

IntelliJ IDEA 개발환경 구성

@taewanme

Web Application 개 발환경 준비



개발 환경 준비

- Java: <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
 - sdkman을 이용한 설치 완료
- Maven: <https://maven.apache.org>
 - sdkman을 이용한 설치 완료
- Tomcat: <https://tomcat.apache.org>
- IntelliJ: <https://www.jetbrains.com/idea>

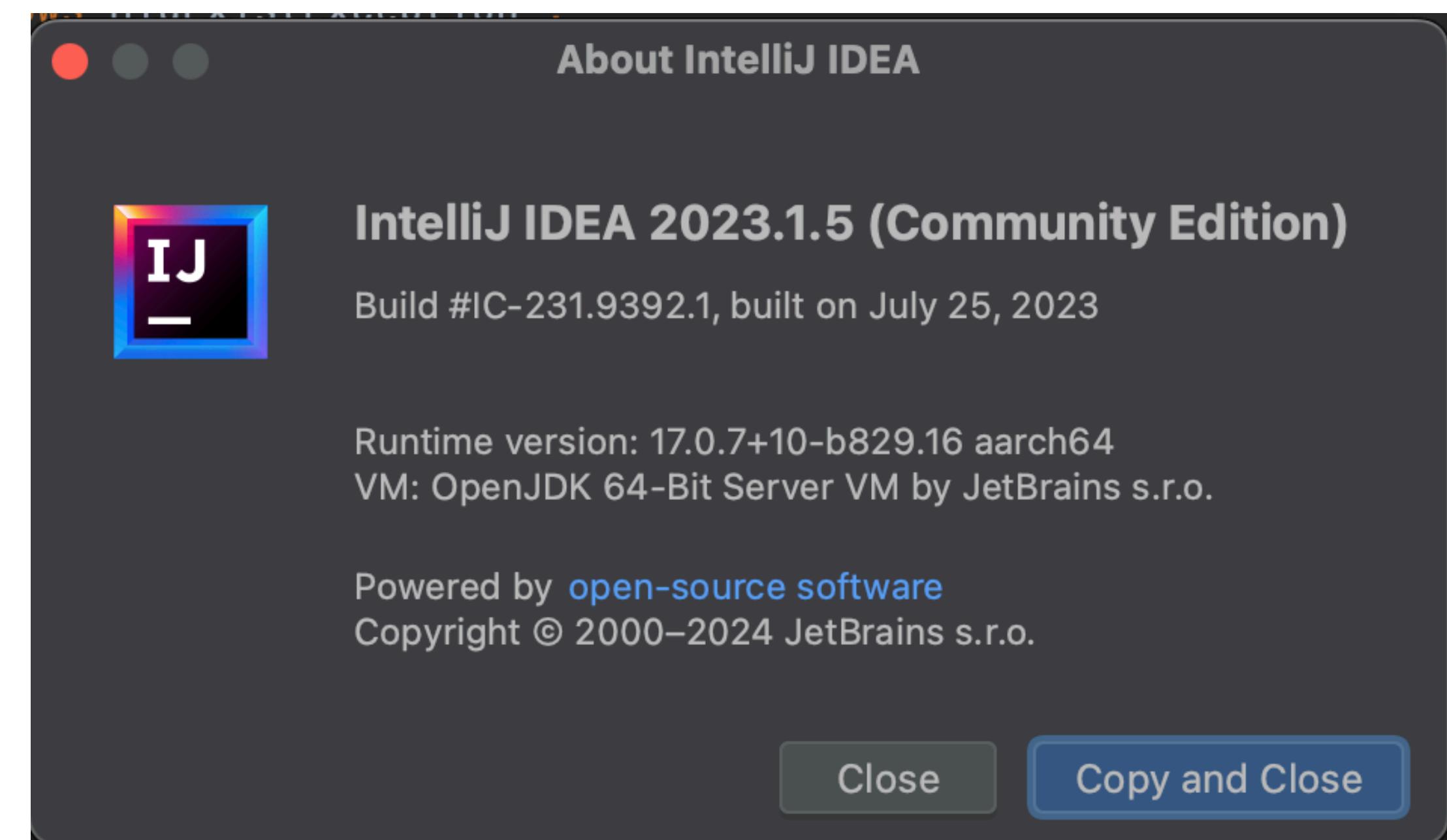
Web Application Server (WAS)

웹 애플리케이션과 서버 환경을 만들어 동작하는 SW: Middleware

- Apache Tomcat
- JBoss
- Jetty
- Netty

IDEA: IntelliJ

- IDE: Integrated Development Environment
- IntelliJ IDEA, Eclipse, Netbeans
-



Java 환경 구성

LTS[Long Time Support] 버전 확인

- sdkman 설치
 - curl -s "https://get.sdkman.io" | bash
- Java 설치
 - sdk list java
 - sdk install java [version]
 - [ex]sdk install java 21.0.3-oracle
- 환경 변수: JAVA_HOME

- [window] PATH 환경 변수 수정

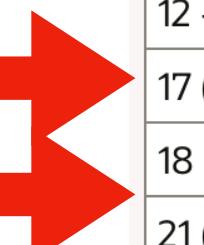
- %JAVA_HOME%/bin

- 설치 확인

- java -version
 - javac -version

<https://www.oracle.com/kr/java/technologies/java-se-support-roadmap.html>

Oracle Java SE Support Roadmap*†				
Release	GA Date	Premier Support Until	Extended Support Until	Sustaining Period
8 (LTS)**	March 2014	March 2022	December 2030****	Indefinite
9 - 10 (non-LTS)	September 2017 - March 2018	March 2018 - September 2018	Not Available	Indefinite
11 (LTS)	September 2018	September 2023	January 2032****	Indefinite
12 - 16 (non-LTS)	March 2019 - March 2021	September 2019 - September 2021	Not Available	Indefinite
17 (LTS)	September 2021	September 2026****	September 2029****	Indefinite
18 - 20 (non-LTS)	March 2022 - March 2023	September 2022 - September 2023	Not Available	Indefinite
21 (LTS)	September 2023	September 2028****	September 2031****	Indefinite
22 (non-LTS)	March 2024	September 2024	Not Available	Indefinite
23 (non-LTS)***	September 2024	March 2025	Not Available	Indefinite
24 (non-LTS)***	March 2025	September 2025	Not Available	Indefinite
25 (LTS)***	September 2025	September 2030	September 2033	Indefinite



maven

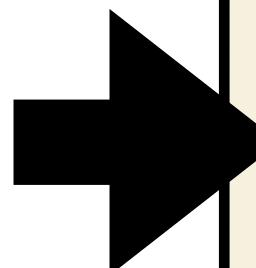
- sdkman 설치
 - sdk list maven
 - sdk install maven 3.8.6
- 환경 변수: MAVEN_HOME
 - [window] PATH 환경 변수 수정
 - %MAVEN_HOME%/bin
- 설치 확인
 - mvn clean compile package

```
=====
Available Maven Versions
=====
4.0.0-beta-3      3.8.2          3.2.2
3.9.7              3.8.1          3.2.1
3.9.6              3.6.3          3.1.1
3.9.5              3.6.2          3.1.0
3.9.4              3.6.1          3.0.5
3.9.3              3.6.0          3.0.4
3.9.2              3.5.4
3.9.1              3.5.3
3.9.0              3.5.2
3.8.8              3.5.0
3.8.7              3.3.9
> * 3.8.6          3.3.3
      3.8.5          3.3.1
      3.8.4          3.2.5
      3.8.3          3.2.3
=====
+ - local version
* - installed
> - currently in use
=====
(END)
```

Apache Tomcat 설치

<https://tomcat.apache.org>

- Tomcat 9
- 다운로드
 - core
 - zip
- 64-bit Windows
- version: 9.0.89
- 압축풀기
- 환경변수 등록



The screenshot shows the Apache Tomcat® website. At the top left is the Tomcat logo (a yellow cat). To its right is a search bar with a 'GO' button. Below the search bar is a banner for 'COMMUNITY OVER CODE' with the text 'Merge with the ASF EUniverse' and 'June 3-5, 2024 Bratislava, Slovakia'. The main content area has a header 'Apache Tomcat'. It contains text about the software being an open source implementation of Jakarta Servlet, Jakarta Pages, Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations, and Jakarta Authentication. It also mentions the Jakarta EE platform and the Apache License version 2. A 'click here' link is present. Below this is a section titled 'Tomcat 10.1.24 Released' with a note about the release being part of the Jakarta EE 10 platform. It discusses applications running on Tomcat 9 and earlier not running on Tomcat 10 without changes, and provides a migration tool. A 'Download' link is at the bottom of this section. Another section below is titled 'Tomcat 11.0.0-M20 Released', announcing a milestone release targeted at Jakarta EE 11. It notes the move from Java EE to Jakarta EE and the availability of a migration tool for applications.

Apache Tomcat®

Apache Tomcat

The Apache Tomcat® software is an open source implementation of the [Jakarta Servlet](#), [Jakarta Pages](#), [Jakarta Expression Language](#), [Jakarta WebSocket](#), [Jakarta Annotations](#) and [Jakarta Authentication](#).

The Jakarta EE platform is the evolution of the Java EE platform. Tomcat 10 and later implement specifications developed as part of Jakarta EE. Tomcat 9 and earlier implement specifications developed as part of Java EE.

The Apache Tomcat software is developed in an open and participatory environment and released under the [Apache License version 2](#). The Apache Tomcat project is intended to be a collaboration development project. To learn more about getting involved, [click here](#).

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the [User Stories](#) page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat 10.1.24 Released

The Apache Tomcat Project is proud to announce the release of version 10.1.24 of Apache Tomcat. This release implements specifications that are part of the Jakarta EE 10 platform.

Applications that run on Tomcat 9 and earlier will not run on Tomcat 10 without changes. Java EE based applications designed for Tomcat 9 and earlier may be placed in the `$CATALINA_BASE/webapps` directory. This conversion is performed using the [Apache Tomcat migration tool for Jakarta EE tool](#) which is also available as a separate [download](#) for off-line use.

The notable changes in this release are:

- Correct error handling for asynchronous requests
- Refactor HTTP header parsing to use common parsing code and fix non-blocking reads of chunked request bodies including trailer fields
- WebDAV locking handling fixes

Full details of these changes, and all the other changes, are available in the [Tomcat 10.1 changelog](#).

[Download](#)

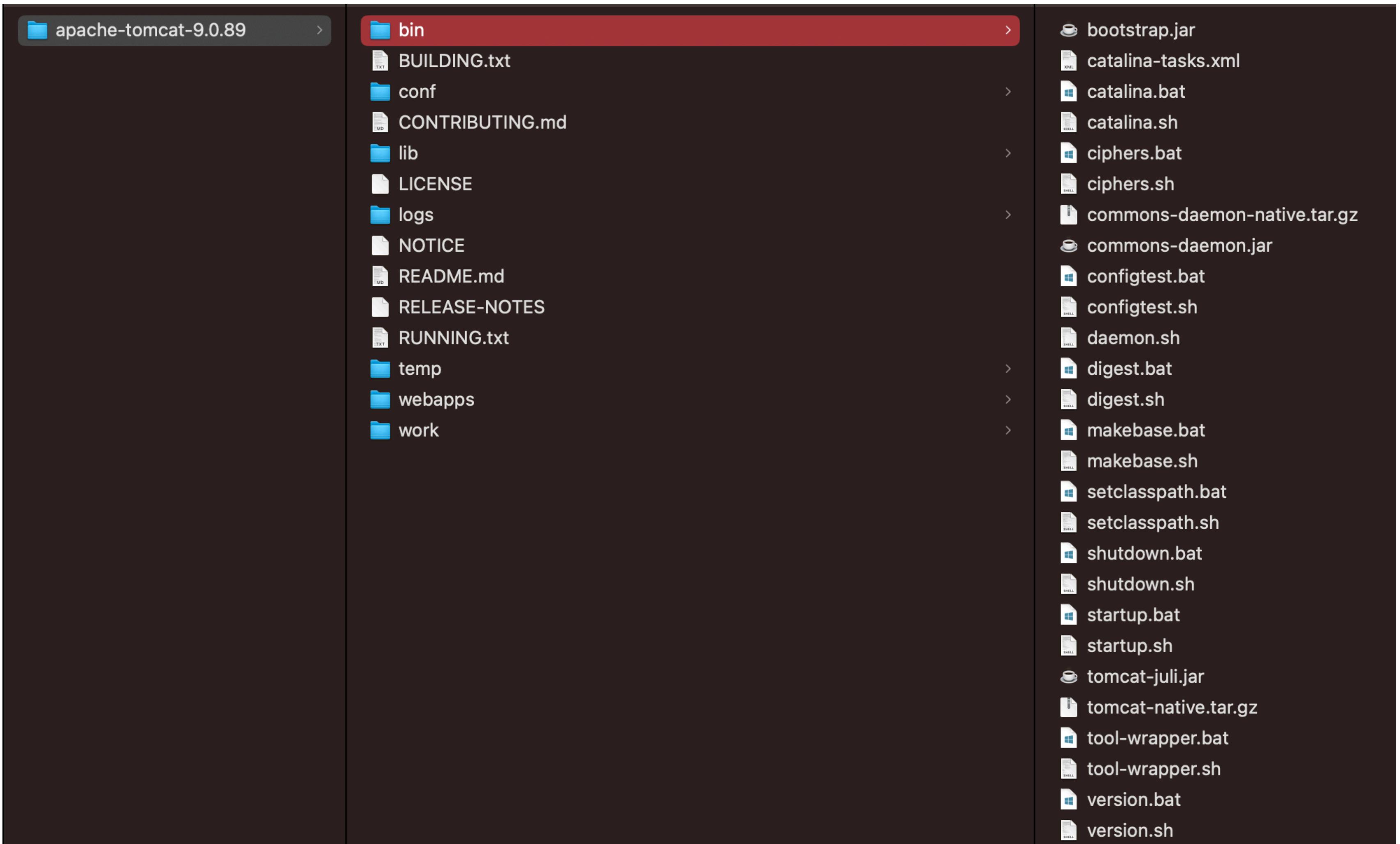
Tomcat 11.0.0-M20 Released

The Apache Tomcat Project is proud to announce the release of version 11.0.0-M20 (alpha) of Apache Tomcat. This release is a milestone release and is targeted at Jakarta EE 11.

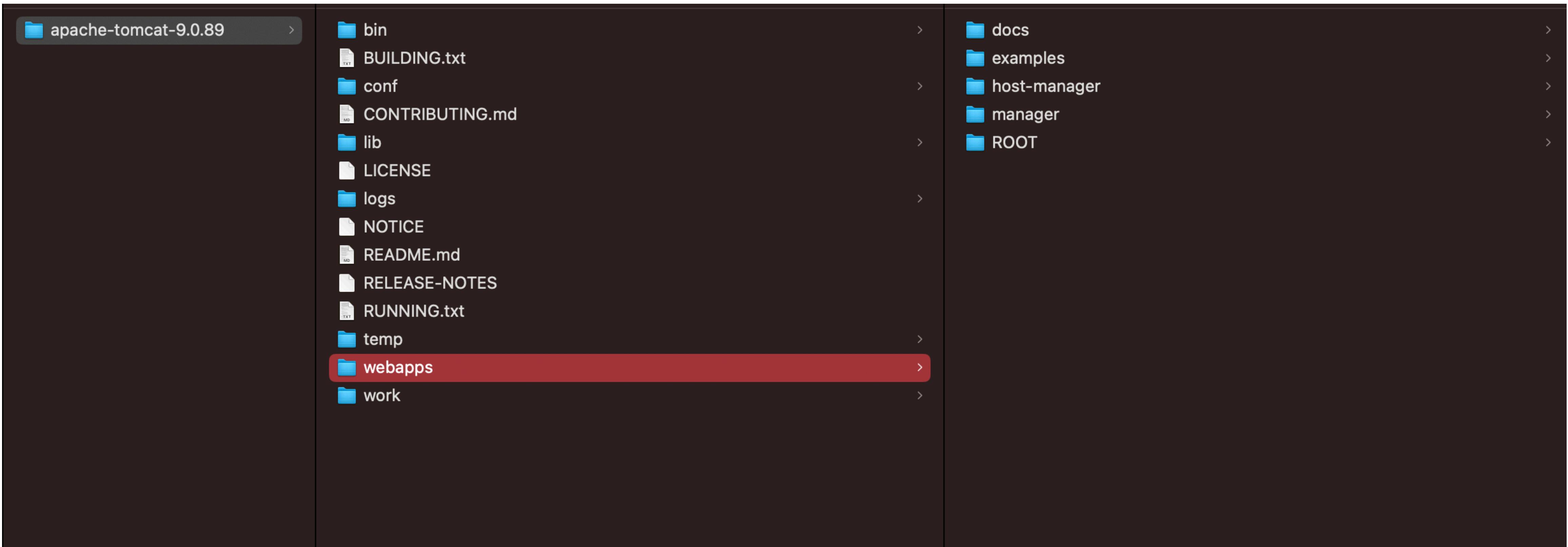
Users of Tomcat 10 onwards should be aware that, as a result of the move from Java EE to Jakarta EE as part of the transfer of Java EE to the Eclipse Foundation, the primary package for all implemented Java EE 8 and 9 APIs has changed. This means that some Java EE 8 and 9 APIs are no longer available in the standard Java API packages. Instead, they are now available in the Jakarta API packages. This change enables applications to migrate from Tomcat 9 and earlier to Tomcat 10 and later. A [migration tool](#) is available to aid this process.

The notable changes in this release are:

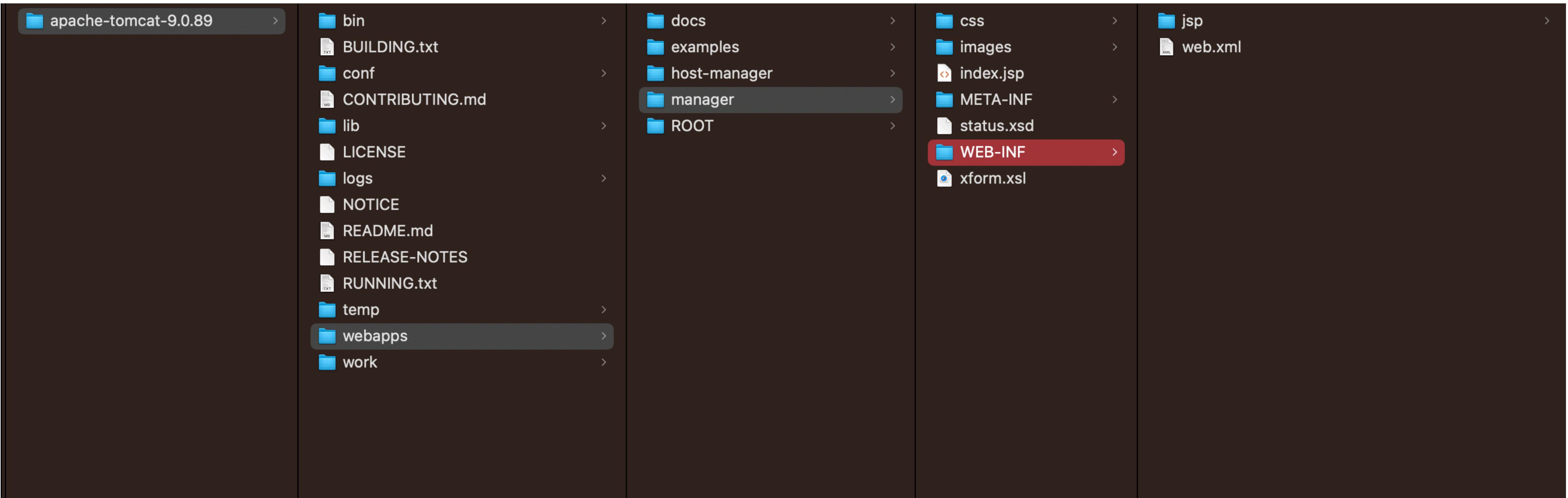
Apache Tomcat 구조



Apache Tomcat 구조



Apache Tomcat 구조

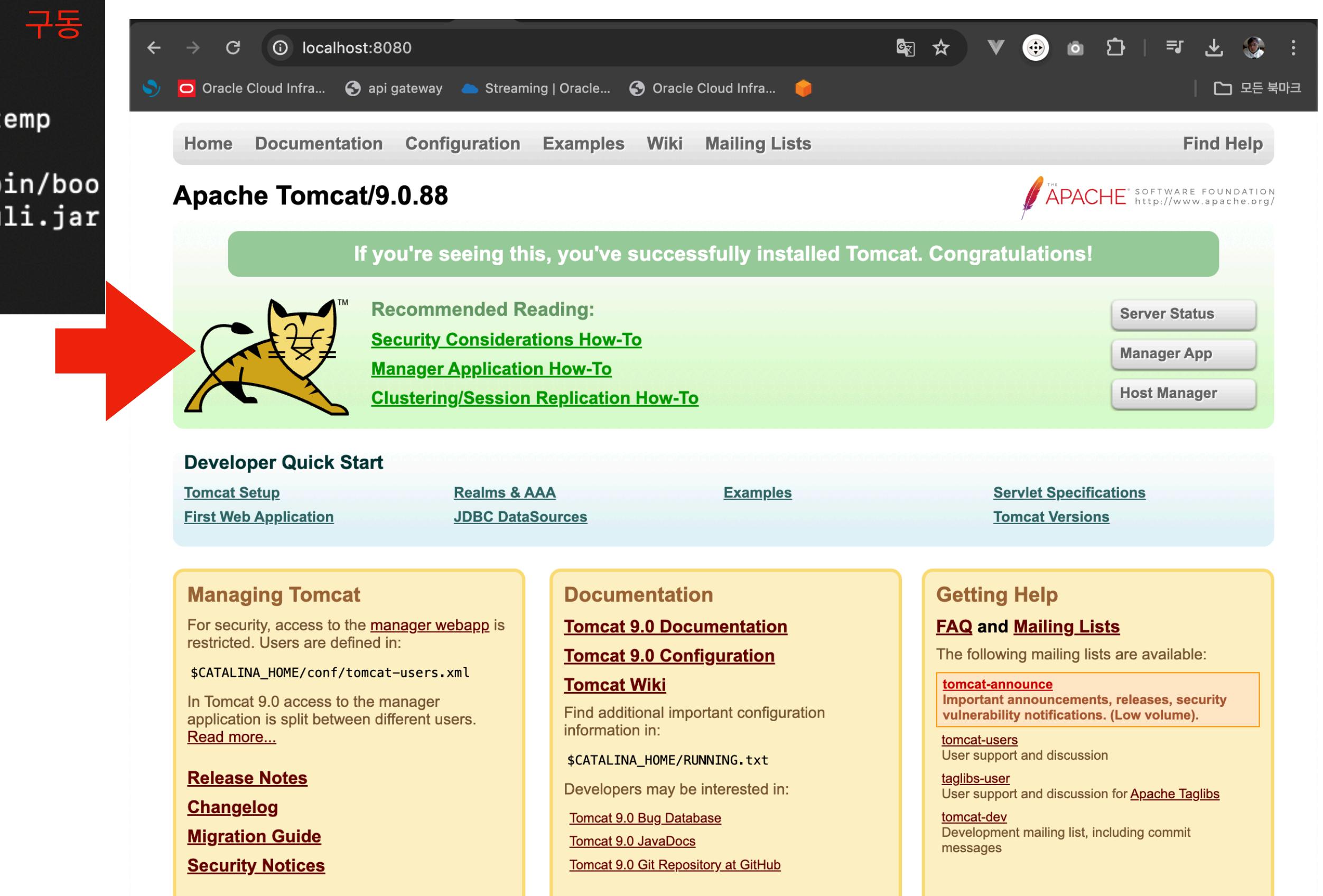


tomcat 구동 및 종료

위치: <TOMCAT_HOME>/bin

```
> ./startup.sh
Using CATALINA_BASE:  /Users/devtainer/.sdkman/candidates/tomcat/9.0.88
Using CATALINA_HOME:   /Users/devtainer/.sdkman/candidates/tomcat/9.0.88
Using CATALINA_TMPDIR: /Users/devtainer/.sdkman/candidates/tomcat/9.0.88/temp
Using JRE_HOME:        /Users/devtainer/.sdkman/candidates/java/current
Using CLASSPATH:       /Users/devtainer/.sdkman/candidates/tomcat/9.0.88/bin/bootstrap.jar:/Users/devtainer/.sdkman/candidates/tomcat/9.0.88/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

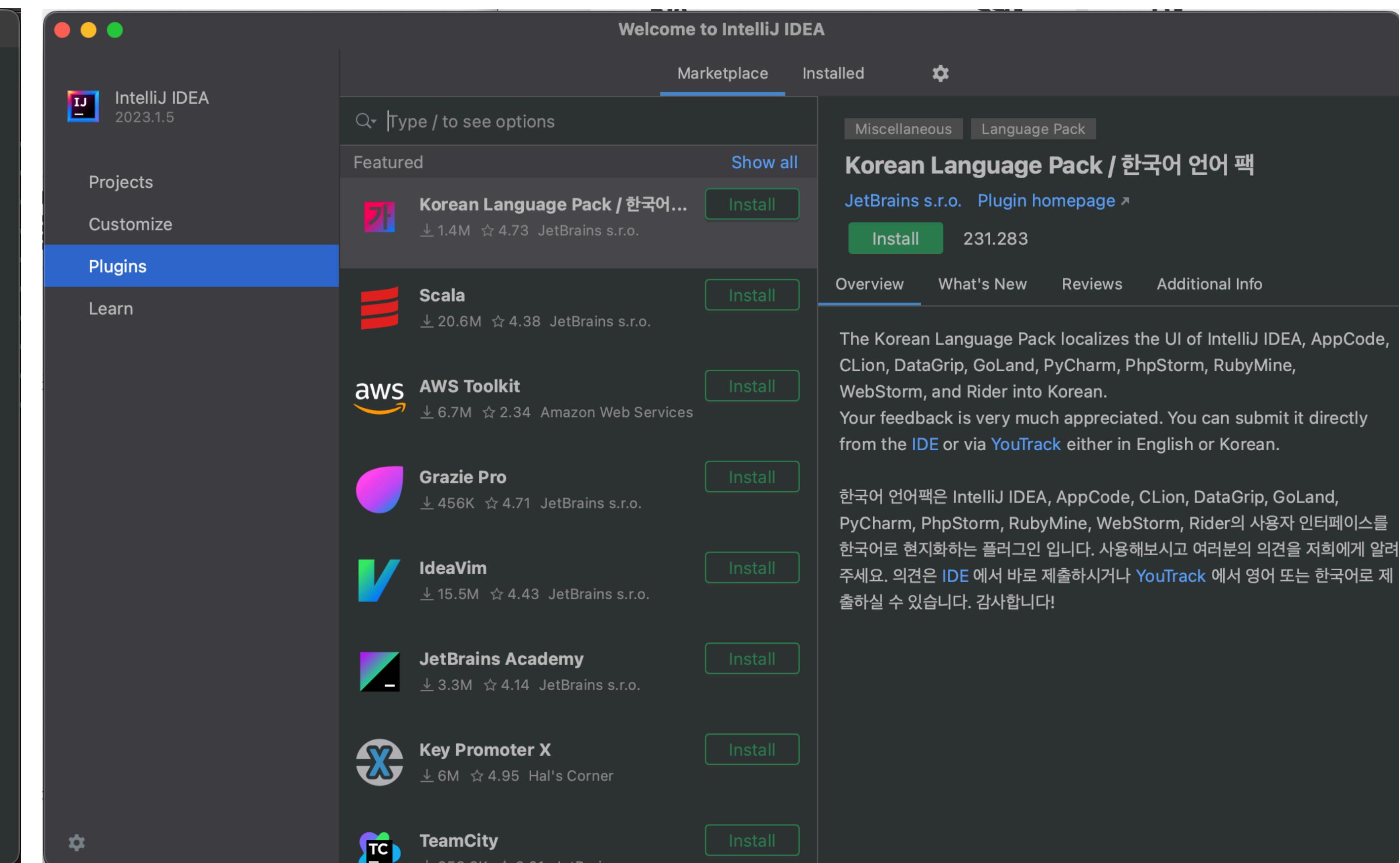
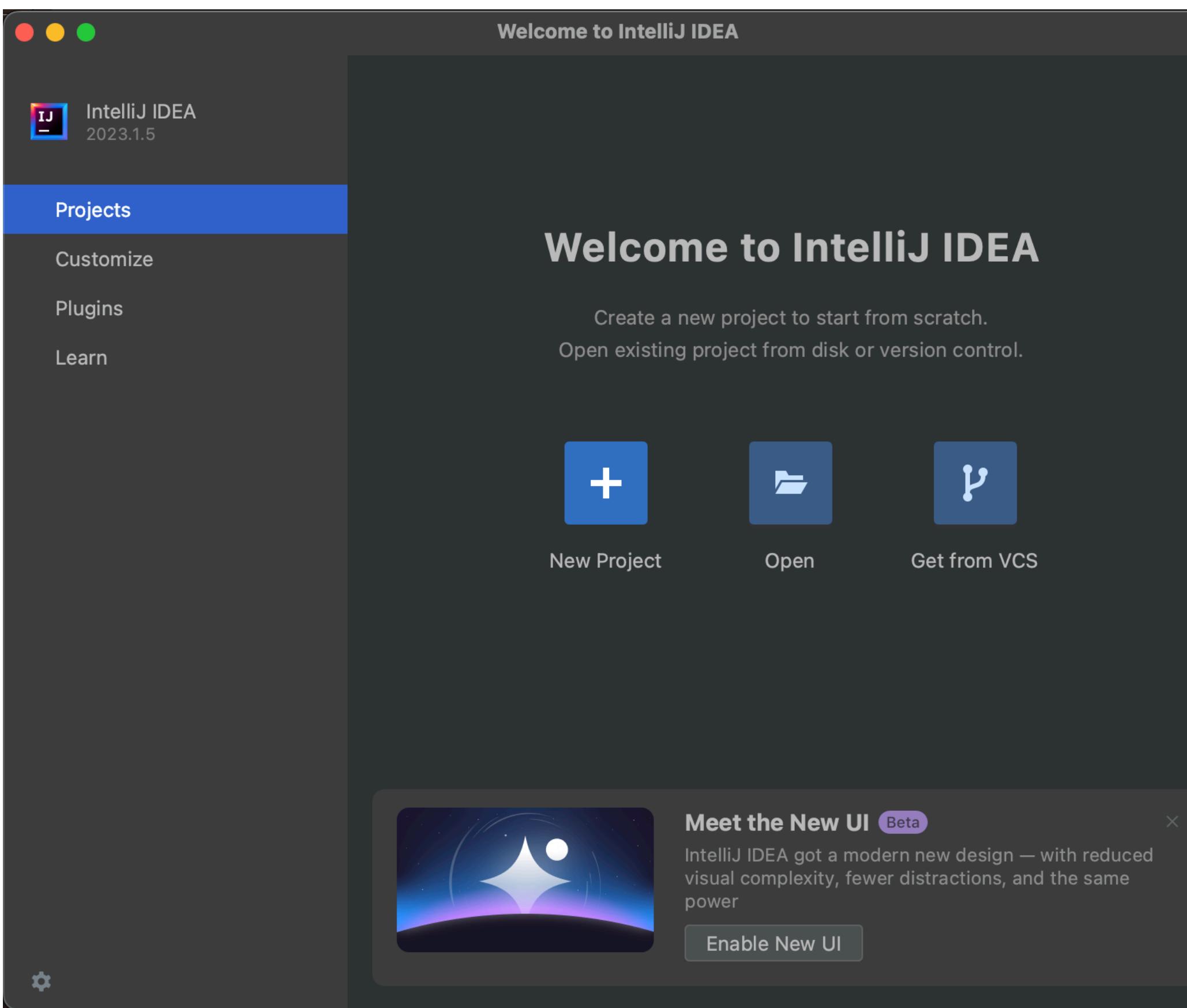
```
> ls
bootstrap.jar          makebase.sh
catalina-tasks.xml     setclasspath.bat
catalina.bat            setclasspath.sh
catalina.sh             shutdown.bat
ciphers.bat            shutdown.sh
ciphers.sh             startup.bat
commons-daemon-native.tar.gz
commons-daemon.jar      startup.sh
configtest.bat          tomcat-juli.jar
configtest.sh           tomcat-native.tar.gz
daemon.sh               tool-wrapper.bat
digest.bat              tool-wrapper.sh
digest.sh               version.bat
makebase.bat            version.sh
> shutdown.sh
Using CATALINA_BASE:  /Users/devtainer/.sdkman/candidates/tomcat/current
Using CATALINA_HOME:   /Users/devtainer/.sdkman/candidates/tomcat/current
Using CATALINA_TMPDIR: /Users/devtainer/.sdkman/candidates/tomcat/current/temp
Using JRE_HOME:        /Users/devtainer/.sdkman/candidates/java/current
Using CLASSPATH:       /Users/devtainer/.sdkman/candidates/tomcat/current/bin/bootstrap.jar:/Users/devtainer/.sdkman/candidates/tomcat/current/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --
--add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --
--add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/
/sun.rmi.transport=ALL-UNNAMED
```



IntelliJ IDEA 설정



IntelliJ IDEA 실행



Eclipse vs IntelliJ

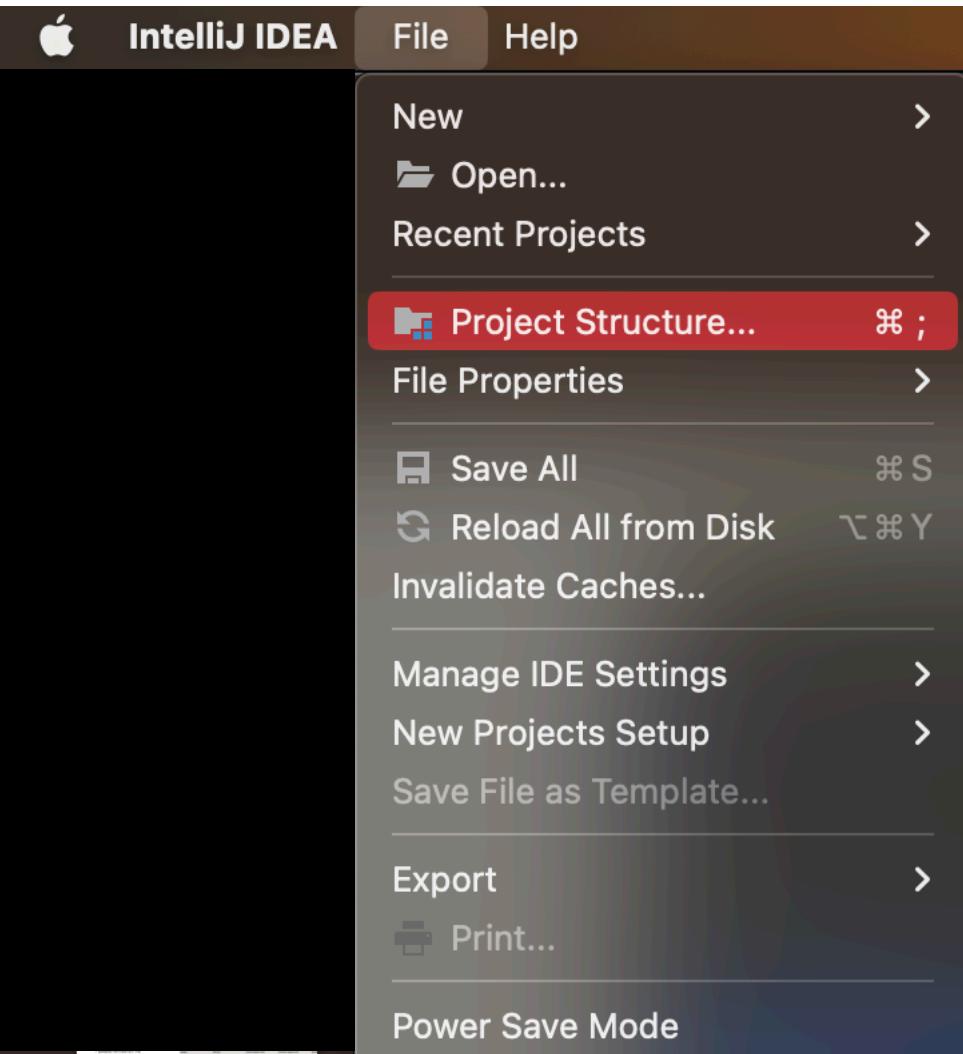
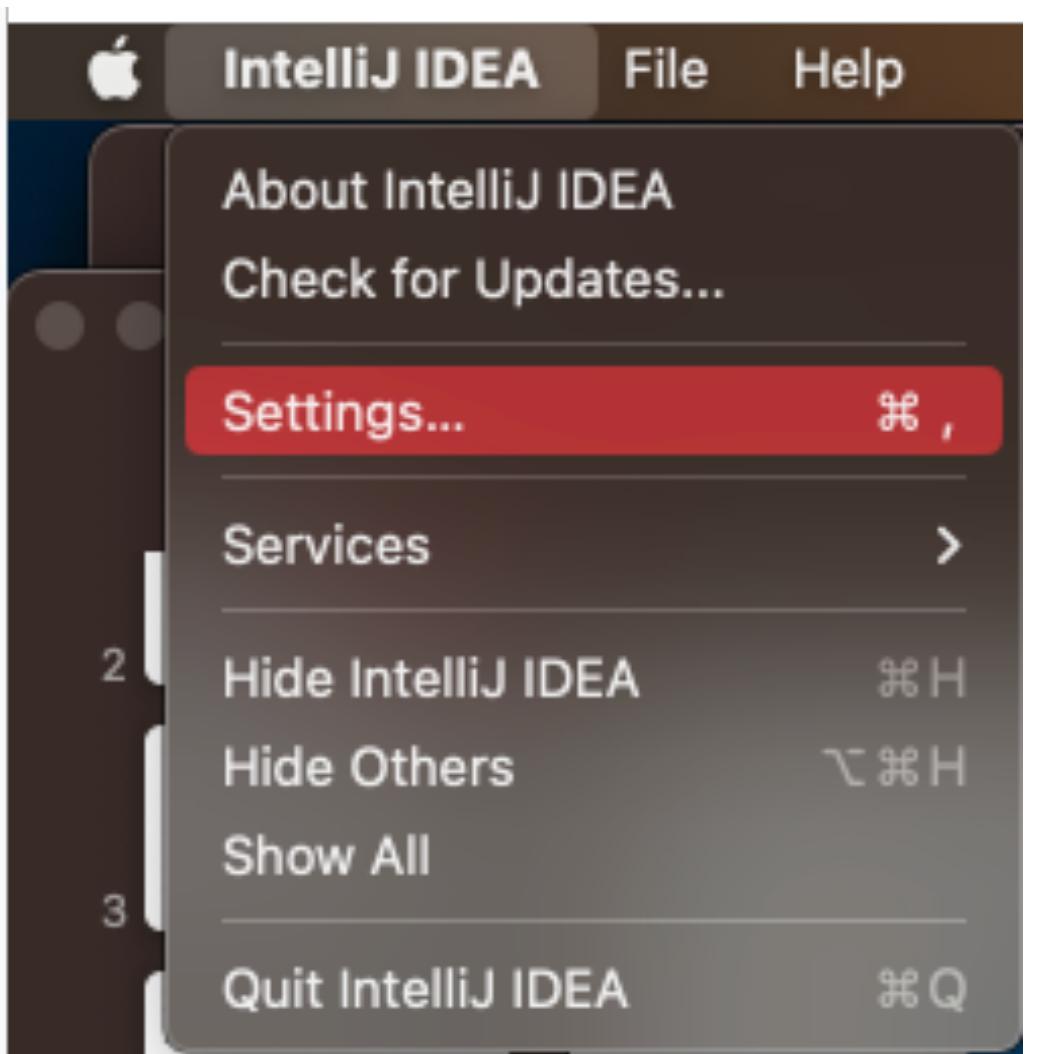
Eclipse	IntelliJ IDEA
Workspace	Project
Project	Module
Facet	Facet
Library	Library
JRE	SDK
Classpath variable	Path variable

- No workspace
- No perspectives
- Tool windows
- Multiple windows

	Eclipse IDE	IntelliJ IDEA
Overall Rating	 ★★★★☆ 4.36 (148)	 ★★★★★ 4.73 (1139)
Ease-of-Use	4 / 5	4.5 / 5
Value for Money	4.5 / 5	4.5 / 5
Customer Support	4 / 5	4.5 / 5
Functionality	4.5 / 5	4.5 / 5
Last Review Written	May 15, 2023	May 16, 2023
Pros/Cons	<p>Pros</p> <ul style="list-style-type: none"> ▲ I can say everything is best in this tool. Right from GUI, easy of use, Integration/collaborations with other tools. ▲ Development with Eclipse is a hiccup free experience and has pretty reliable integration with GIT and contains lots of plugins to extend its features. <p>Cons</p> <ul style="list-style-type: none"> ▼ It's starting more and more to feel outdated, and there's no decent dark mode that does everything. Upgrading to new versions was always a hassle. ▼ Setting up of eclipse is a tad tiresome. If you are not used to use this IDE, it will be hard to understand the project explorer. 	<p>Pros</p> <ul style="list-style-type: none"> ▲ As a Java developer this IDE is my all-time favorite. I highly suggest the paid version which adds many cool features, but even the community edition is so powerful and pleasing to the eye. ▲ It integrates perfectly with Android and supports Kotlin out of the box. <p>Cons</p> <ul style="list-style-type: none"> ▼ I dislike importing from the git. It is very hard to find space as all your projects can easily get mixed into one another. ▼ Periodically need to reinstall it due to failure of plugins. The software just fails to start and then need to remove plugin directory, install everything again etc.
	Read all user reviews	Read all user reviews

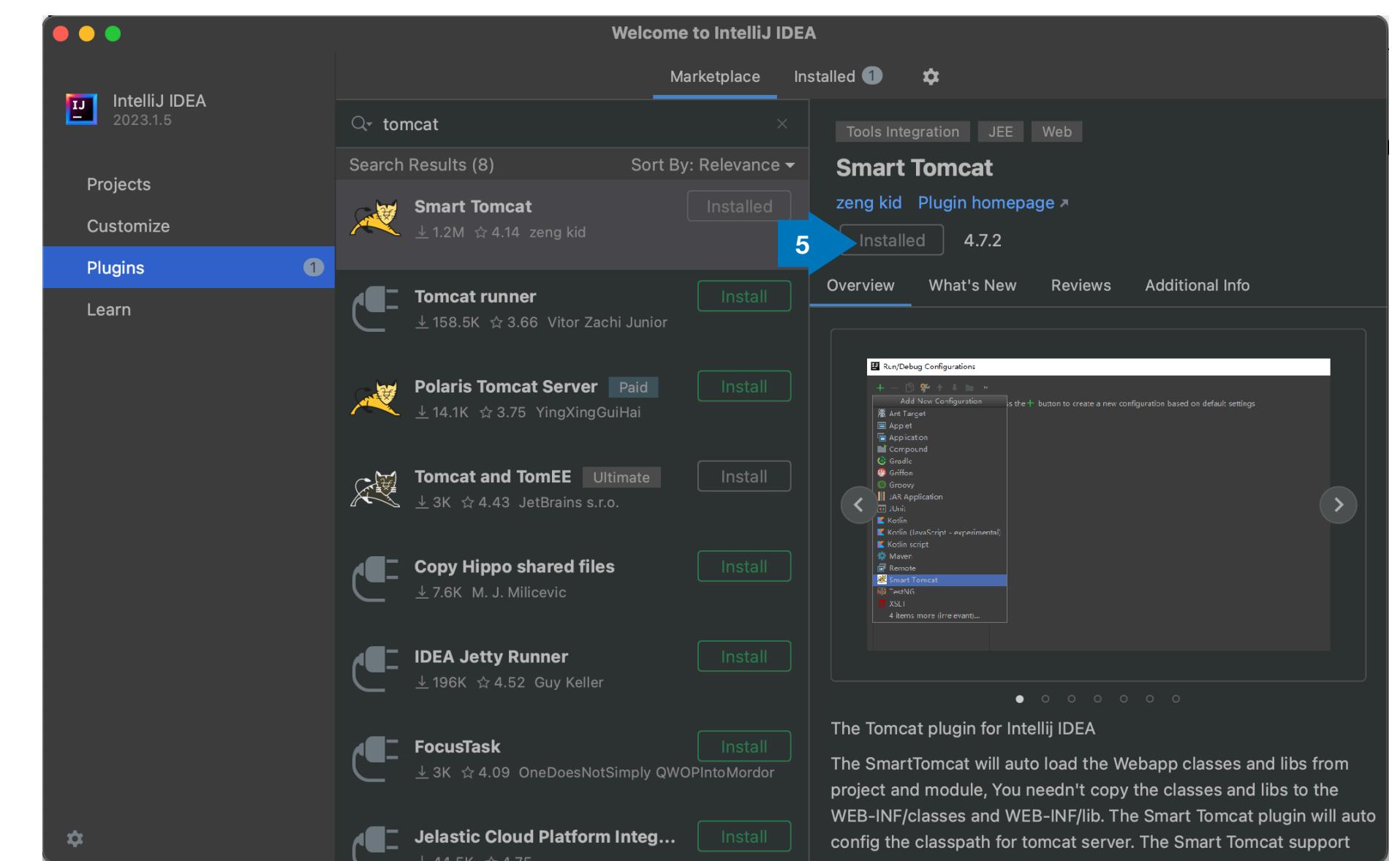
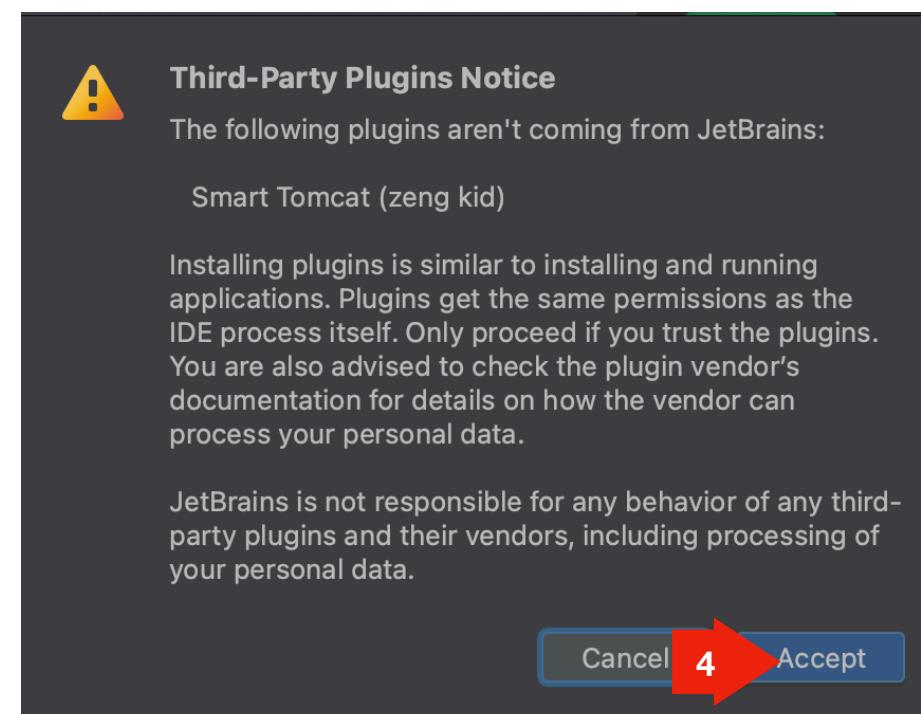
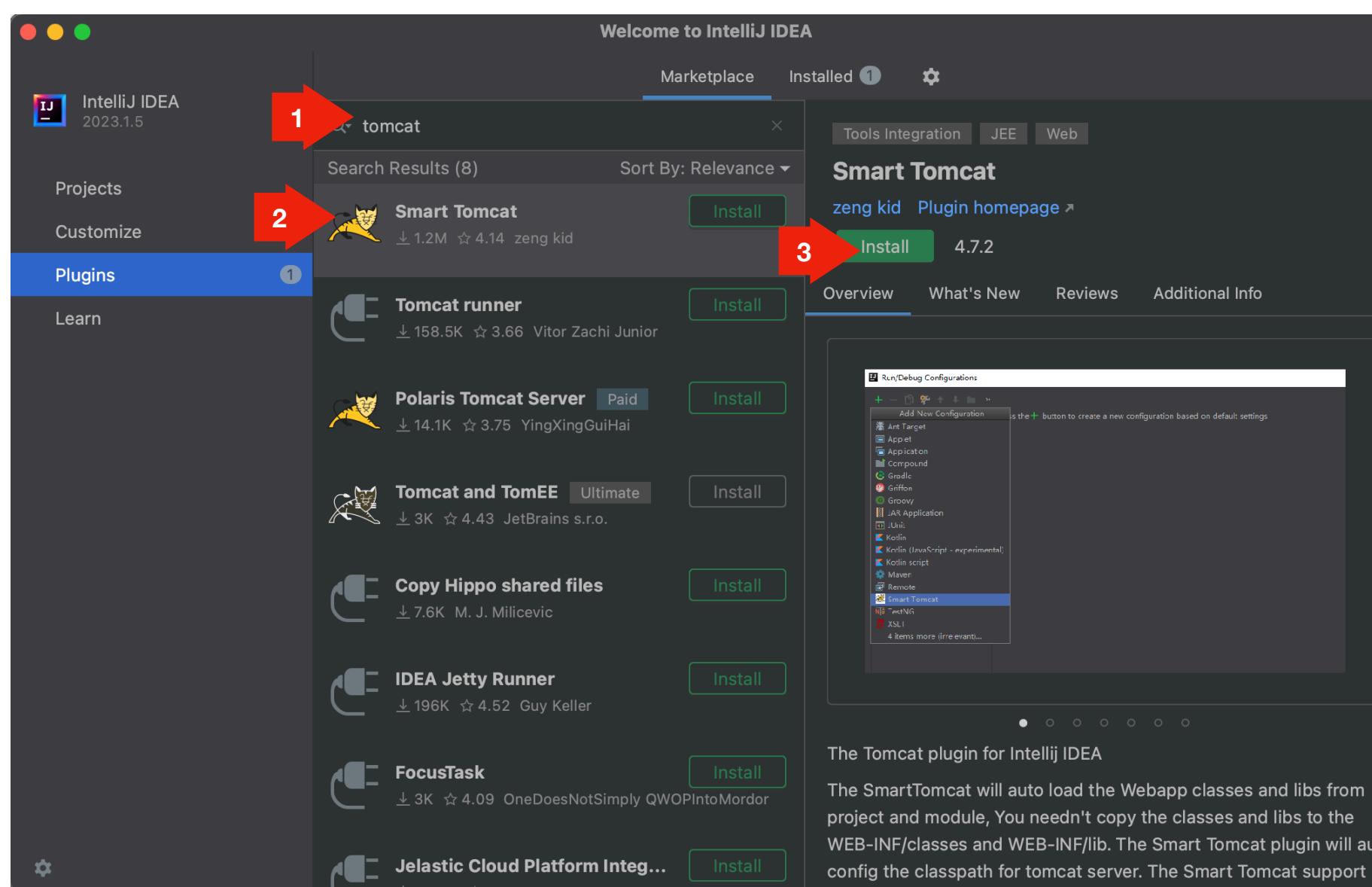
환경설정

- Settings: 글로벌 설정
- Project Structure: 프로젝트 단위 설정



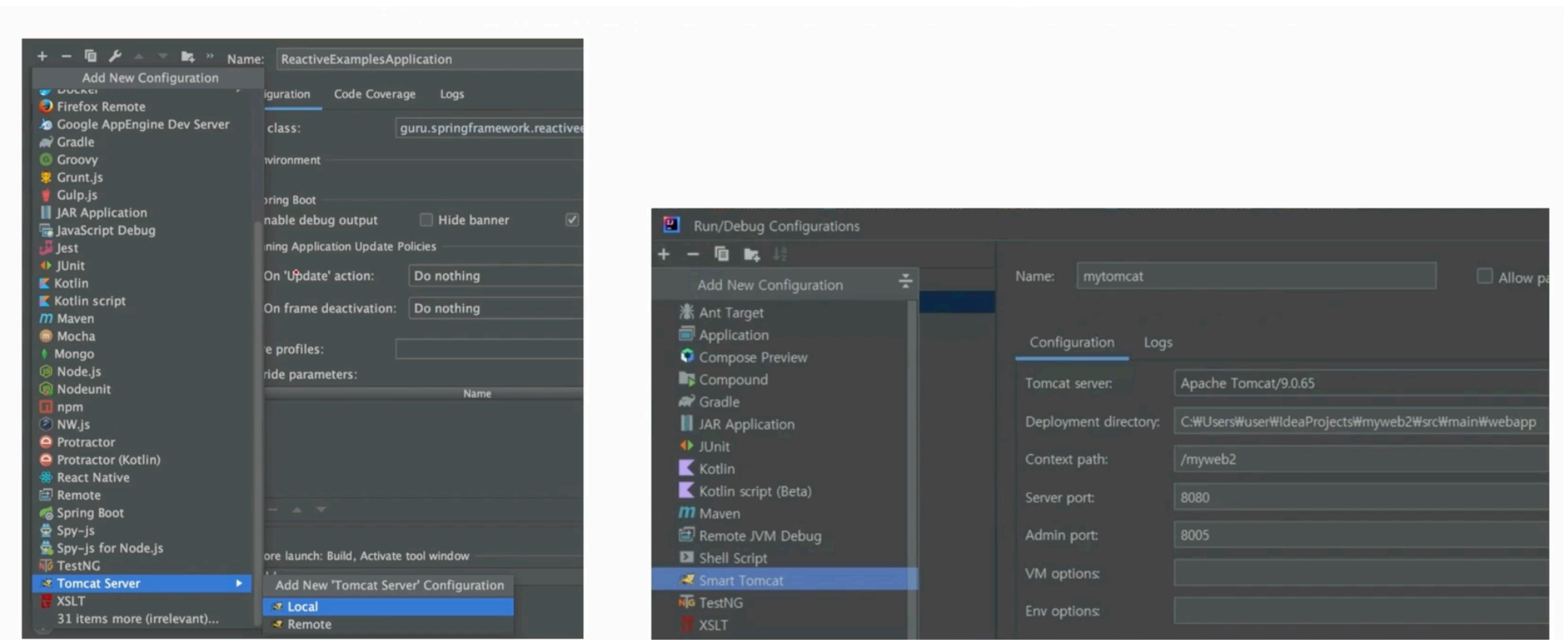
Tomcat Plugin 설치

Smart Tomcat Plugin



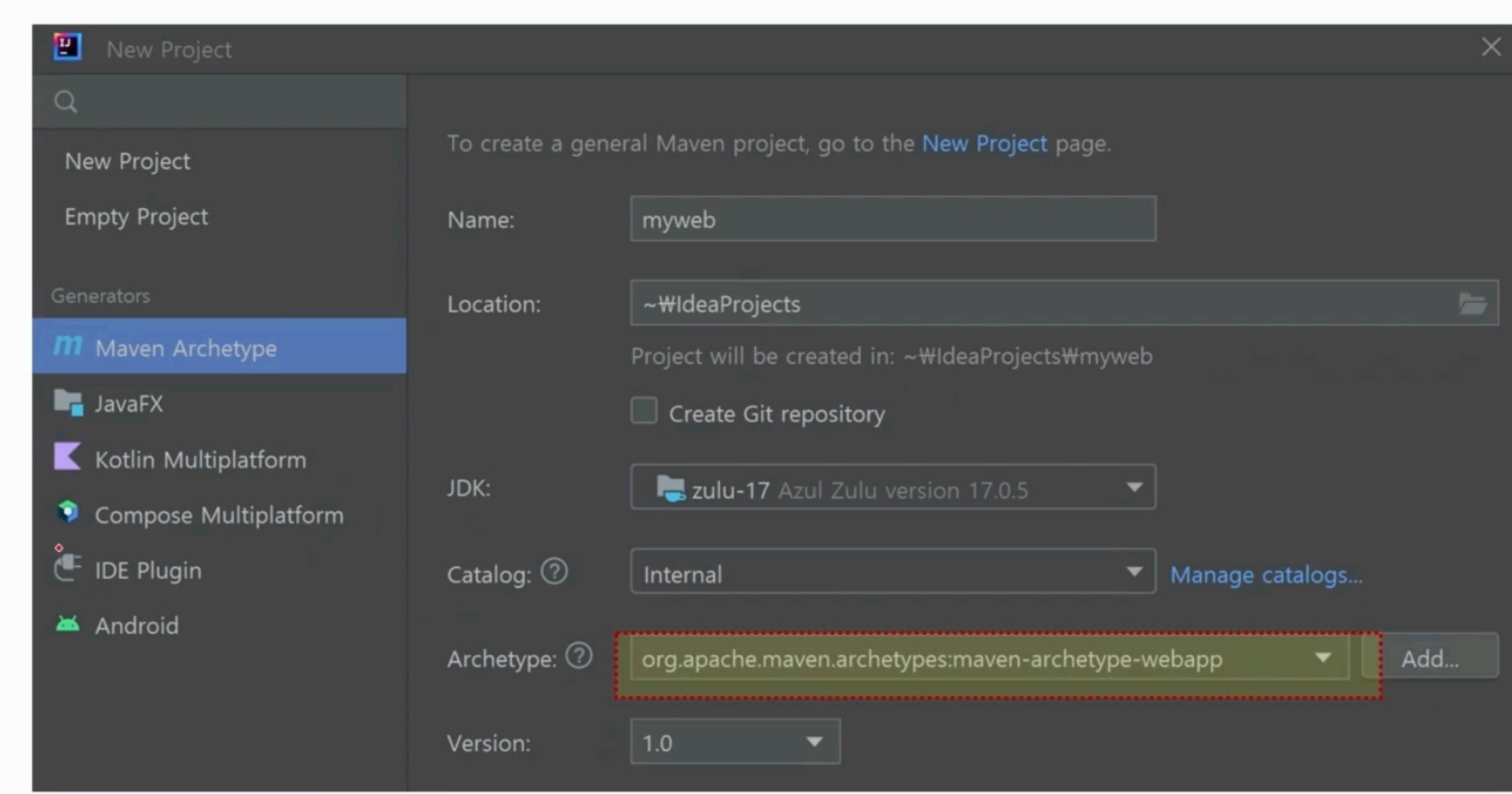
Tomcat 서버 설정

- Run -> Edit Configurations

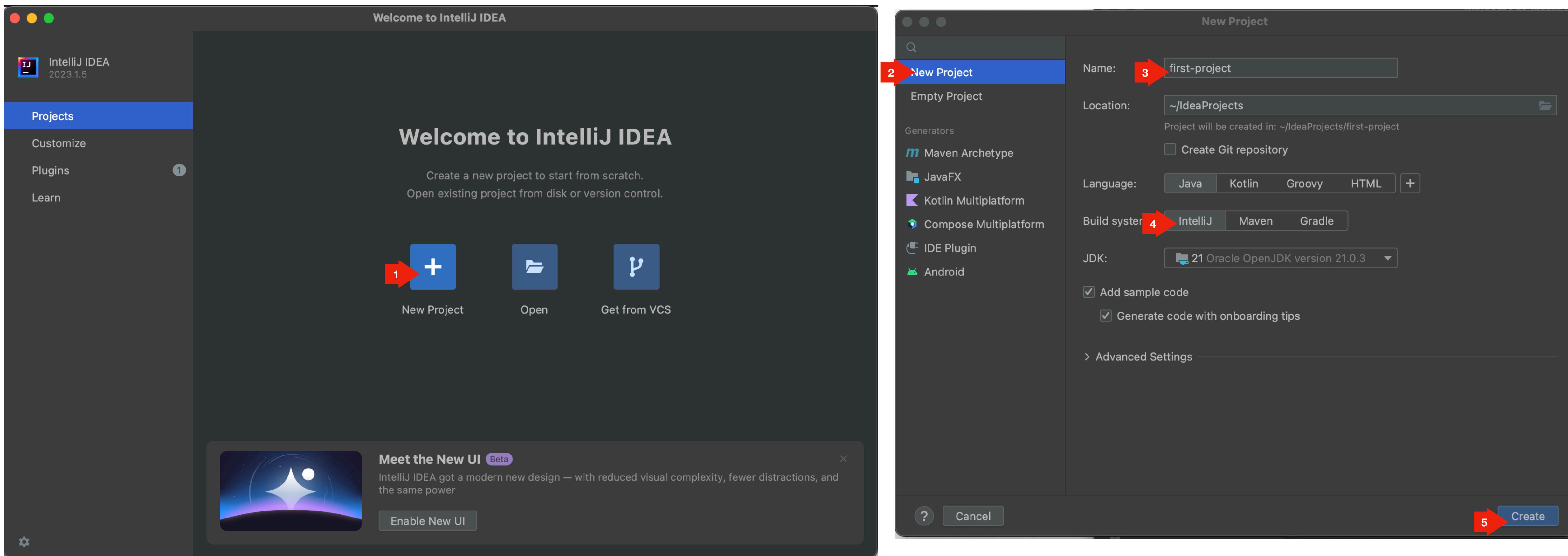


Maven 설정

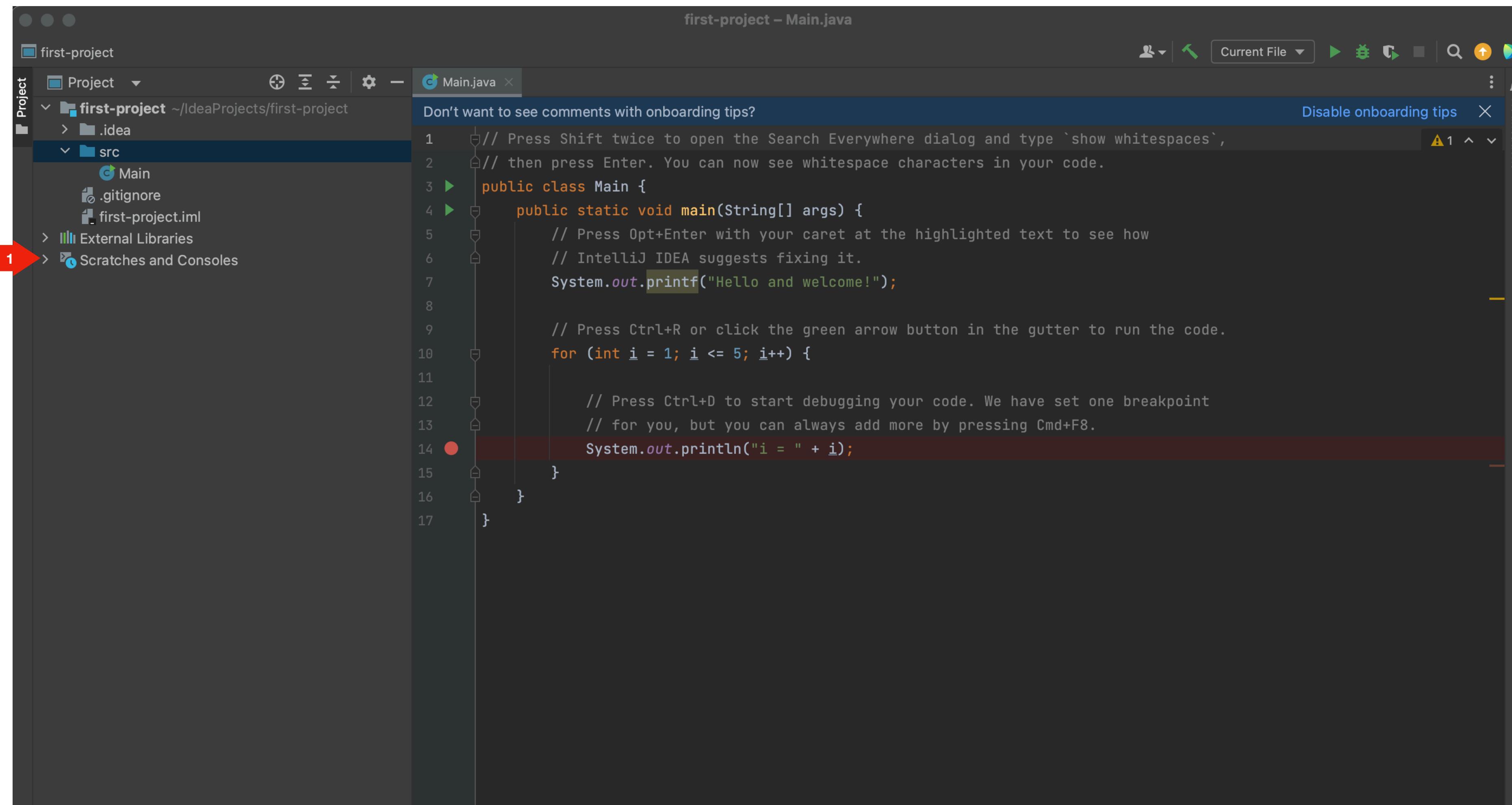
- Maven Archetype
 - org.apache.maven.archetypes:maven-archetype-webapp



Java Project 생성

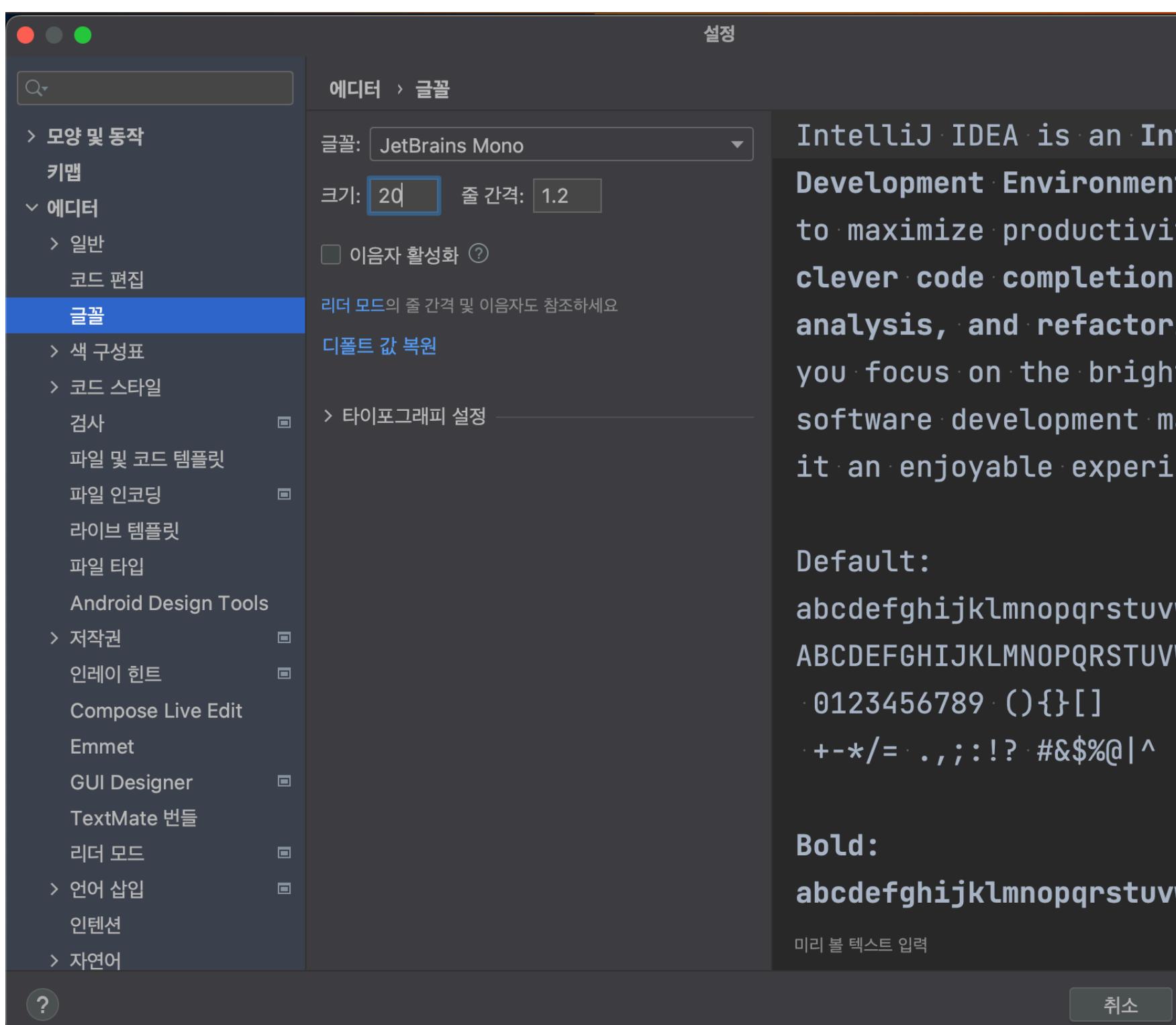
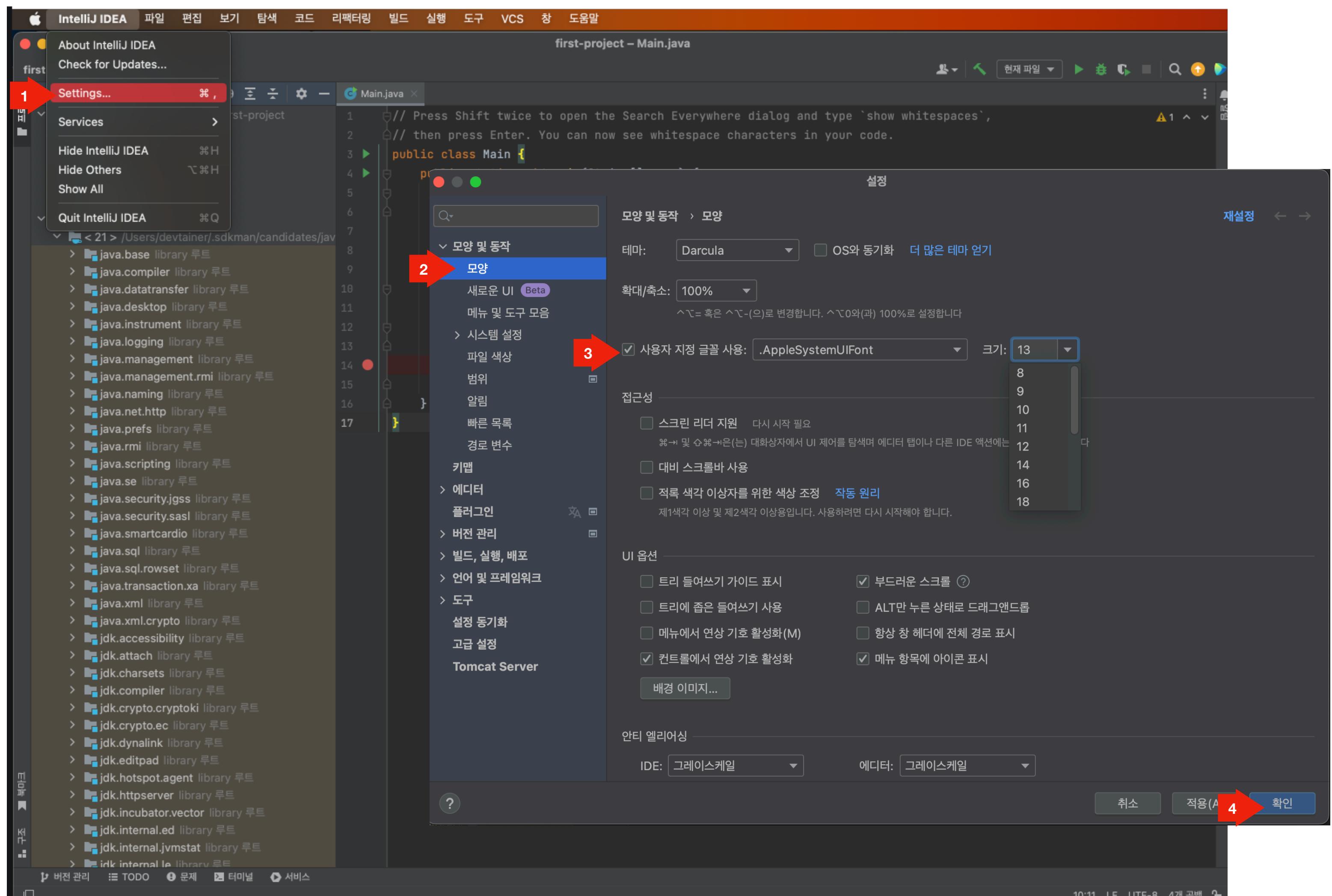


Java Project 생성



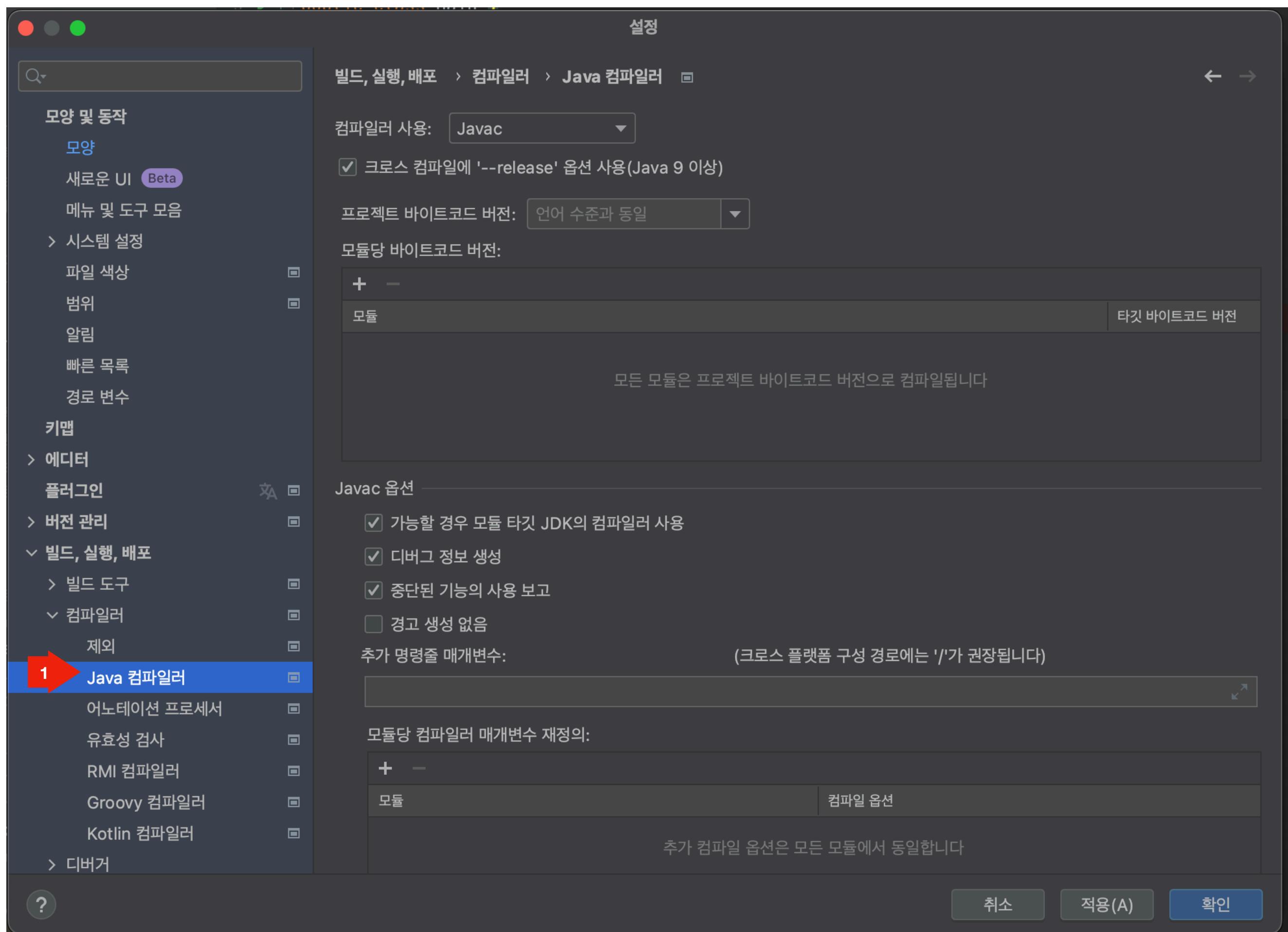
Java Project 생성

글로벌 설정 변경 -> 폰트 크기



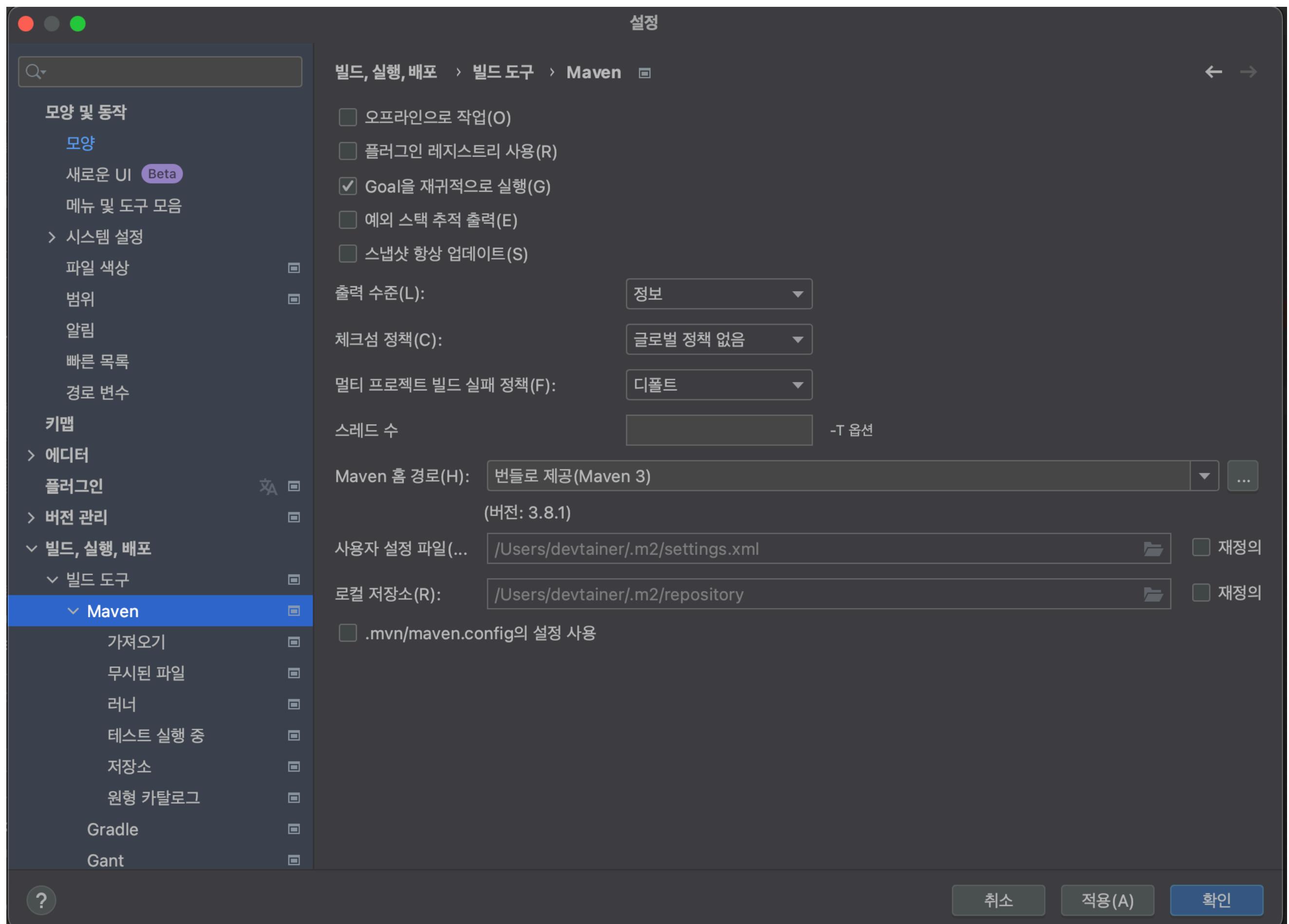
Java Project 생성

컴파일러 변경



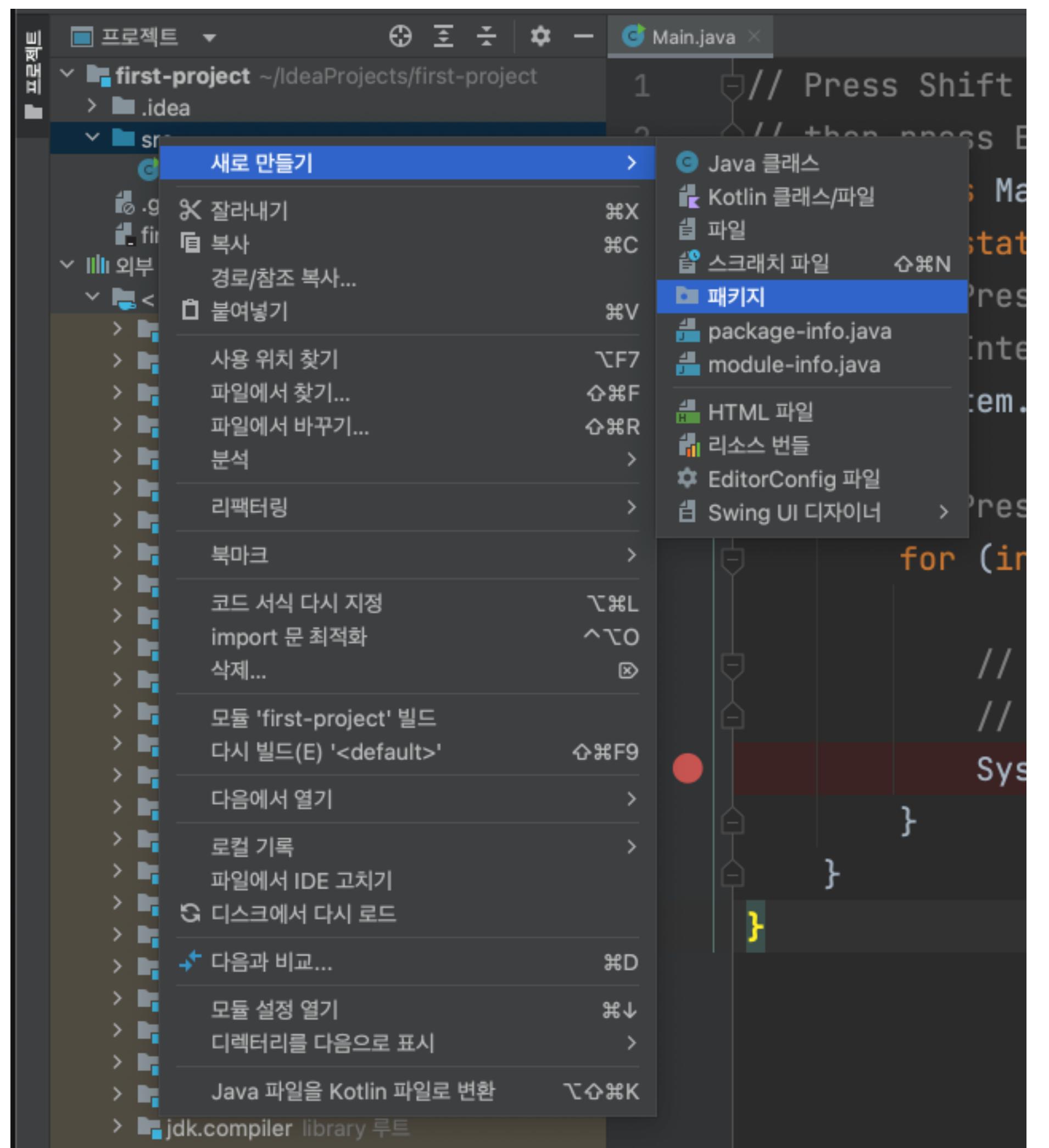
Java Project 생성

빌드 툴 변경

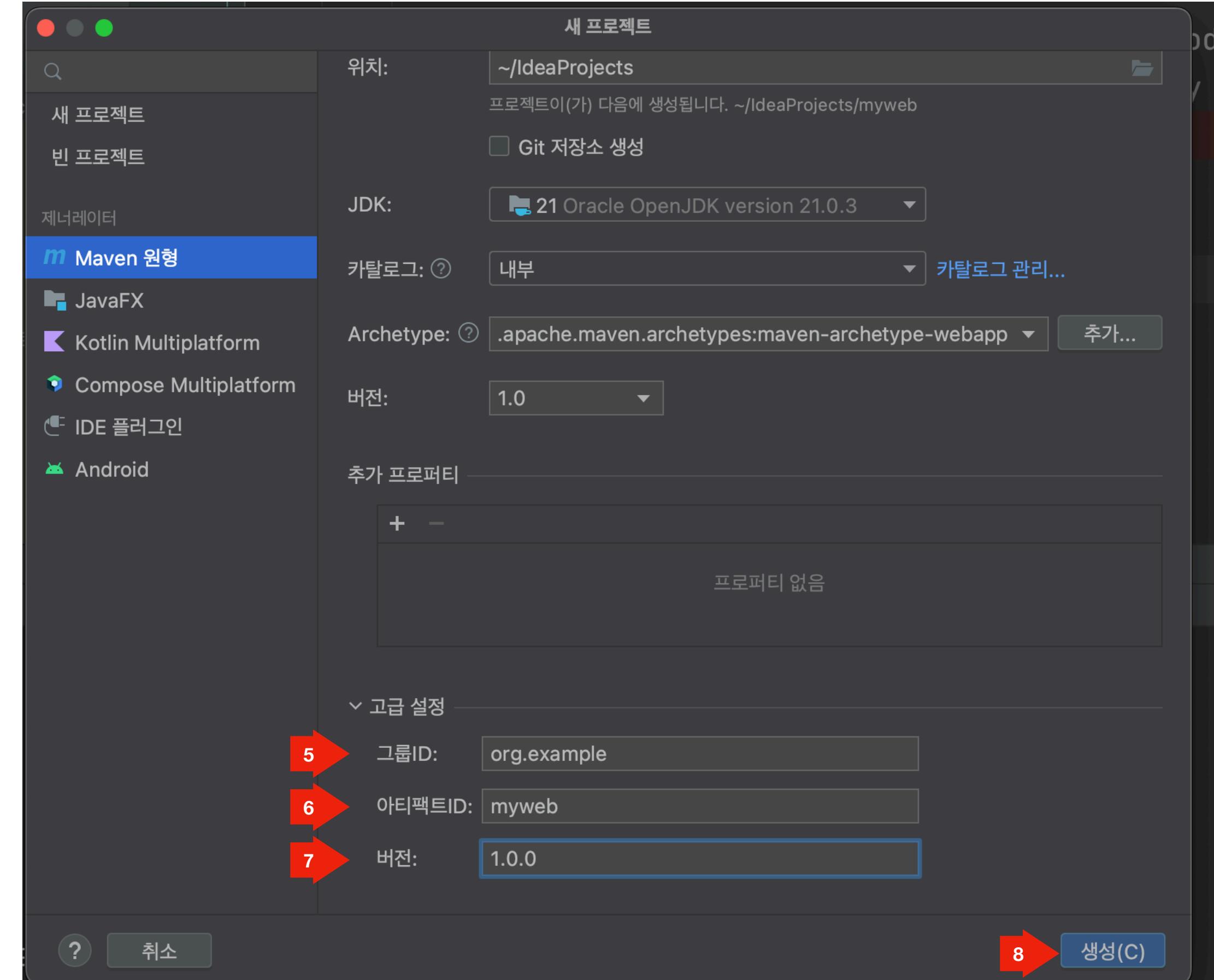
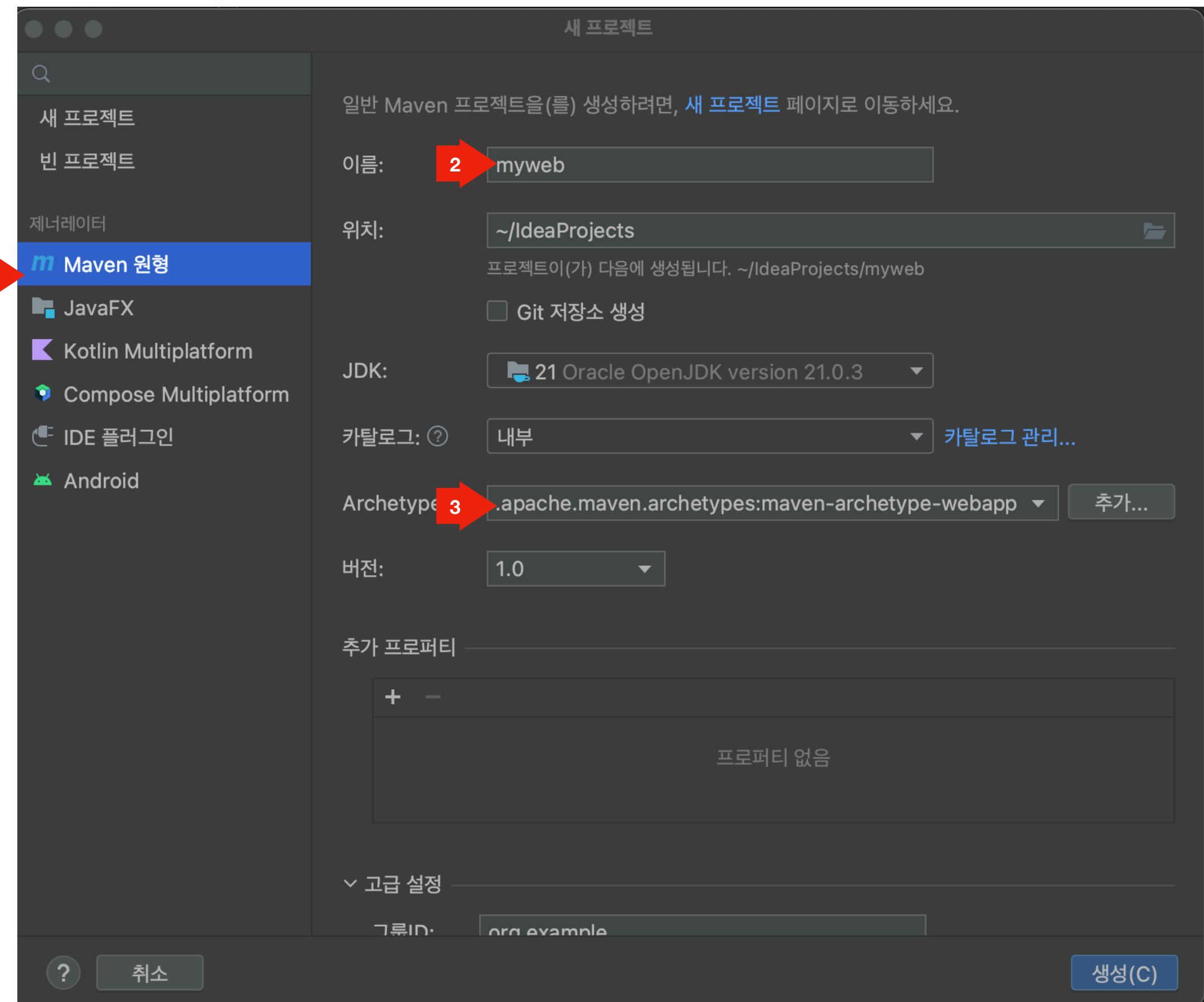


Java Project 생성

클래스 생성



Maven Project 생성



Maven Project 생성

The screenshot shows the IntelliJ IDEA IDE interface with a Maven project named "myweb".

Project Structure:

- myweb (Project)
- .idea
- src
 - main
 - resources
 - webapp
 - WEB-INF
 - web.xml
 - index.jsp
 - .gitignore
 - pom.xml

pom.xml Content:

```
<modelVersion>4.0.0</modelVersion>
<groupId>org.example</groupId>
<artifactId>myweb</artifactId>
<packaging>war</packaging>
<version>1.0.0</version>
<name>myweb Maven Webapp</name>
<url>http://maven.apache.org</url>
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Console Output:

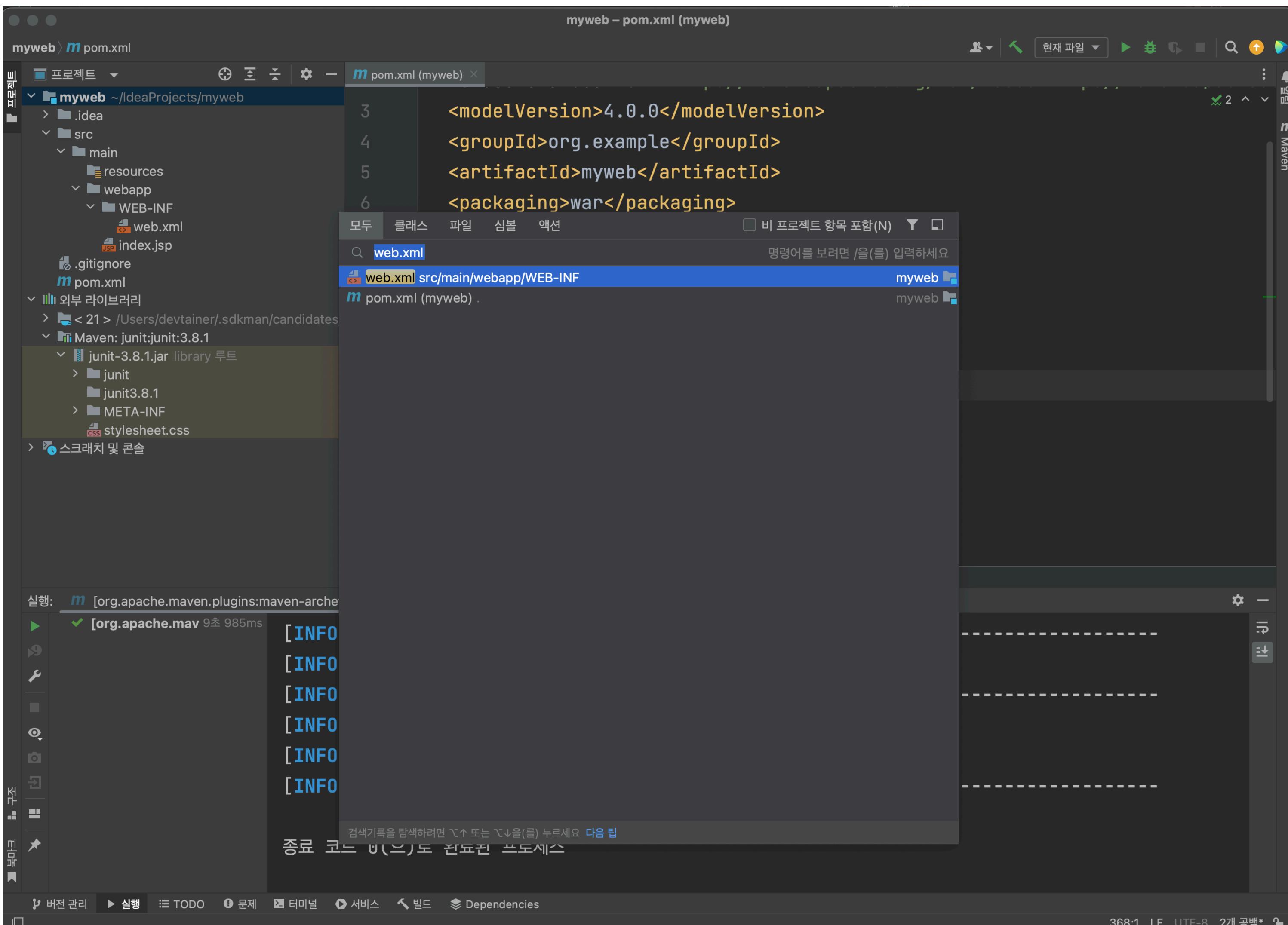
```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  9.071 s
[INFO] Finished at: 2024-06-12T04:31:04+09:00
[INFO] -----
```

종료 코드 0(으)로 완료된 프로세스

Bottom navigation bar: 버전 관리, 실행, TODO, 문제, 터미널, 서비스, 빌드, Dependencies

Maven Project 생성

파일 검색: shift key 2번 입력



Maven Project 생성

pom.xml 리뷰

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

- Project Structure:** The left sidebar shows the project structure for "myweb". It includes a .idea folder, a src directory containing main and resources, a webapp directory with WEB-INF, web.xml, and index.jsp, a .gitignore file, and a pom.xml file which is currently selected.
- pom.xml Content:** The main editor pane displays the XML configuration for the Maven project. The code is as follows:

```
<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 http://maven.apache.org/xsd/pom-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>myweb</artifactId>
    <packaging>war</packaging>
    <version>1.0.0</version>
    <name>myweb Maven Webapp</name>
    <url>http://maven.apache.org</url>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <finalName>myweb</finalName>
    </build>
</project>
```

The "Dependencies" section contains one dependency for the junit artifact. The "build" section specifies a final name for the war file.

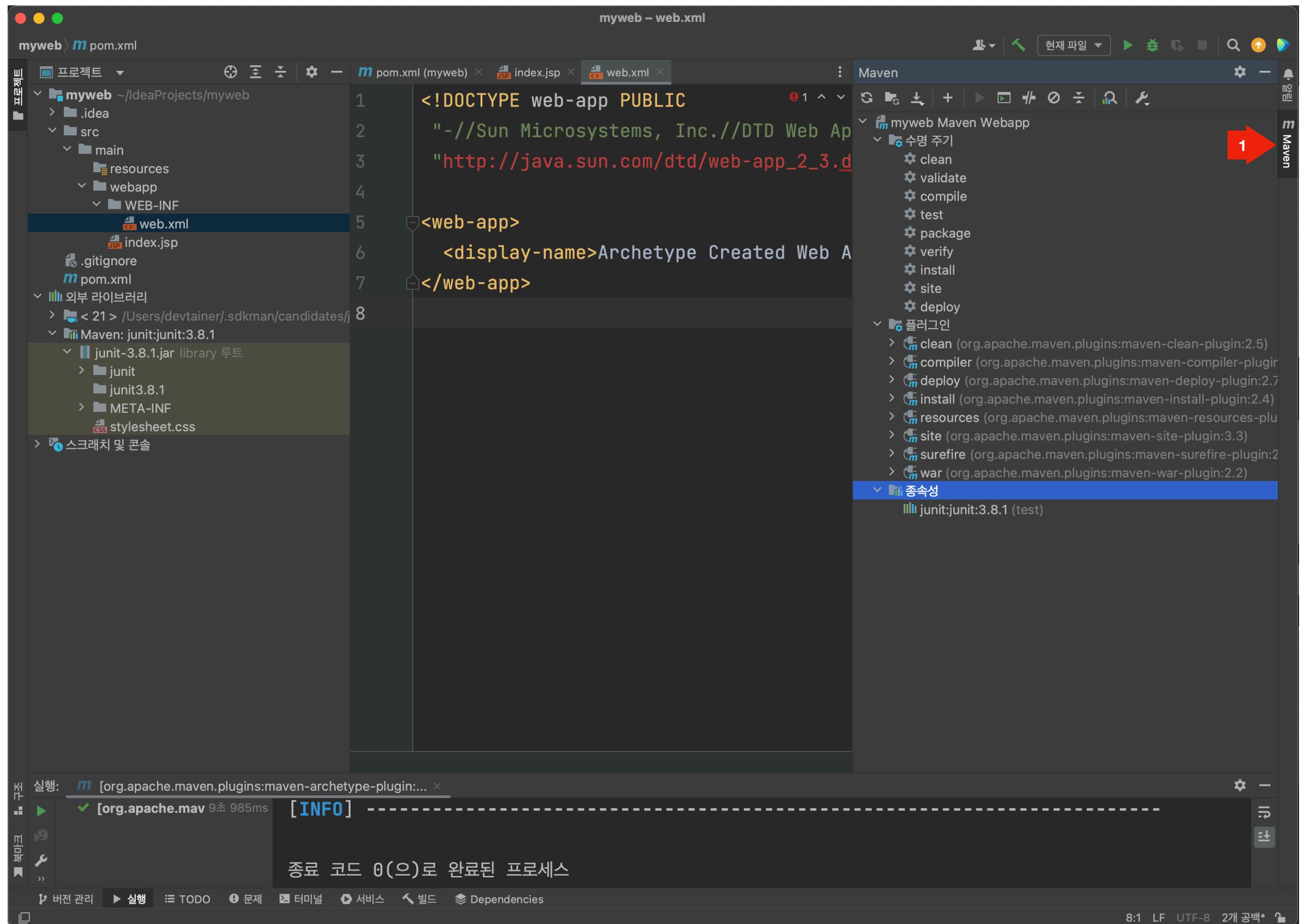
Run Tab: The bottom-left tab bar shows the "Run" tab is active, with the command "org.apache.maven.plugins:maven-archetype-plugin:..." selected.

Output Tab: The bottom-right tab bar shows the "Output" tab is active, displaying the Maven build log:

```
[INFO] [org.apache.maven] 9초 985ms [INFO] -
```

Status Bar: The bottom status bar shows the current time as 12:11, the encoding as UTF-8, and two open files.

Maven Project 빌드



Maven Project 빌드

터미널 빌드

- 터미널 위치: project root
- command: mvn clean compile



The screenshot shows a terminal window within an IDE interface. The terminal tab is active, and the command `mvn clean compile` is being run. The output shows the Maven process scanning for projects, building a specific web application named `myweb`, and performing a clean operation by deleting the target directory.

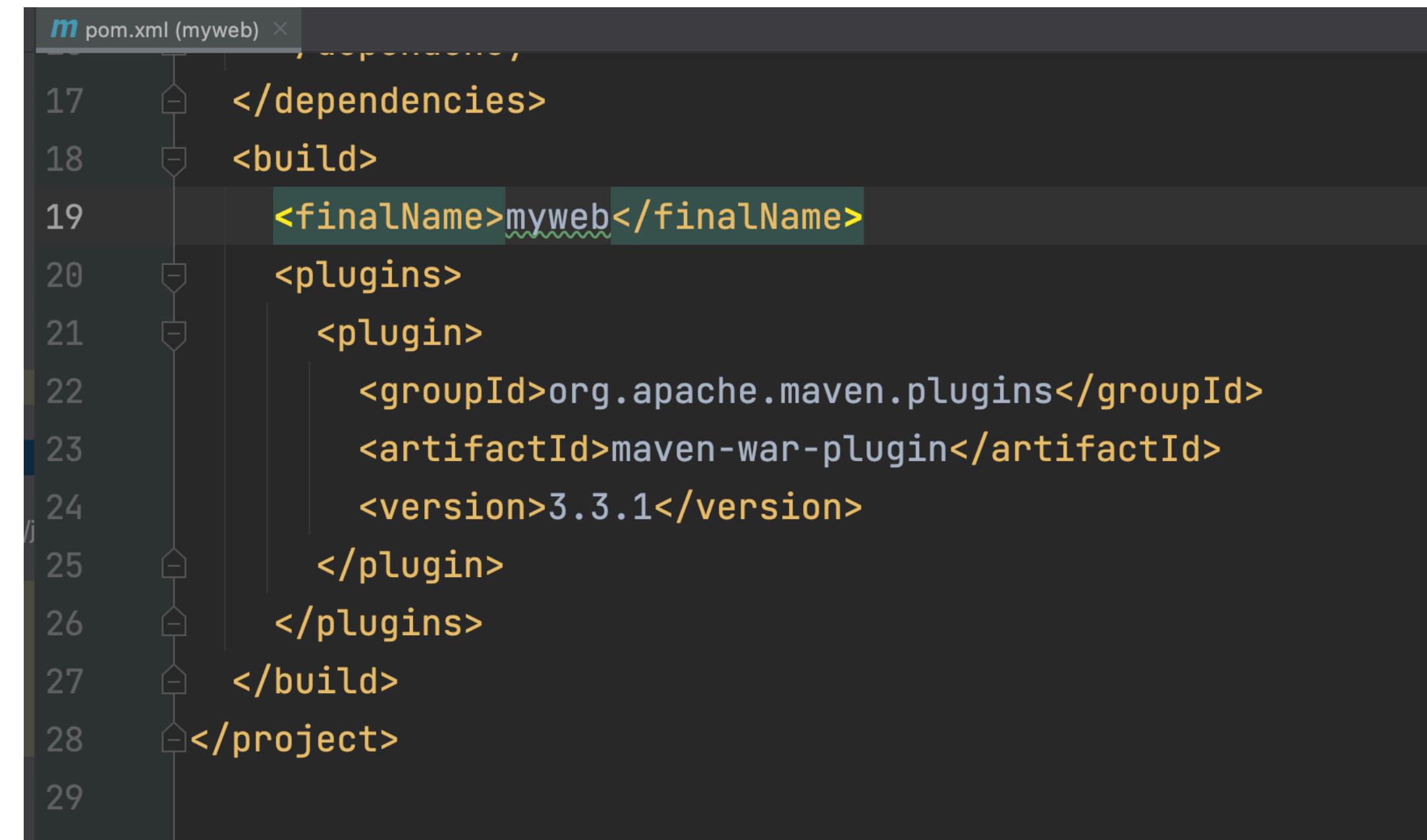
```
터미널: 로컬 × + ▾
mvn clean compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.example:myweb >-----
[INFO] Building myweb Maven Webapp 1.0.0
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ myweb ---
[INFO] Deleting /Users/devtainer/IdeaProjects/myweb/target
```

At the bottom of the terminal window, there are several tabs: 버전 관리, 실행, TODO, 문제, 터미널, 서비스, 빌드, and Dependencies. The 터미널 tab is currently selected. The status bar at the bottom right indicates LF, UTF-8, 2개 공백*, and a small icon.

Maven Project 빌드

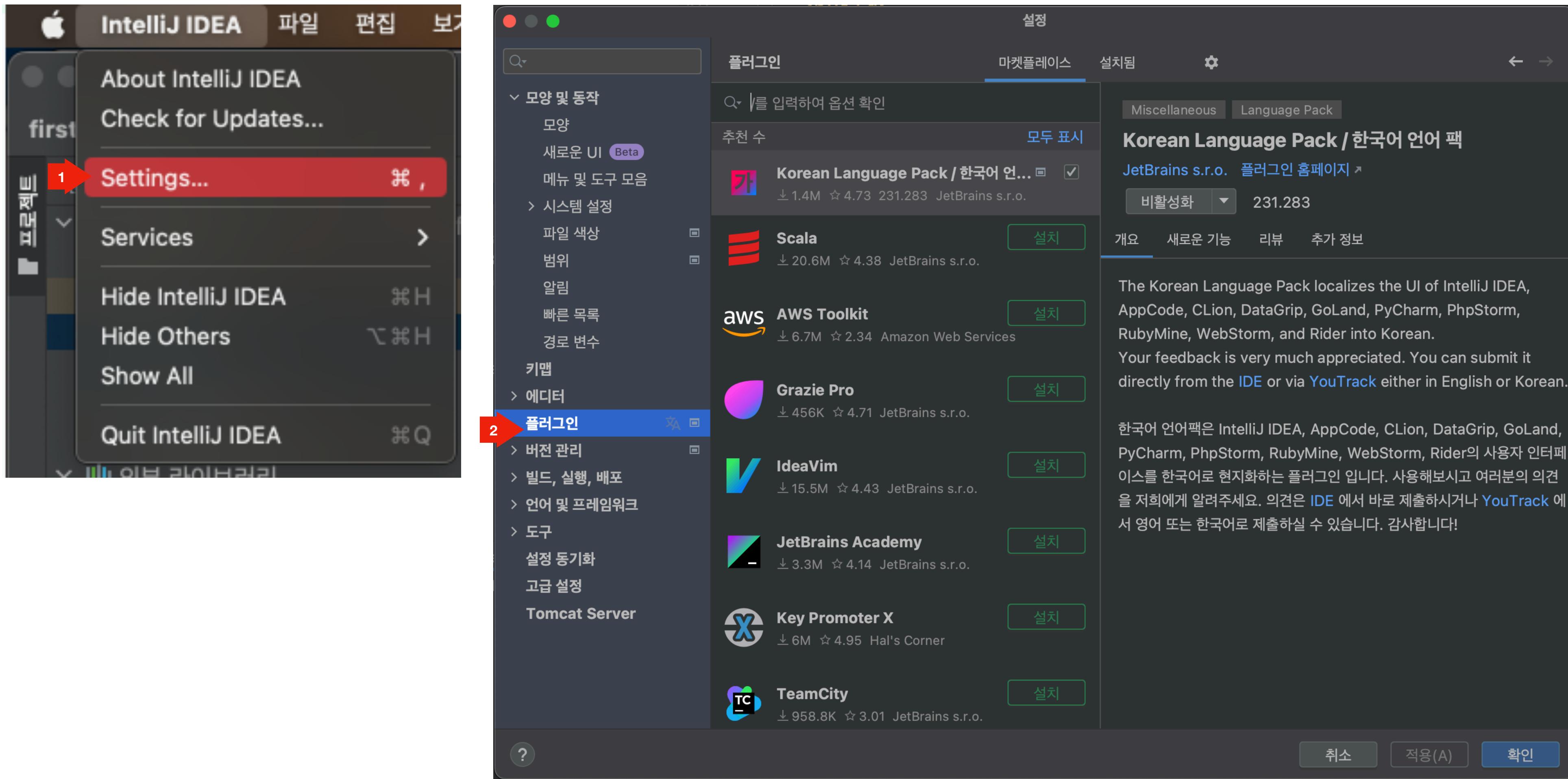
패키지 오류시 다음 코드 추가

```
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.3.1</version>
  </plugin>
</plugins>
```



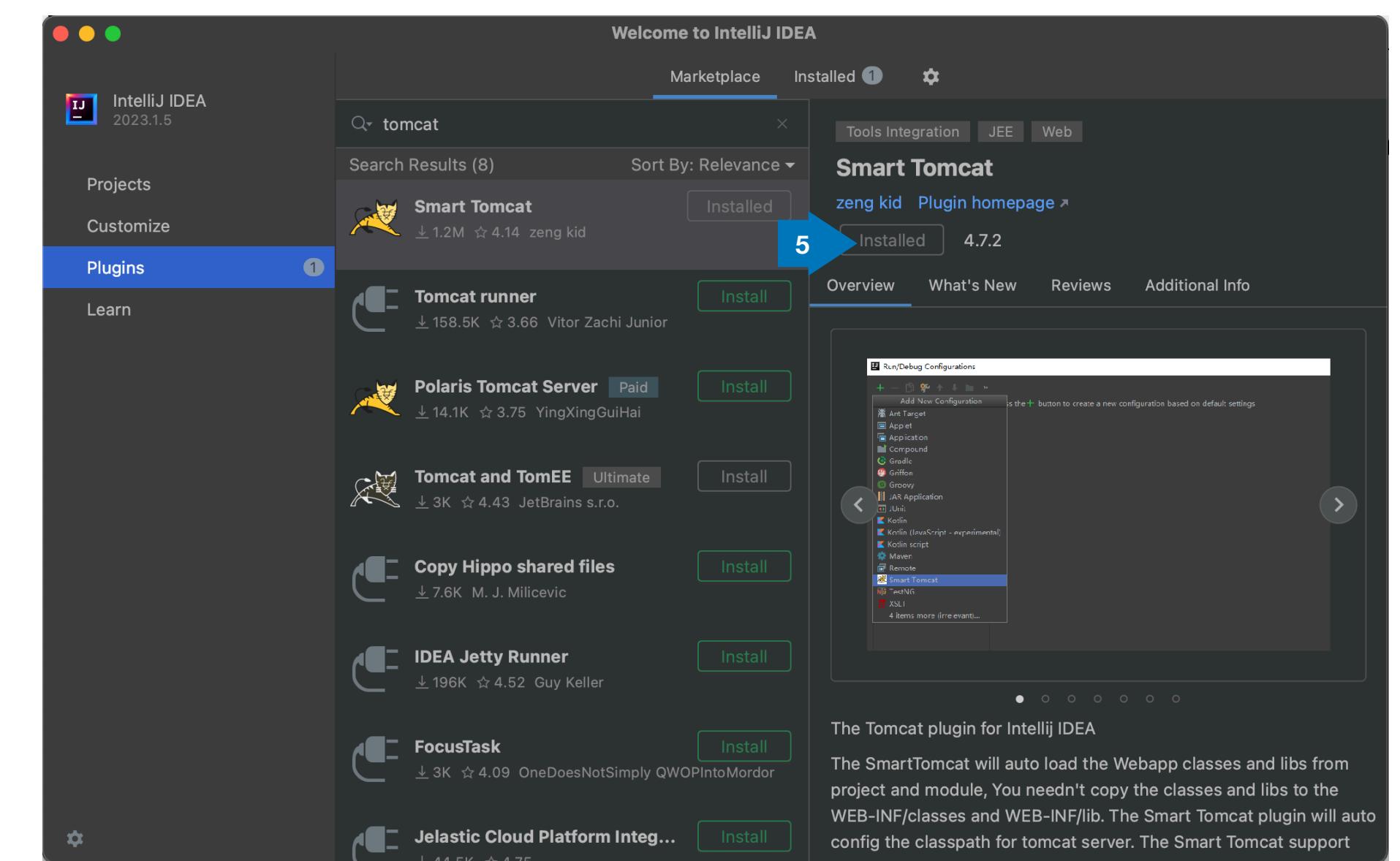
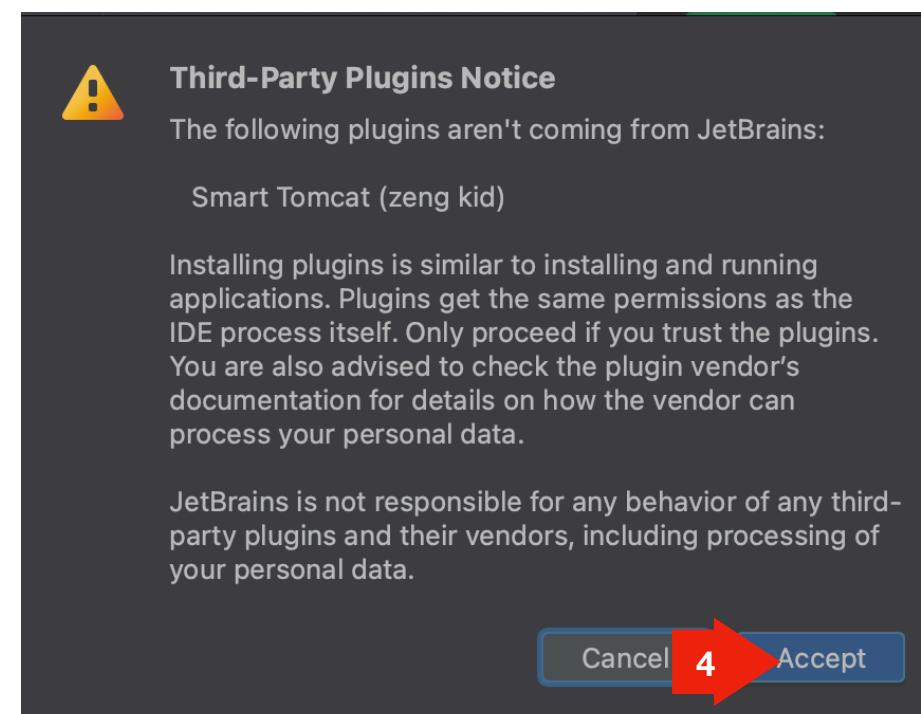
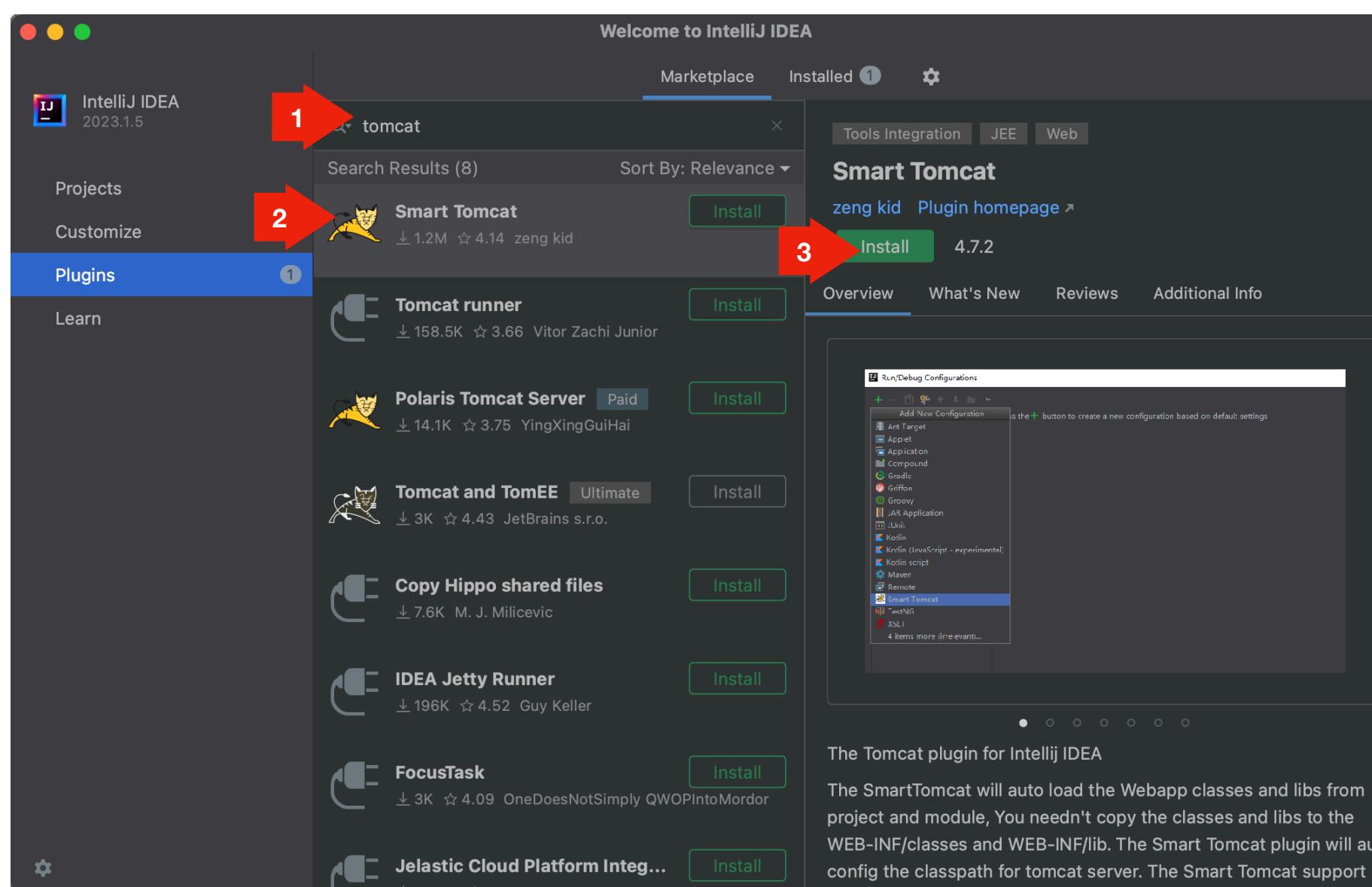
```
m pom.xml (myweb) ×
17   </dependencies>
18   <build>
19     <finalName>myweb</finalName>
20   <plugins>
21     <plugin>
22       <groupId>org.apache.maven.plugins</groupId>
23       <artifactId>maven-war-plugin</artifactId>
24       <version>3.3.1</version>
25     </plugin>
26   </plugins>
27 </build>
28 </project>
29
```

tomcat 플러그인 설치



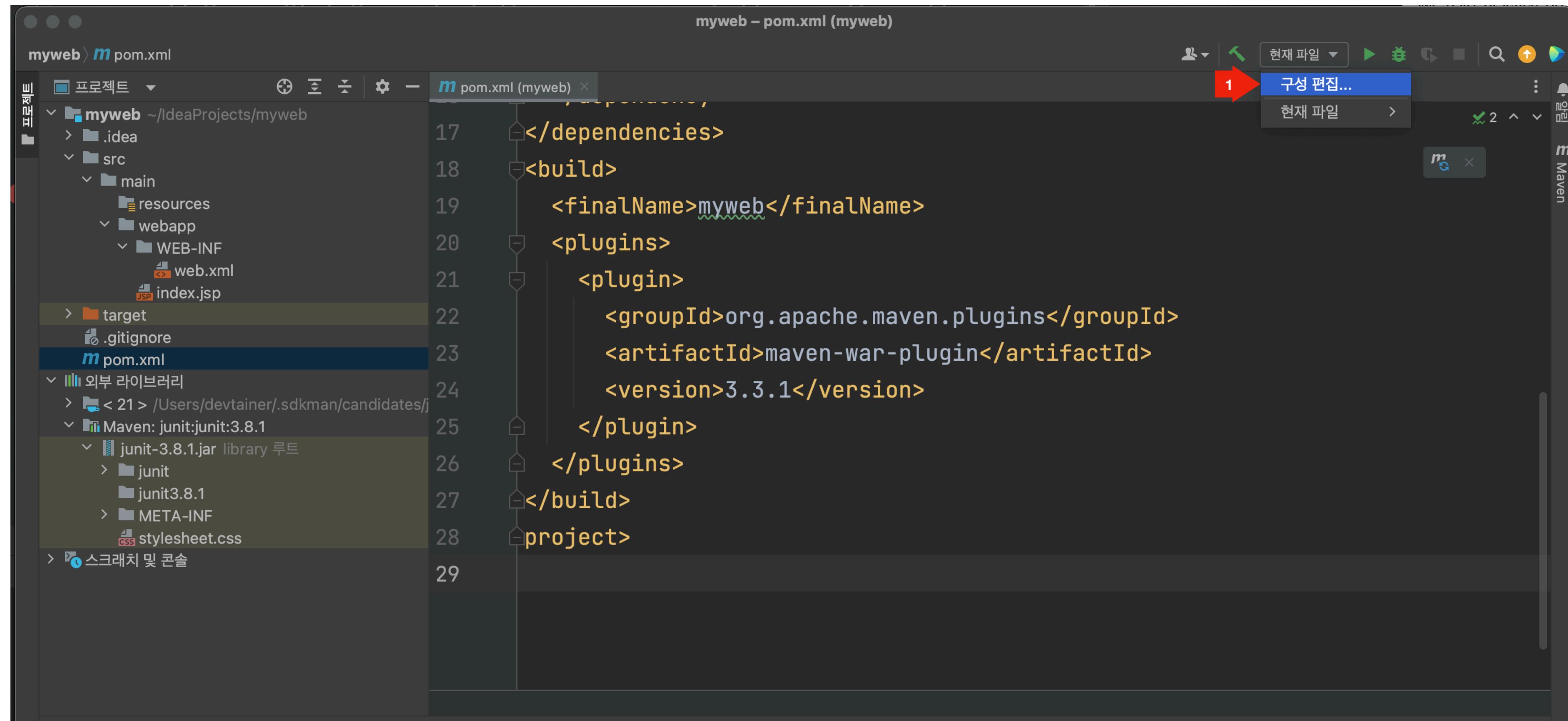
Tomcat Plugin 설치

Smart Tomcat Plugin



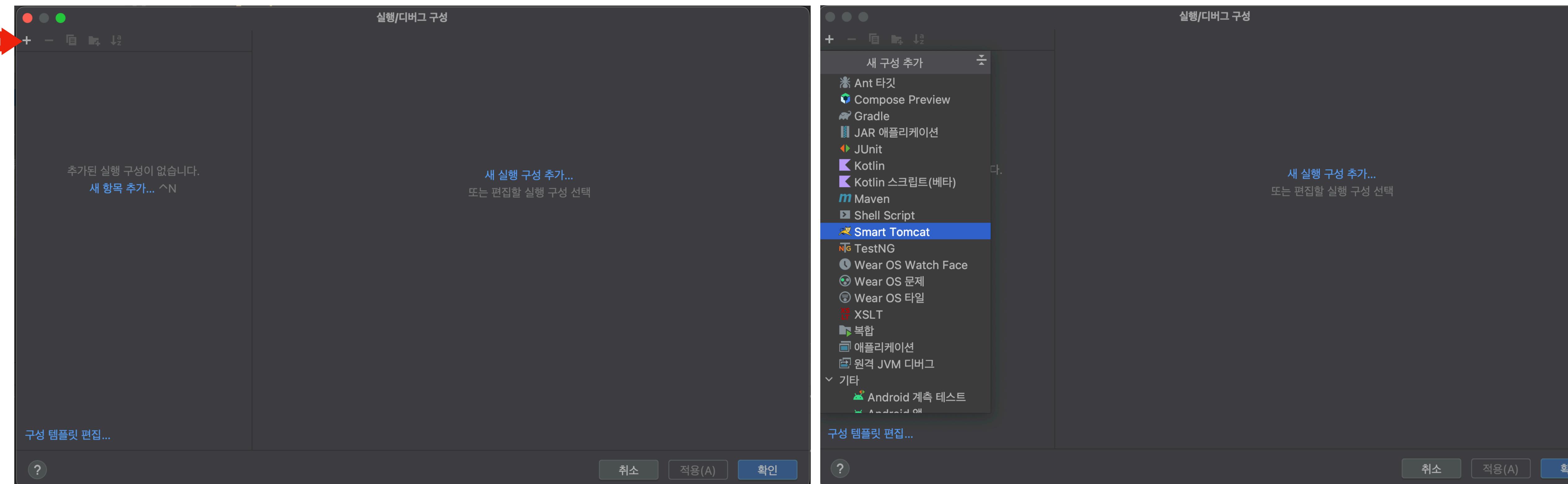
Tomcat Plugin 설정

Smart Tomcat Plugin



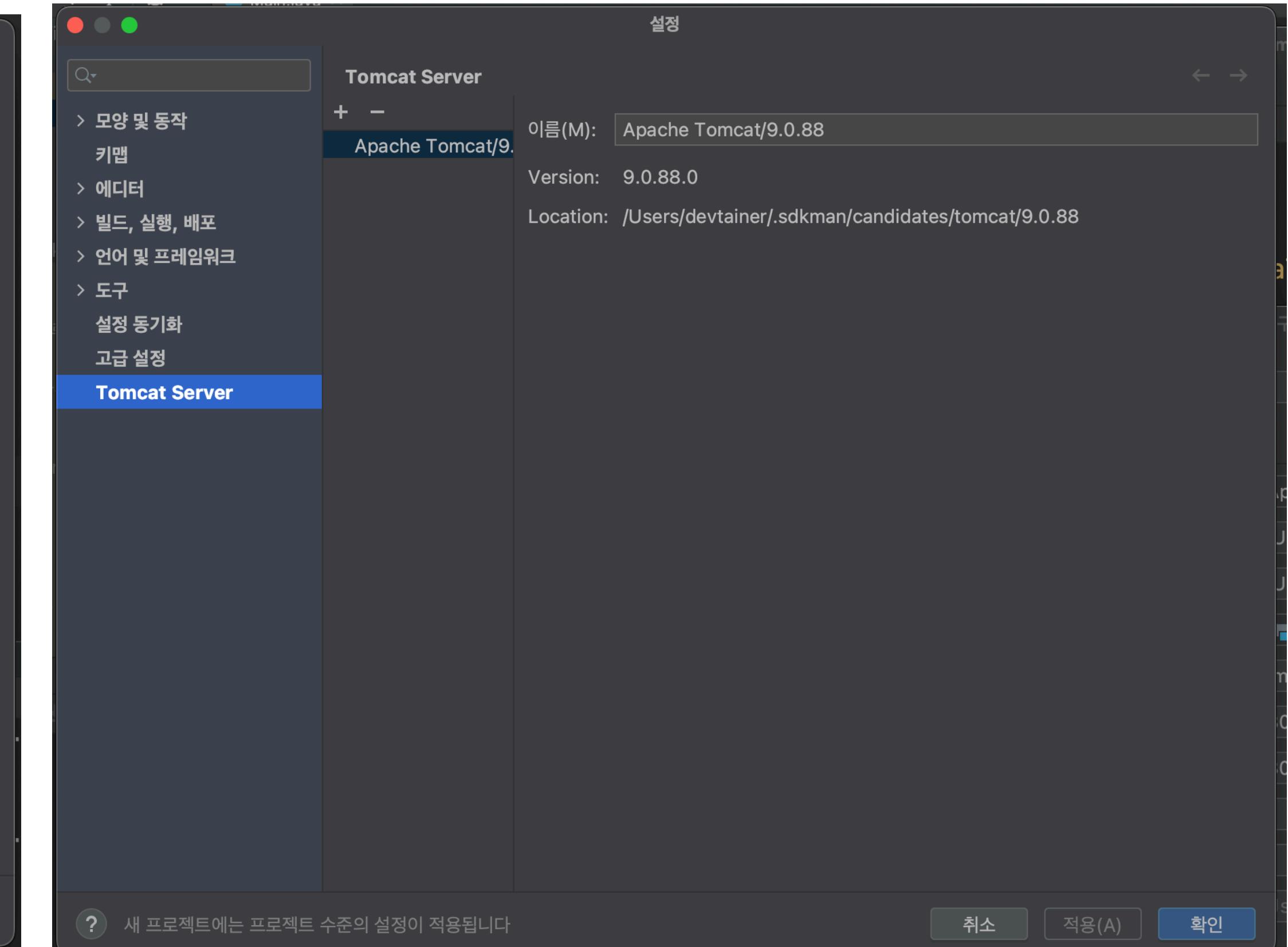
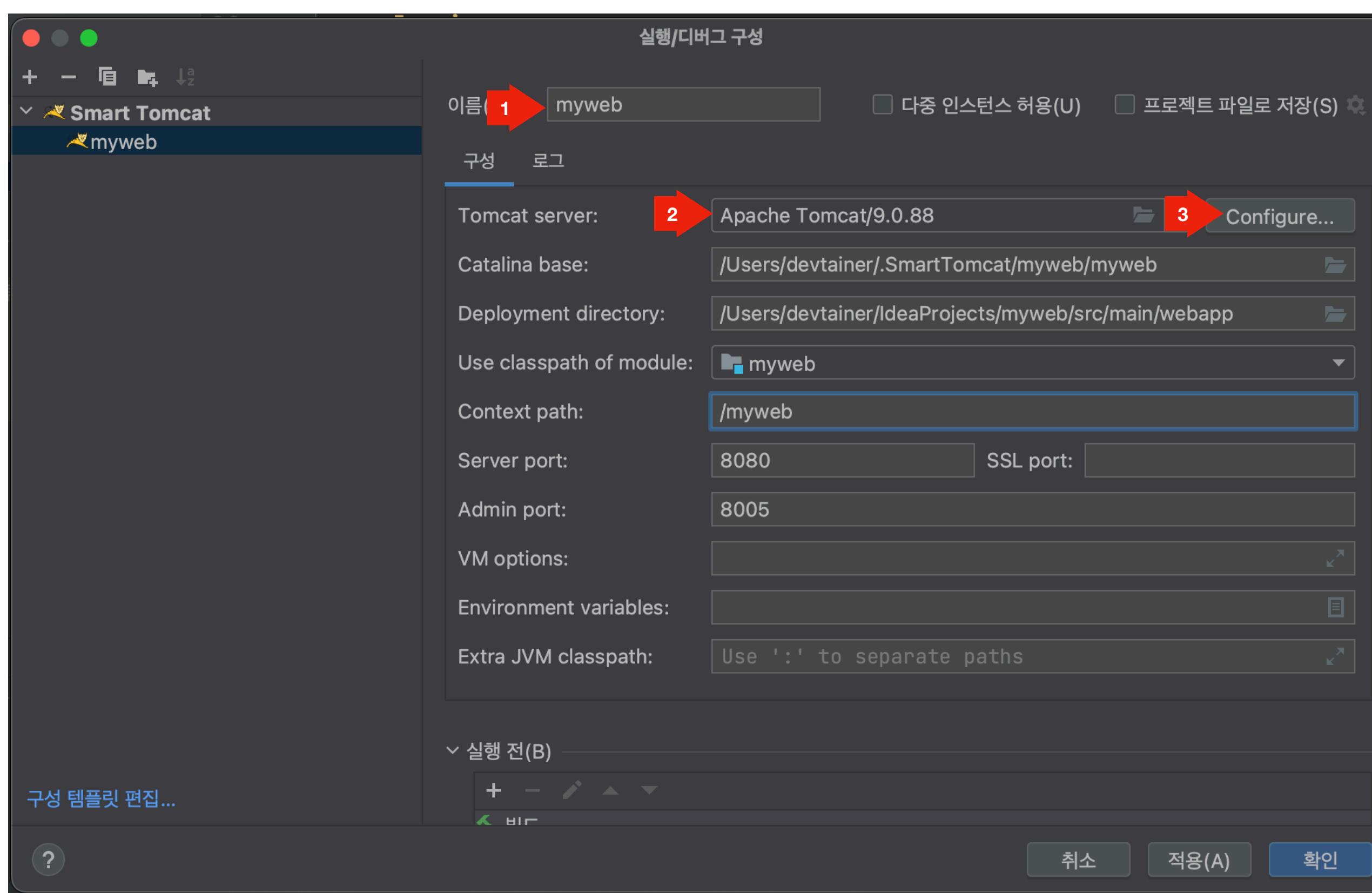
Tomcat Plugin 설정

Smart Tomcat Plugin



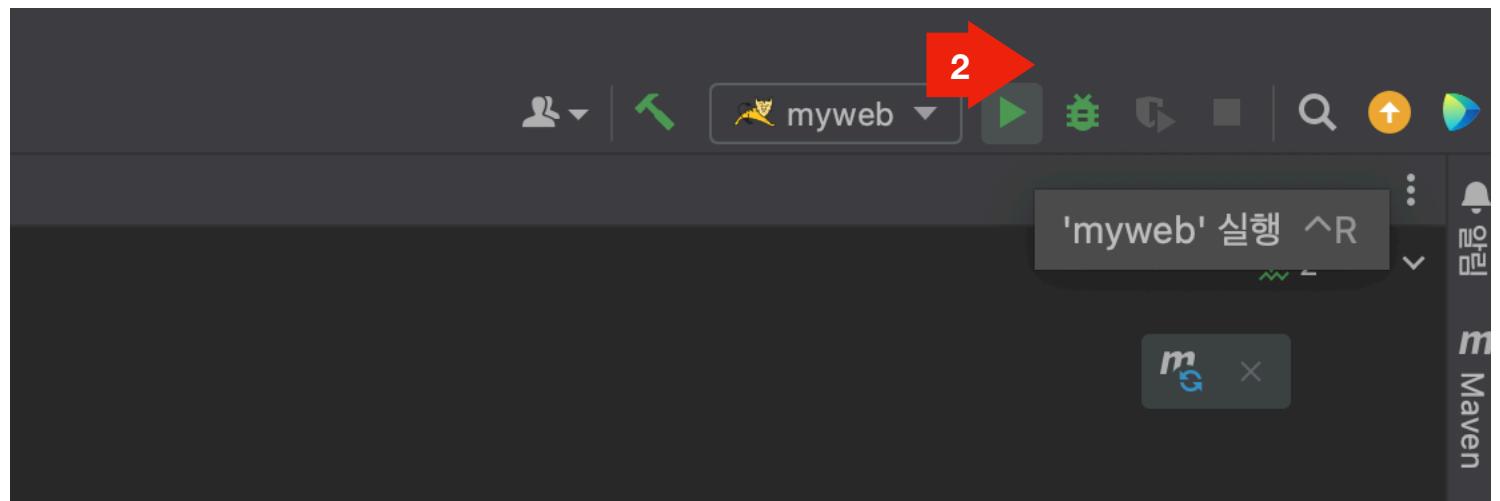
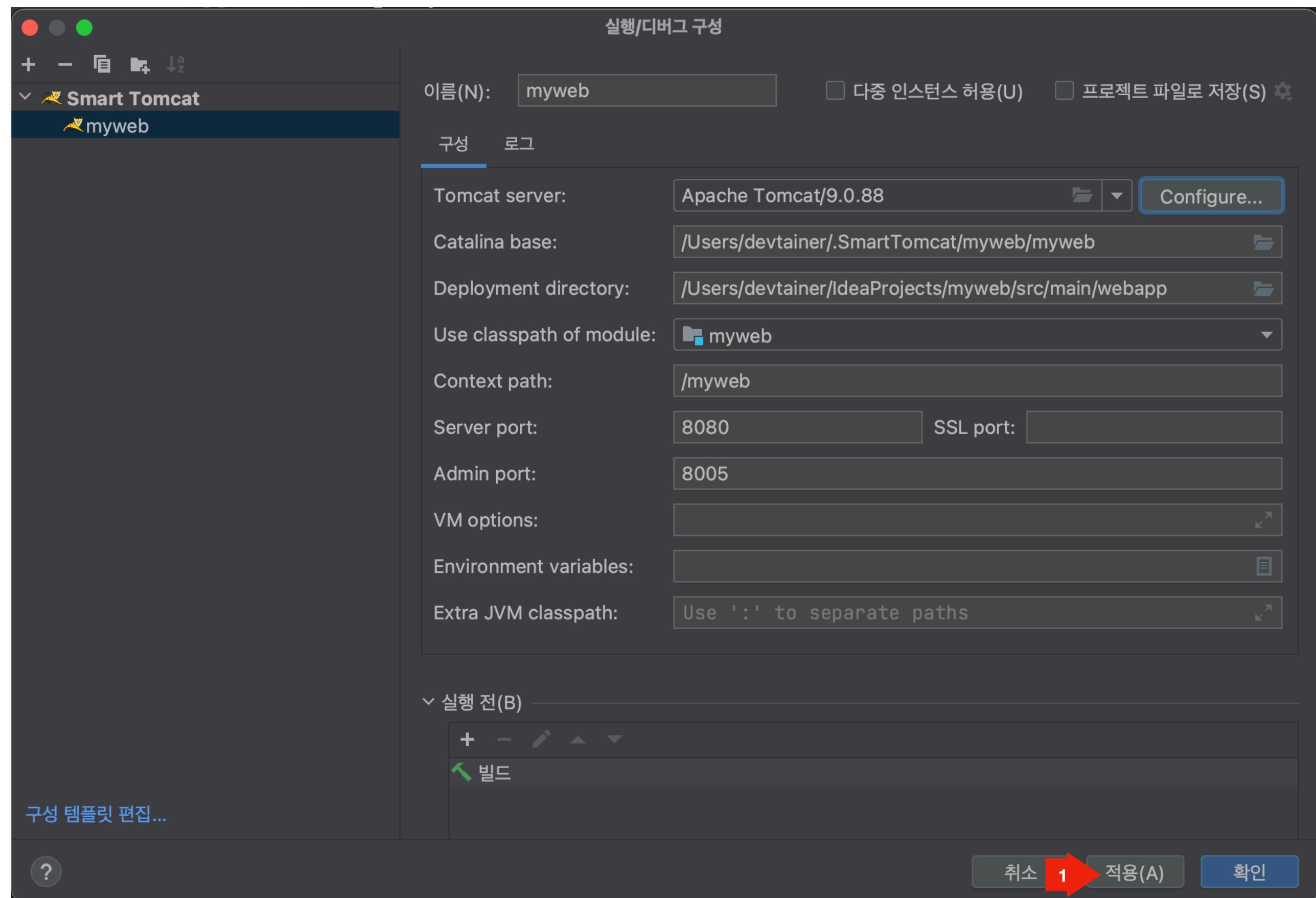
Tomcat Plugin 설정

Smart Tomcat Plugin



Tomcat Plugin 설정

Smart Tomcat Plugin



Tomcat Plugin 설정

Smart Tomcat Plugin

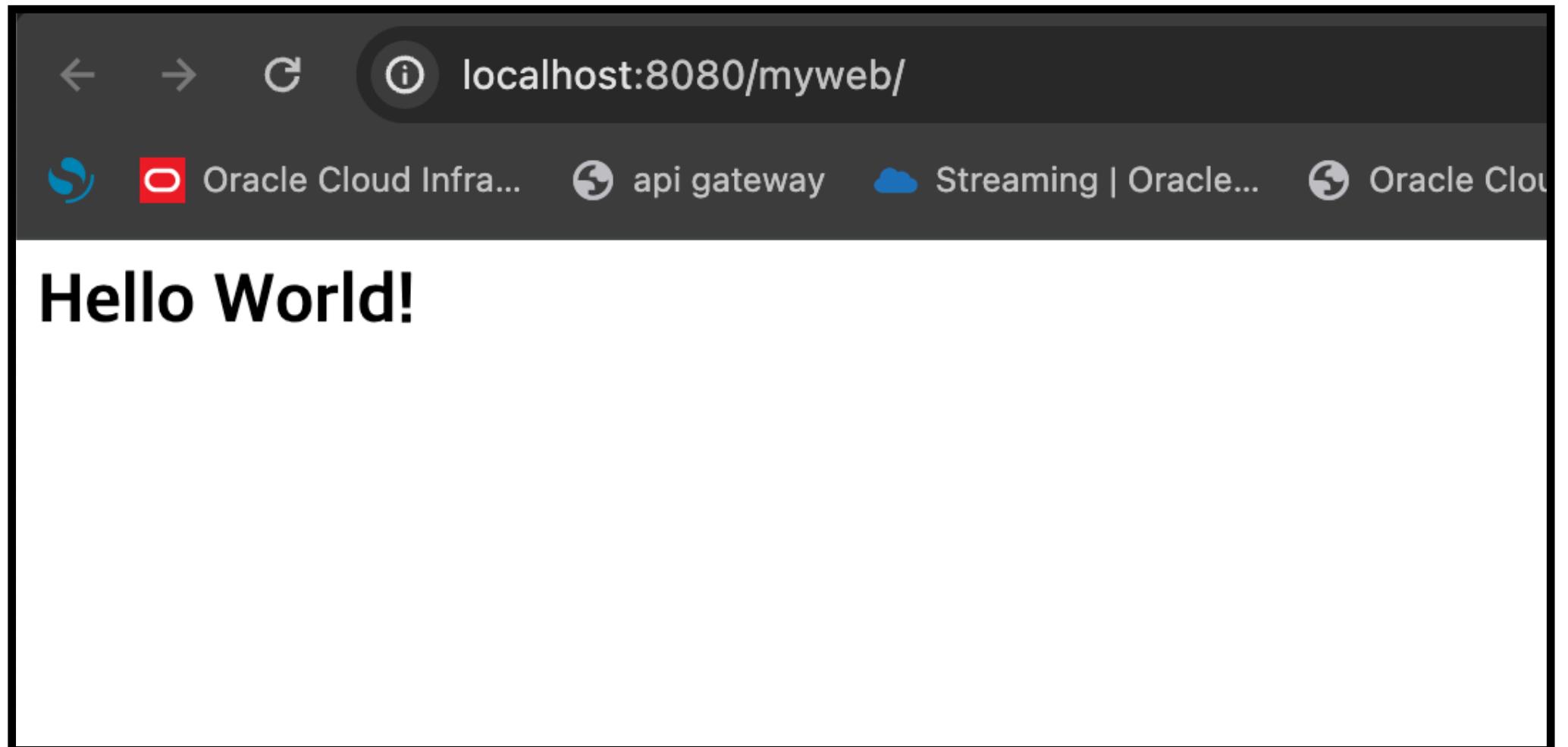
The screenshot shows the IntelliJ IDEA interface with the project 'myweb' open. The central part of the screen displays the `pom.xml` file content:

```
myweb - pom.xml (myweb)
...
</dependencies>
<build>
    <finalName>myweb</finalName>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-war-plugin</artifactId>
            <version>3.3.1</version>
        </plugin>
    </plugins>
</build>
<project>
```

The `<finalName>myweb</finalName>` element is highlighted. Below the code editor, the Maven tool window shows the following log output:

```
실행: myweb
[1] 12-Jun-2024 05:02:28.720 INFO [main] org.apache.catalina.core.StandardService.startInternal 서비스 [Catalina]를 시작합니다.
[1] 12-Jun-2024 05:02:28.720 INFO [main] org.apache.catalina.core.StandardEngine.startInternal 서버 엔진을 시작합니다.
[1] 12-Jun-2024 05:02:28.724 INFO [main] org.apache.catalina.startup.HostConfig.deployDescriptor 배치 descriptor
[1] 12-Jun-2024 05:02:28.876 INFO [main] org.apache.catalina.startup.HostConfig.deployDescriptor 배치 descriptor
[1] 12-Jun-2024 05:02:28.878 INFO [main] org.apache.coyote.AbstractProtocol.start 프로토콜 핸들러 ["http-nio-8080"]
[1] 12-Jun-2024 05:02:28.884 INFO [main] org.apache.catalina.startup.Catalina.start 서버가 [185] 밀리초 내에 시작되었습니다.
[1] http://localhost:8080/myweb
```

A red arrow points to the URL `http://localhost:8080/myweb` in the log output.



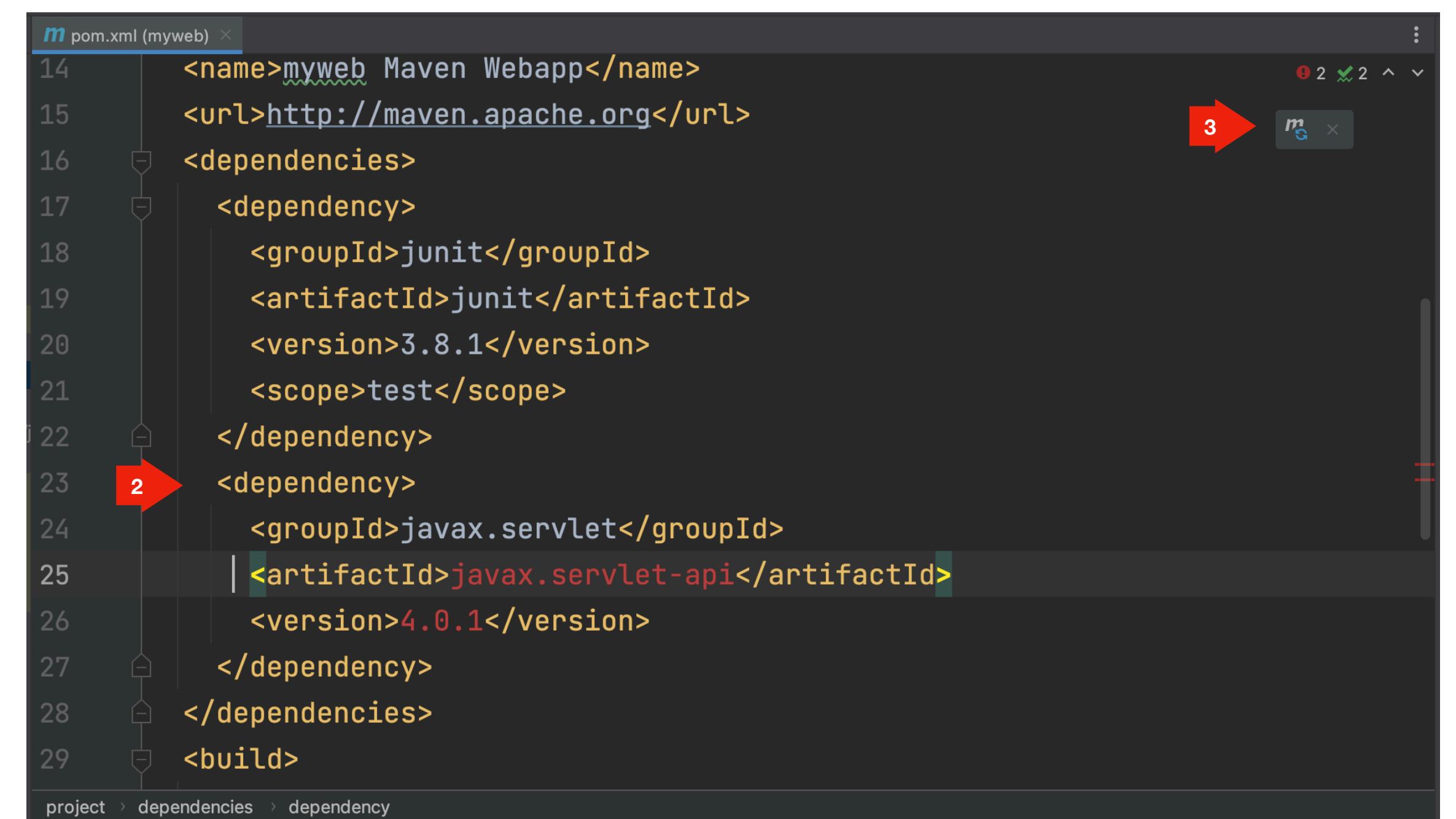
JSP / Servlet



Servlet 등록 pom.xml 변경



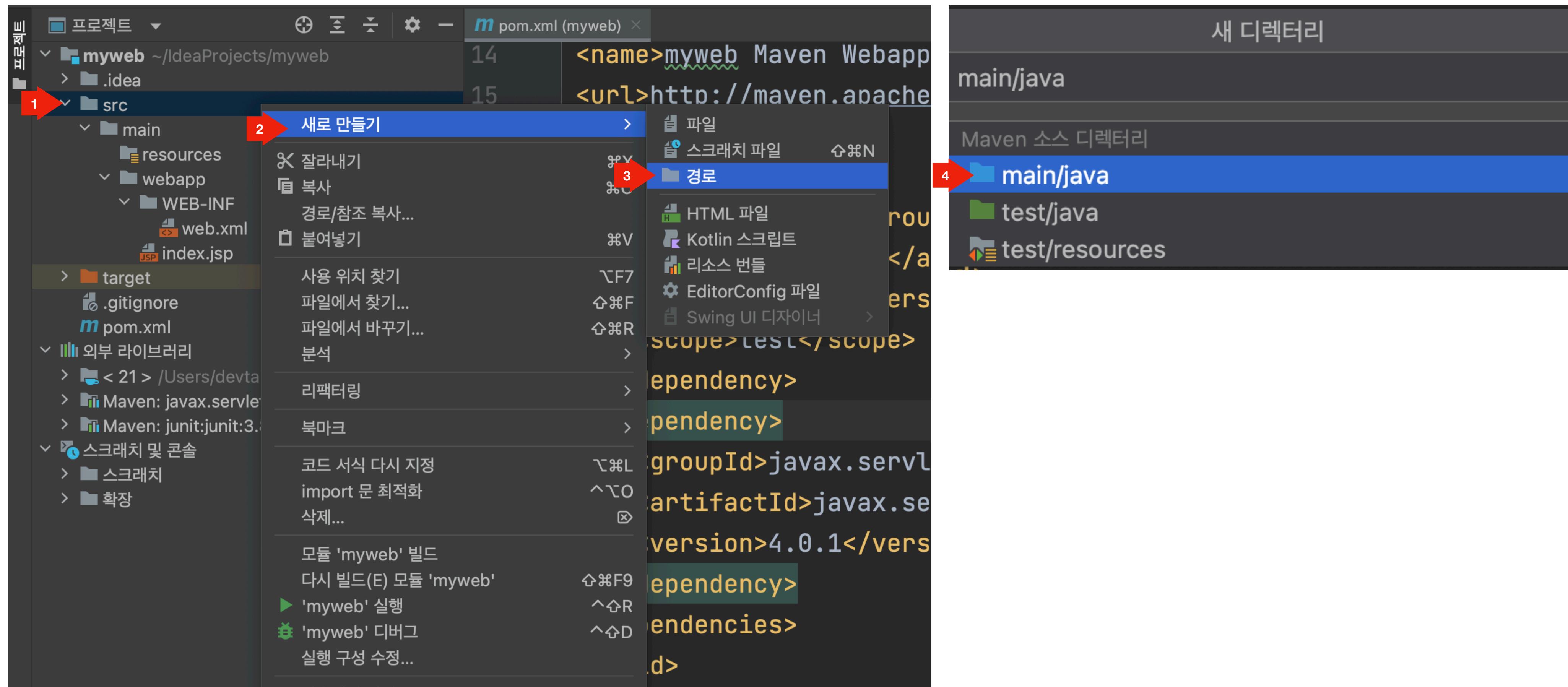
```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>org.example</groupId>
4   <artifactId>myweb</artifactId>
5
6   <properties>
7     <maven.compiler.source>21</maven.compiler.source>
8     <maven.compiler.target>21</maven.compiler.target>
9   </properties>
10
11  <packaging>war</packaging>
12  <version>1.0.0</version>
13  <name>myweb Maven Webapp</name>
14  <url>http://maven.apache.org</url>
15
16  <dependencies>
```



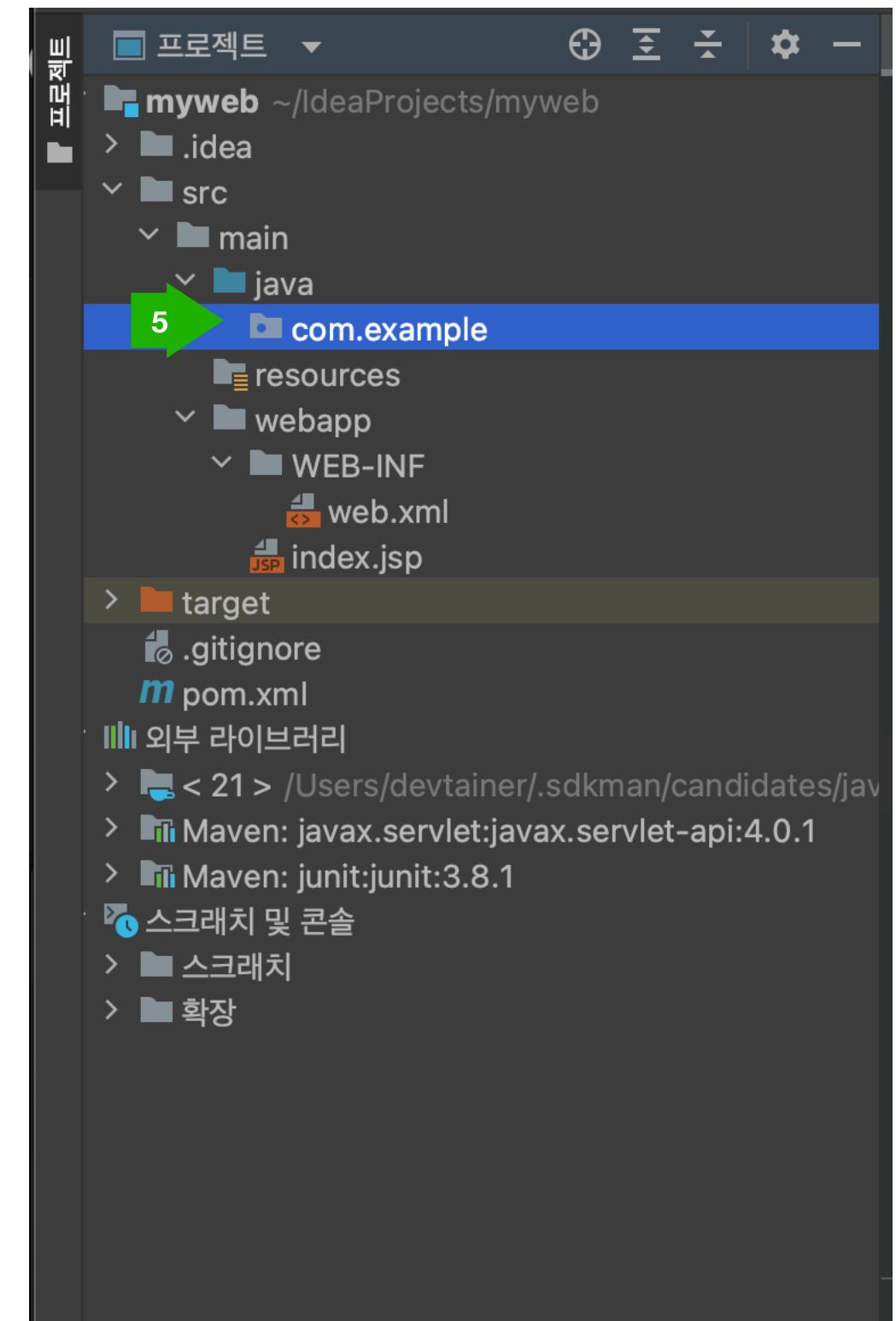
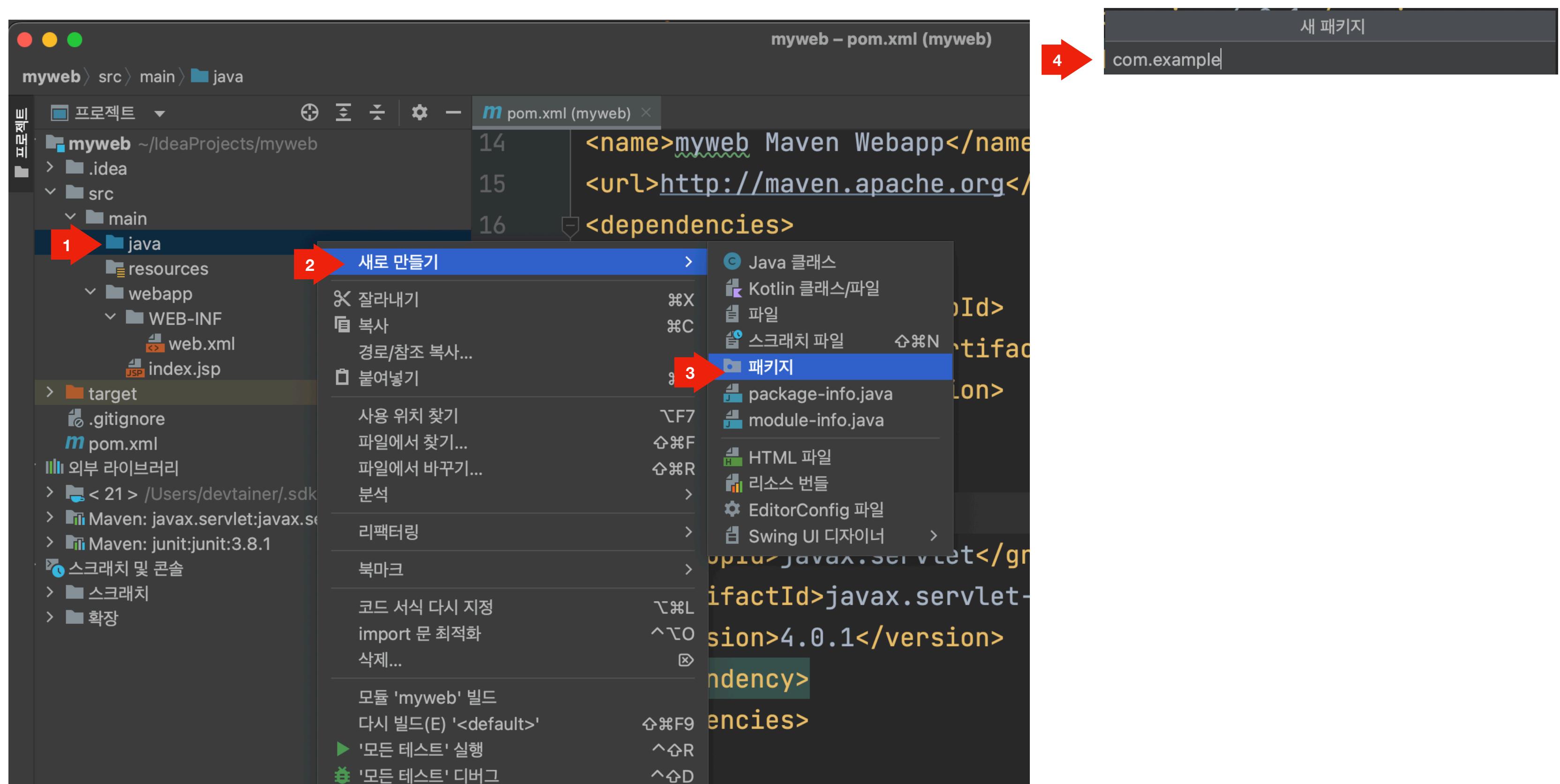
```
14  <name>myweb Maven Webapp</name>
15  <url>http://maven.apache.org</url>
16  <dependencies>
17    <dependency>
18      <groupId>junit</groupId>
19      <artifactId>junit</artifactId>
20      <version>3.8.1</version>
21      <scope>test</scope>
22    </dependency>
23    <dependency>
24      <groupId>javax.servlet</groupId>
25      <artifactId>javax.servlet-api</artifactId>
26      <version>4.0.1</version>
27    </dependency>
28  </dependencies>
29  <build>
```

Servlet 등록

경로 생성

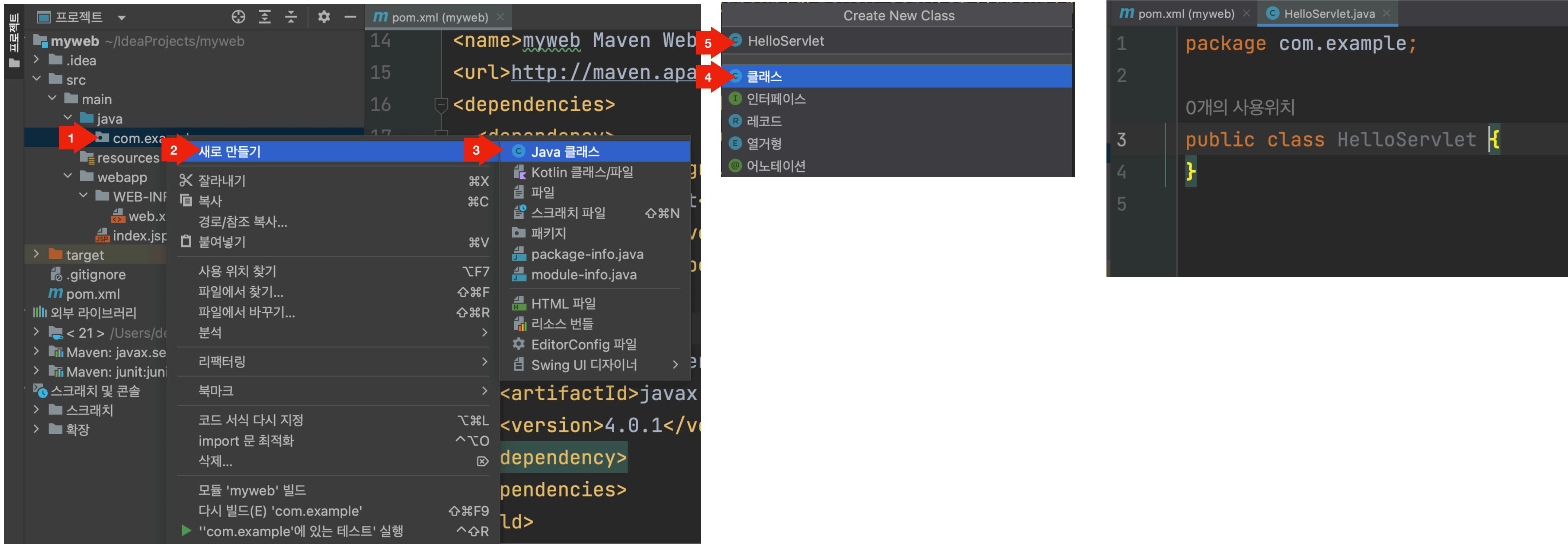


Servlet 등록 패키지 생성



Servlet 등록

서블릿 파일 생성



Servlet 등록

서블릿 파일 코드 변경

The screenshot shows a Java code editor with the following code:

```
10 public class HelloServlet extends HttpServlet {  
11  
12 }  
13  
14
```

A context menu is open at the end of the class definition (line 12). Step 1 highlights the "생성..." (Create...) option. Step 2 highlights the "메서드 재정의..." (Override Method...) option in the submenu. Step 3 highlights the "doGet(req:HttpServletRequest, resp:HttpServletResponse)" method in the "재정의/구현할 메서드 선택" (Select Overridden/Implemented Method) dialog.

Below the code editor, the generated code is shown:

```
10 public class HelloServlet extends HttpServlet {  
11     1개 사용 위치  
12     @Override  
13     4 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException  
14         super.doGet(req, resp);  
15     }  
16 }
```

Step 4 highlights the newly generated `doGet` method call.

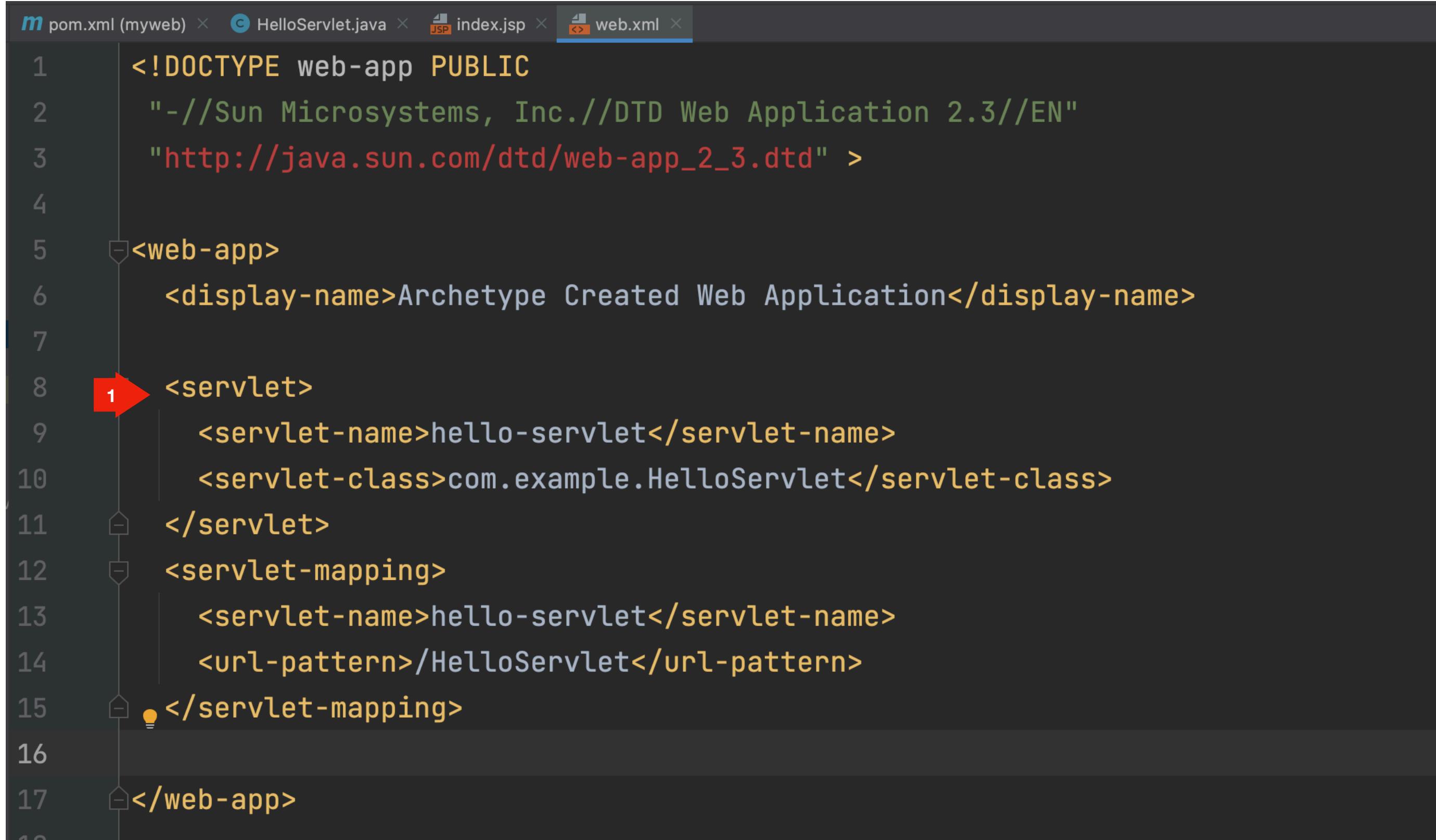
Servlet 등록

서블릿 파일 코드 변경

```
10 public class HelloServlet extends HttpServlet {  
11  
12     // 0개의 사용위치  
13     @Override  
14     protected void doGet(HttpServletRequest req, HttpServletResponse resp)  
15         throws ServletException, IOException {  
16         resp.setContentType("text/html; charset=UTF-8");  
17         PrintWriter out = resp.getWriter();  
18         out.println("Hello, Servlet!");  
19         out.close();  
20     }  
21 }
```

Servlet 등록

서블릿 파일 등록



```
pom.xml (myweb) × C HelloServlet.java × index.jsp × web.xml ×  
1 <!DOCTYPE web-app PUBLIC  
2   "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
3   "http://java.sun.com/dtd/web-app_2_3.dtd" >  
4  
5 <web-app>  
6   <display-name>Archetype Created Web Application</display-name>  
7  
8   1 <servlet>  
9     <servlet-name>hello-servlet</servlet-name>  
10    <servlet-class>com.example.HelloServlet</servlet-class>  
11  </servlet>  
12  <servlet-mapping>  
13    <servlet-name>hello-servlet</servlet-name>  
14    <url-pattern>/HelloServlet</url-pattern>  
15  </servlet-mapping>  
16  
17 </web-app>
```

Maven Build

- mvn clean package

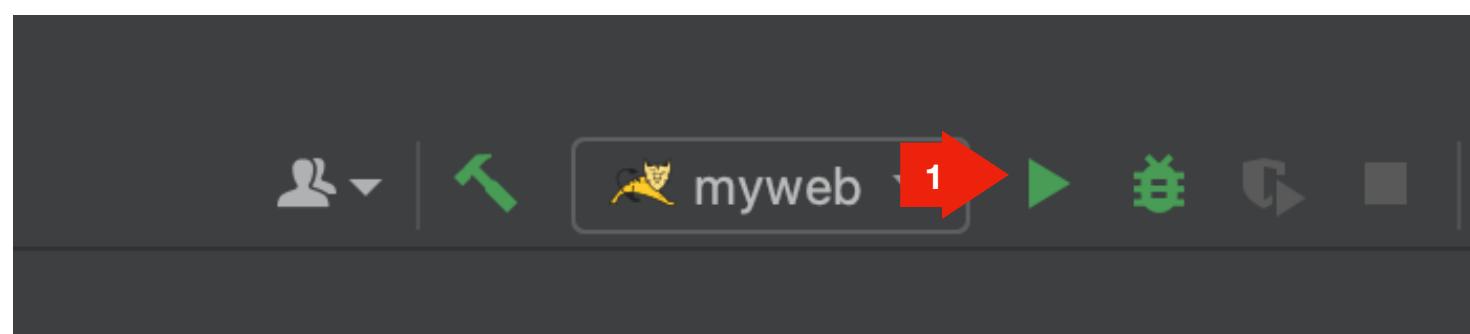
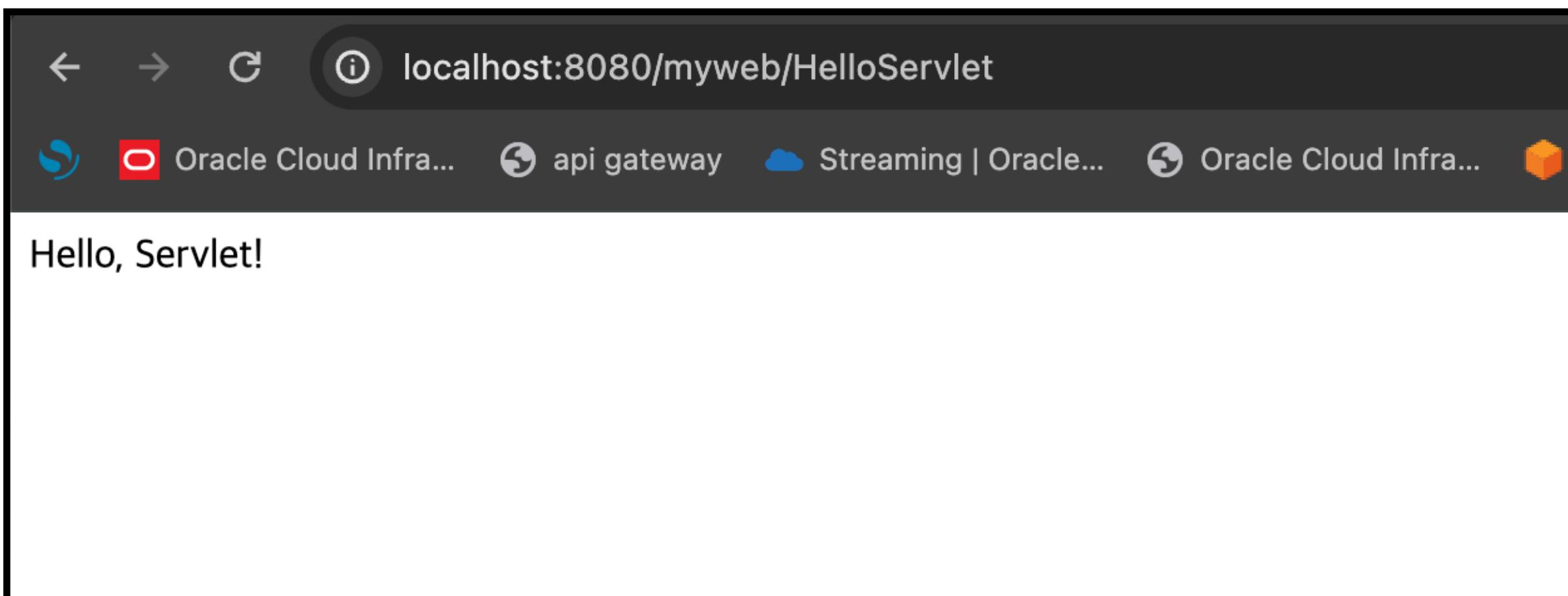


A screenshot of a terminal window titled "터미널: 로컬". The window displays the output of a Maven build command. The output includes:

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.811 s
[INFO] Finished at: 2024-06-12T06:08:07+09:00
[INFO] -----
```

The terminal has a dark background with light-colored text. The title bar is at the top, and there are some icons and text on the left side of the window.

서버 실행

A screenshot of the IDE's terminal window titled 'myweb [compile]'. It contains several log entries from Tomcat. The entries are in red text and include: '12-Jun-2024 06:11:03.630 INFO [main] org.apache.catalina.startup.HostConfig.deployDescriptor()', '12-Jun-2024 06:11:03.788 INFO [main] org.apache.jasper.servlet.TldScanner...', '12-Jun-2024 06:11:03.815 INFO [main] org.apache.catalina.startup.HostConfig...', '12-Jun-2024 06:11:03.818 INFO [main] org.apache.coyote.AbstractProtocol...', and '12-Jun-2024 06:11:03.825 INFO [main] org.apache.catalina.startup.Cat...'. At the bottom of the log, there is a blue hyperlink: 'http://localhost:8080/myweb'.

2nd Servlet 추가

The screenshot shows a Java development environment with two tabs open: `HiServlet.java` and `HelloServlet.java`. The `HiServlet.java` tab is active, displaying the following code:

```
package com.example;

public class HiServlet {
```

The `HelloServlet.java` tab shows the following code:

```
package com.example;

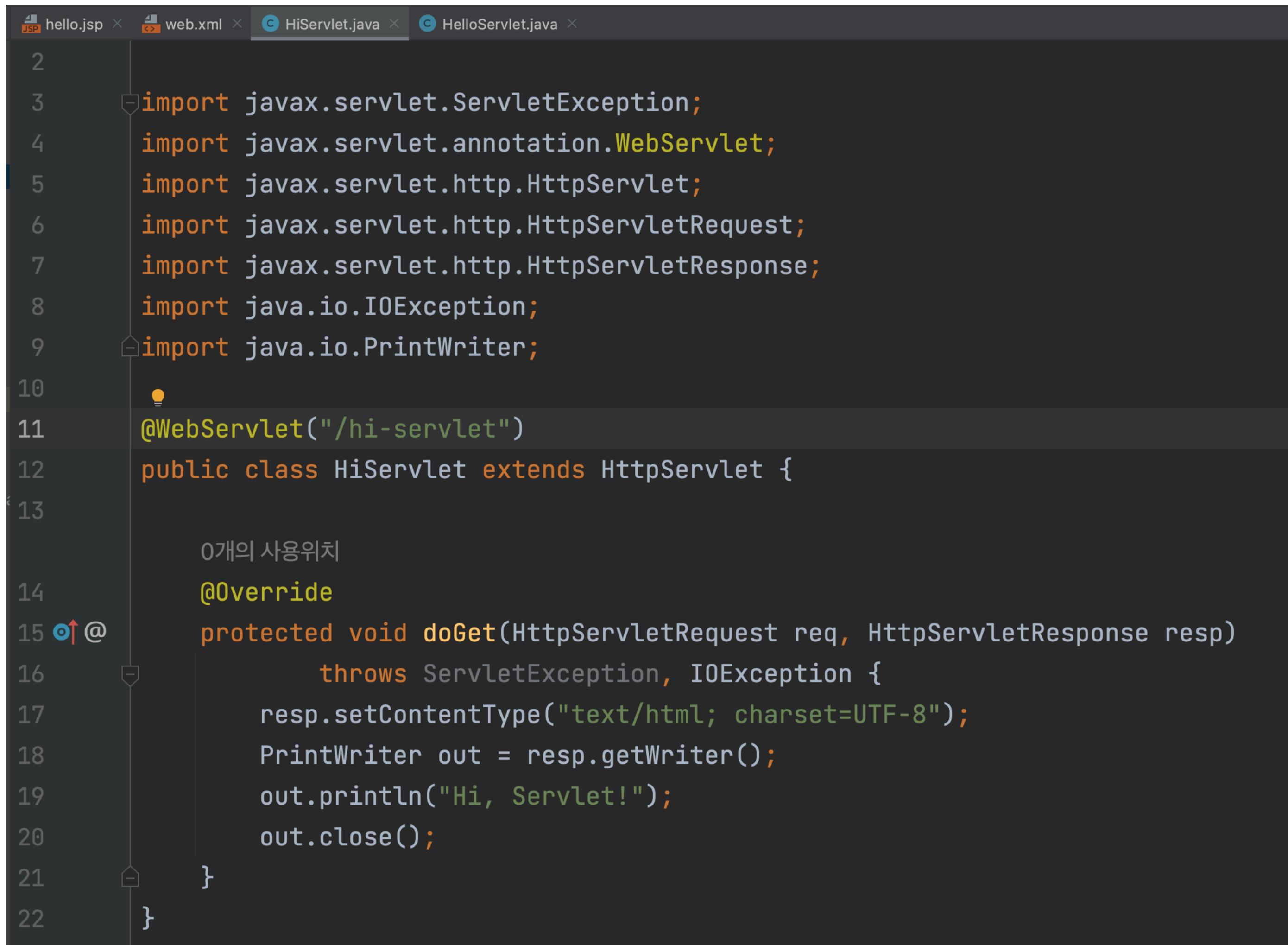
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

public class HelloServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {
        resp.setContentType("text/html; charset=UTF-8");
        PrintWriter out = resp.getWriter();
        out.println("Hi, Servlet!");
        out.close();
    }
}
```

The project structure on the left side of the IDE shows a package named `com.example` containing two classes: `HelloServlet` and `HiServlet`.

Servlet Annotation 등록

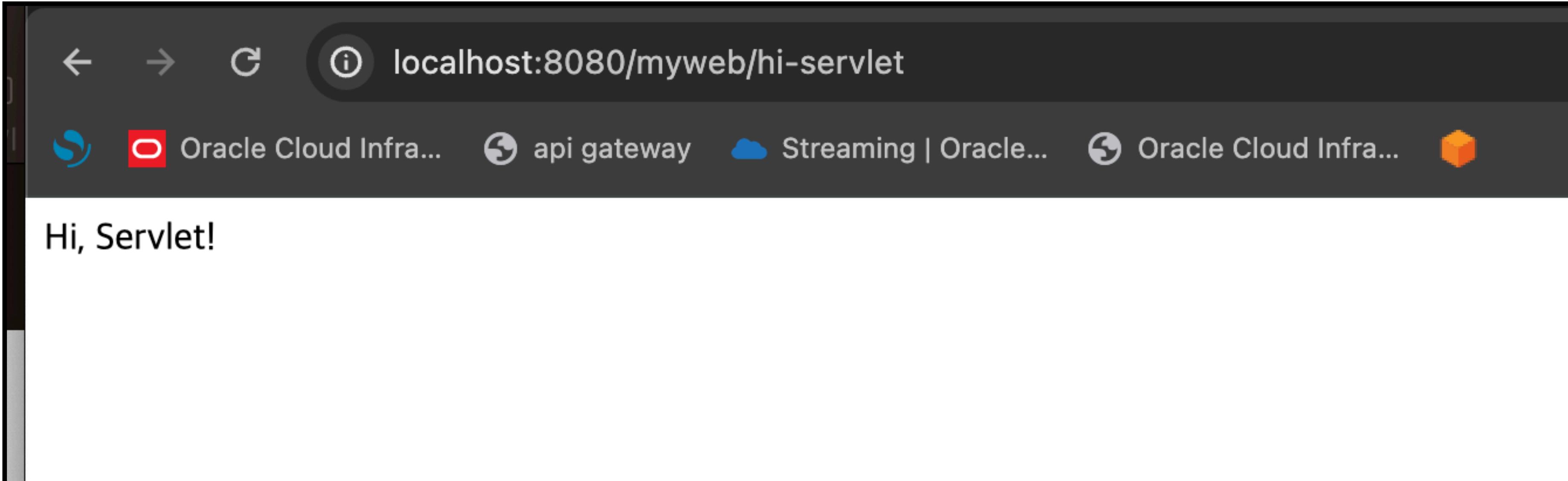


The screenshot shows a Java code editor with a dark theme. The file is named `HiServlet.java`. The code defines a servlet class `HiServlet` that extends `HttpServlet`. It uses the `@WebServlet` annotation to map the URL `/hi-servlet`. The code also includes methods for handling GET requests and setting the response content type to `text/html; charset=UTF-8`.

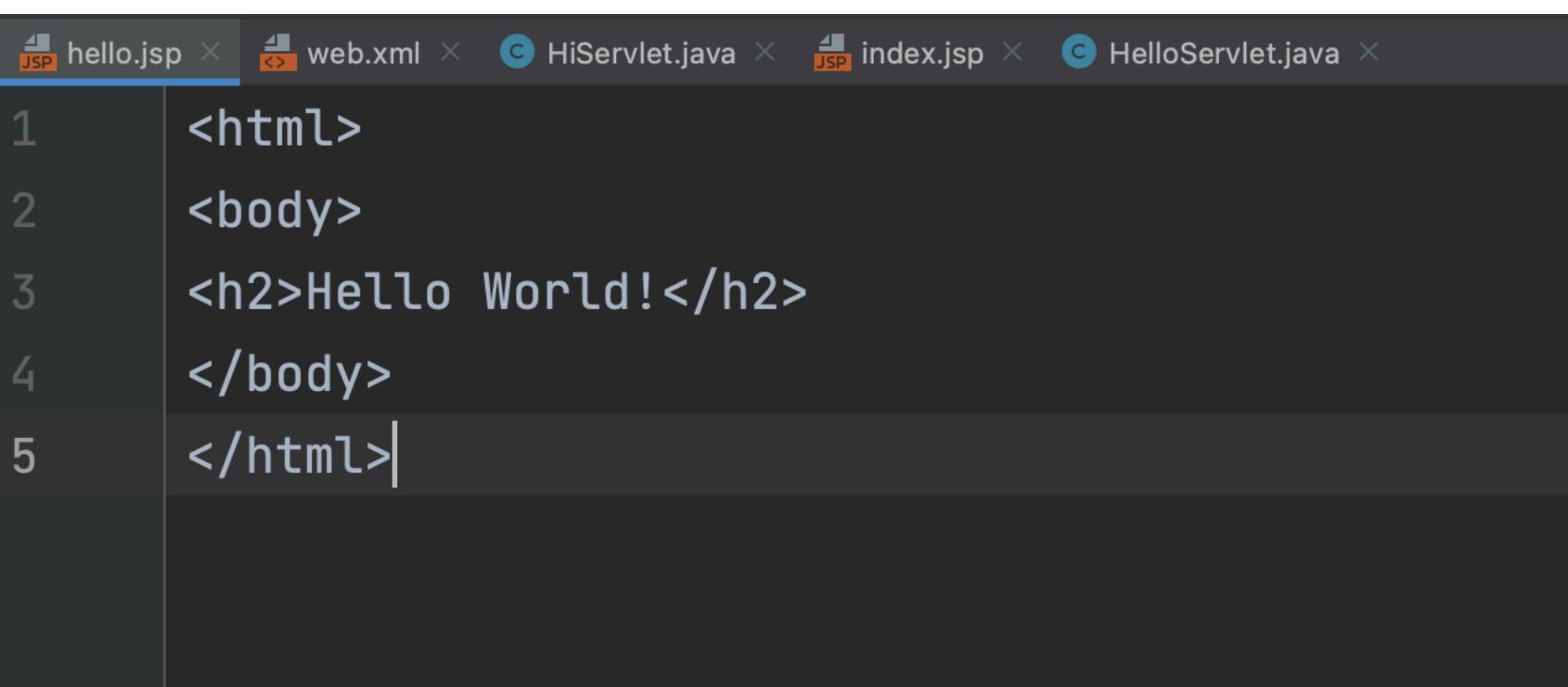
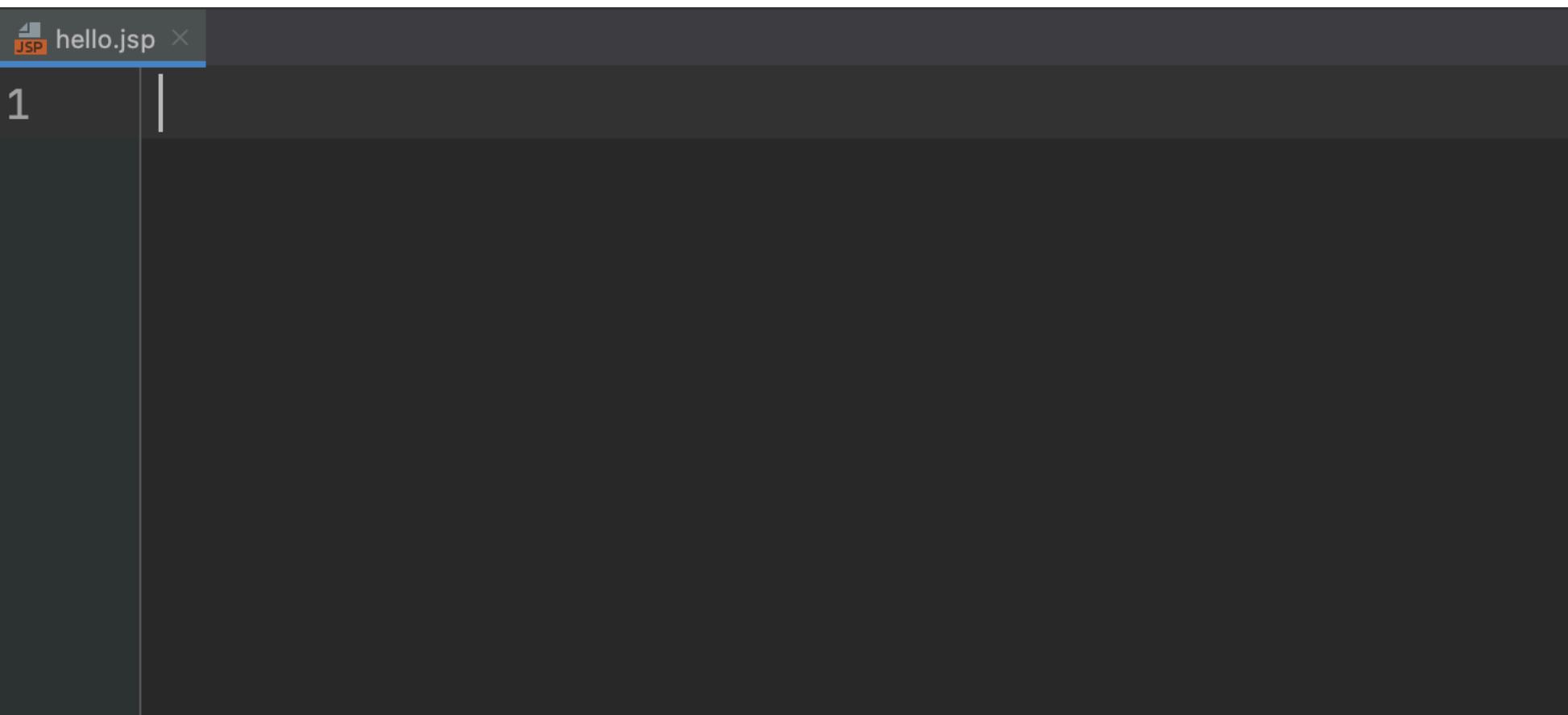
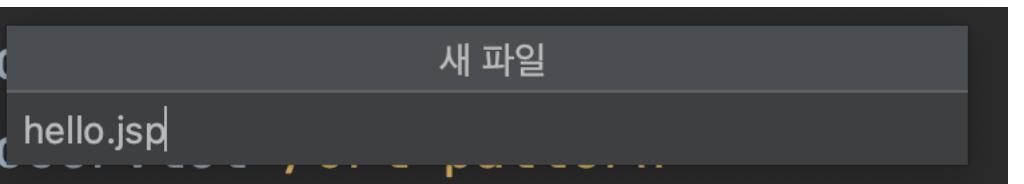
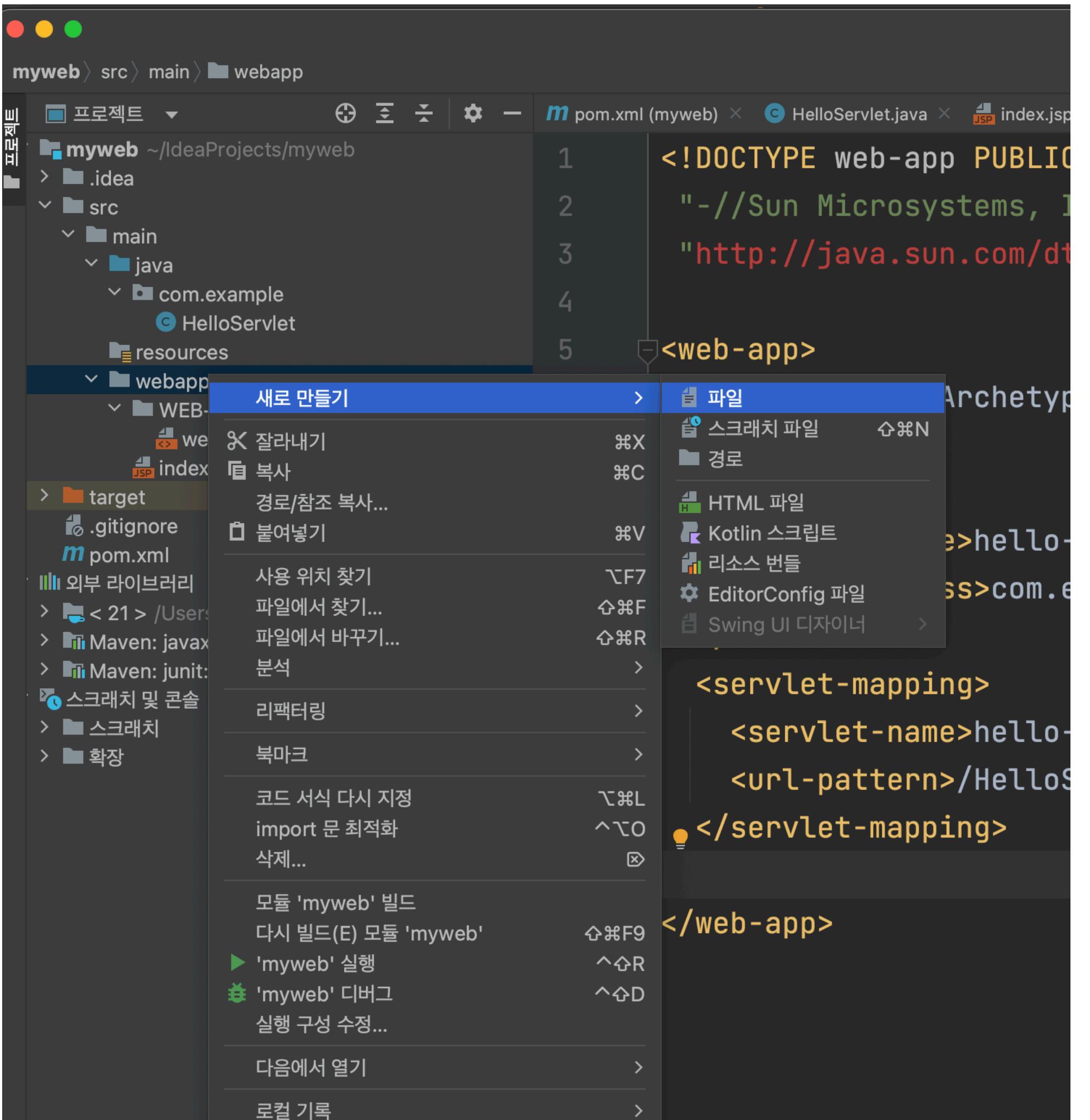
```
1  package com.tistory;
2
3  import javax.servlet.ServletException;
4  import javax.servlet.annotation.WebServlet;
5  import javax.servlet.http.HttpServlet;
6  import javax.servlet.http.HttpServletRequest;
7  import javax.servlet.http.HttpServletResponse;
8  import java.io.IOException;
9  import java.io.PrintWriter;
10
11 @WebServlet("/hi-servlet")
12 public class HiServlet extends HttpServlet {
13
14     // 0개의 사용위치
15     @Override
16     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
17             throws ServletException, IOException {
18         resp.setContentType("text/html; charset=UTF-8");
19         PrintWriter out = resp.getWriter();
20         out.println("Hi, Servlet!");
21         out.close();
22     }
}
```

Servlet 실행

- mvn clean package
- 서버 실행

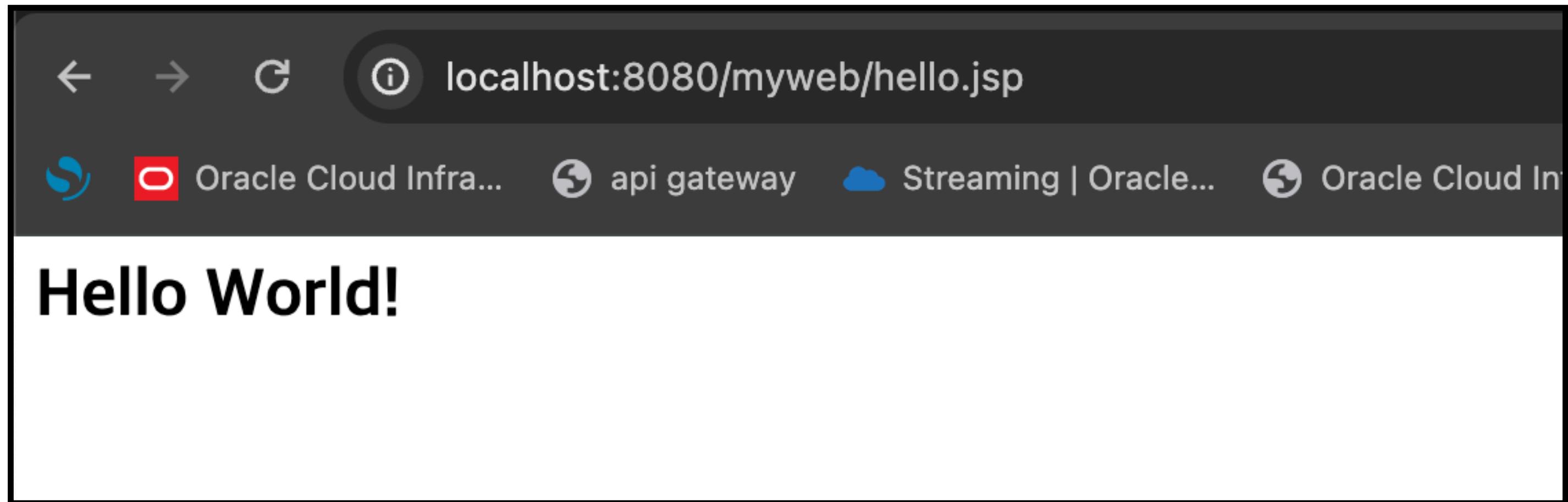


JSP 등록



JSP 등록

서블릿 파일 등록



운영 환경 구축



서버 운영 환경

- Local PC => IntelliJ => Tomcat
- Dev AWS => EC2 => Tomcat
- Integration
- QA
- Staging
- Production AWS => EC2 => Tomcat
 - LB
 - MySQL

Tomcat User 등록

- %TOMCAT_HOME%/conf/tomcat-users.xml

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>

<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>
<user username="deployer" password="deployer" roles="manager-script"/>
<user username="tomcat" password="tomcat" roles="manager-gui"/>
```

Tomcat Manager

<http://localhost:8080/manager>

Tomcat Web Application Manager

Message: OK

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
					Start Stop Reload Undeploy

Sprint Boot를 이용한 개발

