

1. Database Configuration

1) 계정 생성

```
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.xml 파일에 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>
```

```

60         <version>4.3.17.RELEASE</version>
61     </dependency>
62     <dependency>
63         <groupId>junit</groupId>
64         <artifactId>junit</artifactId>
65         <version>4.12</version>
66         <scope>test</scope>
67     </dependency>
68     <dependency>
69         <groupId>com.oracle</groupId>
70         <artifactId>ojdbc8</artifactId>
71         <version>12.2</version>
72     </dependency>
73     <dependency>
74         <groupId>org.mybatis</groupId>
75         <artifactId>mybatis</artifactId>
76         <version>3.4.6</version>
77     </dependency>
78     <dependency>
79         <groupId>org.mybatis</groupId>
80         <artifactId>mybatis-spring</artifactId>
81         <version>1.3.2</version>
82     </dependency>
83     <dependency>
84         <groupId>javax.servlet</groupId>
85         <artifactId>jstl</artifactId>
86         <version>1.2</version>
87     </dependency>
88     <dependency>
89         <groupId>org.springframework</groupId>
90         <artifactId>spring-webmvc</artifactId>
91         <version>4.3.17.RELEASE</version>
92     </dependency>
93     <dependency>
94         <groupId>com.fasterxml.jackson.core</groupId>
95         <artifactId>jackson-databind</artifactId>
96         <version>2.9.5</version>
97     </dependency>
98     <dependency>
99         <groupId>com.fasterxml.jackson.core</groupId>
100        <artifactId>jackson-core</artifactId>
101        <version>2.9.5</version>
102    </dependency>
103    <dependency>
104        <groupId>com.fasterxml.jackson.core</groupId>
105        <artifactId>jackson-annotations</artifactId>
106        <version>2.9.5</version>
107    </dependency>
108    <dependency>
109        <groupId>org.springframework</groupId>
110        <artifactId>spring-test</artifactId>
111        <version>4.3.17.RELEASE</version>
112        <scope>test</scope>
113    </dependency>
114</dependencies>
115

```

7) Build path에 config folder 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">
175     <display-name>SpringBoard</display-name>
176     <welcome-file-list>
177         <welcome-file>index.html</welcome-file>
178     </welcome-file-list>
179
180
181 </web-app>
182
183 -Ctrl + Spacebar를 하면 나타나는 Context Menu에서 [#contextloaderlistener -
ContextLoaderListener] 를 선택하면 아래의 코드가 자동 삽입
184
185 <!-- needed for ContextLoaderListener -->
186 <context-param>
187     <param-name>contextConfigLocation</param-name>
188     <param-value>location</param-value>
189 </context-param>
190
191 <!-- Bootstraps the root web application context before servlet initialization -->
192 <listener>
193
194     <listener-class>org.springframework.web.context.ContextLoaderListener</listene
r-class>
195 </listener>
196
197 --아래 코드로 변환
198 <context-param>
199     <param-name>contextConfigLocation</param-name>
200     <param-value>classpath:applicationContext.xml</param-value>
201 </context-param>
202
203 3)DispatcherServlet Class 추가
204 -web.xml에서 Ctrl + Spacebar 하면 나타나는 Context Menu에서 [#dispatcherservlet -
DispatcherServlet declaration] 선택하면 아래의 코드가 자동 추가된다.
205
206 <!-- The front controller of this Spring Web application, responsible for handling all
application requests -->
207 <servlet>
208     <servlet-name>springDispatcherServlet</servlet-name>
209
210     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
211
212     <init-param>
213         <param-name>contextConfigLocation</param-name>
214         <param-value>location</param-value>
215     </init-param>
216     <load-on-startup>1</load-on-startup>
217 </servlet>
218
219 <!-- Map all requests to the DispatcherServlet for handling -->
220 <servlet-mapping>
221     <servlet-name>springDispatcherServlet</servlet-name>
222     <url-pattern>url</url-pattern>
223 </servlet-mapping>
224
225 --아래의 코드로 변환
226 <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext*.xml</param-value>
</init-param>

```

```
227
228     <servlet-mapping>
229         <servlet-name>springDispatcherServlet</servlet-name>
230         <url-pattern>/</url-pattern>
231     </servlet-mapping>
232
233
```

234 5. index.html 생성 및 정적 페이지 처리하기

235 1)applicationContext.xml file 수정

```
236     <mvc:default-servlet-handler /> 추가
237
```

238 2)index.html 생성

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

240 -File name : index.html > Finish

```
241
242     <!DOCTYPE html>
243     <html>
244         <head>
245             <meta charset="UTF-8">
246             <title>Spring Board</title>
247         </head>
248         <body>
249             <h1>Spring Board</h1>
250         </body>
251     </html>
252
```

253 3)style.css file 생성

254 -WebContent > right-click > New > Folder

255 -Folder name : css > Finish

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

257 -File name : style.css > Finish

```
258
259     @charset "UTF-8";
260     body{
261         background-color: yellow;
262     }
263     h1 {
264         color : blue;
265     }
266
```

267 4)jQuery file 추가하기

268 -WebContent > right-click > New > Folder

269 -Folder name : js > Finish

270 -이미 download 받은 jquery-1.12.4.js 파일 넣기

271

272 5)index.html 수정

```
273
274     <!DOCTYPE html>
275     <html>
276         <head>
277             <meta charset="UTF-8">
278             <title>Spring Board</title>
279             <link rel="stylesheet" type="text/css" href="css/style.css" />
280             <script src="js/jquery-1.12.4.js"></script>
281             <script>
282                 $(document).ready(function(){
283                     alert("Hello Spring Board");
284                 });
285             </script>
286         </head>
```

```
287     <body>
288         <h1>Spring Board</h1>
289     </body>
290 </html>
```

291
292 6)Project > right-click > Run As > Run on Server > Finish

293
294 7)http://localhost:8080/SpringBoard/

295 296 297 6. Java Package 및 관련 Class file 생성

298 1)package 생성

```
299     -src/com.javasoft.vo
300     -src/com.javasoft.dao
301     -src/com.javasoft.service
302     -src/com.javasoft.controller
```

303 304 2)BoardVO class 생성

```
305     -src/com.javasoft.vo.UserVO class 생성
```

```
306  
307     package com.javasoft.vo;
```

```
308  
309     public class BoardVO {
310         private int idx, readnum, grp, lev, step;
311         private String name, email, title, contents;
312         private String writedate;
```

```
313  
314         public BoardVO() {}
```

```
315  
316         public BoardVO(int grp, String name, String email, int idx, String title, int lev, int
step, String writedate, int readnum) {
```

```
317             this.idx = idx;    this.readnum = readnum;
318             this.grp = grp;    this.lev = lev;
319             this.step = step;   this.name = name;
320             this.email = email; this.title = title;
321             this.writedate = writedate;
```

```
322     }
```

```
323  
324     public BoardVO(int grp, String name, String email, int idx, String title, int lev, int
step, String Contents, String writedate, int readnum) {
```

```
325         this.idx = idx;    this.readnum = readnum;
326         this.grp = grp;    this.lev = lev;
327         this.step = step;   this.name = name;
328         this.email = email; this.title = title;
329         this.contents = contents;
330         this.writedate = writedate;
```

```
331     }
```

```
332  
333     public int getIdx() {
334         return idx;
```

```
335     }
```

```
336     public void setIdx(int idx) {
```

```
337         this.idx = idx;
```

```
338     }
```

```
339     public int getReadnum() {
```

```
340         return readnum;
```

```
341     }
```

```
342     public void setReadnum(int readnum) {
```

```
343         this.readnum = readnum;
```

```
344     }
```

```

345     public int getGrp() {
346         return grp;
347     }
348     public void setGrp(int grp) {
349         this.grp = grp;
350     }
351     public int getLev() {
352         return lev;
353     }
354     public void setLev(int lev) {
355         this.lev = lev;
356     }
357     public int getStep() {
358         return step;
359     }
360     public void setStep(int step) {
361         this.step = step;
362     }
363     public String getName() {
364         return name;
365     }
366     public void setName(String name) {
367         this.name = name;
368     }
369     public String getEmail() {
370         return email;
371     }
372     public void setEmail(String email) {
373         this.email = email;
374     }
375     public String getTitle() {
376         return title;
377     }
378     public void setTitle(String title) {
379         this.title = title;
380     }
381     public String getContents() {
382         return contents;
383     }
384     public void setContents(String contents) {
385         this.contents = contents;
386     }
387     public String getWritedate() {
388         return writedate;
389     }
390     public void setWritedate(String writedate) {
391         this.writedate = writedate;
392     }
393     @Override
394     public String toString() {
395         return "BoardVO [idx=" + idx + ", readnum=" + readnum + ", grp=" + grp +
396             ", lev=" + lev + ", step=" + step
397             + ", name=" + name + ", email=" + email + ", title=" + title + ",
398             contents=" + contents + ", writedate="
399             + writedate + "]\n";
400     }
401 }

```

3)BoardService 생성

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

```
403
404     package com.javasoft.service;
405
406     import java.util.List;
407
408     import com.javasoft.vo.BoardVO;
409
410     public interface BoardService {
411         void insertBoard(BoardVO board);
412         BoardVO selectBoard(int idx);
413         List<BoardVO> select();
414         void updateBoard(BoardVO board);
415         void deleteBoard(int idx);
416         void answerBoard(BoardVO board);
417         int getTotalCount();
418     }
419
420 -src/com.javasoft.service.BoardServiceImpl class 생성
```

```
421
422     package com.javasoft.service;
423
424     import java.util.List;
425
426     import com.javasoft.vo.BoardVO;
427
428     public class BoardServiceImpl implements BoardService {
429
430         @Override
431         public void insertBoard(BoardVO board) {
432             // TODO Auto-generated method stub
433
434         }
435
436         @Override
437         public BoardVO selectBoard(int idx) {
438             // TODO Auto-generated method stub
439             return null;
440         }
441
442         @Override
443         public List<BoardVO> select() {
444             // TODO Auto-generated method stub
445             return null;
446         }
447
448         @Override
449         public void updateBoard(BoardVO board) {
450             // TODO Auto-generated method stub
451
452         }
453
454         @Override
455         public void deleteBoard(int idx) {
456             // TODO Auto-generated method stub
457
458         }
459
460         @Override
461         public void answerBoard(BoardVO board) {
462             // TODO Auto-generated method stub
```



```

463
464     }
465
466     @Override
467     public int getTotalCount() {
468         // TODO Auto-generated method stub
469         return 0;
470     }
471 }

```

4)BoardDao 객체 생성

-src/com.javasoft.dao.BoardDao interface

```

475
476     package com.javasoft.dao;
477
478     import java.util.List;
479
480     import com.javasoft.vo.BoardVO;
481
482     public interface BoardDao {
483         void create(BoardVO board);
484         BoardVO read(int idx);
485         List<BoardVO> readAll();
486         void update(BoardVO board);
487         void delete(int idx);
488         void answer(BoardVO board);
489         int getTotalCount();
490     }

```

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

```

493
494     package com.javasoft.dao;
495
496     import java.util.List;
497
498     import com.javasoft.vo.BoardVO;
499
500     public class BoardDaoImpl implements BoardDao {
501
502         @Override
503         public void create(BoardVO board) {
504             // TODO Auto-generated method stub
505
506         }
507
508         @Override
509         public BoardVO read(int idx) {
510             // TODO Auto-generated method stub
511             return null;
512         }
513
514         @Override
515         public List<BoardVO> readAll() {
516             // TODO Auto-generated method stub
517             return null;
518         }
519
520         @Override
521         public void update(BoardVO board) {
522             // TODO Auto-generated method stub

```

```

523
524     }
525
526     @Override
527     public void delete(int idx) {
528         // TODO Auto-generated method stub
529
530     }
531
532     @Override
533     public void answer(BoardVO board) {
534         // TODO Auto-generated method stub
535
536     }
537
538     @Override
539     public int getTotalCount() {
540         // TODO Auto-generated method stub
541         return 0;
542     }
543
544 }
545

```

5)BoardController 객체 생성

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

```

548
549     package com.javasoft.controller;
550
551     public class BoardController {
552
553     }
554

```

6)applicationContext.xml 수정

```

556
557     <mvc:annotation-driven />
558     <mvc:resources location="/views/**" mapping="/WEB-INF/views/" />
559
560     <context:component-scan base-package="com.javasoft" />
561     <context:property-placeholder location="classpath:dbinfo.properties"/>
562
563     <bean id="dataSource"
564         class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
565         <property name="driverClass" value="${db.driver}" />
566         <property name="url" value="${db.url}" />
567         <property name="username" value="${db.username}" />
568         <property name="password" value="${db.password}" />
569     </bean>
570
571     <bean id="sqlSessionFactory"
572         class="org.mybatis.spring.SqlSessionFactoryBean">
573         <property name="dataSource" ref="dataSource" />
574         <property name="configLocation"
575             value="classpath:mybatis-config.xml" />
576         <property name="mapperLocations">
577             <list>
578                 <value>classpath:board-mapper.xml</value>
579             </list>
580         </property>
581     </bean>
582

```

```

583 <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate">
584     <constructor-arg ref="sqlSessionFactory" name="sqlSessionFactory" />
585 </bean>
586
587 <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
588     <property name="prefix" value="/WEB-INF/views/" />
589     <property name="suffix" value=".jsp" />
590 </bean>

```

7)/WEB-INF/views folder 생성

```

592 -WEB-INF > right-click > New > Folder
593 -Folder name : views > Finish

```

8)WebContent > images folder 생성

```

596 -images folder에 아래와 같은 image 파일들 추가할 것
597 -banner.jpg
598 -board.png
599 -reply.jpg
600

```

9)/WEB-INF/views/index.jsp 생성

```

602 -WEB-INF > right-click > New > Other > Web > JSP File > Next
603 -File name : index.jsp > Finish

```

```

606 <%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
607 <!DOCTYPE html>
608 <html>
609     <head>
610         <meta charset="UTF-8">
611         <title>Spring Board</title>
612         <link rel="stylesheet" href="../css/style.css" type="text/css">
613         <script src="../js/jquery-1.12.4.js"></script>
614     </head>
615     <body>
616         
617         <div id="banner">
618             
619             <span>SpringBoard</span>
620         </div>
621         <div>
622             <table border="0">
623                 <tr>
624                     <td>[<a href="write">새 글쓰기</a>]
625                     <td>[<span id="count"></span>]
626                 </tr>
627             </table>
628         </div>
629         <div>
630             <table border="1">
631                 <colgroup>
632                     <col width="10%" />
633                     <col width="10%" />
634                     <col width="60%" />
635                     <col width="10%" />
636                     <col width="10%" />
637                 </colgroup>
638                 <thead>
639                     <tr>

```

```

641         <th>번호</th>
642         <th>작성자</th>
643         <th>제목</th>
644         <th>작성일</th>
645         <th>조회수</th>
646     </tr>
647 </thead>
648 <tbody id="result"></tbody>
649 </table>
650 </div>
651 <p></p>
652 <div id="paging"></div>
653 </body>
654 </html>

```

656 10)index.html 수정

```

657
658 <script>
659     document.location = "board/";
660 </script>

```

662 11)BoardController.java 수정

```

663
664 @Controller
665 public class BoardController {
666     @Autowired
667     private BoardService boardService;
668
669     @RequestMapping(value = "board/", method = RequestMethod.GET)
670     public String index() {
671         return "index";
672     }
673

```

674 12)BoardServiceImpl.java 수정

```

675
676 @Service("boardService")
677 public class BoardServiceImpl implements BoardService {
678     @Autowired
679     private BoardDao boardDao;
680 }
681

```

682 13)BoardDaoImpl.java 수정

```

683
684 @Repository("boardDao")
685 public class BoardDaoImpl implements BoardDao {
686     @Autowired
687     private SqlSession sqlSession;
688
689 }
690

```

691 14)style.css 수정

```

692
693 body {
694     margin-top : 0px;
695     margin-left : 0px;
696     margin-right:0px;
697 }
698 #banner{
699     text-align:center;
700 }

```

```

701     #banner span {
702         font-size:2em;
703     }
704     div:nth-of-type(2) table {
705         width:80%;
706         margin-left:auto; margin-right:auto;
707     }
708     div:nth-of-type(2) table td:nth-of-type(1) {
709         text-align:left;
710     }
711     div:nth-of-type(2) table td:nth-of-type(2) {
712         text-align:right;
713     }
714     div:nth-of-type(3) table {
715         width:80%;
716         margin-left:auto; margin-right:auto;
717         border-collapse: collapse;
718     }
719     div:nth-of-type(3) table th{
720         height:50px;
721         background-color: navy;
722         color:white;
723     }

```

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()
                    })
                });
            });
        });
    </script>

```

```

761         }},
762         success : function(data){
763             if(data.code){
764                 alert("Insert Success");
765             }
766             location.href = "/SpringBoard";
767         }
768     });
769 });
770 });
771 </script>
772 </head>
773 <body>
774     <h1>게시판 글 쓰기</h1>
775     <form>
776         <ul style="list-style-type:none">
777             <li>작성자 : <input type="text" id="name" /></li>
778             <li>Email : <input type="email" id="email" size="30" /></li>
779             <li>제목 : <input type="text" id="title" size="40"/></li>
780             <li>내용 : <textarea id="contents" rows="10" cols="40"></textarea></li>
781         </ul>
782         <input type="button" value="글 쓰기" />
783     </form>
784 </body>
785 </html>

```

787 3)BoardController.java 코드 추가

```

788
789 @RequestMapping(value = "board/", method = RequestMethod.POST)
790 @ResponseBody
791 public Map insert(@RequestBody BoardVO board) {
792     Map<String, Object> result = new HashMap<String, Object>();
793     this.boardService.insertBoard(board);
794     result.put("code", Boolean.TRUE);
795     return result;
796 }
797

```

798 4)BoardServiceImpl.java 코드 추가

```

799
800 @Override
801 public void insertBoard(BoardVO board) {
802     this.boardDao.create(board);
803 }
804

```

805 5)BoardDaoImpl.java code 추가

```

806
807 @Override
808 public void create(BoardVO board) {
809     this.sqlSession.insert("Board.insertSP", board);
810 }
811

```

812 6)Database에 StoredProcedure 추가

```

813
814 CREATE OR REPLACE PROCEDURE sp_board_insert
815 (
816     v_name          IN          board.name%TYPE,
817     v_email         IN          board.email%TYPE,
818     v_title         IN          board.title%TYPE,
819     v_contents      IN          board.contents%TYPE
820 )

```

```

821 IS
822 BEGIN
823     INSERT INTO Board(idx, name, email, title, contents, grp)
824     VALUES(seq_board_idx.NEXTVAL, v_name, v_email, v_title, v_contents,
825             seq_board_idx.CURRVAL);
826 COMMIT;
827 END;

```

7)mybatis-config.xml 코드 추가

```

829
830 <typeAliases>
831     <typeAlias type="com.javasoft.vo.BoardVO" alias="boardVO"/>
832 </typeAliases>
833

```

8)board-mapper.xml 코드 추가

```

834
835
836 <parameterMap type="boardVO" id="insertMap">
837     <parameter property="name" mode="IN"/>
838     <parameter property="email" mode="IN"/>
839     <parameter property="title" mode="IN"/>
840     <parameter property="contents" mode="IN"/>
841 </parameterMap>
842 <insert id="insertSP" parameterMap="insertMap">
843     { call sp_board_insert(?, ?, ?, ?) }
844 </insert>
845

```

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

9)POSTMAN 으로 테스트

```

848 POST localhost:8080/SpringBoard/board/
849 Body
850 raw > JSON
851 {
852     "name" : "백두산",
853     "email" : "javaexpert@nate.com",
854     "title" : "첫 글입니다",
855     "contents" : "첫 글 내용입니다."
856 }
857
858 Send
859 -결과
860 {
861     "code": true
862 }
863

```

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

```

871
872
873 @RequestMapping(value = "board/{page}", method = RequestMethod.GET)
874 @ResponseBody
875 public Map select(@PathVariable int page) {
876     Map<String, Object> result = new HashMap<String, Object>();
877     List<BoardVO> list = this.boardService.select();
878     result.put("code", Boolean.TRUE);
879     result.put("data", list);

```

```
880         return result;
881     }
882
```

2)/WEB-INF/views/index.jsp code 추가

```
885     <script>
886         $(document).ready(function(){
887             $.ajax({
888                 url : 1,
889                 dataType : "json",
890                 method : "GET",
891                 success : function(result){
892                     display(result);
893                 }
894             });
895         });
896         function display(result){
897             var data = result.data;
898             var count = data.length;
899             $("#count").text(count);
900             var str = null;
901             if(count == 0) {
902                 str = "<tr>";
903                 str += "<td colspan='5' align='center'>";
904                 str += "글이 없습니다.</td>";
905                 str += "</tr>";
906             }else if(count > 0){
907                 for(var i = 0 ; i < count ; i++){
908                     var board = data[i];
909                     var writedate = board.writedate.substr(0, 10);
910                     str += "<tr>";
911                     str += "<td style='text-align:center'>" + board.grp + "</td>" +
912                         "<td style='text-align:center'><a href='mailto:" + board.email +
913                             "'>" + board.name + "</a></td>" +
914                             "<td>&nbsp;" + "<a href='" + board.idx + "'>" + board.title +
915                             "</a></td>" +
916                             "<td style='text-align:center'>" + writedate + "</td>" +
917                             "<td style='text-align:center'>" + board.readnum + "</td>";
918                     str += "</tr>";
919                 }
920             }
921             $("#result").html(str);
922         }
923     </script>
```

3)BoardServiceImpl.java 코드 추가

```
924
925     @Override
926     public List<BoardVO> select() {
927         return this.boardDao.readAll();
928     }
929
```

4)BoardDaoImpl.java code 추가

```
930
931
932     @Override
933     public List<BoardVO> readAll() {
934         Map<String, Object> map = new HashMap<String, Object>();
935         this.sqlSession.selectList("Board.selectAllSP", map);
936         List<BoardVO> list = (List<BoardVO>)map.get("results");
937         return list;
938     }
```



```
}
```

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"
```

```

998     },
999     {
1000         "idx": 1,
1001         "readnum": 0,
1002         "grp": 1,
1003         "lev": 0,
1004         "step": 0,
1005         "name": "백두산",
1006         "email": "javaexpert@nate.com",
1007         "title": "첫 글입니다",
1008         "contents": null,
1009         "writedate": "2018-06-13 09:31:43.0"
1010     }
1011 ]
1012 }
1013

```

9)정상적으로 게시글 목록이 나오면 성공

9. 한 게시글 보기

1)BoardController.java 코드 추가

```

1020 @RequestMapping(value = "board/{page}/{idx}", method = RequestMethod.GET)
1021 public ModelAndView select(@PathVariable int page, @PathVariable int idx) {
1022     BoardVO board = this.boardService.selectBoard(idx);
1023     String contents = board.getContents();
1024     contents = contents.replaceAll("\n", "<br />");
1025     board.setContents(contents);
1026     ModelAndView mav = new ModelAndView();
1027     mav.addObject("code", Boolean.TRUE);
1028     mav.addObject("data", board);
1029     mav.addObject("page", page);
1030     mav.setViewName("view"); // /WEB-INF/html/view.jsp
1031     return mav;
1032 }
1033

```

2)/WEB-INF/views/view.jsp 생성하기

```

1036 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
1037 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
1038 <c:set var="board" value="${data}" />
1039 <!DOCTYPE html>
1040 <html>
1041     <head>
1042         <meta charset="UTF-8">
1043         <title>Insert title here</title>
1044         <script src="<c:url value='/js/jquery-1.12.4.js'/>"></script>
1045         <script>
1046             $(document).ready(function(){
1047                 $("#idx").text("${board.idx}");
1048                 $("#writedate").text("${board.writedate}");
1049                 $("#name").text("${board.name}");
1050                 $("#readnum").text("${board.readnum}");
1051                 $("#title").text("${board.title}");
1052                 $("#contents").html("${board.contents}");
1053             });
1054         </script>
1055     </head>
1056     <body>
1057         <h1>게시판 글 확인</h1>

```

```

1058         <ul>
1059             <li>Number : <span id="idx"></span></li>
1060             <li>Date : <span id="writedate"></span></li>
1061             <li>Writer : <span id="name"></span></li>
1062             <li>Readnum : <span id="readnum"></span></li>
1063             <li>Title : <span id="title"></span></li>
1064             <li>Contents : <div id="contents"
1065                 style="width:500px;height:100px;background-color:yellow"></div></li>
1066         </ul>
1067         <div>
1068             <button onclick="javascript:location.href='/Board';">목록으로</button>
1069             <button id="btnUpdate">수정하기</button>
1070             <button id="btnDelete">삭제하기</button>
1071             <button id="btnAnswer">답변하기</button>
1072         </div>
1073         <hr />
1074         <div>
1075         </div>
1076     </body>
1077 </html>

```

3)BoardServiceImpl.java 코드 추가

```

1080 @Override
1081 public BoardVO selectBoard(int idx) {
1082     return this.boardDao.read(idx);
1083 }

```

4)BoardDaoImpl.java code 추가

```

1087 @Override
1088 public BoardVO read(int idx) {
1089     Map<String, Object> map = new HashMap<String, Object>();
1090     map.put("idx", idx);
1091     this.sqlSession.selectOne("Board.selectByIdx", map);
1092     List<BoardVO> list = (List<BoardVO>)map.get("result");
1093     return list.get(0);
1094 }

```

5)Database에 StoredProcedure 추가

```

1098 CREATE OR REPLACE PROCEDURE sp_board_select_byidx
1099 (
1100     v_idx          IN          Board.idx%TYPE,
1101     v_record       OUT         SYS_REFCURSOR
1102 )
1103 AS
1104 BEGIN
1105     OPEN v_record FOR
1106     SELECT idx, grp, name, email, title, contents, writedate, readnum, lev, step
1107     FROM Board WHERE idx = v_idx;
1108
1109     UPDATE Board SET readnum = readnum + 1
1110     WHERE idx = v_idx;
1111     COMMIT;
1112 END;

```

6)board-mapper.xml 코드 추가

```

1116 <parameterMap type="hashMap" id="selectOneMap">

```

```

1117         <parameter property="idx" javaType="Integer"
1118             jdbcType="INTEGER" mode="IN"/>
1119         <parameter property="result" javaType="ResultSet"
1120             jdbcType="CURSOR" mode="OUT" resultMap="selectMap"/>
1121     </parameterMap>
1122     <select id="selectByIdx" parameterMap="selectOneMap"
1123         statementType="CALLABLE">
1124         { call sp_board_select_byidx(?, ?) }
1125     </select>
1126
1127 7)index.jsp 수정
1128     var page = 1; //첫줄에 추가
1129
1130     else if(count > 0){
1131         for(var i = 0 ; i < count ; i++){
1132             var board = data[i];
1133             var writedate = board.writedate.substr(0, 10);
1134             str += "<tr>";
1135             str += "<td style='text-align:center>'>" + board.grp + "</td>" +
1136                 "<td style='text-align:center><a href='mailto:" + board.email + "'>" +
1137                 board.name + "</a></td>" +
1138                 "<td>&nbsp;" + "<a href='" + page + "/" + board.idx + "'>" +
1139                 board.title + "</a></td>" + //여기를 수정
1140                 "<td style='text-align:center>'>" + writedate + "</td>" +
1141                 "<td style='text-align:center>'>" + board.readnum + "</td>";
1142             str += "</tr>";
1143         }
1144     }

```

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

9)POSTMAN 으로 테스트

GET localhost:8080/SpringBoard/board/1/2

Send

-결과

```

1150     <!DOCTYPE html>
1151     <html>
1152         <head>
1153             <meta charset="UTF-8">
1154             <title>Insert title here</title>
1155             <script src="../js/jquery-1.12.4.js"></script>
1156             <script>
1157                 $(document).ready(function(){
1158                     $("#idx").text("2");
1159                     $("#writedate").text("2018-06-13 09:53:16.0");
1160                     $("#name").text("한라산");
1161                     $("#readnum").text("4");
1162                     $("#title").text("두번째 글");
1163                     $("#contents").html("두번째 글입니다.");
1164                 });
1165             </script>
1166         </head>
1167         <body>
1168             <h1>게시판 글 확인</h1>
1169             <ul>
1170                 <li>Number :
1171                     <span id="idx"></span>
1172                 </li>
1173                 <li>Date :
1174                     <span id="writedate"></span>

```

```

1175         </li>
1176         <li>Writer :
1177             <span id="name"></span>
1178         </li>
1179         <li>Readnum :
1180             <span id="readnum"></span>
1181         </li>
1182         <li>Title :
1183             <span id="title"></span>
1184         </li>
1185         <li>Contents :
1186             <div id="contents"
1187                 style="width:500px;height:100px;background-color:yellow"></div>
1188         </li>
1189     </ul>
1190     <div>
1191         <button
1192             onclick="javascript:location.href='/SpringBoard/';">목록으로</button>
1193         <button id="btnUpdate">수정하기</button>
1194         <button id="btnDelete">삭제하기</button>
1195         <button id="btnAnswer">답변하기</button>
1196     </div>
1197 </div>
1198 </body>
1199 </html>
1200

```

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

10. 글 삭제하기

1)BoardController.java 코드 추가

```

1205
1206
1207     @RequestMapping(value = "board/{page}/{idx}", method = RequestMethod.DELETE)
1208     @ResponseBody
1209     public Map delete(@PathVariable int idx, @PathVariable int page) {
1210         this.boardService.deleteBoard(idx);
1211         Map<String, Object> result = new HashMap<String, Object>();
1212         result.put("code", Boolean.TRUE);
1213         return result;
1214     }
1215

```

2)/WEB-INF/views/view.jsp 코드 수정하기

```

1216
1217
1218     $("#btnDelete").bind("click", function(){
1219         $.ajax({
1220             url : ${board.idx},
1221             method : "DELETE",
1222             success : function(data){
1223                 if(data.code){
1224                     location.href = "/SpringBoard";
1225                 }
1226             }
1227         });
1228     });
1229

```

3)BoardServiceImpl.java 코드 추가

```

1230
1231
1232     @Override

```

```
1233     public void deleteBoard(int idx) {
1234         this.boardDao.delete(idx);
1235     }
1236
```

4)BoardDaoImpl.java code 추가

```
1239     @Override
1240     public void delete(int idx) {
1241         this.sqlSession.delete("Board.deleteSP", idx);
1242     }
1243
```

5)Database에 StoredProcedure 추가

```
1246     CREATE OR REPLACE PROCEDURE sp_board_delete
1247     (
1248         v_idx      IN      Board.idx%TYPE
1249     )
1250     IS
1251     BEGIN
1252         DELETE FROM Board
1253         WHERE idx = v_idx;
1254
1255         COMMIT;
1256     END;
1257
```

6)board-mapper.xml 코드 추가

```
1260     <delete id="deleteSP" parameterType="Integer">
1261         { call sp_board_delete(#{idx}) }
1262     </delete>
1263
```

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

8)POSTMAN 으로 테스트

```
1267     DELETE localhost:8080/SpringBoard/board/1/1
1268     Send
1269     -결과
1270     {
1271         "code": true
1272     }
1273
```

9)원하는 글 삭제 성공

11. 글 수정하기

1)BoardController.java 코드 추가

```
1280     @RequestMapping(value = "board/{page}/{idx}", method = RequestMethod.PUT)
1281     @ResponseBody
1282     public Map update(@RequestBody BoardVO board) {
1283         this.boardService.updateBoard(board);
1284         Map<String, Object> result = new HashMap<String, Object>();
1285         result.put("code", Boolean.TRUE);
1286         return result;
1287     }
1288
```

2)/WEB-INF/views/view.jsp 코드 수정하기

```
1291         var flag = true;
1292         $("#btnUpdate").click(function(){
```

```

1293     if(flag){
1294         var title = $("#title").text();
1295         var str = "<input type='text' id='title' value='" + title + "' size='30' />";
1296         $("#title").replaceWith(str);
1297         var contents = $("#contents").text();
1298         str = "<textarea id='contents' cols='40' rows='5'>" + contents +
            "</textarea>";
1299         $("#contents").replaceWith(str);
1300         $("#btnUpdate").text("갱신하기");
1301         flag = false;
1302     }else if(!flag){
1303         $.ajax({
1304             url : "",
1305             method : "PUT",
1306             dataType : "json",
1307             contentType : "application/json;charset=utf-8",
1308             data : JSON.stringify({
1309                 "idx" : parseInt($("#idx").text()),
1310                 "title" : $('#title').val(),
1311                 "contents" : $('#contents').val()
1312             }),
1313             success : function(data){
1314                 if(data.code){
1315                     location.reload();
1316                 }
1317             }
1318         });
1319     }
1320 });
1321

```

1322 3)BoardServiceImpl.java 코드 추가

```

1323
1324     @Override
1325     public void updateBoard(BoardVO board) {
1326         this.boardDao.update(board);
1327     }
1328

```

1329 4)BoardDaoImpl.java code 추가

```

1330
1331     @Override
1332     public void update(BoardVO board) {
1333         this.sqlSession.update("Board.updateSP", board);
1334     }
1335

```

1336 5)Database에 StoredProcedure 추가

```

1337
1338     CREATE OR REPLACE PROCEDURE sp_board_update
1339     (
1340         v_title      IN      Board.title%TYPE,
1341         v_contents   IN      Board.contents%TYPE,
1342         v_idx        IN      Board.idx%TYPE
1343     )
1344     IS
1345     BEGIN
1346         UPDATE Board SET title = v_title, contents = v_contents
1347         WHERE idx = v_idx;
1348
1349         COMMIT;
1350     END;
1351

```

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;&nbsp;&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;&nbsp;&nbsp;&nbsp;&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;&nbsp;&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;
```

```

1470         if(page == 1) {
1471             pageStr = "&lt;&lt;&nbsp;&nbsp;&nbsp;";
1472             pageStr += "&lt;&nbsp;&nbsp;&nbsp;";
1473         }else if(page != 1) {
1474             pageStr = "<a href='javascript:goPage(1)'>&lt;&lt;</a>&nbsp;&nbsp;&nbsp;";
1475             pageStr += "<a href='javascript:goPage(" + (startPage - pageSize) +
1476                 ")'>&lt;</a>&nbsp;&nbsp;&nbsp;";
1477         }
1478         for(var k = startPage ; k <= endPage ; k++){
1479             if(page == k) pageStr += "<span style='color:red;font-weight:bold'>" +
1480                 k + "</span>&nbsp;&nbsp;&nbsp;";
1481             else pageStr += "<a href='javascript:goPage(" + k + ")'>" + k +
1482                 "</a>&nbsp;&nbsp;&nbsp;";
1483         }
1484         if(page == totalPage) {
1485             pageStr += "&gt;</a>&nbsp;&nbsp;&nbsp;";
1486             pageStr += "&gt;&gt;";
1487         }else if(page != totalPage){
1488             pageStr += "<a href='javascript:goPage(" + (startPage + pageSize) +
1489                 ")'>&gt;</a>&nbsp;&nbsp;&nbsp;";
1490             pageStr += "<a href='javascript:goPage(" + totalPage +
1491                 ")'>&gt;&gt;</a>";
1492         }
1493         $("#paging").html(pageStr);
1494     }

```

4)BoardController.java에서 아래 코드 수정

```

1495     @RequestMapping(value = "board/{page}", method = RequestMethod.GET)
1496     @ResponseBody
1497     public Map select(@PathVariable int page) {
1498         int totalCount = this.boardService.getTotalCount();
1499         int pageSize = 10;
1500         int totalPage = (totalCount % pageSize == 0) ? totalCount / pageSize : totalCount
1501             / pageSize + 1;
1502         if(totalPage < page) page = totalPage;
1503
1504         int pageCount = 10;
1505
1506         int start = (page - 1) * pageCount + 1;
1507         int end = start + pageCount - 1 ;
1508
1509         int startPage = ((page - 1) / pageCount) * pageCount + 1;
1510         int endPage = startPage + pageCount - 1 ;
1511
1512         if(endPage> totalPage) endPage = totalPage;
1513         Map<String, Object> result = new HashMap<String, Object>();
1514         List<BoardVO> list = this.boardService.select(start, end);
1515         result.put("code", Boolean.TRUE);
1516         result.put("data", list);
1517         result.put("startPage", startPage);
1518         result.put("endPage", endPage);
1519         result.put("totalPage", totalPage);
1520         result.put("pageSize", pageSize);
1521         result.put("page", page);
1522         return result;
1523     }

```

5)BoardService.java와 BoardServiceImpl.java 코드 수정

-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",
```

```

1642         dataType : "json",
1643         contentType : "application/json;charset=utf-8",
1644         data : JSON.stringify({
1645             "idx" : parseInt($("#idx").text()),
1646             "name" : $("#name").val(),
1647             "email" : $("#email").val(),
1648             "title" : $("#title").val(),
1649             "contents" : $("#contents").val(),
1650             "grp" : parseInt($("#board.grp").val()),
1651             "lev" : parseInt($("#board.lev").val()),
1652             "step" : parseInt($("#board.step").val())
1653         }),
1654         success : function(data){
1655             if(data.code){
1656                 location.href = "/Board";
1657             }
1658         }
1659     });
1660 }
1661 });

```

3)BoardServiceImpl.java 코드 추가

```

1665 @Override
1666 public void answerBoard(BoardVO board) {
1667     this.boardDao.answer(board);
1668 }

```

4)BoardDaoImpl.java code 추가

```

1672 @Override
1673 public void answer(BoardVO board) {
1674     this.answer_update(board.getGrp(), board.getStep());
1675     this.sqlSession.insert("Board.answerInsertSP", board);
1676 }
1677
1678 private void answer_update(int grp, int step) {
1679     Map<String, Integer> map = new HashMap<String, Integer>();
1680     map.put("grp", grp);
1681     map.put("step", step);
1682     this.sqlSession.update("Board.answerUpdateSP", map);
1683 }

```

5)Database에 StoredProcedure 추가

```

1687 CREATE OR REPLACE PROCEDURE sp_board_answer_update
1688 (
1689     v_grp          IN          board.grp%TYPE,
1690     v_step         IN          board.step%TYPE
1691 )
1692 IS
1693 BEGIN
1694     UPDATE Board SET step = step + 1
1695     WHERE grp = v_grp AND step >= v_step;
1696     COMMIT;
1697 END;
1698 /
1699
1700 CREATE OR REPLACE PROCEDURE sp_board_answer_insert
1701 (

```

```

1702         v_name          IN    Board.name%TYPE,
1703         v_email          IN    Board.email%TYPE,
1704         v_title          IN    Board.title%TYPE,
1705         v_contents       IN    Board.contents%TYPE,
1706         v_grp            IN    Board.grp%TYPE,
1707         v_lev            IN    Board.lev%TYPE,
1708         v_step           IN    Board.step%TYPE
1709     )
1710 IS
1711 BEGIN
1712     INSERT INTO Board(idx, name, email, title, contents, grp, lev, step)
1713     VALUES(seq_board_idx.NEXTVAL, v_name, v_email, v_title, v_contents, v_grp,
1714            v_lev, v_step);
1715     COMMIT;
1716 END;
1717 /

```

6)board-mapper.xml 코드 추가

```

1720     <parameterMap type="boardVO" id="answerMap">
1721         <parameter property="name" mode="IN" />
1722         <parameter property="email" mode="IN" />
1723         <parameter property="title" mode="IN" />
1724         <parameter property="contents" mode="IN" />
1725         <parameter property="grp" mode="IN" />
1726         <parameter property="lev" mode="IN" />
1727         <parameter property="step" mode="IN" />
1728     </parameterMap>
1729     <insert id="answerInsertSP" parameterMap="answerMap">
1730         { call sp_board_answer_insert(?,?,?,?,?,?) }
1731     </insert>
1732     <update id="answerUpdateSP" parameterType="hashMap">
1733         { call sp_board_answer_update(#{grp}, #{step}) }
1734     </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

```

1750     CREATE SEQUENCE seq_reply_idx
1751     MAXVALUE 9999

```

-댓글 테이블을 생성하고, 댓글 참조키를 설정한다.

```

1755     CREATE TABLE Reply
1756     (
1757         idx          NUMBER(4),
1758         parent       NUMBER(4) DEFAULT 0,  //FK는 null이 가능하기 때문에
1759         writer       VARCHAR2(20) NOT NULL,
1760         text         VARCHAR2(2000) NOT NULL,

```

```

1761         regdate DATE DEFAULT SYSDATE,
1762         updatedate DATE DEFAULT SYSDATE,
1763         CONSTRAINT reply_idx_pk PRIMARY KEY(idx),
1764         CONSTRAINT reply_idx_fk FOREIGN KEY(parent) REFERENCES Board(idx)
1765     )
1766

```

3) 댓글을 위한 VO 객체 생성

-com.javasoft.vo.ReplyVO.java

```

1770 package com.javasoft.vo;
1771
1772 public class ReplyVO {
1773     private int idx;
1774     private int parent;
1775     private String writer;
1776     private String text;
1777     private String regdate;
1778     private String updatedate;
1779     public ReplyVO() {}
1780     public ReplyVO(int idx, int parent, String writer, String text, String regdate,
1781                     String updatedate) {
1782         this.idx = idx;
1783         this.parent = parent;
1784         this.writer = writer;
1785         this.text = text;
1786         this.regdate = regdate;
1787         this.updatedate = updatedate;
1788     }
1789     public int getIdx() {
1790         return idx;
1791     }
1792     public void setIdx(int idx) {
1793         this.idx = idx;
1794     }
1795     public int getParent() {
1796         return parent;
1797     }
1798     public void setParent(int parent) {
1799         this.parent = parent;
1800     }
1801     public String getWriter() {
1802         return writer;
1803     }
1804     public void setWriter(String writer) {
1805         this.writer = writer;
1806     }
1807     public String getText() {
1808         return text;
1809     }
1810     public void setText(String text) {
1811         this.text = text;
1812     }
1813     public String getRegdate() {
1814         return regdate;
1815     }
1816     public void setRegdate(String regdate) {
1817         this.regdate = regdate;
1818     }
1819     public String getUpdatedate() {
1820         return updatedate;
1821     }

```

```

1820     }
1821     public void setUpdatedate(String updatedate) {
1822         this.updatedate = updatedate;
1823     }
1824     @Override
1825     public String toString() {
1826         return "ReplyVO [idx=" + idx + ", parent=" + parent + ", writer=" + writer +
            ", text=" + text + ", regdate="
1827             + regdate + ", updatedate=" + updatedate + "]";
1828     }
1829 }

```

4)ReplyDao interface와 ReplyDaoImpl.java 생성

```

1832 -com.javasoft.dao.ReplyDao.java
1833
1834     package com.javasoft.dao;
1835
1836     import java.util.List;
1837
1838     import com.javasoft.vo.ReplyVO;
1839
1840     public interface ReplyDao {
1841         List<ReplyVO> list(int idx);
1842         void create(ReplyVO replyVO);
1843         void update(ReplyVO replyVO);
1844         void delete(int idx);
1845     }
1846
1847 -com.javasoft.dao.ReplyDaoImpl.java
1848
1849     package com.javasoft.dao;
1850
1851     import java.util.List;
1852
1853     import com.javasoft.vo.ReplyVO;
1854
1855     @Repository("replyDao")
1856     public class ReplyDaoImpl implements ReplyDao {
1857         @Autowired
1858         private SqlSession sqlSession;
1859
1860         @Override
1861         public List<ReplyVO> list(int idx) {
1862             // TODO Auto-generated method stub
1863             return null;
1864         }
1865
1866         @Override
1867         public void create(ReplyVO replyVO) {
1868             // TODO Auto-generated method stub
1869
1870         }
1871
1872         @Override
1873         public void update(ReplyVO replyVO) {
1874             // TODO Auto-generated method stub
1875
1876         }
1877
1878         @Override

```



```
1879         public void delete(int idx) {
1880             // TODO Auto-generated method stub
1881
1882         }
1883     }
1884
```

5)config/mybatis-config.xml 코드 추가

```
1887     <typeAliases>
1888         <typeAlias type="com.javasoft.vo.BoardVO" alias="boardVO"/>
1889         <typeAlias type="com.javasoft.vo.ReplyVO" alias="replyVO"/>    <--추가
1890     </typeAliases>
1891
```

6)config/reply-mapper.xml 추가

```
1892     <?xml version="1.0" encoding="UTF-8" ?>
1893     <!DOCTYPE mapper
1894         PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
1895             "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
1896
1897     <mapper namespace="Reply">
1898
1899
1900     </mapper>
1901
```

7)config/applicationContext.xml 코드 추가

```
1902
1903
1904     <property name="mapperLocations">
1905         <list>
1906             <value>classpath:board-mapper.xml</value>
1907             <value>classpath:reply-mapper.xml</value>
1908         </list>
1909     </property>
1910
```

8)ReplyService interface와 ReplyServiceImpl.java 생성 -com.javasoft.service.ReplyService.java

```
1911
1912
1913     package com.javasoft.service;
1914
1915     import java.util.List;
1916
1917     import com.javasoft.vo.ReplyVO;
1918
1919     public interface ReplyService {
1920         List<ReplyVO> selectReply(int idx);
1921         void insertReply(ReplyVO replyVO);
1922         void updateReply(ReplyVO replyVO);
1923         void deleteReply(int idx);
1924     }
1925
```

-com.javasoft.service.ReplyServiceImpl.java

```
1926
1927
1928     package com.javasoft.service;
1929
1930     import java.util.List;
1931
1932     import org.springframework.beans.factory.annotation.Autowired;
1933     import org.springframework.stereotype.Service;
1934
1935     import com.javasoft.dao.ReplyDao;
1936     import com.javasoft.vo.ReplyVO;
1937
1938
```

```

1939 @Service("replyService")
1940 public class ReplyServiceImpl implements ReplyService {
1941     @Autowired
1942     private ReplyDao replyDao;
1943
1944     @Override
1945     public List<ReplyVO> selectReply(int idx) {
1946         // TODO Auto-generated method stub
1947         return null;
1948     }
1949
1950     @Override
1951     public void insertReply(ReplyVO replyVO) {
1952         // TODO Auto-generated method stub
1953
1954     }
1955
1956     @Override
1957     public void updateReply(ReplyVO replyVO) {
1958         // TODO Auto-generated method stub
1959
1960     }
1961
1962     @Override
1963     public void deleteReply(int idx) {
1964         // TODO Auto-generated method stub
1965
1966     }
1967
1968 }
1969

```

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

```

1972 package com.javasoft.controller;
1973
1974 import org.springframework.beans.factory.annotation.Autowired;
1975 import org.springframework.stereotype.Controller;
1976
1977 import com.javasoft.service.ReplyService;
1978
1979 @Controller
1980 public class ReplyController {
1981     @Autowired
1982     private ReplyService replyService;
1983 }
1984

```

10)댓글 입력하기

-Database에 StoredProcedure 생성하기

```

1988 CREATE OR REPLACE PROCEDURE sp_reply_insert
1989 (
1990     v_parent    IN    Reply.parent%TYPE,
1991     v_writer    IN    Reply.writer%TYPE,
1992     v_text      IN    Reply.text%TYPE
1993 )
1994 IS
1995 BEGIN
1996     INSERT INTO Reply (idx, parent, writer, text)
1997     VALUES (seq_reply_idx.NEXTVAL, v_parent, v_writer, v_text);
1998 COMMIT;

```

1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056

```
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>  
      
    <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></td>  
        </tr>  
    </table>  
</form>  
</div>
```

```

2057         <ul id="replies" style="list-style-type: none">
2058             </ul>
2059     </div>
2060

```

--script

```

2061
2062
2063     var idx = null;    <---코드 추가
2064     $(document).ready(function(){
2065         $("#idx").text("${board.idx}");
2066         idx = parseInt("${board.idx}");    <---코드 추가
2067         $("#writedate").text("${board.writedate}");
2068         $("#name").text("${board.name}");
2069
2070         ...
2071         ...
2072
2073         $("#replyWrite").click(function(){
2074             // 화면으로부터 입력 받은 변수값의 처리
2075             var replyText = $("#newReplyText");
2076             var replyWriter = $("#newReplyWriter");
2077
2078             var replyTextVal = replyText.val();
2079             var replyWriterVal = replyWriter.val();
2080
2081             $.ajax({
2082                 method : "POST",
2083                 url : "/SpringBoard/reply",
2084                 contentType : "application/json;charset=utf-8",
2085                 dataType : "json",
2086                 data : JSON.stringify({
2087                     "parent" : idx,
2088                     "text" : replyTextVal,
2089                     "writer" : replyWriterVal
2090                 }),
2091                 success : function (result) {
2092                     if(data.code){
2093                         alert("Insert Success");
2094                     }
2095                     //getReplies(); // 댓글 목록 출력 함수 호출
2096                     replyText.val(""); // 댓글 내용 초기화
2097                     replyWriter.val(""); // 댓글 작성자 초기화
2098                 }
2099             });
2100         });
2101

```

-reply-mapper.xml 코드 추가

```

2102
2103
2104     <parameterMap type="replyVO" id="replyInsertMap">
2105         <parameter property="parent" javaType="Integer"
2106             jdbcType="INTEGER" mode="IN" />
2107         <parameter property="writer" javaType="String"
2108             jdbcType="VARCHAR" mode="IN" />
2109         <parameter property="text" javaType="String"
2110             jdbcType="VARCHAR" mode="IN" />
2111     </parameterMap>
2112     <insert id="create" parameterMap="replyInsertMap">
2113         { call sp_reply_insert (?, ?, ?) }
2114     </insert>
2115

```

-Insert를 위한 ReplyDaoImpl.java 수정

```

2117
2118     @Override
2119     public void create(ReplyVO replyVO) {
2120         this.sqlSession.insert("Reply.create", replyVO);
2121     }
2122

```

2123 -Insert를 위한 ReplyServiceImpl.java 수정

```

2124
2125     @Override
2126     public void insertReply(ReplyVO replyVO) {
2127         this.replyDao.create(replyVO);
2128     }
2129

```

2130 11)댓글 목록 보기

2131 -Database에 StoredProcedure 생성하기

```

2132
2133     CREATE OR REPLACE PROCEDURE sp_reply_select
2134     (
2135         v_parent      IN      Reply.parent%TYPE,
2136         v_reply        OUT     SYS_REFCURSOR
2137     )
2138     IS
2139     BEGIN
2140         OPEN v_reply FOR
2141         SELECT idx, parent, writer, text, regdate, updatedate
2142         FROM Reply
2143         WHERE parent = v_parent
2144         ORDER BY idx;
2145     END;
2146

```

2147 -ReplyController.java 코드 추가

```

2148
2149     @RequestMapping(value = "/reply/all/{parent}", method = RequestMethod.GET)
2150     @ResponseBody
2151     public Map list(@PathVariable int parent) {
2152         Map<String, Object> map = new HashMap<String, Object>();
2153         List<ReplyVO> list = this.replyService.selectReply(parent);
2154         map.put("code", Boolean.TRUE);
2155         map.put("data", list);
2156         return map;
2157     }
2158 }
2159

```

2160 -view.jsp 코드 추가

```

2161     --script
2162
2163     $(document).ready(function(){
2164         $("#idx").text("${board.idx}");
2165         idx = parseInt("${board.idx}");
2166         getReplies();          <--코드 추가
2167
2168         ...
2169         ...
2170         success : function (result) {
2171             if(result.code){
2172                 alert("Insert Success");
2173             }
2174             getReplies(); //주석 해제
2175             ...
2176             ...

```

```

2177     function getReplies() {
2178     $.getJSON("/SpringBoard/reply/all/" + idx, function (result) {
2179         var data = result.data;
2180         var str = "";
2181         $(data).each(function () {
2182             str += "<li class='replyLi' id='" + this.idx + "'>"
2183                 + "<p class='text'" + this.text + "</p>"
2184                 + "<p class='writer'" + this.writer + "</p>"
2185                 + "<button type='button' class='update'
2186                     style='background-color:green;color:white;font-weight:bold;'>댓글
2187                     수정</button>"
2188                 + "<button type='button' class='delete'
2189                     style='background-color:red;color:white;font-weight:bold;'>댓글
2190                     삭제</button>"
2191                 + "</li>"
2192                 + "<hr/>";
2193         });
2194         $("#replies").html(str);
2195     });
2196 }

```

-reply-mapper.xml 코드 추가

```

2196     <resultMap id="replyResultMap" type="replyVO">
2197         <id property="idx" column="idx" />
2198         <result property="parent" column="parent" />
2199         <result property="writer" column="writer" />
2200         <result property="text" column="text" />
2201         <result property="regdate" column="regdate" />
2202         <result property="updatedate" column="updatedate" />
2203     </resultMap>
2204
2205     <parameterMap type="hashMap" id="replySelectMap">
2206         <parameter property="parent" mode="IN"
2207             javaType="Integer" jdbcType="INTEGER"/>
2208         <parameter property="results" mode="OUT"
2209             javaType="ResultSet" jdbcType="CURSOR" resultMap="replyResultMap"/>
2210     </parameterMap>
2211     <select id="list" parameterMap="replySelectMap" statementType="CALLABLE">
2212         { call sp_reply_select(?, ?) }
2213     </select>

```

-Select를 위한 ReplyDaoImpl.java 수정

```

2217     @Override
2218     public List<ReplyVO> list(int parent) {
2219         Map<String, Object> result = new HashMap<String, Object>();
2220         result.put("parent", parent);
2221         this.sqlSession.selectList("Reply.list", result);
2222         List<ReplyVO> list = (List<ReplyVO>)result.get("results");
2223         return list;
2224     }

```

-Select를 위한 ReplyServiceImpl.java 수정

```

2228     @Override
2229     public List<ReplyVO> selectReply(int parent) {
2230         return this.replyDao.list(parent);
2231     }
2232

```

12) 댓글 수정하기

-view.jsp 코드 추가

--jQuery코드에서 주목해서 봐야할 것은 클릭 이벤트 선택자가 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 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">
```

```

2290         <parameter property="text" javaType="String"
2291             jdbcType="VARCHAR" mode="IN"/>
2292         <parameter property="idx" javaType="Integer"
2293             jdbcType="INTEGER" mode="IN"/>
2294     </parameterMap>
2295     <update id="update" parameterMap="replyUpdateMap">
2296         { call sp_reply_update(?, ?) }
2297     </update>

```

2298 -ReplyDaoImpl.java 코드 수정

```

2300
2301     @Override
2302     public void update(ReplyVO replyVO) {
2303         Map<String, Object> map = new HashMap<String, Object>();
2304         map.put("text", replyVO.getText());
2305         map.put("idx", replyVO.getIdx());
2306         this.sqlSession.update("Reply.update", map);
2307     }

```

2308 -ReplyServiceImpl.java 코드 수정

```

2309
2310     @Override
2311     public void updateReply(ReplyVO replyVO) {
2312         this.replyDao.update(replyVO);
2313     }

```

2314 -ReplyController.java 코드 추가

```

2315
2316     @RequestMapping(value = "/reply", method = RequestMethod.PUT)
2317     @ResponseBody
2318     public Map update(@RequestBody ReplyVO replyVO) {
2319         Map<String, Object> map = new HashMap<String, Object>();
2320         this.replyService.updateReply(replyVO);
2321         System.out.println(replyVO);
2322         map.put("code", Boolean.TRUE);
2323         return map;
2324     }

```

2325 13)댓글 삭제하기

2326 -view.jsp 코드 추가

```

2327
2328     $("#replies").on("click", ".replyLi .delete", function () {
2329         var replyIdx = $(this).parent().attr("id");
2330         $.ajax({
2331             url : "/SpringBoard/reply/" + parseInt(replyIdx),
2332             method : "DELETE",
2333             dataType : "json",
2334             success : function(result){
2335                 getReplies();
2336             }
2337         });
2338     });

```

2339 -Database StoredProcedure 추가하기

```

2340
2341     CREATE OR REPLACE PROCEDURE sp_reply_delete
2342     (
2343         v_idx IN Reply.idx%TYPE
2344     )
2345     IS
2346     BEGIN

```



```
2350         DELETE FROM Reply WHERE idx = v_idx;
2351         COMMIT;
2352     END;
2353
2354 -reply-mapper.xml 코드 추가
2355
2356     <delete id="delete" parameterType="Integer">
2357         { call sp_reply_delete(#{idx}) }
2358     </delete>
2359
2360 -ReplyDaoImpl.java 코드 수정
2361
2362     @Override
2363     public void delete(int idx) {
2364         this.sqlSession.delete("Reply.delete", idx);
2365     }
2366
2367 -ReplyServiceImpl.java 코드 수정
2368
2369     @Override
2370     public void deleteReply(int idx) {
2371         this.replyDao.delete(idx);
2372     }
2373
2374 -ReplyController.java 코드 추가
2375
2376     @RequestMapping(value = "/reply/{idx}", method = RequestMethod.DELETE)
2377     @ResponseBody
2378     public Map delete(@PathVariable int idx) {
2379         Map<String, Object> map = new HashMap<String, Object>();
2380         this.replyService.deleteReply(idx);
2381         map.put("code", Boolean.TRUE);
2382         return map;
2383     }
```