License is distributed on an "AS IS" BASIS,
14 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15 See the License for the specific language governing permissions and
16 limitations under the License.
17 -->
18 <!-- The contents of this file will be loaded for each web application -->
19 <Context>
20
21 <!-- Default set of monitored resources. If one of these changes, the -->
22 <!-- web application will be reloaded. -->
23 <WatchedResource>WEB-INF/web.xml</WatchedResource>
24 <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
25
26 <Resource name="jdbc/mysql_connect"
27     auth="Container"
28     type="javax.sql.DataSource"
29     maxActive="100"
30     maxIdle="30"
31     maxWait="10000"
32     username="root"
33     password="root1234"
34     driverClassName="com.mysql.jdbc.Driver"
35     url="jdbc:mysql://localhost:3306/HW5"
36 />
```

Write your database(mysql)  
username & password

```
<Resource name="jdbc/mysql_connect"
    auth="Container"
    type="javax.sql.DataSource"
    maxActive="100"
    maxIdle="30"
    maxWait="10000"
    username="dbuser"
    password="dbuser1234"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/HW"
/>
```

# Tomcat configuration

Step4) Open the 'Tomcat 8.5(apache-tomcat-8.5.72)/webapps/ROOT/WEB-INF/web.xml'

- Put the <resource-ref> </resource-ref> tag

C: > Program Files > Apache Software Foundation > Tomcat 8.5 > webapps > ROOT > WEB-INF > web.xml

```
17  -->
18  <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
19  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
20  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
21  | | | | | http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
22  version="3.1"
23  metadata-complete="true">
24
25  <display-name>Welcome to Tomcat</display-name>
26  <description>
27  | Welcome to Tomcat
28  </description>
29  <resource-ref>
30  | <res-ref-name>jdbc/mysql_connect</res-ref-name>
31  | <res-type>javax.sql.DataSource</res-type>
32  | <res-auth>Container</res-auth>
33  </resource-ref>
34  </web-app>
```

<res-ref-name> is same as  
Resource name that we set at '/conf/context.xml'

```
<resource-ref>
  <res-ref-name>
    jdbc/mysql_connect
  </res-ref-name>
  <res-type>
    javax.sql.DataSource
  </res-type>
  <res-auth>
    Container
  </res-auth>
</resource-ref>
```

# How to Browse your JSP file in PC

- ◆ Restart the Tomcat Server
- ◆ Browse your JSP file with 'Open in Browser'
  - You can access the files with any web browser

The screenshot illustrates the process of browsing a JSP file in VS Code. On the left, the 'TOMCAT SERVERS' panel shows a context menu for a Tomcat instance with options like 'Start', 'Restart', and 'Open in Browser'. The 'Open in Browser' option is highlighted. In the center, the file explorer shows a project structure with a 'src' directory containing 'index.jsp'. On the right, the 'index.jsp' file is open, displaying the following code:

```
1 <%@ page contentType="text/html; charset=UTF-8"%>
2 <html>
3   <body>
4     <%= String hello="Hello, World!"; %>
5     <p><%=hello%></p>
6   </body>
7 </html>
```

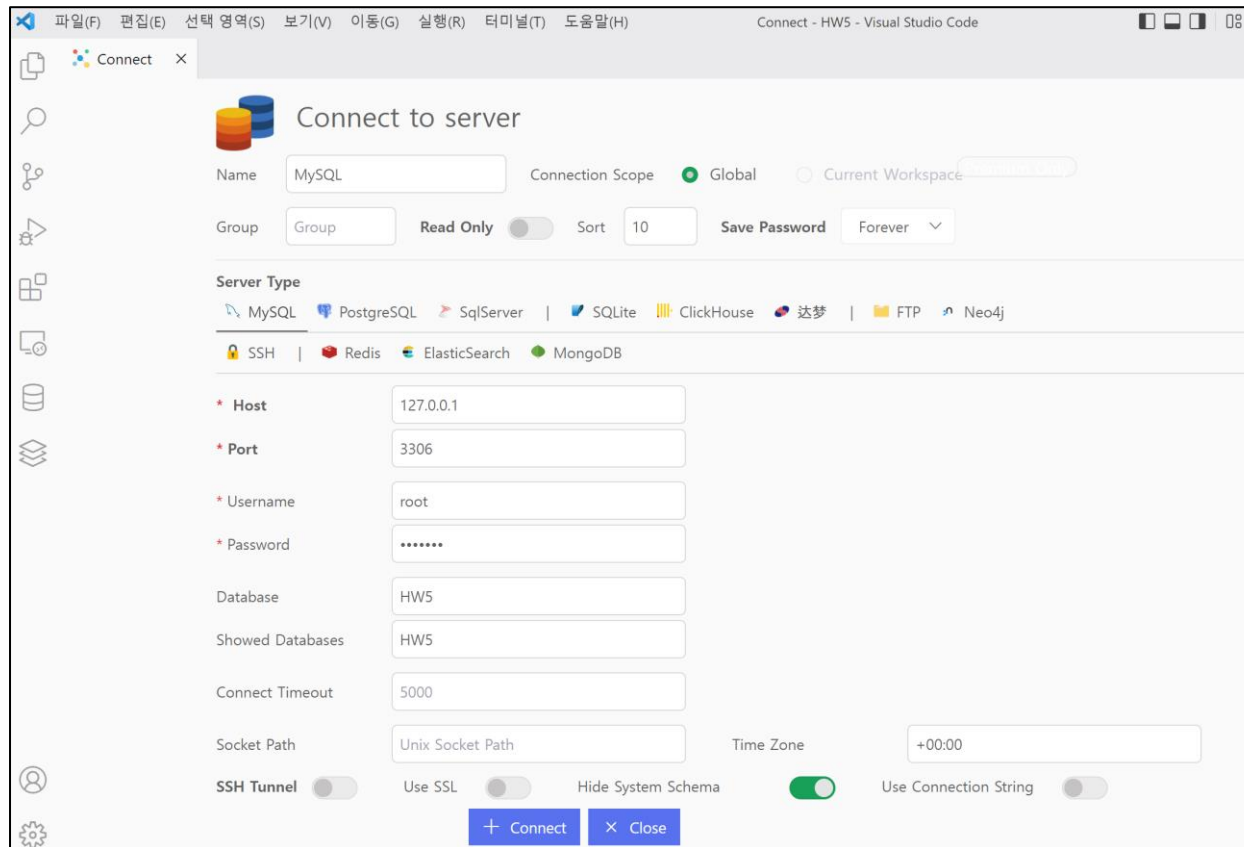
Below the code, a browser window is shown with the address bar displaying 'localhost:8080/index.jsp' and the page content 'Hello, World!'. Below the browser window, the URL format is explained:

<http://localhost:portnumber/<filename>>  
ex) <http://localhost:9090/index.jsp>

# Connect to MySQL

## Connect to MySQL

- Connect to your MySQL server that we used in Assignment 1
- Create the database name



# Example

```
<body>
  <table border="1">
    <tr>
      <td> model </td>
      <td> marker </td>
      <td> color </td>
      <td> type </td>
      <td> price </td>
      <td> delete </td>
    <%
      try {
        stmt = con.createStatement();
        String query = "SELECT product.model, maker, color, printer.type, price
                        FROM product, printer
                        WHERE product.model=printer.model ORDER BY model";
        rs = stmt.executeQuery(query);
        while(rs.next()) {
          %>
          <tr>
            <td><%=rs.getString(1)%></td>
            <td><%=rs.getString(2)%></td>
            <td><%=rs.getString(3)%></td>
            <td><%=rs.getString(4)%></td>
            <td><%=rs.getString(5)%></td>
            <td><a href="del_register.jsp?model=<%=rs.getString(1)%>">del</a></td>
          <tr>
        <%
      }
    } catch (SQLException e) {
      out.println(e.toString());
    }
  %>
</table>
</body>
```

Search.jsp

← → ↻ ⓘ localhost:9090/search.jsp

| model | marker | color | type    | price | delete              |
|-------|--------|-------|---------|-------|---------------------|
| 3001  | D      | true  | ink-jet | 1999  | <a href="#">del</a> |
| 3002  | D      | true  | ink-jet | 2499  | <a href="#">del</a> |
| 3003  | B      | false | laser   | 3599  | <a href="#">del</a> |
| 3004  | H      | true  | laser   | 3499  | <a href="#">del</a> |
| 3010  | I      | true  | dry     | 2000  | <a href="#">del</a> |
| 3015  | A      | false | ink-jet | 2456  | <a href="#">del</a> |
| 3017  | X      | true  | ink-jet | 1300  | <a href="#">del</a> |