

IBM WAS Liberty 서버 기반
전자정부 프레임워크(Spring)
개발 가이드 01

이정운과장 (juwlee@kr.ibm.com)
IBM Korea, WebSphere Technical Sales

목 차

1	개발 환경 구성	3
1.1	이클립스 및 WDT 구성	3
1.2	IBM WAS Liberty 서버 설치	4
1.3	Eclipse 에 IBM WAS Liberty 서버 연동	6
1.4	IBM WAS Liberty 서버에 JVM 옵션 추가하는 방법	9
2	전자정부 프레임워크(Spring) 개발을 위한 환경 설정	11
2.1	전자정부 프레임워크(Spring) 연동 설정	11

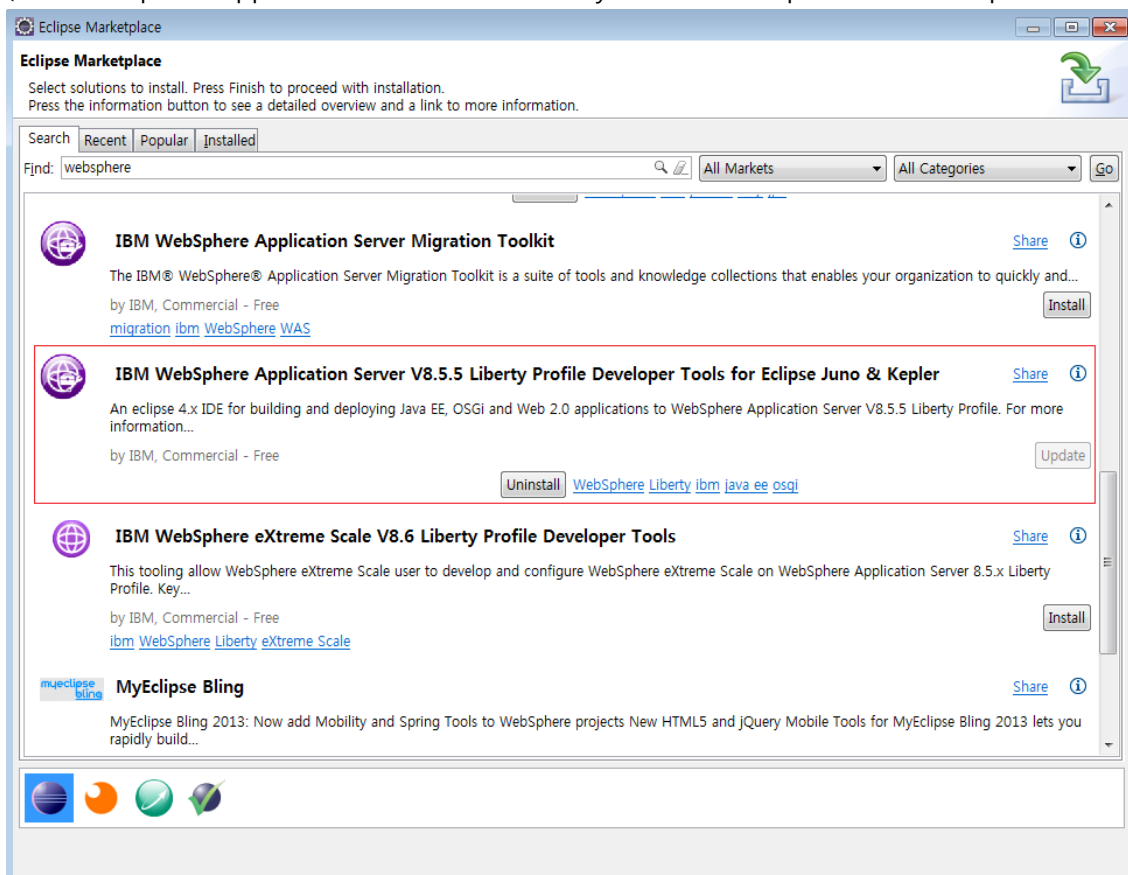
1 개발 환경 구성

1.1 이클립스 및 WDT 구성

웹서비스 어플리케이션 개발을 위해서 이클립스와 WebSphere Developer Tools(WDT) for Eclipse 를 설치 구성합니다.

최신 WDT 설치를 위해서는 이클립스 Juno 및 Kepler 버전이 가능하며 이클립스 설치 후 **Help > Eclipse marketplace** 에서 **WebSphere** 라는 키워드로 검색해서 아래와 같이 WDT for Eclipse 툴을 선택하여 설치만 하시면 됩니다.

(IBM WebSphere Application Server v8.5.5 Liberty Profile Developer Tools for Eclipse Juno & Kepler)



만약, Internet 이 안되고 반드시 offline 으로 작업해야 하는 상황이라서 WDT plugin 을 업데이트 하기 위한 file 을 다운로드 하고자 한다면 하단의 링크로 접속하여 zip 파일을 다운로드 받아서 사용하면 됩니다.

<https://www.ibmdev.net/wasdev/downloads/liberty-profile-using-eclipse/>

1.2 IBM WAS Liberty 서버 설치

WDT 를 설치하였으면 이제 그와 연동되는 IBM WAS Liberty 서버를 설치 해야 합니다. 해당 image 를 받기 위해서 하단의 링크에 접속합니다.

(<https://www.ibmmdw.net/wasdev/downloads/liberty-profile-using-non-eclipse-environments/>)

Download Liberty profile using non-Eclipse environments

Install the Liberty profile runtime in build environments or non-Eclipse IDEs.

1. Download the Liberty profile runtime:

Download

2. Run the following command then follow the prompts to install the Liberty profile runtime:

```
java -jar wlp-developers-runtime-8.5.5.2.jar
```

3. From the wlp/bin directory in your Liberty profile runtime installation, run the following command to create a new server:

```
server create server-name
```

The new server is created in `wlp/usr/servers/server-name`.

4. Run the following command to start the server:

```
server start server-name
```

5. To deploy an application, copy the .war file to the server's dropins directory:

```
usr/servers/server-name/dropins
```

해당 링크에 접속했으면 Download 버튼을 클릭한 후 license 를 accept 한 후에 해당 jar 파일을 다운로드 받습니다.



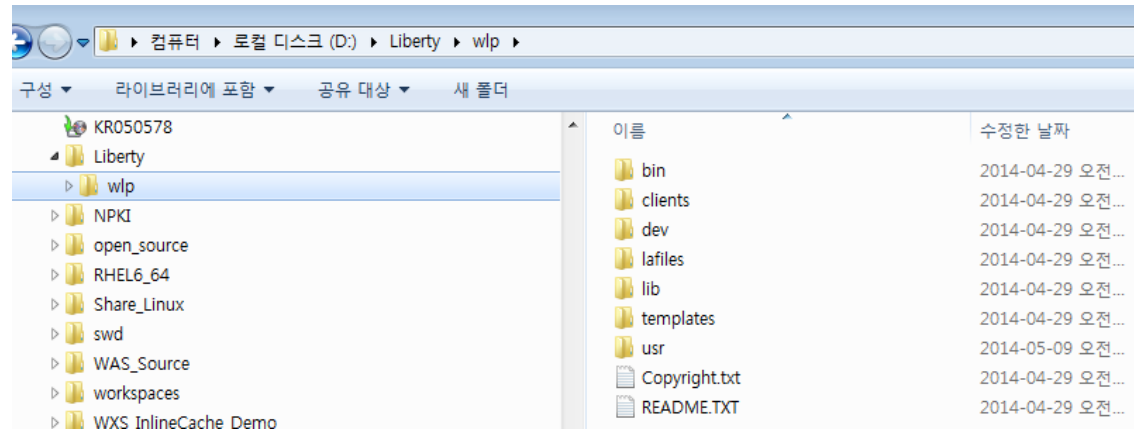
IBM WAS Liberty 서버는 별도의 설치과정이 필요 없습니다. 다운로드 받은 Jar 파일을 하단과 같이 압축을 해제하면 설치를 완료할 수 있습니다. (java -jar wlp-developers-runtime-8.5.5.2.jar)

```
D:\Weclipses\Weclipse-spring-liberty>java -jar wlp-developers-runtime-8.5.5.2.jar
IBM WebSphere Application Server for Developers V8.5.5을<를> 사용하거나 압축을
풀거나 설치하려면 먼저 무보증 프로그램에 관한 국제 라이선스 계약의 이용 약관 및
추가 라이선스 정보에 동의해야 합니다. 다음
라이선스 계약을 주의깊게 읽으십시오.

--viewLicenseAgreement 옵션을 사용하여 라이선스 계약을 별도로 볼 수 있습니다.

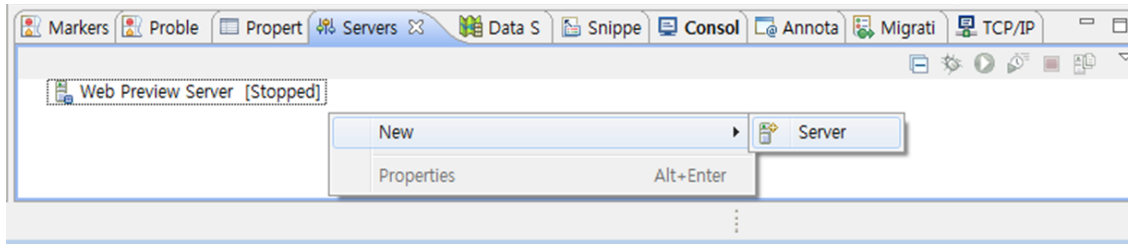
지금 라이선스 이용 약관을 표시하려면 Enter를 누르고 건너뛰려면 'x'를 누르십시오.
```

압축이 정상적으로 해제하면 하단과 같은 폴더 구조로 설치가 완료된 것을 확인할 수 있습니다.

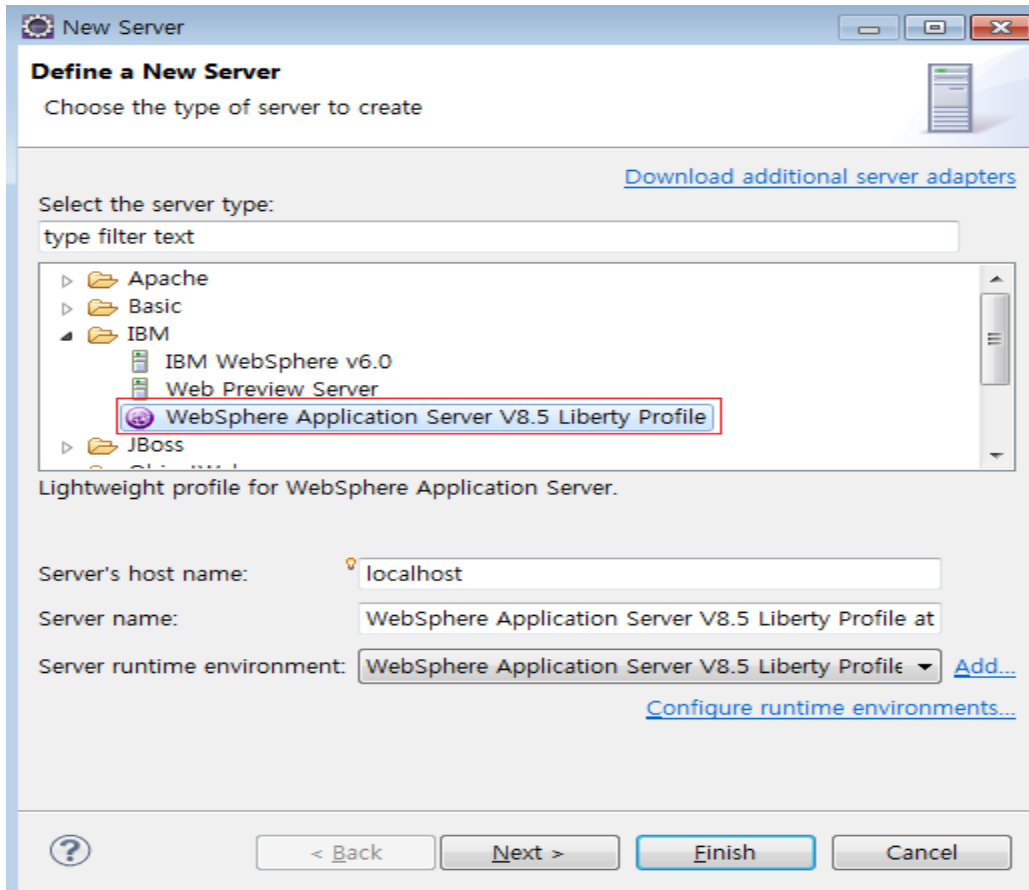


1.3 Eclipse 에 IBM WAS Liberty 서버 연동

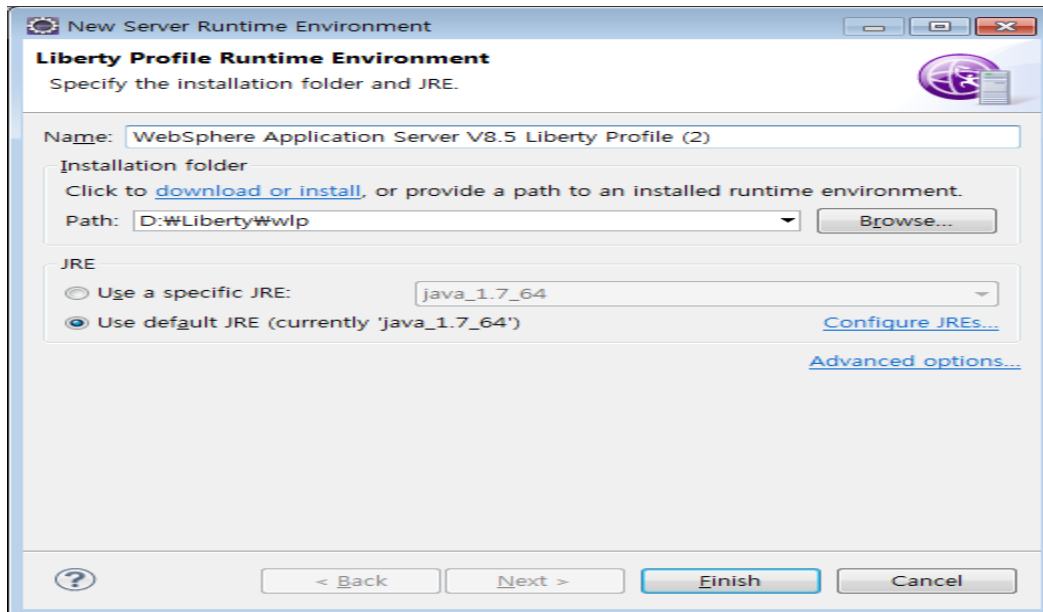
Eclipse 를 위한 WDT 설치와 IBM WAS Liberty 서버 설치가 완료되었으면 하단과 같이 Eclipse 의 Servers 탭을 선택하여 마우스 우 클릭하여 **New > Server** 를 선택합니다.



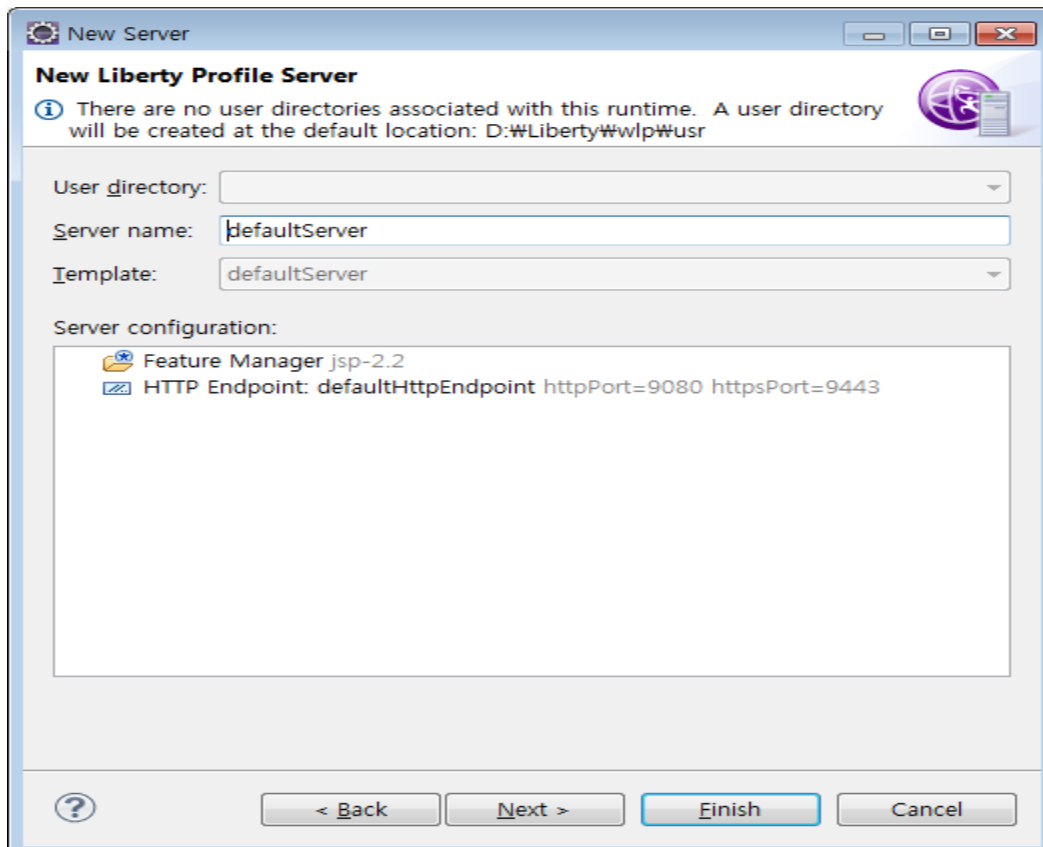
Server type 에서 **WebSphere Application Server v8.5 Liberty Profile**를 선택하고 **Server runtime environment** 항목에서 **Add** 버튼을 클릭합니다.



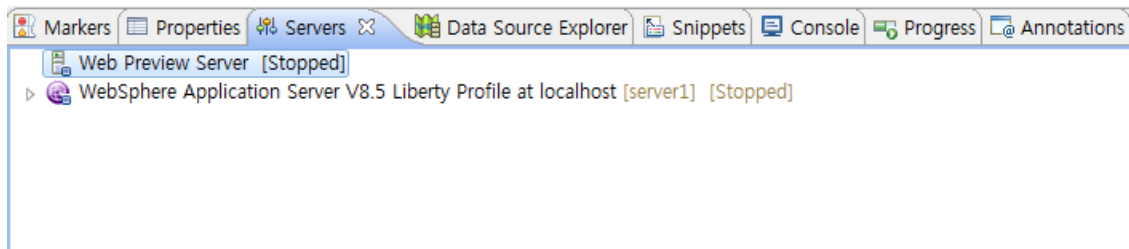
그 후 방금 설치한 IBM WAS Liberty 서버의 위치를 지정하고 사용하고자 하는 Java 버전을 선택 하신 후 **Finish** 버튼을 클릭하여 서버 추가를 마무리 합니다.



다시 서버 추가 화면이 나오면 **Next** 를 클릭하여 서버 설정을 생성한 후 **Finish** 를 클릭하여 설정을 마무리 합니다. (여기서는 서버에 별다른 추가 설정 없이 기본으로 사용합니다.)



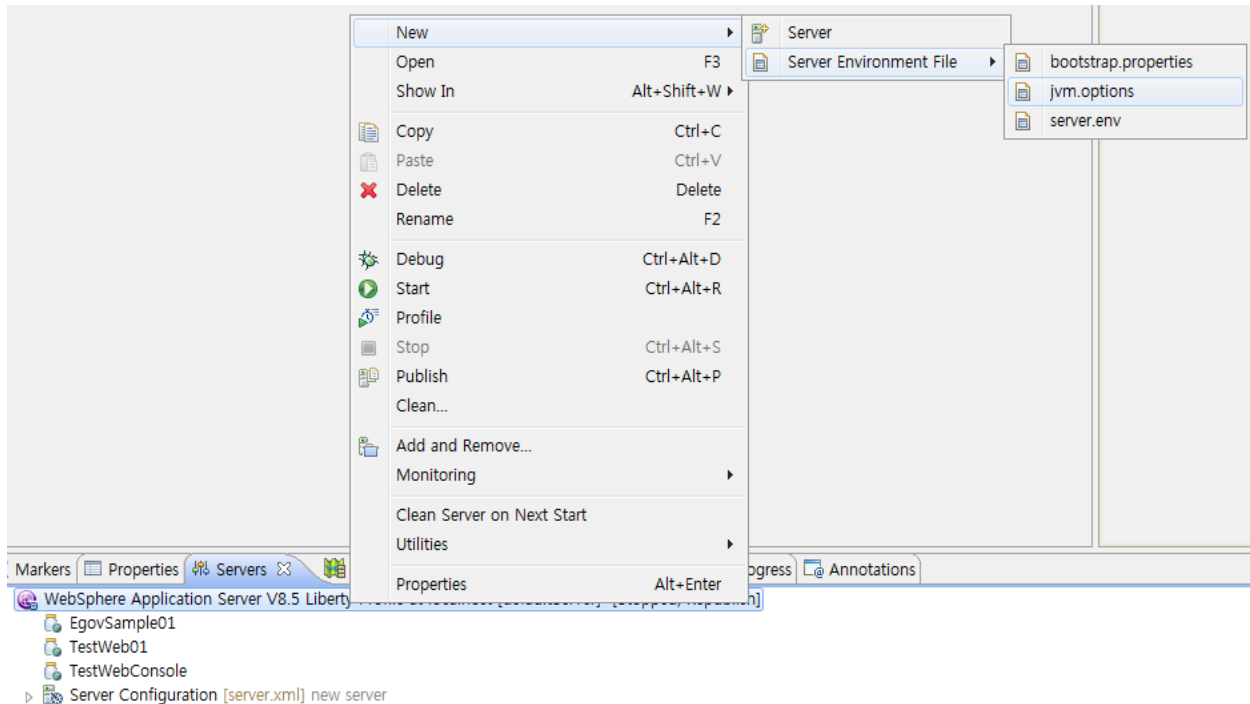
해당 설정을 문제없이 완료하시면 다음과 같이 IBM WAS Liberty 서버가 추가된 화면을 Eclipse 의 Servers 탭에서 확인하실 수 있습니다.



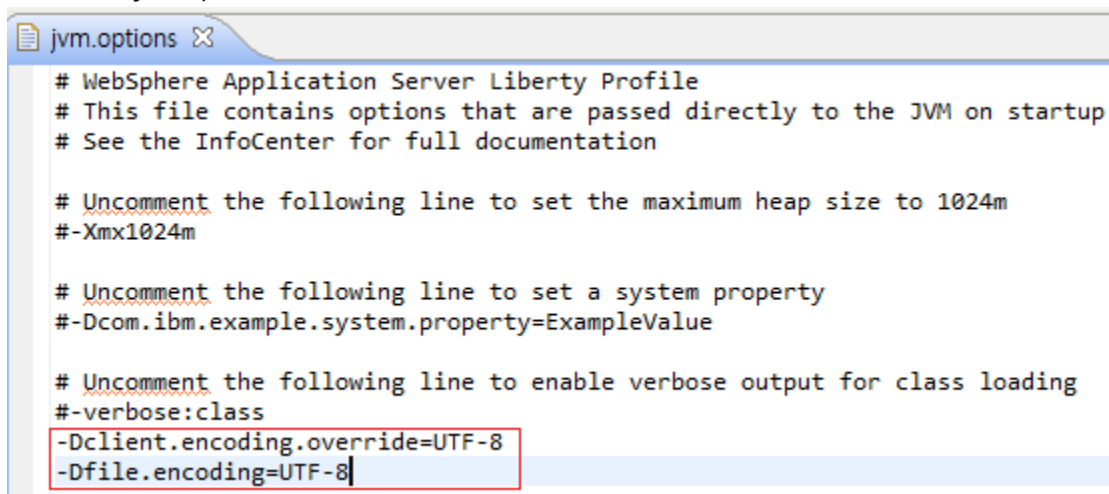
1.4 IBM WAS Liberty 서버에 JVM 옵션 추가하는 방법

필요에 따라서 IBM WAS Liberty 서버에 JVM 옵션을 추가해서 넣어야 할 경우가 많이 있습니다. 이 경우에는 기존 IBM WAS 와 다르게 Liberty 서버 디렉토리에 jvm.options 라는 파일을 추가한 후 옵션을 넣으면 됩니다.

1) eclipse 의 Server 탭에서 마우스 우 클릭한 후 New > Server Environment File > jvm.options 선택



2) 생성된 jvm.options 파일에 원하는 JVM 옵션을 추가하면 됩니다.

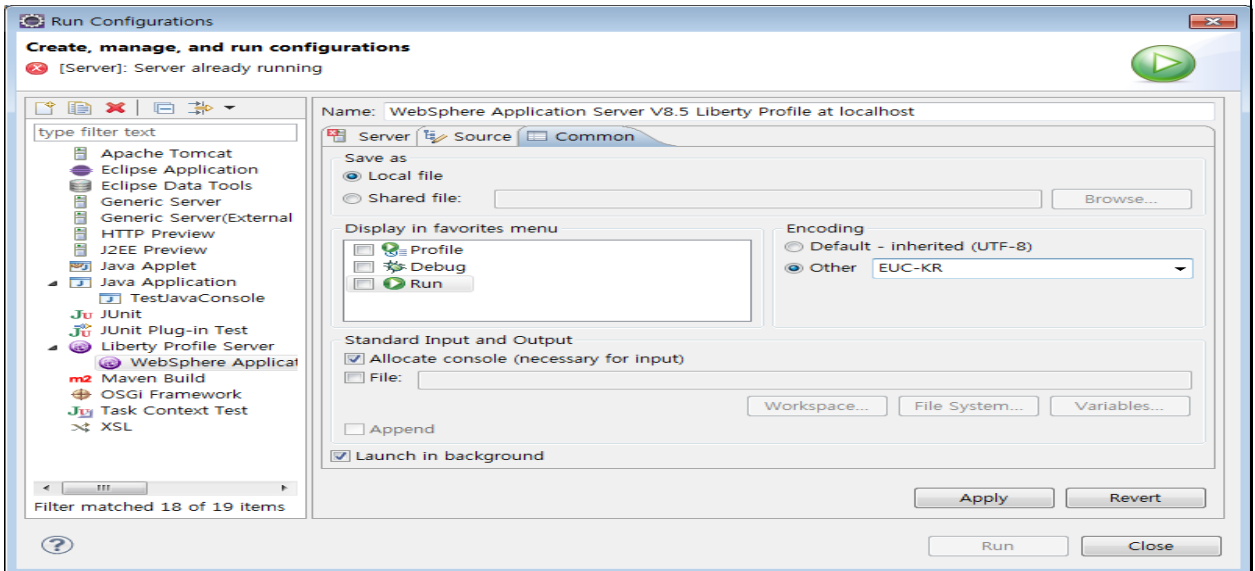


팁 #1) 전자정부 프레임워크 사이트에서 받은 eclipse 를 이용해서 IBM WAS Liberty 를 실행시키고 테스트 해보면 하단과 같이 eclipse 의 console 에서 한글이 깨지는 경우를 확인할 수 있습니다. (Text editor 로 열어보면 한글이 깨져있는 것은 아니고 console 에서만 깨져 보임)

```

IBM J9 VM0000 defaultServer(WebSphere Application Server 8.5.5.2/wlp-1.0.5.cl50220140403-1858) ****, **
[AUDIT ] CWWKE0001I: defaultServer 000000 000000000000.
[AUDIT ] CWWKZ0058I: ***** dropins*****.
[AUDIT ] CWWKT0016I: ***** (default_host): http://localhost:19080/TestWebConsole/
[AUDIT ] CWWKZ0001I: 0.305*** TestWebConsole*****.
[AUDIT ] CWWKT0016I: ***** (default_host): http://localhost:19080/TestWeb01/
[AUDIT ] CWWKZ0001I: 0.498*** TestWeb01*****.
[AUDIT ] CWWKT0016I: ***** (default_host): http://localhost:19080/web/
[AUDIT ] CWWKZ0001I: 2.258*** EgovSample01*****.
[AUDIT ] CWWKF0011I: defaultServer 000000 000000 000000 000000 000000.
  
```

이 경우 Run > Run Configurations 메뉴에서 Liberty profile 을 선택하고 common 탭에서 Encoding 을 하단과 같이 EUC-KR 로 변경하면 console 에서 정상적으로 한글을 확인할 수 있습니다. (선택이 되지 않으므로 직접 입력해야 합니다.)



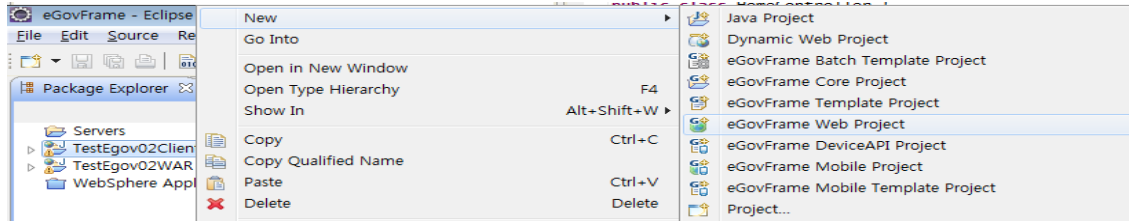
```

IBM J9 VM에서 defaultServer(WebSphere Application Server 8.5.5.2/wlp-1.0.5.cl50220140403-1858) ****, **
[AUDIT ] CWWKE0001I: defaultServer 서버가 실행되었습니다.
[AUDIT ] CWWKZ0058I: 애플리케이션에 대해 dropins(물) 모니터링하는 중입니다.
[AUDIT ] CWWKT0016I: 애플리케이션 시작 가능(default_host): http://localhost:19080/TestWebConsole/
[AUDIT ] CWWKZ0001I: 0.317초 후에 애플리케이션 TestWebConsole이(가) 시작됩니다.
[AUDIT ] CWWKT0016I: 애플리케이션 시작 가능(default_host): http://localhost:19080/TestWeb01/
[AUDIT ] CWWKZ0001I: 0.570초 후에 애플리케이션 TestWeb01이(가) 시작됩니다.
[AUDIT ] CWWKT0016I: 애플리케이션 시작 가능(default_host): http://localhost:19080/web/
[AUDIT ] CWWKZ0001I: 3.203초 후에 애플리케이션 EgovSample01이(가) 시작됩니다.
[AUDIT ] CWWKF0011I: defaultServer 서버가 스마터 플랫폼을 실행할 준비가 되었습니다.
  
```

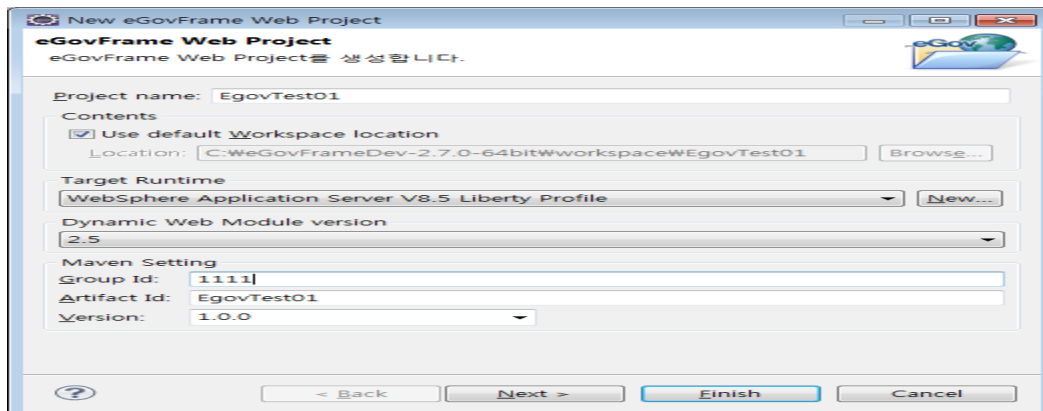
2 전자정부 프레임워크(Spring) 개발을 위한 환경 설정

2.1 전자정부 프레임워크(Spring) 연동 설정

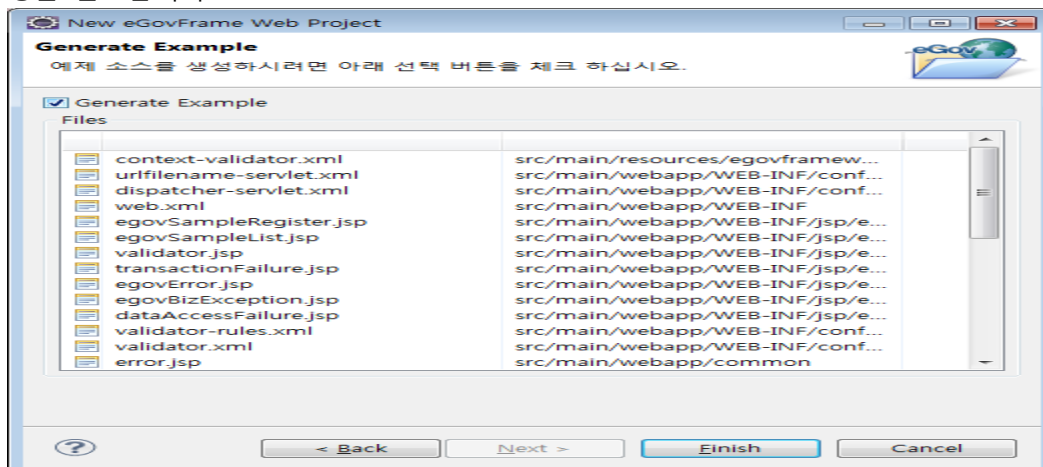
먼저 전자정부 프레임워크 프로젝트를 생성하기 위해서 eGovFrame Web Project 를 하나 생성합니다.



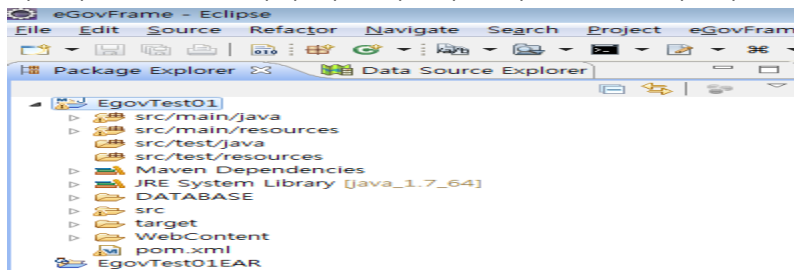
이를 좀 더 자세히 말씀드리면 eGovFrame Web Project 를 생성하기 위해서 해당 메뉴를 클릭하게 되면 하단과 같이 프로젝트 생성 마법사가 나오며 샘플로 사용하기 위해서 기본 값을 입력하여 하단과 같이 생성합니다.



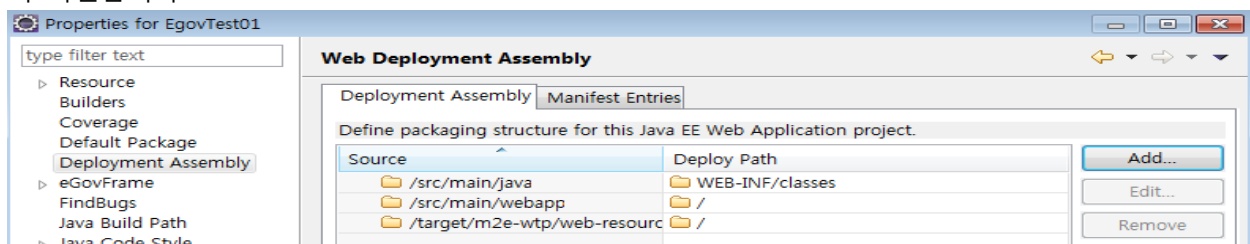
"Next" 버튼을 클릭한 후 다음과 같이 "Generate Example" 을 선택하고 "Finish" 를 클릭하여 샘플 생성을 완료합니다.



이렇게 샘플을 생성하게 되면 하단과 같이 샘플 프로젝트가 생성됩니다.

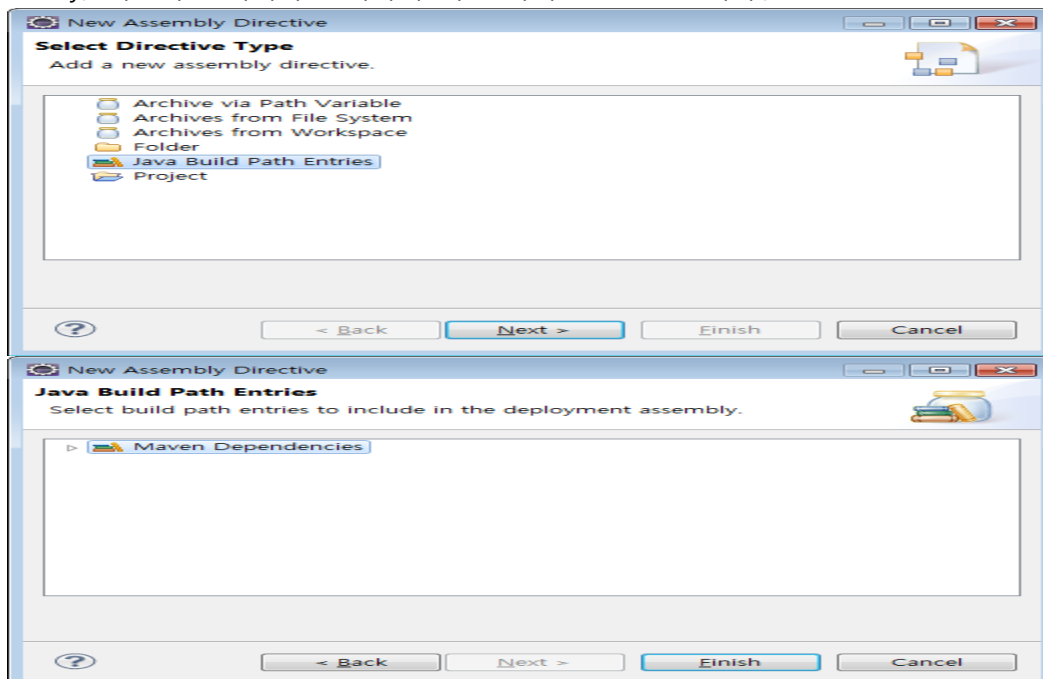


샘플 프로젝트가 생성되었다면 Maven 의 Dependency 를 연결하기 위하여 해당 프로젝트에서 마우스 우 클릭하여 **Properties** 를 선택한 후 **Deployment Assembly** 에 **Maven Dependencies** 가 있는지 확인합니다.



Maven Dependencies 가 만약 설정 되지 않았다면 해당 메뉴에서 **Add** 클릭 후에 **Java Build Path Entries** 를 선택하여 **Maven Dependencies** 를 추가합니다.

(Maven Dependencies 는 Maven 에 설정된 여러 dependencies 들을(spring 을 포함하여 각종 utility library) 배포되는 서버가 인식하기 위한 목적으로 설정합니다.)



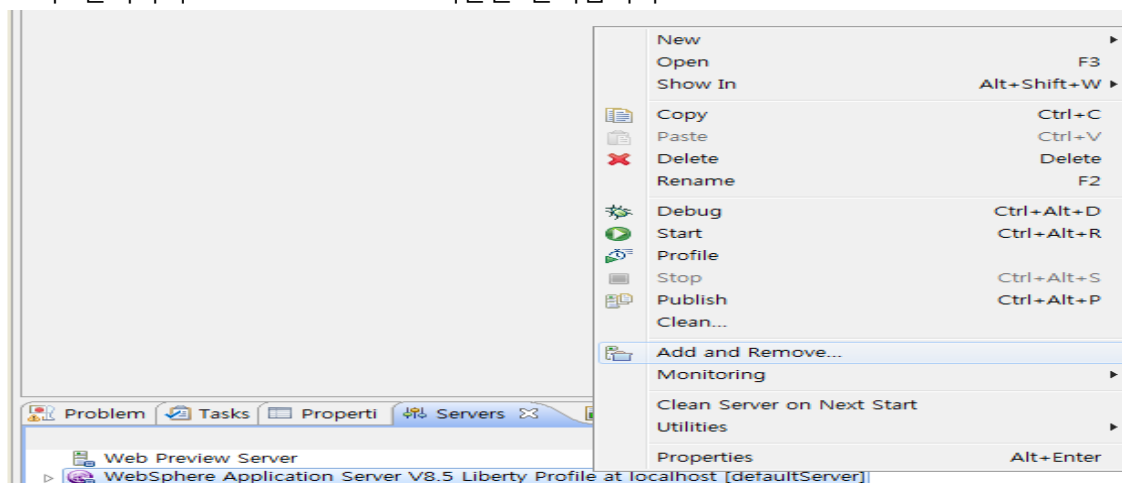
다음으로 maven 에서 IBM WAS Liberty 서버를 target 서버로 인식하고 서버 library 를 참고할 수 있도록 pom.xml 파일에 다음과 같은 dependency 를 추가합니다 .

```
<dependency>
  <groupId>com.ibm.tools.target</groupId>
  <artifactId>was-liberty</artifactId>
  <version>8.5.x.1</version>
  <type>pom</type>
  <scope>provided</scope>
</dependency>
```

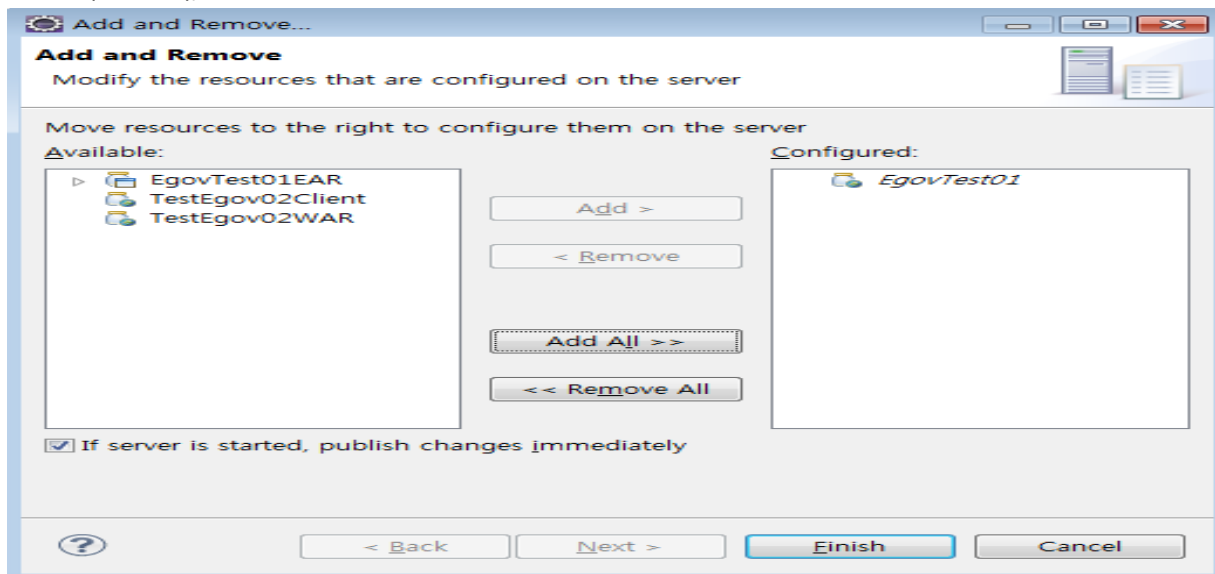
또한, Java 7.0 기반으로 개발하기 위하여 하단과 같이 maven-compiler-plugin 을 1.7 로 변경합니다.

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <configuration>
    <source>1.7</source>
    <target>1.7</target>
    <encoding>UTF-8</encoding>
  </configuration>
</plugin>
```

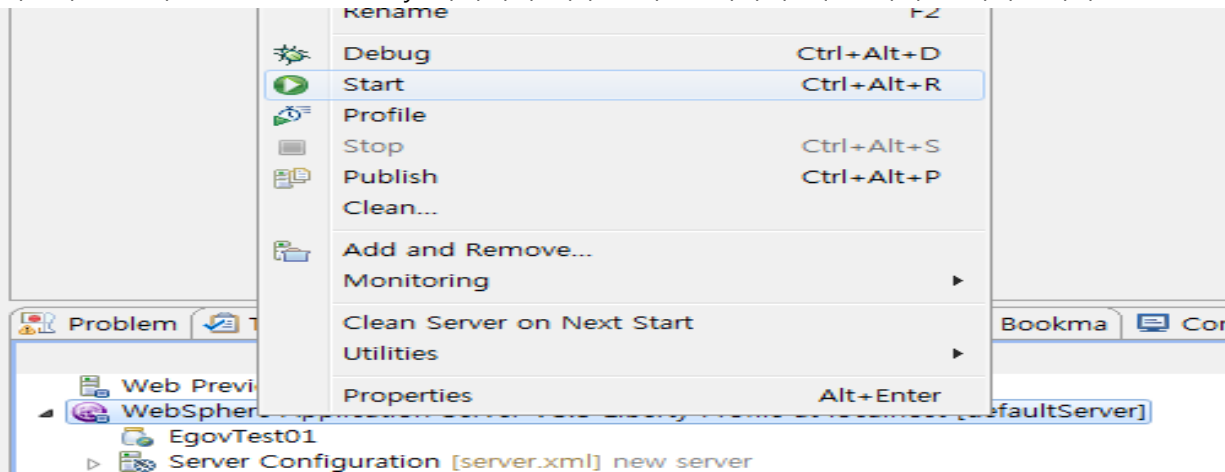
샘플 프로젝트 생성 및 Maven 설정을 완료하였으면 이미 만들어둔 IBM WAS Liberty 서버에서 마우스 우 클릭하여 **Add and Remove** 버튼을 클릭합니다.



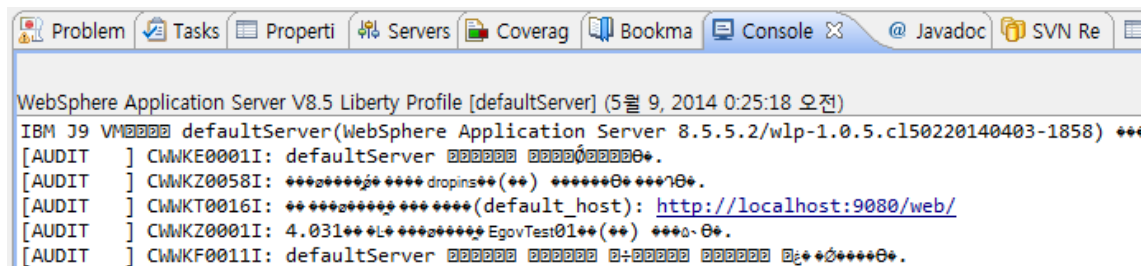
그 후, 샘플 프로젝트를 IBM WAS Liberty 서버에 연동하여 추가합니다. (해당 프로젝트를 선택한 후 **Add** 버튼 클릭)



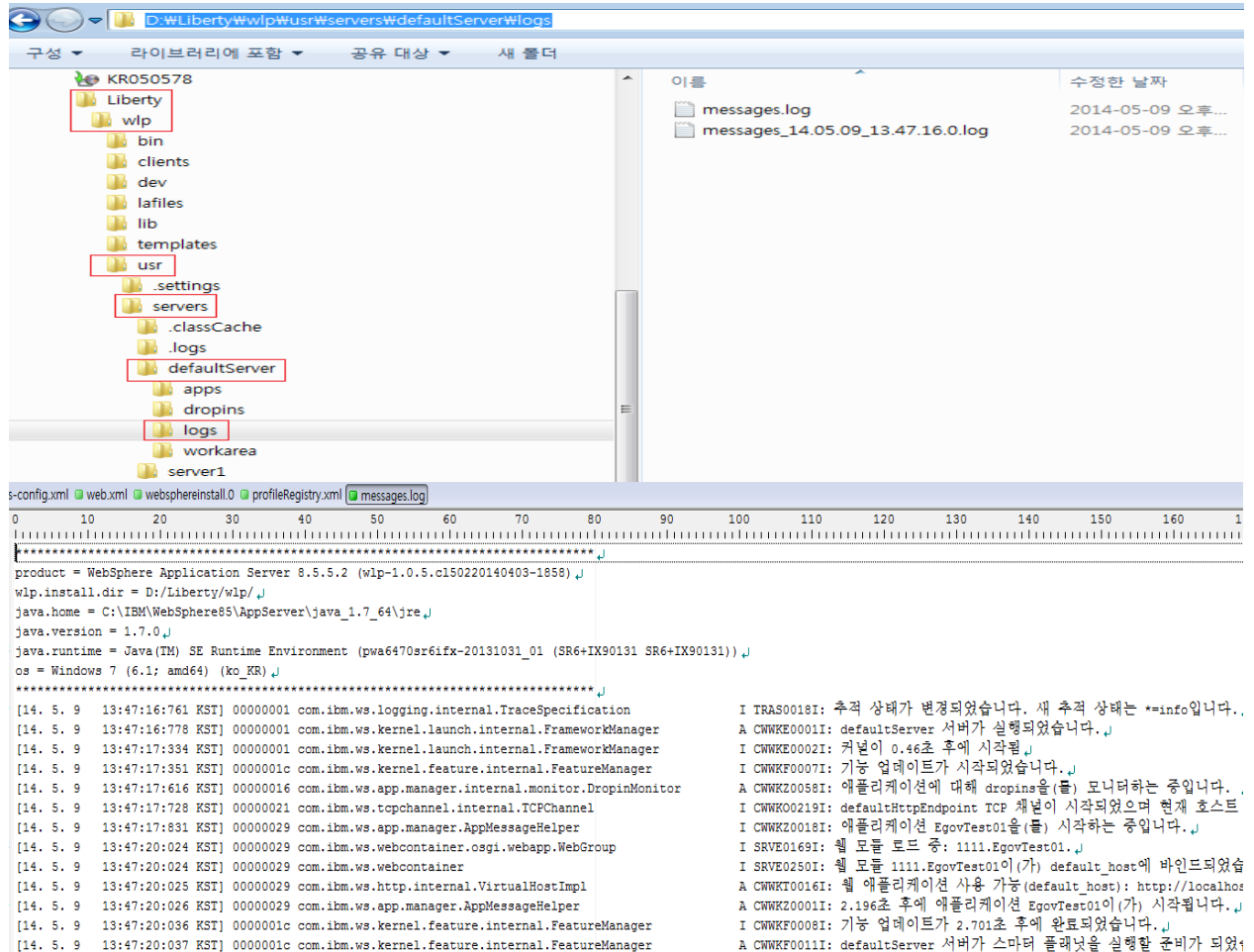
배포가 완료되면 IBM WAS Liberty 서버에서 마우스 우 클릭하여 해당 서버를 시작합니다.



해당 서버가 문제없이 시작된 것을 확인하고 샘플 애플리케이션이 구동된 것을 확인합니다.

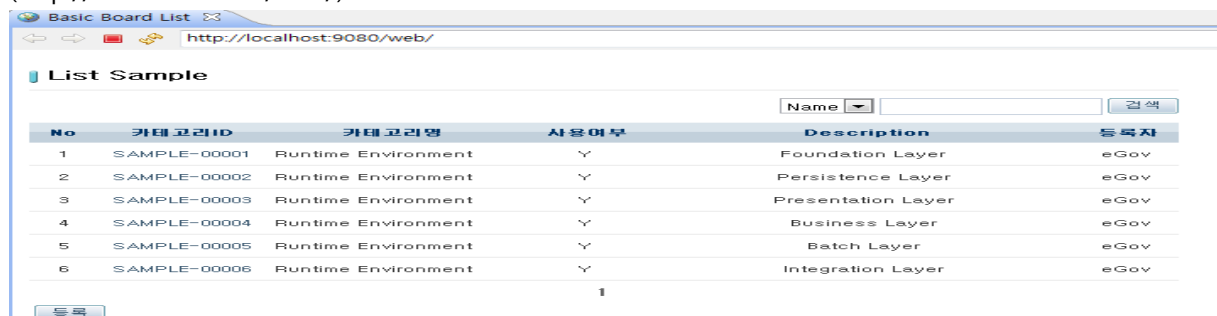


결과를 file 형태로 확인하고자 하신다면 IBM WAS Liberty 서버가 설치된 폴더(D:\Liberty\wlp) 하위의 usr > servers > 서버명(defaultServer) > logs 폴더에서 messages.log 파일을 확인하시면 됩니다.



샘플 페이지를 호출해서 정상적으로 결과가 나오는지 확인합니다.

(http://localhost:9080/web/)

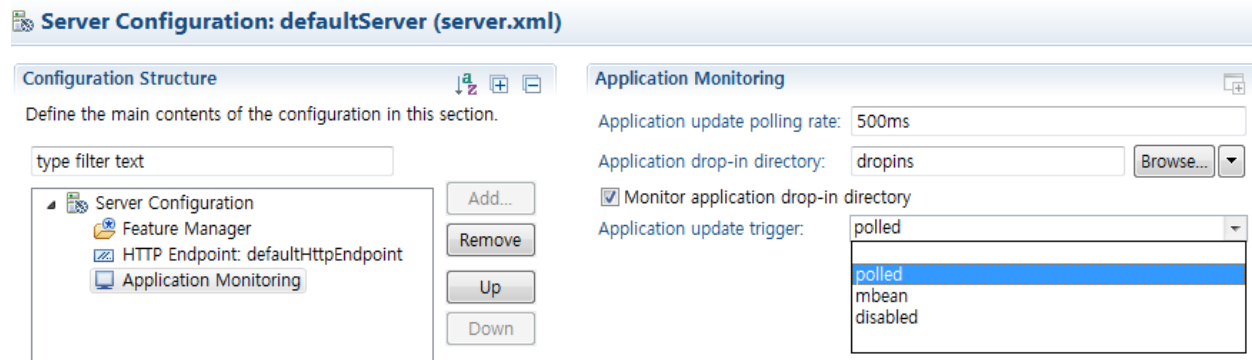


여기까지 무사히 작업하셨다면 IBM WAS Liberty 서버를 이용한 전자정부 프레임워크(Spring) 개발 작업을 위한 연동 작업을 잘 마무리 하신 것입니다. 이제 개발을 진행하신 후 저장 or 빌드를 수행하시면 별도의 작업 없이 변경된 내용을 바로 바로 확인하시면서 개발하실 수 있습니다.

팁 #2) 본 가이드에서 언급한 것과 같은 IBM WAS Liberty 서버에 바로 연결해서 작업하는 것 이외에 만약 Maven 을 이용해서 build 나 install 시에 해당 애플리케이션을 IBM WAS Liberty 서버에 배포하고 확인하는 작업을 수행하시는 것을 선호하신다면 하단과 같이 작업하셔도 동일한 결과를 확인할 수 있습니다.

1안 : Maven 의 pom.xml 설정 변경을 통한 방법

1) server.xml 을 클릭한 후 Application Monitoring 항목에서 Update trigger 항목을 polled 로 변경합니다.



2) pom.xml 에서 tomcat7-maven-plugin 을 주석처리하고 하단과 같이 maven-war-plugin 을 추가합니다. 이때 outputDirectory 는 IBM WAS Liberty 가 설치된 위치/usr/servers/서버명/dropins 폴더 입니다.

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>2.4</version>
  <configuration>
    <warName>${artifactId}-${version}</warName>
    <outputDirectory>D:\weclipses\WeGovFrameDev-2.7.0-64bit_Liberty\wlp\usr\servers\defaultServer\dropins</outputDirectory>
  </configuration>
</plugin>
```

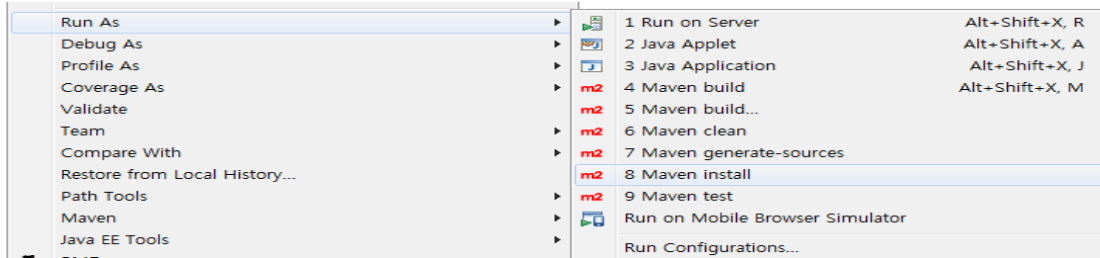


```

<pluginManagement>
  <plugins>
    <!--
      <plugin>
        <groupId>org.apache.tomcat.maven</groupId>
        <artifactId>tomcat7-maven-plugin</artifactId>
        <version>2.2</version>
        <configuration>
          <port>80</port>
          <path>/</path>
          <systemProperties>
            <JAVA_OPTS>-Xms64m -Xmx768m -XX:MaxPermSize=256m</JAVA_OPTS>
          </systemProperties>
        </configuration>
      </plugin>
    -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <warName>${artifactId}-${version}</warName>
        <outputDirectory>D:\EgovDev\wlp\usr\servers\defaultServer\dropins</outputDirectory>
      </configuration>
    </plugin>
  </plugins>
</pluginManagement>

```

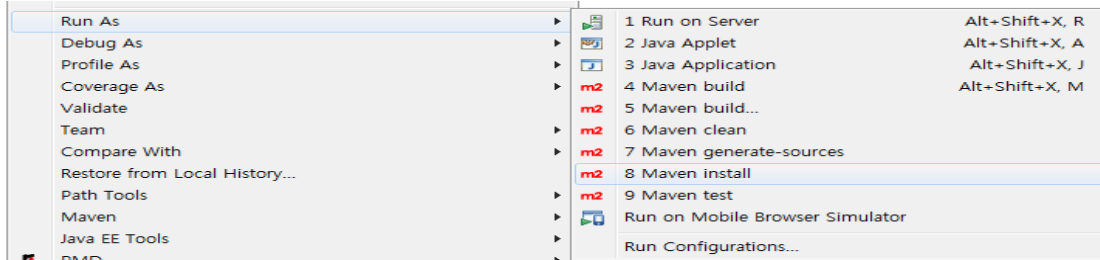
해당 프로젝트를 Run As > Maven install 을 선택하여 build 및 install 을 수행합니다.



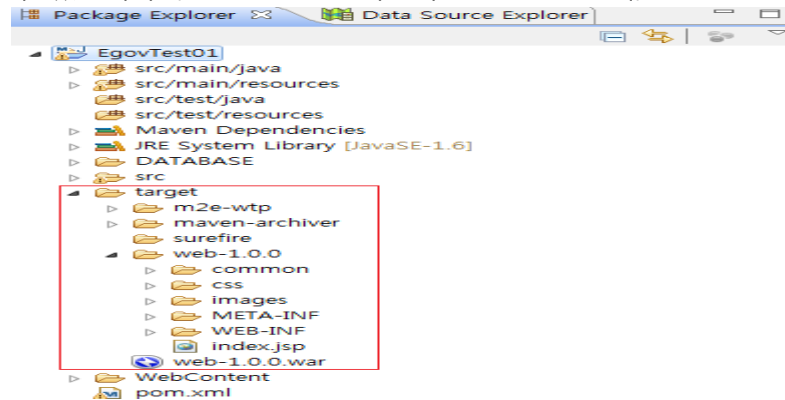
그러면 자동으로 war 파일이 IBM WAS Liberty 서버로 배포되고 이를 동적으로 인식하여 해당 애플리케이션이 업데이트 됩니다.

2안 : Maven install 된 target 폴더를 직접 application 으로 link 거는 방안

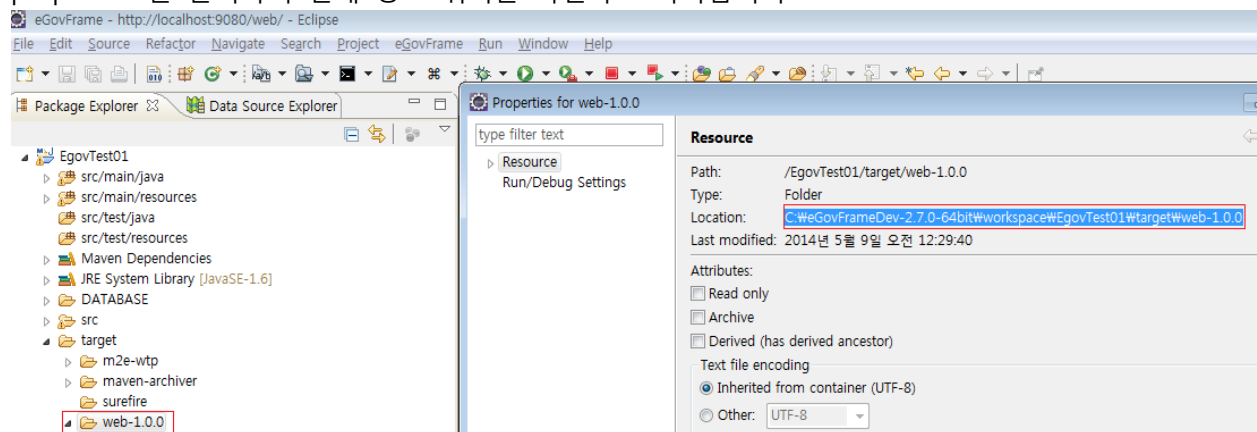
1) 해당 프로젝트를 Run As > Maven install 을 선택하여 build 및 install 을 수행합니다.



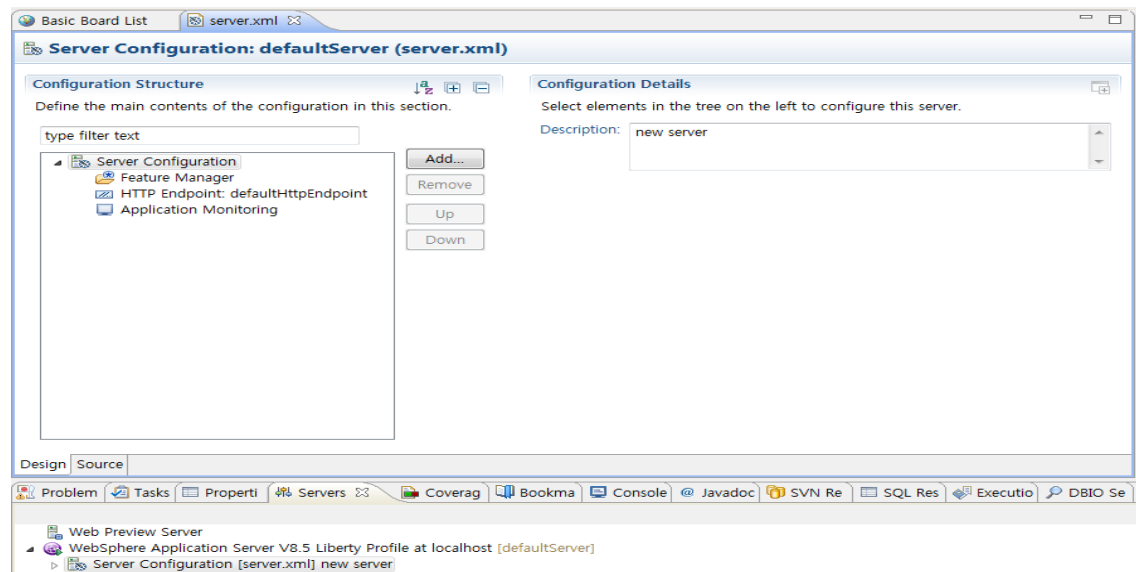
2) 해당 작업이 문제 없이 완료되면 하단과 같이 target 디렉토리에 output 이 생성된 것을 확인할 수 있습니다. (web-1.0.0.war 파일과 web-1.0.0 폴더)



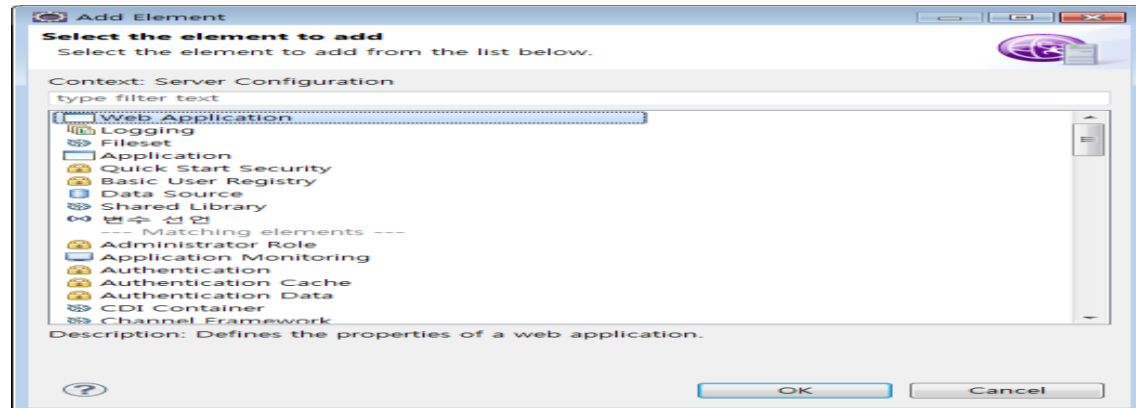
3) 그러면 target 디렉토리(예:Web-1.0.0. 폴더)의 생성된 output 폴더에서 마우스 우 클릭 > properties 를 클릭하여 절대 경로 위치를 확인하고 복사합니다.



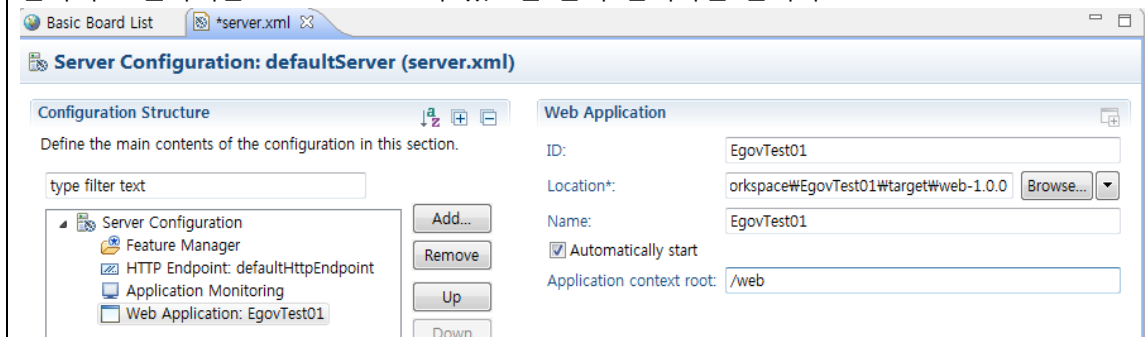
4) 프로젝트의 경로가 확인되었으면 IBM WAS Liberty 서버의 server.xml 을 클릭하여 설정 파일을 확인한 후 **Add** 버튼을 클릭합니다.



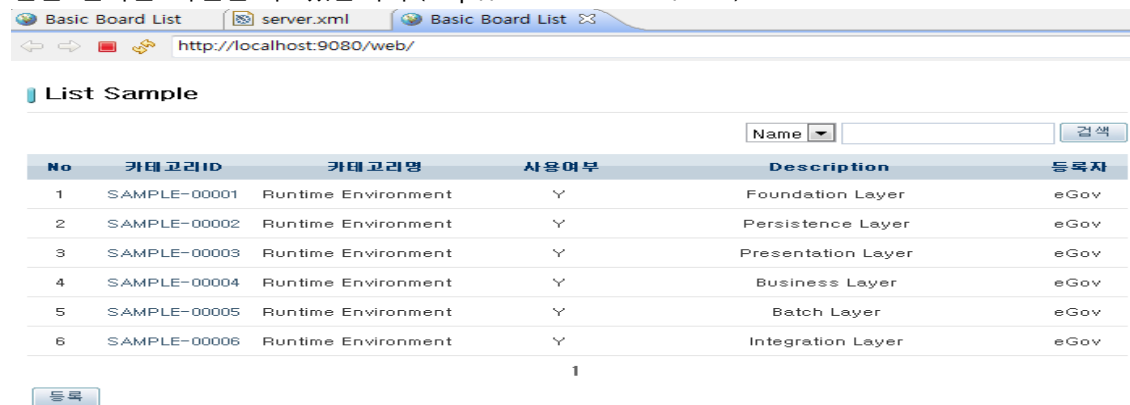
5) 설정 추가 마법사에서 Web Application 을 선택합니다.



6) 선택한 Web Application 에서 설정을 입력합니다. 이때 ID 나 Name 은 원하는 형태로 하시면 되시면 Location 만 이전에 확인했던 target 밑에 생성된 output 폴더의 절대 경로를 입력합니다. 마찬가지로 원하시는 context root 가 있으면 같이 입력하면 됩니다.



7) 이렇게 수행하시고 웹 브라우저를 활용하여 결과를 확인하시면 이전 가이드에서 언급한 것과 동일한 결과를 확인할 수 있습니다.(<http://localhost:9080/web/>)

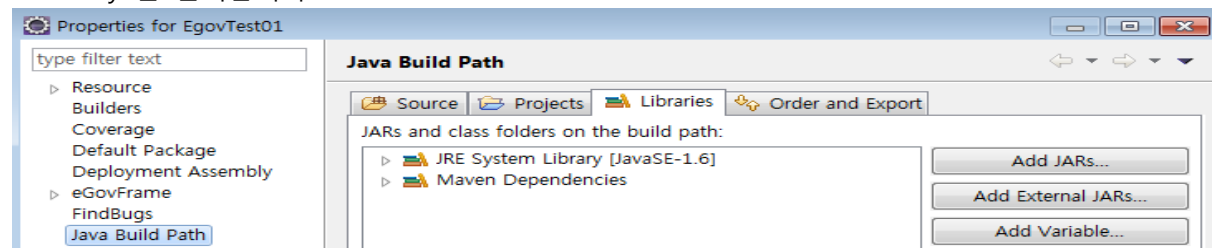


No	카테고리ID	카테고리명	사용여부	Description	등록자
1	SAMPLE-00001	Runtime Environment	Y	Foundation Layer	eGov
2	SAMPLE-00002	Runtime Environment	Y	Persistence Layer	eGov
3	SAMPLE-00003	Runtime Environment	Y	Presentation Layer	eGov
4	SAMPLE-00004	Runtime Environment	Y	Business Layer	eGov
5	SAMPLE-00005	Runtime Environment	Y	Batch Layer	eGov
6	SAMPLE-00006	Runtime Environment	Y	Integration Layer	eGov

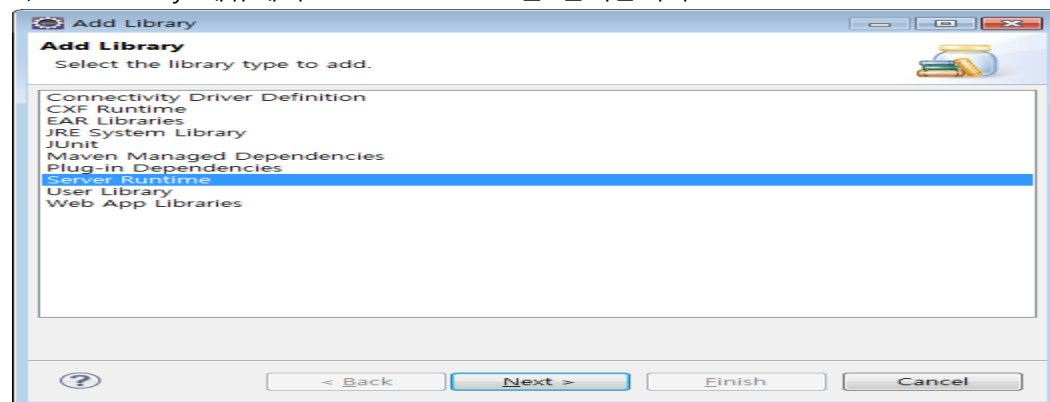
8) 뿐만 아니라 개발 진행 후 Run As > Maven install or build 를 수행하게 되시면 바로 바로 변경된 사항을 IBM WAS Liberty 서버를 통해 반영된 것을 확인할 수 있습니다.

팁 #3) Spring 형태의 프로젝트를 생성한 후 Java EE 와 연관된 library 를 사용해야 할 경우에는 다음과 같이 Java Build Path 를 추가하시면 됩니다.

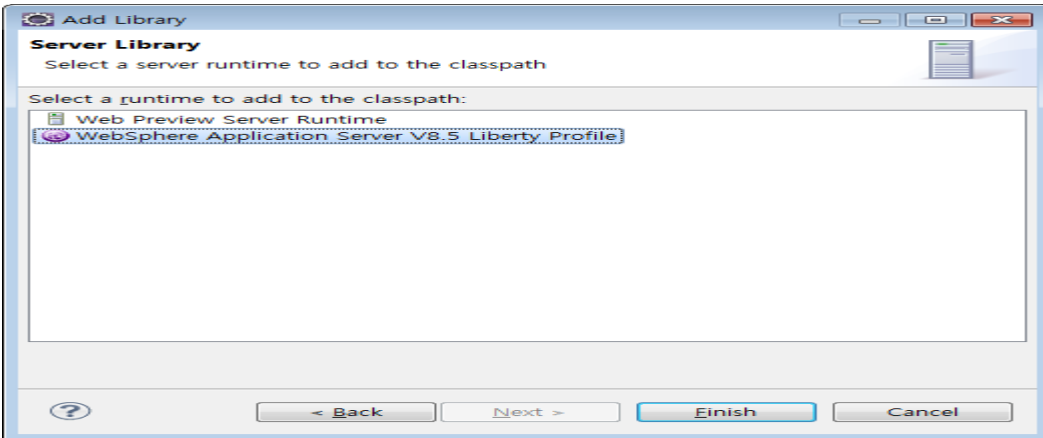
1) 해당 프로젝트에서 마우스 우클릭한 후 properties 를 선택한 후 Java Build path 에서 Add Library 를 클릭합니다.



2) Add Library 메뉴에서 Server Runtime 을 선택합니다.



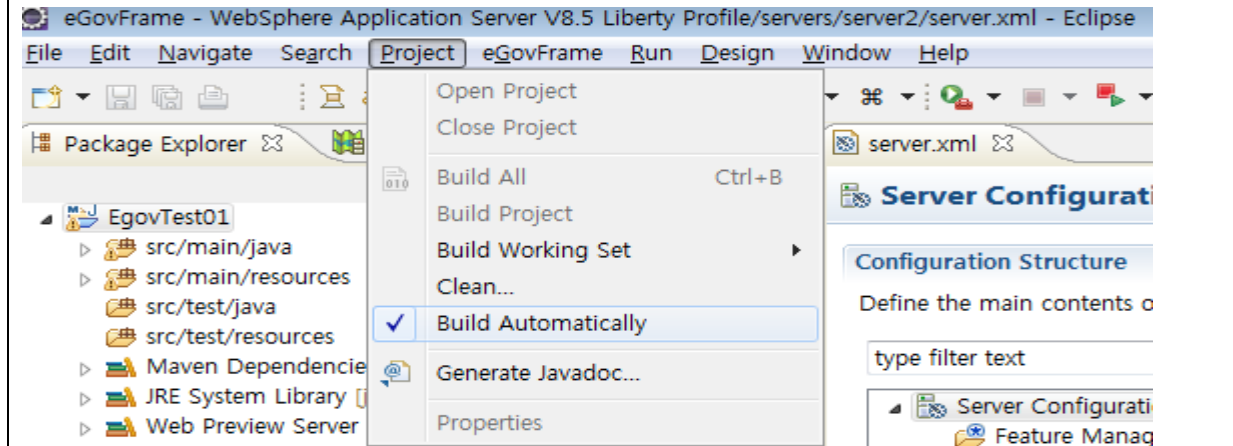
3) Server Library 중에서 WebSphere Application Server v8.5 Liberty Profile 과 Web Preview Server Runtime 을 둘 다 선택해서 추가하시면 됩니다.



또는, Maven 의 pom.xml 파일에 하단과 같이 javaee-api dependency 를 추가해도 동일한 결과를 얻을 수 있습니다.

```
<dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-api</artifactId>
    <version>6.0</version>
</dependency>
```

팁 #4) 만약, 매번 소스 수정을 하고 저장을 할 때 마다 IBM WAS Liberty 서버에 반영되는 것이 번거로우신 분들은 하단과 같이 Project > Build Automatically 를 설정해제 하시게 되면 해당 프로젝트를 Build 할 때만 서버에 결과를 반영하고 확인하실 수도 있습니다.



팁 #5) eclipse 가 느려서 eclipse 튜닝을 원하시는 분들은 하단의 링크를 참고하세요

<http://nguyendangquockhanh.blogspot.kr/2013/08/speed-up-your-eclipse-as-super-fast-ide.html>

또는, 하단과 같이 Windows > Preferences 를 선택한 후 General > Appearance 를 Classic 으로 변경하고 Validation 에 Suspend all validators 를 선택하셔서 validate 기능을 disable 만 해도 속도가 많이 빨라지는 것을 느끼실 것 입니다.

