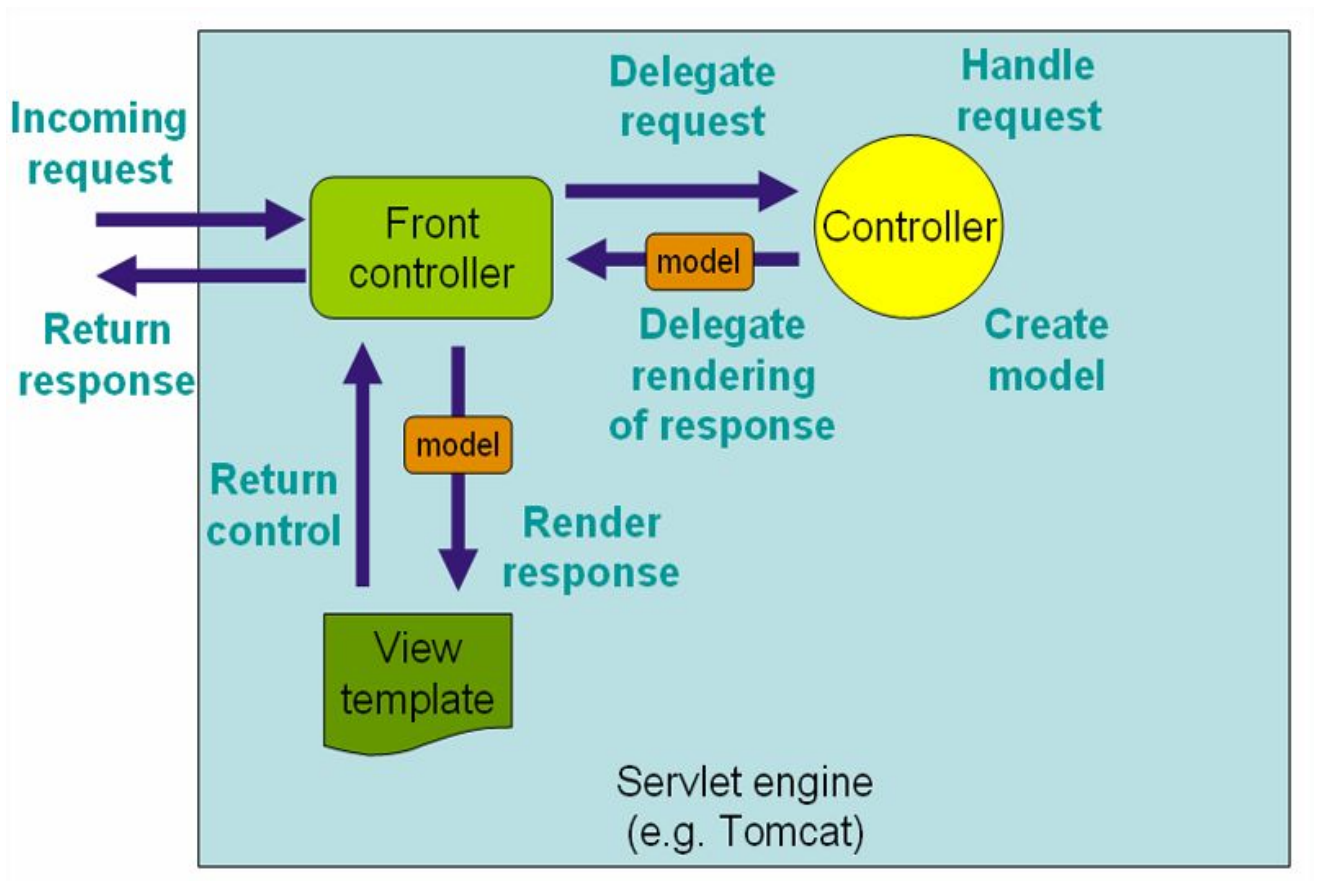
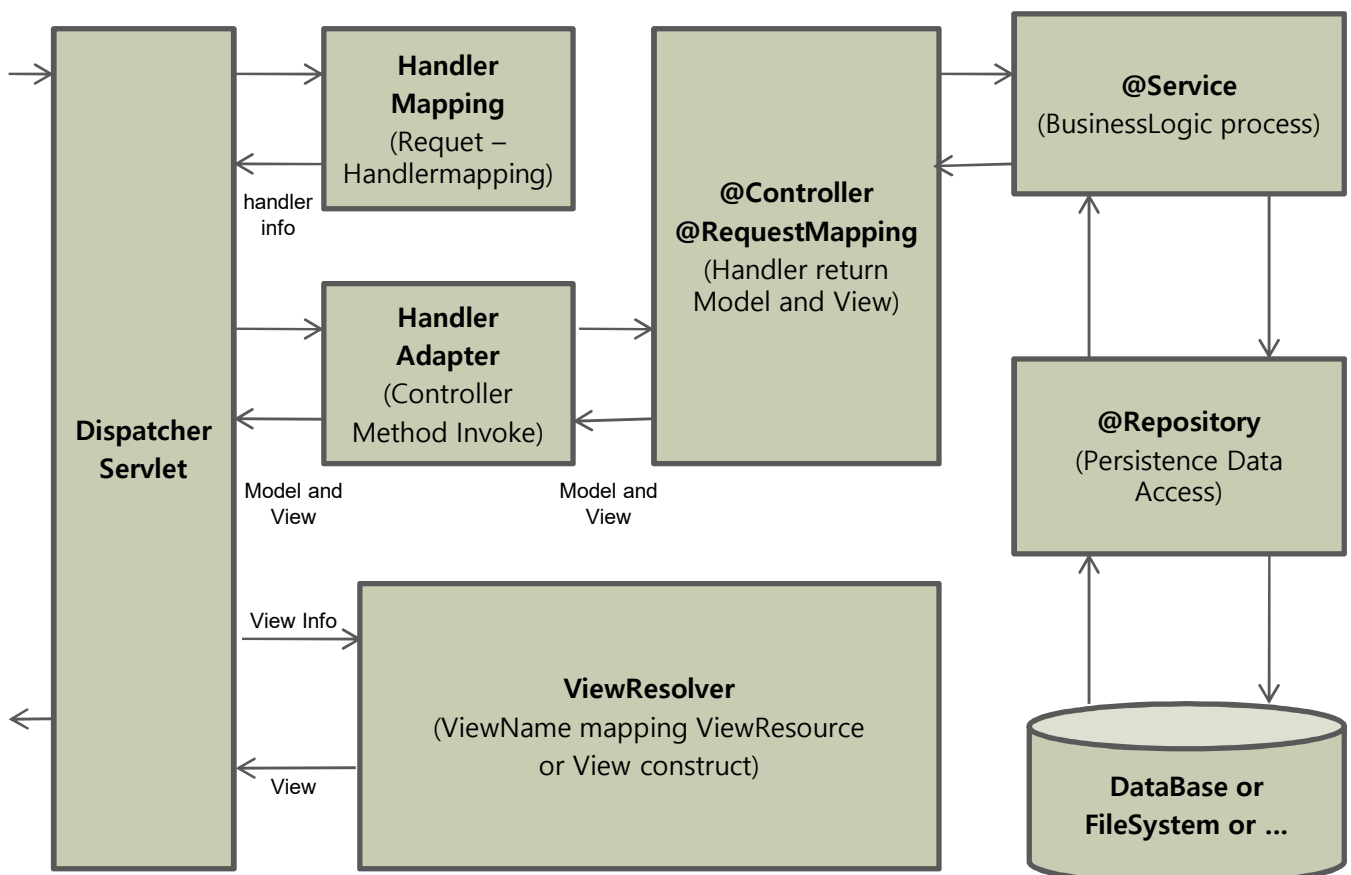


Spring Framework

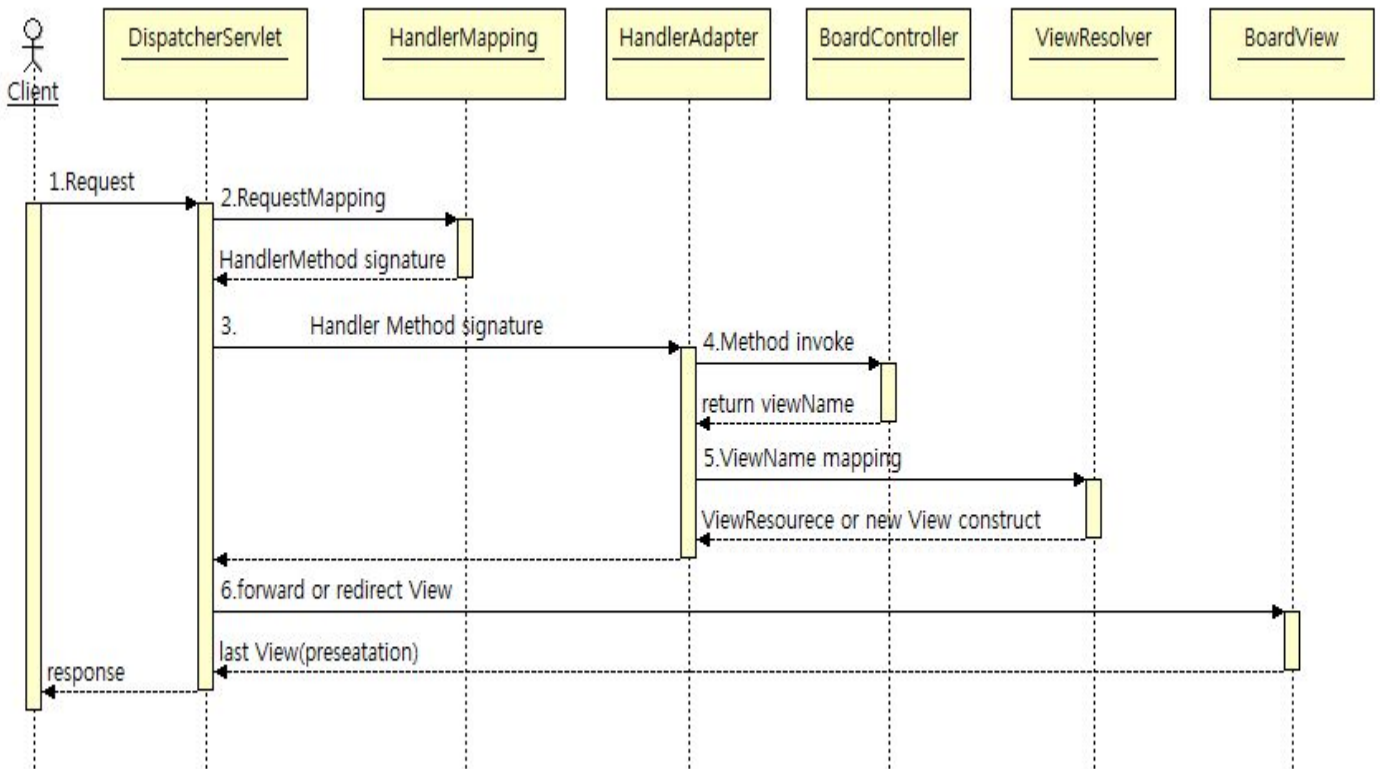
Spring XML Config& MyBatis & Tiles & Maven 게시판 Tutorial

□ 본 교재는 스프링 과 메이븐, 타일즈 및 마이바티스 교육을 목적으로 작성된 실습용 교재로, 구현 코드는 스프링 4.X, 메이븐 3.X, 타일즈 3.X, 마이바티스 3.x 버전에 맞춰 작성되었으니, 참고하시기 바랍니다.

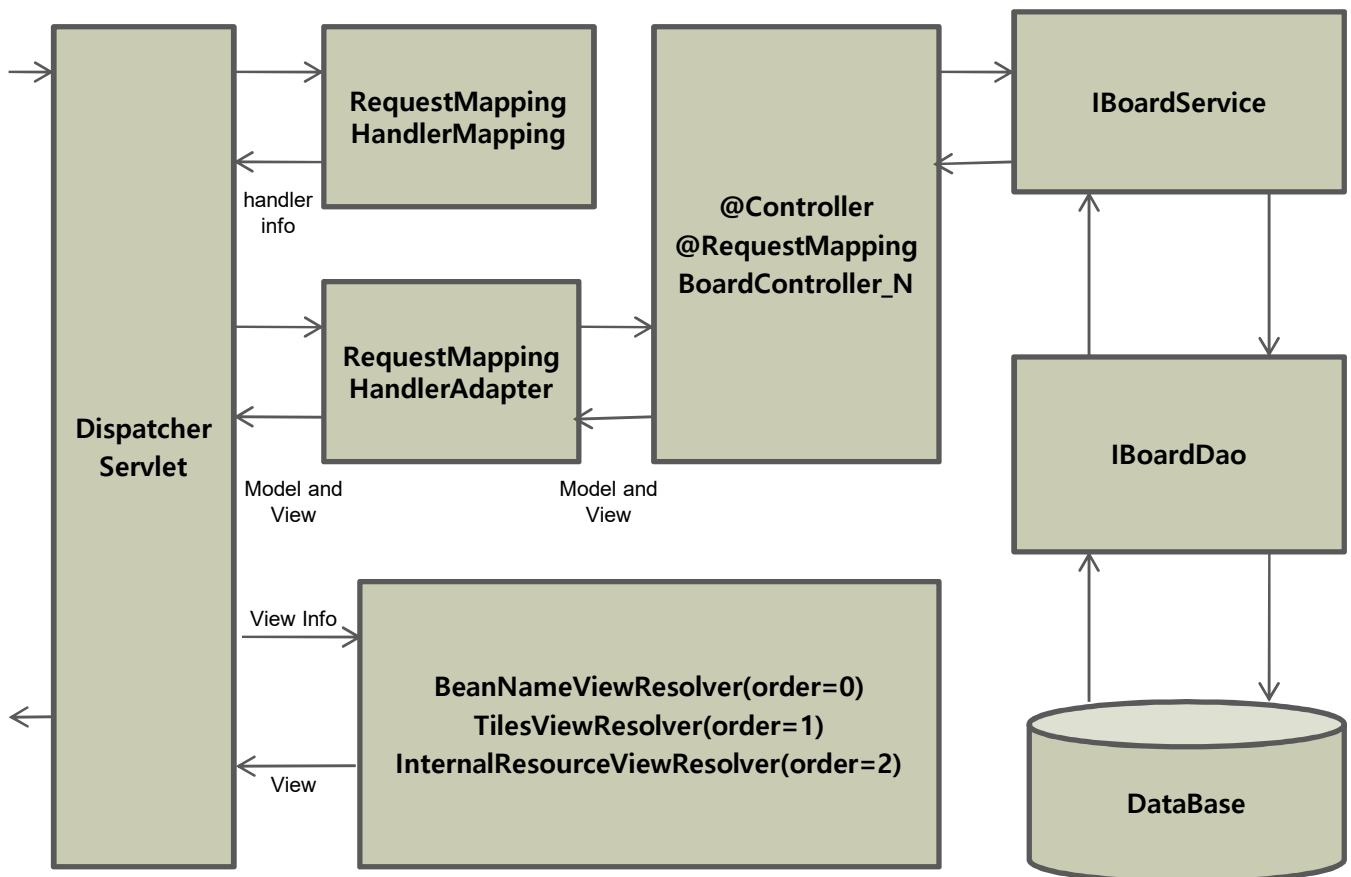
❑ Spring MVC module Architecture

출처 : spring.io (<https://docs.spring.io/spring/docs/4.3.26.RELEASE/spring-framework-reference>)

Request Process Flow

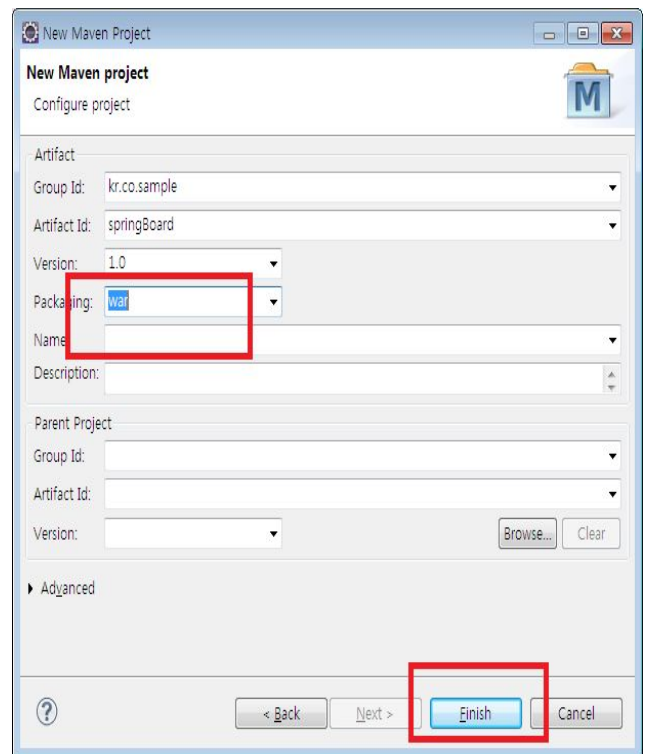
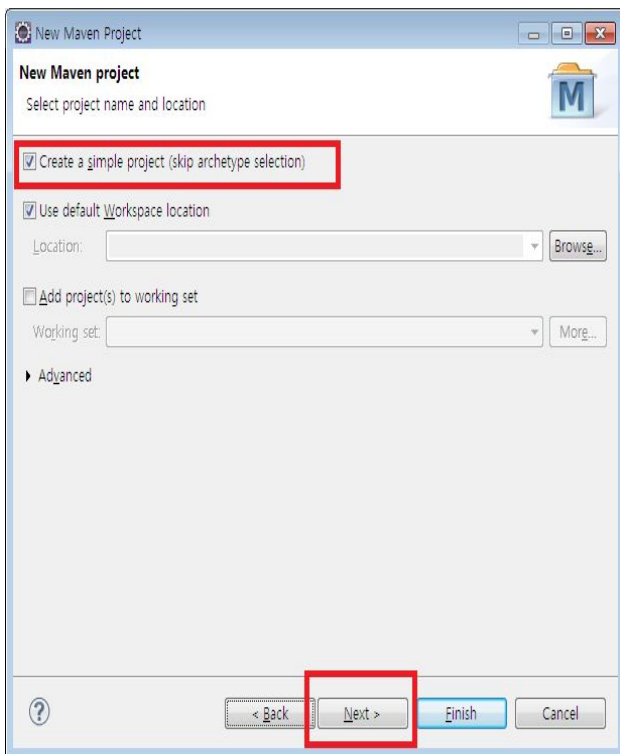


게시판 처리 흐름



□ 개발환경 설정(maven 및 필수 plugin 이 설치된 ide를 가정함.)

- Maven Project 생성



- 의존성 관리 및 플러그인 설정(pom.xml)

- 데이터베이스 드라이버는 별도 등록이 필요함

참고 : maven local repository 에 ojdbc6.jar 추가하는 방법

D:\W>mvn install:install-file -Dfile=ojdbc6.jar -DgroupId=com.oracle -DartifactId=ojdbc6

-Dversion=11.2.0.2.0 -Dpackaging=jar

(ojdbc6.jar 는 루트에 있음을 가정함.)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>kr.co.sample</groupId>
    <artifactId>springBoard</artifactId>
    <version>1.0</version>
    <packaging>war</packaging>
    <properties>
        <java-version>1.8</java-version>
        <log4j-version>2.13.0</log4j-version>
        <org.slf4j-version>1.7.9</org.slf4j-version>
        <org.springframework-version>4.3.26.RELEASE</org.springframework-version>
        <org.springframework.security-
version>4.2.13.RELEASE</org.springframework.security-version>
    </properties>
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-framework-bom</artifactId>
                <version>${org.springframework-version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
</project>
```

```

        <exclusions>
            <exclusion>
                <groupId>commons-logging</groupId>
                <artifactId>commons-logging</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
    <dependency>
        <groupId>org.springframework.security</groupId>
        <artifactId>spring-security-bom</artifactId>
        <version>${org.springframework.security-version}</version>
        <type>pom</type>
        <scope>import</scope>
    </dependency>
</dependencies>
</dependencyManagement>
<dependencies>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>javax.servlet-api</artifactId>
        <version>3.0.1</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>javax.servlet.jsp</groupId>
        <artifactId>jsp-api</artifactId>
        <version>2.2</version>
        <scope>provided</scope>
    </dependency>
    <!-- 스프링 의존성 -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
    </dependency>
    <dependency>
        <groupId>javax.inject</groupId>
        <artifactId>javax.inject</artifactId>
        <version>1</version>
    </dependency>
    <dependency>
        <groupId>org.aspectj</groupId>
        <artifactId>aspectjweaver</artifactId>
        <version>1.8.14</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
    </dependency>
    <!-- WebSocket 지원 -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-websocket</artifactId>
    </dependency>

```

```
<!-- spring - mybatis 연동 -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-orm</artifactId>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>2.0.3</version>
</dependency>
<!-- Spring Security -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-taglibs</artifactId>
</dependency>
<!-- 데이터베이스 접속 -->
<dependency>
    <groupId>com.oracle</groupId>
    <artifactId>ojdbc</artifactId>
    <version>11.2.0.2.0</version>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.3</version>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-dbcp2</artifactId>
    <version>2.4.0</version>
    <exclusions>
        <exclusion>
            <groupId>commons-logging</groupId>
            <artifactId>commons-logging</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<!-- Domain layer -->
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.10</version>
</dependency>
<!-- Marshalling -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.9.8</version>
</dependency>
```

```
<!-- commons 계열 -->
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.9</version>
</dependency>
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-compress</artifactId>
    <version>1.17</version>
</dependency>
<dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.13</version>
</dependency>
<!-- logging framework -->
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-api</artifactId>
    <version>${log4j-version}</version>
</dependency>
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>${log4j-version}</version>
</dependency>
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-slf4j-impl</artifactId>
    <version>${log4j-version}</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>${org.slf4j-version}</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>jcl-over-slf4j</artifactId>
    <version>${org.slf4j-version}</version>
</dependency>
<dependency>
    <groupId>org.lazyluke</groupId>
    <artifactId>log4jdbc-remix</artifactId>
    <version>0.2.7</version>
</dependency>
```

```
<!-- View layer 용 -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
<!-- Tiles framework -->
<dependency>
    <groupId>org.apache.tiles</groupId>
    <artifactId>tiles-jsp</artifactId>
    <version>3.0.8</version>
</dependency>
<!-- JSR-303 객체 검증용 -->
<dependency>
    <groupId>javax.validation</groupId>
    <artifactId>validation-api</artifactId>
    <version>1.1.0.Final</version>
</dependency>
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>4.3.2.Final</version>
</dependency>
<!-- test case 용 -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
<scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${org.springframework-version}</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-test</artifactId>
    <version>${org.springframework.security-version}</version>
    <scope>test</scope>
</dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
            <configuration>
                <source>${java-version}</source>
                <target>${java-version}</target>
            </configuration>
        </plugin>
    </plugins>
</build>
```

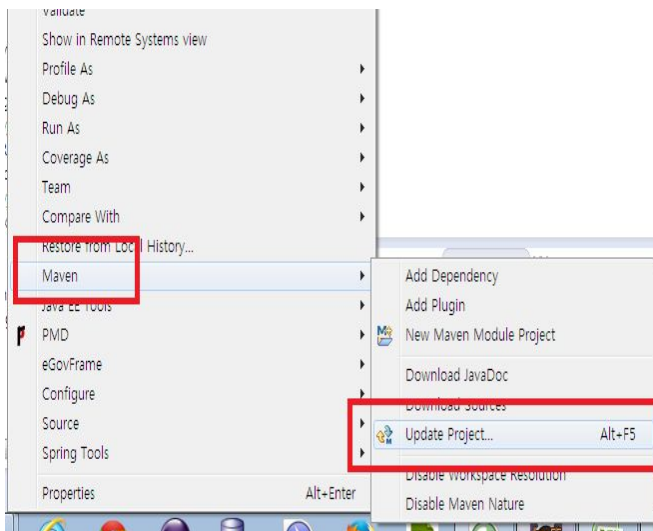


```

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>3.2.1</version>
  <configuration>
    <warSourceDirectory>src/main/webapp</warSourceDirectory>
    <failOnMissingWebXml>>false</failOnMissingWebXml>
  </configuration>
</plugin>
</plugins>
</build>
</project>

```

- Project update (프로젝트 우클릭, Maven -> Update Project)



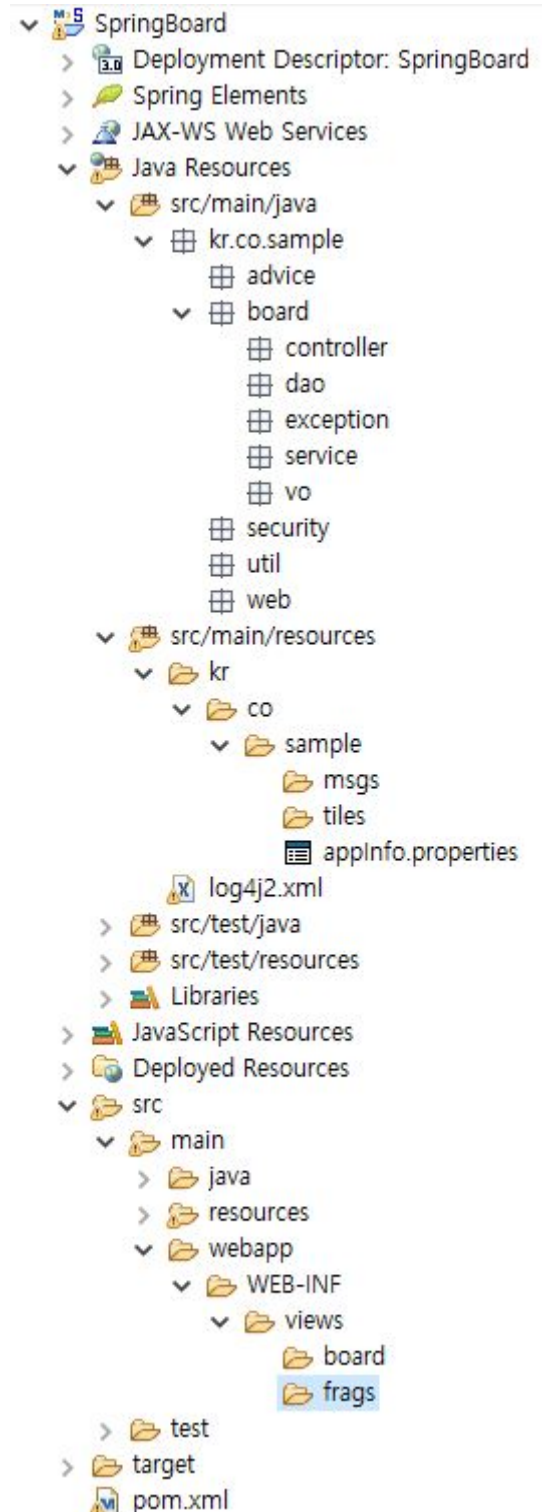
- Deployment Descriptor 추가



□ 구현

- 개발 영역 분리

- domain layer : basepackage.model명.vo
- persistence layer : basepackage.model명.dao
- business logic layer : basepackage.model명.service
- controller layer : basepackage.model명.controller
- presentation(view) layer : /WEB-INF/views/model명/
- advice 영역 : basepackage.advice
- web component 영역 : basepackage.web.???
- mapper 설정 영역 : basepackage.moel명.dao 내에 매퍼인터페이스와 같은 이름으로 작성함.
- message resource 영역 : basepackage.msgs
- bean meta data 등록 파일 영역
 1. 상위 : com.sample.conf.RootConfig...*
 2. 하위 : com.sample.conf.web.AppServletConfig



- 상/하위 컨텍스트를 형성할 Context listener 및 Front Controller 등록(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"
  version="3.0">
  <display-name>springBoard</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/*-context.xml</param-value>
  </context-param>
  <context-param>
    <param-name>spring.profiles.active</param-name>
    <param-value>dev</param-value>
  </context-param>
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <listener>
    <listener-class>org.springframework.security.web.session.HttpSessionEventPublisher</listener-class>
  </listener>
  <filter>
    <filter-name>CharacterEncodingFilter</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>forcedEncoding</param-name>
      <param-value>true</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>CharacterEncodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <filter>
    <filter-name>MultipartFilter</filter-name>
    <filter-class>org.springframework.web.multipart.support.MultipartFilter</filter-class>
    <init-param>
      <param-name>multipartResolverBeanName</param-name>
      <param-value>filterMultipartResolver</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>MultipartFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <!-- <filter> --> <!-- Spring Security 를 사용하는 경우 필요한 설정 -->
    <!-- <filter-name>springSecurityFilterChain</filter-name> -->
    <!-- <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class> -->
  <!-- </filter> -->
```

```

<!-- <filter-mapping> -->
    <!-- <filter-name>springSecurityFilterChain</filter-name> -->
    <!-- <url-pattern>/*</url-pattern> -->
<!-- </filter-mapping> -->
<servlet>
    <servlet-name>springDispatcherServlet</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>springDispatcherServlet</servlet-name>
    <url-pattern>/</url-pattern>
    <url-pattern>/index.html</url-pattern>
</servlet-mapping>
</web-app>

```

- 로깅을 위한 설정 파일 작성(src/main/resources/log4j2.xml)

```

<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
    <Appenders>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%-5level %logger{1.} - %msg%n" />
        </Console>
    </Appenders>
    <Loggers>
        <Logger name="jdbc.resultsettable" level="debug" />
        <Logger name="kr.co.sample" level="info" />
        <Logger name="org.springframework.security" level="debug" />
        <Logger name="org.springframework.web" level="info" />
        <Root level="error">
            <AppenderRef ref="Console" />
        </Root>
    </Loggers>
</Configuration>

```

- 게시글과 첨부파일을 위한 스키마(데이터베이스는 오라클이며, 이미 계정이 생성돼 있음을 가정함.)

```

-- 게시판 스키마
CREATE TABLE BOARD (
    BO_NO      NUMBER(7) NOT NULL,      -- 글번호(PK)
    BO_TITLE   VARCHAR2(100 CHAR) NOT NULL, -- 제목
    BO_WRITER  VARCHAR2(10 CHAR) NOT NULL, -- 글쓴이
    BO_PASSWD  VARCHAR2(20) NOT NULL,      -- 비번
    BO_EMAIL   VARCHAR2(100),             -- 이메일
    BO_IP      VARCHAR2(20) ,              -- 글쓴이 IP
    BO_CONTENT CLOB ,                      -- 내용
    BO_HIT     NUMBER(5) DEFAULT 0,        -- 조회수
    BO_DATE    DATE DEFAULT SYSDATE,       -- 최초 등록일
    --- 답변형을 위한 컬럼
    GROUP_NO   NUMBER(7) NOT NULL,        -- 계층에서 최상위에 있는 부모글 번호
    SEQUENCE_NO CHAR(6) NOT NULL          -- 답글 순서번호, 최대 깊이 3, 동일레벨에서 99개까지 허용함.
);

```

- 게시글과 첨부파일을 위한 스키마(데이터베이스는 오라클이며, 이미 계정이 생성돼 있음을 가정함.)

```

CREATE SEQUENCE BOARD_SEQ MAXVALUE 99999999;
COMMENT ON TABLE BOARD IS '게시판 테이블';
COMMENT ON COLUMN BOARD.BO_NO IS '글번호(PK)';
COMMENT ON COLUMN BOARD.BO_TITLE IS '제목';
COMMENT ON COLUMN BOARD.BO_WRITER IS '글쓴이';
COMMENT ON COLUMN BOARD.BO_PASSWD IS '비번';
COMMENT ON COLUMN BOARD.BO_EMAIL IS '이메일';
COMMENT ON COLUMN BOARD.BO_IP IS '글쓴이 IP';
COMMENT ON COLUMN BOARD.BO_CONTENT IS '내용';
COMMENT ON COLUMN BOARD.BO_HIT IS '조회수';
COMMENT ON COLUMN BOARD.BO_DATE IS '최초 등록일';
COMMENT ON COLUMN BOARD.GROUP_NO IS '그룹번호';
COMMENT ON COLUMN BOARD.SEQUENCE_NO IS '답글순서번호';
-- 첨부파일 스키마
CREATE TABLE PDS(
  PDS_NO NUMBER(7) NOT NULL,      -- PK
  PDS_REF_KEY NUMBER(7),          -- BOARD 의 BO_NO
  PDS_SAVENAME VARCHAR2(300 CHAR), -- 서버에 물리적으로 저장된 이름
  PDS_FILENAME VARCHAR2(100 CHAR), -- 사용자가 올렸던 원래 파일명
  PDS_FILESIZE NUMBER(10),        -- 파일의 사이즈 , LONG
  PDS_FANCYSIZE VARCHAR2(10 CHAR), -- CF: '4.56KB'
  PDS_DOWNCOUNT NUMBER(5) DEFAULT 0, -- 다운 횟수
  PDS_DATE DATE DEFAULT SYSDATE,    -- 업로드 날짜
  CONSTRAINT PK_PDS PRIMARY KEY (PDS_NO),
  CONSTRAINT FK_PDS FOREIGN KEY(PDS_REF_KEY) REFERENCES BOARD(BO_NO) ON DELETE CASCADE
);
CREATE SEQUENCE PDS_SEQ MAXVALUE 99999999 ;
COMMENT ON TABLE PDS IS '첨부파일 테이블';
COMMENT ON COLUMN PDS.PDS_NO IS '첨부파일번호(PK)';
COMMENT ON COLUMN PDS.PDS_REF_KEY IS '첨부게시물번호';
COMMENT ON COLUMN PDS.PDS_SAVENAME IS '서버에 저장시 사용된 파일명(경로미포함)';
COMMENT ON COLUMN PDS.PDS_FILENAME IS '사용자가 선택했던 원래 파일명';
COMMENT ON COLUMN PDS.PDS_FILESIZE IS '파일 크기';
COMMENT ON COLUMN PDS.PDS_FANCYSIZE IS '파일크기 축약형(뷰)';
COMMENT ON COLUMN PDS.PDS_DOWNCOUNT IS '다운 횟수';
COMMENT ON COLUMN PDS.PDS_DATE IS '업로드 날짜';

```

- 테스트용 dummy 데이터 입력

```

INSERT
INTO BOARD
(
  BO_NO,      BO_TITLE,      BO_WRITER,
  BO_PASSWD,  BO_EMAIL,      BO_IP,
  BO_CONTENT, BO_HIT,        BO_DATE, GROUP_NO, SEQUENCE_NO
)
SELECT BOARD_SEQ.NEXTVAL, '테스트 입력 제목_'||BOARD_SEQ.CURRVAL,
       '작성자_'||BOARD_SEQ.CURRVAL, '0000', 'TESTER@GMAIL.COM',
       '255.255.255.255',
       '내용정보 입력<BR />제목_'||BOARD_SEQ.CURRVAL||'<BR />작성자_'
       ||BOARD_SEQ.CURRVAL||'<BR />님이 작성한 글의 내용입니다.',
       BOARD_SEQ.CURRVAL*100, SYSDATE, BOARD_SEQ.CURRVAL, '9999999'
FROM DUAL;

```

- 스프링 상위 컨텍스트 bean metadata configuration file 작성. (/WEB-INF/spring/datasource-context.xml)
 - 참고 : pdf 마지막장의 GlobalResource 등록 참고.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:c="http://www.springframework.org/schema/c"
  xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:jee="http://www.springframework.org/schema/jee"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring-1.2.xsd
    http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-4.3.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-4.3.xsd
    http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd">

  <tx:annotation-driven/>
  <bean id="dataSource" class="net.sf.log4jdbc.Log4jdbcProxyDataSource"
    c:realDataSource-ref="realDataSource"
  />
  <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"
    p:dataSource-ref="dataSource"
    p:typeAliasesPackage="kr.co.sample"
    p:mapperLocations="classpath:kr/co/sample/board/dao/*.xml"
  />
  <mybatis-spring:scan base-package="kr.co.sample.**.dao"
    factory-ref="sqlSessionFactory"
  />
  <!-- 등록된 빈들을 그룹별로 관리하기 위한 profile 설정 -->
  <beans profile="dev" >
    <context:property-placeholder location="classpath:kr/co/sample/db.properties" />
    <bean id="realDataSource" class="org.apache.commons.dbcp2.BasicDataSource"
      p:driverClassName="${db.driverClassName}"
      p:url="${db.url}"
      p:username="${db.username}"
      p:password="${db.password}"
      p:maxTotal="${db.maxActive}"
      p:maxWaitMillis="${db.maxWait}"
    />
  </beans>
  <beans profile="stage" >
    <jee:jndi-lookup id="realDataSource" jndi-name="jdbc/mainDB"
      expected-type="javax.sql.DataSource"></jee:jndi-lookup>
  </beans>
</beans>
```

- /WEB-INF/spring/root-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:util="http://www.springframework.org/schema/util"
    xmlns:c="http://www.springframework.org/schema/c"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="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-4.3.xsd
        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
        http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.3.xsd">

    <context:component-scan base-package="kr.co.sample">
        <context:include-filter type="annotation" expression="org.aspectj.lang.annotation.Aspect"/>
        <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"/>
        <context:exclude-filter type="annotation"
            expression="org.springframework.web.bind.annotation.ControllerAdvice"/>
    </context:component-scan>

    <aop:aspectj-autoproxy />

    <util:properties id="appInfo" location="classpath:kr/co/sample/appInfo.properties" />

    <!--
Spring Security 를 사용하는 경우, springSecurityFilterChain 보다 먼저 multipart request 에 대한 파싱이 필요함.
이 경우, MultipartFilter 를 등록해서 사용하게 되는데, 서블릿 컨테이너에 등록된 필터에서 Spring 컨테이너 내에
등록된 전략 빈을 주입받는 경우, 해당 빈은 상위 컨테이너에 등록되어 있어야 함.
multipartResolver 의 id 를 변경하는 경우, MultipartFilter 에 targetBeanName 파라미터 설정이 필요함.
-->
    <bean id="filterMultipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"
        p:defaultEncoding="#{appInfo.pdsDefaultEncoding}"
        p:uploadTempDir="#{appInfo.pdsTempDir}"
        p:maxInMemorySize="#{appInfo.pdsMaxInMemorySize}"
        p:maxUploadSize="#{appInfo.pdsMaxUploadSize}"
        p:maxUploadSizePerFile="#{appInfo.pdsMaxUploadSizePerFile}"
    />

    <bean id="messageSource" class="org.springframework.context.support.ReloadableResourceBundleMessageSource"
        p:basenames="classpath:kr/co/sample/msgs/message,classpath:org/springframework/security/messages"
        p:cacheSeconds="-1"
    />

    <bean id="messageSourceAccessor" class="org.springframework.context.support.MessageSourceAccessor"
        c:messageSource-ref="messageSource"
    />
</beans>
```

- Spring Security 를 이용하는 경우
- /WEB-INF/spring/security-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans
    xmlns="http://www.springframework.org/schema/security"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:c="http://www.springframework.org/schema/c"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-4.2.xsd">
    <!-- 인증 처리에 필요한 설정 -->
    <beans:bean id="userDetailsService" class="kr.co.sample.security.CustomUserDetailsService"
        p:dataSource-ref="dataSource"
        p:enableAuthorities="true"
        p:usersByUsernameQuery="SELECT * FROM USERS WHERE USER_ID = ?"
        p:authoritiesByUsernameQuery="SELECT * FROM USERROLES WHERE USER_ID = ?"
    />
    <beans:bean id="passwordEncoder" class="org.springframework.security.crypto.factory.PasswordEncoderFactories"
        factory-method="createDelegatingPasswordEncoder" />
    <beans:bean id="authenticationProvider"
        class="org.springframework.security.authentication.dao.DaoAuthenticationProvider"
        p:userDetailsService-ref="userDetailsService"
        p:passwordEncoder-ref="passwordEncoder"
        p:hideUserNotFoundExceptions="false"
    />
    <authentication-manager id="authenticationManager" erase-credentials="false">
        <authentication-provider ref="authenticationProvider" />
    </authentication-manager>

    <beans:bean id="grantedAuthorityDefaults" class="org.springframework.security.config.core.GrantedAuthorityDefaults"
        c:rolePrefix="ROLE_"
    />
    <beans:bean id="roleHierarchyString" class="kr.co.sample.security.hierarchicalRole.RoleHierarchyStringFactoryBean"
        p:dataSource-ref="dataSource"
    />
    <!-- 권한 계층 구조에 따른 접근 제어를 위한 전략 -->
    <beans:bean id="roleHierarchy" class="org.springframework.security.access.hierarchicalroles.RoleHierarchyImpl"
        p:hierarchy-ref="roleHierarchyString"
    />
    <!-- 접근 제어 기반의 인가 처리에 필요한 설정 -->
    <beans:bean id="accessDecisionManager" class="org.springframework.security.access.vote.AffirmativeBased"
        p:allowIfAllAbstainDecisions="false">
        <beans:constructor-arg name="decisionVoters">
            <beans:list>
                <beans:bean class="org.springframework.security.access.vote.AuthenticatedVoter" />
                <!-- 권한 소유자에 한해 접근을 허용하기 위한 voter -->
                <beans:bean class="org.springframework.security.access.vote.RoleVoter" />
                <beans:bean class="org.springframework.security.access.vote.RoleHierarchyVoter"
                    c:roleHierarchy-ref="roleHierarchy"
                />
            </beans:list>
        </beans:constructor-arg>
    </beans:bean>
```

```

        <beans:bean
            class="org.springframework.security.web.access.expression.WebExpressionVoter" />
    </beans:list>
</beans:constructor-arg>
</beans:bean>
<!-- DB 에 있는 접근 제어 정보를 조회하기 위한 전략 -->
<beans:bean id="requestMap" class="kr.co.sample.security.RequestMapFactoryBean"
    p:dataSource-ref="dataSource"
/>
<!-- spEL 의 표현식을 security expression 형태로 eval 하는 객체 -->
<beans:bean id="webSecurityExpressionHandler"
    class="org.springframework.security.web.access.expression.DefaultWebSecurityExpressionHandler" />
<!-- expressionHandler 를 이용하여, 표현식으로부터 적절한 접근제어 설정객체를 생성하는 전략-->
<beans:bean id="filterInvocationSecurityMetadataSource"
    class="org.springframework.security.web.access.expression.ExpressionBasedFilterInvocationSecurityMetadataSource"
    c:requestMap-ref="requestMap"
    c:expressionHandler-ref="webSecurityExpressionHandler"
/>
<!-- Filter의 역할을 수행하는 전략 -->
<beans:bean id="filterSecurityInterceptor"
    class="org.springframework.security.web.access.intercept.FilterSecurityInterceptor"
    p:authenticationManager-ref="authenticationManager"
    p:accessDecisionManager-ref="accessDecisionManager"
    p:securityMetadataSource-ref="filterInvocationSecurityMetadataSource"
/>
<!-- <security:authorize url="/member/memberList.do"></security:authorize> 와 같은 형태의 선택적 랜더링 적용을 위한 전략빈 -->
<beans:bean class="org.springframework.security.web.access.DefaultWebInvocationPrivilegeEvaluator"
    c:securityInterceptor-ref="filterSecurityInterceptor"
/>
<!-- Spring security filter chain 으로 필터링 할 필요 없는, 보호가 필요없는 자원에 대한 ignore 설정 -->
<http pattern="/resources/**" security="none" />
<!-- http 요청에 대해 spring security filter chain 을 단계적으로 통과시켜 보호하기 위한 설정 -->
<http
    authentication-manager-ref="authenticationManager"
    access-decision-manager-ref="accessDecisionManager"
    use-expressions="true">
    <form-login login-page="/login" login-processing-url="/login"
        username-parameter="user_id" password-parameter="password"
        default-target-url="/" always-use-default-target="false"/>
    <logout logout-url="/logout" logout-success-url="/" />
    <custom-filter ref="filterSecurityInterceptor" before="FILTER_SECURITY_INTERCEPTOR" />
    <remember-me remember-me-parameter="rememberMe" token-validity-seconds="#{60*60*24*7}"/>
    <session-management>
        <concurrency-control max-sessions="1" expired-url="/login?error=true"/>
    </session-management>
    <headers>
        <xss-protection block="true" enabled="true"/>
    </headers>
</http>

<!-- matrix 변수에 대한 처리를 위해 세미콜론이 포함된 url 을 허용하기 위한 설정 -->
<beans:bean id="httpFirewall" class="org.springframework.security.web.firewall.StrictHttpFirewall"
    p:allowSemicolon="true" />

<http-firewall ref="httpFirewall" />
</beans:beans>

```


- 스프링 하위 컨텍스트 bean metadata config class 작성. (/WEB-INF/spring/appServlet/servlet-context.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:c="http://www.springframework.org/schema/c"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="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-4.3.xsd
        http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd">

    <context:component-scan base-package="kr.co.sample" use-default-filters="false">
        <context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>
        <context:include-filter type="annotation"
            expression="org.springframework.web.bind.annotation.ControllerAdvice"/>
    </context:component-scan>

    <!-- enable-matrix-variables="true" , java config 의 urlPathHelper.setRemoveSemicolonContent(false); -->
    <mvc:annotation-driven enable-matrix-variables="true" >
        <!-- request 로부터 핸들러 메소드의 아규먼트를 추출하기 위한 전략, 아규먼트 타입별 처리자 -->
        <mvc:argument-resolvers>
            <bean class="kr.co.sample.web.PdsVOMethodArgumentResolver" />
            <bean
                class="org.springframework.security.web.method.annotation.AuthenticationPrincipalArgumentResolver" />
        </mvc:argument-resolvers>

        <!--
        @RequestBody 나 @ResponseBody 를 사용하는 경우, request 로부터의 객체 변환,
        response 로의 객체 변환을 처리하는 전략
        -->
        <mvc:message-converters>
            <bean class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter"
                p:supportedMediaTypes="application/json;charset=UTF-8"
            />
        </mvc:message-converters>
    </mvc:annotation-driven>

    <bean id="boardValidator" class="kr.co.sample.web.validator.BoardValidator" />

    <mvc:tiles-configurer preparer-factory="org.springframework.web.servlet.view.tiles3.SpringBeanPreparerFactory">
        <mvc:definitions location="classpath:kr/co/sample/tiles/*.xml"/>
    </mvc:tiles-configurer>

    <bean id="downloadView" class="kr.co.sample.web.view.DownloadView" />
    <bean id="BNVR" class="org.springframework.web.servlet.view.BeanNameViewResolver"
        p:order="1"
    />
</beans>
```

```
<!-- 일반적인 ViewResolver들의 order 값이 Ordered.LOWEST_PRECEDENCE 임에 반해 -->
<!--
ContentNegotiatingViewResolver 는 반대로 HIGHH_PRECEDENCE 로 되어있어, 명시적인 order 설정이 필요함.
-->
<bean id="CNVR" class="org.springframework.web.servlet.view.ContentNegotiatingViewResolver"
      p:order="#{T(org.springframework.core.Ordered).LOWEST_PRECEDENCE}">
  <property name="defaultViews">
    <list>
      <bean class="org.springframework.web.servlet.view.json.MappingJackson2JsonView" />
      <ref bean="downloadView"/>
    </list>
  </property>
  <property name="viewResolvers">
    <list>
      <bean class="org.springframework.web.servlet.view.tiles3.TilesViewResolver" />
      <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"
            p:prefix="/WEB-INF/views/" p:suffix=".jsp" />
    </list>
  </property>
</bean>

<bean id="localeResolver" class="org.springframework.web.servlet.i18n.CookieLocaleResolver"
      p:cookiePath="/"
      p:cookieMaxAge="#{60*60*24*7}" />

<mvc:interceptors>
<bean id="localeChangeInterceptor" class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor"
      p:paramName="lang" />
</mvc:interceptors>

<mvc:resources location="/resources/" mapping="/resources/**" cache-period="0" />

<mvc:default-servlet-handler />

<mvc:view-controller path="/index.html" view-name="index"/>

</beans>
```

- 시스템 기본 설정 파일 작성 (src/main/resources/kr/co/sample/applInfo.properties)

```
#업로드 위치
pdsSaveUrl=/upload
#업로드 임시 디렉토리
pdsTempDir=file:d:/temp
#최대 업로드사이즈 및 메모리 저장 임계치 : CommonsMultipartResolver 를 사용하는 경우.
pdsMaxUploadSize=1048576000
pdsMaxUploadSizePerFile=104857600
pdsMaxInMemorySize=10240
#기본 인코딩
pdsDefaultEncoding=UTF-8
#답글 최대 깊이
LEVELLIMIT=3
#동일레벨 최대 답글수를 의미하는 '9'의 갯수
SAMELEVELCHARS=2
```

- 다국적 메시지 처리를 위한 메시지 번들 작성(src/main/resources/kr/co/sample/messages/message_ko.properties)

```
#한글 메시지
# Spring Code resolver 의 메시지 검색 단계.
#1. ErrorCode.model명.field명
#2. ErrorCode.field명
#3. ErrorCode.model명
#4. ErrorCode
board.bo_no=글번호
board.bo_title=글제목
board.bo_writer=작성자
board.bo_email=이메일
board.bo_passwd=비밀번호
board.bo_content=글내용
board.bo_date=등록일
board.bo_hit=조회수
## 에러처리를 위한 메시지
NotEmpty={0}, 필수 입력 항목
Length={0} 은 {1}~{2}문자 이내로 입력
Email=이메일 형식 확인
```

- 게시글과 첨부파일 및 페이징 처리를 위한 VO 작성

```
package kr.co.sample.board.vo;
@Data
@NoArgsConstructor
public class PdsVO implements Serializable{
    public PdsVO(MultipartFile uploadFile){
        this.uploadFile = uploadFile;
        pds_filename = uploadFile.getOriginalFilename();
        pds_filesize = uploadFile.getSize();
        pds_fancysize = FileUtils.byteCountToDisplaySize(pds_filesize);
        pds_savename = UUID.randomUUID().toString();
    }
    private MultipartFile uploadFile; //업로드된 파일
    private Integer pds_no; //첨부파일번호(PK)
    private Integer pds_ref_key; //첨부게시물번호
    private String pds_savename; //서버에 저장시 사용된 파일명(경로포함)
    private String pds_filename; //사용자가 선택했던 원래 파일명
    private Long pds_filesize; //파일 크기
```

```

private String pds_fancysize; //파일크기 축약형(뷰)
private Integer pds_downcount; //다운 횟수
private String pds_date; //업로드 날짜
private MultipartFile uploadFile; //업로드된 파일
public void saveToFolder(File folder) throws IOException{
    if(uploadFile.isEmpty()) return;
    uploadFile.transferTo(new File(folder, pds_savename));
}
public byte[] getBytes() throws IOException{
    if(uploadFile == null || uploadFile.isEmpty()) return null;
    return uploadFile.getBytes();
}
}

```

```

package kr.co.sample.board.vo;
public class BoardVO implements Serializable{
    private Integer bo_no; //글번호(PK)
    private String bo_title; //제목
    private String bo_writer; //글쓴이
    private String bo_passwd; //비번
    private String bo_email; //이메일
    private String bo_ip; //글쓴이 IP
    private String bo_content; //내용
    private Integer bo_hit; //조회수
    private String bo_date; //최초 등록일

    public static final int LEVELLIMIT = 3;
    public static final int SAMELEVELCHARS=2;

    private Integer group_no; //그룹 번호
    private String sequence_no; //답글 순서번호
    private String bo_parent; //답글일 경우 부모글번호 (부모글 검증에만 사용됨)

    private Integer pdssize; // 첨부파일 개수 조회시 사용
    private Integer[] delPdsNos; //삭제할 첨부파일들의 번호

    private Integer pdsNoStart; // 여러 개의 첨부파일 업로드시 pds_no 시작값을 받기위한 프로퍼티
    private List<PdsVO> pdsList; // getter , setter 선언.
    public int getLevel(){
        Pattern pattern = Pattern.compile("([9]{"+SAMELEVELCHARS+"})$");
        if(sequence_no==null){
            return -1;
        }else if(sequence_no.length() != (LEVELLIMIT * SAMELEVELCHARS)){
            return -1;
        }else{
            Matcher matcher = pattern.matcher(sequence_no);
            // 레벨당 '9' 문자수(2)횟수로 반복되는 패턴이 없다면, 가장 마지막 레벨임을 의미함.
            int findIdx = LEVELLIMIT * SAMELEVELCHARS;
            if(matcher.find()){
                findIdx = matcher.start(); // '99'가 처음 시작되는 위치
            }
            // 찾아진 index 범위 : 0(0레벨, 루트글), 1~2(1레벨), 3~4(2레벨), 5~6(3레벨)...
            return (findIdx + (SAMELEVELCHARS - 1)) / SAMELEVELCHARS;
        }
    }
}

```

```

package kr.co.sample.board.vo;

import java.util.List;
/**
 * paging 처리를 위한 프로퍼티를 갖는 VO
 */
public class PagingInfoVO<T> {
    private int blockSize=5;
    private int screenSize=10;
    private int currentPage;
    private int startRow;
    private int endRow;
    private int startPage;
    private int endPage;
    private int totalPages;
    private int totalRecords;
    private String pagingHTML;
    private String jsFuncName = "goPage";
    private T data;
    private List<T> dataList;
    public PagingInfoVO(){
    public PagingInfoVO(int blockSize, int screenSize) {
        this.blockSize = blockSize;
        this.screenSize = screenSize;
    }
    public void setTotalRecords(int totalRecords) {
        this.totalRecords = totalRecords;
        totalPages = (int) Math.ceil((totalRecords/(double)screenSize));
        endRow = currentPage * screenSize;
        startRow = endRow - (screenSize-1);
        startPage = ((currentPage-1)/blockSize)*blockSize + 1;
        endPage = startPage + (blockSize-1);
        endPage = totalPages<endPage? totalPages : endPage;
    }
    public String getPagingHTML(){
        StringBuffer sb = new StringBuffer("");
        if (startPage > 1) {
            sb.append("<a href='javascript:' + jsFuncName + \"(" + (startPage - blockSize) + \")'>이전</a>");
        }
        for (int i = startPage; i <= endPage; i++) {
            if (currentPage == i) {
                sb.append("[ " + i + " ]");
            } else {
                sb.append("<a href='javascript:' + jsFuncName + \"(" + i + \")'>" + i + "</a>");
            }
        }
        if (endPage < totalPages) {
            sb.append("<a href='javascript:' + jsFuncName + \"(" + (endPage + 1) + \")'>다음</a>");
        }
        return sb.toString();
    }
    // 그외 getter, setter 선언
}

```

- 게시글과 첨부파일 처리를 위한 DAO(Data Access Object) 작성

```
package kr.co.sample.board.dao;

/**
 * 게시판 관리를 위한 DataAccessObject interface, Persistence Layer
 */
public interface IBoardDAO {
    /**
     * 부모글과 부모글 바로 다음에 작성된 동일 레벨의 글 사이의 시퀀스 번호들 중 가장 작은
     * @param group_no, maxSeq 부모글의 시퀀스번호(989999), minSeq(980000)
     * @return
     */
    public String selectMinSequenceNoWithSameGroup(Map<String, Object> paramMap);
    /**
     * 새글 등록
     * @param board
     * @return
     */
    public Integer insertBoard(BoardVO board);
    /**
     * 특정 글 조회
     * @param bo_no 조회할 글 번호
     * @return
     */
    public BoardVO selectBoard(Integer bo_no);
    /**
     * 글 조회수 카운트
     * @param bo_no
     * @return
     */
    public int incrementHit(Integer bo_no);
    /**
     * 글 목록 조회
     * @param pagingInfoVO
     * @return
     */
    public List<BoardVO> selectBoardList(PagingInfoVO<BoardVO> pagingInfoVO);
    /**
     * 게시글 목록 수 조회
     * @param pagingInfoVO
     * @return
     */
    public int selectBoardCount(PagingInfoVO<BoardVO> pagingInfoVO);
    /**
     * 글 수정
     * @param board
     * @return
     */
    public int updateBoard(BoardVO board);
    /**
     * 글 삭제
     * @param bo_no
     * @return
     */
    public int deleteBoard(Integer bo_no);
}
```

```

package kr.co.sample.board.dao;
/**
 * 첨부파일의 메타데이터를 위한 CRUD interface
 */
public interface IPdsDAO {

    /**
     * 첨부파일 저장을 위한 메소드
     * @param board 첨부파일을 가진 게시글
     * @return
     */
    public int insertPdsList(BoardVO board);

    /**
     * 첨부파일 조회를 위한 메소드
     * @param pds_no
     * @return
     */
    public PdsVO selectPds(Integer pds_no);

    /**
     * downcount 증가를 위한 메소드
     * @param pds_no
     * @return
     */
    public int incrementDowncount(Integer pds_no);

    /**
     * 첨부파일 삭제를 위한 메소드
     * @param pds_no
     * @return
     */
    public int deletePds(Integer pds_no);

    /**
     * 여러 개의 첨부파일 삭제를 위한 메소드
     * @param board
     * @return
     */
    public int deletePdses(BoardVO board);
}

```

- Persistence layer 에서 데이터에 접근하기 위한 매핑 파일 작성
(src/main/java/kr/co/sample/board/dao/IPdsDAO.xml)

```

<?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="kr.co.sample.board.dao.IPdsDAO">
    <sql id="pdsSelectFrag">
        PDS_NO, PDS_REF_KEY, PDS_SAVENAME,
        PDS_FILENAME, PDS_FILESIZE, PDS_FANCYSIZE,
        PDS_DOWNCOUNT, TO_CHAR(PDS_DATE, 'YYYY-MM-DD') PDS_DATE
    </sql>

```

```

<insert id="insertPdsList" parameterType="boardVO">
    <select id="pdsSelectFrag" order="BEFORE" id="pdsNoStart">
        SELECT PDS_SEQ.NEXTVAL FROM DUAL
    </select>
    INSERT ALL
    <foreach collection="pdsList" item="pds" index="idx">
        INTO PDS (
            PDS_NO, PDS_REF_KEY, PDS_SAVENAME,
            PDS_FILENAME, PDS_FILESIZE, PDS_FANCYSIZE,
            PDS_DOWNCOUNT, PDS_DATE
        ) VALUES (
            #{pdsNoStart}+#{idx}, #{pds.pds_ref_key}, #{pds.pds_savename},
            #{pds.pds_filename}, #{pds.pds_filesize}, #{pds.pds_fancysize},
            0, SYSDATE
        )
    </foreach>
    SELECT * FROM DUAL
</insert>

<select id="selectPds" parameterType="int" resultType="pdsVO">
    SELECT
    <include refid="pdsSelectFrag" />
    FROM PDS
    WHERE PDS_NO = #{pds_no}
</select>

<select id="selectPdsListByBo_no" parameterType="int" resultType="pdsVO">
    SELECT
    <include refid="pdsSelectFrag" />
    FROM PDS
    WHERE PDS_REF_KEY = #{bo_no}
</select>

<update id="incrementDowncount" parameterType="int">
    UPDATE PDS
    SET
    PDS_DOWNCOUNT = PDS_DOWNCOUNT+1
    WHERE PDS_NO = #{pds_no}
</update>

<delete id="deletePds" parameterType="int">
    DELETE
    FROM PDS
    WHERE PDS_NO = #{pds_no}
</delete>

<delete id="deletePdses" parameterType="boardVO">
    DELETE FROM PDS
    WHERE PDS_NO IN
    <foreach collection="delPdsNos" item="delNo" open="(" close=")" separator=",">
        #{delNo}
    </foreach>
</delete>
</mapper>

```


- Persistence layer 에서 데이터를 접근하기 위한 매핑 파일 작성
(src/main/resources/kr/co/sample/board/dao/IBoardDAO.xml)

```
<?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="kr.co.sample.board.dao.IBoardDAO">
    <resultMap type="boardVO" id="boardMap" >
        <id property="bo_no" column="BO_NO" />
        <collection property="pdsList" javaType="java.util.List" ofType="pdsVO"
            autoMapping="true">
            <id property="pds_no" column="PDS_NO" />
        </collection>
    </resultMap>

    <sql id="boardSelectFrag">
        BO_NO, BO_TITLE, BO_WRITER, BO_PASSWD,
        BO_EMAIL, BO_IP, BO_CONTENT, BO_HIT,
        TO_CHAR(BO_DATE, 'yyyy-MM-dd') BO_DATE,
        GROUP_NO, SEQUENCE_NO
    </sql>

    <select id="selectBoard" parameterType="int" resultMap="boardMap">
        SELECT
        <include refid="boardSelectFrag" />
        ,
        <include refid="kr.co.sample.board.dao.IPdsDAO.pdsSelectFrag" />
        FROM BOARD LEFT OUTER JOIN PDS ON (BOARD.BO_NO = PDS_REF_KEY)
        WHERE BOARD.BO_NO = #{bo_no}
    </select>

    <update id="incrementHit" parameterType="int">
        UPDATE BOARD
        SET
        BO_HIT = BO_HIT + 1
        WHERE BO_NO = #{bo_no}
    </update>

    <!-- 검색 조건 -->
    <sql id="searchFrag">
        <where>
            <if test="@org.apache.commons.lang3.StringUtils@isNotBlank(data.bo_title)">
                BO_TITLE LIKE ('%' || #data.bo_title# || '%')
            </if>
            <if test="@org.apache.commons.lang3.StringUtils@isNotBlank(data.bo_writer)">
                BO_WRITER LIKE ('%' || #data.bo_writer# || '%')
            </if>
            <if test="@org.apache.commons.lang3.StringUtils@isNotBlank(data.bo_content)">
                BO_CONTENT LIKE ('%' || #data.bo_content# || '%')
            </if>
        </where>
    </sql>
```

```

<select id="selectBoardCount" resultType="int" parameterType="pagingInfoVO">
    SELECT COUNT(BO_NO)
    FROM BOARD
    <include refid="searchFrag" />
</select>

<select id="selectBoardList" resultType="boardVO" parameterType="pagingInfoVO">
    SELECT B.*
    FROM (SELECT ROWNUM RNUM, A.*
          FROM (
              SELECT
                  <include refid="boardSelectFrag" />
                  , (SELECT COUNT(*) FROM PDS WHERE PDS_REF_KEY = BOARD.BO_NO)
                  PDSSIZE
              FROM BOARD
              <include refid="searchFrag" />
              ORDER BY GROUP_NO DESC, SEQUENCE_NO DESC
          ) A
        ) B
    <![CDATA[
    WHERE RNUM >= #{startRow} AND RNUM <= #{endRow}
    ]]>
</select>

<!-- 동일 그룹내 동일 레벨의 다음 답글을 작성하기 위해 sequence_no 를 조회함 -->
<select id="selectMinSequenceNoWithSameGroup" resultType="string" parameterType="map">
    SELECT MIN(SEQUENCE_NO)
    FROM BOARD
    <![CDATA[ WHERE GROUP_NO = #{group_no} AND
    ( SEQUENCE_NO <= #{maxSeq} AND SEQUENCE_NO >= #{minSeq} ) ]]>
</select>

<update id="updateBoard" parameterType="boardVO">
    UPDATE BOARD
    SET
    BO_TITLE =#{bo_title},
    BO_WRITER =#{bo_writer},
    BO_EMAIL =#{bo_email},
    BO_IP =#{bo_ip},
    BO_CONTENT =#{bo_content}
    WHERE BO_NO =#{bo_no}
</update>

<!-- 게시글 삭제시, 해당 글을 루트로 시작된 계층구조 상의 모든 답글을 함께 삭제함 -->
<delete id="deleteBoard" parameterType="int">
    DELETE
    FROM BOARD
    WHERE GROUP_NO = (SELECT GROUP_NO FROM BOARD WHERE BO_NO = #{bo_no})
    AND SEQUENCE_NO LIKE (SELECT REPLACE(SEQUENCE_NO, '99', '') || '%'
    FROM BOARD
    WHERE BO_NO = #{bo_no})
</delete>

```

```
<insert id="insertBoard" parameterType="boardVO">
    <selectKey resultType="int" keyProperty="bo_no" order="BEFORE">
        <!-- PK를 생성하는 select 쿼리 -->
        SELECT BOARD_SEQ.NEXTVAL FROM DUAL
    </selectKey>
    INSERT INTO BOARD
    (
        BO_NO, BO_TITLE, BO_WRITER, BO_PASSWD, BO_EMAIL,
        BO_IP, BO_CONTENT, BO_HIT, BO_DATE, GROUP_NO
        <if test="@org.apache.commons.lang3.StringUtils@isNotBlank(sequence_no)">
            , SEQUENCE_NO
        </if>
    )
    VALUES
    (
        #{bo_no}, #{bo_title}, #{bo_writer},
        #{bo_passwd}, #{bo_email}, #{bo_ip},
        #{bo_content}, 0, SYSDATE
        <if test="group_no != null">
            , #{group_no}
        </if>
        <if test="group_no == null">
            , #{bo_no}
        </if>
        <if
            test="@org.apache.commons.lang3.StringUtils@isNotBlank(sequence_no)">
            , #{sequence_no}
        </if>
    )
</insert>
</mapper>
```

- Connection 설정 정보 파일 관리(src/main/resources/kr/co/sample/db.properties)

```
db.driverClassName=oracle.jdbc.driver.OracleDriver
db.url=jdbc:oracle:thin:@localhost:1521:xe
db.username=sem
db.password=java
db.initialSize=5
db.maxActive=20
db.maxWait=120000
```

- 게시글과 첨부파일 처리를 위한 Business Logic Object(Service 객체) 작성

```
package kr.co.sample;

public enum CommonResult {
    OK, FAIL, INVALIDPASSWORD, NOTEXIST, PKDUPLICATE
}
```

```
package kr.co.sample.board.service;

/**
 * 게시판 관리를 위한 Business Logic Object interface
 */
public interface IBoardService {
    /**
     * 신규 글 생성
     * @param board
     * @return
     */
    public CommonResult createBoard(BoardVO board) throws IOException;
    /**
     * 특정 글 조회
     * @param bo_no
     * @return
     */
    public BoardVO getBoard(Integer bo_no);
    /**
     * 특정 구간의 게시글 목록 조회(페이징 처리에 따른 조회)
     * 게시글 목록수 조회
     * @param pagingInfoVO
     * @return
     */
    public int getBoardCount(PagingInfoVO<BoardVO> pagingInfoVO);
    /**
     * @param pagingInfoVO
     * @return
     */
    public List<BoardVO> getBoardList(PagingInfoVO<BoardVO> pagingInfoVO);
    /**
     * 게시글 수정
     * @param board
     * @return
     */
    public CommonResult modifyBoard(BoardVO board) throws IOException;
    /**
     * 게시글 삭제
     * @param board
     * @return
     */
    public CommonResult removeBoard(BoardVO board) throws IOException;
    /**
     * 특정 첨부파일 조회
     * @param pds_no 조회할 첨부파일 번호
     * @return
     */
    public PdsVO getPds(Integer pds_no);
}
```

```

package kr.co.sample.board.service;
// import

public class BoardServiceImpl implements IBoardService{
    private static final Logger log = LoggerFactory.getLogger(this.getClass());
    @Inject
    private IBoardDao boardDAO;
    @Inject
    private IPdsDAO pdsDAO;

    @Value("#{appInfo.pdsSaveUrl}")
    private String saveUrl;
    private File saveFolder;
    @Inject
    private ResourceLoader loader;

    @Value("#{appInfo.LEVELLIMIT}")
    int LEVELLIMIT;
    @Value("#{appInfo.SAMELEVELCHARS}")
    int SAMELEVELCHARS;

    @PostConstruct
    public void init() throws IOException{
        saveFolder = loader.getResource(saveUrl).getFile();
        if (!saveFolder.exists()) saveFolder.mkdirs();
    }

    /**
     * 첨부파일의 메타데이터와 실제 파일바디 부분을 분리해 저장하기 위한 메소드.<br />
     * @param board
     * @throws IOException
     */
    private void savePdsMetaDataAndFileBody(BoardVO board)
    throws IOException{
        List<PdsVO> pdsList = board.getPdsList();
        if (pdsList != null && pdsList.size() > 0) {
            int rowcnt = pdsDAO.insertPds(pds);
            if (rowcnt == pdsList.size()) {
                for(PdsVO pds : pdsList){
                    Pds.saveToFolder(saveFolder);
                }
            } else {
                throw new BoardException("첨부파일 업로드 실패");
            } // inner if end
        }
    }
}

```

```

/**
 * 게시글 수정이나 삭제시 기존 첨부파일의 파일바디를 삭제하기 위한 메소드.
 * @param delPdsSaveNames
 * @throws IOException
 */
private void deletePdsFileBody(String[] delPdsSaveNames) throws IOException{
    if(delPdsSaveNames!=null){
        for (String saveName : delPdsSaveNames) {
            File delFile = new File(saveFolder, saveName);
            if (delFile.exists() && delFile.isFile()) {
                delFile.delete();
            } else{
                log.error("삭제 과정에서 해당 파일이 존재하지 않거나, 삭제 대상이 파일의
                형태로 존재하지 않습니다.");
            }
        }
    }
}

/**
 * 부모글이 존재하는지 답글의 깊이는 적절한지 확인하기 위한 메소드
 * @param board
 * @return
 */
private boolean validParent(BoardVO parent){
    if(parent==null){
        throw new BoardException("부모글이 존재하지 않습니다.");
    }else if(parent.getLevel()<0 || parent.getLevel() >= LEVELLIMIT){
        throw new BoardException(parent.getGroup_no()+"번 글에는 더 이상 답글 안대영");
    }else{
        return true;
    }
}

/**
 * 답글의 시퀀스 번호를 생성
 * @param parent
 * @return
 */
private String generateSequenceNo(BoardVO parent){
    int depth = LEVELLIMIT * SAMELEVELCHARS; // sequence_no 길이
    int parentLevel = parent.getLevel();
    long parentSeq = Long.parseLong(parent.getSequence_no());
    // 부모글의 최대레벨에서 부모글의 레벨을 뺀 10의 승수 값. ex) 부모글이 1레벨이면, 10000
    long levelDeterminant = (long) Math.pow(10, SAMELEVELCHARS * (LEVELLIMIT -
    parentLevel));
    // 답글의 레벨을 결정짓은 순차번호만 10의 배수로 계산. ex) 990000
    long minSeq = (parentSeq / levelDeterminant) * levelDeterminant;
    DecimalFormat formatter = new DecimalFormat(String.format("%0"+depth+"d", 0));
    formatter.setGroupingUsed(false);
    Map<String, Object> paramMap = new HashMap<String, Object>();
    paramMap.put("group_no", parent.getGroup_no());
    paramMap.put("maxSeq", parent.getSequence_no());
    paramMap.put("minSeq", formatter.format(minSeq));
}

```

```

        String minSeqSameLevel = boardDAO.selectMinSequenceNoWithSameGroup(paramMap);
        if(minSeqSameLevel==null){// 자식글이 아직 없는 경우, 부모글의 시퀀스 번호를 사용
            minSeqSameLevel = parent.getSequence_no();
        }
        // 자식글 레벨로 1레벨을 낮춰서 시퀀스 번호를 결정하는 승수.
        levelDeterminant = levelDeterminant / (long)Math.pow(10, SAMELEVELCHARS);
        long offset = levelDeterminant - 1; // 레벨 이후의 하위 레벨 자리의 숫자는 '9'로 넣기
        // 위해 9999, 혹은 99 를 읍셋값으로 사용함.
        long sequence_no = (Long.parseLong(minSeqSameLevel)/levelDeterminant-1) *
        levelDeterminant + offset;
        if(sequence_no < minSeq){
            throw new BoardException("동일 레벨의 허용 댓글 수 초과. 해당 레벨의 마지막
            자식글이 있는 경우임.");
        }
        return formatter.format(sequence_no);
    }
    /**
     * 댓글 등록전 답글에 대한 기본 검증과 시퀀스 번호 생성을 위한 기본 설정
     * @param board
     */
    private void settingReplyBoard(BoardVO board) {
        BoardVO parent = getBoard(board.getBo_parent());
        if(validParent(parent)){
            board.setGroup_no(parent.getGroup_no());
            board.setSequence_no(generateSequenceNo(parent));
        }
    }
    @Override
    public CommonResult createBoard(BoardVO board) throws IOException{
        if(board.getBo_parent()!=null){
            settingReplyBoard(board);
        }
        Integer pk = boardDAO.insertBoard(board);
        if (pk == null || pk <= 0) {
            return CommonResult.FAIL;
        } else {
            savePdsMetaDataAndFileBody(board);
            return CommonResult.OK;
        }
    }

    @Override
    public BoardVO getBoard(Integer bo_no) {
        BoardVO board = boardDAO.selectBoard(bo_no);
        if (board == null) {
            throw new RuntimeException("게시글이 존재하지 않음");
        } else {
            boardDAO.incrementHit(bo_no);
            return board;
        }
    }
}

```

```
@Override
public List<BoardVO> getBoardList(PagingInfoVO<BoardVO> pagingInfoVO) {
    return boardDAO.selectBoardList(pagingInfoVO);
}

@Override
public int getBoardCount(PagingInfoVO<BoardVO> pagingInfoVO) {
    return boardDAO.selectBoardCount(pagingInfoVO);
}

@Override
public CommonResult modifyBoard(BoardVO board) throws IOException {
    Integer bo_no = board.getBo_no();
    BoardVO savedBoard = getBoard(bo_no);
    String savedPasswd = savedBoard.getBo_passwd();
    String inputPasswd = board.getBo_passwd();
    if (savedPasswd.equals(inputPasswd)) {
        int result = boardDAO.updateBoard(board);
        if (result > 0) {
            Integer[] delPdsNos = board.getDelPdsNos();
            String[] delPdsSaveNames = null;
            if (delPdsNos != null && delPdsNos.length > 0) {
                delPdsSaveNames = new String[delPdsNos.length];
                for (int i=0; i<delPdsNos.length; i++) {
                    PdsVO delPds = pdsDAO.selectPds(delPdsNos[i]);
                    delPdsSaveNames[i]=delPds.getPds_savename();
                } // for end
                pdsDAO.deletePdses(board);
                deletePdsFileBody(delPdsSaveNames);
            }
            savePdsMetaDataAndFileBody(board);
            return CommonResult.OK;
        } else {
            return CommonResult.FAIL;
        } // if(result > 0) end
    } else {
        return CommonResult.INVALIDPASSWORD;
    }
}
// continue
```



```

@Override
public CommonResult removeBoard(BoardVO board) throws IOException {
    Integer bo_no = board.getBo_no();
    BoardVO savedBoard = getBoard(bo_no);
    String savedPasswd = savedBoard.getBo_passwd();
    String inputPasswd = board.getBo_passwd();
    if (savedPasswd.equals(inputPasswd)) {
        List<PdsVO> pdsList = savedBoard.getPdsList();
        String[] delPdsSaveNames = null;
        // PDS 테이블이 가진 cascade 옵션에 따라 BOARD 테이블에서 레코드가 삭제될시
        // PDS 의 자식 레코드가 동시에 삭제됨.
        // 이 경우, ORM 의 lazyLoadingEnabled 속성이 활성화되어있을시
        // BOARD 레코드가 지워지고 CASCADE 로 PDS 레코드가 지워진후
        // 이미 삭제된 레코드를 조회할 가능성이 있음.
        // 따라서 파일의 위치를 미리 캐싱해뒀다가 삭제할 수 있음.
        if (pdsList != null) {
            delPdsSaveNames = new String[pdsList.size()];
            for (int idx = 0; idx < pdsList.size(); idx++) {
                delPdsSaveNames[idx] = pdsList.get(idx).getPds_savename();
            }
        }
        int result = boardDAO.deleteBoard(bo_no);
        if (result > 0) {
            if (delPdsSaveNames.length > 0) deletePdsFileBody(delPdsSaveNames);
            return CommonResult.OK;
        } else {
            return CommonResult.FAIL;
        }
    } else {
        return CommonResult.INVALIDPASSWORD;
    }
}

@Override
public PdsVO getPds(Integer pds_no) {
    PdsVO pds = pdsDAO.selectPds(pds_no);
    if (pds == null) {
        throw new RuntimeException("해당 첨부파일이 존재하지 않음.");
    } else {
        pdsDAO.incrementDowncount(pds_no);
        return pds;
    }
}
}

```

- List<PdsVO>, PdsVO[], PdsVO ;타입의 핸들러 메소드 아규먼트 처리를 위한 컴포넌트.

```
package kr.co.sample.web;
// import
/**
 * PdsVO, PdsVO[], List<PdsVO> 형태의 핸들러 메소드 아규먼트를 처리.
 *
 */
public class PdsVOMethodArgumentResolver implements HandlerMethodArgumentResolver {
    private static enum ArgType{ VO, ARRAY, LIST, NONE}
    private ArgType argType = ArgType.NONE;

    @Override
    public boolean supportsParameter(MethodParameter parameter) {
        Class<?> paramType = parameter.getParameterType();
        boolean support = false;
        if(PdsVO.class.isAssignableFrom(paramType) ){
            this.argType = ArgType.VO;
            support = true;
        }else if (List.class.isAssignableFrom(paramType)){
            Class<?> elementType =
                GenericCollectionTypeResolver.getCollectionParameterType(parameter);
            if (elementType != null && elementType.equals(PdsVO.class)) {
                this.argType = ArgType.LIST;
                support = true;
            }
        }else if(paramType.isArray()){
            if(PdsVO.class.isAssignableFrom(paramType.getComponentType())){
                this.argType = ArgType.ARRAY;
                support = true;
            }
        }
        return support;
    }

    private void assertIsMultipartRequest(HttpServletRequest request) {
        String contentType = request.getContentType();
        if(contentType == null ||
            !contentType.toLowerCase().startsWith("multipart/")){
            throw new MultipartException("현재 리퀘스트는 multipart 리퀘스트가 아님. form
            enctype 을 확인할것.");
        }
    }
}
```

```

@Override
public Object resolveArgument(MethodParameter parameter,
ModelAndViewContainer mavContainer, NativeWebRequest webRequest,
WebDataBinderFactory binderFactory) throws Exception {
    HttpServletRequest request = webRequest.getNativeRequest(HttpServletRequest.class);
    assertIsMultipartRequest(request);
    MultipartHttpServletRequest multipartRequest = WebUtils.getNativeRequest(request,
MultipartHttpServletRequest.class);
    String partName = parameter.getParameterName();
    List<MultipartFile> files = multipartRequest.getFiles(partName);
    if(files==null){
        return null;
    }else{
        List<PdsVO> pdsList = new ArrayList<PdsVO>();
        for(MultipartFile file : files){
            if(file.isEmpty()) continue;
            pdsList.add(new PdsVO(file));

        } // for end
        if(pdsList.size()>0){
            return makeArgument(parameter, pdsList);
        }else{
            return null;
        }
    } // else end
}

private Object makeArgument(MethodParameter parameter, List<PdsVO> pdsList){
    Object arg = null;
    if(pdsList!=null && pdsList.size()>0){
        switch(this.argType){
            case VO :
                arg = pdsList.get(0);
                break;
            case ARRAY :
                PdsVO[] array = new PdsVO[pdsList.size()];
                arg = pdsList.toArray(array);
                break;
            case LIST :
                arg = pdsList;
                break;
            default :
                arg = null;
        }
    }
    return arg;
}
}

```

- 게시글 검증을 위한 Validator 작성

```
package kr.co.sample.web.validator;

// import 구문
/**
 * 게시글에 대한 검증을 수행하는 {@link Validator}
 */
public class BoardValidator implements Validator{
    @Inject
    MessageSourceAccessor accessor;

    @Override
    public boolean supports(Class<?> arg0) {
        return BoardVO.class.isAssignableFrom(arg0);
    }

    @Override
    public void validate(Object commandObject, Errors bindingResult) {
        BoardVO board = (BoardVO) commandObject;
        //검증 결과 객체 속성 결정 방법.
        //1. Errors 를 직접 사용하는 경우
        //    : rejectValue(필드명, 에러코드, 에러메시지 인자, 디폴트 메시지)
        //2. ValidationUtils 를 사용하는 경우
        //    : ValidationUtils.rejectIfEmpty(Errors객체, 필드명, 에러코드,
        //    에러메시지 인자, 디폴트 메시지);
        //
        if(StringUtils.isBlank(board.getBo_title())){
            bindingResult.rejectValue("bo_title", "NotEmpty",
                new Object[]{accessor.getMessage("board.bo_title")},
                "제목은 필수입력사항");
        }

        ValidationUtils.rejectIfEmpty(bindingResult, "bo_writer",
            "NotEmpty", new Object[]{accessor.getMessage("board.bo_writer")},
            "작성자 필수 입력사항");

        ValidationUtils.rejectIfEmpty(bindingResult, "bo_passwd",
            "NotEmpty", new Object[]{accessor.getMessage("board.bo_passwd")},
            "비번 필수 입력사항");

        String bo_passwd = board.getBo_passwd();

        if(bo_passwd!=null &&
            (bo_passwd.length()<4 || bo_passwd.length()>12)){
            bindingResult.rejectValue("bo_passwd", "Length",
                new Object[]{accessor.getMessage("board.bo_passwd"), 4, 12},
                "비번의 길이는 4~12");
        }

        ValidationUtils.rejectIfEmpty(bindingResult, "bo_content", "NotEmpty",
            new Object[]{accessor.getMessage("board.bo_content")},
            "내용 필수");
    }
}
```

- 게시판 핸들러에 공통으로 위빙할 advice

```
package kr.co.sample.board.controller;

import javax.inject.Inject;

import kr.co.sample.web.validator.BoardValidator;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;

/**
 * 게시판 공통 예외 처리 컴포넌트<br />
 * 스프링 3.2 부터 추가된 {@link ControllerAdvice}를 통해<br />
 * 동일한 Component-scan 에 의해 등록된 컨트롤러들에 공통적으로 위빙 가능한 advice를 지정할수있음. <br />
 * 이 advice 를 통해 공통 예외처리(@{@link ExceptionHandler})나 <br />
 * 공통 모델(@{@link ModelAttribute}) <br />
 * 혹은 공통 바인더(@{@link InitBinder}) 설정이 가능함.
 *
 */
@ControllerAdvice
public class BoardControllerAdvice {

    private static Logger log = LoggerFactory.getLogger(BoardControllerAdvice.class);

    @Inject
    BoardValidator validator;

    /**
     * "board" 혹은 "boardVO" 라는 모델명을 가진 command object 에 대해
     * 공통 검증을 수행하기 위해 바인더에 validator({@link BoardValidator}) 를 설정하기 위한 메소드
     * @param binder
     */
    @InitBinder({"board", "boardVO"})
    public void bindingInitializer(WebDataBinder binder) {
        binder.setValidator(validator);
    }

    /**
     * 게시판에 대해 특정 예외가 발생시 공통 처리하기 위한 핸들러 메소드
     * @param e
     * @param response
     * @return
     */
    @ExceptionHandler({Exception.class })
    public String exceptionHandler(Throwable e, HttpServletResponse response){
        response.reset();
        log.error("게시판 예외 발생", e);
        return "error/exceptionView";
    }
}
```

- 게시글과 첨부파일 처리를 위한 Controller layer 작성

```
package kr.co.sample.board.controller;
//import 구문
/**
 * 게시글 목록조회, 상세조회, 게시글 수정을 위한 controller
 */
@Controller
@RequestMapping("/board")
public class BoardController1 {
    @Inject
    private IBoardService service;
    /**
     * @param searchType 검색 조건
     * @param searchWord 검색 키워드
     * @param currentPage 요청 페이지
     * @param model
     * @return 논리적인 view name
     */
    @GetMapping("/boardList.do")
    public String boardList(
        @RequestParam(required = false) String searchType,
        @RequestParam(required = false) String searchWord,
        @RequestParam(required = false, defaultValue = "1") int currentPage,
        Model model)
    {
        BoardVO board = new BoardVO();
        if ("bo_title".equals(searchType)) {
            board.setBo_title(searchWord);
        } else if ("bo_writer".equals(searchType)) {
            board.setBo_writer(searchWord);
        } else if ("bo_content".equals(searchType)) {
            board.setBo_content(searchWord);
        } else {
            board.setBo_content(searchWord);
            board.setBo_writer(searchWord);
            board.setBo_title(searchWord);
        }

        PagingInfoVO<BoardVO> pagingInfoVO = new PagingInfoVO<BoardVO>();
        pagingInfoVO.setData(board);

        pagingInfoVO.setCurrentPage(currentPage);
        int totalRecords = service.getBoardCount(pagingInfoVO);
        pagingInfoVO.setTotalRecords(totalRecords);

        List<BoardVO> boardList = service.getBoardList(pagingInfoVO);
        pagingInfoVO.setDataList(boardList);

        model.addAttribute("pagingInfoVO", pagingInfoVO);
        return "board/boardList";
    }
}
```

```

/**
 * produces : Accept 리퀘스트 헤더에 대한 매핑 설정.
 * @param searchType 검색 조건
 * @param searchWord 검색 키워드
 * @param currentPage 요청 페이지
 * @param model
 * @return @ResponseBody 에 의해 마샬링 대상이 되는 객체
 */
@GetMapping(value = "/boardList.do", produces = MediaType.APPLICATION_JSON_VALUE)
@ResponseBody
public PagingInfoVO<BoardVO> boardList(
    @RequestParam(required = false) String searchType,
    @RequestParam(required = false) String searchWord,
    @RequestParam(required = false, defaultValue = "1") int currentPage,
    Model model)
{
    boardList(searchType, searchWord, currentPage, model);
    return (PagingInfoVO<BoardVO>) model.asMap().get("pagingInfoVO");
}

/**
 * 경로변수(@{@link PathVariable})를 사용한 게시글 상세 조회 핸들러 메소드
 * @param bo_no 경로변수로 전달된 조회할 글번호, 경로변수는 requestScope 를 통해 뷰로 전달.
 * @param model
 * @return
 */
@GetMapping("/boardView/{bo_no}")
public String boardView(@PathVariable("bo_no") int bo_no, Model model) {
    BoardVO board = service.getBoard(bo_no);
    model.addAttribute("board", board);
    return "board/boardView";
}

/**
 * 게시글 수정 양식 제공을 위한 핸들러 메소드, 동일 URI에 대해 Method 로 리퀘스트 매핑 조건을 설정.
 * RESTful URI 방식은 아니나, 관행적으로 많이 사용되던 URI 형식을 사용하고 있음.
 * 모든 커맨드에 대해 RESTful 구조를 적용한다면, 이와 같이 불필요한 핸들러 메소드를 중복이 해결됨.
 * @param bo_no 수정할 글번호
 * @param model
 * @return
 */
@GetMapping("/boardUpdate.do")
public String boardUpdate(int bo_no, Model model) {
    boardView(bo_no, model);
    return "board/boardEdit";
}

// continue

```

```

/**
 * 게시글 수정데이터를 처리하기 위한 핸들러 메소드.
 * @param board 수정할 데이터를 가진 command object
 * @param bo_files {@link PdsVOMethodArgumentResolver}를 통해 multipart 로부터 추출된 파일에 대한
   정보를 가진 객체
 * @param model
 * @param status 커맨드가 정상 종료되는 시점에 세션 스코프에 있는 속성을 정리하기 위한 {@link
   SessionStatus}
 * @return
 * @throws IOException
 */
@PostMapping(value = "/boardUpdate.do", Consumes=MediaType.MULTIPART_FORM_DATA_VALUE)
public String boardUpdate(
    @Valid @ModelAttribute("board") BoardVO board, // @Validated 로 group hint 설정 가능
    Errors errors,
    ArrayList<PdsVO> bo_files,
    Model model) throws IOException {
    if(errors.hasErrors()){
        return "board/boardEdit";
    }
    String viewName = null;
    String message = null;

    if(bo_files!=null){
        List<PdsVO> pdsList = board.getPdsList();
        if(pdsList!=null){
            pdsList.addAll(bo_files);
        }else{
            board.setPdsList(bo_files);
        }
    }

    CommonResult result = service.modifyBoard(board);
    if (result == CommonResult.INVALIDPASSWORD) {
        viewName = "board/boardEdit";
        message = "비밀번호 오류, 다시 입력하셈.";
    } else if (result == CommonResult.FAIL) {
        viewName = "board/boardEdit";
        message = "서버오류. 다시 시도 하셈.";
    } else {
        viewName = "redirect:/board/boardView/" + board.getBo_no();
    }
    model.addAttribute("message", message);
    return viewName
}
}

```



```

package kr.co.sample.board.controller;

// import 구문
/**
 * 게시글 등록 및 삭제를 위한 controller
 *
 */
@Controller
@RequestMapping("/board")
public class BoardController2 {
    @Inject
    IBoardService service;
    @Value("#{appInfo.LEVELLIMIT}")
    int LEVELLIMIT;
    @Value("#{appInfo.SAMELEVELCHARS}")
    int SAMELEVELCHARS;
    /**
     * 게시글 삭제를 위한 핸들러 메소드
     * @param bo_no 삭제할 글 번호
     * @param bo_passwd 인증용 비밀번호
     * @param redirectAttributes FlashAttribute 를 사용하기 위해 주입받는 객체
     * @return
     * @throws IOException
     */
    @RequestMapping("/boardDelete.do")
    public String boardDelete(String bo_no, String bo_passwd,
        RedirectAttributes redirectAttributes) throws IOException{
        BoardVO board = new BoardVO();
        board.setBo_no(new Integer(bo_no));
        board.setBo_passwd(bo_passwd);
        CommonResult result = service.removeBoard(board);
        String viewName = null;
        String message = null;
        if (result == CommonResult.OK) {
            viewName = "redirect:/board/boardList.do";
        } else if (result == CommonResult.FAIL) {
            message = "서버 오류 다시 지우셈.";
            viewName = "redirect:/board/boardView/" + bo_no;
        } else if (result == CommonResult.INVALIDPASSWORD) {
            message = "비밀번호 오류 다시 지우셈.";
            viewName = "redirect:/board/boardView/" + bo_no;
        }
        redirectAttributes.addFlashAttribute("message", message);
        return viewName;
    }
    /**
     * 게시글 작성을 위한 양식을 제공하기 위한 핸들러 메소드
     * @param board command object
     * @return
     */
    @GetMapping("/boardInsert.do")
    public String boardInsert(
        @ModelAttribute("board") BoardVO board, RedirectAttributes redirectAttributes){
        String viewName = "board/boardForm";
        Integer bo_parent = board.getBo_parent();
    }
}

```

```

        if(bo_parent!=null && bo_parent > 0){ // 답글인 경우, validation
            BoardVO parent = boardDAO.selectBoard(bo_parent);
            if(parent==null){
                redirectAttributes.addFlashAttribute("msg", "부모글이 존재하지 않습니다.
원글이 그 사이 서버에서 삭제된 것 같아요~");
                viewName = "redirect:/board/boardList.do";
            }else if(parent.getLevel() >= LEVELLIMIT){
                redirectAttributes.addFlashAttribute("msg",
                String.format("특정글에 대해 허용하는 답글의 깊이는 %d입니다. 그이상은
시저염~", BoardVO.LEVELLIMIT));
                viewName = "redirect:/board/boardList.do";
            }
        }
        return viewName;
    }
}

/**
 * 양식을 통해 작성된 게시글을 등록하기 위한 핸들러 메소드
 * @param board 등록할 글 정보를 가진 command object
 * @param bindingResult 바인더 설정에 따라 검증 결과를 가진 객체({@link
BoardControllerAdvice#bindingInitializer})
 * @param bo_files {@link PdsVOMethodArgumentResolver}를 통해 multipart 로부터 추출된 파일에 대한
정보를 가진 객체
 * @param model
 * @return
 * @throws IOException
 */
@PostMapping("/boardInsert.do")
public String boardInsert(
    @Valid @ModelAttribute("board") BoardVO board,
    BindingResult bindingResult,
    ArrayList<PdsVO> bo_files,
    Model model) throws IOException {
    if (bindingResult.hasErrors()) {
        return "board/boardForm";
    }
    if(board.getBo_parent()!=null){
        board.setBo_title("[RE] "+board.getBo_title());
    }
    String message = null;
    String viewName = null;
    board.setPdsList(bo_files);
    CommonResult result = service.createBoard(board);
    if (result == CommonResult.OK) {
        viewName = "redirect:/board/boardList.do";
    } else {
        viewName = "board/boardForm";
        message = "서버 오류, 잠시뒤 다시 시도해주셈.";
        model.addAttribute("message", message);
    }
    return viewName;
}
}

```

```

package kr.co.sample.board.controller;
// import 구문
/**
 * 첨부파일 다운로드는 각기 다른 방식으로 처리할수 있음.<br />
 * 1. MessageConverter를 사용하는 방식(download1)
 * 2. BeanNameViewResolver 로 custom view
 * 를 사용하는 방식(download2)<br />
 * 두 메소드의 "@RequestMapping" 메소드를 주석처리 해가며 테스트 할것.
 */
@Controller
public class DownloadController {
    @Inject
    private IBoardService service;
    @Value("#{appInfo.pdsSaveUrl}")
    private String saveUrl;
    @Inject
    private ResourceLoader loader;
    /**
     * {@link ResourceHttpMessageConverter} 동작
     * @param pds_no
     * @param response
     * @return
     * @throws IOException
     */
    @RequestMapping("/board/download.do")
    @ResponseBody
    public Resource download1(
        @RequestHeader(name="user-agent", required=true) String agent,
        @RequestParam(required=true) int pds_no,
        HttpServletResponse response) throws UnsupportedEncodingException {
        response.setContentType("application/octet-stream");

        PdsVO pds = service.getPds(pds_no);
        String filename = pds.getPds_filename();

        if(StringUtils.containsAny(agent.toLowerCase(), "trident", "msie", "edge")) {
            filename = URLEncoder.encode(filename, "UTF-8");
            filename = filename.replace("+", "%20");
        } else {
            filename = new String(filename.getBytes(), "iso-8859-1");
        }

        response.setHeader("Content-Disposition", "attachment;filename=\""
            + filename + "\"");
        long filesize = pds.getPds_filesize();
        response.setHeader("Content-Length", filesize + "");
        Resource resource = this.loader.getResource(saveUrl+"/"+pds.getPds_savename());
        if(!resource.exists()){
            throw new FileNotFoundException(pds.getPds_no()+" , 해당 파일이 존재하지
            않습니다. 서버 오류. ");
        }
        return resource;
    }
}

```

```

/**
 * 논리적인 view name 을 이용한 MVC 구조식
 * @param pds_no
 * @param model
 * @return
 */
public String download2(int pds_no, Model model) {
    PdsVO pds = service.getPds(pds_no);
    model.addAttribute("pds", pds);
    return "downloadView";
}
}

```

- 파일 다운로드를 위한 Custom View, download2 메소드에서 사용할 View 객체

```

package kr.co.sample.web.view;

/**
 * {@link BeanNameViewResolver} 를 통해 뷰를 선택하고 있는 다운로드 컨트롤러를 위한 뷰
 * @see {@link AbstractView#renderMergedOutputModel}
 */
public class DownloadView extends AbstractView {
    @Value("#{appInfo.pdsSaveUrl}")
    private String saveUrl;
    @Inject
    private ResourceLoader loader;
    @Override
    protected void renderMergedOutputModel(Map<String, Object> model,
        HttpServletRequest request, HttpServletResponse response)
        throws Exception {
        PdsVO pds = (PdsVO) model.get("pds");
        response.setContentType("application/octet-stream");
        String filename = pds.getPds_filename();
        if(StringUtils.containsAny(agent.toLowerCase(), "trident", "msie", "edge")) {
            filename = URLEncoder.encode(filename, "UTF-8");
            filename = filename.replace("+", "%20"); // 공백 처리
        } else {
            filename = new String(filename.getBytes(), "iso-8859-1");
        }
        response.setHeader("Content-Disposition", "attachment;filename=\""
            + filename + "\"");
        long filesize = pds.getPds_filesize();
        response.setHeader("Content-Length", filesize + "");

        Resource file = this.loader.getResource(saveUrl+"/"+pds.getPds_savename());
        if(!file.exists()){
            throw new FileNotFoundException(pds.getPds_no()+" 에 해당하는 파일이
            존재하지 않습니다.");
        }
        try(OutputStream os = response.getOutputStream()){
            FileUtils.copyFile(file.getFile(), os);
            os.flush();
        } catch (Exception e) {
            throw e;
        }
    }
}
}

```

- TilesViewResolver 를 위한 definition 정의 파일 작성(kr/co/sample/tiles/defs.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 2.1//EN"
    "http://tiles.apache.org/dtds/tiles-config_2_1.dtd">
<tiles-definitions>
    <definition name="parent" template="/WEB-INF/views/template.jsp">
        <put-attribute name="title" value="Spring Board Tutorial" />
        <put-attribute name="scriptTags" value="/WEB-INF/views/frags/scripts.jsp" />
        <put-attribute name="styleTags" value="/WEB-INF/views/frags/styles.jsp" />
    </definition>

    <definition extends="parent" name="board/*">
        <put-attribute name="content" value="/WEB-INF/views/board/{1}.jsp" />
    </definition>

    <definition extends="parent" name="matrix/*">
        <put-attribute name="content" value="/WEB-INF/views/matrix/{1}.jsp" />
    </definition>

    <definition extends="parent" name="error/*">
        <put-attribute name="content" value="/WEB-INF/views/error/{1}.jsp" />
    </definition>
</tiles-definitions>
```

- Presentation layer 를 구성하는 view(jsp) 작성
 - Tiles definition 을 구성하는 template 및 attribute 작성.
 - /WEB-INF/views/template.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title><tiles:getAsString name="title"/></title>
        <tiles:insertAttribute name="scriptTags" />
        <tiles:insertAttribute name="styleTags" />
    </head>
    <body>
        <tiles:insertAttribute name="content" />
    </body>
</html>
```

- /WEB-INF/views/frags/scripts.jsp (공통 스크립트 로딩 attribute)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<script type="text/javascript" src="https://code.jquery.com/jquery-git1.min.js"></script>
```

- /WEB-INF/views/frags/styles.jsp (공통 스타일 로딩 attribute)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<link rel="stylesheet" href="${pageContext.request.contextPath }/resources/css/main.css" />
```

- /WEB-INF/views/board/boardForm.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<h4>새글 등록 양식</h4>
<div class="form">
    <form:form action="${pageContext.request.contextPath }/board/boardInsert.do"
        method="post" enctype="multipart/form-data" modelAttribute="board">
        <input type="hidden" name="bo_parent" value="${board.bo_parent}" />
        <table>
            <tr>
                <th>제목</th>
                <td>
                    <input type="text" name="bo_title"
                        value="${board.bo_title}" />
                    <form:errors path="bo_title" element="span"
                        cssClass="errors"></form:errors>
                </td>
            </tr>
            <tr>
                <th>글쓴이</th>
                <td>
                    <input type="text" name="bo_writer"
                        value="${board.bo_writer}" />
                    <form:errors path="bo_writer" element="span"
                        cssClass="errors"></form:errors>
                </td>
            </tr>
            <tr>
                <th>비번</th>
                <td>
                    <input type="text" name="bo_passwd"
                        value="${board.bo_passwd}" />
                    <form:errors path="bo_passwd" element="span" cssClass="errors"
                        delimiter="&nbsp;"></form:errors>
                </td>
            </tr>
            <tr>
                <th>이메일</th>
                <td>
                    <input type="email" name="bo_email"
                        value="${board.bo_email}" />
                </td>
            </tr>
            <tr>
                <th>글쓴이 IP</th>
                <td>
                    <input type="text" name="bo_ip" readonly="readonly"
                        value="${pageContext.request.remoteAddr}" />
                </td>
            </tr>
            <tr>
                <th>첨부파일</th>
                <td>
                    <input type="file" name="bo_files" />
                    <input type="file" name="bo_files" />
                    <input type="file" name="bo_files" />
                </td>
            </tr>
        </table>
    </form:form>
</div>
```

```

        <tr>
            <th>내용</th>
            <td><textarea name="bo_content"
id="ir1">${board.bo_content }</textarea>
            <form:errors path="bo_content" element="span"
cssClass="errors"></form:errors>
            </td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" value="등록" /></td>
        </tr>
    </table>
</form:form>
</div>

```

- /WEB-INF/views/board/boardList.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://www.springframework.org/tags" prefix="spring" %>
<script type="text/javascript">
    function ${pagingInfoVO.jsFuncName} (currentPage){
        var form = document.getElementById("searchForm");
        form.searchType.value="${param.searchType}";
        form.searchWord.value="${param.searchWord}";
        form.currentPage.value = currentPage;
        form.submit();
    }

    <c:set var="message" value="${empty param.message? message : param.message}" />
    <c:if test="${!empty message }">
        alert("${message }");
    </c:if>
</script>
<h4>게시글 목록 조회</h4>
<input type="button" value="새글쓰기"
onclick="window.Location.href= '${pageContext.request.contextPath}/board/boardInsert.do'"
/>

<div class="list">
    <table>
        <thead>
            <tr>
                <th><spring:message code="board.bo_no"/></th>
                <th><spring:message code="board.bo_title"/></th>
                <th><spring:message code="board.bo_writer"/></th>
                <th><spring:message code="board.bo_date"/></th>
                <th><spring:message code="board.bo_hit"/></th>
            </tr>
        </thead>

```

- [illegible]

- /WEB-INF/views/board/boardView.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<script type="text/javascript">
    function handler(command){
        var form = document.boardHiddenForm;
        if(command == 'list'){
            window.location.href="${pageContext.request.contextPath}/board/boardList.do";
        }else if(command == 'update'){
            form.action = "${pageContext.request.contextPath}/board/boardUpdate.do";
            form.submit();
        }else if(command == 'delete'){
            form.action = "${pageContext.request.contextPath}/board/boardDelete.do";
            var bo_passwd = prompt("비밀번호 입력하셈");
            form.bo_passwd.value=bo_passwd;
            form.submit();
        }else if(command == 'reply'){
            form.action = "${pageContext.request.contextPath}/board/boardInsert.do";
            form.submit();
        }
    }
</script>
<c:set var="message" value="${empty param.message? message : param.message}" />
<c:if test="${!empty message}">
    alert("${message}");
</c:if>
</script>
<form name="boardHiddenForm">
    <input type="hidden" name="bo_no" value="${board.bo_no}" />
    <input type="hidden" name="bo_passwd" />
    <input type="hidden" name="bo_parent" value="${board.bo_no }">
</form>
<div class="view">
    <table>
        <tr>
            <th>글번호 (PK)</th>
            <td>${board.bo_no}</td>
        </tr>
        <tr>
            <th>제목</th>
            <td>${board.bo_title}</td>
        </tr>
        <tr>
            <th>글쓴이</th>
            <td>${board.bo_writer}</td>
        </tr>
        <tr>
            <th>비번</th>
            <td>테스트용: ${board.bo_passwd}</td>
        </tr>
        <tr>
            <th>이메일</th>
            <td>${board.bo_email}</td>
        </tr>
    </table>
</div>
```

- /WEB-INF/views/board/boardViewjsp 계속

```

        <tr>
            <th>글쓴이 IP</th>
            <td>${board.bo_ip}</td>
        </tr>
        <tr>
            <th>첨부파일</th>
            <td>
                <c:forEach items="${board.pdsList }" var="pds"
                    varStatus="vs">
                    <c:url value="/board/download.do" var="downUrl">
                        <c:param name="pds_no" value="${pds.pds_no }" />
                    </c:url>
                    <a href="${downUrl }"
                        title="다운횟수:${pds.pds_downcount },
                        ${pds.pds_fancysize}">${pds.pds_filename }</a>
                    <c:if test="${!vs.Last }">,</c:if>
                </c:forEach>
            </td>
        </tr>
        <tr>
            <th>내용</th>
            <td>${board.bo_content}</td>
        </tr>
        <tr>
            <th>조회수</th>
            <td>${board.bo_hit}</td>
        </tr>
        <tr>
            <th>최초 등록일</th>
            <td>${board.bo_date}</td>
        </tr>
        <tr>
            <td colspan="2">
                <spring:eval expression="@appInfo" var="info"></spring:eval>
                <td colspan="2">
                    <c:if test="${board.level lt info.LEVELLIMIT }">
                        <input type="button" value="답글쓰기" onclick=" handler('reply') " />
                    </c:if>
                    <input type="button" value="수정" onclick="handler('update') " />
                    <input type="button" value="삭제" onclick=" handler('delete') " />
                    <input type="button" value="목록으로" onclick=" handler('list') " />
                </td>
            </tr>
    </table>
</div>

```

- /WEB-INF/views/board/boardEdit.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<script type="text/javascript">
    <c:set var="message" value="${empty param.message? message : param.message}" />
    <c:if test="${!empty message }">
        alert("${message }");
    </c:if>

    $(document).ready(function(){
        $(".delBtn").on("click", function(event){
            var span = $(this).parent();
            $(span).hide();
            var pds_no = $(this).attr("id");
            var html =
                "<input type='hidden' name='delPdsNos' value='"+pds_no+"' />";
            $("#boardUpdateForm").append(html);
            event.preventDefault();
        });
    });
</script>
<h4>수정 양식</h4>
<div class="form">
    <form action="${pageContext.request.contextPath }/board/boardUpdate.do"
        method="post"
        enctype="multipart/form-data"
        id="boardUpdateForm"
    >
        <input type="hidden" name="bo_no" value="${board.bo_no }"/>
        <table>
            <tr>
                <th>제목</th>
                <td>
                    <input type="text" name="bo_title"
                        value="${board.bo_title}" />
                </td>
            </tr>
            <tr>
                <th>글쓴이</th>
                <td>
                    <input type="text" name="bo_writer"
                        value="${board.bo_writer}" />
                </td>
            </tr>
            <tr>
                <th>비번(테스트용비번:${board.bo_passwd})</th>
                <td>
                    <input type="text" name="bo_passwd"
                        value="" />
                </td>
            </tr>
        </table>
    </form>
</div>

```

- /WEB-INF/views/board/boardEditjsp 계속

```

        <tr>
            <th>이메일</th>
            <td>
                <input type="email" name="bo_email"
                value="${board.bo_email}" />
            </td>
        </tr>
        <tr>
            <th>글쓴이 IP</th>
            <td>
                <input type="text" name="bo_ip" readonly="readonly"
                value="${pageContext.request.remoteAddr}" />
            </td>
        </tr>
        <tr>
            <th>기존첨부파일</th>
            <td>
                <c:forEach items="${board.pdsList}" var="pds">
                    <span>${pds.pds_filename }
                    <input type="image"
                    src="${pageContext.request.contextPath }/images/del.jpg"
                    id="${pds.pds_no }"
                    class="delBtn"
                    style="width: 15px; height: 15px;"
                    />
                    &nbsp;
                </span>
                </c:forEach>
            </td>
        </tr>
        <tr>
            <th>신규파일</th>
            <td>
                <input type="file" name="bo_files"/>
                <input type="file" name="bo_files"/>
                <input type="file" name="bo_files"/>
            </td>
        </tr>
        <tr>
            <th>내용</th>
            <td>
                <textarea name="bo_content" >${board.bo_content }</textarea>
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <input type="submit" value="수정하기" />
            </td>
        </tr>
    </table>
</form>
</div>

```

- 게시판 공통 예외 뷰(/WEB-INF/views/error/exceptionView.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isErrorPage="true"%>
<div class="view">
    <h4>게시판 공통 예외 처리 뷰</h4>
    <table>
        <thead>
            <tr>
                <th>상태코드</th>
                <th>발생위치</th>
                <th>예외클래스</th>
                <th>예외메시지</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>${pageContext.errorData.statusCode }</td>
                <td>${pageContext.errorData.requestURI }</td>
                <td>${pageContext.exception.getClass().getSimpleName() }</td>
                <td>${pageContext.exception.message }</td>
            </tr>
        </tbody>
    </table>
</div>
```

- Business logic 에 대한 아규먼트와 리턴값을 로깅하기 위한 advice

```
package kr.co.sample.advice;

// import 구문

@Aspect
public class LoggingAdvice {
    Logger log = LoggerFactory.getLogger(this.getClass());
    @Pointcut("execution(* kr.co.sample.*.service.*(..))")
    public void dummy(){}

    @Around(value="dummy()")
    public Object around(ProceedingJoinPoint joinpoint)
    throws Throwable{
        String targetName = joinpoint.getTarget().getClass().getSimpleName();
        String methodName = joinpoint.getSignature().getName();
        Object[] args = joinpoint.getArgs();
        log.info("{}.{ } 호출", targetName, methodName);
        log.info("넘어간 인자 : {}", Arrays.toString(args));
        Object retValue = joinpoint.proceed(args);
        log.info("결과값 : {}", retValue.toString());
        return retValue;
    }
}
```

□ 스프링 3.2 부터 추가된 MatrixVariable 의 활용

- (참고 : annotation-driven 에 enableMatrixVariables="true" 설정을 하거나 HandlerMapping 에 대해 removeSemicolonContent="false" 설정이 필요함.)
- 컨트롤러 추가.

```
package kr.co.sample.matrix.controller;
// import
/**
 * 스프링 3.2 에서 추가된 {@link MatrixVariable}의 기능을 테스트하기 위한 컨트롤러.<br />
 * '{@link MatrixVariable}'는 URI 포함된 matrix variable 를 확인할수 있는 어노테이션.<br />
 * 매트릭스 데이터가 포함된 URI 형태는 "/common/cars;colors=green,red,yellow;year=2014"와 같으며, <br />
 * 이때 cars 라는 인자는 경로변수의 형태로 {@link PathVariable}를 통해 받을수 있고, <br />
 * colors 와 year 라는 매트릭스 변수는 {@link MatrixVariable}를 통해 받을수 있다.
 */
@Controller
@RequestMapping("/matrix")
public class MatrixVariabelTestController {

    private Logger log = LoggerFactory.getLogger(this.getClass());
    @RequestMapping
    public String matrixForm(){
        return "matrix/matrixForm";
    }

    @RequestMapping(value="/{product}", produces={"text/plain;charset=UTF-8"})
    @ResponseBody
    public String matrixVariableTest(
        @PathVariable String product,
        @MatrixVariable(pathVar="product", value="model", required=true) String model,
        @MatrixVariable(pathVar="product", value="colors", required=false) String[] colors,
        @MatrixVariable(pathVar="product", value="year", required=true) int year,
        @MatrixVariable Map<String, Object> matrixMap
    ){
        log.info("product{};model={};colors={};year={}", product, model,
            Arrays.toString(colors), year);
        log.info("Matrix Map : {}", matrixMap.toString());
        StringBuffer productDesc = new StringBuffer("");
        productDesc.append(model+" , 연식 : "+year+" , 색상 : "+Arrays.toString(colors));
        return productDesc.toString();
    }

    @RequestMapping(value="/{product}", produces={"text/html", "application/xml"})
    public String matrixVariableTest(
        @PathVariable String product,
        @MatrixVariable(pathVar="product", value="model", required=true) String model,
        @MatrixVariable(pathVar="product", value="colors", required=false) String[] colors,
        @MatrixVariable(pathVar="product", value="year", required=true) int year,
        @MatrixVariable Map<String, Object> matrixMap,
        Model modelMap
    ){
        String productDesc = matrixVariableTest(product, model, colors, year, matrixMap);
        modelMap.addAttribute("productDesc", productDesc);
        return "matrix/matrixView";
    }
}
```

– /WEB-INF/views/matrix/matrixForm.jsp.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<c:url value="/matrix/cars" var="matrixUrl"/>
<script>
    $(function(){
        $("[name='year']").on("click", function(){
            $("#submitBtn").trigger("click");
        });
    });
    function submitFunc(){
        var formArray = $("#carsForm").serializeArray();
        var text = "";
        $.each(formArray, function(index, element){
            text += ";" + element.name + "=" + element.value;
        });
        $.ajax({
            url: "${matrixUrl}" + text,
            method: "post",
            dataType: "text",
            // dataType: "html",
            success: function(response){
                $("#result").html(response);
            },
            error: function(errorResponse){
                $("#result").hide();
                $("#errorFrame").contents().find("body").html(errorResponse.responseText);
                $("#errorFrame").show();
            }
        });
    }
}
</script>
<div class="form">
    <h4>제품선택</h4>
    <form id="carsForm" method="post">
    <table style="text-align: center;">
        <thead><tr><th>분류</th><th>색상</th><th>연식</th></tr></thead>
        <tbody>
            <tr>
                <td>
                    <select name="model" size="10">
                        <option value="avante">아반떼</option>
                        <option value="sonata">소나타</option>
                        <option value="malibu">말리부</option>
                        <option value="korando">코란도</option>
                    </select>
                </td>
            </tr>
        </tbody>
    </table>
    </div>
```

```

        <td>
            <label><input type="checkbox" name="colors" value="red" />빨강
            </label><br />
            <label><input type="checkbox" name="colors" value="yellow" />노랑
            </label><br />
            <label><input type="checkbox" name="colors" value="blue" />파랑
            </label><br />
            <label><input type="checkbox" name="colors" value="gray" />회색
            </label><br />
            <label><input type="checkbox" name="colors" value="black" />검정
            </label><br />
        </td>
        <td>
            <label><input type="radio" name="year"
            value="2014"/>2014</label><br />
            <label><input type="radio" name="year"
            value="2013"/>2013</label><br />
            <label><input type="radio" name="year"
            value="2012"/>2012</label><br />
            <label><input type="radio" name="year"
            value="2011"/>2011</label><br />
        </td>
    </tr>
    <tr><td colspan="3"><input type="button" id="submitBtn" value="선택완료"
    onclick="submitFunc()" /></td></tr>
</tbody>
</table>
</form>
<div id="result"></div>
<iframe id="errorFrame" style="display: none; width: 100%;"></iframe>
</div>

```

– /WEB-INF/views/matrix/matrixView.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<h4>매트릭스 데이터</h4>
<div style="border: 1px solid red;">
    경로 변수로 확보된 인자 : ${product }<br />
    <c:set var="carsAttribute"
    value="${requestScope['org.springframework.web.servlet.HandlerMapping.matrixVariables'].product }" />
    매트릭스 변수로 확보된 인자 :[${carsAttribute.model } , ${carsAttribute.colors },
    ${carsAttribute.year }]
    == ${productDesc }
    <br />
    <c:forEach
    items="${requestScope['org.springframework.web.servlet.HandlerMapping.matrixVariables']}"
    var="matrixEntry">
        ${matrixEntry.key }=${matrixEntry.value }<br />
    </c:forEach>
</div>
<h4>뷰로 전달된 데이터 종류</h4>
<div style="border: 1px solid red;">
    <c:forEach items="${requestScope }" var="entry">
        <span style="color: red;">${entry.key }</span>=${entry.value }<br />
    </c:forEach>
</div>

```


- 스타일 및 테스트를 위한 부가 파일 작성.
 - /src/main/webapp/resources/css/main.css

```
@CHARSET "UTF-8";
.back {
border: 1px solid black;
}
.errors{
color: red;
}
#result{
background-color: lime;
}
#errorResult{
background-color: silver;
}
/* css/main.css */
body {margin:0;padding:0;background-color:#fff;text-align:center;}
hr {display:none;clear:both;} /* 레이아웃 구분(top/left/contents/footer)에만 적용 */
img,fieldset {border:0 none;}
dl,ul,ol,li {list-style: none;}
h1,h2,h3,h4,h5,dl,dt,dd,ul,li,ol,th,td,p,blockquote,form,fieldset,legend {margin:0;padding:0;}
table {border-collapse:collapse;}
a {color:#333;text-decoration:none;} a:hover, a:active {text-decoration:underline;}
#WRAP .hover, #WRAP a:hover {color:#cc0000;}
input,select,textarea {font-size:100%; vertical-align:middle;}
/*body,th,td {font-size:12px;font-family:굴림,gulim,sans-serif;color:#333;}*/
body,th,td {font-size:12px;font-family:돋움,Dotum,AppleGothic,serif; color:#333;
border:1px solid black;
}
/*기본 CSS - 추가 가능*/
h1,h2{display:none;} /*h2까지 템플릿에서 사용, 각 작업은 h3부터*/
/*h3 { font:bold 12px 굴림,gulim,sans-serif; }*/
h3 {font-size:12px;}
em{font-style:normal;}
.hidden{display:none;}
.ls {letter-spacing:-1px;}
.fl{float:left !important;}
.fr{float:right !important;}
.b {font-weight:bold !important; letter-spacing:-1px;}
.clb,.clr {clear:both;}
.border{border:1px #f00 solid;}
img.middle {vertical-align:middle; }
img.top {vertical-align:top; }
.cursor{cursor:pointer;}
.png { tmp:expression(setPngImg(this));}
p.title {font-size: 130%; font-weight: bold; font-style: italic;
text-align: left; padding-left: 10px; margin-top:15px;}
.left {text-align: left; padding-left: 5px;}
.right{text-align: right; padding-right: 5px;}
.center{text-align: center}
.bg_gray {background-color: #dddddd;}
.bg_pinky {background-color: #f04040;}
.wd_60 { width: 60px}
.wd_300 { width: 300px}
```

- /src/main/webapp/resources/css/main.css 계속

```
.buttonArea {text-align: center; line-height: 25px;}
.buttonArea button {width: 120px; text-align: center; }
.buttonArea a {border: 1px solid;
padding: 3px 30px 3px 30px;
background-color: #d0d0d0;}

/* board */
.list table {width: 80%; min-width: 600px; margin: 0 auto;}
.list th{ min-width:60px; border: 1px red solid; background-color: skyblue;}
.list th,
.list td {line-height: 18px; }
.list tbody tr:hover {background-color: #d0d0d0; }
.list tbody td {border: 1px black solid; }
.list h3, #board h4 {text-align: left; margin: 15px 20px; }

.form input {border:1px solid #a0a0; }
.form table {width: 80%; min-width: 600px; margin: 0 auto;}
.form th{ min-width:60px; border: 1px red solid; background-color: skyblue;}
.form th,
.form td {line-height: 20px; }
.form td {text-align: left; padding-left: 10px }
.form tbody td {border: 1px black solid; }
.form h3, .form h4 {text-align: left; margin: 15px 20px; }

/* 페이징 */
#pagingNew {clear:both;font:13px Verdana;margin:41px 0 4px 0;text-align:center;width:743px;}
#pagingNew a {border:1px solid #ecec; height:23px;text-decoration:none;}
#pagingNew a:hover {border:1px solid #95a5ff;color:#6671ff;}
#pagingNew .sel {border:1px solid #95a5ff;color:#6671ff;font-weight:bold;height:23px;padding:2px 5px 0 5px;}
#pagingNew .numBox {color:#000;font-weight:bold;padding:2px 5px 0 5px;}
#pagingNew .numPrev
{background:url("http://i1.daumcdn.net/imgsrc.search/search_all/2008_new/200806/bLL_01.gif") no-repeat 6px 6px;color:#000;height:23px;margin-right:4px;overflow:hidden;padding:3px 6px 4px 17px;*padding:5px 6px 5px 17px;font-size:12px;*position:relative;
*top:-1px;}
#pagingNew .numNext {color:#000;margin-left:4px;padding:3px 3px 4px 3px;*padding:3px 2px 0px 2px;font-size:12px;*position:relative;
*top:-1px;}
```

- 테스트용 인덱스 (/WEB-INF/views/index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Index view</title>
    </head>
    <body>
        <a href='<c:url value="/board/boardList.do"/>'>게시판</a><br />
        <a href='<c:url value="/matrix"/>'>매트릭스변수테스트</a>
    </body>
</html>
```

□ DataSource JNDI 설정 방법.

1. GlobalResource 등록(naming) 방법 - dataSource 를 서버에서 일괄 관리하는 경우.

1) catalina.properties 에 dataSource 생성 프로퍼티 등록.

```
mainDB.driverClassName=oracle.jdbc.driver.OracleDriver
mainDB.url=jdbc:oracle:thin:@localhost:1521:xe
mainDB.username=sem
mainDB.password=java
```

2) servlet.xml 에 GlobalResources 등록.

참고 : tomcat7 부터 DataSource 생성 Factory 로 commons-dbcپ를 사용하지 않고, toomcat-jdbc.jar 내에 있는 Factory 를 사용함.

```
<GlobalNamingResources>
  <Resource
    auth="Container"
    factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
    driverClassName="${mainDB.driverClassName}"
    name="jdbc/mainDB"
    password="${mainDB.password}"
    type="javax.sql.DataSource"
    url="${mainDB.url}"
    username="${mainDB.username}" />
</GlobalNamingResources>
```

3) context.xml 에 ResourceLink 등록.

```
<Context>
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <ResourceLink name="jdbc/mainDB"
    global="jdbc/mainDB"
    type="javax.sql.DataSource" />
</Context>
```

2. 어플리케이션별로 별도 관리하는 dataSource 등록(naming) 방법.

web content 영역(context root 아래 META-INF/context.xml 생성 후 Resource 등록.

```
<Context>
  <Resource
    auth="Application"
    driverClassName="${mainDB.driverClassName}"
    factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
    name="jdbc/mainDB"
    type="javax.sql.DataSource"
    url="${mainDB.url}"
    username="${mainDB.username}"
    password="${mainDB.password}"
  />
</Context>
```

- 참고사이트

- <http://tomcat.apache.org/tomcat-7.0-doc/config/globalresources.html>
- http://tomcat.apache.org/tomcat-7.0-doc/config/context.html#Resource_Definitions
- <http://tomcat.apache.org/tomcat-7.0-doc/jdbc-pool.html>

□ 계층형 게시판 참고 (일반형 게시판을 계층형으로 변경하는 경우)

- 재귀쿼리를 지원하지 않는 DB를 사용하는 경우, 계층형을 처리하기 위해 순차번호(sequence_no)를 이용한다.

1. 최초의 루트글에 대해 깊이가 증가하는 답글들에 대해 동일 그룹을 의미하는 group_no 를 부여함.
2. 답글의 깊이를 의미하는 level 을 부여함(깊이가 증가함에 따라 level + 1).
3. 특정글에 대해 답글의 깊이를 어디까지 허용할 것인지 결정(LEVELLIMIT).
4. 특정글의 동일 레벨 답글들의 개수를 몇 개까지 허용할 것인지 결정(SAMELEVELCHARS).
 - 1) 98~0까지 (99개 허용)이라면, SAMELEVELCHARS=2
 - EX) SAMELEVELCHARS=2, LEVELLIMIT=5 라면, sequence_no의 자릿수는 10 ('9999999999')

bo_no	group_no	sequence_no	level
1	1	9999999999	0
2	1	->9899999999	1
3	1	->9799999999	1
5	1	->9798999999	2
8	1	->9798989999	3
9	1	->9798979999	3
10	1	->9798969999	3
11	1	->9798969899	4
12	1	->9798969898	5
6	1	->9797999999	2
7	1	->9796999999	2
4	1	->9699999999	1

5. 게시글의 저장 영역(BOARD)에 group_no, sequence_no 를 추가.
 - group_no : 동일 부모의 자식글임을 알수있도록, 최상위 부모글의 번호를 그룹번호로 설정.
 - sequence_no : default value 로 '99999999999999'를 설정

```
ALTER TABLE BOARD ADD ( GROUP_NO NUMBER, SEQUENCE_NO CHAR(10) DEFAULT '9999999999');
UPDATE BOARD SET GROUP_NO = BO_NO; COMMIT;
```

6. 답글의 깊이(레벨)은 BoardVO에서 sequence_no를 통해 산출한다.

```
/**
 * 답글의 깊이 제한 , 깊이에 제한에 따라 데이터베이스 BOARD 테이블의 SEQUENCE_NO의 컬럼 사이즈와 기본값도 달라져야 한다.
 */
public static final int LEVELLIMIT = 5;
/**
 * 동일 레벨 답글에 대해 허용되는 문자 수 (각 레벨당 답글 수는 99개까지 허용, 답글 정책에 따라 수정 가능).
 * 99, 98, 97, ... 00
 */
public static final int SAMELEVELCHARS=2;
private Integer group_no;
private String sequence_no;
private Integer bo_parent;
/**
 * if(sequence_no.endsWith("9999999999")) return 0;
 * else if(sequence_no.endsWith("99999999")) return 1;
 * else if(sequence_no.endsWith("999999")) return 2;
 * else if(sequence_no.endsWith("9999")) return 3;
 * else if(sequence_no.endsWith("99")) return 4;
 * else return 5;
 * 한 레벨당 '9'문자수(2)에 따라 9문자의 반복횟수를 가진 패턴을 사용하면, 좀더 확장성 있는 코드를 구현할 수 있다.
 */
public int getLevel(){
    Pattern pattern = Pattern.compile("(\\[9\\{ "+SAMELEVELCHARS+"\\}\\}\\$"); // 정규식 : ([9]{2,})
    if(sequence_no==null || sequence_no.length() != (LEVELLIMIT * SAMELEVELCHARS)){
        return -1;
    }else{
        Matcher matcher = pattern.matcher(sequence_no);
        // 레벨당 '9' 문자수(2)횟수로 반복되는 패턴이 없다면, 이는 가장 마지막 레벨임을 의미함.
        int findIdx = LEVELLIMIT * SAMELEVELCHARS;
        if(matcher.find()){
            findIdx = matcher.start(); // '99'가 처음 시작되는 위치
        }
        // 찾아진 index 범위 : 0(0레벨, 루트글), 1~2(1레벨), 3~4(2레벨), 5~6(3레벨)...
        return (findIdx + (SAMELEVELCHARS - 1)) / SAMELEVELCHARS;
    }
}
```


2) 그룹번호와 순차번호를 삽입하기 위해 insert 쿼리 수정.

```
<insert id="insertBoard" parameterClass="boardVO">
  <selectKey resultClass="int" keyProperty="bo_no">
    SELECT BOARD_SEQ.NEXTVAL FROM DUAL
  </selectKey>
  INSERT INTO BOARD
  (
    BO_NO,      BO_TITLE,    BO_WRITER,
    BO_PASSWD,   BO_EMAIL,    BO_IP,
    BO_CONTENT,  BO_HIT,      BO_DATE
    , GROUP_NO
    <dynamic prepend=" , ">
      <isEmpty property="sequence_no">
        SEQUENCE_NO
      </isEmpty>
    </dynamic>
  )
  VALUES
  (
    #bo_no#,      #bo_title#,    #bo_writer#,
    #bo_passwd#,   #bo_email#,    #bo_ip#,
    #bo_content#,  0,            SYSDATE
    <dynamic>
      <isEmpty property="group_no" prepend=" , ">
        #group_no#
      </isEmpty>
      <isEmpty property="group_no" prepend=" , ">
        #bo_no#
      </isEmpty>
      <isEmpty property="sequence_no" prepend=" , ">
        #sequence_no#
      </isEmpty>
    </dynamic>
  )
</insert>
```

3) 부모글의 동일 레벨 자식글 들 중 가장 마지막 자식글의 순차 번호를 조회하는 메소드 추가.

```
/**
 * 동일 그룹 내에 있는 시퀀스번호 중에서 부모글과 부모글 바로 다음에 작성된 동일 레벨의 글 사이의 시퀀스 번호
 * 중 가장 작은 시퀀스번호를 조회
 * @param group_no 그룹번호, maxSeq 부모글의 시퀀스번호, minSeq 부모글과 동일 레벨의 글 중 바로 다음에
 * 작성된 글의 시퀀스 번호를 가진 map.
 */
public String selectMinSequenceNoWithSameGroup(Map<String, Object> paramMap);
```

13. 신규글 등록 로직을 가진 BoardServiceImpl 에서 현재 글이 답글일 경우에 대한 로직 추가.

1) 부모글 검증을 위한 메소드(validParent) 추가 : createBoard 에서 새글을 등록(dao.insertBoard)하기 전에 부모글이 존재하는지 혹은 더 이상의 답글이 존재할 수 있는지(LEVELLIMIT 체크) 등을 확인함.

```
/**
 * 부모글이 존재하는지 답글의 깊이는 적절한지 확인하기 위한 메소드
 */
private boolean validParent(BoardVO parent){
  if(parent == null){
    throw new RuntimeException(" 부모글이 존재하지 않습니다. ");
  }else if(parent.getLevel()<0||parent.getLevel()>=BoardVO.LEVELLIMIT){
    throw new RuntimeException(parent.getGroup_no()+"번 글에는 더이상 답글을 허용하지 않습니다.");
  }else{
    return true;
  }
}
```

- 2) 답글일 경우, 순차번호(sequence_no)를 결정하기 위한 메소드 추가.
- 3) createBoard 에서 신규글을 등록하기 전, 답글인지 여부와 답글이 작성될 수 있는지 여부를 확인하고, 답글의 그룹번호와 순차 번호를 결정한 다음 insert 를 한다.

```

/**
 * 부모글의 시퀀스 번호를 사용해 자식글의 시퀀스 번호를 결정.
 */
private String getSequenceNO(BoardVO parent) {
    int sequenceDepth = BoardVO.LEVELLIMIT * BoardVO.SAMELEVELCHARS; // sequence_no 길이
    int level = parent.getLevel(); // 부모글의 레벨
    long parentSeq = Long.parseLong(parent.getSequence_no()); // 부모글의 시퀀스 번호
    long minSeq = 0;
    // 답글의 깊이가 증가함에 따라 순차번호의 증가 범위를 찾아내기 위한 결정자 1010, 108, 106, 104, 102
    long levelDeterminant = (long) Math.pow(10, BoardVO.SAMELEVELCHARS * (BoardVO.LEVELLIMIT - level));
    // ex) 9799999999 부모글의 자식글들은 9798999999 ~ 9700999999 까지 시퀀스 번호를 가짐.
    minSeq = (parentSeq / levelDeterminant) * levelDeterminant; // ex) 9700000000
    // 레벨제한과 각 레벨당 답글수제한에 따라 숫자 포맷이 달라짐.
    // ex) 5레벨까지 99개를 허용하기 위해 10자리 포맷 '0000000000'을 결정.
    DecimalFormat formatter = new DecimalFormat(String.format("%0"+sequenceDepth+"d", 0));
    Map<String, Object> paramMap = new HashMap<String, Object>();
    paramMap.put("group_no", parent.getGroup_no());
    paramMap.put("maxSeq", parent.getSequence_no());
    paramMap.put("minSeq", formatter.format(minSeq));

    // 현재 부모글의 동일 레벨 자식글 중 가장 마지막 자식글의 시퀀스 번호를 가져옴.
    String minSeqSameGroup = boardDAO.selectMinSequenceNoWithSameGroup(paramMap);
    if (minSeqSameGroup == null) { // 자식글이 없는 경우, 부모글의 시퀀스 번호를 다시 사용.
        minSeqSameGroup = parent.getSequence_no();
    }
    // 부모글에서 1레벨을 낮춰 자식글의 시퀀스번호를 결정하기 위한 결정자 수정
    // ex) 9799999999 부모글의 자식글들을 9798999999 ~ 9700999999 까지 시퀀스 번호를 가짐.
    levelDeterminant = levelDeterminant / (long) Math.pow(10, BoardVO.SAMELEVELCHARS);
    long offset = levelDeterminant - 1; // 새로운 답글의 시퀀스번호에서 부모글의 하위 레벨을 결정하는
    // 자릿수는 '9'로 설정.
    long sequence_no = (Long.parseLong(minSeqSameGroup) / levelDeterminant - 1) * levelDeterminant
    + offset; // ex) 9798999999
    if (sequence_no < minSeq) {
        throw new RuntimeException(" 동일 레벨에 대해 허용 가능한 댓글수를 넘었습니다. 마지막
        자식글이 이미 있는 거지요.");
    }
    return formatter.format(sequence_no);
}

@Override
@Transactional(propagation=Propagation.REQUIRED)
public ServiceResult createBoard(BoardVO board) {
    // 답글쓰기로 올라온 글이라면, 해당 답글의 그룹번호와 시퀀스 번호를 설정하기 위한 로직이 필요하다.
    Integer parent_no = board.getBo_parent();
    BoardVO parent = null;
    if (parent_no != null && parent_no > 0) {
        parent = boardDAO.selectBoard(parent_no);
        boolean valid = validParent(parent);
        if (!valid) {
            throw new CommonException(parent_no + "번 글에 대해 답글을 작성할 수
            없습니다.");
        }
        // 자식글의 그룹번호는 부모글의 그룹번호를 따라감.
        board.setGroup_no(parent.getGroup_no());
        // 계산된 시퀀스번호 설정
        board.setSequence_no(getSequenceNO(parent));
    }
    Integer pk = boardDAO.insertBoard(board);
    //... 이후는 동일
}

```