

4장 웹 애플리케이션 이해하기

4.1 웹 애플리케이션

4.2 웹 애플리케이션의 기본 구조

4.3 컨테이너에서 웹 애플리케이션 실행하기

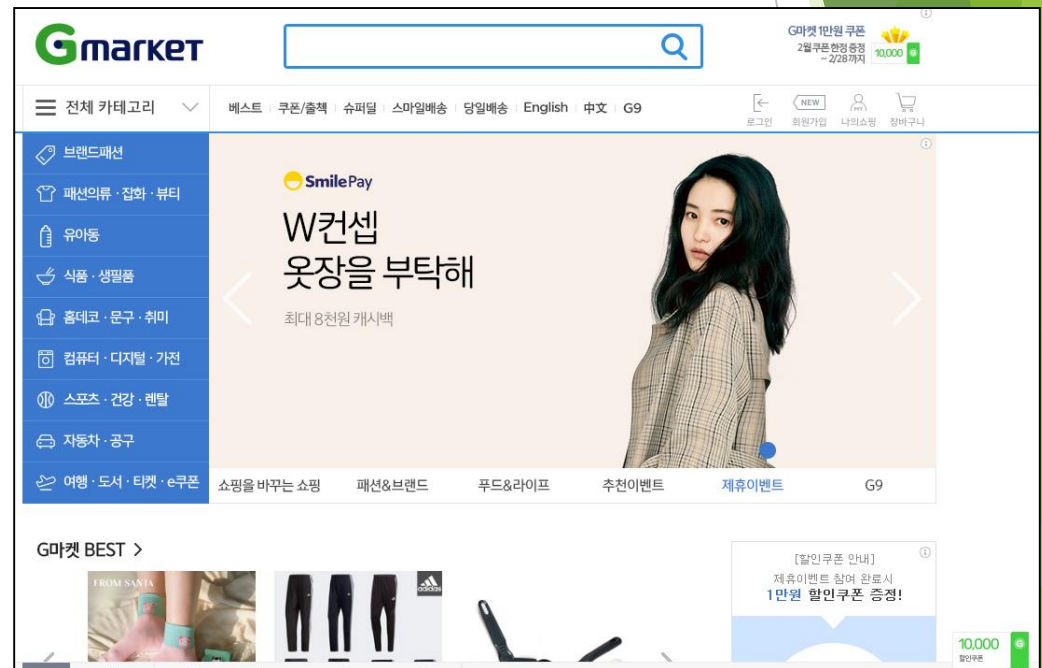
4.4 이클립스에서 웹 애플리케이션 실습하기

4.5 웹 애플리케이션 서비스하기

4.1 웹 애플리케이션

• 4.1.1 웹 애플리케이션 정의

- 기존의 정적 웹 애플리케이션 기능을 그대로 사용하면서 서버릿, JSP, 자바 클래스들을 추가하여 사용자에게 동적인 서비스를 제공하는 프로그램

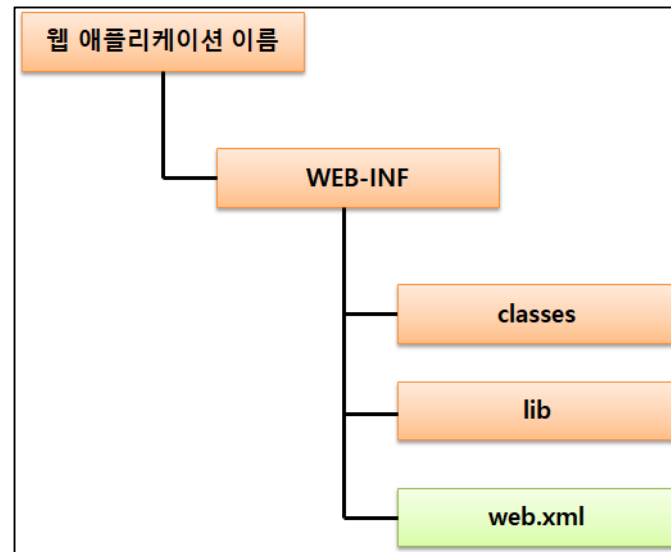


쇼핑몰, 구인 구직 사이트 ...

4.2 웹 애플리케이션의 기본 구조

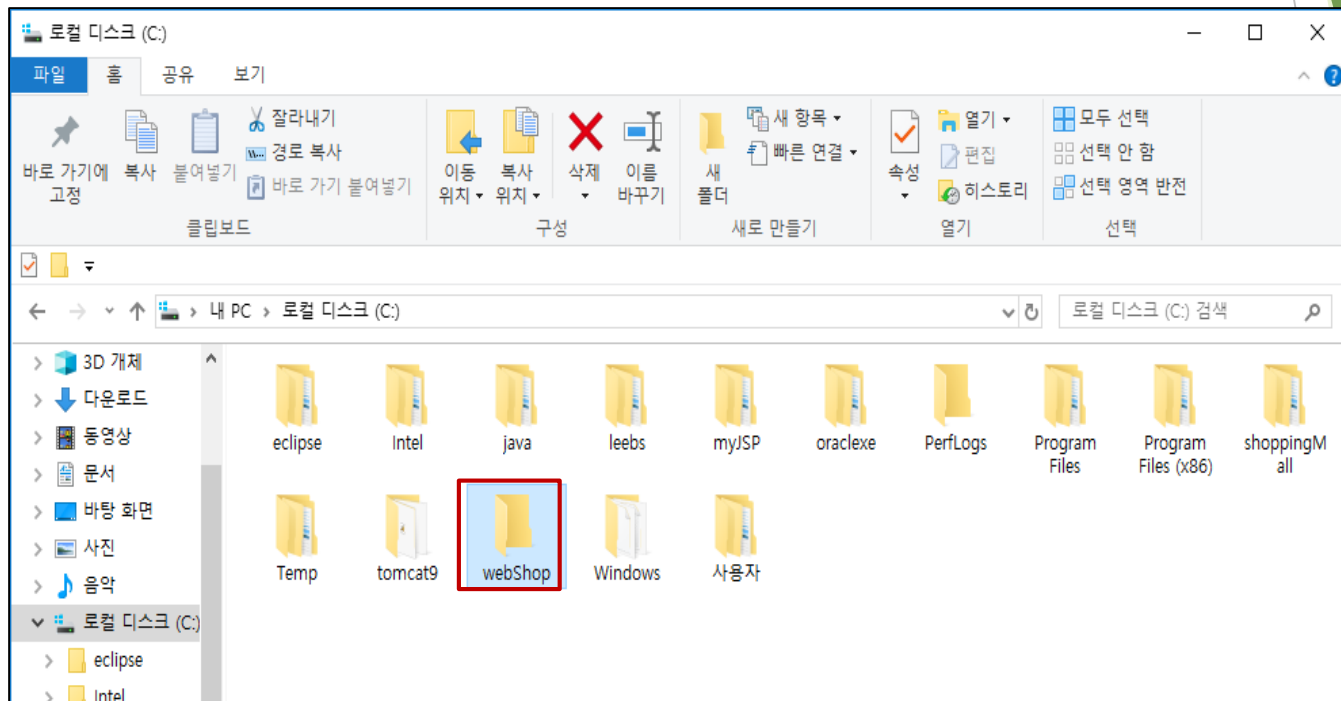
- 웹 애플리케이션 기본 구조

➤ 톰캣 같은 컨테이너에서 실행되는 웹 애플리케이션은 일정한 디렉토리 구조를 갖추어야 함.



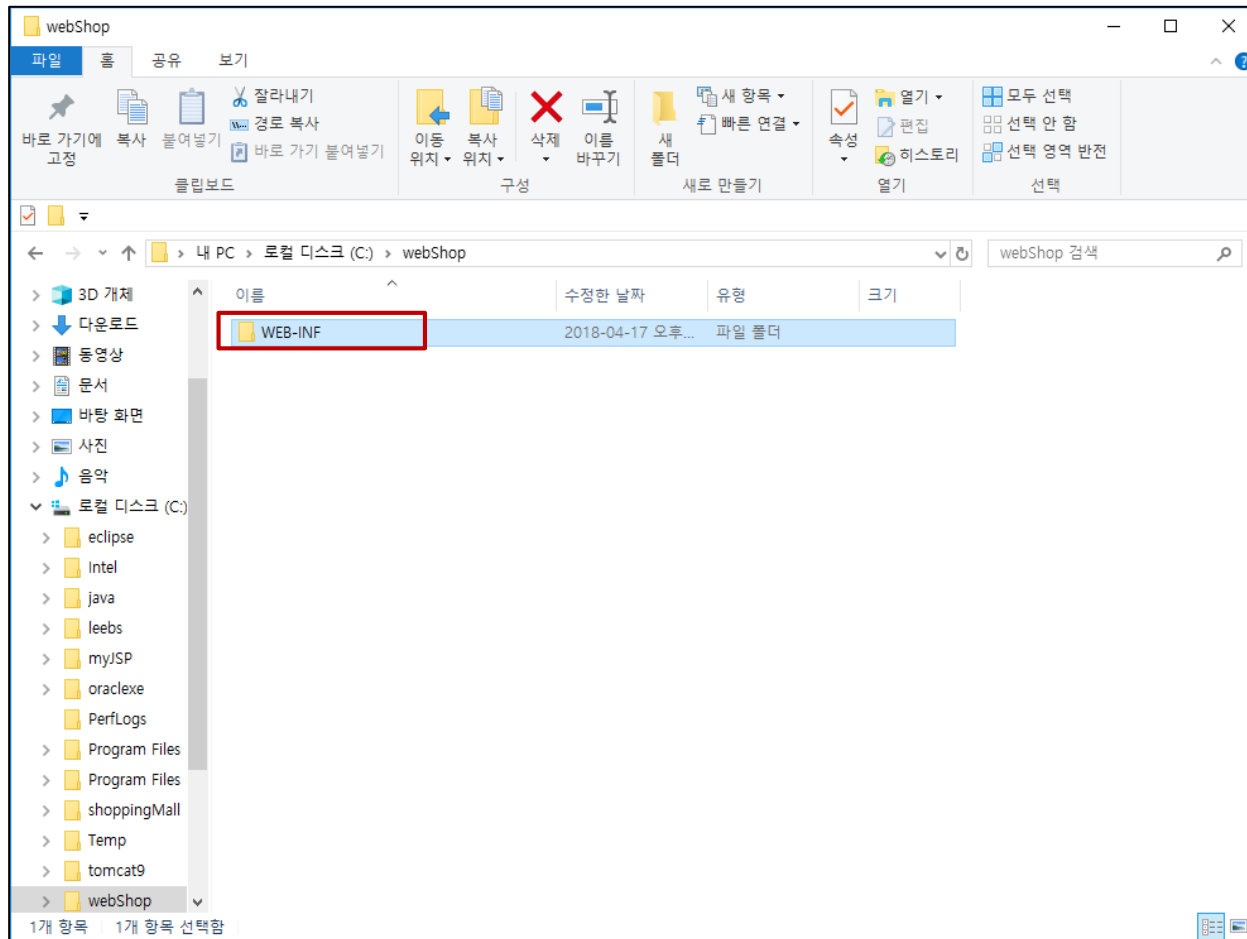
4.2 웹 애플리케이션의 기본 구조

- 윈도 탐색기로 웹 애플리케이션 만들기
 - 윈도 탐색기로 C 드라이브에 webShop 폴더를 생성합니다.



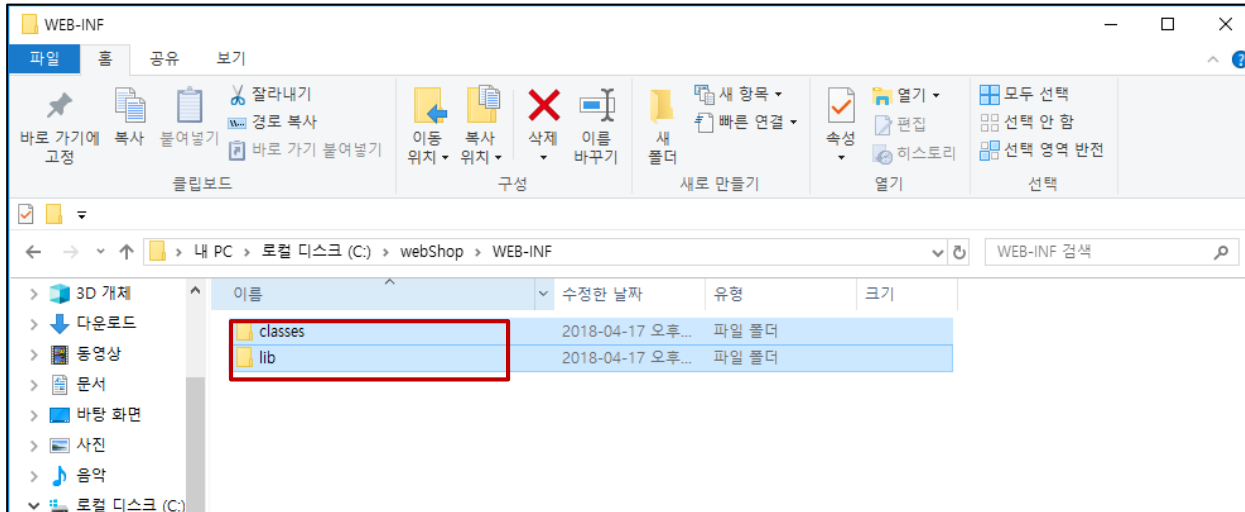
4.2 웹 애플리케이션의 기본 구조

2. webShop 폴더 안에 WEB-INF 폴더를 생성합니다



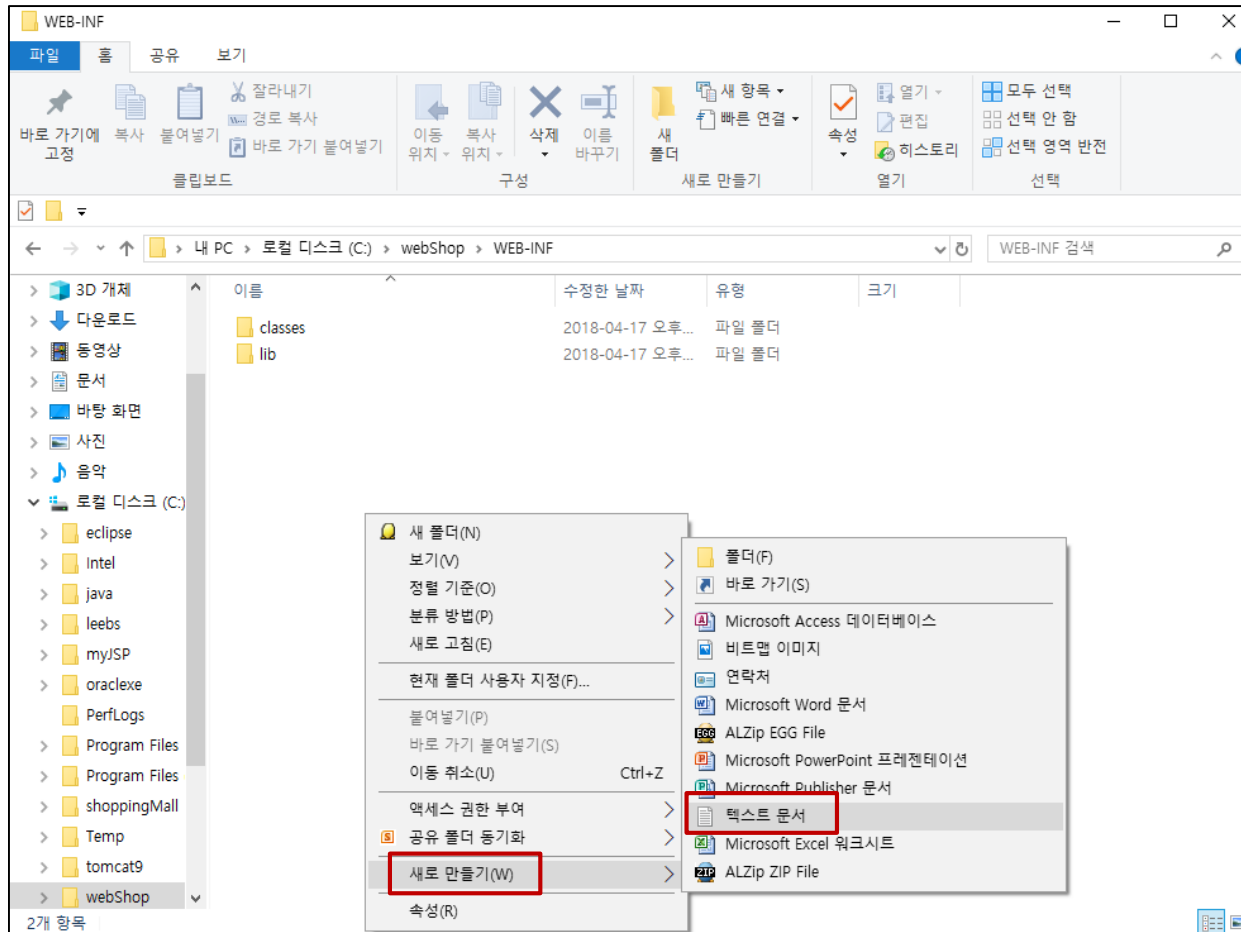
4.2 웹 애플리케이션의 기본 구조

3. WEB-INF 폴더에 classes와 lib 폴더를 각각 생성합니다.

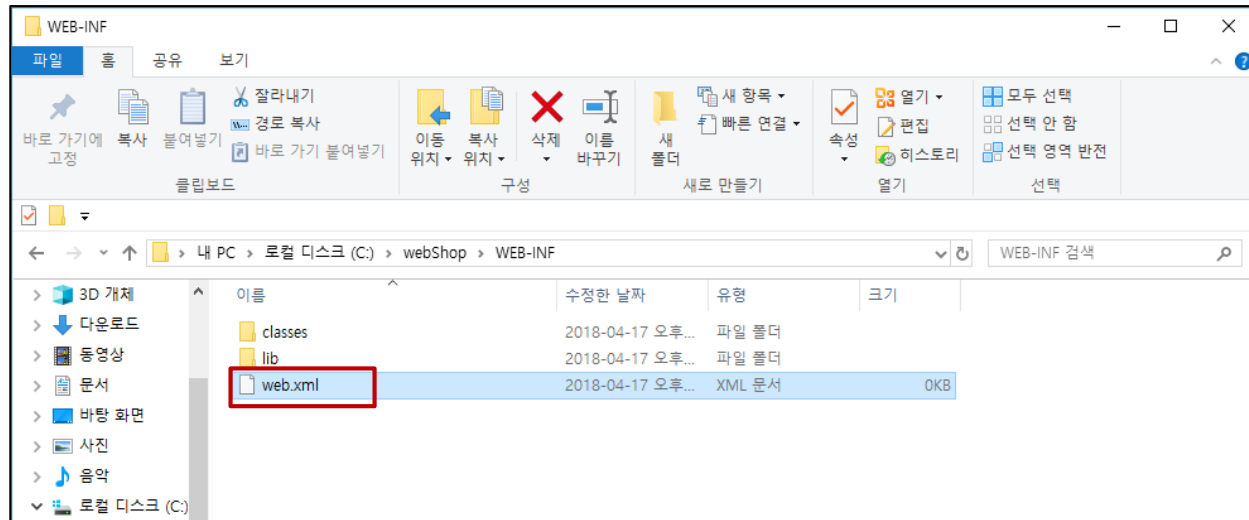


4.2 웹 애플리케이션의 기본 구조

4. WEB-INF 폴더를 선택하고 마우스 오른쪽 버튼을 클릭한 후 새로 만들기 > 텍스트 문서를 선택해 텍스트 파일을 생성하고 이름을 web.xml로 변경합니다.



4.2 웹 애플리케이션의 기본 구조



4.2 웹 애플리케이션의 기본 구조

5. VS Code로 web.xml 파일을 열고 다음 코드를 작성합니다. 단, 오타 방지를 위해 이 책과 함께 제공하는 파일에서 복사하여 붙여 넣으세요.

```
web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://java.sun.com/xml/ns/javaee"
4         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
5         id="WebApp_ID" version="3.0">
6
7 </web-app>
```

코드 pro04/web.xml

~~webShop/WebContent/WEB-INF/web.xml~~

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
</web-app>
```

❖ Note

톰캣 8 버전까지는 이 코드를 추가하지 않아도 자동으로 생성이 되었습니다. 하지만 톰캣 9 버전부터는 xml을 추가해야 합니다.

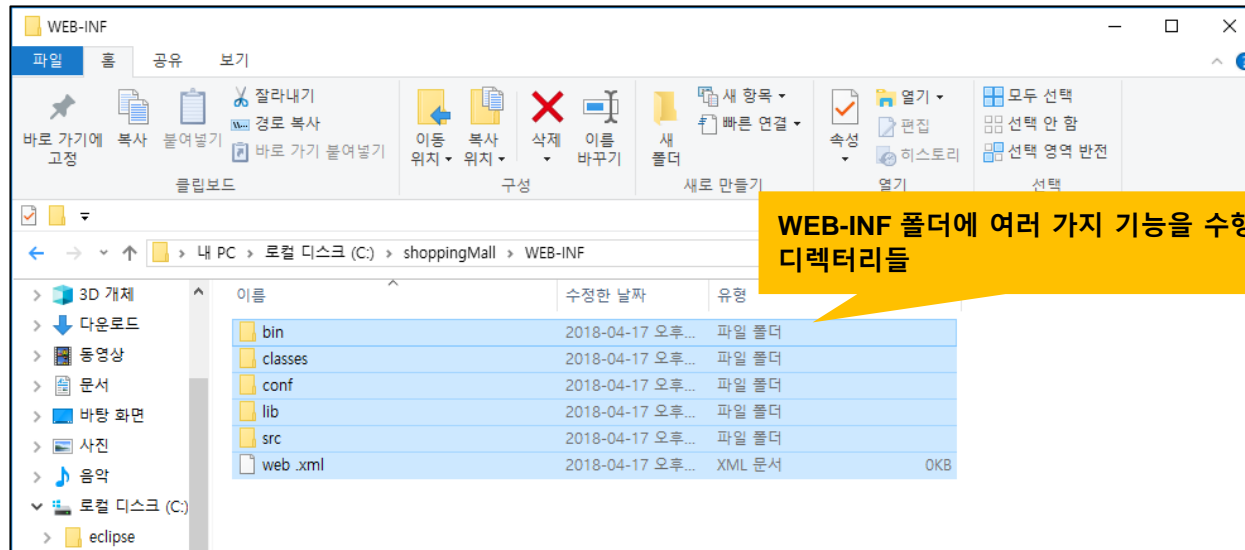
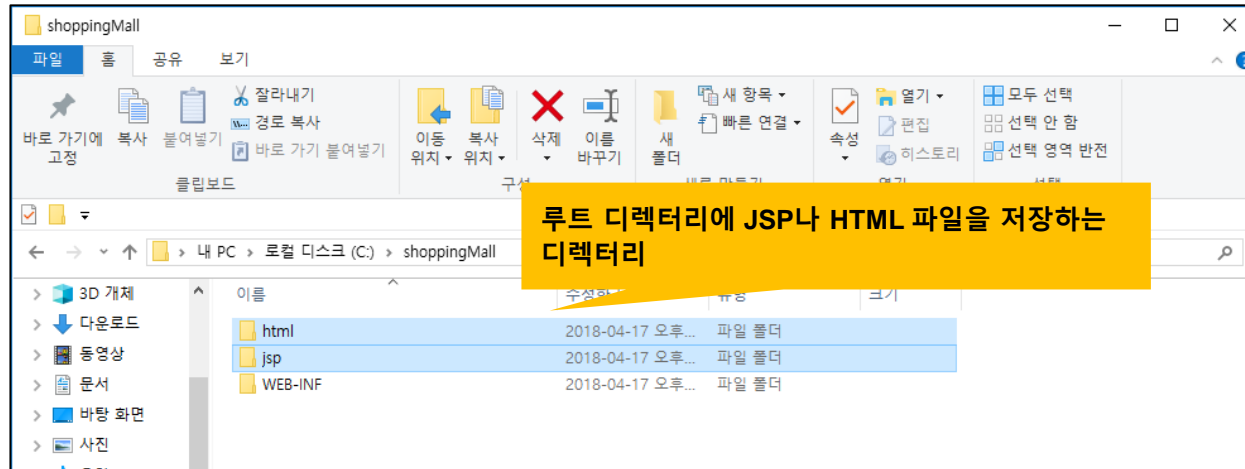
4.2 웹 애플리케이션의 기본 구조

웹 애플리케이션 구성 요소의 기능

구성 요소	기능
webShop	웹 애플리케이션의 루트 디렉터리. 다른 웹 애플리케이션 이름과 중복을 허용하지 않으며, 여기에는 JSP HTML 파일이 저장됩니다.
WEB-INF	웹 애플리케이션에 관한 정보가 저장되는 곳입니다. 이 디렉터리는 외부에서 접근할 수 없습니다.
classes	웹 애플리케이션이 수행하는 서블릿과 다른 일반 클래스들이 위치하는 곳입니다.
lib	웹 애플리케이션에서 사용되는 여러 가지 라이브러리 압축 파일(jar 파일)이 저장되는 곳입니다. DB 연동 드라이버나 프레임워크 기능 관련 jar 파일이 여기에 저장됩니다. lib 디렉터리의 jar는 클래스 패스가 자동으로 설정됩니다.
web.xml	배치 지시자(deployment descriptor)로서 일종의 환경 설정 파일입니다. 웹 애플리케이션에 대한 여러 가지 설정을 할 때 사용됩니다.

4.2 웹 애플리케이션의 기본 구조

실제 웹 애플리케이션 디렉터리 구조



4.2 웹 애플리케이션의 기본 구조

웹 애플리케이션에 추가된 구성 요소의 기능

구성 요소	기능
jsp/html	JSP 파일과 HTML 파일이 저장된 곳입니다
css	스타일시트 파일이 저장된 곳입니다
image	웹 애플리케이션에서 사용되는 이미지가 저장된 곳입니다
js	자바스크립트 파일이 저장된 곳입니다.
bin	애플리케이션에서 사용되는 각종 실행 파일이 저장된 곳입니다.
conf	프레임워크에서 사용하는 각종 설정 파일이 저장된 곳입니다.
src	자바 소스 파일이 저장된 곳입니다.

4.3 컨테이너에서 웹 애플리케이션 실행하기

• 4.3.1 컨테이너에 웹 애플리케이션 등록

- 웹 애플리케이션은 일반 PC에서 실행되는 자바 애플리케이션과 달리 단독으로 실행할 수 없음
- 웹 애플리케이션은 톰캣과 같은 웹 컨테이너에서 실행되는 구조임.
- 웹 애플리케이션을 실행하려면 우선 톰캣에 등록부터 해야함.

톰캣 컨테이너에 웹 애플리케이션 등록방법

- ① %CATALINA_HOME%webApp 디렉터리에 애플리케이션을 저장
- ② server.xml에 직접 웹 애플리케이션을 등록

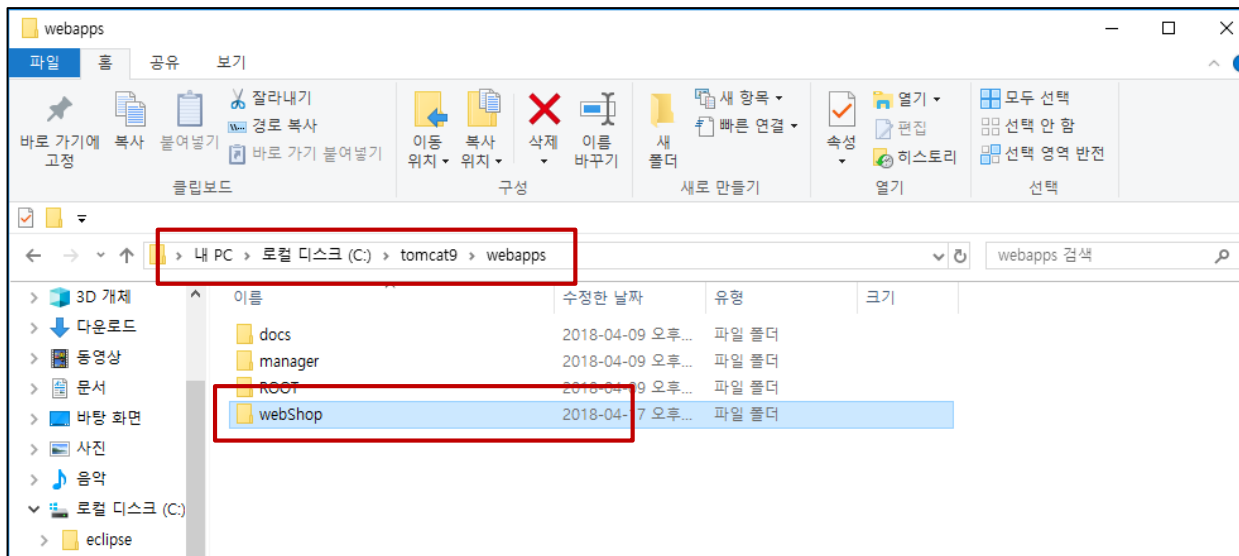
❖ Note

JDK 경로를 환경 변수에 등록하면서 JDK 루트 디렉터리를 JAVA_HOME이라고 불렀습니다. 톰캣의 루트 디렉터리는 보통 CATALINA_HOME으로 묵시적으로 사용하고 있습니다.

4.3 컨테이너에서 웹 애플리케이션 실행하기

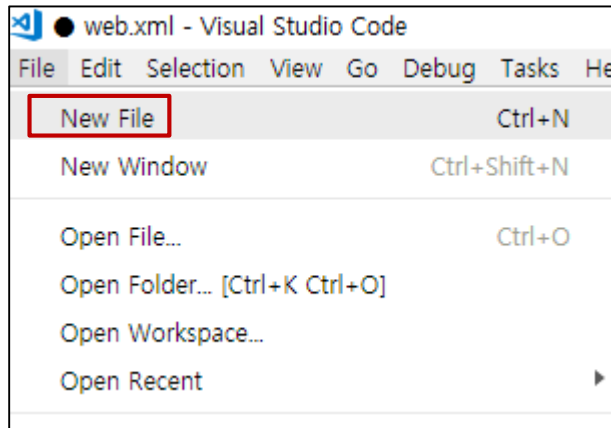
- **%CATALINA_HOME%webApp 디렉터리에 애플리케이션을 저장하는 방법**

1. C 드라이브의 webShop 폴더 전체를 복사한 후 톰캣 루트 디렉터리의 하위에 있는 webapps폴더에 붙여 넣습니다.



4.3 컨테이너에서 웹 애플리케이션 실행하기

2. VS Code를 실행한 후 상단 메뉴의 File > New File을 선택합니다.

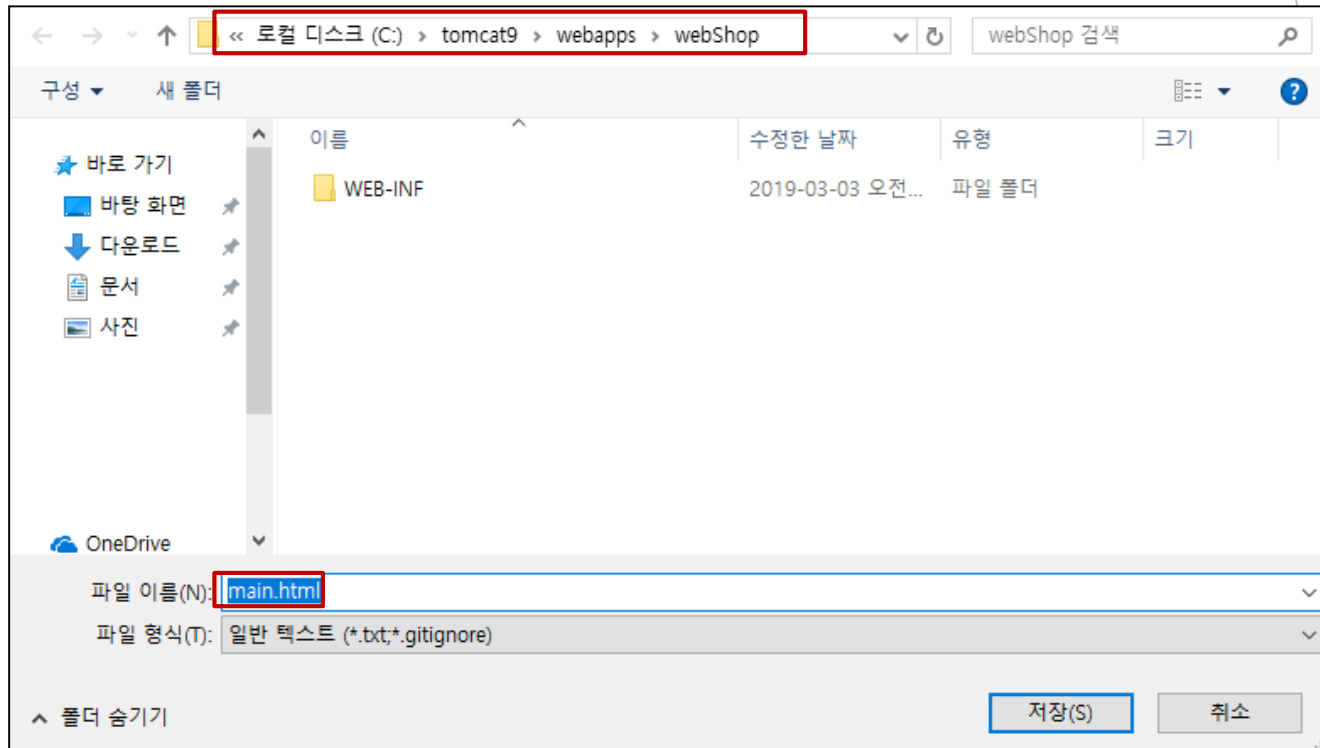


3. 다음과 같이 HTML 코드를 입력합니다.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>Hello JSP!</title>
</head>
<body>
  Hello JSP!!<br>
  안녕하세요!!
</body>
</html>
```

4.3 컨테이너에서 웹 애플리케이션 실행하기

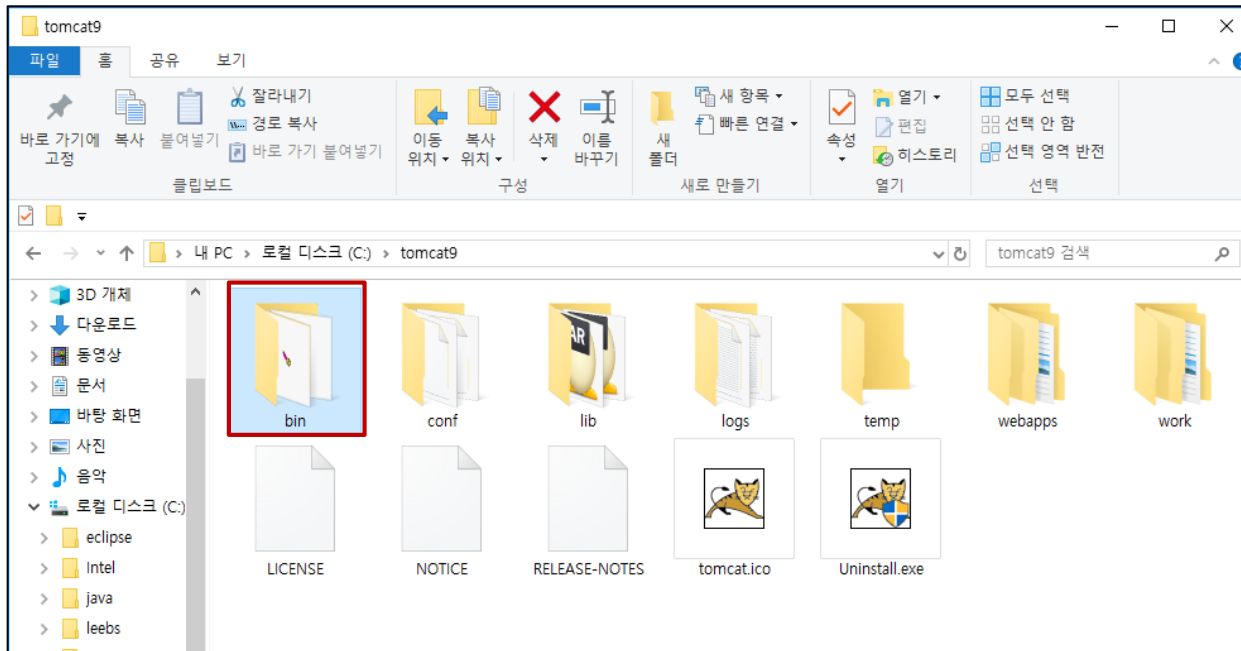
4. 상단 메뉴의 File > Save를 선택하고 파일 이름을 main.html로 수정한 후 webShop 폴더에 저장합니다.



4.3 컨테이너에서 웹 애플리케이션 실행하기

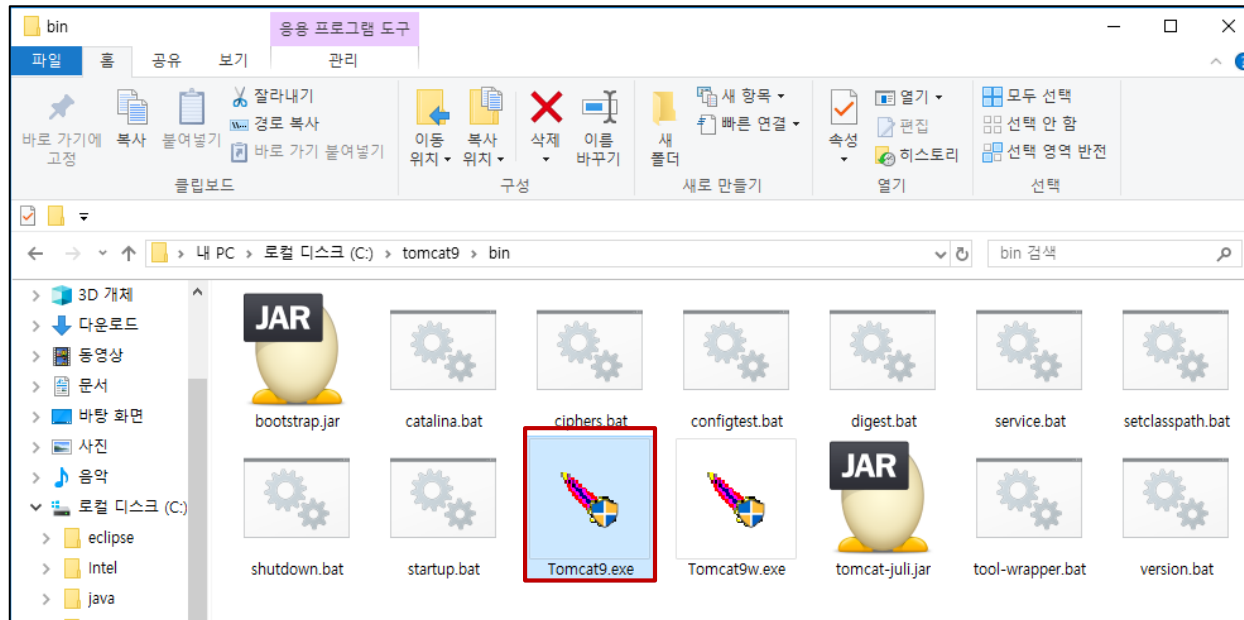
• 4.3.2 톰캣 실행하기

1. 톰캣을 설치한 루트 디렉터리의 bin 폴더로 이동합니다.



4.3 컨테이너에서 웹 애플리케이션 실행하기

2. bin 디렉터리의 Tomcat9.exe를 더블클릭해 실행합니다.



4.3 컨테이너에서 웹 애플리케이션 실행하기

3. 콘솔창이 나타나면서 톰캣 컨테이너가 실행됩니다.



```
C:\tomcat9\bin\Tomcat9.exe
22-Aug-2018 20:10:08.246 정보 [main] org.apache.catalina.startup.Catalina.load Initialization processed in 1672 ms
22-Aug-2018 20:10:08.375 정보 [main] org.apache.catalina.core.StandardService.startInternal Starting service [Catalina]
22-Aug-2018 20:10:08.376 정보 [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet Engine: Apache Tomcat/9.0.6
22-Aug-2018 20:10:09.486 경고 [main] org.apache.catalina.util.SessionIdGeneratorBase.createSecureRandom Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [329] milliseconds.
22-Aug-2018 20:10:09.536 정보 [main] org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive [C:\tomcat9\webapps\webShop.war]
22-Aug-2018 20:10:09.627 정보 [main] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive [C:\tomcat9\webapps\webShop.war] has finished in [91] ms
22-Aug-2018 20:10:09.629 정보 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\tomcat9\webapps\docs]
22-Aug-2018 20:10:09.692 정보 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\tomcat9\webapps\docs] has finished in [63] ms
22-Aug-2018 20:10:09.693 정보 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\tomcat9\webapps\manager]
22-Aug-2018 20:10:09.801 정보 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\tomcat9\webapps\manager] has finished in [108] ms
22-Aug-2018 20:10:09.804 정보 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\tomcat9\webapps\ROOT]
22-Aug-2018 20:10:09.860 정보 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\tomcat9\webapps\ROOT] has finished in [58] ms
22-Aug-2018 20:10:09.868 정보 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8090"]
22-Aug-2018 20:10:09.883 정보 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
22-Aug-2018 20:10:09.892 정보 [main] org.apache.catalina.startup.Catalina.start Server startup in 1645 ms
```

4.3 컨테이너에서 웹 애플리케이션 실행하기

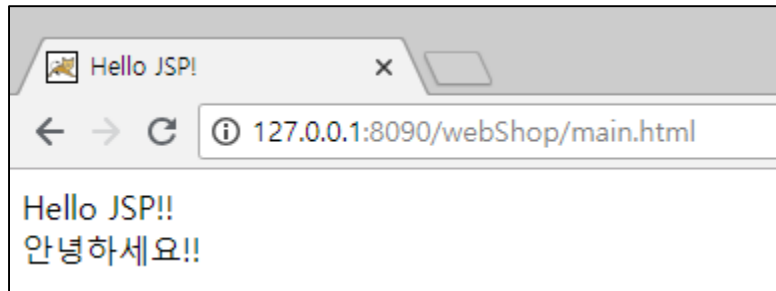
- 4.3.3 브라우저에서 웹 애플리케이션 요청

웹 브라우저에서 웹 애플리케이션을 요청하는 방법

`http://IP주소:포트번호/컨텍스트이름/요청파일이름`



`http://127.0.0.1:8090/webShop/main.html`



❖ Tip

현재는 톰캣이 로컬 PC에 설치되어 있기 때문에 127.0.0.1 대신 localhost라고 입력해도 됩니다.

4.3 컨테이너에서 웹 애플리케이션 실행하기

• 4.3.4 컨텍스트란?

- 톰캣의 `server.xml`에 등록하는 웹 애플리케이션을 컨텍스트(Context)라 함.
- 즉, 톰캣 입장에서 인식하는 한 개의 웹 애플리케이션임.

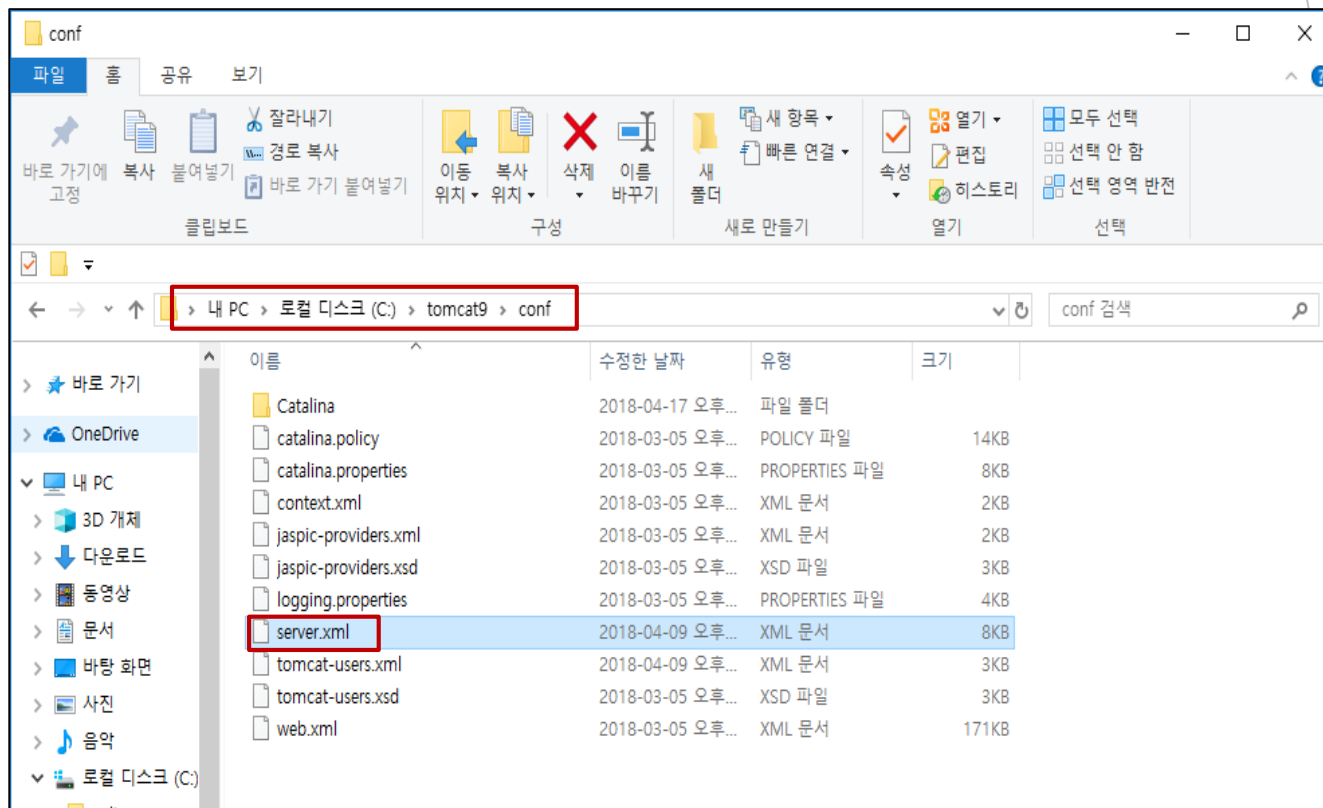
컨텍스트의 특징

- 웹 애플리케이션당 하나의 컨텍스트가 등록됨
- 웹 애플리케이션 이름과 같을 수도 있고 다를 수도 있음
- 컨텍스트 이름은 중복되면 안됨
- 웹 애플리케이션의 의미를 가장 잘 나타낼 수 있는 명사형으로 지정
- 대소문자 구분
- `server.xml`에 등록

4.3 컨테이너에서 웹 애플리케이션 실행하기

• 4.3.5 톰캣 컨테이너에 컨텍스트 등록하기

톰캣 설치 루트 디렉터리의 conf 디렉터리의 server.xml을 설정합니다.



4.3 컨테이너에서 웹 애플리케이션 실행하기

<Context> 태그란?

- 톰캣은 모든 설정 정보를 XML로 저장한 후 실행 시 정보를 읽어와 설정대로 실행함.
- 우리가 만든 웹 애플리케이션도 미리 <Context> 태그를 이용해서 server.xml에 등록해 두어야 톰캣이 설정한 대로 웹 애플리케이션을 실행함.

<Context> 태그 구성요소의 기능

구성요소	기능
path	웹 애플리케이션의 컨텍스트 이름입니다. 웹 애플리케이션 이름과 다를 수도 있으며, 웹 브라우저에서 실제 웹 애플리케이션을 요청하는 이름입니다.
docBase	컨텍스트에 대한 실제 웹 애플리케이션이 위치한 경로입니다. WEB-INF 상위 폴더까지의 경로를 나타냅니다.
reloadable	실행 중 소스 코드가 수정될 경우 바로 갱신할지를 설정합니다. 만약 false로 설정하면 톰캣을 다시 실행해야 추가한 소스 코드의 기능이 반영됩니다.

4.3 컨테이너에서 웹 애플리케이션 실행하기

server.xml에 <Context> 태그를 이용해서 등록

```
<Context path="/컨텍스트 이름"  
         docBase="실제 웹 애플리케이션의 WEB-INF 디렉터리 위치"  
         reloadable="true 또는 false" />
```



```
<Context path="/webMal"  
         docBase="C:\www\webShop"  
         reloadable="true" />
```

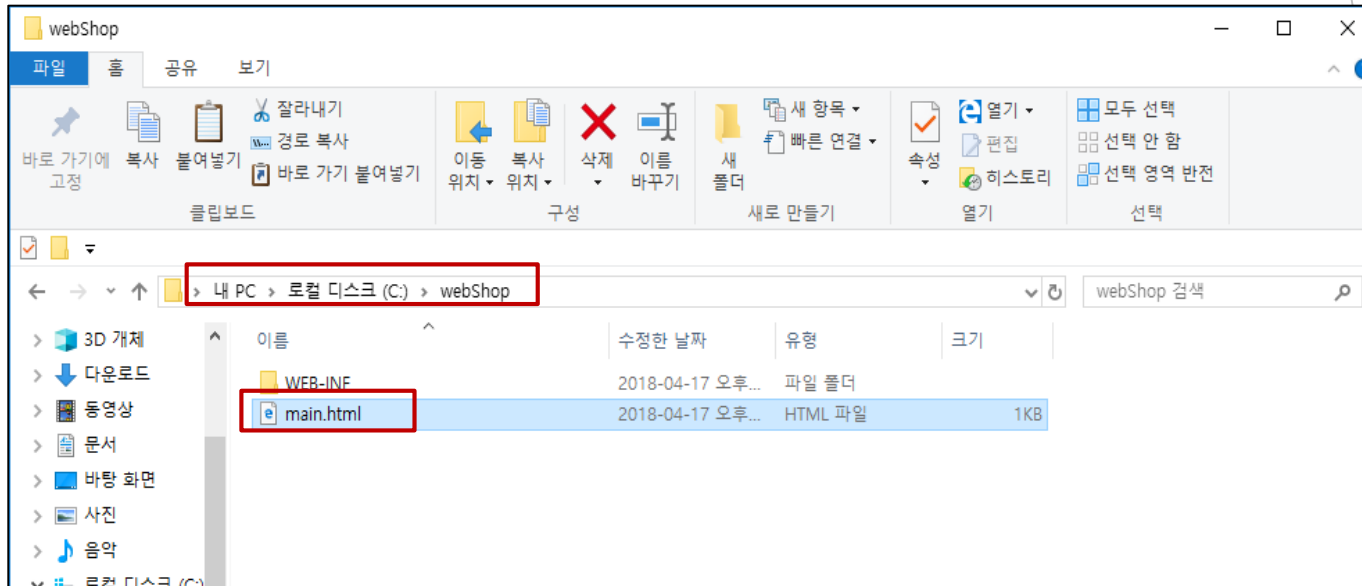

4.3 컨테이너에서 웹 애플리케이션 실행하기

```
149 <Host name="localhost" appBase="webapps"
150       unpackWARs="true" autoDeploy="true">
151
152     <Context path="/webMal"
153             docBase="C:\webShop"
154             reloadable="true" />
155
156     <!-- SingleSignOn valve, share authentication between web applications
157          Documentation at: /docs/config/valve.html -->
158     <!--
159     <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
160     -->
161
162     <!-- Access log processes all example.
163          Documentation at: /docs/config/valve.html
164          Note: The pattern used is equivalent to using pattern="common" -->
165     <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
166           prefix="localhost_access_log" suffix=".txt"
167           pattern="%h %l %u %t &quot;%r&quot; %s %b" />
168
169 </Host>
```

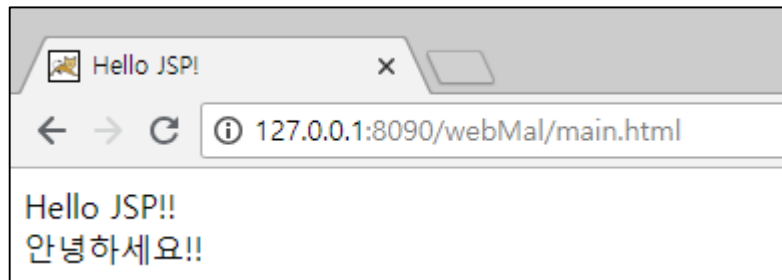
```
<Context path = "/webMal"
         docBase = "C:\webShop"
         reloadable = "true" />
```

4.3 컨테이너에서 웹 애플리케이션 실행하기

첫 번째 톰캣 실행 시 사용한 main.html을 복사해서 C:\webShop 아래에 복사



브라우저에서 <http://127.0.0.1:8090/webMal/main.html>로 요청



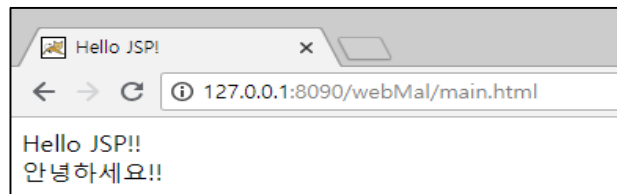
4.3 컨테이너에서 웹 애플리케이션 실행하기

- 4.3.6 톰캣 컨테이너에서의 웹 애플리케이션 동작 과정

<http://localhost:8090/webMal/main.html>

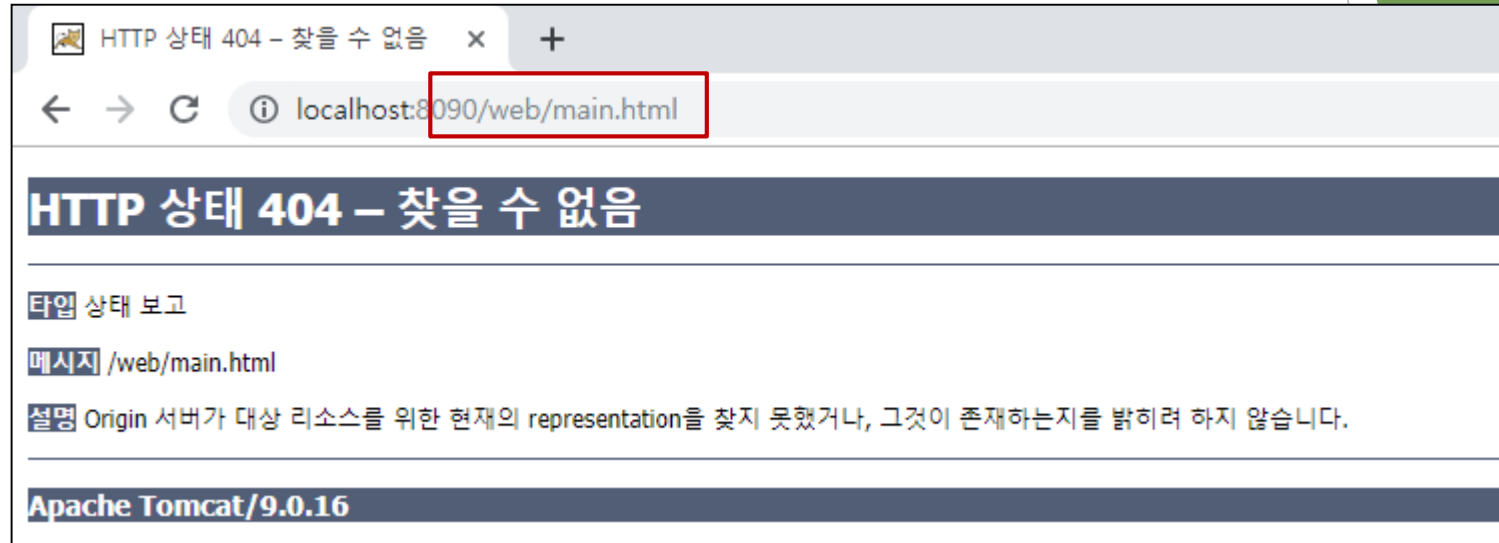


- ① 웹 브라우저에서 컨텍스트 이름(webMal)으로 요청
- ② 요청을 받은 톰캣 컨테이너는 요청한 컨텍스트 이름이 server.xml에 있는지 확인
- ③ 해당 컨텍스트 이름이 있으면 컨텍스트 이름에 대한 실제 웹 애플리케이션이 있는 경로(C:\webShop)로 가서 요청한 main.html을 클라이언트 웹 브라우저로 전송
- ④ 웹 브라우저는 전송된 main.html을 브라우저에 표시



4.3 컨테이너에서 웹 애플리케이션 실행하기

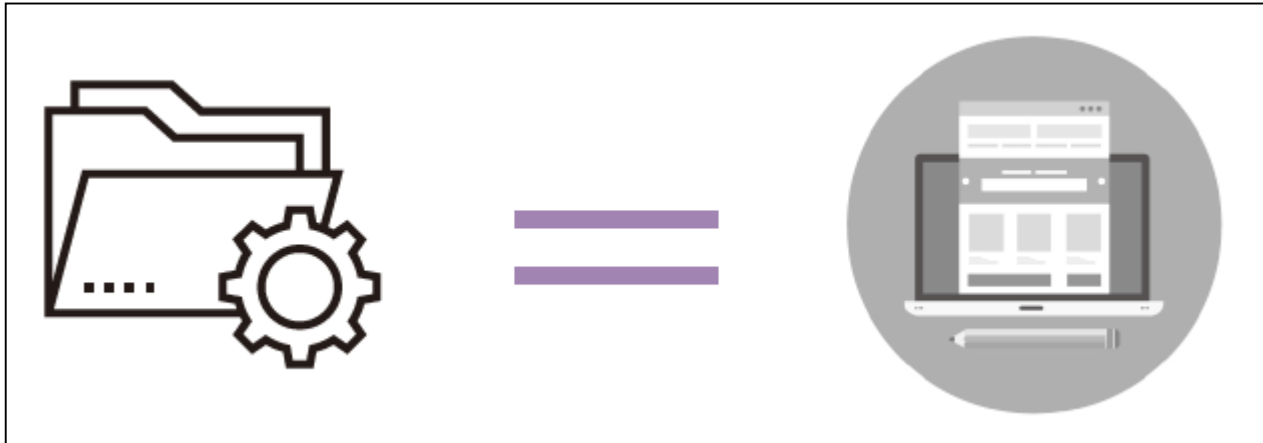
- 등록되지 않은 컨텍스트 이름으로 요청시 오류 발생



4.4 이클립스에서 웹 애플리케이션 실습하기

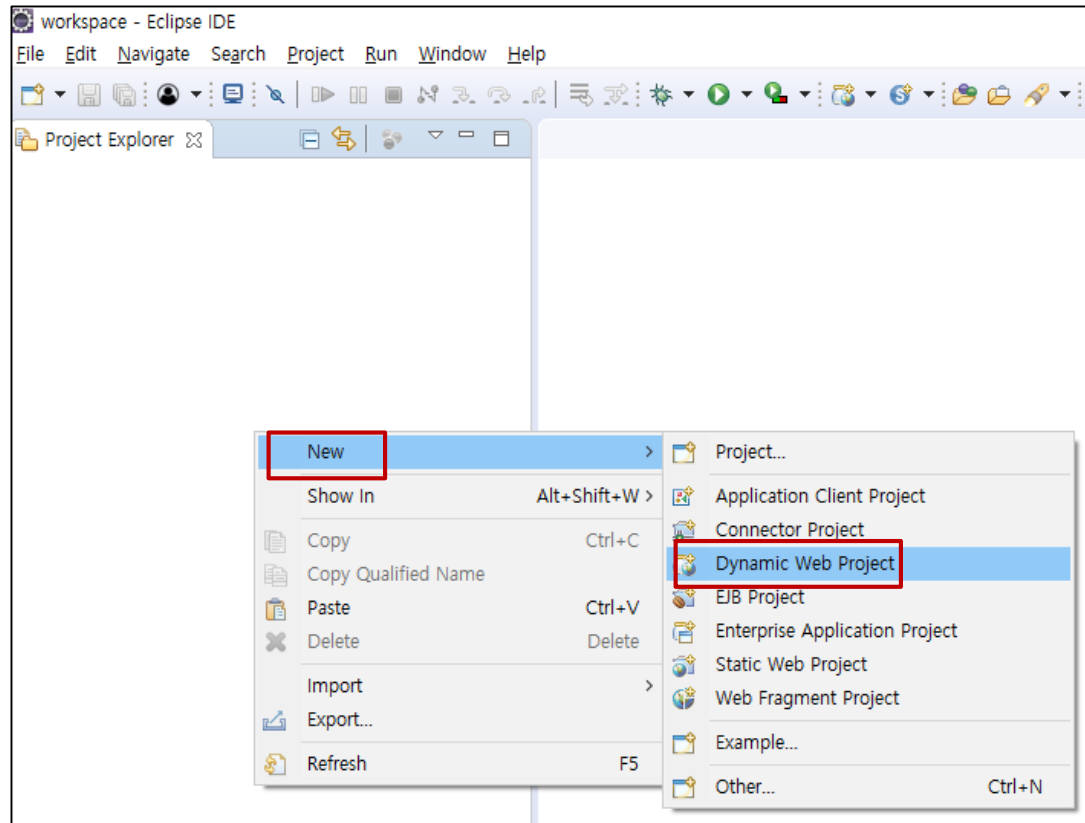
• 4.4.1 이클립스에서 웹 프로젝트 생성

- 이클립스에서는 한 개의 프로젝트가 한 개의 웹 애플리케이션임
- 프로젝트 이름이 바로 웹 애플리케이션 이름임



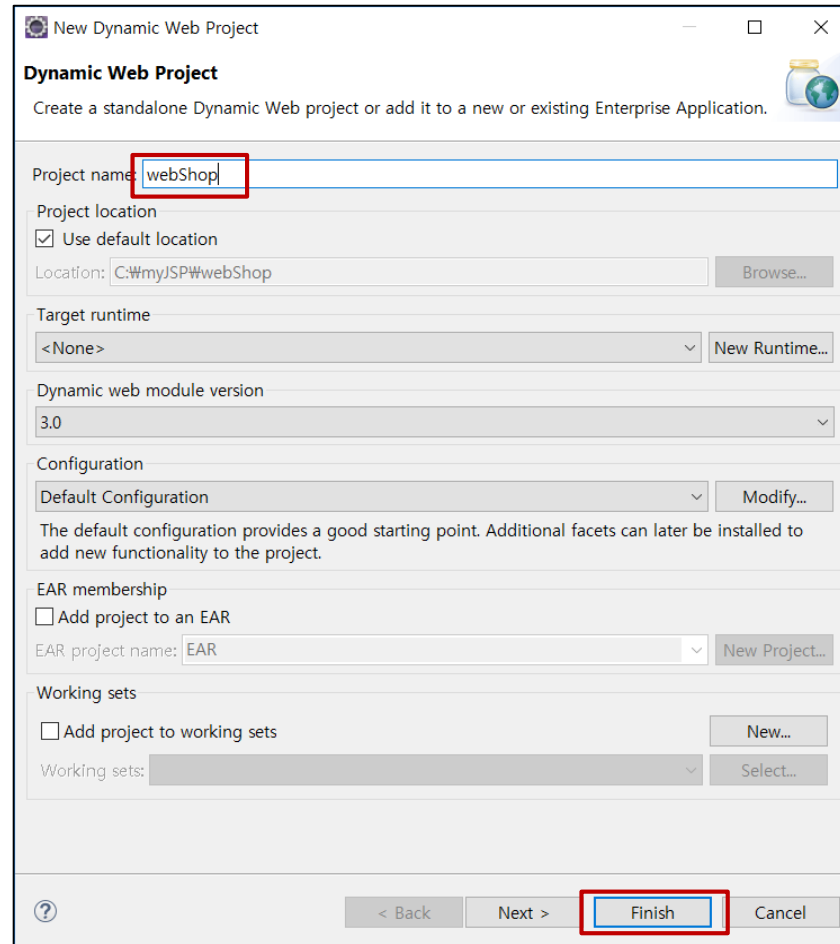
4.4 이클립스에서 웹 애플리케이션 실습하기

1. 이클립스를 열고 Project Explorer 영역에서 마우스 오른쪽 버튼을 클릭한 후 New > Dynamic Web Project를 선택합니다.



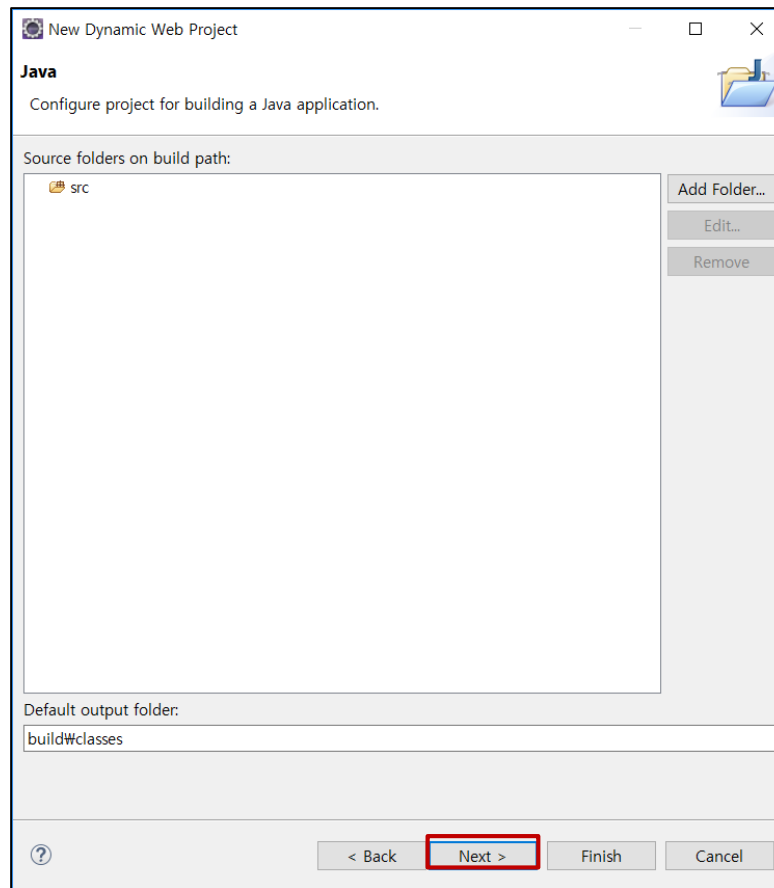
4.4 이클립스에서 웹 애플리케이션 실습하기

2. 프로젝트 이름에 webShop이라고 입력한 후 Next를 클릭합니다.



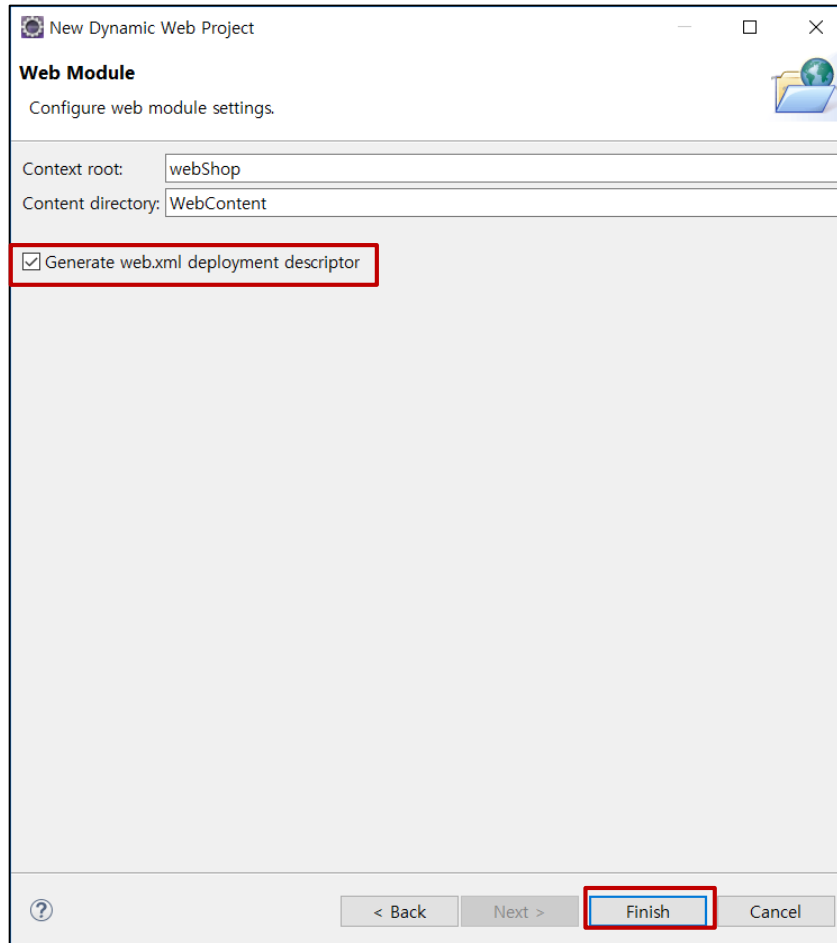
4.4 이클립스에서 웹 애플리케이션 실습하기

3. 이클립스에서 지정한 기본값 그대로 두고 Next를 클릭합니다.



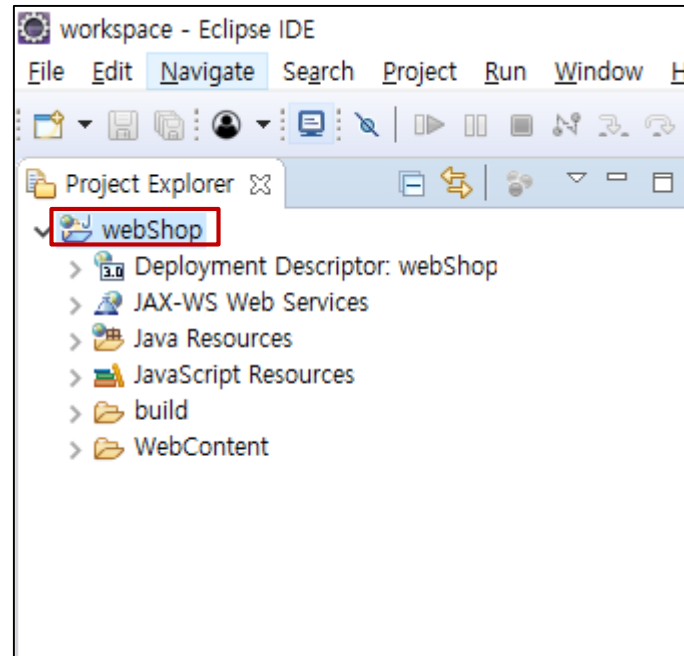
4.4 이클립스에서 웹 애플리케이션 실습하기

4. web.xml을 생성할 것인지 묻는 체크박스에 체크한 후 Finish를 클릭합니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

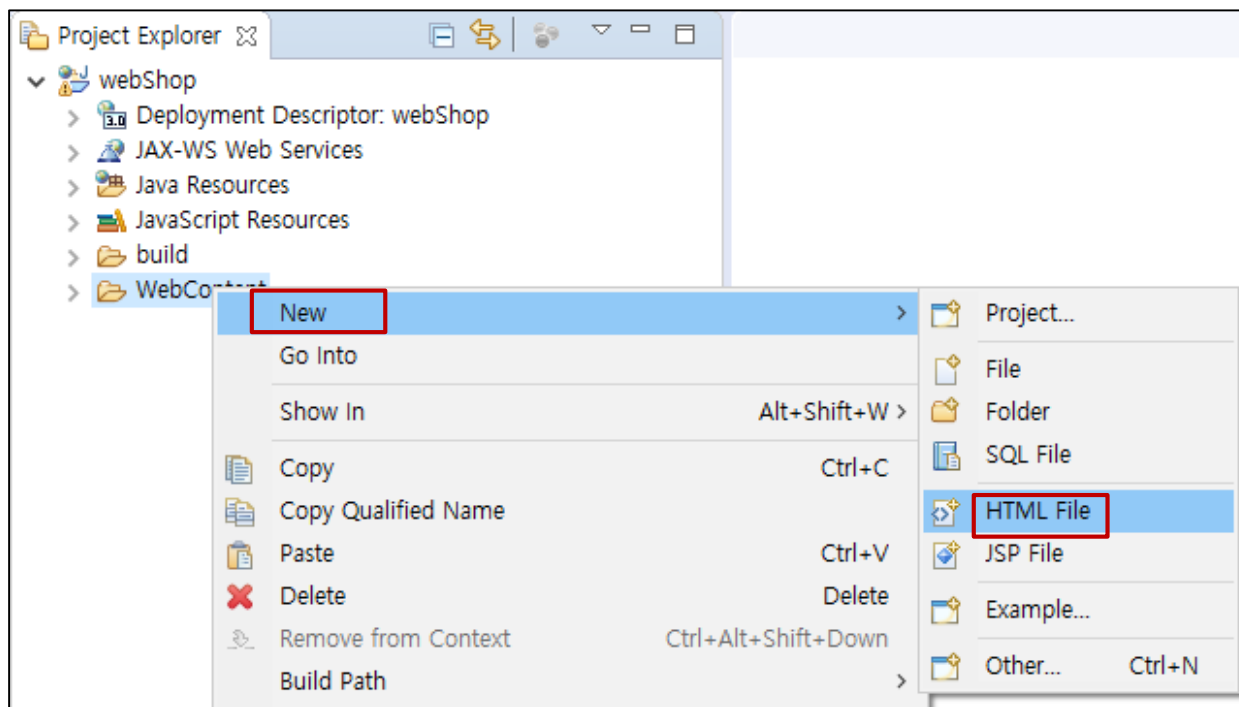
5. Project Explorer에 webShop 프로젝트가 생성된 것을 확인할 수 있습니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

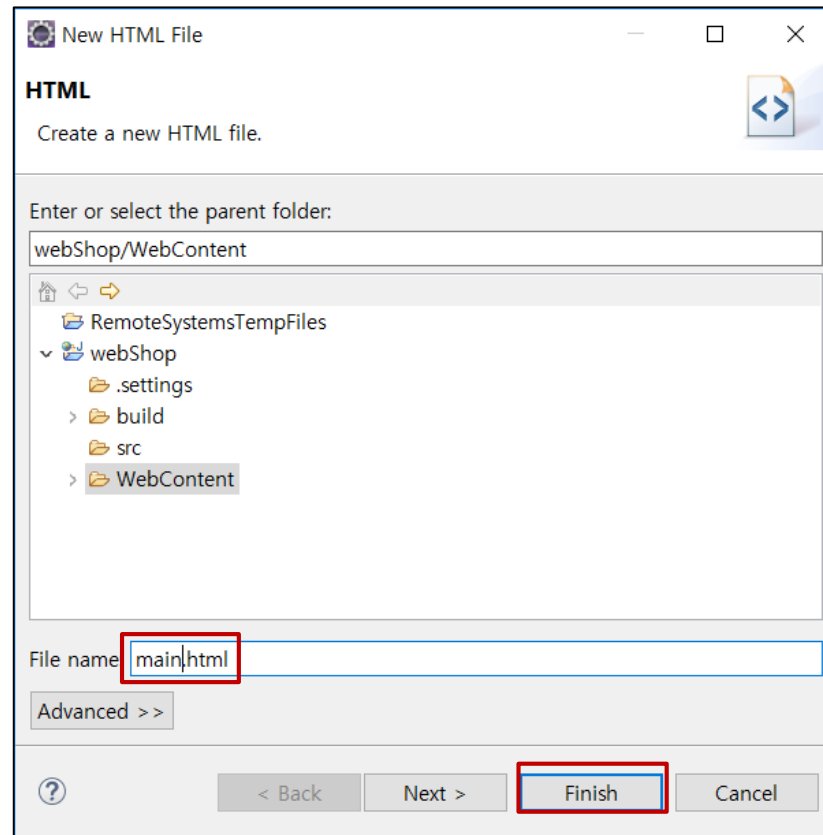
• 4.4.2 이클립스에서 HTML 파일 생성

1. 프로젝트 하위 메뉴에서 WebContent를 선택하고 마우스 오른쪽 버튼을 클릭한 후 New >HTML File을 선택합니다.



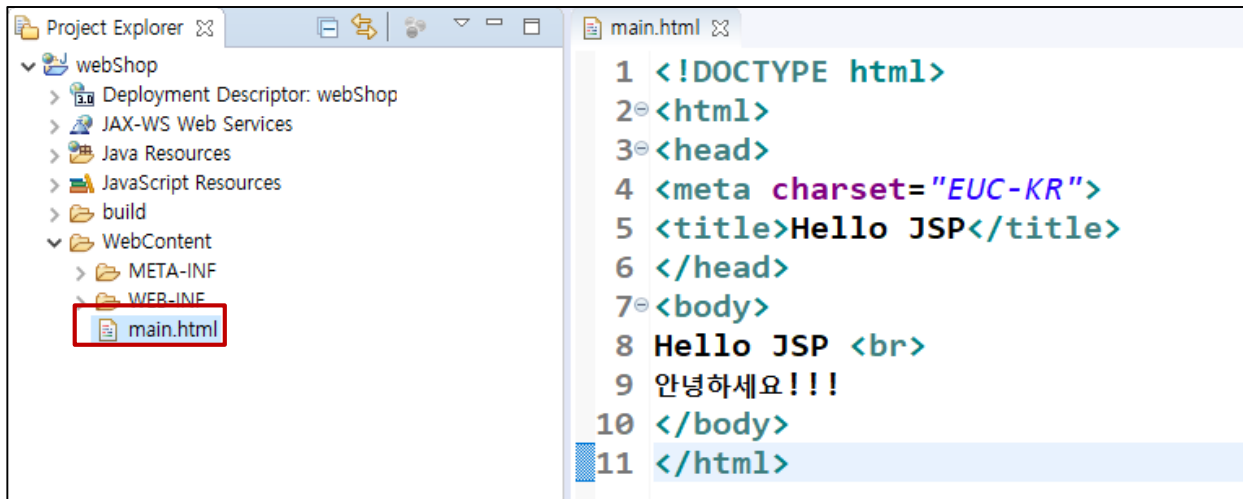
4.4 이클립스에서 웹 애플리케이션 실습하기

2. 파일 이름으로 main.html을 입력한 후 Finish를 클릭합니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

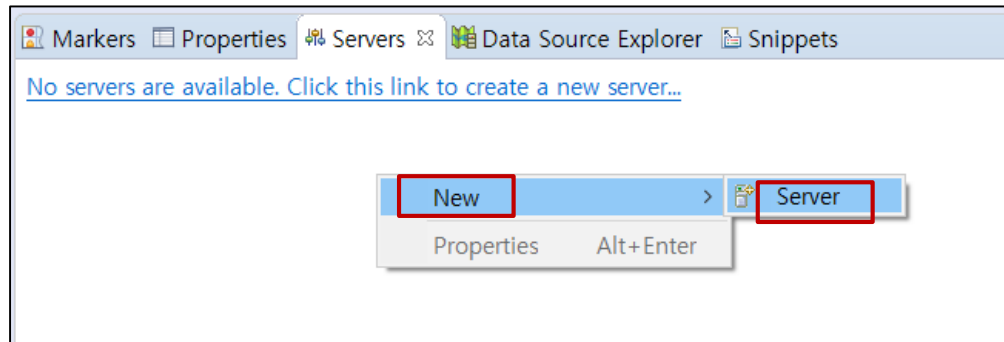
3. main.html 파일에 다음과 같이 간단한 HTML 코드를 작성한 후 CTRL+S를 눌러 저장합니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

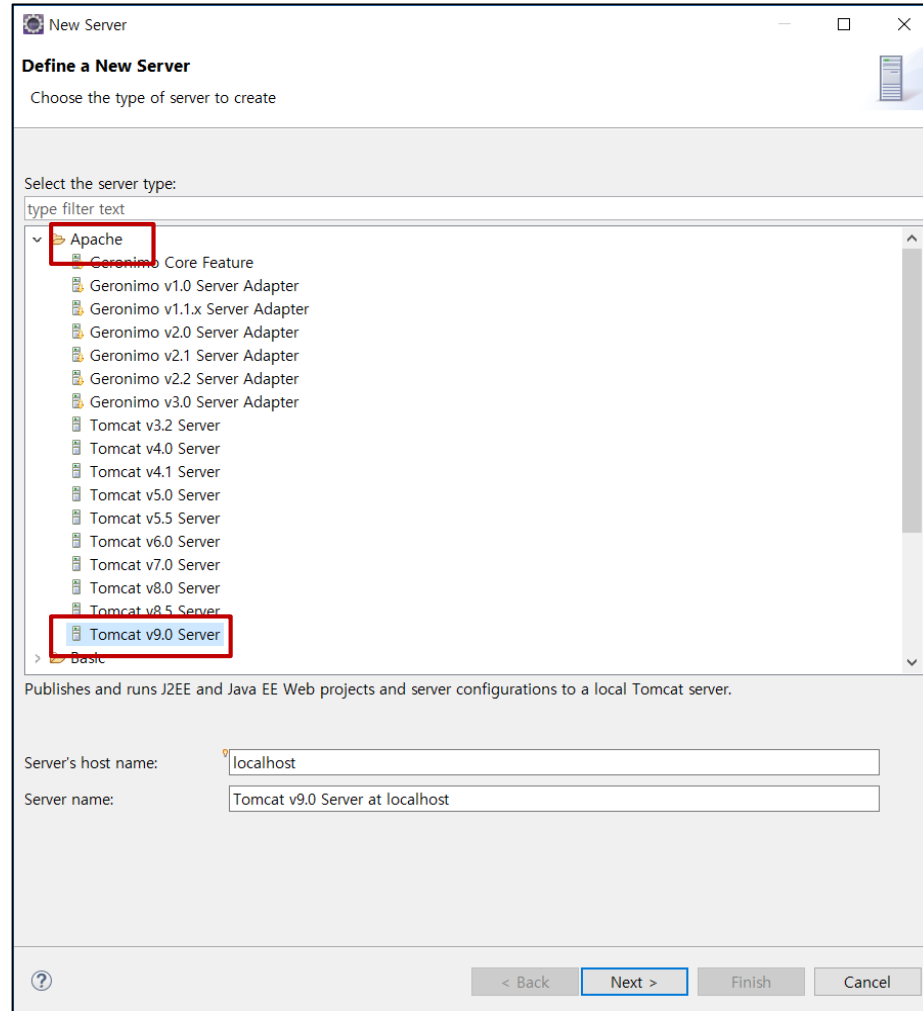
- 4.4.3 이클립스와 톰캣 연동

1. 이클립스 하단의 Servers 탭을 선택하고 마우스 오른쪽 버튼을 클릭한 후 New > Server를 선택합니다.



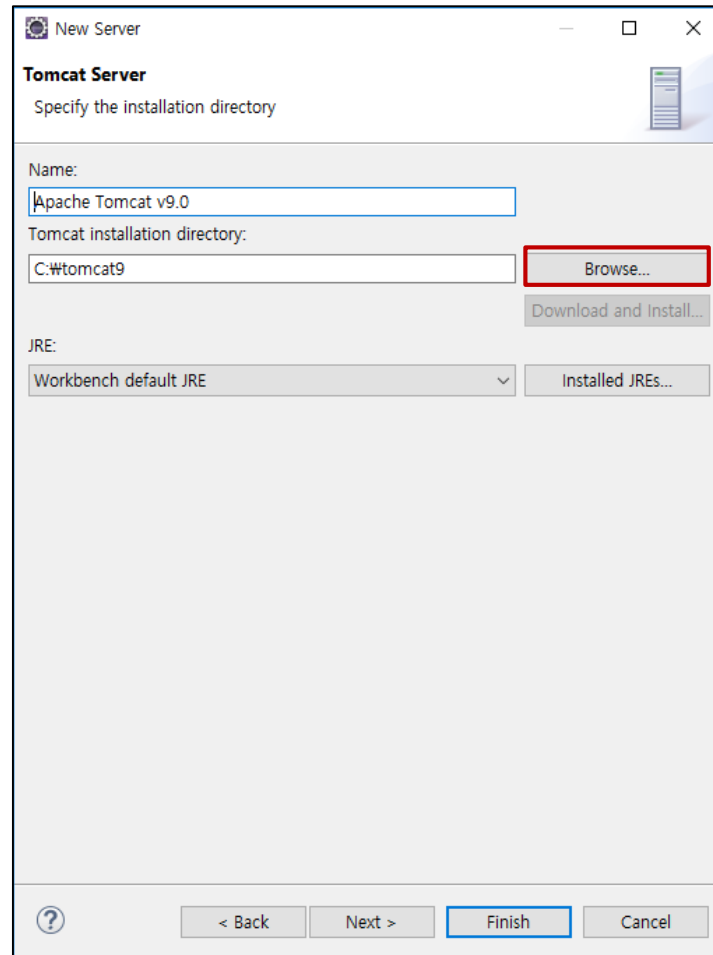
4.4 이클립스에서 웹 애플리케이션 실습하기

2. 서버 설정창의 Apache 항목에서 Tomcat v9.0 Server를 선택한 후 Next를 클릭합니다.

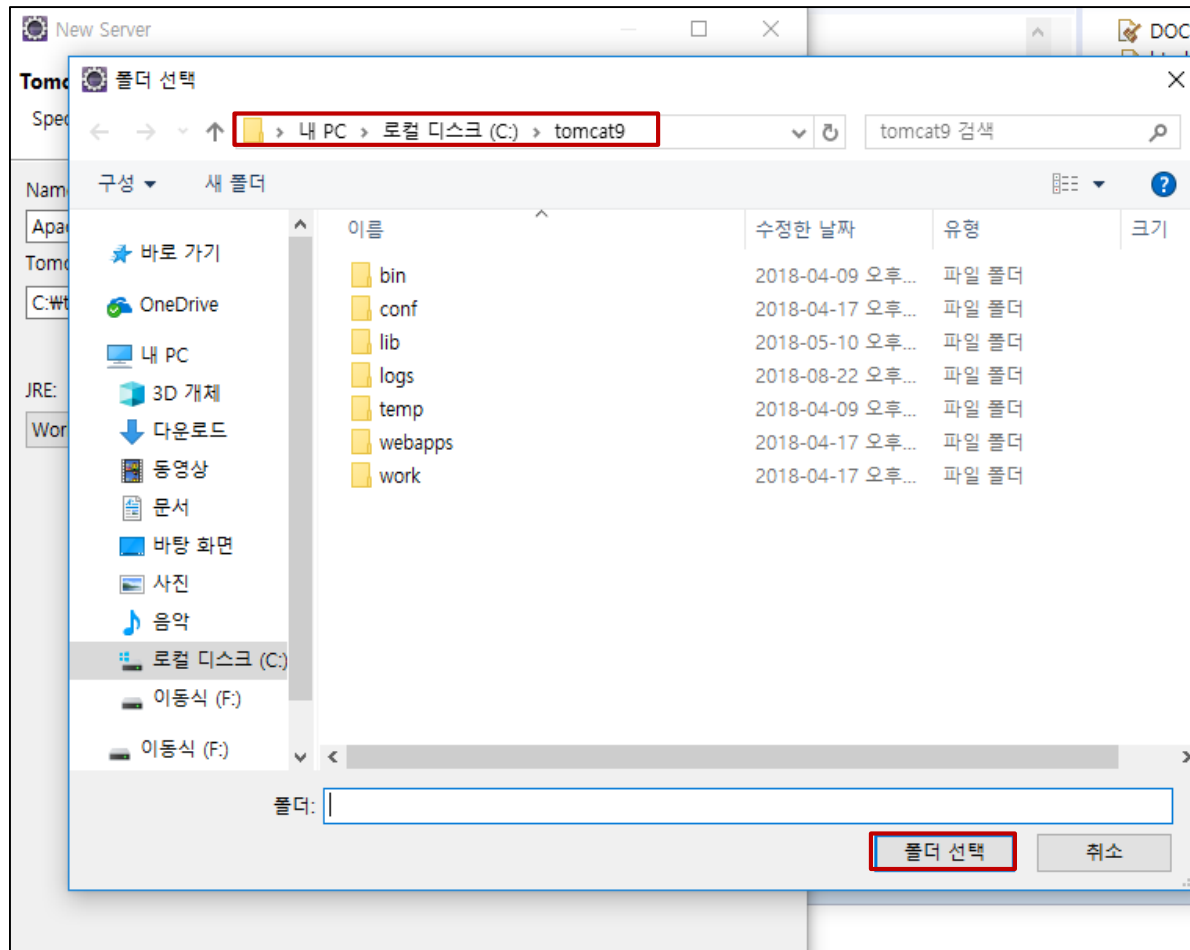


4.4 이클립스에서 웹 애플리케이션 실습하기

3. 톰캣 설치 디렉토리를 설정하기 위해 Browse...를 클릭하여 톰캣 홈 디렉토리(C:\tomcat9) 경로를 선택한 후 폴더 선택을 클릭합니다.

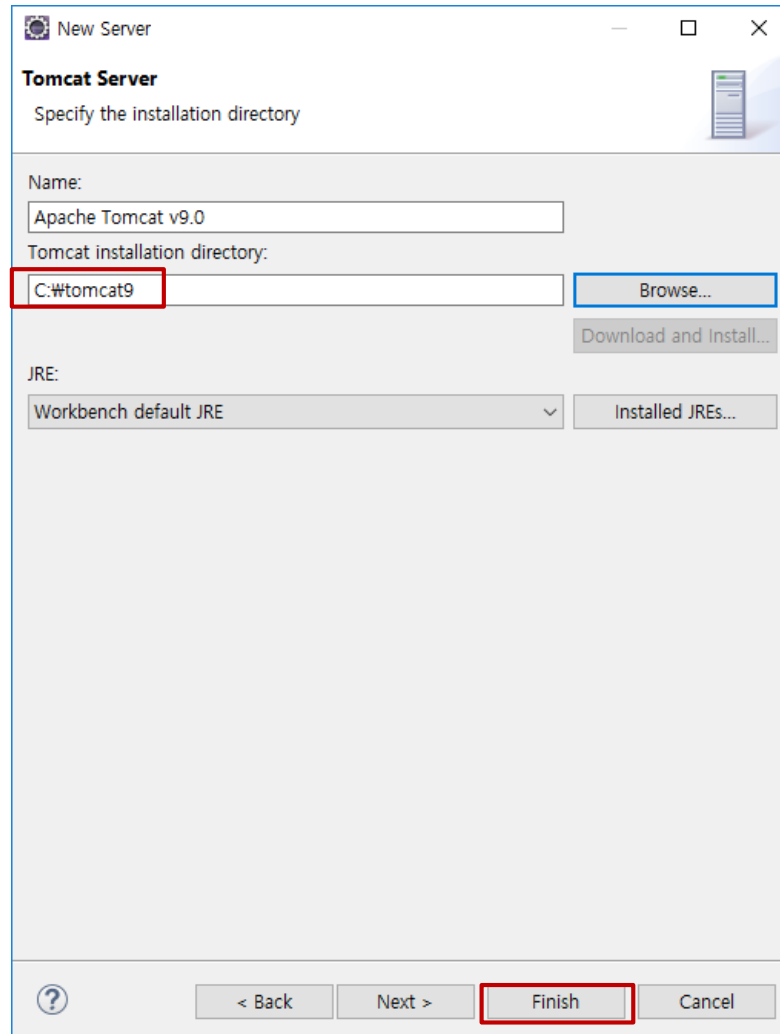


4.4 이클립스에서 웹 애플리케이션 실습하기



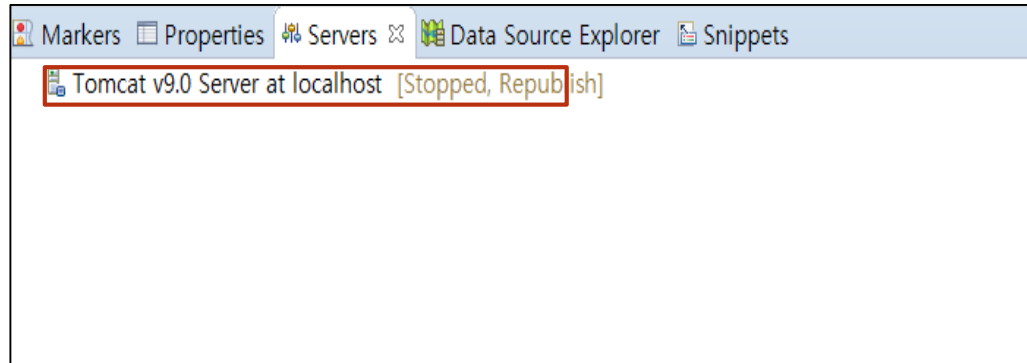
4.4 이클립스에서 웹 애플리케이션 실습하기

4. 설정 경로를 한 번 더 확인하고 Finish를 클릭합니다.



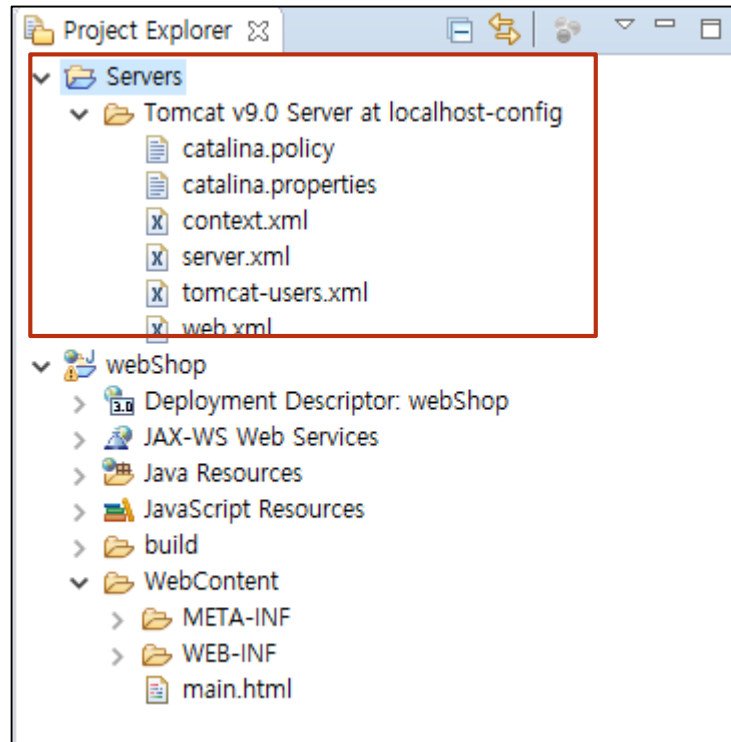
4.4 이클립스에서 웹 애플리케이션 실습하기

5. 이클립스 하단의 Servers 탭을 보면 다음과 같이 표시됩니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

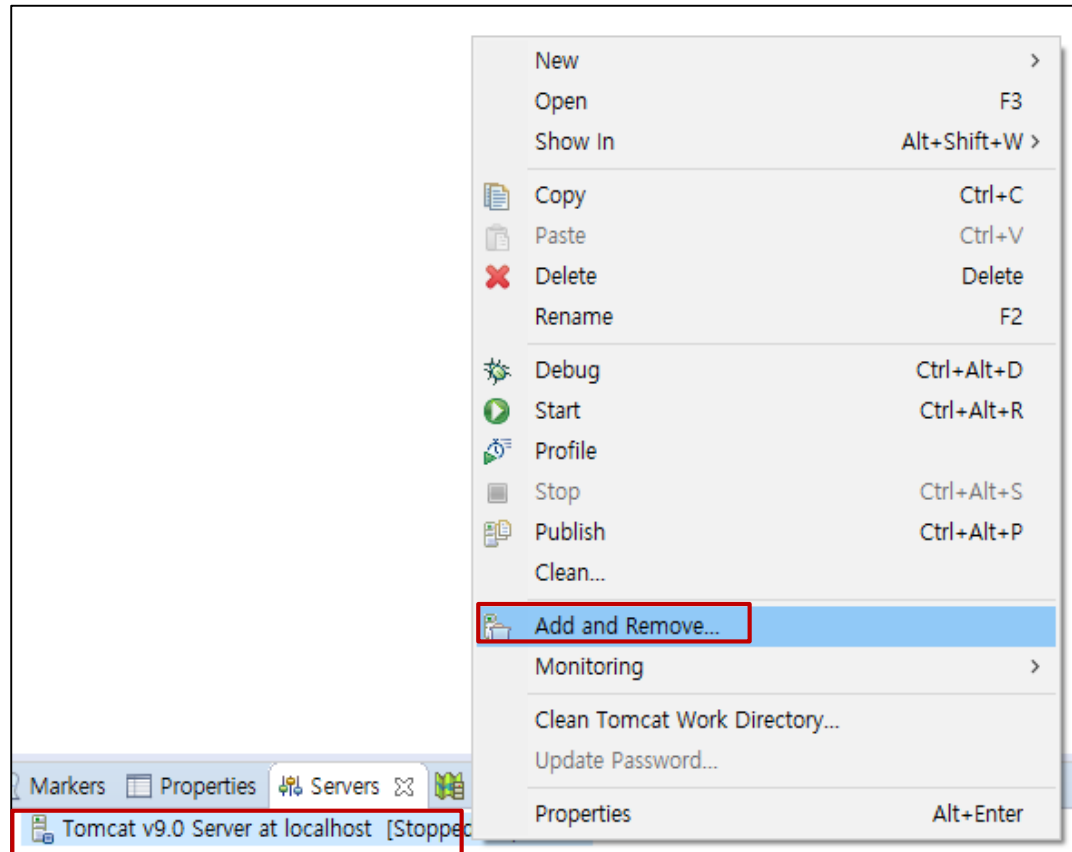
6. 이클립스에서 톰캣 컨테이너를 추가하면 다음과 같이 프로젝트 탐색기에 Servers 항목이 추가되면서 톰캣과 관련된 xml 파일들이 나타납니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

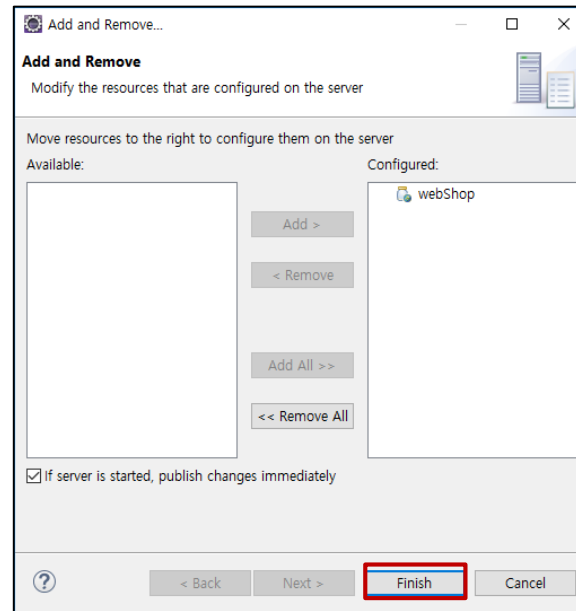
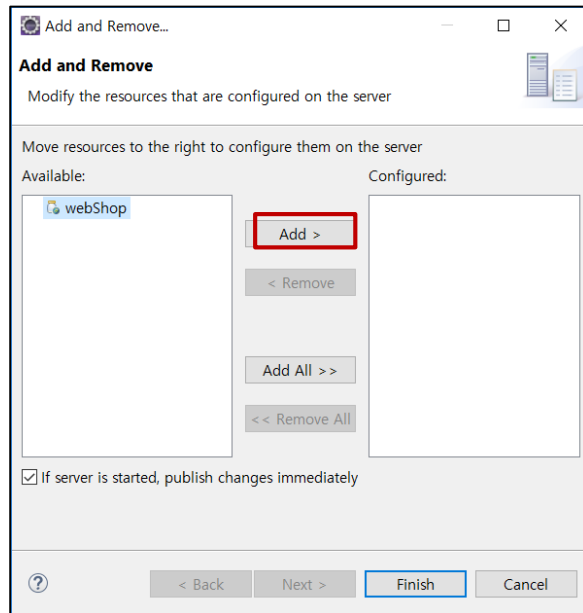
- 4.4.4 이클립스와 연동한 톰캣에 프로젝트 등록

1. Servers 탭 아래에 등록된 톰캣 서버 Tomcat v9.0 Server at localhost [Stopped]를 선택한 후 마우스 오른쪽 버튼을 클릭하여 Add and Remove를 선택합니다.



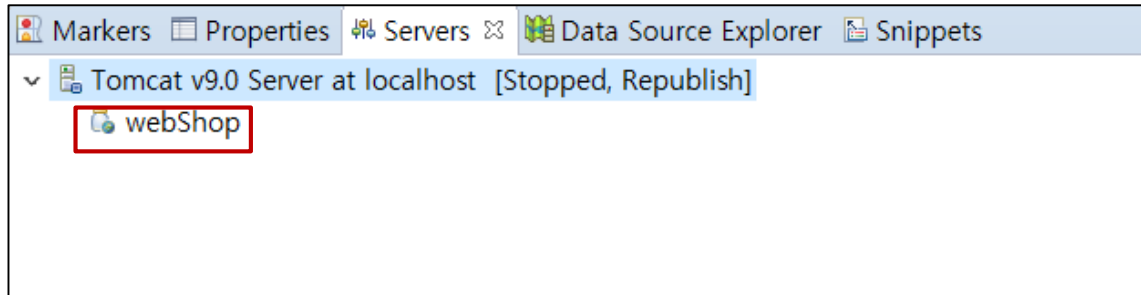
4.4 이클립스에서 웹 애플리케이션 실습하기

2. 추가할 프로젝트, 즉 여기서는 webShop을 선택한 후 Add를 클릭해 톱켓에 추가(등록)합니다. 이어서 Finish를 클릭하여 등록을 마칩니다.



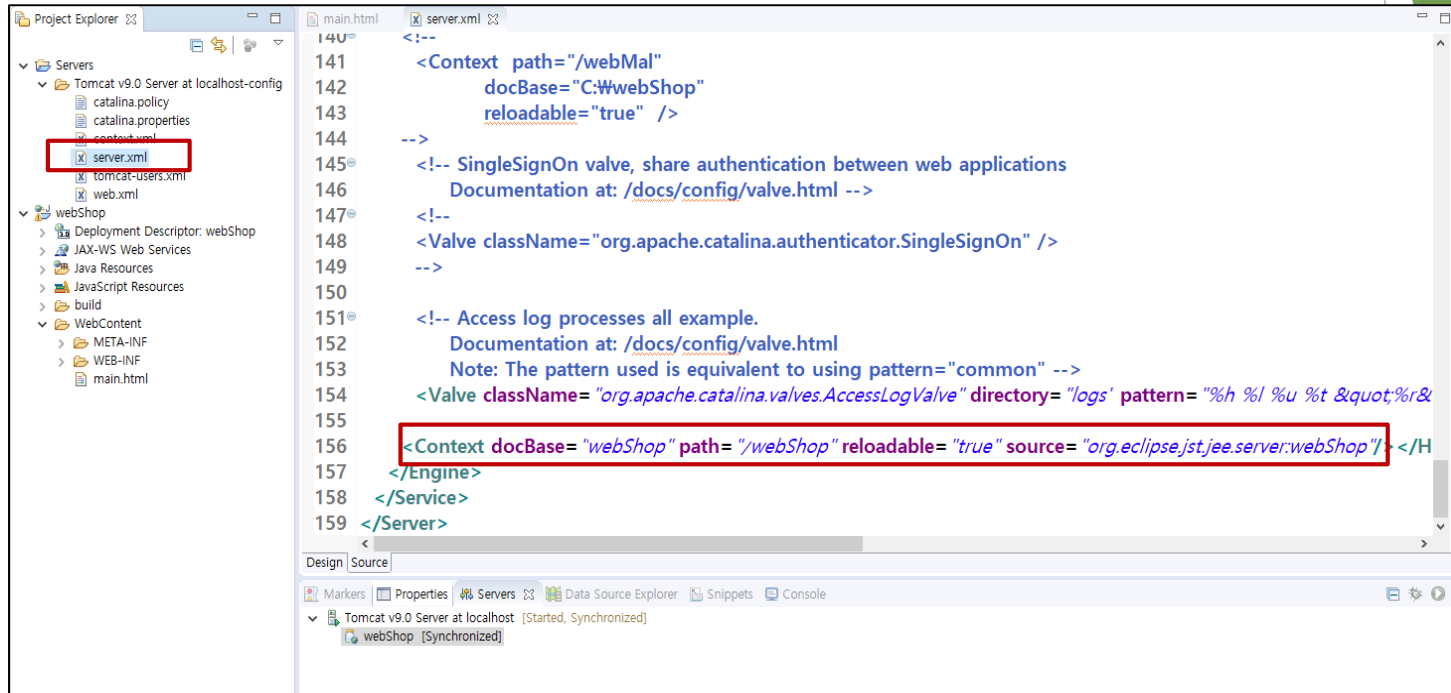
4.4 이클립스에서 웹 애플리케이션 실습하기

3. 톰캣 서버에 webShop 프로젝트가 등록된 것을 확인할 수 있습니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

4.server.xml을 클릭해서 프로젝트에 대한 컨텍스트가 자동으로 추가된 것을 확인합니다.

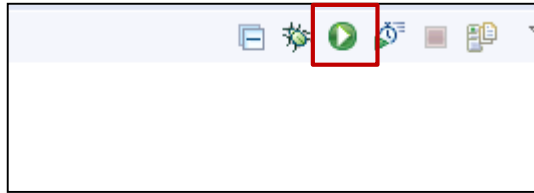


프로젝트 이름 = 컨텍스트 이름

4.4 이클립스에서 웹 애플리케이션 실습하기

• 4.4.5 웹 브라우저에서 요청하기

1. Servers 탭 오른쪽에 있는 녹색 실행 버튼을 클릭해 서버를 실행합니다.

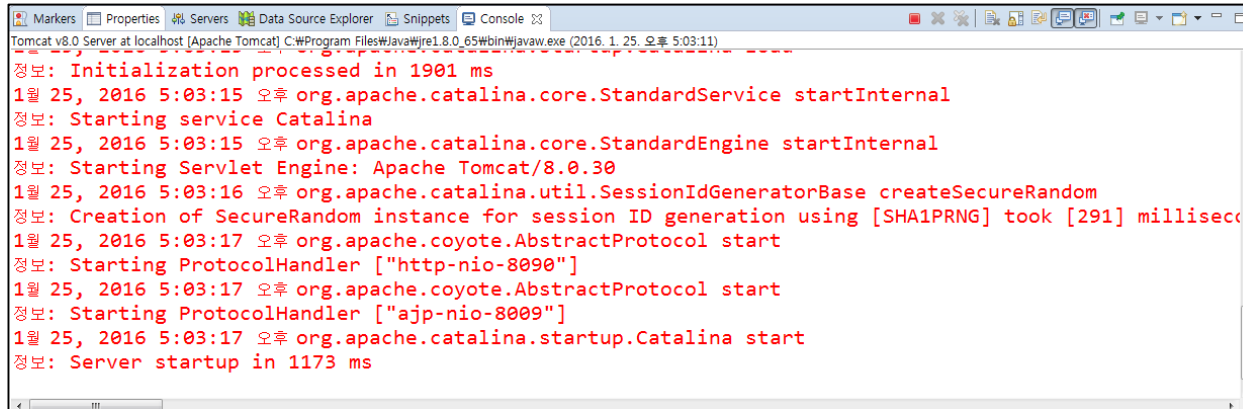


2. 다음과 같은 보안 경고창이 나타나면 액세스 허용을 클릭합니다.



4.4 이클립스에서 웹 애플리케이션 실습하기

3. 이클립스 Console 탭에 로그가 출력되면서 톰캣이 실행됩니다.



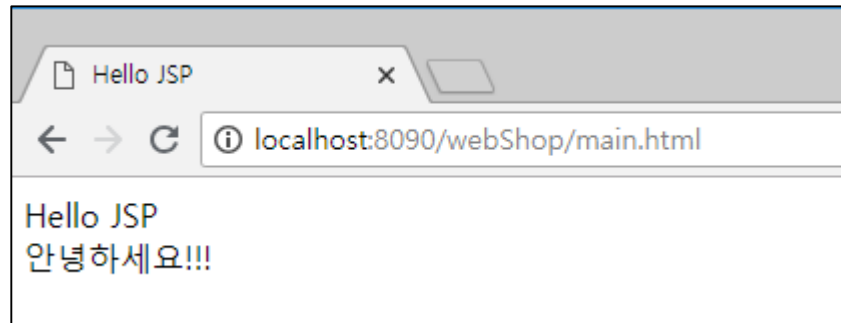
The screenshot shows the Eclipse IDE's Console window. The title bar indicates it's for 'Tomcat v8.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_65\bin\javaw.exe (2016. 1. 25. 오후 5:03:11)'. The console output, displayed in red text, shows the following sequence of events:

```
정보: Initialization processed in 1901 ms
1월 25, 2016 5:03:15 오후 org.apache.catalina.core.StandardService startInternal
정보: Starting service Catalina
1월 25, 2016 5:03:15 오후 org.apache.catalina.core.StandardEngine startInternal
정보: Starting Servlet Engine: Apache Tomcat/8.0.30
1월 25, 2016 5:03:16 오후 org.apache.catalina.util.SessionIdGeneratorBase createSecureRandom
정보: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [291] milliseconds
1월 25, 2016 5:03:17 오후 org.apache.coyote.AbstractProtocol start
정보: Starting ProtocolHandler ["http-nio-8090"]
1월 25, 2016 5:03:17 오후 org.apache.coyote.AbstractProtocol start
정보: Starting ProtocolHandler ["ajp-nio-8009"]
1월 25, 2016 5:03:17 오후 org.apache.catalina.startup.Catalina start
정보: Server startup in 1173 ms
```

4.4 이클립스에서 웹 애플리케이션 실행하기

4. 이제 웹 브라우저에서 다음 주소로 요청하면 브라우저에 메시지가 출력됩니다.

- <http://localhost:8090/webShop/main.html>



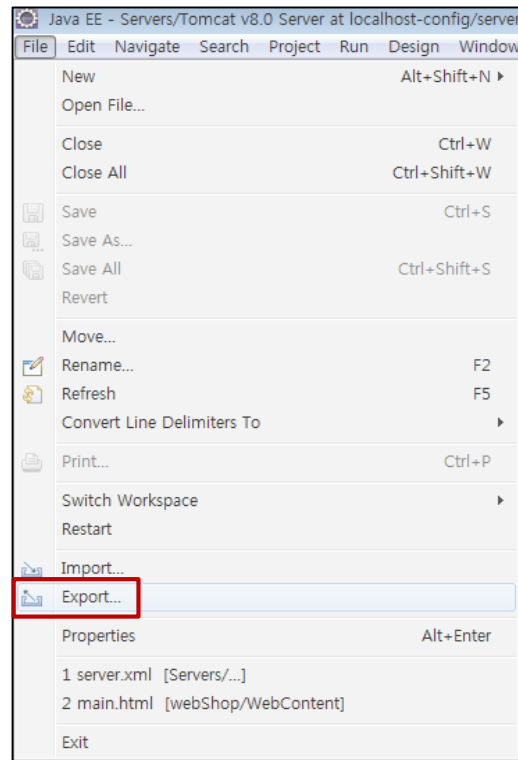
4.5 웹 애플리케이션 서비스하기

배포(Deploy)란?

개발이 완료된 웹 애플리케이션을 실제 사용자들에게 서비스하기 위한 작업

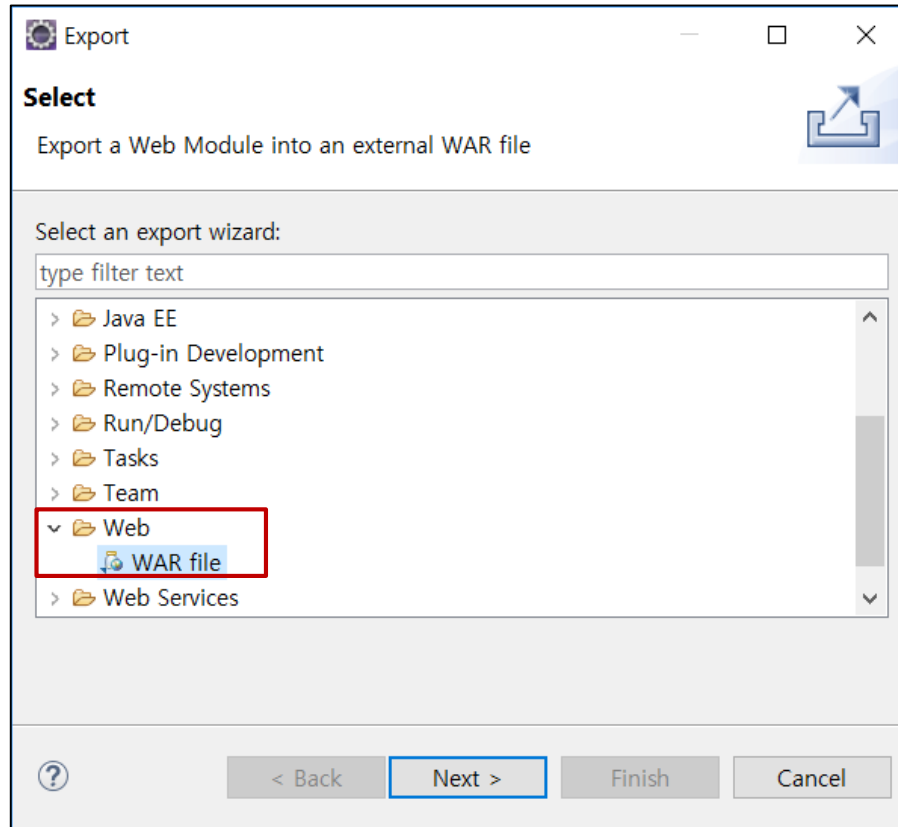
- 4.5.1 톰캣에 배치하기

1. 이클립스 상단 메뉴에서 File > Export...를 선택합니다.



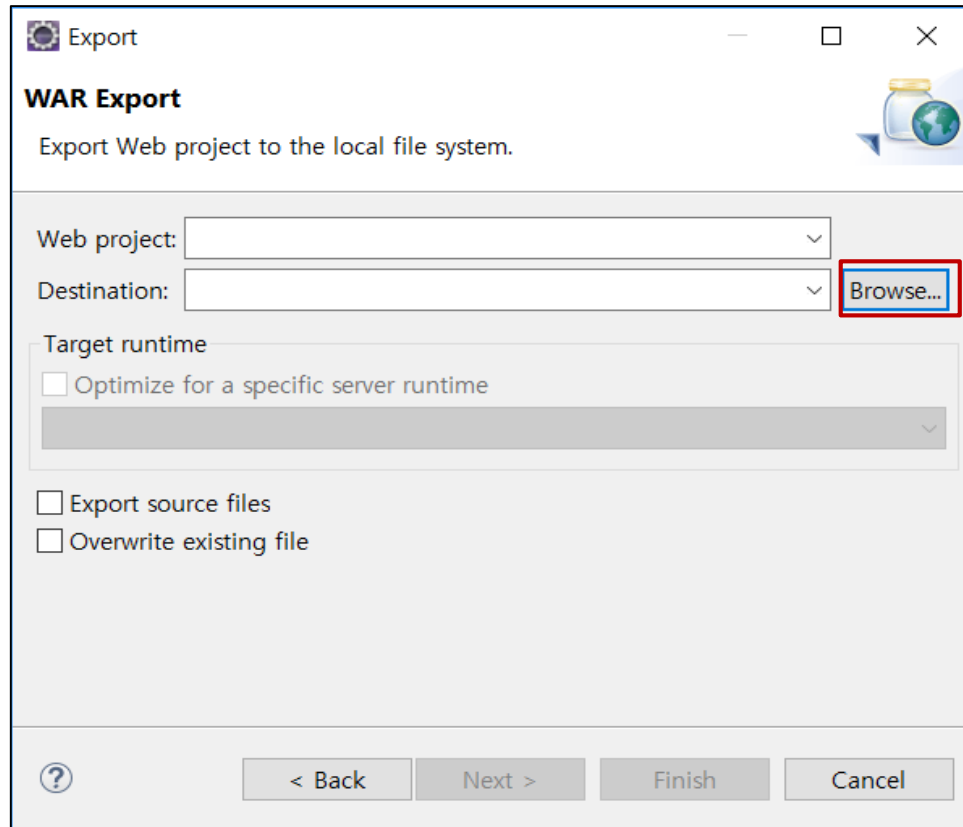
4.5 웹 애플리케이션 서비스하기

2. Web 항목의 WAR file을 선택한 후 Next를 클릭합니다.



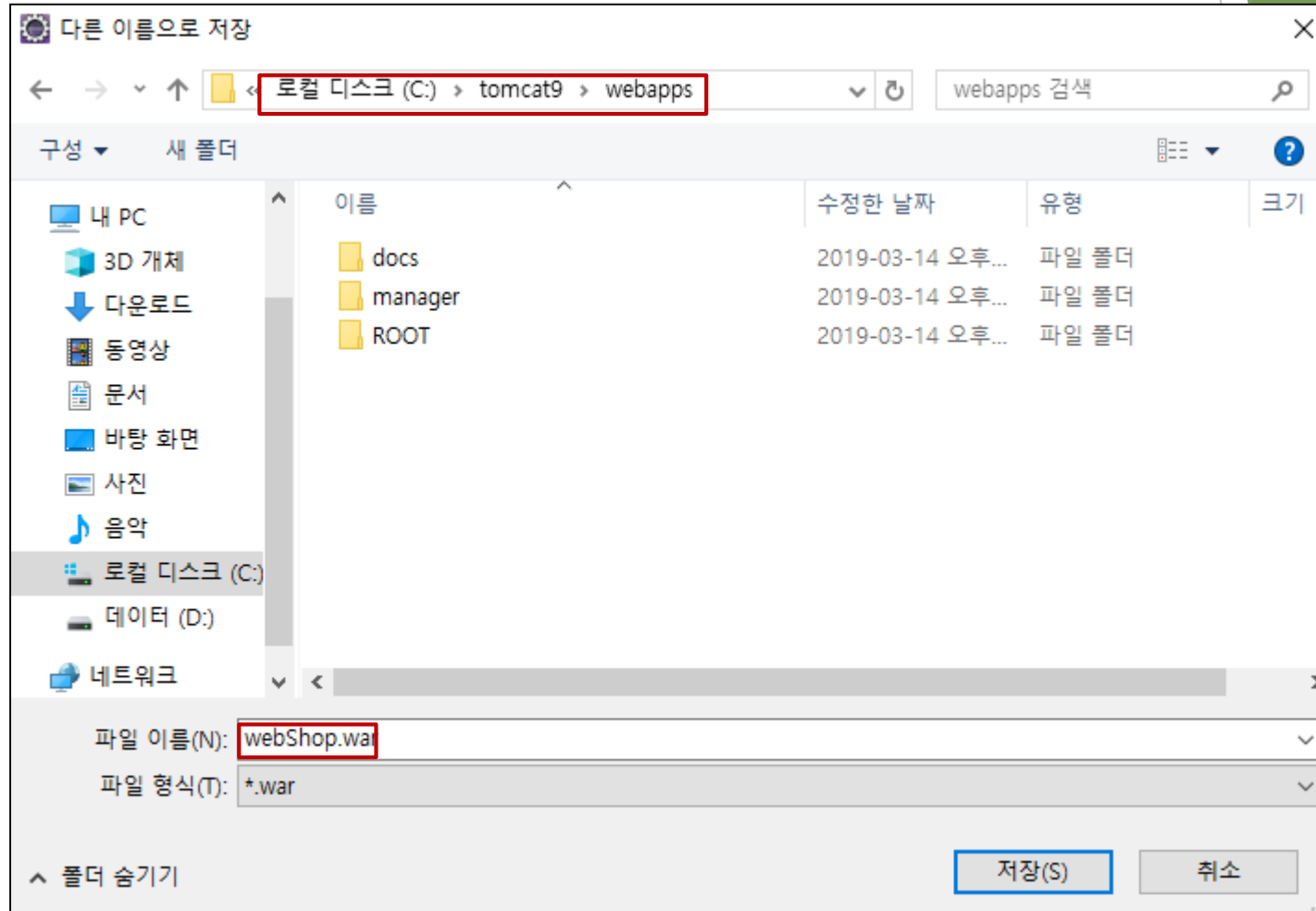
4.5 웹 애플리케이션 서비스하기

3. Browse...를 클릭해 war 파일을 저장할 위치를 지정합니다.



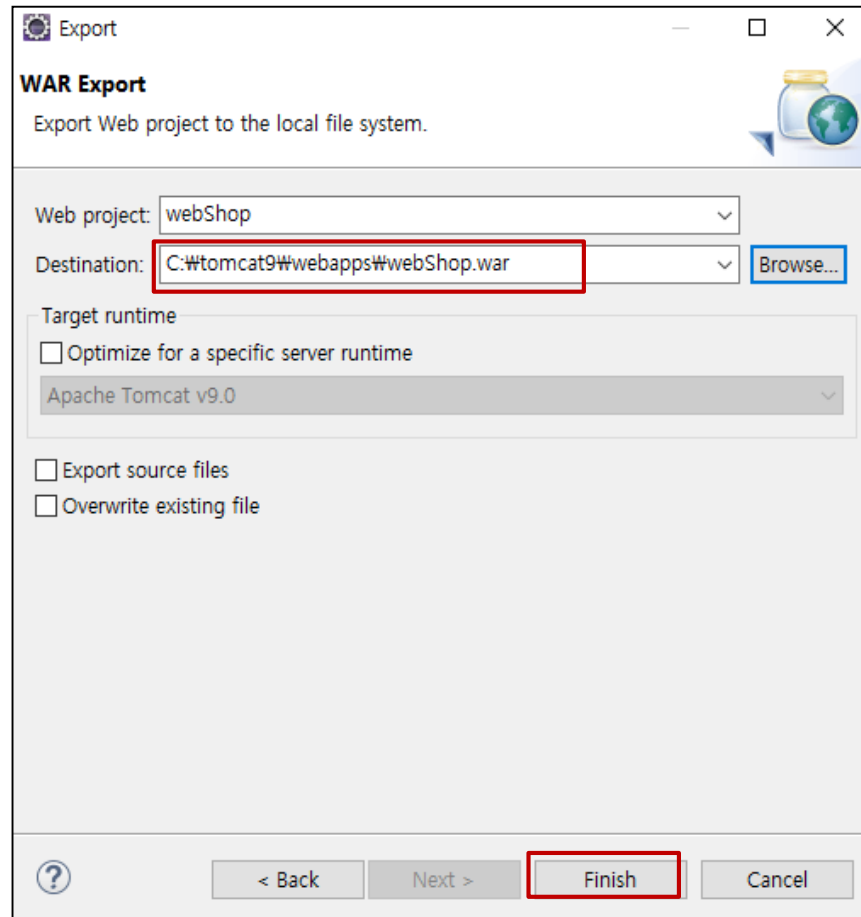
4.5 웹 애플리케이션 서비스하기

4. 톰캣 폴더의 webapps 디렉터리를 지정하고 webShop.war라는 파일 이름으로 저장합니다.



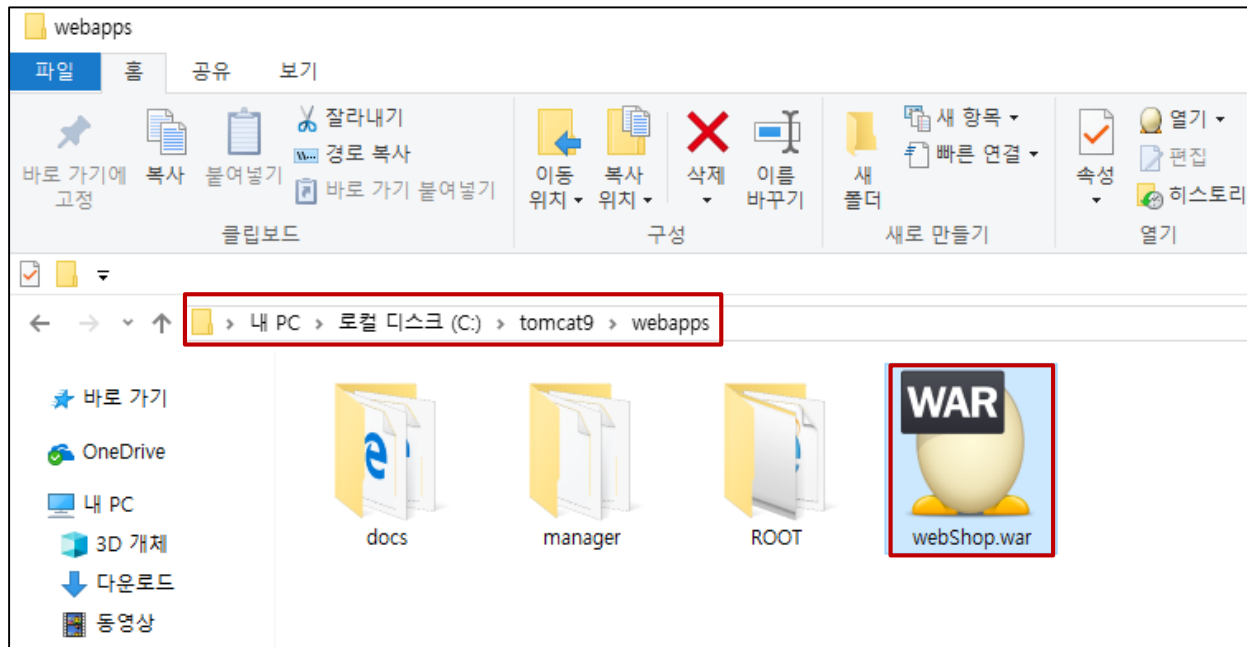
4.5 웹 애플리케이션 서비스하기

5. 지정한 war 파일의 경로를 최종 확인한 후 Finish를 클릭해 war 파일을 내보냅니다(export).



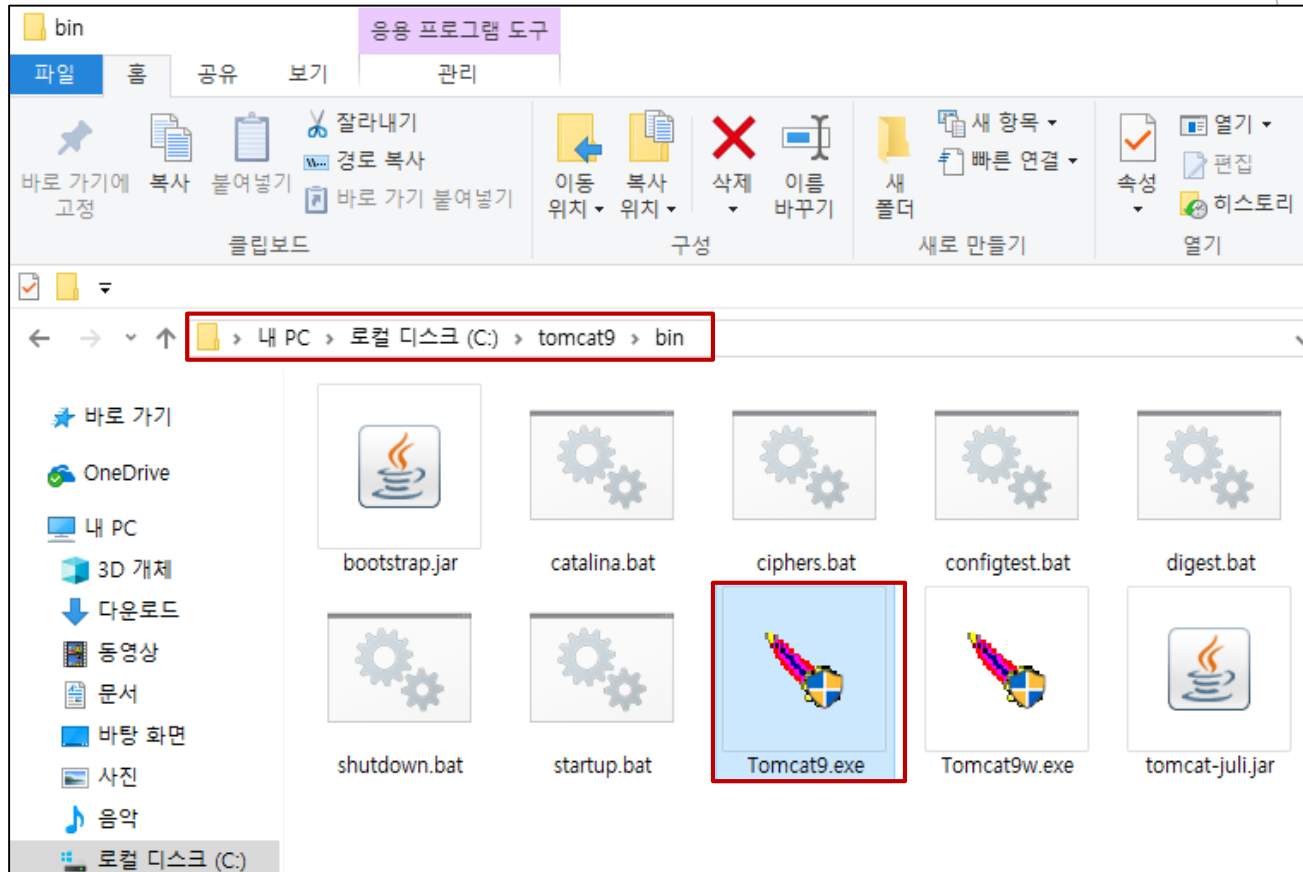
4.5 웹 애플리케이션 서비스하기

6. 톰캣 디렉터리의 webapps 폴더를 보면 war 파일이 생긴 것을 확인할 수 있습니다.



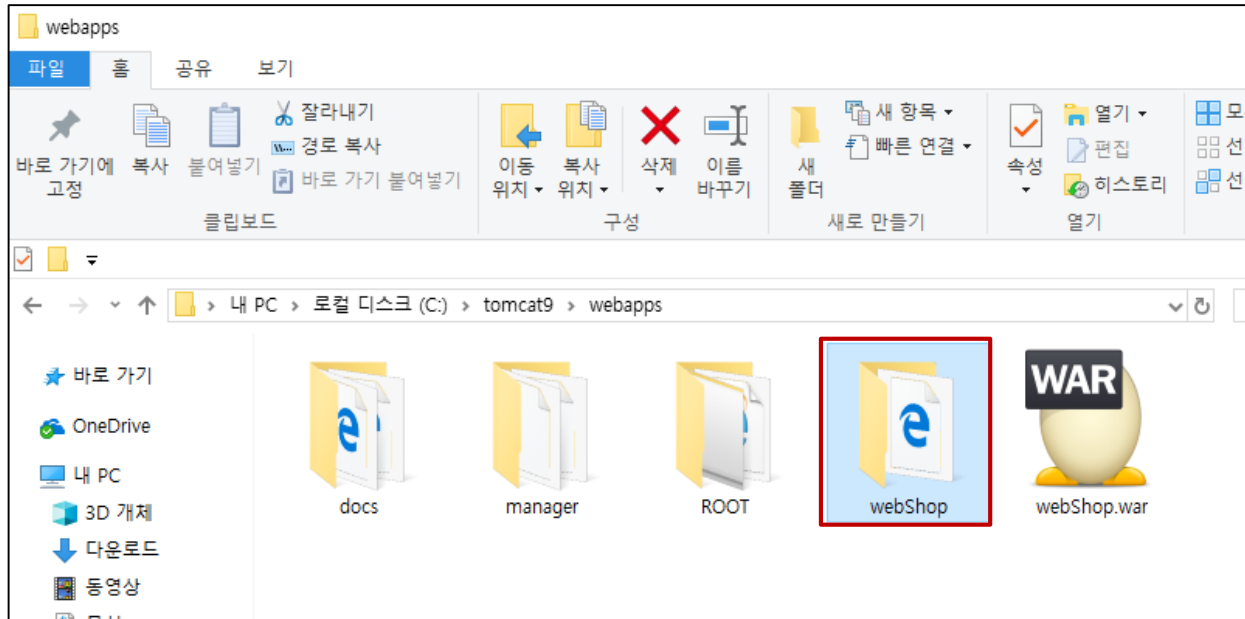
4.5 웹 애플리케이션 서비스하기

7. 톰캣 루트 디렉터리 하위의 bin 폴더에서 Tomcat9.exe를 더블클릭하여 실행합니다.



4.5 웹 애플리케이션 서비스하기

8. 톰캣은 webapps의 war 파일을 압축을 해제함과 동시에 자동으로 컨텍스트로 등록합니다.

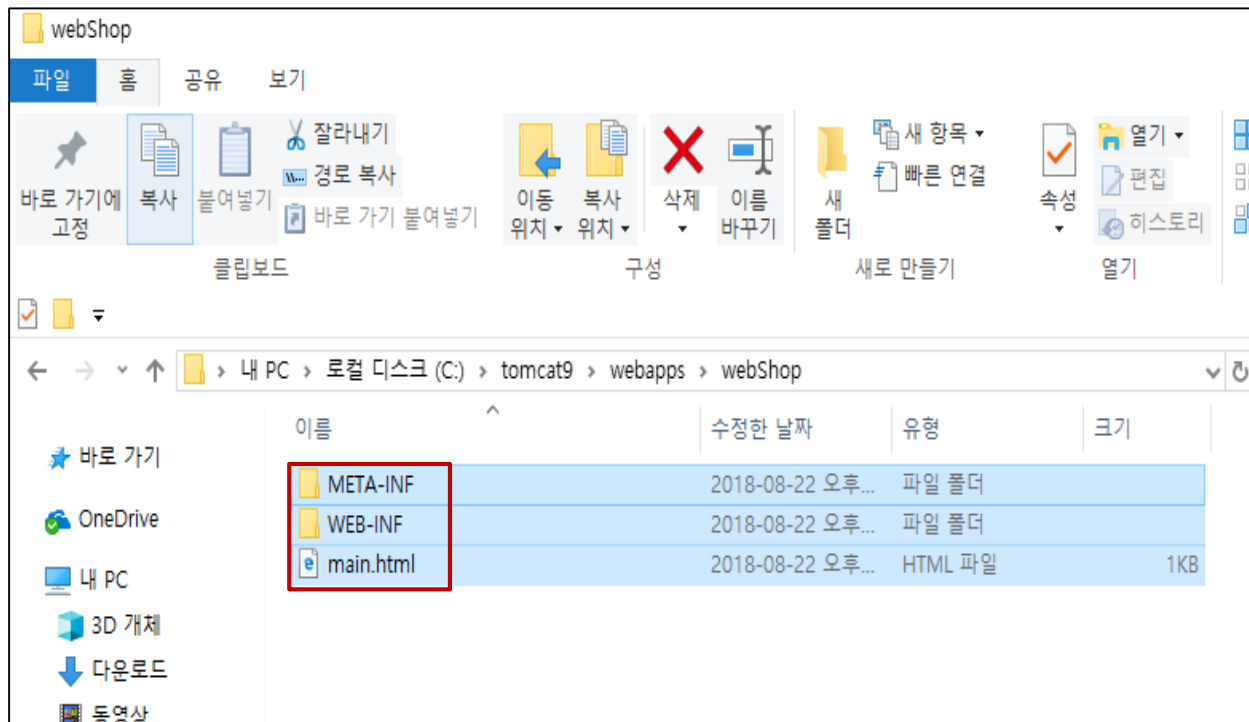


❖ Tip

webShop.war를 배치하면 webShop 폴더가 생성되므로 4.3절에서 수동으로 생성한 webShop 폴더는 삭제해야 합니다.

4.5 웹 애플리케이션 서비스하기

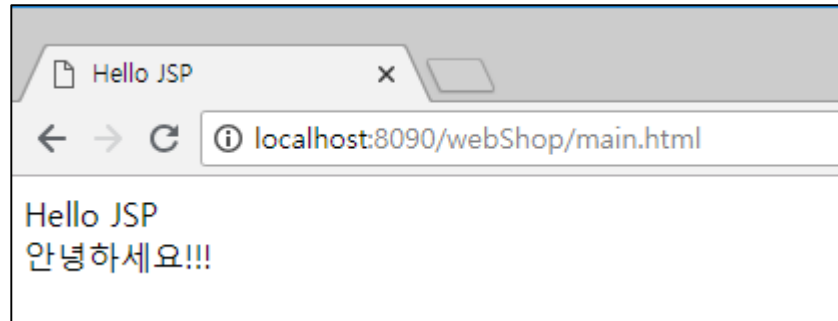
9. 압축을 해제한 webShop 폴더를 확인해 보면 이클립스에서 생성한 WEB-INF 폴더와 main.html 파일이 보입니다.



4.5 웹 애플리케이션 서비스하기

10. 이제 웹 브라우저에서 다음과 같이 컨텍스트 이름으로 요청하면 앞의 이클립스에서 실행한 것과 동일한 메시지가 출력됩니다.

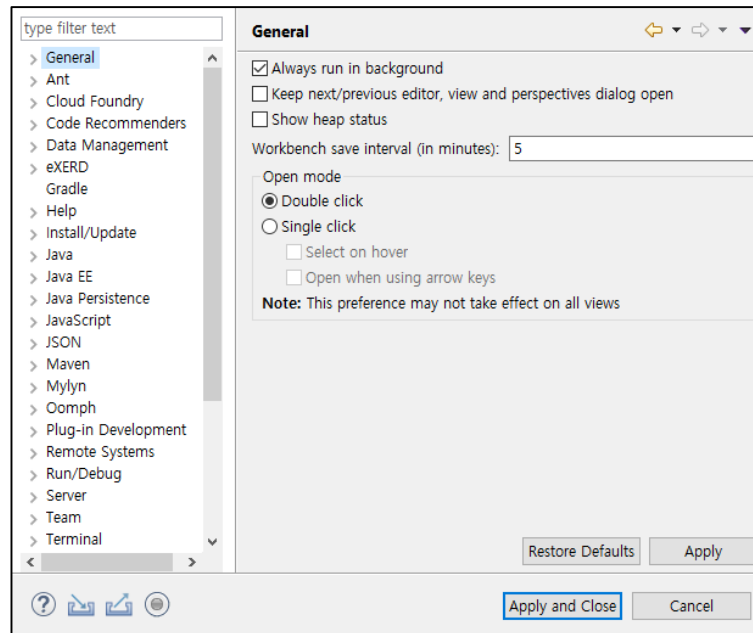
- <http://localhost:8090/webShop/main.html>



4.5 웹 애플리케이션 서비스하기

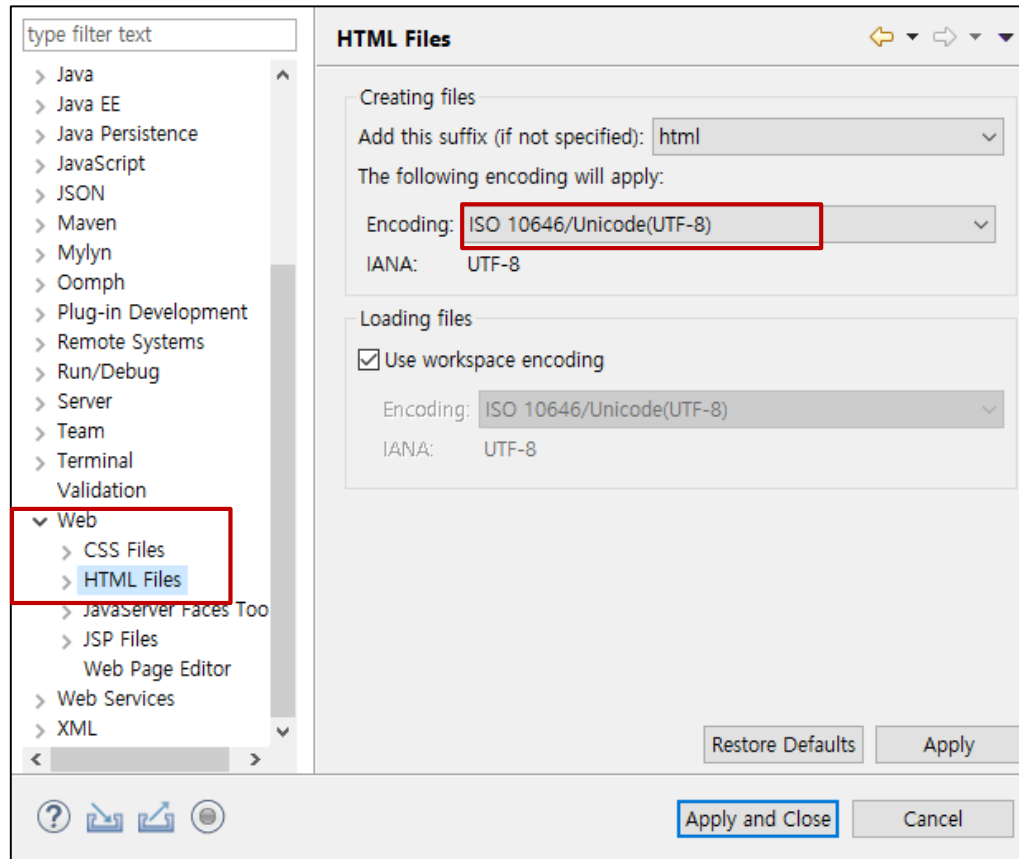
- 이클립스에서 **HTML**과 **JSP** 파일 한글 이코딩 **UTF-8**로 설정하기

1. 이클립스 상단 메뉴의 Window > Preference를 클릭하여 속성창을 엽니다.



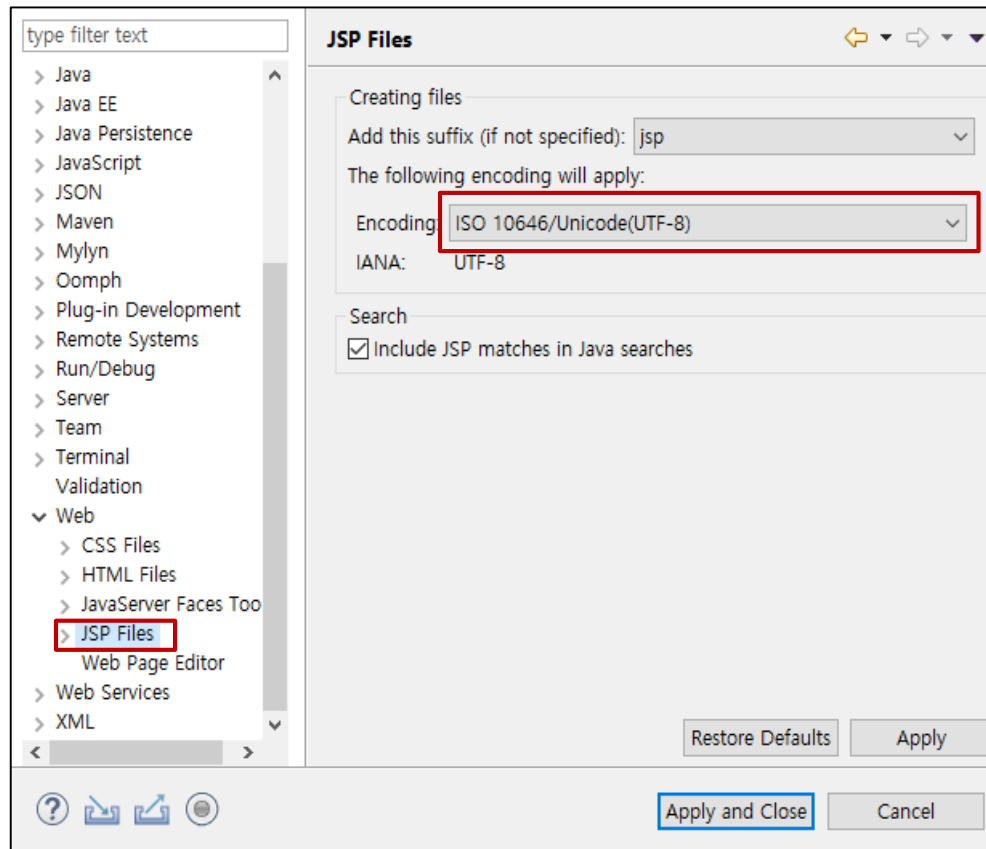
4.5 웹 애플리케이션 서비스하기

2. 속성창 왼쪽 메뉴에 있는 Web > HTML Files를 선택한 후 Encoding을 UTF-8로 변경합니다.



4.5 웹 애플리케이션 서비스하기

3. 마찬가지로 JSP Files을 선택한 후 Encoding을 UTF-8로 변경하고 Apply and Close를 클릭합니다.



4.5 웹 애플리케이션 서비스하기

4. 프로젝트에서 test.html을 생성하면 charset이 UTF-8로 설정된 것을 확인할 수 있습니다.

