



2018 JANUARY 12

코드로 배우는 스프링 웹 프로젝트

SPRING STUDY

SEOKRAE KIM



내용

| | |
|-----------------------------------|----|
| I. 스프링 웹 프로젝트 스터디..... | 1 |
| 1. 프로젝트의 기본 구조 구성..... | 1 |
| 1) 개발 환경의 설정..... | 1 |
| 2) MySQL의 설정과 스프링 테스트..... | 8 |
| 3) 스프링 + MyBatis + MySQL의 설정..... | 9 |
| 4) 모델 2 방식과 스프링 MVC | 14 |
| 5) 스프링 + MyBatis..... | 20 |
| 6) 정리 | 27 |

I. 스프링 웹 프로젝트 스터디

본 문서는 “코드로 배우는 스프링 웹 프로젝트”책의 내용을 기반으로 하여 라이브러리의 필요성과 사용 목적, 코딩실습에 관련된 내용만이 작성되었습니다.

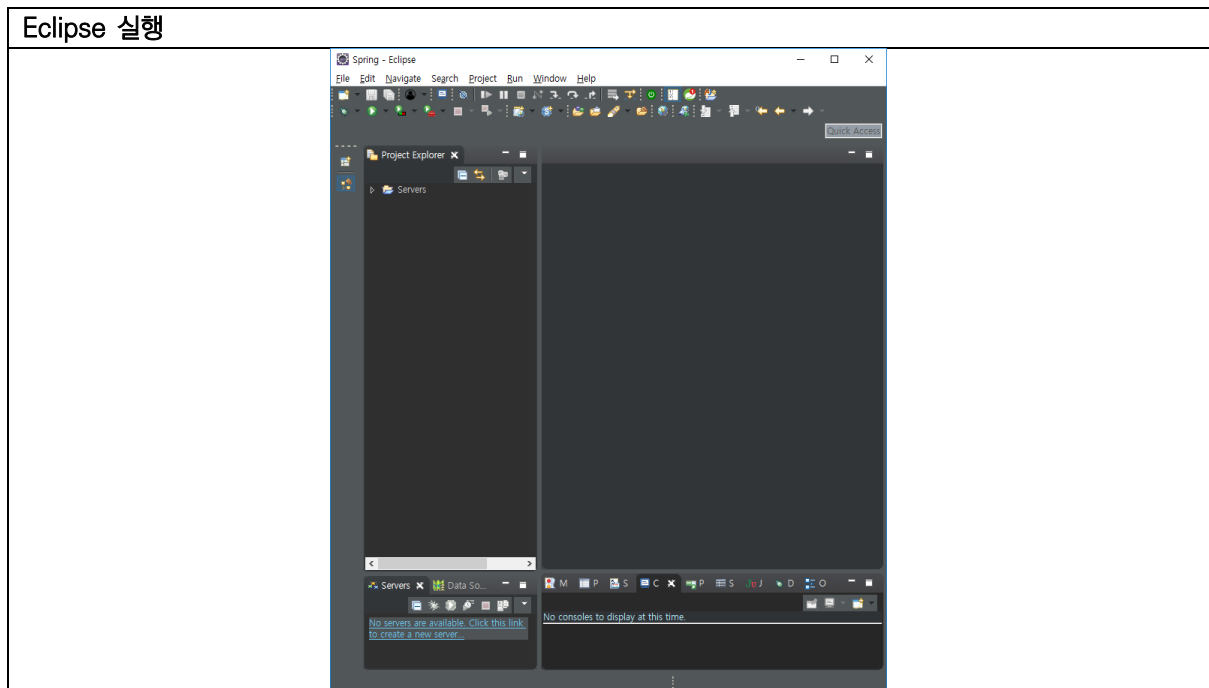
스프링의 개념과 이론적인 부분에 대해서는 Spring Study Theory를 참고해주세요.

1. 프로젝트의 기본 구조 구성

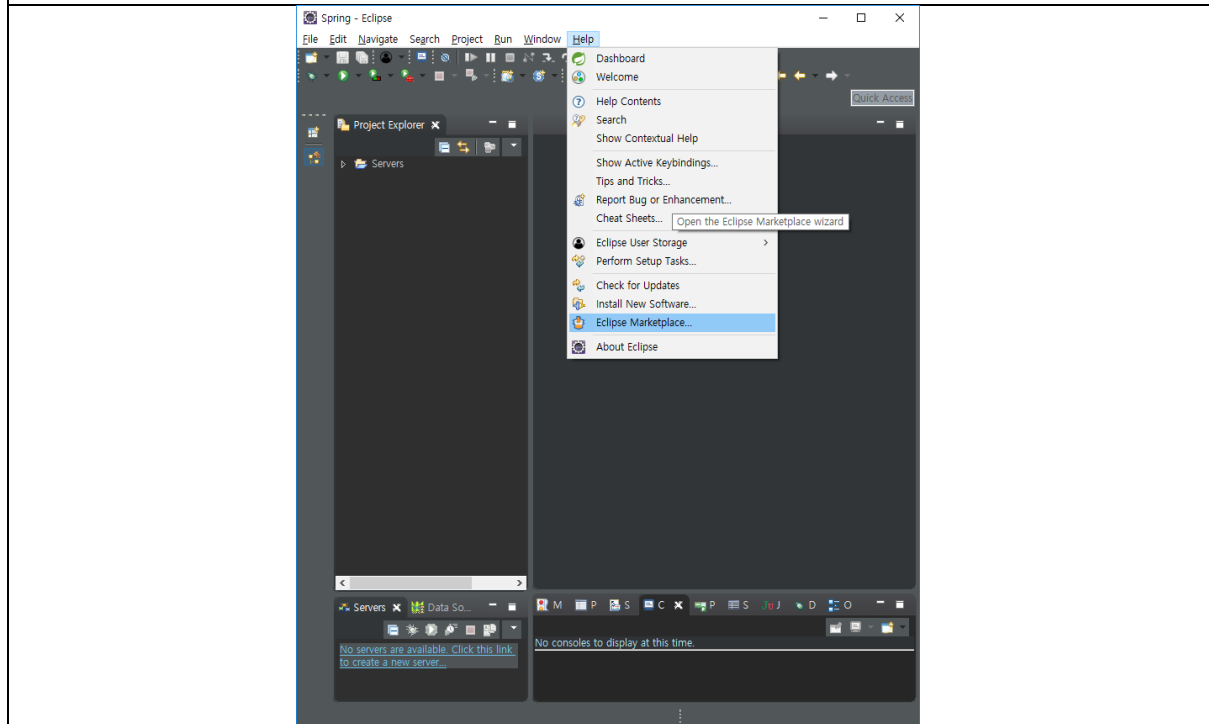
“프로젝트의 기본 구조 및 구성”에서는 Spring Project를 위해서 “프로젝트 생성”, “라이브러리의 초기화”, “프로젝트 실행” 순으로 진행하여 **스프링이 실행되는 것을 확인**하고, **스프링 프로젝트의 폴더 구조를 분석**하는 것을 목표로 한다.

1) 개발 환경의 설정

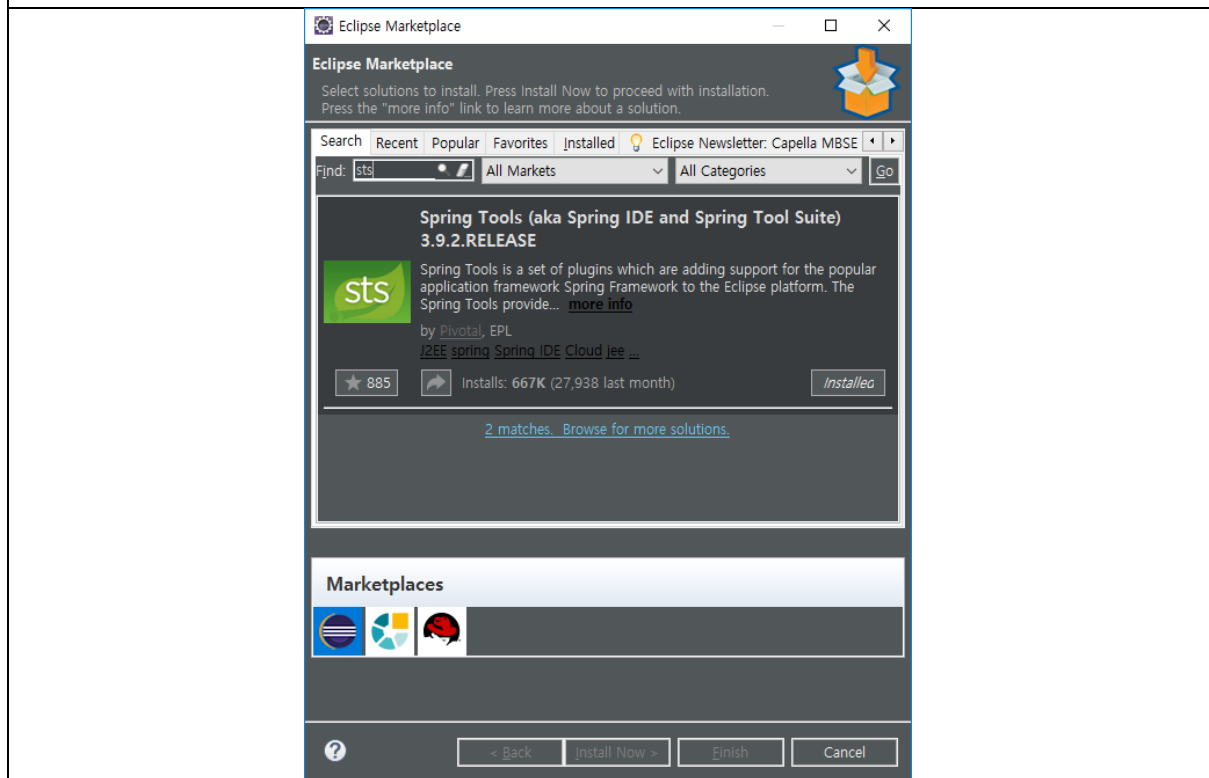
(1) Eclipse에 STS 플러그인 추가



Help -> Eclipse Marketplace



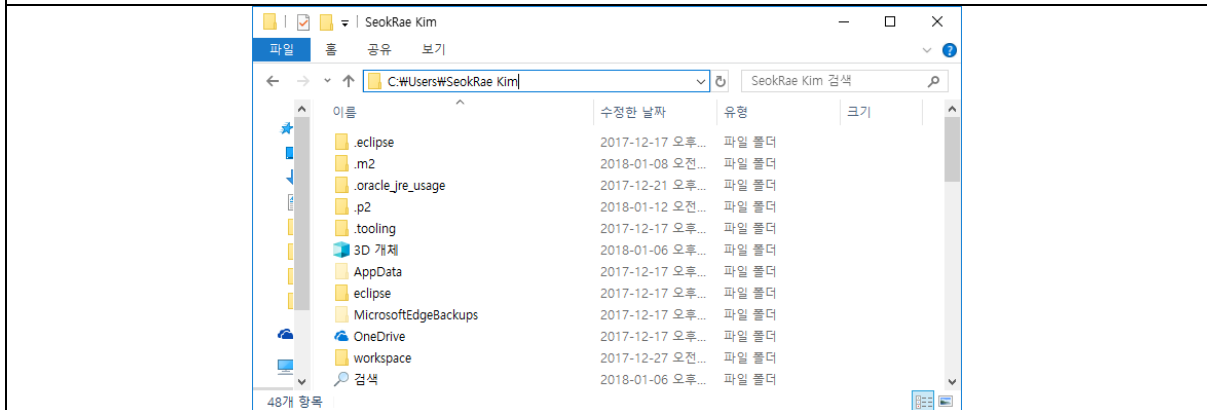
Find에 “sts”를 검색하여 Install



① Maven 빌드 시 오류 날 때

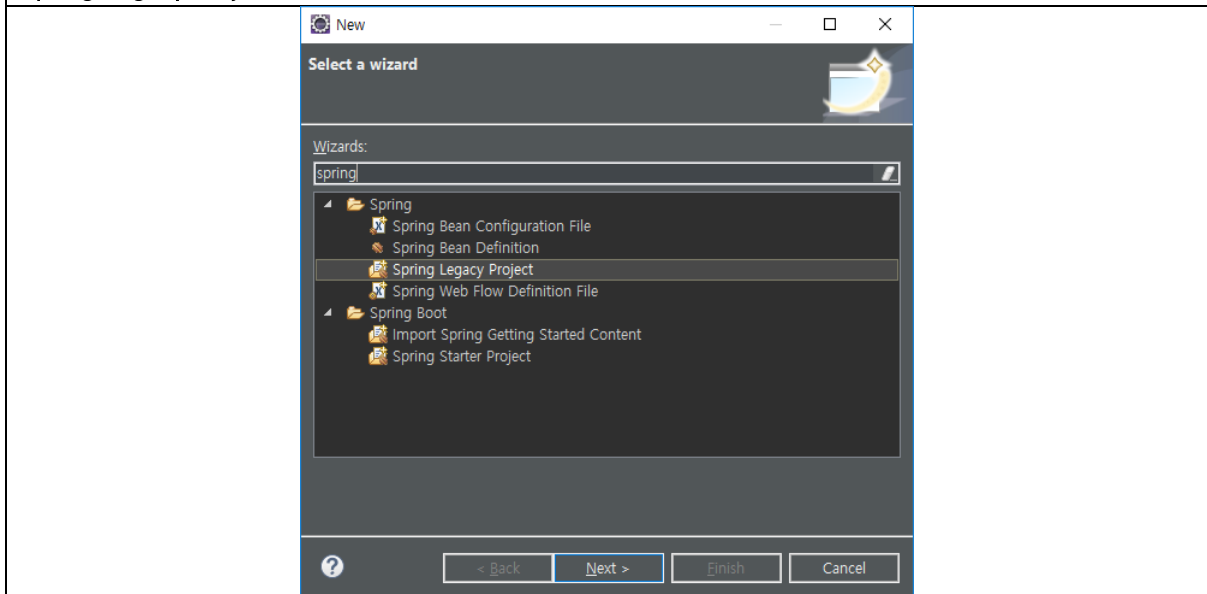
- maven이라는 빌드 도구가 필요한 jar파일들과 프로젝트를 구성하게 된다.
- 이 과정에서 네트워크의 문제나 기존의 라이브러리와 충돌로 인하여 제대로 안받아질 수도 있다.
- **.m2를 삭제 후 재시작하여** Maven 라이브러리를 다시 받도록 하는 것이 좋다.
- 또한 사용자의 이름과 폴더가 영문이 아닐 경우 제대로 설치가 안되는 경우도 있다.

.m2 폴더 위치

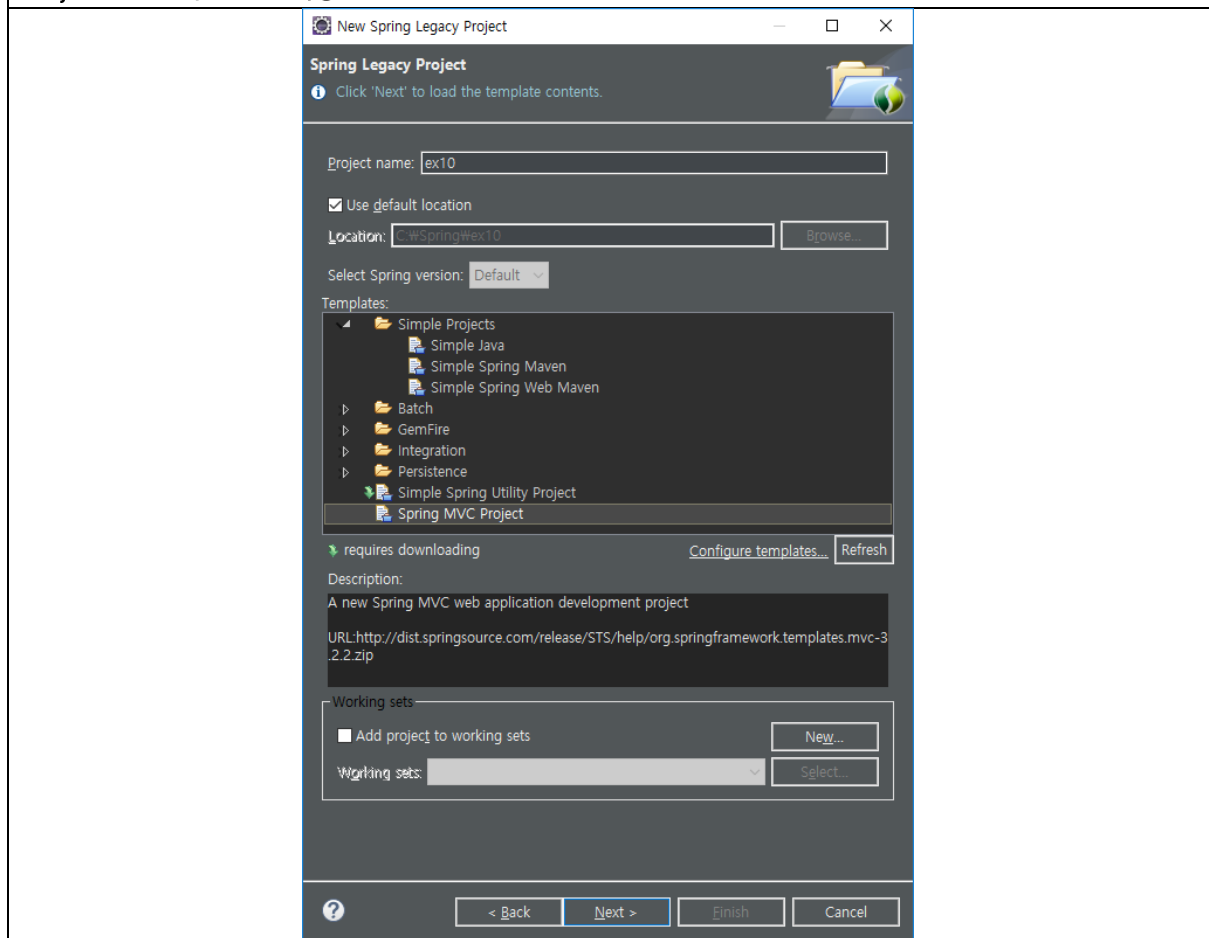


(2) Spring Project를 이용한 프로젝트 생성

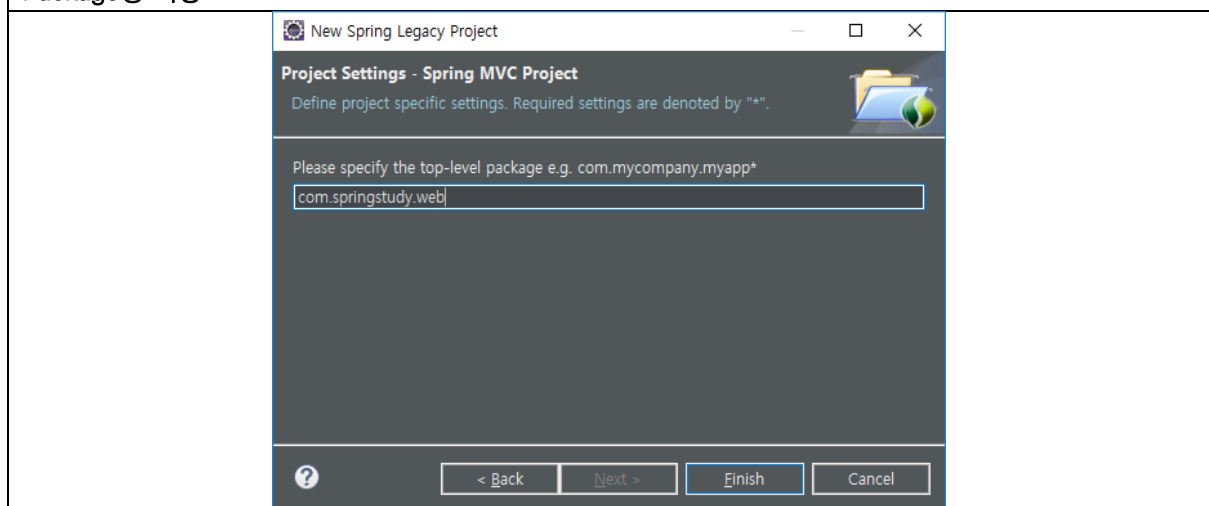
Spring Legacy Project 생성



Project name에 "ex10"작성



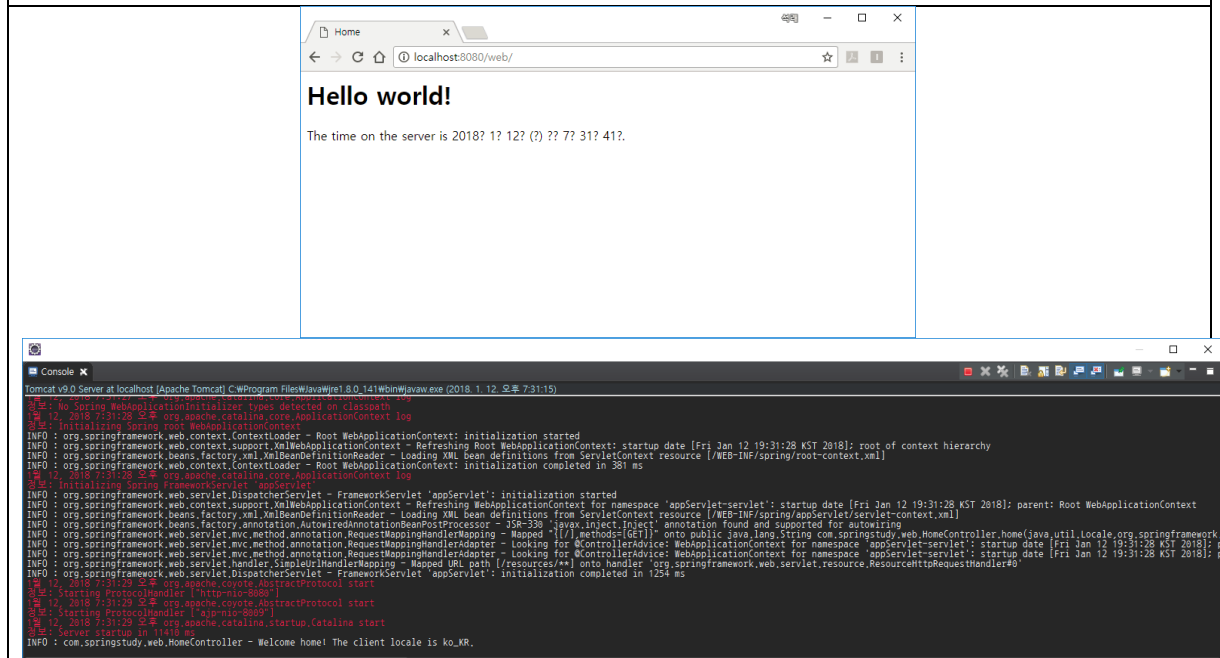
Package명 작성



(3) 라이브러리 초기화

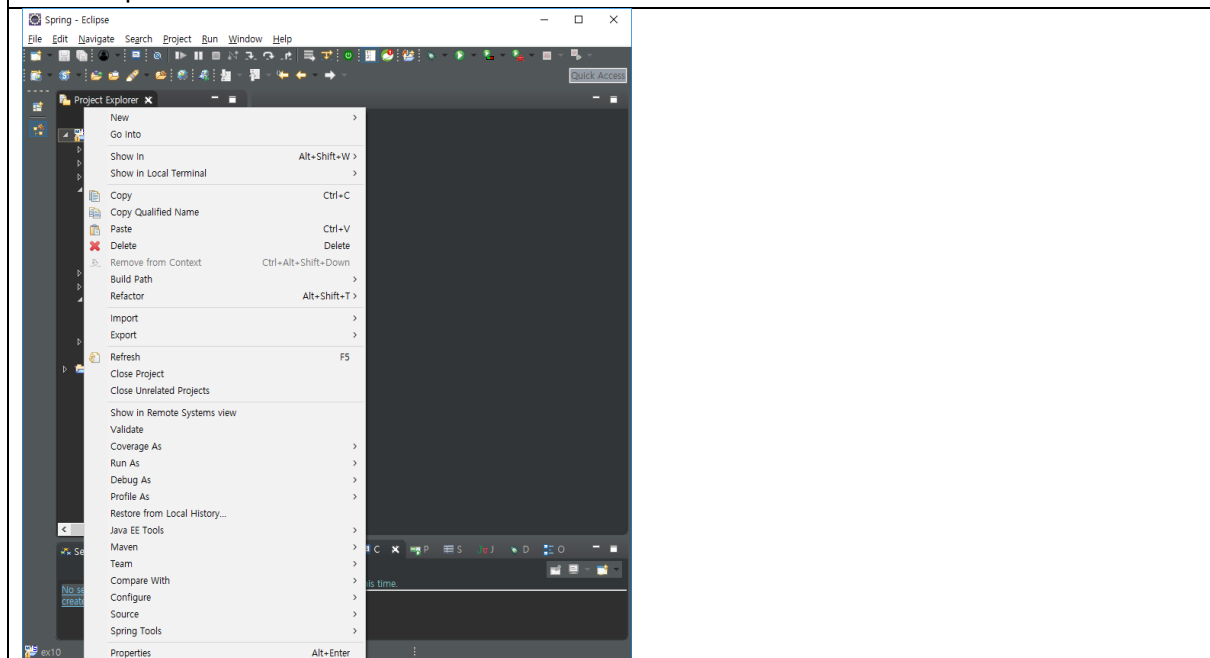
① 프로젝트 실행 테스트

프로젝트 실행 테스트

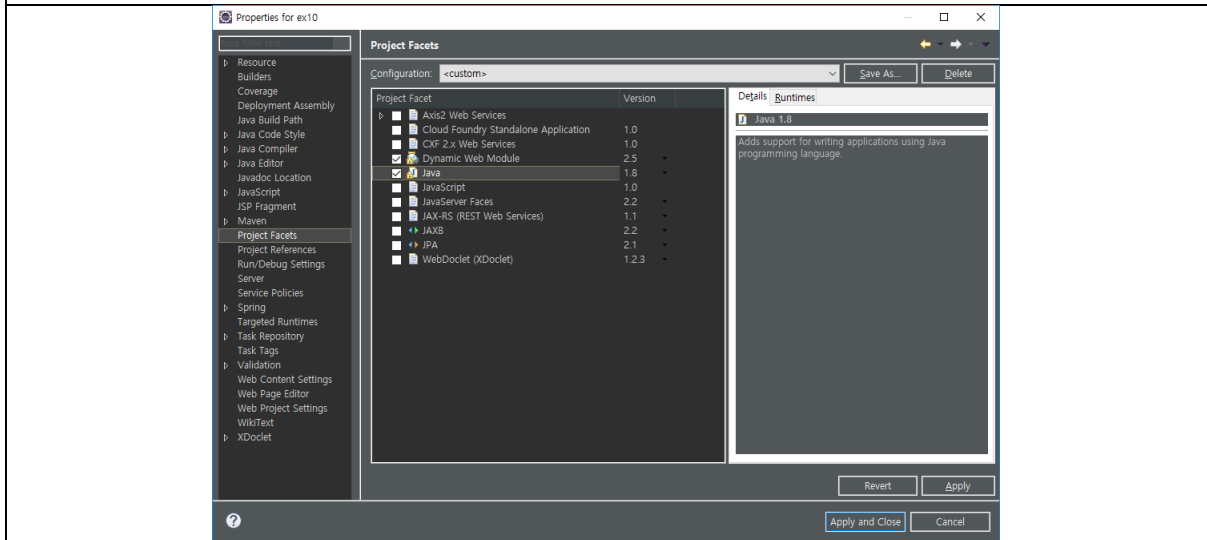


② JDK 버전의 처리

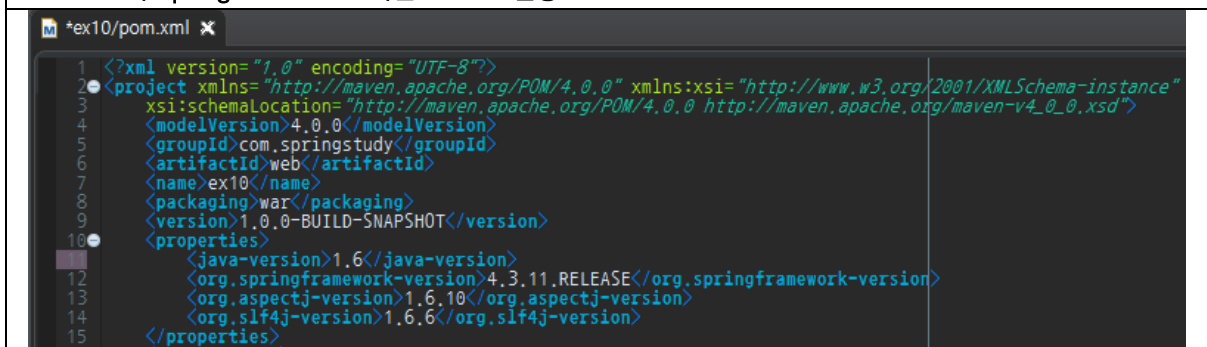
프로젝트 preferences



Project Facets -> Java 버전 1.8로 수정



Pom.xml의 springframework 버전 4.3.11 설정



<properties>

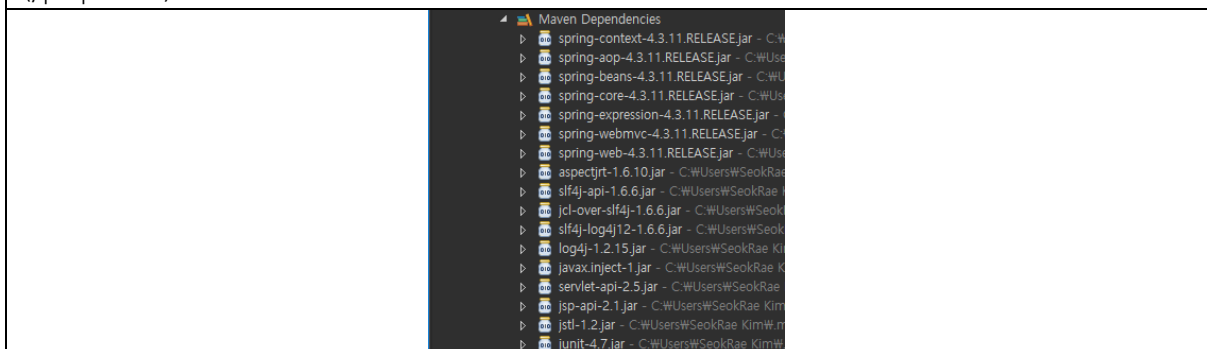
<java-version>1.6</java-version>

<org.springframework-version>4.3.11.RELEASE</org.springframework-version>

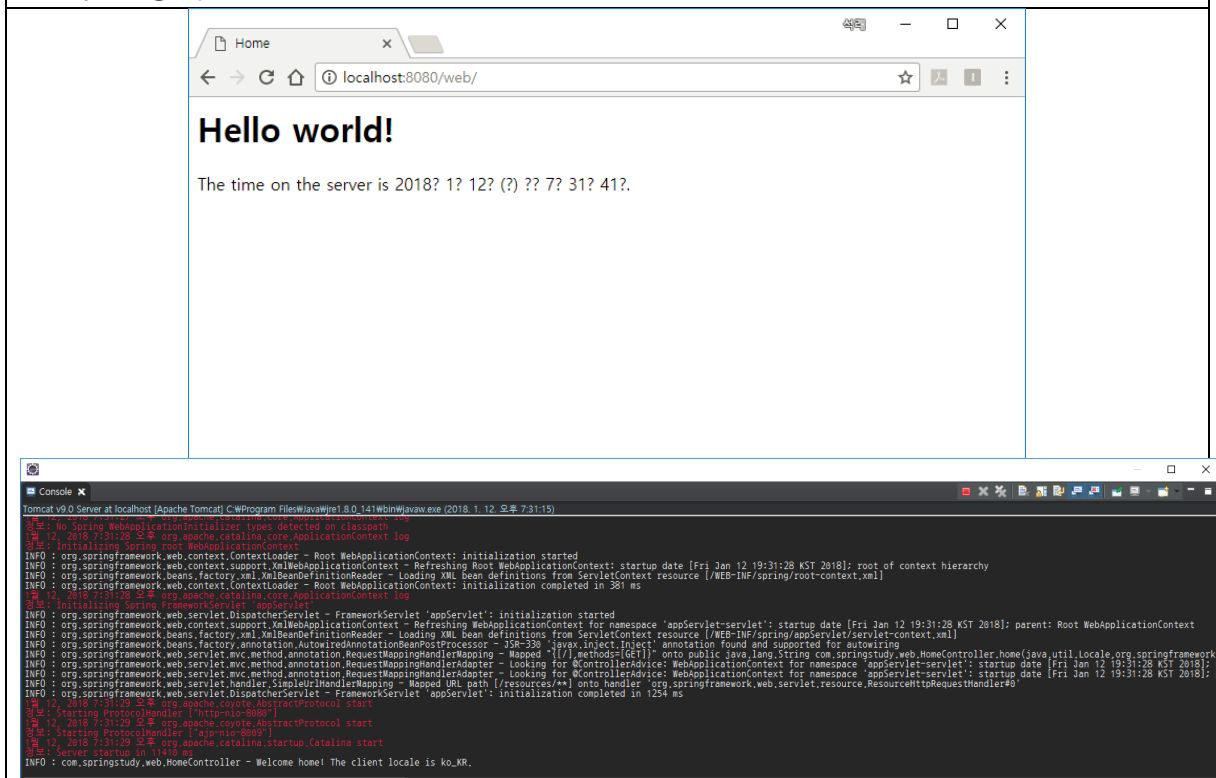
<org.aspectj-version>1.6.10</org.aspectj-version>

<org.slf4j-version>1.6.6</org.slf4j-version>

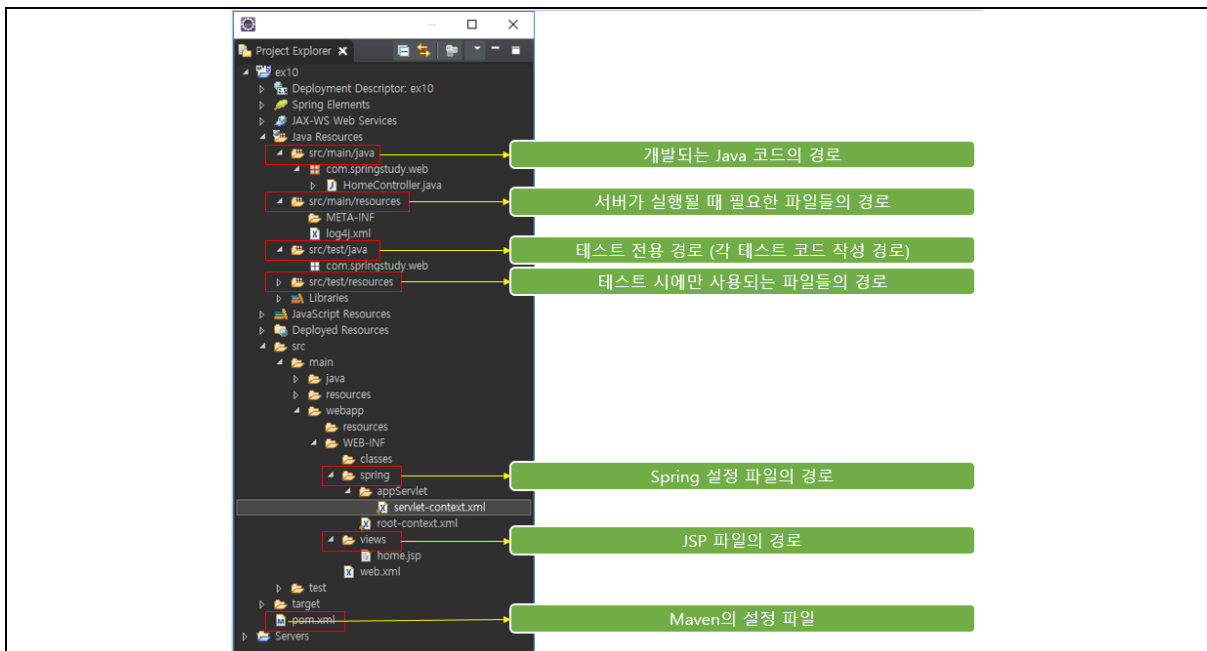
</properties>



프로젝트 실행 테스트



③ 스프링 MVC 프로젝트 템플릿 구조



④ 설치 작업 정리

- Eclipse에 STS 플러그인을 다운로드
- pom.xml을 이용해서 스프링 4.3.11버전에 맞는 라이브러리를 추가
- Tomcat 서버를 설정 이후 테스트

2) MySQL의 설정과 스프링 테스트

(1) MySQL 연결 테스트와 junit

① MySQL 테스트 관련 라이브러리

MySQL Connector/J JDBC 라이브러리 추가

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.41</version>
</dependency>
```

JDK의 버전 변경

```
<properties>
  <java-version>1.8</java-version>
  <org.springframework-version>4.3.11.RELEASE</org.springframework-version>
  <org.aspectj-version>1.6.10</org.aspectj-version>
  <org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
```

jUnit의 버전 변경

```
<!-- Test -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
</dependency>
```

② JDBC 연결 테스트 try-with 코딩

test/java/패키지/MySQLConnectionTest.java

```
public class MySQLConnection {
  private static final String DRIVER = "com.mysql.jdbc.Driver";
  private static final String URL = "jdbc:mysql://localhost:3306/{데이터베이스명}?useSSL=false";
  private static final String USER = "{DB계정명}";
  private static final String PW = "{DB비밀번호}";
```

```

@Test
public void testConnection() throws Exception {
    Class.forName(DRIVER);
    try (Connection con = DriverManager.getConnection(URL, USER, PW)) {
        System.out.println(con);
    } catch (Exception e) {
        System.out.println(e);
    }
}
}

```

Markers Properties Snippets Progress SQL Results JUnit Debug Outline Console

<terminated> MySQLConnection (1) [JUnit] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (2018. 1. 14. 오후 3:21:44)
 com.mysql.jdbc.JDBC4Connection@64b8f8f4

(2) 작업 내역 정리

- MySQL 설치
- MySQL JDBC 라이브러리 5.1.41 버전 추가
- Java 1.8 버전 수정
- junit 4.12 버전 수정
- JDBC 연결 테스트 코드

3) 스프링 + MyBatis + MySQL의 설정

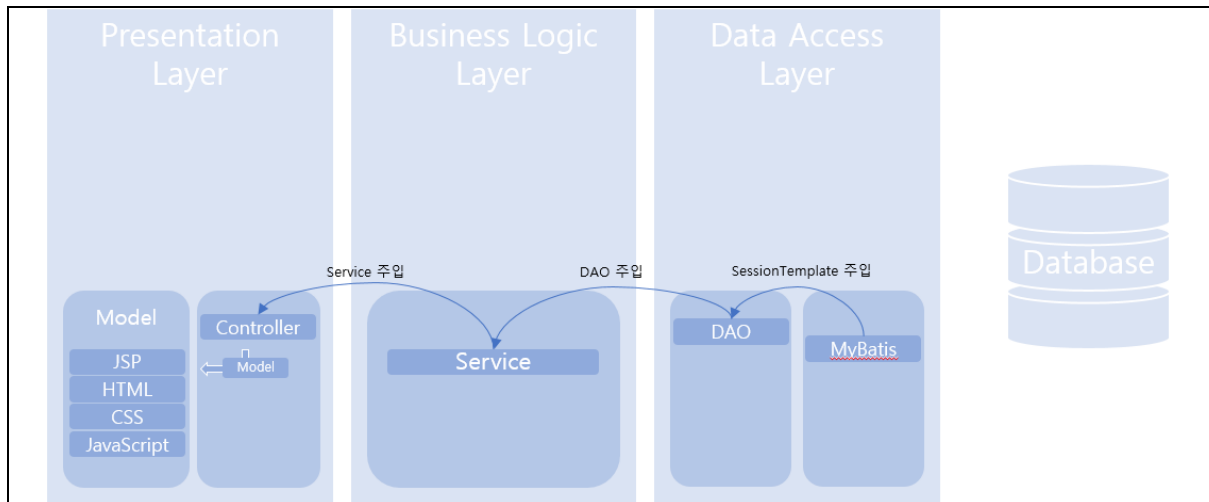
작업 목표

- 스프링과 MyBatis를 연동하기 위한 라이브러리 설정
- 데이터베이스와의 연결을 담당하는 DataSource 객체 설정
- MyBatis의 핵심인 SqlSessionFactory 객체 설정 및 테스트

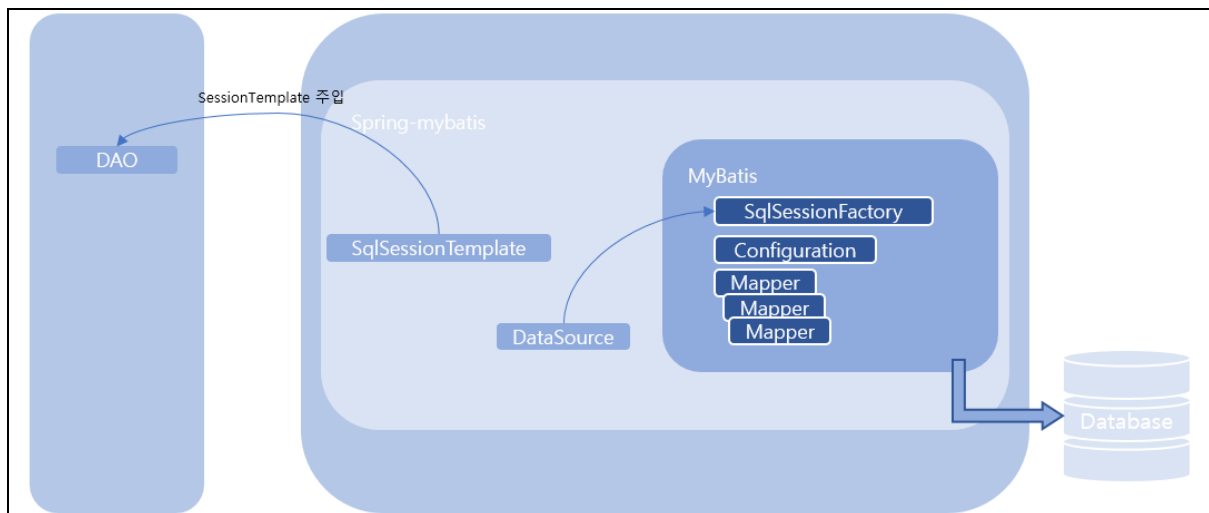
(1) 일반적인 스프링 웹 프로젝트의 구성

① 이 책의 프로젝트 구성

| | | |
|----------------------|------------|---|
| Presentation Layer | | UI를 담당하는 구성 요소 |
| | Model | 데이터를 다루는 영역 |
| | controller | 모델과 뷰 영역간의 조정 역할 |
| Business Logic Layer | | 서비스 계층으로 고객의 요구사항을 반영하는 계층 |
| | Service | 사용자 레벨 관리 기능 추가 |
| Data Access Layer | | Persistence Layer로 데이터 처리를 전문으로 담당 |
| | DAO | DB를 사용해 데이터를 조회하거나 조작하는 기능을 전담하도록 만든 오브젝트 |
| | MyBatis | 개발자가 지정한 SQL, 저장프로시저를 제공하는 퍼시스턴스 프레임워크 |



(2) MyBatis와의 구성



(3) MyBatis 연동을 위한 준비

모듈 추가

- **spring-jdbc**: JDBC의 장점과 단순성을 유지하는 간결한 api를 제공
- **spring-test**: 스프링 컴포넌트의 junit 혹은 TestNG와 관련된 유닛 테스트와 통합테스트 지원

① Spring-jdbc, spring-test, MyBatis, mybatis-spring 추가















| 이름 | pom.xml |
|---------|--|
| MyBatis | <pre> <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis --> <dependency> <groupId>org.mybatis</groupId> <artifactId>mybatis</artifactId> <version>3.4.1</version> </pre> |

| | |
|----------------|--|
| | <code></dependency></code> |
| MyBatis-Spring | <code><!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring --></code> <code><dependency></code> <code> <groupId>org.mybatis</groupId></code> <code> <artifactId>mybatis-spring</artifactId></code> <code> <version>1.3.0</version></code> <code></dependency></code> |
| spring-jdbc | <code><dependency></code> <code> <groupId>org.springframework</groupId></code> <code> <artifactId>spring-jdbc</artifactId></code> <code> <version>\${org.springframework-version}</version></code> <code></dependency></code> |
| spring-test | <code><dependency></code> <code> <groupId>org.springframework</groupId></code> <code> <artifactId>spring-test</artifactId></code> <code> <version>\${org.springframework-version}</version></code> <code></dependency></code> |

② Spring Project에서 root-context.xml 파일의 수정

- 웹 자원과 관련되지 않은 모든 자원의 설정은 root-context
- Namespaces 탭을 이용해 사용 가능한 XML 태그의 폭을 넓힘

pom.xml에 라이브러리 설정 시 root-context.xml에 Namespace 태그 추가 가능

| | |
|--|---|
| | <div> <input checked="" type="checkbox"/>  aop - http://www.springframework.org/schema/aop <input checked="" type="checkbox"/>  beans - http://www.springframework.org/schema/beans <input type="checkbox"/>  c - http://www.springframework.org/schema/c <input type="checkbox"/>  cache - http://www.springframework.org/schema/cache <input checked="" type="checkbox"/>  context - http://www.springframework.org/schema/context <input checked="" type="checkbox"/>  jdbc - http://www.springframework.org/schema/jdbc <input type="checkbox"/>  jee - http://www.springframework.org/schema/jee <input type="checkbox"/>  lang - http://www.springframework.org/schema/lang <input type="checkbox"/>  mvc - http://www.springframework.org/schema/mvc <input checked="" type="checkbox"/>  mybatis-spring - http://mybatis.org/schema/mybatis-spring <input type="checkbox"/>  p - http://www.springframework.org/schema/p <input type="checkbox"/>  task - http://www.springframework.org/schema/task <input type="checkbox"/>  tx - http://www.springframework.org/schema/tx <input type="checkbox"/>  util - http://www.springframework.org/schema/util </div> |
|--|---|

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:aop="http://www.springframework.org/schema/aop"
5       xmlns:context="http://www.springframework.org/schema/context"
6       xmlns:jdbc="http://www.springframework.org/schema/jdbc"
7       xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"
8       xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.3.xsd
9                           http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring-1.2.xsd
10                          http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
11                          http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
12                          http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd">
13
14   <!-- Root Context: defines shared resources visible to all other web components -->
15
16 </beans>

```

(4) MySQL과의 연결을 담당하는 DataSource 설정하기

- DataSource는 JDBC의 커넥션을 처리하는 기능을 갖고 있다.

spring-jdbc 모듈 클래스를 이용해서 DataSource 추가

```

<!-- spring-jdbc 모듈의 클래스를 이용해서 DataSource 추가 -->
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
    <property name="url" value="jdbc:mysql://{아이피}:3306/{데이터베이스명}?useSSL=false"/>
    <property name="username" value="{DB아이디}"/>
    <property name="password" value="{DB비밀번호}"/>
</bean>

```

① DataSource의 테스트 진행

- 인스턴스 변수의 @Inject 어노테이션 처리된 DataSource는 스프링이 생성해서 주입
- jUnit의 버전 4.11 이상인지 확인

/test/패키지/DataSourceTest.java

```

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = { "file:src/main/webapp/WEB-INF/spring/**/root-context.xml" })
public class DataSourceTest {
    @Inject
    private DataSource ds;
    @Test
    public void testConnection() {
        try(Connection con = ds.getConnection()){
            System.out.println(con);
        }catch(Exception e) {
            System.out.println(e);
        }
    }
}

```

```

Markers Properties Snippets Progress SQL Results JUnit Debug Outline Console
terminated- DataSourceTest (S) [JUnit] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (2018. 1. 14. 오후 6:05:34)
INFO : org.springframework.test.context.support.DefaultTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-INF/spring.factories]: [org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.DefaultTestContextBootstrapper, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DefaultTestContextBootstrapper]
INFO : org.springframework.test.context.support.DefaultTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DefaultTestContextBootstrapper]
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from file [C:\Spring\ex10\src\main\webapp\WEB-INF\spring\root-context.xml]
INFO : org.springframework.context.support.GenericApplicationContext - Refreshing org.springframework.context.support.GenericApplicationContext@4883b407: startup date [Sun Jan 14 18:05:34 2018]
INFO : org.springframework.jdbc.datasource.DriverManagerDataSource - Loaded JDBC driver: com.mysql.jdbc.Driver
INFO : org.springframework.jdbc.datasource.DriverManagerDataSource - Loaded JDBC driver: com.mysql.jdbc.Driver
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springframework.context.support.GenericApplicationContext@4883b407: startup date [Sun Jan 14 18:05:34 2018]

```

(5) MyBatis 연결

- DataSource가 정상적으로 설정 이후 MyBatis와 MySQL을 연동시키는 작업을 한다.

① SqlSessionFactory 객체 설정

- SqlSessionFactory는 데이터베이스와의 연결과 SQL의 실행에 대한 모든 것을 가진 중요한 객체
- SqlSessionFactoryBean을 이용하여 SqlSessionFactory를 생성

SqlSessionFactoryBean을 root-context.xml에 설정

```
<!-- MyBatis-Spring 모듈을 받아 SqlSessionFactoryBean을 추가하여 SqlSessionFactory 사용 -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource"/>
</bean>
```

② mybatis-config.xml 파일의 추가

- XML 파일의 DTD나 XML 스키마 필요
- 'mybatis config dtd' 추가

src/main/resources/mybatis-config.xml 생성

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>

</configuration>
```

root-context.xml에 mybatis-config.xml 파일 적용 설정

```
<!-- MyBatis-Spring 모듈을 받아 SqlSessionFactoryBean을 추가하여 SqlSessionFactory 사용 -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource"/>
    <property name="configLocation" value="classpath:/mybatis-config.xml"/>
</bean>
```

③ MyBatis의 연결 테스트

- root-context.xml파일에서 설정한 SqlSessionFactoryBean 설정 시 SqlSessionFactory을 주입

test/패키지/MyBatisTest.java

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = { "file:src/main/webapp/WEB-INF/spring/**/root-context.xml" })
```

```

public class MyBatisTest {

    @Inject
    private SqlSessionFactory sqlFactory;

    @Test
    public void testFactory() {
        System.out.println(sqlFactory);
    }

    @Test
    public void testSession() {
        try(SqlSession session = sqlFactory.openSession()){
            System.out.println(session);
        }catch(Exception e) {
            System.out.println(e);
        }
    }
}

```

```

<terminated> MyBatisTest (3) [JUnit] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (2018. 1. 14. 오후 7:05:47)
INFO : org.springframework.test.context.support.DefaultTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-INF/spring.factories]
INFO : org.springframework.test.context.support.DefaultTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.web.ServletTestExecutionLi
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from file [C:\Spring\ex10\src\main\webapp\WEB-INF\spring\root-context.x
INFO : org.springframework.context.support.GenericApplicationContext - Refreshing org.springframework.context.support.GenericApplicationContext@853b32d7: startup date [Sun
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annotation found and supported for autowiring
INFO : org.springframework.jdbc.datasource.DriverManagerDataSource - Loaded JDBC driver: com.mysql.jdbc.Driver
org.apache.ibatis.session.defaults.DefaultSqlSessionFactory@49e53c76
org.apache.ibatis.session.defaults.DefaultSqlSessionFactory@49e53c76
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springframework.context.support.GenericApplicationContext@853b32d7: startup date [Sun Jan

```

(6) 작업 내역의 정리

- MyBatis, MyBatis-Spring, spring-jdbc, spring-test 모듈 추가
- root-context.xml에 DataSource설정 추가
- DataSource 테스트
- MyBatis 연결을 위한 SqlSessionFactoryBean 설정, mybatis-config.xml 파일 생성
- MyBatis 연결 테스트

4) 모델 2 방식과 스프링 MVC

(1) 모델2 패턴의 이해

① 스프링 MVC의 구조에서 컨트롤러 생성

- MVC 패턴의 이해(이론)
- servlet-context.xml 구조 분석
- 컨트롤러를 중심으로 테스트(void, String, Object, redirect, JSON)

i. void 리턴 타입

SampleController – return void

@Controller


```

public class SampleController {

    private static final Logger logger = LoggerFactory.getLogger(SampleController.class);

    @RequestMapping("doA")
    public void doA() {
        logger.info("doA called.....");
    }

    @RequestMapping("doB")
    public void doB() {
        logger.info("doB called.....");
    }

}

```

```

INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization started
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationContext for namespace 'appServlet-servlet': startup date [Sun Jan 14 19:48:38 KST 2018]; paren
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML Bean definitions from ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annotation found and supported for autowiring
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[]" methods=[GET]] onto public java.lang.String com.springstudy.web.HomeController.home(jav
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[]" onto public void com.springstudy.web.SampleController.doA()
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[]" onto public void com.springstudy.web.SampleController.doB()
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter - Looking for @ControllerAdvice: WebApplicationContext for namespace 'appServlet-servlet': startup date
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter - Looking for @ControllerAdvice: WebApplicationContext for namespace 'appServlet-servlet': startup date
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/resources/**] onto handler 'org.springframework.web.servlet.resource.ResourceHttpRequestHandler#0'
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization completed in 1106 ms
[14:2018 7:48:40 오후 org.apache.coyote.AbstractProtocol start
[14:2018 7:48:40 오후 org.apache.coyote.AbstractProtocol start
[14:2018 7:48:40 오후 org.apache.catalina.startup.Catalina start
[14:2018 7:48:40 오후 org.apache.catalina.startup.Catalina start
INFO : Server startup in 10400 ms
INFO : com.springstudy.web.SampleController - doA called.....

```

ii. String 리턴 타입

SampleController2 – return String

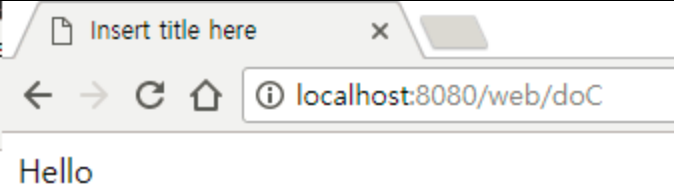
```

/**
 * @String 리턴타입은 '문자열 + .jsp'파일을 찾아서 실행한다.
 * @author SeokRae
 *
 */
@Controller
public class SampleController2 {
    private static final Logger logger = LoggerFactory.getLogger(SampleController2.class);

    @RequestMapping("doC")
    public String doC(@ModelAttribute("msg") String msg) {
        logger.info("doC called .....");
        return "result";
    }

}

```

| |
|--|
| <pre> tomcat v9.0 Server at localhost/[Apache Tomcat] C:\Program Files\Java\jre1.8.0_141\bin\java.exe (2018. 1. 15. 오후 8:08:08) INFO : org.springframework.web.context.ContextLoader - Root WebApplicationContext: initialization started INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing Root WebApplicationContext: startup date [Mon Jan 15 20:08:17 KST 2018]; root of context hierarchy INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/spring/root-context.xml] INFO : org.springframework.web.context.ContextLoader - Root WebApplicationContext: initialization completed in 713 ms [출 15, 2018 8:08:18 오후 org.apache.catalina.core.ApplicationContext log 오류: Initializing Spring FrameworkServlet 'appServlet' INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization started INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationContext for namespace 'appServlet-servlet': startup date [Mon Jan 15 20:08:18 KST 2018]; parent: Root Web INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml] INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'java.lang.Inject' annotation found and supported for autowiring INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[/].methods=[GET]" onto public java.lang.String com.spring.web.HomeController.home(java.util.Locale,c INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[/doB]" onto public void com.spring.web.SampleController.doB() INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[/doA]" onto public void com.spring.web.SampleController.doA() INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[/doC]" onto public java.lang.String com.spring.web.SampleController2.doC(java.lang.String) INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "[/doD]" onto public java.lang.String com.spring.web.SampleController3.doD(org.springframework.ui.Mode INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter - Looking for @ControllerAdvice: WebApplicationContext for namespace 'appServlet-servlet': startup date [Mon Jan INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/resources/*] onto handler org.springframework.web.servlet.resource.ResourceHttpRequestHandler#0 INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization completed in 1196 ms [출 15, 2018 8:08:19 오후 org.apache.coyote.AbstractProtocol start 오류: Starting ProtocolHandler ["http-nio-8080"] [출 15, 2018 8:08:19 오후 org.apache.coyote.AbstractProtocol start 오류: Starting ProtocolHandler ["ajp-nio-8009"] [출 15, 2018 8:08:19 오후 org.apache.catalina.startup.Catalina start 오류: Server startup in 10362 ms INFO : com.spring.web.SampleController2 - doC called </pre> |
| <p>result.jsp</p>  |

iii. Object 리턴 타입

SampleController3 – return Object

@Controller

```

public class SampleController3 {
    private static final Logger logger = LoggerFactory.getLogger(SampleController3.class);

    @RequestMapping("/doD")
    public String doD(Model model) {

        // make sample data
        ProductVO product = new ProductVO("Sample Product", 10000);
        logger.info("doD");

        // 뷰로 데이터 넘기기
        model.addAttribute(product);

        return "productDetail";
    }
}

```

```

INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/spring/root-context.xml]
INFO : org.springframework.web.context.ContextLoader - Root WebApplicationContext: initialization completed in 655 ms
[일 15, 2018 8:32:08 오후 org.apache.catalina.core.ApplicationContext log
[일 15, 2018 8:32:08 오후 org.springframework.web.servlet.FrameworkServlet - appServlet: initialization started
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization started
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationContext for namespace 'appServlet-servlet': startup date [Mon Jan 15 20:32:08 KST 2018]
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annotation found and supported for autowiring
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "(/)" methods=[GET] onto public java.lang.String com.spring.web.HomeController.home()
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "(/doA)" onto public void com.spring.web.SampleController.doA()
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "(/doB)" onto public void com.spring.web.SampleController.doB()
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "(/doC)" onto public java.lang.String com.spring.web.SampleController2.doC(java.lang.String)
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "(/doD)" onto public java.lang.String com.spring.web.SampleController3.doD(org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping)
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "(/doE)" onto public void com.spring.web.SampleController4.doE(java.lang.String)
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping - Mapped "(/doF)" onto public java.lang.String com.spring.web.SampleController4.doF(org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping)
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter - Looking for @ControllerAdvice: WebApplicationContext for namespace 'appServlet-servlet': started
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/resources/*] onto handler 'org.springframework.web.servlet.resource.ResourceHttpRequestHandler'
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization completed in 1269 ms
[일 15, 2018 8:32:10 오후 org.apache.coyote.AbstractProtocol start
[일 15, 2018 8:32:10 오후 org.apache.coyote.AbstractProtocol start
[일 15, 2018 8:32:10 오후 org.apache.coyote.AbstractProtocol start
[일 15, 2018 8:32:10 오후 org.apache.catalina.startup.Catalina start
[일 15, 2018 8:32:10 오후 org.apache.catalina.startup.Catalina start
INFO : com.spring.web.SampleController3 - doD

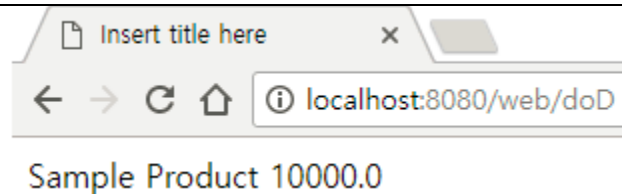
```

ProductDetail.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Insert title here</title>
    </head>
    <body>
        <span>${productVO.name}</span>
        <span>${productVO.price}</span>
    </body>
</html>

```



iv. 리다이렉트의 경우

SampleController4 – redirect

@Controller

```
public class SampleController4 {
```

```
    private static final Logger logger = LoggerFactory.getLogger(SampleController4.class);
```

```
    @RequestMapping("/doE")
```

```
    public String doE(RedirectAttributes rtr) {
```

```
        logger.info("doE called but redirect to /doF.....");
```

```
        rtr.addFlashAttribute("msg", "This is the Message!! with redirected");
```

```
        return "redirect:/doF";
```

```
    }
```

```
    @RequestMapping("/doF")
```

```
    public void doF(@ModelAttribute String msg) {
```

```

        logger.info("doF called ..... " + msg);
    }
}

```

```

Tomcat v9.0. Server at localhost [Apache Tomcat/9.0.14]
15. 2018 8:35:47 오후 org.apache.catalina.core.ApplicationContext log
INFO: Initializing Spring FrameworkServlet 'appServlet'
INFO: org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization started
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Refreshing WebApplicationContext for namespace 'appServlet-servlet': startup date [Mon Jan 15 20:35:47 KST 2018]
INFO: org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]
INFO: org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annotation found and supported for autowiring
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Mapping methods=GET] onto public java.lang.String com.spring.web.HomeController.home()
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Mapping methods=doA] onto public void com.spring.web.SampleController.doA()
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Mapping methods=doB] onto public void com.spring.web.SampleController.doB()
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Mapping methods=doC] onto public java.lang.String com.spring.web.SampleController2.doC(java.lang.String)
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Mapping methods=doD] onto public java.lang.String com.spring.web.SampleController3.doD(org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter)
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Mapping methods=doE] onto public void com.spring.web.SampleController4.doE(java.lang.String)
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Mapping methods=doF] onto public java.lang.String com.spring.web.SampleController4.doF(org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter)
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Looking for @ControllerAdvice: WebApplicationContext for namespace 'appServlet-servlet': sta
INFO: org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter - Looking for @ControllerAdvice: WebApplicationContext for namespace 'appServlet-servlet': sta
INFO: org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/resources/**] onto handler 'org.springframework.web.servlet.resource.ResourceHttpRequestHandler'
INFO: org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization completed in 1188 ms
15. 2018 8:35:48 오후 org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
15. 2018 8:35:48 오후 org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8089"]
15. 2018 8:35:48 오후 org.apache.catalina.startup.Catalina start
INFO: Server startup in 10395 ms
INFO: com.spring.web.SampleController4 - doF called but redirect to /doF.....
INFO: com.spring.web.SampleController4 - doF called .....

```

v. JSON 리턴 타입

- JSON데이터에 대한 처리를 간단하게 할 수 있는 Jackson-databind 라이브러리를 추가

pom.xml

```

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.8.4</version>
</dependency>

```

SampleController5 – JSON

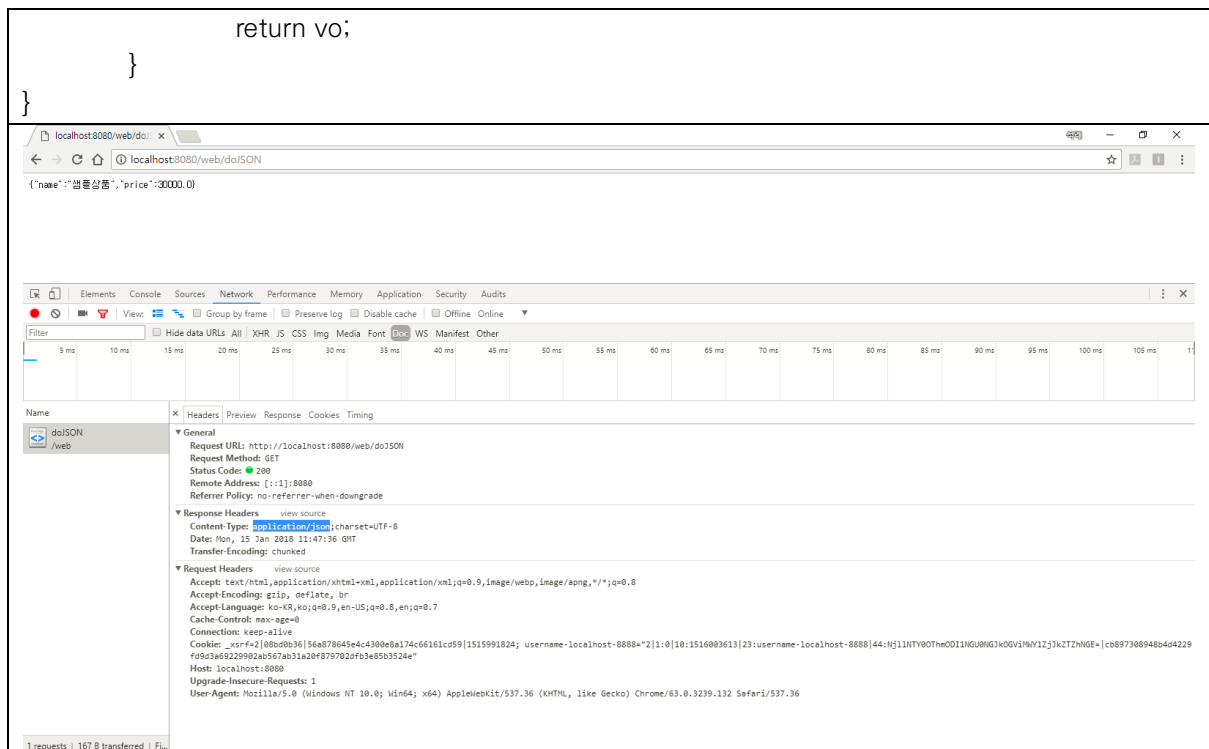
```

@Controller
public class SampleController5 {

    @RequestMapping("/doJSON")
    public @ResponseBody ProductVO doJSON() {

        ProductVO vo = new ProductVO("샘플상품", 30000);
    }
}

```



② WAS없이 컨트롤러 테스트

pom.xml의 servlet 버전 수정

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
</dependency>
```

test/패키지/SampleControllerTest

```
@RunWith(SpringJUnit4ClassRunner.class)
@WebAppConfiguration
@ContextConfiguration(locations = { "file:src/main/webapp/WEB-INF/spring/**/*.xml" })
public class SampleControllerTest {

    private static final Logger logger = LoggerFactory.getLogger(SampleControllerTest.class);

    @Inject
    private WebApplicationContext wac;

    private MockMvc mockMvc;

    @Before
    public void setup() {
        this.mockMvc = MockMvcBuilders.webAppContextSetup(this.wac).build();
    }
}
```

```

        logger.info("setup .....");
    }

    @Test
    public void testDoA() throws Exception {
        mockMvc.perform(MockMvcRequestBuilders.get("/doA"));
    }
}

```

```

INFO : org.springframework.test.context.web.WebTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-INF/spring.factories]: [org.springframework.test.context.web.WebTestContextBootstrapper, org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DirtContextTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DefaultTestExecutionListener, org.springframework.test.context.support.MethodOutputTestExecutionListener]
INFO : org.springframework.test.context.web.WebTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DirtContextTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DefaultTestExecutionListener, org.springframework.test.context.support.MethodOutputTestExecutionListener]
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from file [C:\SpringPractice\ex00\src\main\webapp\WEB-INF\spring\appServlet\servlet-context.xml]
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from file [C:\SpringPractice\ex00\src\main\webapp\WEB-INF\spring\root-context.xml]
INFO : org.springframework.context.support.GenericWebApplicationContext - Refreshing org.springframework.context.support.GenericWebApplicationContext@598867a5: startup date [Mon Jan 15 21:01:56 KST]
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annotation found and supported for autowiring
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/],methods=[GET]}" onto public java.lang.String com.spring.web.HomeController.home(java.util.Locale, java.lang.String)
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/doB]}" onto public void com.spring.web.SampleController.doB()
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/doA]}" onto public void com.spring.web.SampleController.doA()
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/doC]}" onto public java.lang.String com.spring.web.SampleController2.doC(java.lang.String)
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/doB]}" onto public java.lang.String com.spring.web.SampleController3.doB(org.springframework.ui.Model)
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/doE]}" onto public java.lang.String com.spring.web.SampleController4.doE(org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping)
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/doF]}" onto public void com.spring.web.SampleController4.doF(java.lang.String)
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Mapped "{[/doJSON]}" onto public com.spring.domain.ProductVO com.spring.web.SampleController5.doJSON()
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Looking for @ControllerAdvice: org.springframework.context.support.GenericWebApplicationContext@598867a5
INFO : org.springframework.web.servlet.mvc.annotation.AnnotationMethodMapping - Looking for @ControllerAdvice: org.springframework.context.support.GenericWebApplicationContext@598867a5
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/resources/**] onto handler 'org.springframework.web.servlet.resource.ResourceHttpRequestHandler#0'
INFO : org.springframework.jdbc.datasource.DriverManagerDataSource - Loaded JDBC driver: com.mysql.jdbc.Driver
INFO : org.springframework.mock.web.MockServletContext - Initializing Spring FrameworkServlet
INFO : org.springframework.test.web.servlet.TestDispatcherServlet - FrameworkServlet '':: initialization started
INFO : org.springframework.test.web.servlet.TestDispatcherServlet - FrameworkServlet '':: initialization completed in 31 ms
INFO : com.spring.web.SampleControllerTest - setup .....
INFO : com.spring.web.SampleController - doA called .....
INFO : org.springframework.context.support.GenericWebApplicationContext - Closing org.springframework.context.support.GenericWebApplicationContext@598867a5: startup date [Mon Jan 15 21:01:56 KST]

```

(2) 정리

- 타입별 컨트롤러 생성
- WAS없이 컨트롤러 테스트

5) 스프링 + MyBatis

- 테이블 생성, VO 클래스 작성
- DAO 인터페이스 작성 및 구현(@Repository, @Inject SqlSession)
- XML Mapper 작성 및 인식을 위한 설정(root-context.xml)

(1) 테이블 생성 및 개발 준비

① 테이블 생성

Member Table 생성

```

CREATE TABLE tbl_member (
    userid varchar(50) not null
    , userpw varchar(50) not null
    , username varchar(50) not null
    , email varchar(100)
    , regdate timestamp default now()
    , updatedate timestamp default now()
    , primary key(userid)
);

```

② 도메인 객체 생성

테이블의 속성과 같은 멤버 변수를 가진 도메인(VO) 클래스 생성

```
public class MemberVO {  
    private String userid;  
    private String userpw;  
    private String username;  
    private String email;  
    private Date regdate;  
    private Date updatedate;  
}
```

(2) DAO 인터페이스 작성

- 현재 시간을 체크하는 기능
- 데이터 추가 기능

MemberDAO 작성

```
public interface MemberDAO {  
  
    public String getTime();  
  
    public void insertMember(MemberVO vo);  
}
```

(3) XML Mapper 작성

- XML로 작성된 Mapper의 위치 src/main/resources/mappers에 위치
- XML Mapper 파일 작성, DTD 추가
- SQL 작성

mappers/memberMapper.xml 작성

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE mapper  
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
  
<mapper namespace="org.zerock.mapper.MemberMapper">  
    <select id="getTime" resultType="string">  
        select now()  
    </select>  
  
    <insert id="insertMember">  
        INSERT INTO tbl_member (  
            userid
```

```

        , userpw
        , username
        , email
    )
    VALUES (
        , #{userid}
        , #{userpw}
        , #{username}
        , #{email}
    )
</insert>
</mapper>

```

① mybatis-spring에 XML Mapper 설정

root-context.xml 설정 변경

```

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource"></property>
    <property name="configLocation" value="classpath:/mybatis-config.xml"></property>
    <property name="mapperLocations" value="classpath:/mappers/**/*.Mapper.xml"></property>
</bean>

```

(4) DAO 인터페이스의 구현

- DAO인터페이스와 Mapper를 이용한 구현클래스 작성
- MyBatis에서 DAO를 이용하는 경우에는 SqlSessionTemplate를 이용한다.

① SqlSessionTemplate 설정

- 기본적인 트랜잭션의 관리
- 스레드 처리의 안전성을 보장
- 데이터베이스의 연결과 종료를 책임진다.
- SqlSessionTemplate는 SqlSessionFactory를 생성자로 주입해서 설정

root-context.xml에 SqlSessionTemplate 설정

```

<bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate" destroy-method="clearCache">
    <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"></constructor-arg>
</bean>

```

② 구현 클래스 작성

- SqlSessionTemplate을 주입하여 사용
- DAO를 스프링에 인식시키기 위한 어노테이션 @Repository

MemberDAOImpl

```
@Repository
public class MemberDAOImpl implements MemberDAO{

    @Inject
    private SqlSession sqlSession;
    /**
     * sqlSession의 기능
     * selectOne
     * selectList
     * selectMap
     * insert
     * update
     * delete
     *
     */
    private static final String namespace = "com.spring.web.MemberMapper";

    @Override
    public String getTime() {
        return sqlSession.selectOne(namespace + ".getTime");
    }

    @Override
    public void insertMember(MemberVO vo) {
        sqlSession.insert(namespace + ".insertMember", vo);
    }
}
```

(5) 스프링에 빈으로 등록

root-context.xml에 컴포넌트 스캔 처리

```
<!-- 스프링에서 해당 패키지를 스캔, @Repository 어노테이션을 읽음 -->
<context:component-scan base-package="com.spring.persistence"></context:component-scan>
```

(6) 테스트 코드 작성

MemberDAOTest

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"file:src/main/webapp/WEB-INF/spring/**/root-context.xml"})
public class MemberDAOTest {

    @Inject
    private MemberDAO dao;
```

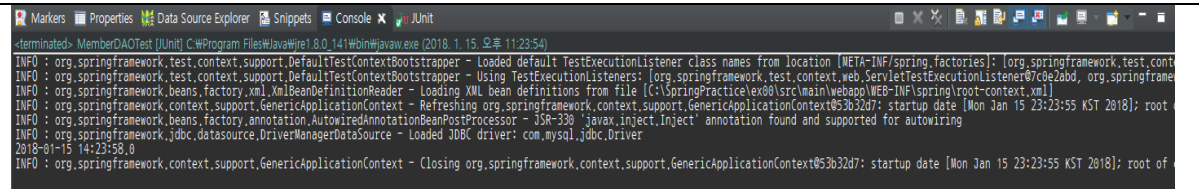
```

@Test
public void testTime() {
    System.out.println(dao.getTime());
}

@Test
public void testInsertMember() {
    MemberVO vo = new MemberVO();
    vo.setUserid("user05");
    vo.setUserpw("user05");
    vo.setUsername("USER05");
    vo.setEmail("user05@aaa.com");

    dao.insertMember(vo);
}
}

```



(7) MyBatis의 로그 log4jdbc-log4j2설정

- 잘못된 SQL이나 잘못된 속성의 이름으로 예외발생을 위한 로그 설정 라이브러리

MyBatis 로그 설정을 위한 log4jdbc-log4j2 라이브러리 추가

```

<!-- https://mvnrepository.com/artifact/org.bgee.log4jdbc-log4j2/log4jdbc-log4j2-jdbc4 -->
<dependency>
    <groupId>org.bgee.log4jdbc-log4j2</groupId>
    <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
    <version>1.16</version>
</dependency>

```

root-context.xml에 데이터베이스와 연결되는 드라이버 클래스와 연결 URL 수정

```

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="net.sf.log4jdbc.sql.jdbcapi.DriverSpy"></property>
    <property name="url" value="jdbc:log4jdbc:mysql://mysqlinstance.c4otsabdm6jt.ap-northeast-2.rds.amazonaws.com:3306/snsdb?useSSL=false"></property>
    <property name="username" value="sdba"></property>
    <property name="password" value="1234qwer"></property>
</bean>

```

- log4jdbc-log4j2가 올바르게 동작하게 하기 위한 로그 관련 설정 파일 추가

log4jdbc.log4j2.properties 파일

```
log4jdbc.spylogdelegator.name=net.sf.log4jdbc.log.slf4j.Slf4jSpyLogDelegator
```

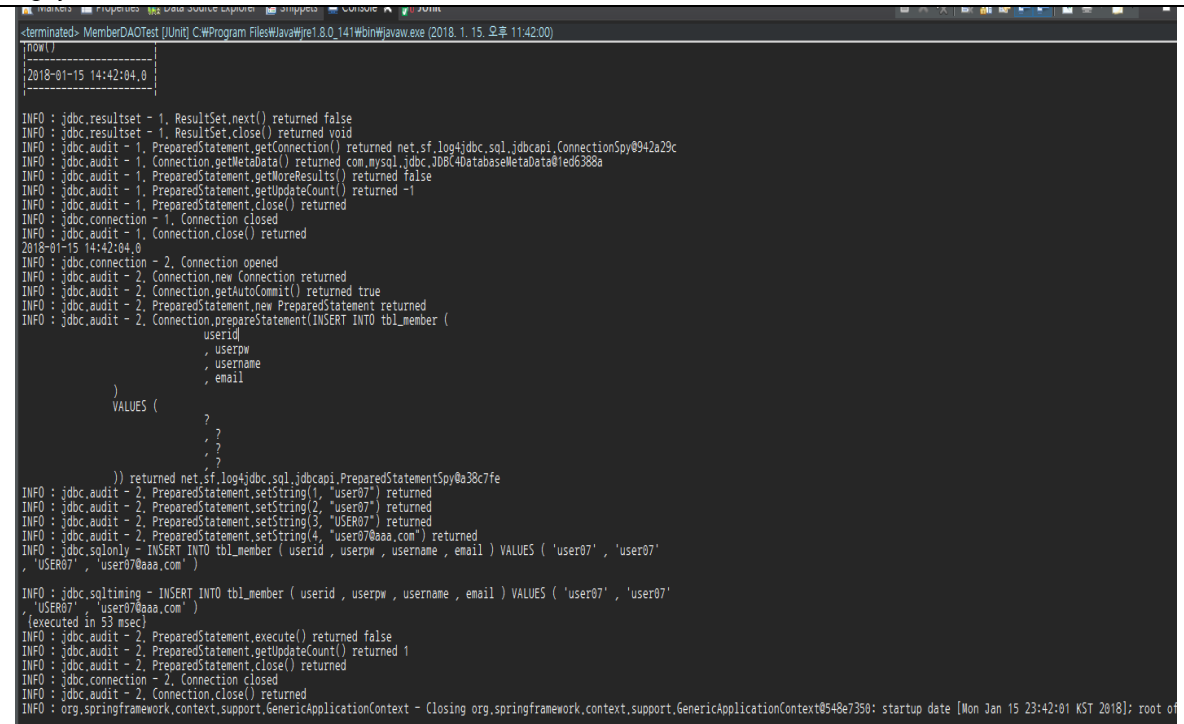
logback.xml 파일

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <include resource="org/springframework/boot/logging/logback/base.xml"/>

    <!-- log4jdbc-log4j2 -->
    <logger name="jdbc.sqlonly" level="DEBUG"/>
    <logger name="jdbc.sqltiming" level="INFO"/>
    <logger name="jdbc.audit" level="WARN"/>
    <logger name="jdbc.resultset" level="ERROR"/>
    <logger name="jdbc.resultsettable" level="ERROR"/>
    <logger name="jdbc.connection" level="INFO"/>

</configuration>
```

log4jdbc 설정 후 결과



```
<terminated> MemberDAOTest [JUnit] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (2018. 1. 15. 오후 11:42:00)
[INFO]
2018-01-15 14:42:04.0
INFO : jdbc.resultset - 1. ResultSet.next() returned false
INFO : jdbc.resultset - 1. ResultSet.close() returned void
INFO : jdbc.audit - 1. PreparedStatement.getConnection() returned net.sf.log4jdbc.sql.jdbcapi.ConnectionSpy@942a29c
INFO : jdbc.audit - 1. Connection.getMetaData() returned com.mysql.jdbc.JDBC4DatabaseMetaData@1ed6388a
INFO : jdbc.audit - 1. PreparedStatement.getMoreResults() returned false
INFO : jdbc.audit - 1. PreparedStatement.getUpdateCount() returned -1
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.connection - 1. Connection closed
INFO : jdbc.audit - 1. Connection.close() returned
2018-01-15 14:42:04.0
INFO : jdbc.connection - 2. Connection opened
INFO : jdbc.audit - 2. Connection.new Connection returned
INFO : jdbc.audit - 2. Connection.setAutoCommit() returned true
INFO : jdbc.audit - 2. PreparedStatement.new PreparedStatement returned
INFO : jdbc.audit - 2. Connection.prepareStatement(INSERT INTO tbl_member (
    userid
    , userpw
    , username
    , email
)
VALUES (
    ?
    , ?
    , ?
    , ?
)) returned net.sf.log4jdbc.sql.jdbcapi.PreparedStatementSpy@a38c7fe
INFO : jdbc.audit - 2. PreparedStatement.setString(1, "user07") returned
INFO : jdbc.audit - 2. PreparedStatement.setString(2, "user07") returned
INFO : jdbc.audit - 2. PreparedStatement.setString(3, "USER07") returned
INFO : jdbc.audit - 2. PreparedStatement.setString(4, "user07@aaa.com") returned
INFO : jdbc.sqlonly - INSERT INTO tbl_member ( userid , userpw , username , email ) VALUES ( 'user07' , 'user07'
, 'USER07' , 'user07@aaa.com' )
INFO : jdbc.sqltiming - INSERT INTO tbl_member ( userid , userpw , username , email ) VALUES ( 'user07' , 'user07'
, 'USER07' , 'user07@aaa.com' )
(executed in 53 msec)
INFO : jdbc.audit - 2. PreparedStatement.execute() returned false
INFO : jdbc.audit - 2. PreparedStatement.getUpdateCount() returned 1
INFO : jdbc.audit - 2. PreparedStatement.close() returned
INFO : jdbc.connection - 2. Connection closed
INFO : jdbc.audit - 2. Connection.close() returned
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springframework.context.support.GenericApplicationContext@548e7350: startup date [Mon Jan 15 23:42:01 KST 2018]; root of
```

(8) MyBatis의 #{} 문법

- 사용자의 등록과 조회 처리

MemberDAO 인터페이스

```
public MemberVO readMember(String userid) throws Exception;
```

```
public MemberVO readWithePW(String userid, String userpw) throws Exception;
```

MemberDAOImpl

@Repository

public class MemberDAOImpl implements MemberDAO{

 @Inject

 private SqlSession sqlSession;

 /**

 * sqlSession의 기능

 * selectOne

 * selectList

 * selectMap

 * insert

 * update

 * delete

 *

 */

 private static final String namespace = "com.spring.web.MemberMapper";

 @Override

 public String getTime() {

 return sqlSession.selectOne(namespace + ".getTime");

 }

 @Override

 public void insertMember(MemberVO vo) {

 sqlSession.insert(namespace + ".insertMember", vo);

 }

 @Override

 public MemberVO readMember(String userid) throws Exception {

 return (MemberVO) sqlSession.selectOne(namespace + ".selectMember", userid);

 }

 @Override

 public MemberVO readWithePW(String userid, String userpw) throws Exception {

 Map<String, Object> paramMap = new HashMap<String, Object>();

 paramMap.put("userid", userid);

 paramMap.put("userpw", userpw);

 return sqlSession.selectOne(namespace + ".readWithPW", paramMap);

 }

}

- 사용자 조회 쿼리문 등록

memberMapper.xml

```
<select id="selectMember" resultType="com.spring.domain.MemberVO">
    SELECT
        *
    FROM tbl_member
    WHERE userid = #{userid}
</select>
<select id="readWithPW" resultType="com.spring.domain.MemberVO">
    SELECT
        *
    FROM tbl_member
    WHERE userid = #{userid}
    AND    userpw = #{userpw}
</select>
```

6) 정리

- ① 개발환경 설정
- ② MySQL 설정
- ③ MyBatis 설정
- ④ Spring + MyBatis 설정