## 1. Database Configuration

1)계정생성

```
sqlplus /nolog

conn SYS as sysdba

ALTER SESSION SET "_ORACLE_SCRIPT"=true;

CREATE USER bbs
IDENTIFIED BY bbs
DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE temp;

ALTER USER bbs DEFAULT TABLESPACE USERS
QUOTA UNLIMITED ON users;

GRANT resource, connect TO bbs;
```

2)Sequence 및 Table 생성

```
CREATE SEQUENCE seq_board_idx
MAXVALUE 9999;

CREATE TABLE Board
(
    idx           NUMBER(4),
    name          VARCHAR2(20)      NOT NULL,
    email         VARCHAR2(100),
    title         VARCHAR2(100)     NOT NULL,
    contents   VARCHAR2(2000)    NOT NULL,
    writedate    DATE     DEFAULT   SYSDATE,
    readnum    NUMBER(4)   DEFAULT   0,
    grp           NUMBER(4)          NOT NULL,
    lev           NUMBER(2)       DEFAULT  0,
    step          NUMBER(2)       DEFAULT  0,
    CONSTRAINT board_idx_pk  PRIMARY KEY(idx)
);
```

## 2. Java Configuration

1)In J2EE Perspective

2)Project Explorer > right-click > New > Dynamic Web Project

3)Project name : SpringBoard > Next > Check [Generate web.xml deployment descriptor] > Finish

4)Convert to Maven Project

-project right-click > Configure > Convert to Maven Project > Finish

5)Add Spring Project Nature

-project right-click > Spring Tools > Add Spring Project Nature

6)새로 생성된 pom.xml파일에 필요한 library 추가 > Maven Clean > Maven Install

```
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>4.3.17.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>4.3.17.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>com.oracle</groupId>
        <artifactId>ojdbc8</artifactId>
        <version>12.2</version>
    </dependency>
    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis</artifactId>
        <version>3.4.6</version>
    </dependency>
    <dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis-spring</artifactId>
        <version>1.3.2</version>
    </dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>4.3.17.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>2.9.5</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-core</artifactId>
        <version>2.9.5</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-annotations</artifactId>
        <version>2.9.5</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-test</artifactId>
        <version>4.3.17.RELEASE</version>
        <scope>test</scope>
    </dependency>
  </dependencies>
```

7)Build path에 config foler 추가

-Project right-click > Build Path > Configure Build Path > Select [Source] tab
-Click [Add Folder] > Select 현재 project > Click [Create New Folder...]
-Folder name : config > Finish > OK > Apply and Close

-Java Resources > config 폴더 확인

3. MyBatis Configuration

1)config folder에 dbinfo.properties file 추가

-config folder > right-click > New > File

-File name : dbinfo.properties

```
db.driver=oracle.jdbc.driver.OracleDriver
db.url=jdbc:oracle:thin:@192.168.56.3:1521:ORCL
db.username=bbs
db.password=bbs
```

2)config folder에 mybatis-config.xml file 생성

-config folder > right-click > New > Other > XML > XML File > Next

-File name : mybatis-config.xml > Finish

```
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
</configuration>
```

3)config folder에 board-mapper.xml file 생성

-config folder > right-click > New > Other > XML > XML File > Next

-File name : board-mapper.xml > Finish

```
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="Board">
</mapper>
```

4. Spring Configuration

1)config folder에 applicationContext.xml 파일 생성

-Spring Perspective로 전환

-config right-click > New > Spring Bean Configuration File

-File name : applicationContext.xml

-생성시 beans,context, mvc 체크

```
<?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"
    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-3.2.xsd">

</beans>
```

2)ContextLoaderListener class 설정

-WebContent/WEB-INF/web.xml 수정

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="
http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="
http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID"
```

```
version="3.1">
    <display-name>SpringBoard</display-name>
    <welcome-file-list>
      <welcome-file>index.html</welcome-file>
    </welcome-file-list>


</web-app>
```

-Ctrl + Spacebar를 하면 나타나는 Context Menu에서 [#contextloaderlistener - ContextLoaderListener] 를 선택하면 아래의 코드가 자동 삽입

```
<!-- needed for ContextLoaderListener -->
<context-param>
   <param-name>contextConfigLocation</param-name>
   <param-value>location</param-value>
</context-param>

<!-- Bootstraps the root web application context before servlet initialization -->
<listener>
   <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

--아래 코드로 변환

```
<context-param>
   <param-name>contextConfigLocation</param-name>
   <param-value>classpath:applicationContext.xml</param-value>
</context-param>
```

3)DispatcherServlet Class 추가

-web.xml에서 Ctrl + Spacebar 하면 나타나는 Context Menu에서 [#dispatcherservlet - DispatcherServlet declaration] 선택하면 아래의 코드가 자동 추가된다.

```
<!-- The front controller of this Spring Web application, responsible for handling all application requests -->
<servlet>
   <servlet-name>springDispatcherServlet</servlet-name>
   <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
   <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>location</param-value>
   </init-param>
   <load-on-startup>1</load-on-startup>
</servlet>

<!-- Map all requests to the DispatcherServlet for handling -->
<servlet-mapping>
   <servlet-name>springDispatcherServlet</servlet-name>
   <url-pattern>url</url-pattern>
</servlet-mapping>
```

--아래의 코드로 변환

```
<init-param>
   <param-name>contextConfigLocation</param-name>
   <param-value>classpath:applicationContext*.xml</param-value>
</init-param>
```

```
<servlet-mapping>
    <servlet-name>springDispatcherServlet</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

5. index.html 생성 및 정적 페이지 처리하기

1)applicationContext.xml file 수정

```
<mvc:default-servlet-handler /> 추가
```

2)index.html 생성

-WebContent > right-click > New > Other > Web > HTML File > Next

-File name : index.html > Finish

```
<!DOCTYPE html>
<html>
    <head>
    <meta charset="UTF-8">
    <title>Spring Board</title>
    </head>
    <body>
        <h1>Spring Board</h1>
    </body>
</html>
```

3)style.css file 생성

-WebContent > right-click > New > Folder

-Folder name : css > Finish

-css > right-click > New > Other > Web > CSS File > Next

-File name : style.css > Finish

```
@charset "UTF-8";
body{
    background-color: yellow;
}
h1 {
    color : blue;
}
```

4)jQuery file 추가하기

-WebContent > right-click > New > Folder

-Folder name : js > Finish

-이미 download 받은 jquery-1.12.4.js 파일 넣기

5)index.html 수정

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Spring Board</title>
        <link rel="stylesheet" type="text/css" href="css/style.css" />
        <script src="js/jquery-1.12.4.js"></script>
        <script>
            $(document).ready(function(){
                alert("Hello Spring Board");
            });
        </script>
    </head>
    <body>
        <h1>Spring Board</h1>
    </body>
</html>
```

6)Project > right-click > Run As > Run on Server > Finish

7)http://localhost:8080/SpringBoard/

## 6. Java Package 및 관련 Class file 생성

1)package 생성

-src/com.javasoft.vo
-src/com.javasoft.dao
-src/com.javasoft.service
-src/com.javasoft.controller

2)BoardVO class 생성

-src/com.javasoft.vo.UserVO class 생성

```
package com.javasoft.vo;

public class BoardVO {
    private int idx, readnum, grp, lev, step;
    private String name, email, title, contents;
    private String writedate;

    public BoardVO() {}

    public BoardVO(int grp, String name, String email, int idx, String title, int lev, int step, String writedate, int readnum) {
        this.idx = idx;     this.readnum = readnum;
        this.grp = grp;     this.lev = lev;
        this.step = step;   this.name = name;
        this.email = email;  this.title = title;
        this.writedate = writedate;
    }

    public BoardVO(int grp, String name, String email, int idx, String title, int lev, int step, String Contents, String writedate, int readnum) {
        this.idx = idx;     this.readnum = readnum;
        this.grp = grp;     this.lev = lev;
        this.step = step;   this.name = name;
        this.email = email;  this.title = title;
        this.contents = contents;
        this.writedate = writedate;
    }

    public int getIdx() {
        return idx;
    }
    public void setIdx(int idx) {
        this.idx = idx;
    }
    public int getReadnum() {
        return readnum;
    }
    public void setReadnum(int readnum) {
        this.readnum = readnum;
    }
    public int getGrp() {
        return grp;
    }
    public void setGrp(int grp) {
        this.grp = grp;
    }
    public int getLev() {
        return lev;
    }
    public void setLev(int lev) {
        this.lev = lev;
    }
    public int getStep() {
        return step;
    }
    public void setStep(int step) {
        this.step = step;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getContents() {
        return contents;
    }
    public void setContents(String contents) {
        this.contents = contents;
    }
    public String getWritedate() {
        return writedate;
    }
    public void setWritedate(String writedate) {
        this.writedate = writedate;
    }
    @Override
    public String toString() {
        return "BoardVO [idx=" + idx + ", readnum=" + readnum + ", grp=" + grp + ", lev=" + lev + ", step=" + step
                + ", name=" + name + ", email=" + email + ", title=" + title + ", contents=" + contents + ", writedate="
                + writedate + "]";
    }
}
```

3)BoardService 생성

-src/com.javasoft.service.BoardService interface 생성

```
package com.javasoft.service;

import java.util.List;

import com.javasoft.vo.BoardVO;

public interface BoardService {
    void insertBoard(BoardVO board);
    BoardVO selectBoard(int idx);
    List<BoardVO> select();
    void updateBoard(BoardVO board);
    void deleteBoard(int idx);
    void answerBoard(BoardVO board);
    int getTotalCount();
}
```

-src/com.javasoft.service.BoardServiceImpl class 생성

```
package com.javasoft.service;

import java.util.List;

import com.javasoft.vo.BoardVO;

public class BoardServiceImpl implements BoardService {

    @Override
    public void insertBoard(BoardVO board) {
        // TODO Auto-generated method stub

    }

    @Override
    public BoardVO selectBoard(int idx) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public List<BoardVO> select() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void updateBoard(BoardVO board) {
        // TODO Auto-generated method stub

    }

    @Override
    public void deleteBoard(int idx) {
        // TODO Auto-generated method stub

    }

    @Override
    public void answerBoard(BoardVO board) {
        // TODO Auto-generated method stub
```

```java
    }

    @Override
    public int getTotalCount() {
        // TODO Auto-generated method stub
        return 0;
    }
}
```

4)BoardDao 객체 생성

-src/com.javasoft.dao.BoardDao interface

```java
package com.javasoft.dao;

import java.util.List;

import com.javasoft.vo.BoardVO;

public interface BoardDao {
    void create(BoardVO board);
    BoardVO read(int idx);
    List<BoardVO> readAll();
    void update(BoardVO board);
    void delete(int idx);
    void answer(BoardVO board);
    int getTotalCount();
}
```

-src/com.javasoft.dao.BoardDaoImpl.java 생성

```java
package com.javasoft.dao;

import java.util.List;

import com.javasoft.vo.BoardVO;

public class BoardDaoImpl implements BoardDao {

    @Override
    public void create(BoardVO board) {
        // TODO Auto-generated method stub

    }

    @Override
    public BoardVO read(int idx) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public List<BoardVO> readAll() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void update(BoardVO board) {
        // TODO Auto-generated method stub
    }

    @Override
    public void delete(int idx) {
        // TODO Auto-generated method stub

    }

    @Override
    public void answer(BoardVO board) {
        // TODO Auto-generated method stub

    }

    @Override
    public int getTotalCount() {
        // TODO Auto-generated method stub
        return 0;
    }

}
```

5)BoardController 객체 생성

-src/com.javasoft.controller.BoardController class 생성

```
package com.javasoft.controller;

public class BoardController {

}
```

6)applicationContext.xml 수정

```
<mvc:annotation-driven />
<mvc:resources location="/views/**" mapping="/WEB-INF/views/" />

<context:component-scan base-package="com.javasoft" />
<context:property-placeholder location="classpath:dbinfo.properties"/>

<bean id="dataSource"
    class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
    <property name="driverClass" value="${db.driver}" />
    <property name="url" value="${db.url}" />
    <property name="username" value="${db.username}" />
    <property name="password" value="${db.password}" />
</bean>

<bean id="sqlSessionFactory"
    class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
    <property name="configLocation"
        value="classpath:mybatis-config.xml" />
    <property name="mapperLocations">
        <list>
            <value>classpath:board-mapper.xml</value>
        </list>
    </property>
</bean>
```

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

    <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/views/" />
        <property name="suffix" value=".jsp" />
    </bean>
```

7)/WEB-INF/views folder 생성
-WEB-INF > right-click > New > Folder
-Folder name : views > Finish

8)WebContent > images folder 생성
-images folder에 아래와 같은 image 파일들 추가할 것
-banner.jpg
-board.png
-reply.jpg

9)/WEB-INf/views/index.jsp 생성
-WEB-INF > right-click > New > Other > Web > JSP File > Next
-File name : index.jsp > Finish

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Spring Board</title>
        <link rel="stylesheet" href="../css/style.css" type="text/css">
        <script src="../js/jquery-1.12.4.js"></script>
    </head>
    <body>
        <img src="../images/banner.jpg" width="100%" height="250" />
        <div id="banner">
            <img src="../images/board.png" width="30" height="30" />
            <span>SpringBoard</span>
        </div>
        <div>
            <table border="0">
                <tr>
                    <td>[<a href="write">새 글쓰기</a>]
                    </td>
                    <td>[<span id="count"></span>]
                    </td>
                </tr>
            </table>
        </div>
        <div>
            <table border="1">
                <colgroup>
                    <col width="10%" />
                    <col width="10%" />
                    <col width="60%" />
                    <col width="10%" />
                    <col width="10%" />
                </colgroup>
                <thead>
                    <tr>
```

```
                            <th>번호</th>
                            <th>작성자</th>
                            <th>제목</th>
                            <th>작성일</th>
                            <th>조회수</th>
                        </tr>
                    </thead>
                    <tbody id="result"></tbody>
                </table>
            </div>
            <p></p>
            <div id="paging"></div>
        </body>
    </html>
```

10)index.html 수정

```
<script>
    document.location = "board/";
</script>
```

11)BoardController.java 수정

```
@Controller
public class BoardController {
    @Autowired
    private BoardService boardService;

    @RequestMapping(value = "board/", method = RequestMethod.GET)
    public String index() {
        return "index";
    }
```

12)BoardServiceImpl.java 수정

```
@Service("boardService")
public class BoardServiceImpl implements BoardService {
    @Autowired
    private BoardDao boardDao;
}
```

13)BoardDaoImpl.java 수정

```
@Repository("boardDao")
public class BoardDaoImpl implements BoardDao {
    @Autowired
    private SqlSession sqlSession;

}
```

14)style.css 수정

```
body {
    margin-top : 0px;
    margin-left : 0px;
    margin-right:0px;
}
#banner{
    text-align:center;
}
```

```
    #banner span {
        font-size:2em;
    }
    div:nth-of-type(2) table {
        width:80%;
        margin-left:auto; margin-right:auto;
    }
    div:nth-of-type(2) table td:nth-of-type(1) {
        text-align:left;
    }
    div:nth-of-type(2) table td:nth-of-type(2) {
        text-align:right;
    }
    div:nth-of-type(3) table {
        width:80%;
        margin-left:auto; margin-right:auto;
        border-collapse: collapse;
    }
    div:nth-of-type(3) table th{
        height:50px;
        background-color: navy;
        color:white;
    }
```

15)project > right-click > Run As > Run on Server > Finish

16)http://localhost:8080/SpringBoard/board/

7. 새 글쓰기

1)BoardController.java 코드 추가

```
    @RequestMapping(value = "board/write", method = RequestMethod.GET)
    public String write() {
        return "write";   //   /WEB-INF/views/write.jsp
    }
```

2)/WEB-INF/views/write.jsp 파일 생성

```
    <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
    <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
    <!DOCTYPE html>
    <html lang="en">
    <head>
         <meta charset="UTF-8">
         <title>게시판 글 쓰기</title>
         <script src="<c:url value='/js/jquery-1.12.4.js'/>"></script>
         <script>
            $(document).ready(function(){
                $("input[type='button']").bind("click", function(){
                    $.ajax({
                        url : "/SpringBoard/board",
                        method : "POST",
                        dataType : "json",
                        contentType : "application/json;charset=utf-8",
                        data : JSON.stringify({
                            "name" : $("#name").val(),
                            "email" : $("#email").val(),
                            "title" : $("#title").val(),
                            "contents" : $("#contents").val()
```

```
          }),
          success : function(data){
            if(data.code){
              alert("Insert Success");
            }
            location.href = "/SpringBoard";
          }
        });
      });
    });
  </script>
</head>
<body>
  <h1>게시판 글 쓰기</h1>
  <form>
    <ul style="list-style-type:none">
        <li>작성자 : <input type="text" id="name" /></li>
        <li>Email : <input type="email" id="email" size="30" /></li>
        <li>제목 : <input type="text" id="title" size="40"/></li>
        <li>내용 : <textarea id="contents" rows="10" cols="40"></textarea></li>
    </ul>
    <input type="button" value="글 쓰기" />
  </form>
</body>
</html>
```

3)BoardController.java 코드 추가

```
@RequestMapping(value = "board/", method = RequestMethod.POST)
@ResponseBody
public Map insert(@RequestBody BoardVO board) {
    Map<String, Object> result = new HashMap<String, Object>();
    this.boardService.insertBoard(board);
    result.put("code", Boolean.TRUE);
    return result;
}
```

4)BoardServiceImpl.java 코드 추가

```
@Override
public void insertBoard(BoardVO board) {
    this.boardDao.create(board);
}
```

5)BoardDaoImpl.java code 추가

```
@Override
public void create(BoardVO board) {
    this.sqlSession.insert("Board.insertSP", board);
}
```

6)Database에 StoredProcedure 추가

```
CREATE OR REPLACE PROCEDURE sp_board_insert
(
    v_name          IN          board.name%TYPE,
    v_email         IN          board.email%TYPE,
    v_title         IN          board.title%TYPE,
    v_contents      IN          board.contents%TYPE
)
```

```
IS
BEGIN
    INSERT INTO Board(idx, name, email, title, contents, grp)
    VALUES(seq_board_idx.NEXTVAL, v_name, v_email, v_title, v_contents, seq_board_idx.CURRVAL);
    COMMIT;
END;
```

7)mybatis-config.xml 코드 추가

```
<typeAliases>
    <typeAlias type="com.javasoft.vo.BoardVO" alias="boardVO"/>
</typeAliases>
```

8)board-mapper.xml 코드 추가

```
<parameterMap type="boardVO" id="insertMap">
    <parameter property="name" mode="IN"/>
    <parameter property="email" mode="IN"/>
    <parameter property="title" mode="IN"/>
    <parameter property="contents" mode="IN"/>
</parameterMap>
<insert id="insertSP" parameterMap="insertMap">
    { call sp_board_insert(?, ?, ?, ?) }
</insert>
```

9)project > right-click > Run As > Run on Server > Finish

9)POSTMAN 으로 테스트

```
POST  localhost:8080/SpringBoard/board/
Body
    raw > JSON
        {
            "name" : "백두산",
            "email" : "javaexpert@nate.com",
            "title" : "첫 글입니다",
            "contents" : "첫 글 내용입니다."
        }
Send
-결과
    {
        "code": true
    }
```

10)페이지에서 [새 글쓰기] 클릭하여 write.jsp를 통해 입력이 성공하는지 테스트한다.

-Insert Success alert창이 나오고

-http://localhost:8080/SpringBoard/board/ 로 redirect 되면 성공이다.

-Database에 가서 입력된 것을 확인한다.

8. 목록보기

1)BoardController.java 코드 추가

```
@RequestMapping(value = "board/{page}", method = RequestMethod.GET)
@ResponseBody
public Map select(@PathVariable int page) {
    Map<String, Object> result = new HashMap<String, Object>();
    List<BoardVO> list = this.boardService.select();
    result.put("code", Boolean.TRUE);
    result.put("data", list);
    return result;
}
```

2)/WEB-INF/views/index.jsp code 추가

```
<script>
    $(document).ready(function(){
        $.ajax({
            url : 1,
            dataType : "json",
            method : "GET",
            success : function(result){
                display(result);
            }
        });
    });
    function display(result){
        var data = result.data;
        var count = data.length;
        $("#count").text(count);
        var str = null;
        if(count == 0) {
            str = "<tr>";
            str += "<td colspan='5' align='center'>";
            str += "글이 없습니다.</td>";
            str += "</tr>";
        }else if(count > 0){
            for(var i = 0 ; i < count ; i++){
                var board = data[i];
                var writedate = board.writedate.substr(0, 10);
                str += "<tr>";
                str += "<td style='text-align:center'>" + board.grp + "</td>" +
                        "<td style='text-align:center'><a href='mailto:" + board.email + "'>" + board.name + "</a></td>" +
                        "<td>&nbsp;" + "<a href='" + board.idx + "'>" + board.title + "</a></td>" +
                        "<td style='text-align:center'>" + writedate + "</td>" +
                        "<td style='text-align:center'>" + board.readnum + "</td>";
                str += "</tr>";
            }
        }
        $("#result").html(str);
    }
</script>
```

3)BoardServiceImpl.java 코드 추가

```
@Override
public List<BoardVO> select() {
    return this.boardDao.readAll();
}
```

4)BoardDaoImpl.java code 추가

```
@Override
public List<BoardVO> readAll() {
    Map<String, Object> map = new HashMap<String, Object>();
    this.sqlSession.selectList("Board.selectAllSP", map);
    List<BoardVO> list = (List<BoardVO>)map.get("results");
    return list;
    }
```

5)Database에 StoredProcedure 추가

```
CREATE OR REPLACE PROCEDURE sp_board_select
(
    v_bbs         OUT          SYS_REFCURSOR
)
AS
BEGIN
    OPEN v_bbs FOR
    SELECT idx, grp, name, email, title, writedate, readnum, lev, step
    FROM Board ORDER BY grp DESC, step ASC;
END;
```

6)board-mapper.xml 코드 추가

```
<resultMap type="boardVO" id="selectMap">
   <result property="idx" column="idx" />
   <result property="readnum" column="readnum" />
   <result property="grp" column="grp" />
   <result property="lev" column="lev" />
   <result property="step" column="step" />
   <result property="name" column="name" />
   <result property="email" column="email" />
   <result property="title" column="title" />
   <result property="contents" column="contents" />
   <result property="writedate" column="writedate"
      javaType="String" jdbcType="DATE" />
</resultMap>

<parameterMap type="hashMap" id="boardMap">
   <parameter property="results" javaType="ResultSet"
      jdbcType="CURSOR" mode="OUT" resultMap="selectMap" />
</parameterMap>
<select id="selectAllSP" statementType="CALLABLE"
   parameterMap="boardMap">
   { call sp_board_select(?) }
</select>
```

7)project > right-click > Run As > Run on Server > Finish

8)POSTMAN 으로 테스트

```
GET  localhost:8080/SpringBoard/board/1
Send
-결과
   {
       "code": true,
       "data": [
           {
               "idx": 2,
               "readnum": 0,
               "grp": 2,
               "lev": 0,
               "step": 0,
               "name": "한라산",
               "email": "javaexpert@nate.com",
               "title": "두번째 글",
               "contents": null,
               "writedate": "2018-06-13 09:53:16.0"
```

```
        },
        {
            "idx": 1,
            "readnum": 0,
            "grp": 1,
            "lev": 0,
            "step": 0,
            "name": "백두산",
            "email": "javaexpert@nate.com",
            "title": "첫 글입니다",
            "contents": null,
            "writedate": "2018-06-13 09:31:43.0"
        }
    ]
}
```

9)정상적으로 게시글 목록이 나오면 성공

9. 한 게시글 보기

1)BoardController.java 코드 추가

```
@RequestMapping(value = "board/{page}/{idx}", method = RequestMethod.GET)
public ModelAndView select(@PathVariable int page, @PathVariable int idx) {
    BoardVO board = this.boardService.selectBoard(idx);
    String contents = board.getContents();
    contents = contents.replaceAll("\n", "<br />");
    board.setContents(contents);
    ModelAndView mav = new ModelAndView();
    mav.addObject("code", Boolean.TRUE);
    mav.addObject("data", board);
    mav.addObject("page", page);
    mav.setViewName("view");   //   /WEB-INF/html/view.jsp
    return mav;
}
```

2)/WEB-INF/views/view.jsp 생성하기

```
<%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:set var="board" value="${data}" />
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Insert title here</title>
        <script src="<c:url value='/js/jquery-1.12.4.js'/>"></script>
        <script>
            $(document).ready(function(){
                $("#idx").text("${board.idx}");
                $("#writedate").text("${board.writedate}");
                $("#name").text("${board.name}");
                $("#readnum").text("${board.readnum}");
                $("#title").text("${board.title}");
                $("#contents").html("${board.contents}");
            });
        </script>
    </head>
    <body>
        <h1>게시판 글 확인</h1>
```

```
                <ul>
                    <li>Number : <span id="idx"></span></li>
                    <li>Date : <span id="writedate"></span></li>
                    <li>Writer : <span id="name"></span></li>
                    <li>Readnum : <span id="readnum"></span></li>
                    <li>Title : <span id="title"></span></li>
                    <li>Contents : <div id="contents" style="width:500px;height:100px;background-color:yellow"></div></li>
                </ul>
                <div>
                    <button onclick="javascript:location.href='/Board';">목록으로</button>
                    <button id="btnUpdate">수정하기</button>
                    <button id="btnDelete">삭제하기</button>
                    <button id="btnAnswer">답변하기</button>
                </div>
                <hr />
                <div>
                </div>
            </body>
        </html>
```

3)BoardServiceImpl.java 코드 추가

```
    @Override
    public BoardVO selectBoard(int idx) {
        return this.boardDao.read(idx);
    }
```

4)BoardDaoImpl.java code 추가

```
    @Override
    public BoardVO read(int idx) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put("idx", idx);
        this.sqlSession.selectOne("Board.selectByIdx", map);
        List<BoardVO> list = (List<BoardVO>)map.get("result");
        return list.get(0);
    }
```

5)Database에 StoredProcedure 추가

```
    CREATE OR REPLACE PROCEDURE sp_board_select_byidx
    (
         v_idx               IN            Board.idx%TYPE,
         v_record            OUT         SYS_REFCURSOR
    )
    AS
    BEGIN
         OPEN v_record FOR
         SELECT idx, grp, name, email, title, contents, writedate, readnum, lev, step
         FROM Board WHERE idx = v_idx;

         UPDATE Board SET readnum = readnum + 1
         WHERE idx = v_idx;
         COMMIT;
    END;
```

6)board-mapper.xml 코드 추가

```
    <parameterMap type="hashMap" id="selectOneMap">
```

```
        <parameter property="idx" javaType="Integer"
            jdbcType="INTEGER" mode="IN"/>
        <parameter property="result" javaType="ResultSet"
            jdbcType="CURSOR" mode="OUT" resultMap="selectMap"/>
    </parameterMap>
    <select id="selectByIdx" parameterMap="selectOneMap"
        statementType="CALLABLE">
        { call sp_board_select_byidx(?, ?) }
    </select>
```

7)index.jsp 수정

```
    var page = 1;   //첫줄에 추가

    else if(count > 0){
        for(var i = 0 ; i < count ; i++){
            var board = data[i];
            var writedate = board.writedate.substr(0, 10);
            str += "<tr>";
            str += "<td style='text-align:center'>" + board.grp + "</td>" +
                     "<td style='text-align:center'><a href='mailto:" + board.email + "'>" +
                     board.name + "</a></td>" +
                     "<td>&nbsp;" + "<a href='" + page + "/" + board.idx + "'>" +
                     board.title + "</a></td>" +     //여기를 수정
                     "<td style='text-align:center'>" + writedate + "</td>" +
                     "<td style='text-align:center'>" + board.readnum + "</td>";
            str += "</tr>";
        }
    }
```

8)project > right-click > Run As > Run on Server > Finish

9)POSTMAN 으로 테스트

```
    GET  localhost:8080/SpringBoard/board/1/2
    Send
    -결과
        <!DOCTYPE html>
        <html>
            <head>
                <meta charset="UTF-8">
                <title>Insert title here</title>
                <script src="../js/jquery-1.12.4.js"></script>
                <script>
                  $(document).ready(function(){
                      $("#idx").text("2");
                      $("#writedate").text("2018-06-13 09:53:16.0");
                      $("#name").text("한라산");
                      $("#readnum").text("4");
                      $("#title").text("두번째 글");
                      $("#contents").html("두번째 글입니다.");
                  });
                </script>
            </head>
            <body>
                <h1>게시판 글 확인</h1>
                <ul>
                    <li>Number :
                        <span id="idx"></span>
                    </li>
                    <li>Date :
                        <span id="writedate"></span>
        </li>
        <li>Writer :
          <span id="name"></span>
        </li>
        <li>Readnum :
          <span id="readnum"></span>
        </li>
        <li>Title :
          <span id="title"></span>
        </li>
        <li>Contents :
          <div id="contents" style="width:500px;height:100px;background-color:yellow"></div>
        </li>
     </ul>
     <div>
        <button onclick="javascript:location.href='/SpringBoard/';">목록으로</button>
        <button id="btnUpdate">수정하기</button>
        <button id="btnDelete">삭제하기</button>
        <button id="btnAnswer">답변하기</button>
     </div>
     <hr />
     <div>
    </div>
  </body>
</html>
```

10)게시판 목록에서 특정 제목을 클릭해서 정상적으로 글 내용이 나오면 성공

10. 글 삭제하기

1)BoardController.java 코드 추가

```
@RequestMapping(value = "board/{page}/{idx}", method = RequestMethod.DELETE)
@ResponseBody
public Map delete(@PathVariable int idx, @PathVariable int page) {
    this.boardService.deleteBoard(idx);
    Map<String, Object> result = new HashMap<String, Object>();
    result.put("code", Boolean.TRUE);
    return result;
}
```

2)/WEB-INF/views/view.jsp 코드 수정하기

```
$("#btnDelete").bind("click", function(){
    $.ajax({
      url : ${board.idx},
      method : "DELETE",
      success : function(data){
        if(data.code){
          location.href = "/SpringBoard";
        }
      }
    });
  });
```

3)BoardServiceImpl.java 코드 추가

```
@Override
```

```
public void deleteBoard(int idx) {
    this.boardDao.delete(idx);
}
```

4)BoardDaoImpl.java code 추가

```
@Override
public void delete(int idx) {
    this.sqlSession.delete("Board.deleteSP", idx);
}
```

5)Database에 StoredProcedure 추가

```
CREATE OR REPLACE PROCEDURE sp_board_delete
(
     v_idx        IN       Board.idx%TYPE
)
IS
BEGIN
     DELETE FROM Board
     WHERE idx = v_idx;

     COMMIT;
END;
```

6)board-mapper.xml 코드 추가

```
<delete id="deleteSP" parameterType="Integer">
    { call sp_board_delete(#{idx}) }
</delete>
```

7)project > right-click > Run As > Run on Server > Finish

8)POSTMAN 으로 테스트

```
DELETE  localhost:8080/SpringBoard/board/1/1
Send
-결과
    {
         "code": true
    }
```

9)원하는 글 삭제 성공

11. 글 수정하기

1)BoardController.java 코드 추가

```
@RequestMapping(value = "board/{page}/{idx}", method = RequestMethod.PUT)
@ResponseBody
public Map update(@RequestBody BoardVO board) {
    this.boardService.updateBoard(board);
    Map<String, Object> result = new HashMap<String, Object>();
    result.put("code", Boolean.TRUE);
    return result;
}
```

2)/WEB-INF/views/view.jsp 코드 수정하기

```
    var flag = true;
        $("#btnUpdate").click(function(){
```

```
        if(flag){
            var title = $("#title").text();
            var str = "<input type='text' id='title' value='" + title + "' size='30' />";
            $("#title").replaceWith(str);
            var contents = $("#contents").text();
            str = "<textarea id='contents' cols='40' rows='5'>" + contents + "</textarea>";
            $("#contents").replaceWith(str);
            $("#btnUpdate").text("갱신하기");
            flag = false;
        }else if(!flag){
            $.ajax({
                url : "",
                method : "PUT",
                dataType : "json",
                contentType : "application/json;charset=utf-8",
                data : JSON.stringify({
                    "idx" : parseInt($("#idx").text()),
                    "title" : $('#title').val(),
                    "contents" : $('#contents').val()
                }),
                success : function(data){
                    if(data.code){
                        location.reload();
                    }
                }
            });
        }
    });
```

3)BoardServiceImpl.java 코드 추가

```
@Override
public void updateBoard(BoardVO board) {
    this.boardDao.update(board);
}
```

4)BoardDaoImpl.java code 추가

```
@Override
public void update(BoardVO board) {
    this.sqlSession.update("Board.updateSP", board);
}
```

5)Database에 StoredProcedure 추가

```
CREATE OR REPLACE PROCEDURE sp_board_update
(
     v_title          IN            Board.title%TYPE,
     v_contents   IN            Board.contents%TYPE,
     v_idx           IN             Board.idx%TYPE
)
IS
BEGIN
     UPDATE Board SET title = v_title, contents = v_contents
     WHERE idx = v_idx;

     COMMIT;
END;
```

6)board-mapper.xml 코드 추가

```
<parameterMap type="boardVO" id="updateMap">
    <parameter property="title" javaType="String" jdbcType="VARCHAR" />
    <parameter property="contents" javaType="String" jdbcType="VARCHAR" />
    <parameter property="idx" javaType="Integer" jdbcType="INTEGER" />
</parameterMap>
<update id="updateSP" parameterMap="updateMap" statementType="CALLABLE">
    { call sp_board_update(?, ?, ?) }
</update>
```

7)project > right-click > Run As > Run on Server > Finish

8)POSTMAN 으로 테스트

```
GET  localhost:8080/SpringBoard/board/1/3
Send
```

-결과에서 다음을 수정

```
{
    "idx" : 3,
    "title" : "첫 글입니다",
    "contents" : "글 내용을 수정합니다."
}
```

-다시 Send를 보내면

```
{
     "code": true
}
```

9)글 수정 성공

## 12. Pagination

1)우선 1000개의 글을 입력한다.

-아래와 같이 BoardController.java에 코드를 추가한다.

```
@RequestMapping(value = "board/test")
public void test() {
    for(int i = 1 ; i <= 1000 ; i++) {
        BoardVO board = new BoardVO();
        board.setName("한지민");
        board.setEmail("aaa@aaa.com");
        board.setTitle(i + "번째 글입니다.");
        board.setContents(i + "번째 글입니다.");
        this.boardService.insertBoard(board);
    }
}
```

2)project > right-click > Run As > Run on Server > Finish

3)index.jsp 코드 추가 및 수정

```
    var page = null;
    $(document).ready(function(){
        $.ajax({
            url : 1,
            dataType : "json",
            method : "GET",
            success : function(result){
                display(result);
            }
        });
    });
    function goPage(pge){  //page 이동
        page = pge;
        $.ajax({
            url : page,
            dataType : "json",
            method : "GET",
            success : function(result){
                display(result);
            }
        });
    }

```

-display() 함수 코드 수정

```
    function display(result){
            var startPage = result.startPage;
            var endPage = result.endPage;
            var totalPage = result.totalPage;
            var pageSize = result.pageSize;
            page = result.page;
            var data = result.data;
            var count = data.length;
            $("#count").text(count);
            var str = null;
            if(count == 0) {
                str = "<tr>";
                str += "<td colspan='5' align='center'>";
                str += "글이 없습니다.</td>";
                str += "</tr>";
            }else if(count > 0){
                for(var i = 0 ; i < count ; i++){
                    var board = data[i];
                    var email = board.email;
                    var lev = board.lev;
                    var writedate = board.writedate.substr(0, 10);
                    var grp = null;
                    var title = "";
                    str += "<tr>";
                    if(lev > 0){
                        grp = "&nbsp;";
                        title = "<img src='../images/reply.jpg' width='20' height='20' />";
                    }else grp = board.grp;
                    for(var j = 0 ; j < lev ; j++){
                        title = "&nbsp;&nbsp;&nbsp;" + title;
                    }
                    str += "<td style='text-align:center'>" + grp + "</td>" +
                             "<td style='text-align:center'><a href='mailto:" + email + "'>" + board.name + "</a></td>" +
                             "<td>&nbsp;" + title + "<a href='" + page + "/" + board.idx + "'>" + board.title + "</a></td>" +
                             "<td style='text-align:center'>" + writedate + "</td>" +
                             "<td style='text-align:center'>" + board.readnum + "</td>";
                    str += "</tr>";
                }
            }
            $("#result").html(str);
            $("#paging").empty();
            $("#paging").attr("style", "text-align:center");

            var pageStr = null;
```

```
                if(page == 1) {
                    pageStr = "&lt;&lt;&nbsp;&nbsp;";
                    pageStr += "&lt;&nbsp;&nbsp;";
                }else if(page != 1) {
                    pageStr = "<a href='javascript:goPage(1)'>&lt;&lt;</a>&nbsp;&nbsp;";
                    pageStr += "<a href='javascript:goPage(" + (startPage - pageSize) + ")'>&lt;</a>&nbsp;&nbsp;";
                }

                for(var k = startPage ; k <= endPage ; k++){
                    if(page == k) pageStr += "<span style='color:red;font-weight:bold'>" + k + "</span>&nbsp;&nbsp;&nbsp;";
                    else pageStr += "<a href='javascript:goPage(" + k + ")'>" + k + "</a>&nbsp;&nbsp;&nbsp;";
                }

                if(page == totalPage) {
                    pageStr += "&gt;</a>&nbsp;&nbsp;";
                    pageStr += "&gt;&gt;";
                }else if(page != totalPage){
                    pageStr += "<a href='javascript:goPage(" + (startPage + pageSize) + ")'>&gt;</a>&nbsp;&nbsp;";
                    pageStr += "<a href='javascript:goPage(" + totalPage + ")'>&gt;&gt;</a>";
                }

                $("#paging").html(pageStr);
            }
```

4)BoardController.java에서 아래 코드 수정

```
    @RequestMapping(value = "board/{page}", method = RequestMethod.GET)
    @ResponseBody
    public Map select(@PathVariable int page) {
        int totalCount = this.boardService.getTotalCount();
        int pageSize = 10;
        int totalPage = (totalCount % pageSize == 0) ? totalCount / pageSize : totalCount / pageSize + 1;
        if(totalPage < page) page = totalPage;

        int pageCount = 10;

        int start = (page - 1) * pageCount + 1;
        int end = start + pageCount - 1 ;

        int startPage = ((page - 1) / pageCount) * pageCount + 1;
        int endPage = startPage + pageCount - 1 ;

        if(endPage> totalPage) endPage = totalPage;
        Map<String, Object> result = new HashMap<String, Object>();
        List<BoardVO> list = this.boardService.select(start, end);
        result.put("code", Boolean.TRUE);
        result.put("data", list);
        result.put("startPage", startPage);
        result.put("endPage", endPage);
        result.put("totalPage", totalPage);
        result.put("pageSize", pageSize);
        result.put("page", page);
        return result;
    }
```

5)BoardService.java와 BoardServiceImpl.java 코드 스정

-BoardService.java

```
List<BoardVO> select(int startPage, int endPage);
```

-BoardServiceImpl.java

```
@Override
public List<BoardVO> select(int startPage, int endPage) {
    return this.boardDao.readAll(startPage, endPage);
}
```

6)BoardServiceImpl.java 코드 수정

```
@Override
public int getTotalCount() {
    return this.boardDao.getTotalCount();
}
```

7)BoardDao.java와 BoardDaoImpl.java code 수정

-BoardDao.java

```
List<BoardVO> readAll(int startPage, int endPage);
```

-BoardDaoImpl.java

```
@Override
public List<BoardVO> readAll(int startPage, int endPage) {
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("startPage", startPage);
    map.put("endPage", endPage);
    this.sqlSession.selectList("Board.selectAllSP", map);
    List<BoardVO> list = (List<BoardVO>)map.get("results");
    return list;
}
```

8)BoardDaoImpl.java 코드 수정

```
@Override
public int getTotalCount() {
    return (Integer)this.sqlSession.selectOne("Board.selectCount");
}
```

9)board-mapper.xml 코드 수정

```
<parameterMap type="hashMap" id="boardMap">
    <parameter property="results" javaType="ResultSet"
        jdbcType="CURSOR" mode="OUT" resultMap="selectMap"/>
    <parameter property="startPage" javaType="Integer"
        jdbcType="INTEGER" mode="IN" />
    <parameter property="endPage" javaType="Integer"
        jdbcType="INTEGER" mode="IN" />
</parameterMap>
<select id="selectAllSP" statementType="CALLABLE"
    parameterMap="boardMap">
    { call sp_board_select(?, ?, ?) }
</select>
```

10)board-mapper.xml 코드 추가

```
<select id="selectCount" resultType="Integer">
    SELECT COUNT(*) FROM Board
</select>
```

11)Database Stored Procedure 수정

```
CREATE OR REPLACE PROCEDURE sp_board_select
(
    v_bbs        OUT         SYS_REFCURSOR,
    v_start      IN           NUMBER,
    v_end        IN           NUMBER
)
AS
BEGIN
    OPEN v_bbs FOR
    SELECT B.idx, grp, name, email, title, writedate, readnum, lev, step
    FROM
        (SELECT rownum AS rnum, A.idx, A.grp, A.name, A.email, A.title, A.writedate, A.readnum, A.lev, A.step
        FROM
            (SELECT idx, grp, name, email, title, writedate, readnum, lev, step
            FROM Board ORDER BY grp DESC, step ASC) A
        WHERE rownum <= v_end) B
    WHERE B.rnum >= v_start;
END;
```

12)project > right-click > Run As > Run on Server > Finish

13. 답글달기

1)BoardController.java 코드 추가

```
@RequestMapping(value = "board/{page}/{idx}", method = RequestMethod.POST)
@ResponseBody
public Map answer(@RequestBody BoardVO board) {
    board.setLev(board.getLev() + 1);
    board.setStep(board.getStep() + 1);
    this.boardService.answerBoard(board);
    Map<String, Object> result = new HashMap<String, Object>();
    result.put("code", Boolean.TRUE);
    return result;
}
```

2)/WEB-INF/views/view.jsp 코드 추가하기

```
    var answer = true;
        $("#btnAnswer").click(function(){
            if(answer){
                $("#name").replaceWith("<input type='text' id='name' />");
                $("#name").after("<li>Email : <input type='text' id='email'></li>");
                var title = $("#title").text();
                var str = "<input type='text' id='title' value='" + title + "' size='30' />";
                $("#title").replaceWith(str);
                var contents = $("#contents").text();
                contents = "\n\n\n\n\n-----------------------------" + contents;
                str = "<textarea id='contents' cols='40' rows='5'>" + contents + "</textarea>";
                $("#contents").replaceWith(str);
                $("#btnAnswer").text("답변완료");
                answer = false;
            }else if(!answer){
                $.ajax({
                    url : "",
                    method : "POST",
                    dataType : "json",
                    contentType : "application/json;charset=utf-8",
                    data : JSON.stringify({
                        "idx" : parseInt($("#idx").text()),
                        "name" : $("#name").val(),
                        "email" : $("#email").val(),
                        "title" : $('#title').val(),
                        "contents" : $('#contents').val(),
                        "grp" : parseInt("${board.grp}"),
                        "lev" : parseInt("${board.lev}"),
                        "step" : parseInt("${board.step}")
                    }),
                    success : function(data){
                        if(data.code){
                            location.href = "/Board";
                        }
                    }
                });
            }
        });
```

3)BoardServiceImpl.java 코드 추가

```
@Override
public void answerBoard(BoardVO board) {
    this.boardDao.answer(board);
}
```

4)BoardDaoImpl.java code 추가

```
@Override
public void answer(BoardVO board) {
    this.answer_update(board.getGrp(), board.getStep());
    this.sqlSession.insert("Board.answerInsertSP", board);
}

private void answer_update(int grp, int step) {
    Map<String, Integer> map = new HashMap<String, Integer>();
    map.put("grp", grp);
    map.put("step", step);
    this.sqlSession.update("Board.answerUpdateSP", map);
}
```

5)Database에 StoredProcedure 추가

```
CREATE OR REPLACE PROCEDURE sp_board_answer_update
(
     v_grp            IN             board.grp%TYPE,
     v_step            IN             board.step%TYPE
)
IS
BEGIN
     UPDATE Board SET step = step + 1
     WHERE grp = v_grp AND step >= v_step;
     COMMIT;
END;
/

CREATE OR REPLACE PROCEDURE sp_board_answer_insert
(
     v_name           IN     Board.name%TYPE,
     v_email          IN     Board.email%TYPE,
     v_title          IN     Board.title%TYPE,
     v_contents       IN     Board.contents%TYPE,
     v_grp            IN     Board.grp%TYPE,
     v_lev            IN     Board.lev%TYPE,
     v_step           IN     Board.step%TYPE
)
IS
BEGIN
     INSERT INTO Board(idx, name, email, title, contents, grp, lev, step)
     VALUES(seq_board_idx.NEXTVAL, v_name, v_email, v_title, v_contents, v_grp, v_lev, v_step);
     COMMIT;
END;
/
```

6)board-mapper.xml 코드 추가

```
    <parameterMap type="boardVO" id="answerMap">
        <parameter property="name" mode="IN" />
        <parameter property="email" mode="IN" />
        <parameter property="title" mode="IN" />
        <parameter property="contents" mode="IN" />
        <parameter property="grp" mode="IN" />
        <parameter property="lev" mode="IN" />
        <parameter property="step" mode="IN" />
    </parameterMap>
    <insert id="answerInsertSP" parameterMap="answerMap">
        { call sp_board_answer_insert(?,?,?,?,?,?,?) }
    </insert>
    <update id="answerUpdateSP" parameterType="hashMap">
        { call sp_board_answer_update(#{grp}, #{step}) }
    </update>
```

7)project > right-click > Run As > Run on Server > Finish

## 14. 댓글달기

1)댓글 정책 결정하기

| Method | URI | 설명 |
|---|---|---|
| GET | /reply/all/123 | 123번 게시글의 모든댓글 리스트 |
| POST | /reply/+데이터 | 새로운데이터추가 |
| PUT/PATCH | /reply/200 + 데이터 | 200 댓글수정 |
| DELETE | /reply/200 | 200 댓글삭제 |

2)댓글을 위한 테이블 생성

-하나의 게시글은 여러개의 댓글을 가지는 1:N관계

```
    CREATE SEQUENCE seq_reply_idx
    MAXVALUE 9999
```

-댓글 테이블을 생성하고, 댓글 참조키를 설정한다.

```
    CREATE TABLE Reply
    (
        idx         NUMBER(4),
        parent      NUMBER(4)   DEFAULT 0,    //FK는 null이 가능하기 때문에
        writer      VARCHAR2(20)    NOT NULL,
        text    VARCHAR2(2000)  NOT NULL,
        regdate   DATE DEFAULT SYSDATE,
        updatedate   DATE  DEFAULT SYSDATE,
        CONSTRAINT reply_idx_pk   PRIMARY KEY(idx),
        CONSTRAINT reply_idx_fk    FOREIGN KEY(parent) REFERENCES Board(idx)
    )
```

3)댓글을 위한 VO 객체 생성

-com.javasoft.vo.ReplyVO.java

```
package com.javasoft.vo;

public class ReplyVO {
    private int idx;
    private int parent;
    private String writer;
    private String text;
    private String regdate;
    private String updatedate;
    public ReplyVO() {}
    public ReplyVO(int idx, int parent, String writer, String text, String regdate, String updatedate) {
        this.idx = idx;
        this.parent = parent;
        this.writer = writer;
        this.text = text;
        this.regdate = regdate;
        this.updatedate = updatedate;
    }
    public int getIdx() {
        return idx;
    }
    public void setIdx(int idx) {
        this.idx = idx;
    }
    public int getParent() {
        return parent;
    }
    public void setParent(int parent) {
        this.parent = parent;
    }
    public String getWriter() {
        return writer;
    }
    public void setWriter(String writer) {
        this.writer = writer;
    }
    public String getText() {
        return text;
    }
    public void setText(String text) {
        this.text = text;
    }
    public String getRegdate() {
        return regdate;
    }
    public void setRegdate(String regdate) {
        this.regdate = regdate;
    }
    public String getUpdatedate() {
        return updatedate;
    }
    public void setUpdatedate(String updatedate) {
        this.updatedate = updatedate;
    }
    @Override
    public String toString() {
        return "ReplyVO [idx=" + idx + ", parent=" + parent + ", writer=" + writer + ", text=" + text + ", regdate="
                + regdate + ", updatedate=" + updatedate + "]";
    }
}
```

4)ReplyDao interface와 ReplyDaoImpl.java 생성

-com.javasoft.dao.ReplyDao.java

```
package com.javasoft.dao;

import java.util.List;

import com.javasoft.vo.ReplyVO;

public interface ReplyDao {
    List<ReplyVO> list(int idx);
    void create(ReplyVO replyVO);
    void update(ReplyVO replyVO);
    void delete(int idx);
}
```

-com.javasoft.dao.ReplyDaoImpl.java

```
package com.javasoft.dao;

import java.util.List;

import com.javasoft.vo.ReplyVO;

@Repository("replyDao")
public class ReplyDaoImpl implements ReplyDao {
    @Autowired
    private SqlSession sqlSession;

    @Override
    public List<ReplyVO> list(int idx) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void create(ReplyVO replyVO) {
        // TODO Auto-generated method stub

    }

    @Override
    public void update(ReplyVO replyVO) {
        // TODO Auto-generated method stub

    }

    @Override
    public void delete(int idx) {
        // TODO Auto-generated method stub

    }
}
```

5)config/mybatis-config.xml 코드 추가

```
<typeAliases>
    <typeAlias type="com.javasoft.vo.BoardVO" alias="boardVO"/>
    <typeAlias type="com.javasoft.vo.ReplyVO" alias="replyVO"/>   <--추가
</typeAliases>
```

6)config/reply-mapper.xml 추가

```
<?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="Reply">

</mapper>
```

7)config/applicationContext.xml 코드 추가

```
<property name="mapperLocations">
    <list>
        <value>classpath:board-mapper.xml</value>
        <value>classpath:reply-mapper.xml</value>
    </list>
</property>
```

8)ReplyService interface와 ReplyServiceImpl.java 생성

-com.javasoft.service.ReplyService.java

```
package com.javasoft.service;

import java.util.List;

import com.javasoft.vo.ReplyVO;

public interface ReplyService {
    List<ReplyVO> selectReply(int idx);
    void insertReply(ReplyVO replyVO);
    void updateReply(ReplyVO replyVO);
    void deleteReply(int idx);
}
```

-com.javasoft.service.ReplyServiceImpl.java

```
package com.javasoft.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.javasoft.dao.ReplyDao;
import com.javasoft.vo.ReplyVO;

```

```
@Service("replyService")
public class ReplyServiceImpl implements ReplyService {
    @Autowired
    private ReplyDao replyDao;

    @Override
    public List<ReplyVO> selectReply(int idx) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void insertReply(ReplyVO replyVO) {
        // TODO Auto-generated method stub

    }

    @Override
    public void updateReply(ReplyVO replyVO) {
        // TODO Auto-generated method stub

    }

    @Override
    public void deleteReply(int idx) {
        // TODO Auto-generated method stub

    }

}
```

9)com.javasoft.controller.ReplyController.java 생성

```
package com.javasoft.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

import com.javasoft.service.ReplyService;

@Controller
public class ReplyController {
    @Autowired
    private ReplyService replyService;
}
```

10)댓글 입력하기

-Database에 StoredProcedure 생성하기

```
CREATE OR REPLACE PROCEDURE sp_reply_insert
(
     v_parent     IN      Reply.parent%TYPE,
     v_writer     IN      Reply.writer%TYPE,
     v_text       IN      Reply.text%TYPE
)
IS
BEGIN
     INSERT INTO Reply (idx, parent, writer, text)
     VALUES (seq_reply_idx.NEXTVAL, v_parent, v_writer, v_text);
     COMMIT;
```

```
END;
```

-ReplyController.java 코드 추가

```
package com.javasoft.controller;

import java.util.HashMap;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import com.javasoft.service.ReplyService;
import com.javasoft.vo.ReplyVO;

@Controller
public class ReplyController {
    @Autowired
    private ReplyService replyService;

    @RequestMapping(value = "/reply", method = RequestMethod.POST)
    @ResponseBody
    public Map register(@RequestBody ReplyVO replyVO) {
        Map<String, Object> map = new HashMap<String, Object>();
        this.replyService.insertReply(replyVO);
        map.put("code", Boolean.TRUE);
        return map;
    }
}
```

-view.jsp 코드 추가

--body

```
<hr />
<div>
    <img src="<c:url value='/images/memo.png' />" width="40" height="40" />
    <span style='font-size: 2em'>댓글 작성</span>
</div>
<form>
    <table>
        <tr>
            <td rowspan="2"><textarea id="newReplyText" name="text"
                    rows="3" cols="50" placeholder="댓글 내용을 입력해주세요"></textarea></td>
            <td><input id="newReplyWriter" name="writer"
                placeholder="댓글작성자..." size="10"></td>
        </tr>
        <tr>
            <td><input type="button" id="replyWrite" value="저장"
                style="width: 100px; background-color: skyblue; color: white; font-weight: 700" />
            </td>
        </tr>
    </table>
</form>
<div>
```

```
        <ul id="replies" style="list-style-type: none">
        </ul>
    </div>
```

--script

```
var idx = null;    <---코드 추가
$(document).ready(function(){
    $("#idx").text("${board.idx}");
    idx = parseInt(${board.idx});        <---코드 추가
    $("#writedate").text("${board.writedate}");
    $("#name").text("${board.name}");

    ...
    ...

$("#replyWrite").click(function(){
    // 화면으로부터 입력 받은 변수값의 처리
    var replyText = $("#newReplyText");
    var replyWriter = $("#newReplyWriter");

    var replyTextVal = replyText.val();
    var replyWriterVal = replyWriter.val();

    $.ajax({
        method : "POST",
        url : "/SpringBoard/reply",
        contentType : "application/json;charset=utf-8",
        dataType : "json",
        data : JSON.stringify({
            "parent" : idx,
            "text" : replyTextVal,
            "writer" : replyWriterVal
        }),
        success : function (result) {
            if(data.code){
                alert("Insert Success");
            }
            //getReplies(); // 댓글 목록 출력 함수 호출
            replyText.val(""); // 댓글 내용 초기화
            replyWriter.val(""); // 댓글 작성자 초기화
        }
    });
});
```

-reply-mapper.xml 코드 추가

```
<parameterMap type="replyVO" id="replyInsertMap">
    <parameter property="parent" javaType="Integer"
        jdbcType="INTEGER" mode="IN" />
    <parameter property="writer" javaType="String"
        jdbcType="VARCHAR" mode="IN" />
    <parameter property="text" javaType="String"
        jdbcType="VARCHAR" mode="IN" />
</parameterMap>
<insert id="create" parameterMap="replyInsertMap">
    { call sp_reply_insert (?, ?, ?) }
</insert>
```

-Insert를 위한 ReplyDaoImpl.java 수정

```
@Override
public void create(ReplyVO replyVO) {
    this.sqlSession.insert("Reply.create", replyVO);
}
```

-Insert를 위한 ReplyServiceImpl.java 수정

```
@Override
public void insertReply(ReplyVO replyVO) {
    this.replyDao.create(replyVO);
}
```

11)댓글 목록 보기

-Database에 StoredProcedure 생성하기

```
CREATE OR REPLACE PROCEDURE sp_reply_select
(
    v_parent        IN          Reply.parent%TYPE,
    v_reply         OUT         SYS_REFCURSOR
)
IS
BEGIN
    OPEN v_reply FOR
    SELECT idx, parent, writer, text, regdate, updatedate
    FROM Reply
    WHERE parent = v_parent
    ORDER BY idx;
END;
```

-ReplyController.java 코드 추가

```
@RequestMapping(value = "/reply/all/{parent}", method = RequestMethod.GET)
@ResponseBody
    public Map list(@PathVariable int parent) {
        Map<String, Object> map = new HashMap<String, Object>();
        List<ReplyVO> list = this.replyService.selectReply(parent);
        map.put("code", Boolean.TRUE);
        map.put("data", list);
        return map;
    }
}
```

-view.jsp 코드 추가

--script

```
$(document).ready(function(){
    $("#idx").text("${board.idx}");
    idx = parseInt("${board.idx}");
    getReplies();               <--코드 추가

    ...
    ...
    success : function (result) {
        if(result.code){
            alert("Insert Success");
        }
        getReplies();   //주석 해제
    ...
    ...
```

```
        function getReplies() {
        $.getJSON("/SpringBoard/reply/all/" + idx, function (result) {
            var data = result.data;
            var str = "";
            $(data).each(function () {
                str += "<li class='replyLi' id='" + this.idx + "'>"
                    +   "<p class='text'>" + this.text + "</p>"
                    +   "<p class='writer'>" + this.writer + "</p>"
                    +   "<button type='button'  class='update'
                    style='background-color:green;color:white;font-weight:bold;'>댓글
                    수정</button>"
                    +   "<button type='button'  class='delete'
                    style='background-color:red;color:white;font-weight:bold;'>댓글
                    삭제</button>"
                    + "</li>"
                    + "<hr/>";
            });
            $("#replies").html(str);
        });
    }
```

-reply-mapper.xml 코드 추가

```
<resultMap id="replyResultMap" type="replyVO">
    <id property="idx" column="idx" />
    <result property="parent" column="parent" />
    <result property="writer" column="writer" />
    <result property="text" column="text" />
    <result property="regdate" column="regdate" />
    <result property="updatedate" column="updatedate" />
</resultMap>

<parameterMap type="hashMap" id="replySelectMap">
    <parameter property="parent" mode="IN"
        javaType="Integer" jdbcType="INTEGER"/>
    <parameter property="results" mode="OUT"
        javaType="ResultSet" jdbcType="CURSOR" resultMap="replyResultMap"/>
</parameterMap>
<select id="list" parameterMap="replySelectMap" statementType="CALLABLE">
    { call sp_reply_select(?, ?) }
</select>
```

-Select를 위한 ReplyDaoImpl.java 수정

```
@Override
public List<ReplyVO> list(int parent) {
    Map<String, Object> result = new HashMap<String, Object>();
    result.put("parent", parent);
    this.sqlSession.selectList("Reply.list", result);
    List<ReplyVO> list = (List<ReplyVO>)result.get("results");
    return list;
}
```

-Select를 위한 ReplyServiceImpl.java 수정

```
@Override
public List<ReplyVO> selectReply(int parent) {
    return this.replyDao.list(parent);
}
```

12)댓글 수정하기

-view.jsp 코드 추가

--jQuery코드에서 주목해서 봐야할 것은 클릭 이벤트 선택자가 <ul>의 id인 replies로 되어있다는 점이다.

--단순히 클릭이 발생한 수정버튼을 선택자로 지정하면 될 것 같지만 Ajax통신 후에 생기는 요소이기 때문에 이벤트처리가 되지 않는다.

--그래서 Ajax통신 이전에 존재하는 <ul>을 이용해서 이벤트를 등록해야만 한다.

```
var replyIdx = null;
var replyUpdateFlag = true;
$("#replies").on("click", ".replyLi button", function () {
    if(replyUpdateFlag){
        var replyLi = $(this).parent();

        replyIdx = replyLi.attr("id");
        var replyText = replyLi.find(".text").text();
        var replyWriter = replyLi.find(".writer").text();

        var replyStr = "<textarea id='updateReplyText' rows='3' cols='60'>" +
        replyText + "</textarea>";
        replyLi.find(".text").replaceWith(replyStr);
        $(this).text("댓글 수정 완료");
        replyUpdateFlag = false;
    }else{
        var replyText = $(this).parent().find("#updateReplyText").val();
        $(this).text("댓글 수정");
        $.ajax({
            url : "/SpringBoard/reply",
            method : "PUT",
            dataType : "json",
            contentType : "application/json;charset=utf-8",
            data : JSON.stringify({
                "idx" : parseInt(replyIdx),
                "text" : replyText
            }),
            success : function(result){
                getReplies();
                replyUpdateFlag = true;
            }
        });
    }
});
```

-Database StoredProcedure 추가하기

```
CREATE OR REPLACE PROCEDURE sp_reply_update
(
    v_text      IN      Reply.text%TYPE,
    v_idx       IN      Reply.idx%TYPE
)
IS
BEGIN
    UPDATE Reply SET text = v_text, updatedate = SYSDATE
    WHERE idx = v_idx;
    COMMIT;
END;
```

-reply-mapper.xml 코드 추가

```
<parameterMap type="hashMap" id="replyUpdateMap">
```

```
        <parameter property="text" javaType="String"
            jdbcType="VARCHAR" mode="IN"/>
        <parameter property="idx" javaType="Integer"
            jdbcType="INTEGER" mode="IN"/>
    </parameterMap>
    <update id="update" parameterMap="replyUpdateMap">
        { call sp_reply_update(?, ?) }
    </update>
```

-ReplyDaoImpl.java 코드 수정

```
    @Override
    public void update(ReplyVO replyVO) {
        Map<String, Object> map = new HashMap<String, Object>();
        map.put("text", replyVO.getText());
        map.put("idx", replyVO.getIdx());
        this.sqlSession.update("Reply.update", map);
    }
```

-ReplyServiceImpl.java 코드 수정

```
    @Override
    public void updateReply(ReplyVO replyVO) {
        this.replyDao.update(replyVO);
    }
```

-ReplyController.java 코드 추가

```
    @RequestMapping(value = "/reply", method = RequestMethod.PUT)
    @ResponseBody
    public Map update(@RequestBody ReplyVO replyVO) {
        Map<String, Object> map = new HashMap<String, Object>();
        this.replyService.updateReply(replyVO);
        System.out.println(replyVO);
        map.put("code", Boolean.TRUE);
        return map;
    }
```

13)댓글 삭제하기

-view.jsp 코드 추가

```
    $("#replies").on("click", ".replyLi .delete", function () {
        var replyIdx = $(this).parent().attr("id");
        $.ajax({
            url : "/SpringBoard/reply/" + parseInt(replyIdx),
            method : "DELETE",
            dataType : "json",
            success : function(result){
                getReplies();
            }
        });
    });
```

-Database StoredProcedure 추가하기

```
    CREATE OR REPLACE PROCEDURE sp_reply_delete
    (
        v_idx   IN   Reply.idx%TYPE
    )
    IS
    BEGIN
        DELETE FROM Reply WHERE idx = v_idx;
        COMMIT;
    END;
```

-reply-mapper.xml 코드 추가

```
<delete id="delete" parameterType="Integer">
    { call sp_reply_delete(#{idx}) }
</delete>
```

-ReplyDaoImpl.java 코드 수정

```
@Override
public void delete(int idx) {
    this.sqlSession.delete("Reply.delete", idx);
}
```

-ReplyServiceImpl.java 코드 수정

```
@Override
public void deleteReply(int idx) {
    this.replyDao.delete(idx);
}
```

-ReplyController.java 코드 추가

```
@RequestMapping(value = "/reply/{idx}", method = RequestMethod.DELETE)
@ResponseBody
public Map delete(@PathVariable int idx) {
    Map<String, Object> map = new HashMap<String, Object>();
    this.replyService.deleteReply(idx);
    map.put("code", Boolean.TRUE);
    return map;
}
```