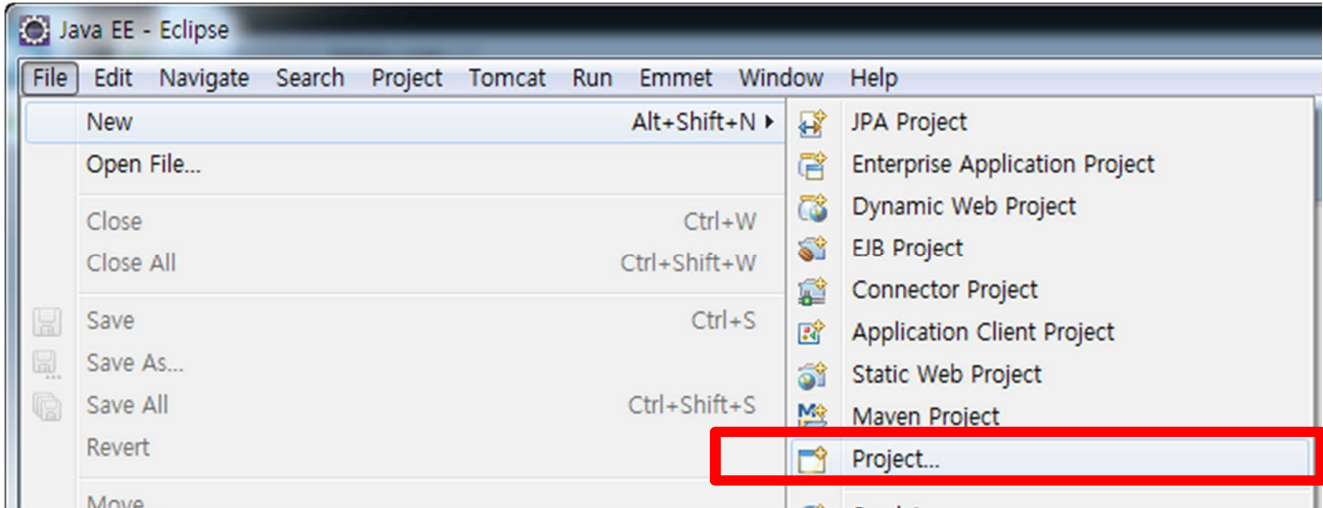


목차

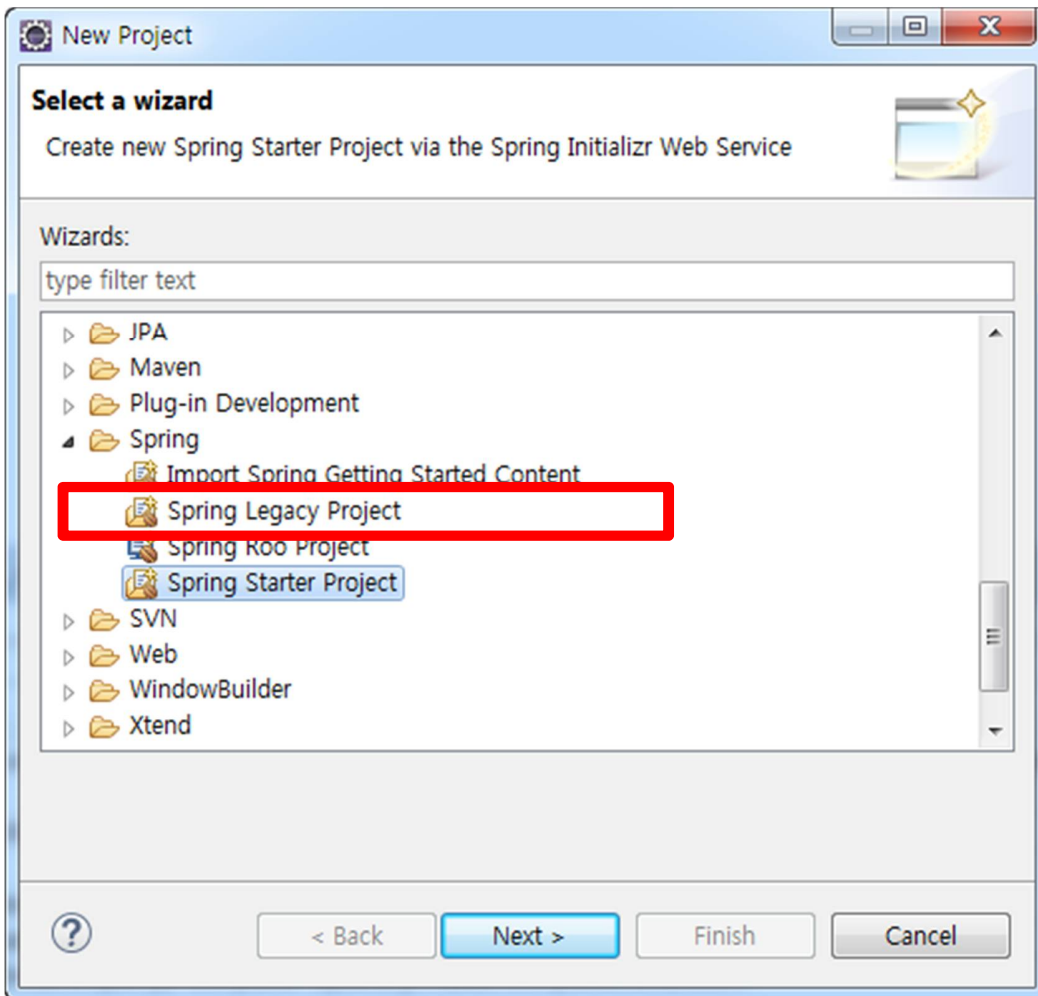
1. 스프링 프로젝트 생성 & 실행	2
2. 스프링 프로젝트의 디렉터리 구조	5
3. Tomcat 서버 설정	6
4. 스프링 웹 프로젝트 설정	7
4.1 web.xml	7
4.2 root-context.xml	9
4.3 servlet-context.xml 수정	9
5. build.gradle 생성	11
6. Converting to Gradle(STS) project	18
7. Gradle(STS) >> Refresh All	18
8. 프로젝트에 Add Gradle Nature 추가	19
9. Context root 수정	20
10. 프로젝트 실행	21
11. 크롬을 기본 브라우저로 설정	22
12. Reference	23

1. 스프링 프로젝트 생성 & 실행

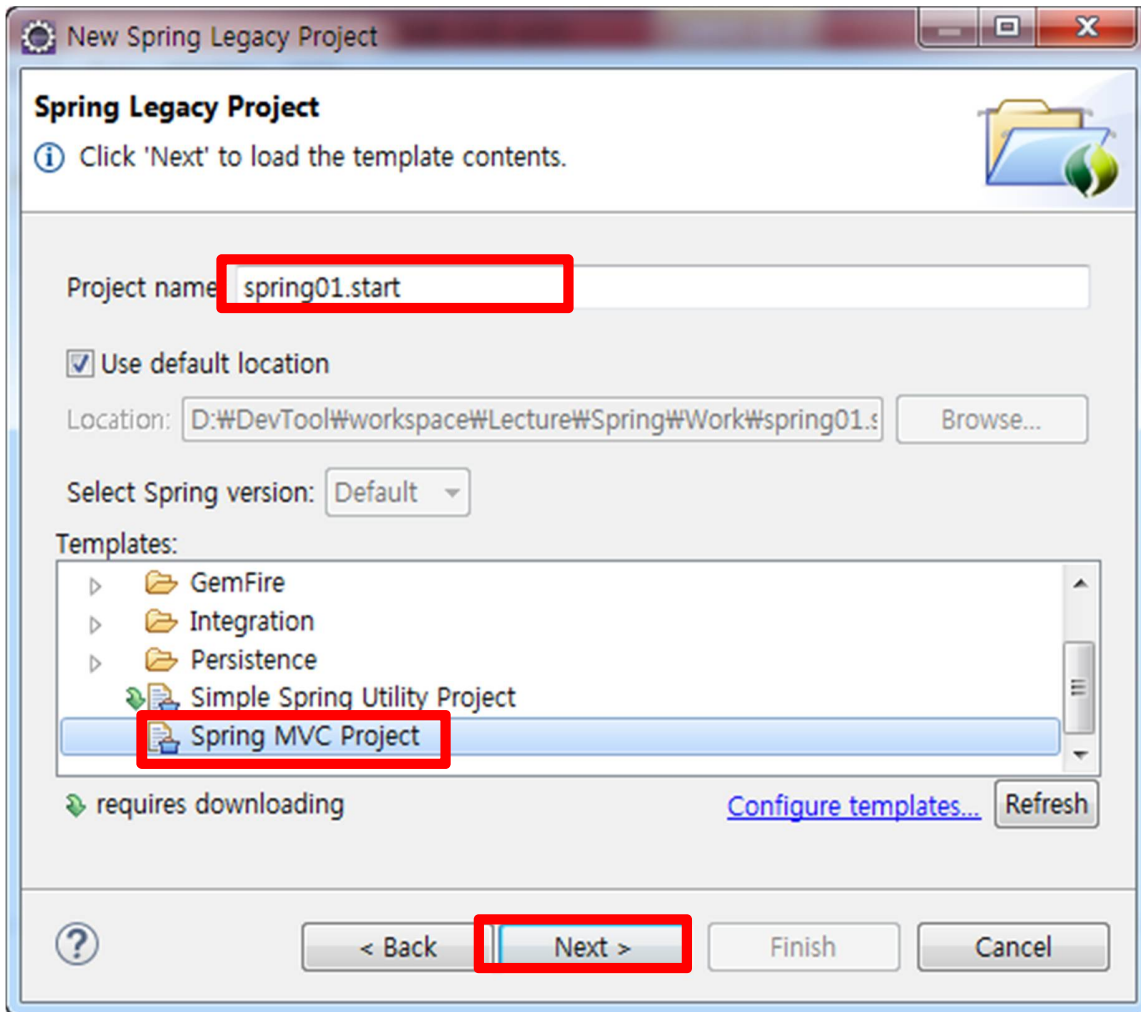
- File > New > Other 를 선택 후 아래와 같이 spring project 를 하나 생성해보자.



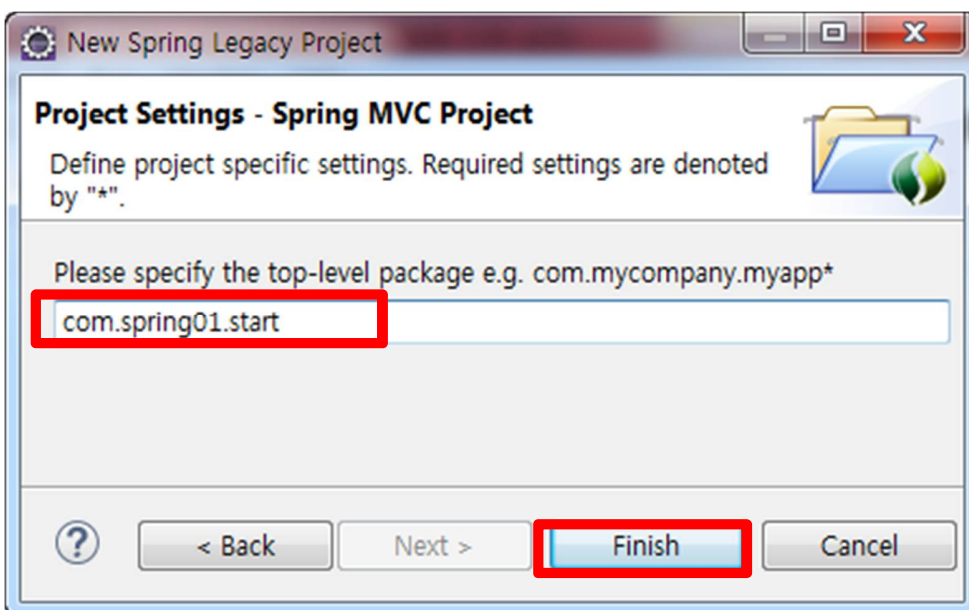
- Sping Legacy Project 를 선택



- Spring Project 를 생성 시 아래를 보면..Spring MVC Project 가 있다.

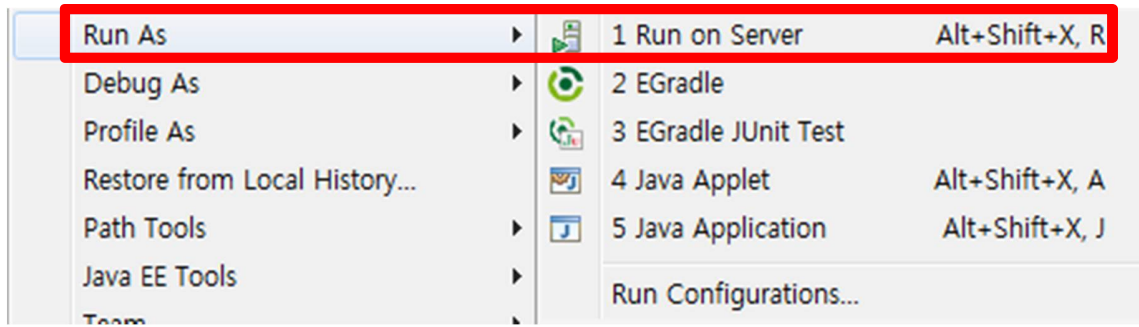


- 패키지 설정

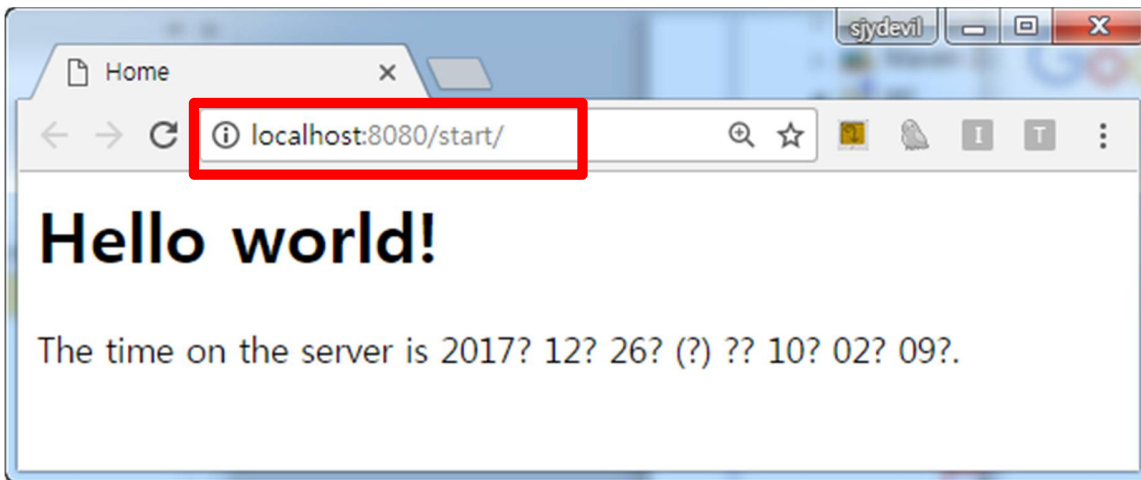


st01.이클립스로 Spring Web 프로젝트 생성

- 서버 실행



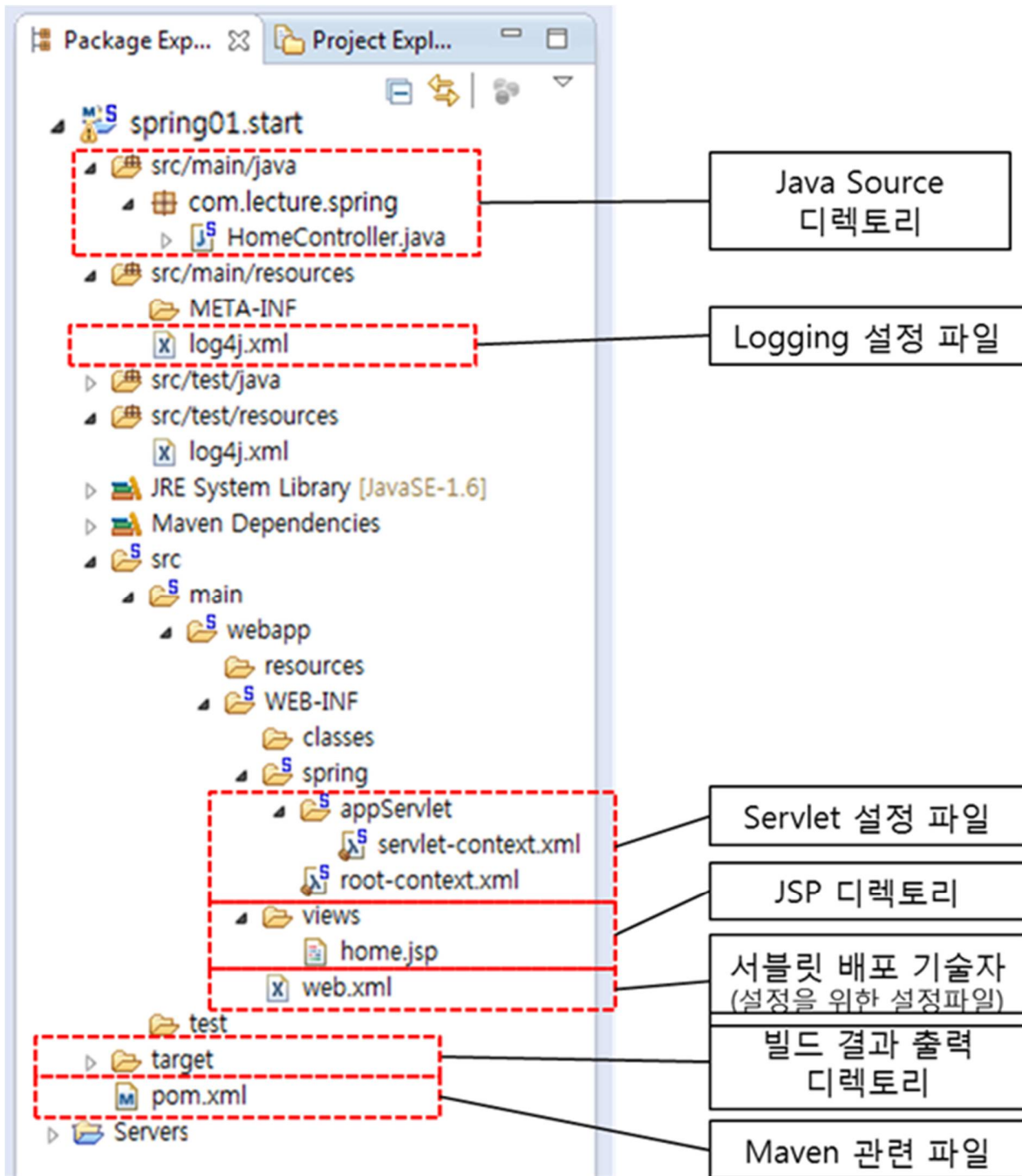
- 실행 화면



- 스프링 실행시 한글 깨짐을 해결하시오.

1. jsp 파일 내에서
2. web.xml 파일에서
3. 톰캣 서버의 server.xml 파일에서.

2. 스프링 프로젝트의 디렉터리 구조



- 스프링 실행시 jsp 파일에서 한글 깨짐을 해결하시오.
 1. jsp 파일 내에서
 2. web.xml 파일에서
 3. 톰캣 서버의 server.xml 파일에서.

3. Tomcat 서버 설정

- {CATALINA_HOME}/conf/server.xml 수정

```
<Connector port="80" protocol="HTTP/1.1"  
            connectionTimeout="20000"  
            redirectPort="8443"  
            URIEncoding="utf-8" />
```

4. 스프링 웹 프로젝트 설정

4.1 web.xml

- src/main/webapp/WEB-INF / 폴더에 있음

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <!-- Spring Web 어플리케이션을 위한 메인 설정파일을 등록한다. -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
  </context-param>

  <!-- Spring Web 어플리케이션 컨테스트를 로딩한다. -->
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>

  <!-- Spring Web 어플리케이션의 맨 앞단 Controller(DispatcherServlet) 를 등록한다. -->
  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>appServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

  <!-- 문자 인코딩 처리 필터 설정 -->
  <filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
    <init-param>
      <param-name>forceEncoding</param-name>
```

```

        <param-value>true</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<!-- -->
<welcome-file-list>
    <welcome-file>index.html</welcome-file>
</welcome-file-list>

```

```

<!-- jsp 파일 utf-8 페이지 인코딩 설정 <%@ page pageEncoding="UTF-8" %> -->
<jsp-config>
    <jsp-property-group>
        <url-pattern>*.jsp</url-pattern>
        <page-encoding>UTF-8</page-encoding>
    </jsp-property-group>
</jsp-config>

```

```

<!-- 에러 페이지 설정 -->
<error-page>
    <error-code>403</error-code>
    <location>/WEB-INF/views/error.jsp</location>
</error-page>

<error-page>
    <error-code>404</error-code>
    <location>/WEB-INF/views/error.jsp</location>
</error-page>

<error-page>
    <error-code>500</error-code>
    <location>/WEB-INF/views/error.jsp</location>
</error-page>

```

```

</web-app>

```


4.2 root-context.xml

- src/main/webapp/WEB-INF/spring/ 폴더에 있음

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

  <!-- Root Context: defines shared resources visible to all other web components -->

</beans>
```

4.3 servlet-context.xml 수정

- src/main/webapp/WEB-INF/spring/appServlet/ 폴더에 있음

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:context="http://www.springframework.org/schema/context"

  xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

  <!-- 스프링의 DispatcherServlet 에게 정적인 자원을 알려준다
  정적자원에는 css 파일, js 파일, 이미지등... -->

  <resources mapping="/resources/**" location="/resources/" />

  <!-- Resolves views : jsp 파일 찾는 경로 설정 -->
  <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
  </beans:bean>
```

st01.이클립스로 Spring Web 프로젝트 생성

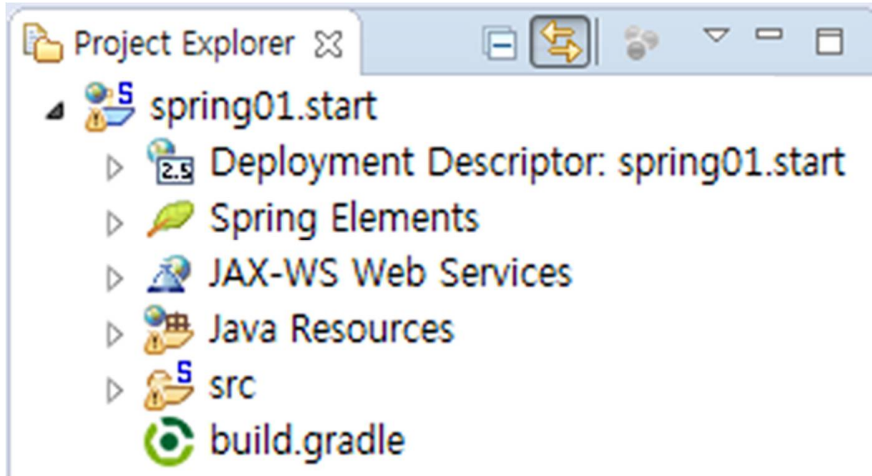
```
<!-- step1. 스프링의 어노테이션을 사용할 수 있도록 하는 설정 -->
<!-- @RequestMapping , @ExceptionHandler 등과 같은 어노테이션을 사용하는 경우 설정해야 함 -->
<annotation-driven />
<context:spring-configured />
<context:annotation-config />

<!-- step2. 어노테이션이 지정된 클래스를 컨테이너에 자동으로 등록되게 하는 설정 -->
<!-- 복수의 패키지를 사용하고 싶은 경우 <context:component-scan> 태그를 여러개 작성 -->
<context:component-scan base-package="com.lecture.spring" />

</beans:beans>
```

5. build.gradle 생성

프로젝트에 build.gradle 을 생성하고 아래의 내용을 복사하여 붙여 넣는다.



```
// tomcat 과 cargo plugin 에 대한 repository 설정입니다.
buildscript {
    repositories {
        jcenter()
    }

    dependencies {
        // Gradle Tomcat plugin
        classpath "com.bmuschko:gradle-tomcat-plugin:2.2.5"

        // Gradle Cargo plugin
        classpath 'com.bmuschko:gradle-cargo-plugin:2.2.3'
    }
}

// Plugin 설정, 만약에 code 에 대한 static analysis 가 필요한 경우에 이곳에 설정.
apply plugin: "java"
apply plugin: "idea"
apply plugin: "groovy"
apply plugin: "eclipse"
//apply plugin: "application"
apply plugin: "eclipse-wtp" // WTP(Web Tools Platform) -> 웹 프로젝트로 인식하도록
apply plugin: "war"
apply plugin: "com.bmuschko.tomcat"
apply plugin: "com.bmuschko.cargo"

//
group = "com.lecture.spring"
version = "1.0.0"
```

```

description = "SpringBoard"
defaultTasks = ["clean", "build", "war", "distZip", "tomcatRun", "installDist"]
def tomcatHome = System.getenv()["CATALINA_HOME"]

// JAVA Version 1.8
compileJava {
    sourceCompatibility = 1.8
    targetCompatibility = 1.8
}

compileTestJava {
    sourceCompatibility = 1.8
    targetCompatibility = 1.8
}

[compileJava, compileTestJava]*.options.collect {options ->
    options.encoding = "UTF-8" // 소스 인코딩 UTF-8 로 지정
    options.debug = true
    options.deprecation = true
    options.compilerArgs.add "-Xlint"
    options.compilerArgs.add "-Xlint:-unchecked"
    options.compilerArgs.add "-Xlint:-serial"
    options.compilerArgs.add "-Xlint:-rawtypes"
}

eclipse {
    wtp {
        facet {
            facet name: "jst.web" , version: "2.5" // Servlet Spec Version 지정, 미 지정시 2.4
            facet name: "jst.java", version: "1.8" // Java Version 지정
        }
    }

    classpath {
        containers.remove("org.eclipse.jdt.launching.JRE_CONTAINER")
        containers
        "org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-1.8"
    }

    //exclude "log4j.properties"
}

```

```
// 메이븐 Central 저장소 사용
repositories {
    // Use "jcenter" for resolving your dependencies.
    // You can declare any Maven/Ivy/file repository here.
    mavenCentral()
    jcenter()

    maven { url "http://mesir.googlecode.com/svn/trunk/mavenrepo" }
    maven { url "https://raw.githubusercontent.com/slorber/gcm-server-
repository/master/releases/" }
    maven { url "http://repo.spring.io/release" }
    maven { url "http://repo.spring.io/milestone" }
    maven { url "https://oss.sonatype.org/content/repositories/releases" }
}

configurations {
    providedRuntime
    providedCompile

    // logback(slf4j)를 사용하기 때문에 모든 의존성에서 commons-logging 는 제외
    //compile.exclude group: "commons-logging", module: "commons-logging"
}

// dependency 버전 정보
def versions = [
    spring          : "4.1.9.RELEASE"
    , security      : "3.2.8.RELEASE"
    , webflow       : "2.4.1.RELEASE"
    , junit         : "4.12"
    , servletApi    : "3.1.0"
    , jstl          : "1.2"
    , slf4j         : "1.7.21"
    , mockito       : "1.9.0"
    , cglib         : "2.2.2"
    , groovy        : "2.2.1"
    , fasterxml     : "2.3.1"
    , aspectj       : "1.8.6"
    , springSpock   : "0.7-groovy-2.0"
    , tiles         : "3.0.5"
    , hibernate     : "4.2.20.Final"
    , tomcatVersion : "7.0.73"
    , cargoVersion  : "1.4.5"
    //, logback      : "1.0.13"
]

// 의존성 설정
dependencies {
    compile fileTree(dir: "lib", include: ["*.jar"])
}
```

```

// Apache Commons Lang, a package of Java utility classes for the classes that are in
java.lang hierarchy,
// or are considered to be so standard as to justify existence in java.lang.
compile group: "org.apache.commons" , name: "commons-lang3" , version: "3.4"
compile group: "org.apache.commons" , name: "commons-dbcp2" , version: "2.0"
compile group: "commons-beanutils" , name: "commons-beanutils" , version: "1.9.2"
compile group: "commons-codec" , name: "commons-codec" , version: "1.10"
compile group: "commons-collections" , name: "commons-collections" , version: "3.2"
compile group: "commons-pool" , name: "commons-pool" , version: "1.6"

// log 라이브러리
compile group: "org.slf4j" , name: "slf4j-api" , version:
"${versions.slf4j}"
runtime group: "org.slf4j" , name: "slf4j-log4j12" , version:
"${versions.slf4j}"
runtime group: "org.slf4j" , name: "jcl-over-slf4j" , version:
"${versions.slf4j}"
runtime group: "log4j" , name: "log4j" , version: "1.2.17"

// log4jdbc library
compile group: "com.googlecode.log4jdbc" , name: "log4jdbc" , version: "1.2"

// spring framework 라이브러리
compile group: "org.springframework" , name: "spring-core" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-beans" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-context" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-webmvc" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-jdbc" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-tx" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-aop" , version:
"${versions.spring}"
runtime group: "org.springframework" , name: "spring-aspects" , version:
"${versions.spring}"

//
compile group: "org.springframework" , name: "spring-context-support" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-websocket" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-messaging" , version:
"${versions.spring}"
compile group: "org.springframework" , name: "spring-orm" , version:

```

```

"${versions.spring}"
    compile group: "org.springframework", name: "spring-expression" , version:
"${versions.spring}"

    // springframework.security
    compile group: "org.springframework.security", name: "spring-security-core" , version:
"${versions.security}"
    compile group: "org.springframework.security", name: "spring-security-web" , version:
"${versions.security}"
    compile group: "org.springframework.security", name: "spring-security-config", version:
"${versions.security}"

    // springframework.webflow
    compile group: "org.springframework.webflow", name: "spring-js" , version:
"${versions.webflow}"
    compile group: "org.springframework.webflow", name: "spring-webflow", version:
"${versions.webflow}"

    // aspectj 라이브러리: AOP 와 연관이 있는 라이브러리
    compile group: "org.aspectj" , name: "aspectjrt" , version: "${versions.aspectj}"
    compile group: "org.aspectj" , name: "aspectjweaver" , version: "${versions.aspectj}"
    compile group: "org.aspectj" , name: "aspectjtools" , version: "${versions.aspectj}"

    // mybatis 라이브러리
    compile group: "org.mybatis", name: "mybatis" , version: "3.2.8"
    compile group: "org.mybatis", name: "mybatis-spring", version: "1.2.2"

    // mysql jdbc 드라이버
    compile group: "mysql", name: "mysql-connector-java", version: "5.1.34"

    // 파일업로드 할 때 사용.
    compile group: "commons-fileupload", name: "commons-fileupload" , version: "1.2.1"
    compile group: "commons-io" , name: "commons-io" , version: "2.4"

    // JSR 330 JAR 를 포함하기 위한 라이브러리. @Inject, @Named 어노테이션 사용 가능
    compile group: "javax.inject" , name: "javax.inject" , version: "1"
    compile group: "javax.persistence" , name: "persistence-api" , version: "1.0.2"

    // JSP Tag 를 위한 라이브러리.
    compile group: "javax.servlet" , name: "jstl" , version: "${versions.jstl}"
    compile group: "javax.servlet.jsp.jstl" , name: "jstl-api" , version: "${versions.jstl}"
    compile group: "taglibs" , name: "standard" , version: "1.1.2"

    // TDD 를 위한 JUnit 라이브러리.
    testCompile group: "junit", name: "junit", version: "${versions.junit}"
    testCompile group: "org.springframework", name: "spring-test", version: "${versions.spring}"
    testCompile group: "org.mockito", name: "mockito-core", version: "${versions.mockito}"

```

```

//
runtime group: "javax.servlet"      , name: "javax.servlet-api", version:
"${versions.servletApi}"
runtime group: "javax.servlet.jsp", name: "javax.servlet.jsp-api", version: "2.3.1"
runtime group: "org.apache.tomcat", name: "tomcat-servlet-api", version:
"${versions.tomcatVersion}"

// tomcat plugin 설정입니다.
tomcat group: "org.apache.tomcat.embed", name: "tomcat-embed-core", version:
"${versions.tomcatVersion}"
tomcat group: "org.apache.tomcat.embed", name: "tomcat-embed-logging-juli", version:
"${versions.tomcatVersion}"
tomcat(group: "org.apache.tomcat.embed", name: "tomcat-embed-jasper", version:
"${versions.tomcatVersion}") {
    exclude group: "org.eclipse.jdt.core.compiler", module: "ecj"
}

// cargo 에 대한 설정입니다.
cargo group: "org.codehaus.cargo", name: "cargo-core-uberjar", version:
"${versions.cargoVersion}"
cargo group: "org.codehaus.cargo", name: "cargo-ant"          , version:
"${versions.cargoVersion}"
}

// main java 소스셋의 디렉토리 추가
sourceSets {
    main {
        java.srcDirs = ["src/main/java" ]
        resources.srcDirs = ["src/main/resources"]
    }
}

assemble {
    //dependsOn taskCopy
}

idea {
    module {
        sourceDirs += file("src/main/java")
    }
}

// TEST 설정
test {
    jvmArgs = ["-ea", "-Xmx256m"]
    logging.captureStandardOutput(LogLevel.INFO)
}

```



```

// testReport = false

systemProperties "property": "value"

testLogging {
    events "started", "passed"
}
}

war {
    baseName = "ROOT"
    version = "${new Date().format('yyyyMMdd')}}"
}

// 프로젝트 초기화
// 1. java source directory 생성 : src/main/java, src/test/java
// 2. resource directory 생성 : src/main/resource, src/test/resource
// 3. web source directory 생성 : src/main/webapp, src/main/webapp/WEB-INF
task initProject(description: "initialize project") {

    sourceSets*.java.srcDirs*.each { it.mkdirs() }
    sourceSets*.resources.srcDirs*.each { it.mkdirs() }

    //create webapp directory
    def webappDir = "$rootDir/src/main/webapp"
    file(webappDir).mkdir()

    //Create the WEB-INF directory
    file(webappDir + "/WEB-INF").mkdir()

    //Create the META-INF directory
    file(webappDir + "/META-INF").mkdir()
}

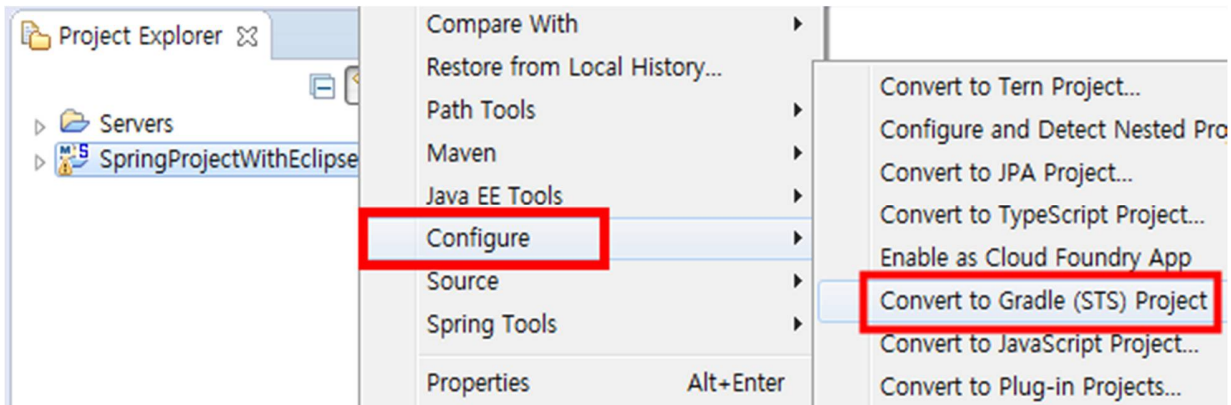
task taskBuild() {
    println "taskBuild!!!"

    //commandLine "cmd", "/c", "gradle build"

    if(war != null){
        println "war path :: " + war.archivePath
        println "war name :: " + war.archiveName
    }
}

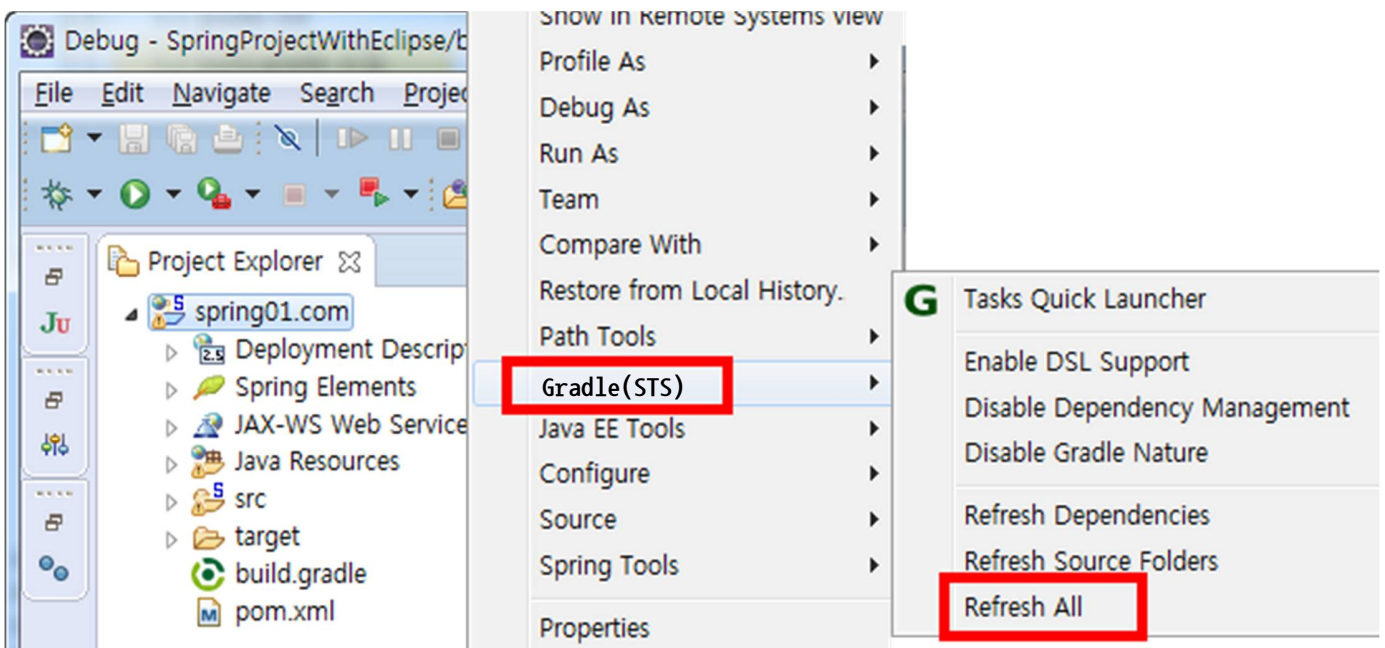
```

6. Converting to Gradle(STS) project

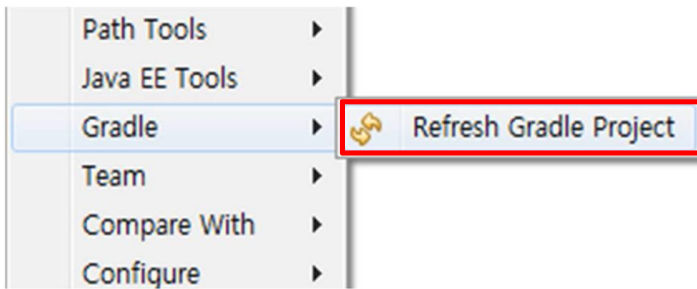
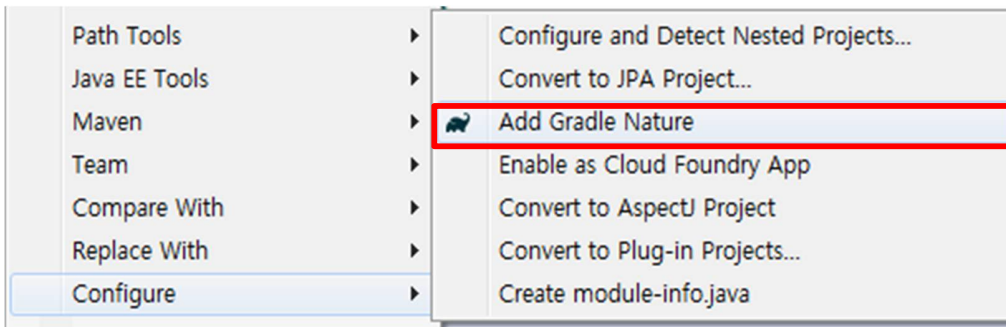


7. Gradle(STS) >> Refresh All

build.gradle 이 수정되면 반드시 "Refresh All" 을 실행 해야 한다.



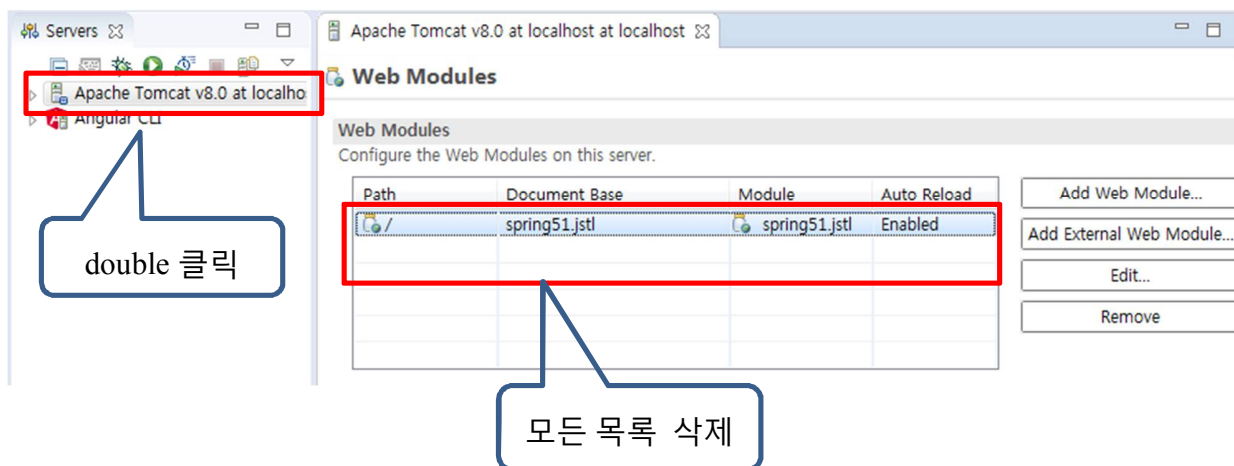
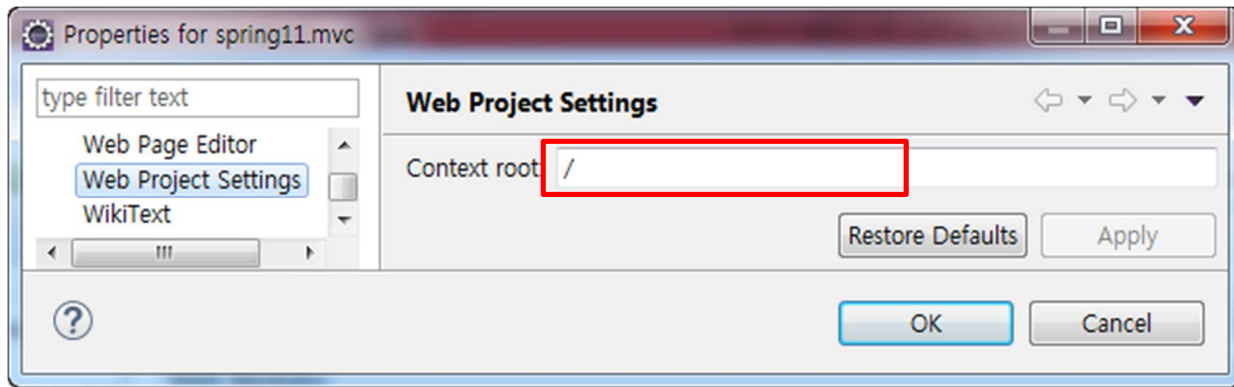
8. 프로젝트에 Add Gradle Nature 추가



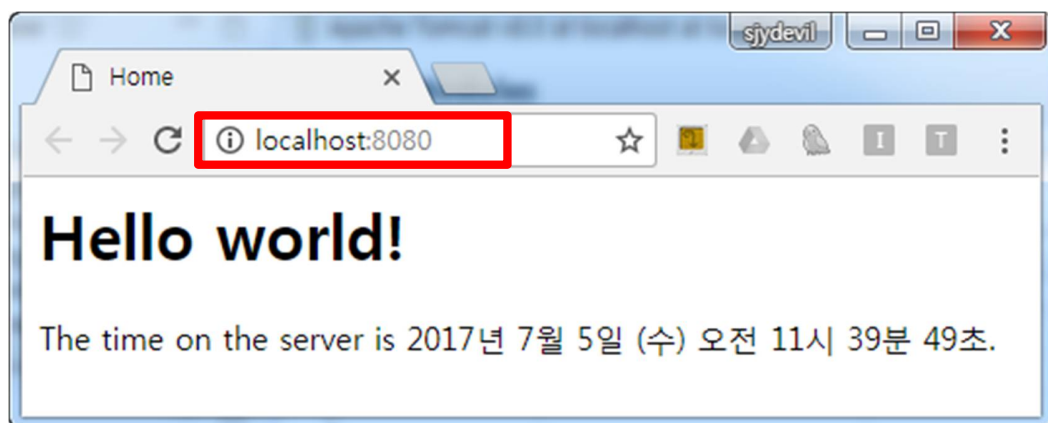
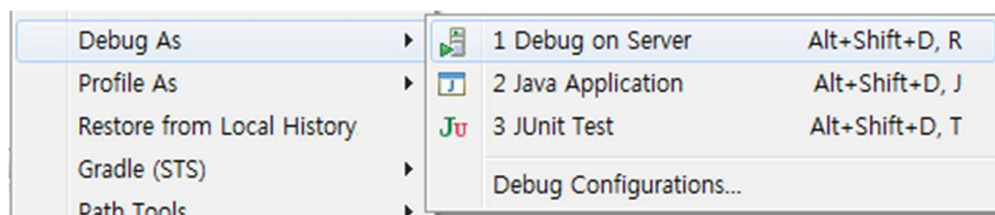
9. Context root 수정

이클립스에서 프로젝트 선택 후 오른쪽 마우스 클릭하여 "Properties" 를 선택한다.

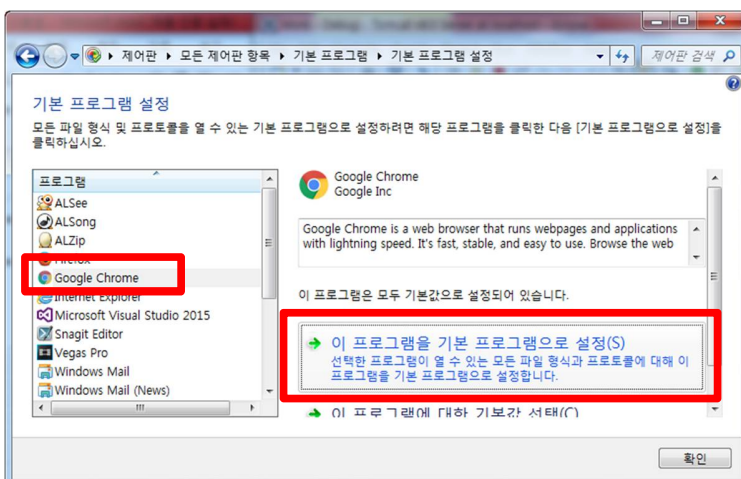
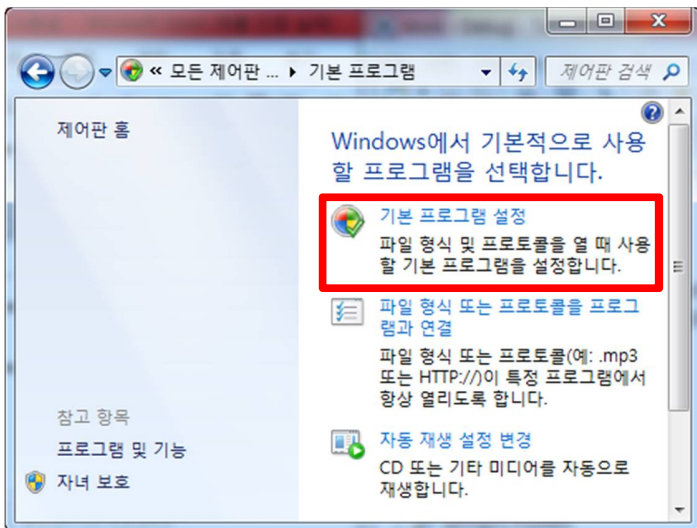
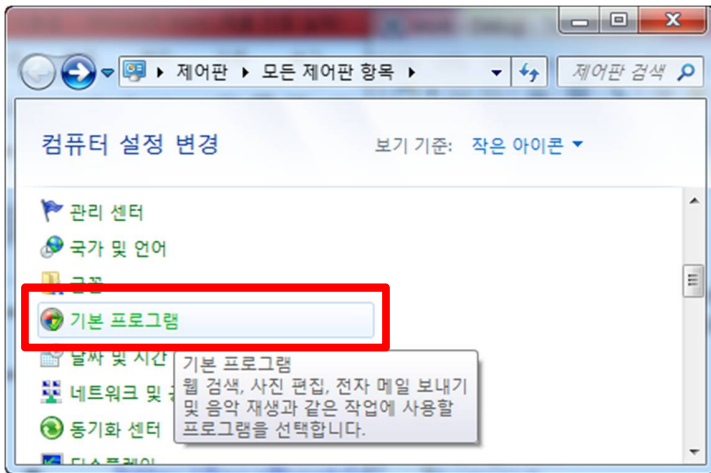
"Properties" 창에서 "Web Project Settings" 메뉴를 클릭하여 "Context root"를 아래와 같이 바꾼다.



10. 프로젝트 실행



11. 크롬을 기본 브라우저로 설정



12. Reference

<http://www.jayway.com/2013/05/12/getting-started-with-gradle/>

<http://www.slipp.net/wiki/pages/viewpage.action?pageId=12878060>

[https://www.credera.com/blog/custom-application-development/converting-spring-boot-project-maven-gradle-
sts/](https://www.credera.com/blog/custom-application-development/converting-spring-boot-project-maven-gradle-
sts/)

<http://stackoverflow.com/questions/13925724/providedcompile-without-war-plugin>

<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/transaction.html>

<http://hellogk.tistory.com/94>

<http://hellowk1.blogspot.kr/2014/02/spring-framework-transaction-aop.html>

<http://hellowk1.blogspot.kr/2015/03/spring-framework-transaction-with.html>

<http://barunmo.blogspot.kr/2013/06/mybatis.html>

<http://egloos.zum.com/springmvc/v/499291>

<http://blog.outsider.ne.kr/870>