

```
1 1. Database Configuration
2 1)계정생성
3   sqlplus /nolog
4
5   conn SYS as sysdba
6
7   ALTER SESSION SET "_ORACLE_SCRIPT"=true;
8
9   CREATE USER bbs
10  IDENTIFIED BY bbs
11  DEFAULT TABLESPACE USERS
12  TEMPORARY TABLESPACE temp;
13
14  ALTER USER bbs DEFAULT TABLESPACE USERS
15  QUOTA UNLIMITED ON users;
16
17  GRANT resource, connect TO bbs;
18
19 2)Sequence 및 Table 생성
20  CREATE SEQUENCE seq_board_idx
21  MAXVALUE 9999;
22
23  CREATE TABLE Board
24  (
25      idx          NUMBER(4),
26      name         VARCHAR2(20)    NOT NULL,
27      email        VARCHAR2(100),
28      title        VARCHAR2(100)   NOT NULL,
29      contents     VARCHAR2(2000)  NOT NULL,
30      writedate    DATE            DEFAULT SYSDATE,
31      readnum      NUMBER(4)       DEFAULT 0,
32      grp          NUMBER(4)       NOT NULL,
33      lev          NUMBER(2)       DEFAULT 0,
34      step         NUMBER(2)       DEFAULT 0,
35      CONSTRAINT board_idx_pk PRIMARY KEY(idx)
36  );
37
38
39 2. Java Configuration
40 1)New > Spring Legacy Project
41 2)Project name : Spring Board
42 3)Spring MVC Project > Finish
43 4)Project Settings > Enter a topLevelPackages : com.example.board > Finish
44 5)pom.xml에서
45   -properties version 수정
46   <properties>
47       <java-version>1.8</java-version>
48       <org.springframework-version>4.3.20.RELEASE</org.springframework-version>
49       <org.aspectj-version>1.9.2</org.aspectj-version>
50       <org.slf4j-version>1.7.25</org.slf4j-version>
51   </properties>
52
53   -기존 dependency version 변경
54   <!-- Servlet -->
55   <dependency>
56       <groupId>javax.servlet</groupId>
57       <artifactId>javax.servlet-api</artifactId>
```

```
58     <version>3.1.0</version>
59     <scope>provided</scope>
60 </dependency>
61 <dependency>
62     <groupId>javax.servlet.jsp</groupId>
63     <artifactId>javax.servlet.jsp-api</artifactId>
64     <version>2.3.0</version>
65     <scope>provided</scope>
66 </dependency>
67 <!-- Test -->
68 <dependency>
69     <groupId>junit</groupId>
70     <artifactId>junit</artifactId>
71     <version>4.12</version>
72     <scope>test</scope>
73 </dependency>
74
75 -새로 생성된 pom.xml파일에 필요한 library 추가 > Maven Clean > Maven Install
76 <dependency>
77     <groupId>org.springframework</groupId>
78     <artifactId>spring-jdbc</artifactId>
79     <version>4.3.20.RELEASE</version>
80 </dependency>
81 <dependency>
82     <groupId>com.oracle</groupId>
83     <artifactId>ojdbc8</artifactId>
84     <version>12.2</version>
85 </dependency>
86 <dependency>
87     <groupId>org.mybatis</groupId>
88     <artifactId>mybatis</artifactId>
89     <version>3.4.6</version>
90 </dependency>
91 <dependency>
92     <groupId>org.mybatis</groupId>
93     <artifactId>mybatis-spring</artifactId>
94     <version>1.3.2</version>
95 </dependency>
96 <dependency>
97     <groupId>com.fasterxml.jackson.core</groupId>
98     <artifactId>jackson-databind</artifactId>
99     <version>2.9.7</version>
100 </dependency>
101 <dependency>
102     <groupId>com.fasterxml.jackson.core</groupId>
103     <artifactId>jackson-core</artifactId>
104     <version>2.9.7</version>
105 </dependency>
106 <dependency>
107     <groupId>com.fasterxml.jackson.core</groupId>
108     <artifactId>jackson-annotations</artifactId>
109     <version>2.9.7</version>
110 </dependency>
111 <dependency>
112     <groupId>org.springframework</groupId>
113     <artifactId>spring-test</artifactId>
114     <version>4.3.20.RELEASE</version>
```

```
115         <scope>test</scope>
116     </dependency>
117
118
119 3. MyBatis Configuration
120 1)oracle.properties file 추가
121     -src/main/resources > right-click > New > File
122     -File name : dbinfo.properties
123
124     db.driver=oracle.jdbc.driver.OracleDriver
125     db.url=jdbc:oracle:thin:@192.168.56.3:1521:ORCL
126     db.username=bbs
127     db.password=bbs
128
129 2)mybatis-config.xml file 생성
130     -src/main/resources > right-click > New > Other > XML > XML File > Next
131     -File name : mybatis-config.xml > Finish
132
133     <!DOCTYPE configuration
134     PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
135     "http://mybatis.org/dtd/mybatis-3-config.dtd">
136
137     <configuration>
138     </configuration>
139
140 3)board-mapper.xml file 생성
141     -src/main/resources > right-click > New > Other > XML > XML File > Next
142     -File name : board-mapper.xml > Finish
143
144     <!DOCTYPE mapper
145     PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
146     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
147
148     <mapper namespace="Board">
149     </mapper>
150
151
152 4. Spring Configuration
153 1)src/main/webapp/WEB-INF/spring/root-context.xml 이름 변경
154     -applicationContext.xml
155
156 2)src/main/webapp/WEB-INF/spring/appServlet/servlet-context.xml 이름 변경
157     -beans.xml
158
159 3)src/main/webapp/WEB-INF/web.xml 수정
160     -아래 코드 추가
161     <display-name>SpringBoard</display-name>
162     <welcome-file-list>
163     <welcome-file>index.html</welcome-file>
164     </welcome-file-list>
165
166     <context-param>
167     <param-name>contextConfigLocation</param-name>
168     <param-value>/WEB-INF/spring/root-context.xml</param-value>
169     </context-param>
170     <servlet>
171     <servlet-name>appServlet</servlet-name>
```

```

172     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
173     <init-param>
174         <param-name>contextConfigLocation</param-name>
175         <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
176     </init-param>
177     <load-on-startup>1</load-on-startup>
178 </servlet>
179

```

-아래로 변경

```

181     <context-param>
182         <param-name>contextConfigLocation</param-name>
183         <param-value>/WEB-INF/spring/applicationContext.xml</param-value>
184     </context-param>
185     <servlet>
186         <servlet-name>appServlet</servlet-name>
187         <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
188         <init-param>
189             <param-name>contextConfigLocation</param-name>
190             <param-value>/WEB-INF/spring/appServlet/beans.xml</param-value>
191         </init-param>
192         <load-on-startup>1</load-on-startup>
193     </servlet>
194
195

```

5. index.html 생성 및 정적 페이지 처리하기

1)beans.xml file 수정

-아래 코드 추가

```
<default-servlet-handler />
```

2)index.html 생성

-src/main/webapp > right-click > New > Other > Web > HTML File > Next

-File name : index.html > Finish

```

205     <!DOCTYPE html>
206     <html>
207     <head>
208         <meta charset="UTF-8">
209         <title>Spring Board</title>
210     </head>
211     <body>
212         <h1>Spring Board</h1>
213     </body>
214 </html>
215

```

3)style.css file 생성

-src/main/webapp > right-click > New > Folder

-Folder name : css > Finish

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

-File name : style.css > Finish

```

222     @charset "UTF-8";
223     body{
224         background-color: yellow;
225     }
226     h1 {
227         color : blue;
228     }

```

```
229
230 4) file 추가하기
231     -src/main/webapp > right-click > New > Folder
232     -Folder name : js > Finish
233     -이미 download 받은 jquery.min.js 파일 넣기
234
235 5)index.html 수정
236
237     <!DOCTYPE html>
238     <html>
239     <head>
240         <meta charset="UTF-8">
241         <title>Spring Board</title>
242         <link rel="stylesheet" type="text/css" href="css/style.css" />
243         <script src="js/jquery.min.js"></script>
244         <script>
245             $(document).ready(function(){
246                 alert("Hello Spring Board");
247             });
248         </script>
249     </head>
250     <body>
251         <h1>Spring Board</h1>
252     </body>
253 </html>
254
255 6)Project > right-click > Run As > Run on Server > Finish
256
257 7))http://localhost:8080/board/
258
259
260 6. Java Package 및 관련 Class file 생성
261 1)package 생성
262     -src/main/java/com.example.vo
263     -src/main/java/com.example.dao
264     -src/main/java/com.example.service
265     -src/main/java/com.example.controller <--- com.example.biz를 변경
266
267 2)BoardVO class 생성
268     -src/main/java/com.example.vo.BoardVO class 생성
269
270     package com.example.vo;
271
272     public class BoardVO {
273         private int idx, readnum, grp, lev, step;
274         private String name, email, title, contents;
275         private String writedate;
276
277         public int getIdx() {
278             return idx;
279         }
280         public void setIdx(int idx) {
281             this.idx = idx;
282         }
283         public int getReadnum() {
284             return readnum;
285         }
286     }
```

```
286     public void setReadnum(int readnum) {
287         this.readnum = readnum;
288     }
289     public int getGrp() {
290         return grp;
291     }
292     public void setGrp(int grp) {
293         this.grp = grp;
294     }
295     public int getLev() {
296         return lev;
297     }
298     public void setLev(int lev) {
299         this.lev = lev;
300     }
301     public int getStep() {
302         return step;
303     }
304     public void setStep(int step) {
305         this.step = step;
306     }
307     public String getName() {
308         return name;
309     }
310     public void setName(String name) {
311         this.name = name;
312     }
313     public String getEmail() {
314         return email;
315     }
316     public void setEmail(String email) {
317         this.email = email;
318     }
319     public String getTitle() {
320         return title;
321     }
322     public void setTitle(String title) {
323         this.title = title;
324     }
325     public String getContents() {
326         return contents;
327     }
328     public void setContents(String contents) {
329         this.contents = contents;
330     }
331     public String getWritedate() {
332         return writedate;
333     }
334     public void setWritedate(String writedate) {
335         this.writedate = writedate;
336     }
337     @Override
338     public String toString() {
339         return "BoardVO [idx=" + idx + ", readnum=" + readnum + ", grp=" + grp + ", lev=" + lev
340             + ", step=" + step
340             + ", name=" + name + ", email=" + email + ", title=" + title + ", contents=" + contents
340             + ", writedate="
```

```
341         + writedate + "];";
342     }
343 }
344
345 3)BoardService 생성
346 -src/example.service.BoardService interface 생성
347
348 package com.example.service;
349
350 import java.util.List;
351
352 import com.example.vo.BoardVO;
353
354 public interface BoardService {
355     void insertBoard(BoardVO board);
356     void selectBoard(Map map);
357     void select(Map map);
358     void updateBoard(BoardVO board);
359     void deleteBoard(int idx);
360     void answerBoard(BoardVO board);
361     void getTotalCount(Map map);
362 }
363
364 -src/main/java/com.example.service.BoardServiceImpl class 생성
365
366 package com.example.service;
367
368 import java.util.Map;
369
370 import org.springframework.beans.factory.annotation.Autowired;
371 import org.springframework.stereotype.Service;
372
373 import com.example.dao.BoardDao;
374 import com.example.vo.BoardVO;
375
376 @Service("boardService")
377 public class BoardServiceImpl implements BoardService {
378     @Autowired
379     BoardDao boardDao;
380
381     @Override
382     public void insertBoard(BoardVO board) {
383         // TODO Auto-generated method stub
384     }
385 }
386
387 @Override
388 public void selectBoard(Map map) {
389     // TODO Auto-generated method stub
390 }
391 }
392
393 @Override
394 public void select(Map map) {
395     // TODO Auto-generated method stub
396 }
397 }
```

```
398
399     @Override
400     public void updateBoard(BoardVO board) {
401         // TODO Auto-generated method stub
402
403     }
404
405     @Override
406     public void deleteBoard(int idx) {
407         // TODO Auto-generated method stub
408
409     }
410
411     @Override
412     public void answerBoard(BoardVO board) {
413         // TODO Auto-generated method stub
414
415     }
416
417     @Override
418     public void getTotalCount(Map map) {
419         // TODO Auto-generated method stub
420
421     }
422 }
423
424
425 4)BoardDao 객체 생성
426 -src/main/java/com.example.dao.BoardDao interface
427
428     package com.example.dao;
429
430     import java.util.List;
431
432     import com.example.vo.BoardVO;
433
434     public interface BoardDao {
435         void create(BoardVO board);
436         void read(Map map);
437         void readAll(Map map);
438         void update(BoardVO board);
439         void delete(int idx);
440         void answer(BoardVO board);
441         void getTotalCount(Map map);
442     }
443
444 -src/main/java/com.example.dao.BoardDaoImpl.java 생성
445
446     package com.example.dao;
447
448     import java.util.Map;
449
450     import org.apache.ibatis.session.SqlSession;
451     import org.springframework.beans.factory.annotation.Autowired;
452     import org.springframework.stereotype.Repository;
453
454     import com.example.vo.BoardVO;
```



```
455
456     @Repository("boardDao")
457     public class BoardDaoImpl implements BoardDao {
458         @Autowired
459         SqlSession sqlSession;
460
461         @Override
462         public void create(BoardVO board) {
463             // TODO Auto-generated method stub
464
465         }
466
467         @Override
468         public void read(Map map) {
469             // TODO Auto-generated method stub
470
471         }
472
473         @Override
474         public void readAll(Map map) {
475             // TODO Auto-generated method stub
476
477         }
478
479         @Override
480         public void update(BoardVO board) {
481             // TODO Auto-generated method stub
482
483         }
484
485         @Override
486         public void delete(int idx) {
487             // TODO Auto-generated method stub
488
489         }
490
491         @Override
492         public void answer(BoardVO board) {
493             // TODO Auto-generated method stub
494
495         }
496
497         @Override
498         public void getTotalCount(Map map) {
499             // TODO Auto-generated method stub
500
501         }
502     }
503
504
505 5)BoardController 객체 생성
506     -src/main/java/com.example.controller.HomeController class 이름 변경
507     -BoardController.java
508
509
510 6)applicationContext.xml 수정
511     -context check
```

```
512 <context:property-placeholder location="classpath:oracle.properties"/>
513
514 <bean id="dataSource"
515     class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
516     <property name="driverClass" value="${db.driver}" />
517     <property name="url" value="${db.url}" />
518     <property name="username" value="${db.username}" />
519     <property name="password" value="${db.password}" />
520 </bean>
521
522 <bean id="sqlSessionFactory"
523     class="org.mybatis.spring.SqlSessionFactoryBean">
524     <property name="dataSource" ref="dataSource" />
525     <property name="configLocation"
526         value="classpath:mybatis-config.xml" />
527     <property name="mapperLocations">
528         <list>
529             <value>classpath:board-mapper.xml</value>
530         </list>
531     </property>
532 </bean>
533
534 <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate">
535     <constructor-arg ref="sqlSessionFactory" name="sqlSessionFactory" />
536 </bean>
```

7)beans.xml 수정

```
537
538 -기존
539 <context:component-scan base-package="com.examp.controller" />
540 <resources mapping="/resources/**" location="/resources/" />
541 <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
542     <beans:property name="prefix" value="/WEB-INF/views/" />
543     <beans:property name="suffix" value=".jsp" />
544 </beans:bean>
545
546
547 -수정
548 <resources mapping="/static/**" location="/static/" />
549 <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
550     <beans:property name="prefix" value="/static/" />
551     <beans:property name="suffix" value=".jsp" />
552 </beans:bean>
553 <context:component-scan base-package="com.example" />
```

8)src/main/webapp/static folder 생성

9)bootstrap을 다운받아 css, js, fonts folder를 webapp/static folder하위에 추가

```
558 -src/main/webapp/static/css
559     --bootstrap.min.css
560 -src/main/webapp/static/js
561     --bootstrap.min.js
562 -src/main/webapp/static/fonts
563 -src/main/webapp/static/index.html
```

10)src/main/webapp/static/index.jsp 생성

```
566 -src/main/webapp > right-click > New > Other > Web > JSP File > Next
567 -File name : index.jsp > Finish
```

```
569 <%@ page language="java" contentType="text/html; charset=UTF-8"
570     pageEncoding="UTF-8"%>
571 <!DOCTYPE html>
572 <html>
573 <head>
574 <meta charset="UTF-8">
575 <title>Spring Board</title>
576 <link rel="stylesheet" href="static/css/bootstrap.min.css"
577     type="text/css">
578 <link rel="stylesheet" href="static/css/style.css" type="text/css">
579 <script src="static/js/jquery.min.js"></script>
580 </head>
581 <body>
582     <div class="text-center">
583         
584     </div>
585     <div>
586         <table>
587             <tr>
588                 <td>[ <a href="write">새 글쓰기</a> ]
589             </td>
590             <td>[ <span id="count"></span> ]
591             </td>
592         </tr>
593     </table>
594 </div>
595 <div>
596     <table class="table table-hover">
597         <colgroup>
598             <col width="10%" />
599             <col width="10%" />
600             <col width="60%" />
601             <col width="10%" />
602             <col width="10%" />
603         </colgroup>
604         <thead>
605             <tr>
606                 <th>번호</th>
607                 <th>작성자</th>
608                 <th>제목</th>
609                 <th>작성일</th>
610                 <th>조회수</th>
611             </tr>
612         </thead>
613         <tbody id="records"></tbody>
614     </table>
615 </div>
616 <p></p>
617 <div id="paging"></div>
618 </body>
619 </html>
620
621 11)src/main/webapp/static/index.html 수정
622
623 <script>
624     document.location.href = "/board";
625 </script>
```

```
626
627 12)BoardController.java 수정
628
629 @Controller
630 public class BoardController {
631     @Autowired
632     private BoardService boardService;
633
634     @RequestMapping(value = "/", method = RequestMethod.GET)
635     public String home() {
636         return "index";
637     }
638
```

```
639 13)style.css 수정
640
641 body {
642     margin-top : 0px;
643     margin-left : 0px;
644     margin-right:0px;
645 }
646 #banner{
647     text-align:center;
648 }
649 #banner span {
650     font-size:2em;
651 }
652 div:nth-of-type(2) table {
653     width:80%;
654     margin-left:auto; margin-right:auto;
655 }
656 div:nth-of-type(2) table td:nth-of-type(1) {
657     text-align:left;
658 }
659 div:nth-of-type(2) table td:nth-of-type(2) {
660     text-align:right;
661 }
662 div:nth-of-type(3) table {
663     width:80%;
664     margin-left:auto; margin-right:auto;
665     border-collapse: collapse;
666 }
667 div:nth-of-type(3) table th{
668     height:50px;
669     background-color: navy;
670     color:white;
671 }
672
```

673 14)project > right-click > Run As > Run on Server > Finish

674
675 15)<http://localhost:8080/board/>

676
677
678 7. 새 글쓰기

679 1)BoardController.java 코드 추가

```
680
681 @RequestMapping(value = "/write", method = RequestMethod.GET)
682 public String write() {
```

```
683     return "write"; // webapp/static/write.jsp
684 }
685
686 2)webapp/static/write.jsp 파일 생성
687
688 <%@ page language="java" contentType="text/html; charset=UTF-8"
689     pageEncoding="UTF-8"%>
690 <!DOCTYPE html>
691 <html>
692 <head>
693 <meta charset="UTF-8">
694 <title>Insert title here</title>
695 <link rel="stylesheet" href="static/css/bootstrap.min.css"
696     type="text/css" />
697 <script src="static/js/jquery.min.js"></script>
698 <script>
699     $(document).ready(function(){
700         $("#btnWrite").bind("click", function(){
701             $.ajax({
702                 url : "write",
703                 dataType : "json",
704                 method : "POST",
705                 contentType : 'application/json; charset=UTF-8',
706                 data : JSON.stringify({
707                     "name" : $("#txtName").val(),
708                     "email" : $("#txtEmail").val(),
709                     "title" : $("#txtTitle").val(),
710                     "contents" : $("#txtContents").val()
711                 }),
712                 success : function(data){
713                     if(data.code) {
714                         alert("Insert Success");
715                         location.href = "/board";
716                     } else alert("Insert Failure\nPlease Try again.");
717                 },
718                 error : function(err){
719                     console.log("Error 발생 : " + err);
720                 }
721             });
722         });
723         $("#btnCancel").bind("click", function(){
724             history.back();
725         });
726     });
727 </script>
728 </head>
729 <body>
730 <h1 class="text-center">게시판 글 쓰기</h1>
731 <form class="form-horizontal">
732 <div class="form-group">
733 <label for="txtName" class="col-sm-3 control-label">작성자</label>
734 <div class="col-sm-6">
735 <input type="text" class="form-control" id="txtName"
736     placeholder="Writer">
737 </div>
738 </div>
739 <div class="form-group">
```

```

740         <label for="txtEmail" class="col-sm-3 control-label">Email</label>
741         <div class="col-sm-6">
742             <input type="email" class="form-control" id="txtEmail"
743                 placeholder="Email">
744         </div>
745     </div>
746     <div class="form-group">
747         <label for="txtTitle" class="col-sm-3 control-label">제목</label>
748         <div class="col-sm-6">
749             <input type="text" class="form-control" id="txtTitle"
750                 placeholder="Title">
751         </div>
752     </div>
753     <div class="form-group">
754         <label for="txtContents" class="col-sm-3 control-label">내용</label>
755         <div class="col-sm-6">
756             <textarea class="form-control" rows="5" id="txtContents"
757                 placeholder="Contents"></textarea>
758         </div>
759     </div>
760     <div class="form-group">
761         <div class="text-center">
762             <button type="button" class="btn btn-primary" id="btnWrite">글 쓰기</button>
763             <button type="button" class="btn btn-success" id="btnCancel">취소</button>
764         </div>
765     </div>
766 </form>
767 </body>
768 </html>

```

3) BoardController.java 코드 추가

```

770
771 @RequestMapping(value = "/write", method = RequestMethod.POST)
772 @ResponseBody
773 public Map insert(@RequestBody BoardVO board) {
774     Map<String, Object> result = new HashMap<String, Object>();
775     String title = board.getTitle();
776     title = title.replace("<", "&lt;");
777     title = title.replace(">", "&gt;");
778     title = title.replace("'", "");
779     board.setTitle(title);
780
781     String contents = board.getContents();
782     contents = contents.replace("<", "&lt;");
783     contents = contents.replace(">", "&gt;");
784     contents = contents.replace("'", "");
785     contents = contents.replace("\r\n", "<br />");
786     board.setContents(contents);
787     this.boardService.insertBoard(board);
788     result.put("code", Boolean.TRUE);
789     return result;
790 }

```

4) BoardServiceImpl.java 코드 추가

```

794 @Override
795 public void insertBoard(BoardVO board) {

```

```
796     this.boardDao.create(board);
797 }
798
799 5)BoardDaoImpl.java code 추가
800
801 @Override
802 public void create(BoardVO board) {
803     this.sqlSession.insert("Board.insertSP", board);
804 }
805
806 6)Database에 StoredProcedure 추가
807
808 CREATE OR REPLACE PROCEDURE sp_board_insert
809 (
810     v_name          IN          board.name%TYPE,
811     v_email         IN          board.email%TYPE,
812     v_title         IN          board.title%TYPE,
813     v_contents      IN          board.contents%TYPE
814 )
815 IS
816 BEGIN
817     INSERT INTO Board(idx, name, email, title, contents, grp)
818     VALUES(seq_board_idx.NEXTVAL, v_name, v_email, v_title, v_contents,
819            seq_board_idx.CURRVAL);
819     COMMIT;
820 END;
821
822 7)mybatis-config.xml 코드 추가
823
824 <typeAliases>
825     <typeAlias type="com.example.vo.BoardVO" alias="boardVo"/>
826 </typeAliases>
827
828 8)board-mapper.xml 코드 추가
829
830 <parameterMap type="boardVo" id="insertMap">
831     <parameter property="name" javaType="String" jdbcType="VARCHAR" mode="IN"/>
832     <parameter property="email" javaType="String" jdbcType="VARCHAR" mode="IN"/>
833     <parameter property="title" javaType="String" jdbcType="VARCHAR" mode="IN"/>
834     <parameter property="contents" javaType="String" jdbcType="VARCHAR" mode="IN"/>
835 </parameterMap>
836 <insert id="insertSP" parameterType="boardVo" parameterMap="insertMap">
837     { call sp_board_insert(?,?,?,?)}
838 </insert>
839
840 9)project > right-click > Run As > Run on Server > Finish
841
842 10)POSTMAN 으로 테스트
843 POST http://localhost:8080/board/write
844 Body
845     raw > JSON
846     {
847         "name" : "백두산",
848         "email" : "javaexpert@nate.com",
849         "title" : "첫 글입니다",
850         "contents" : "첫 글 내용입니다."
851     }
```

```
852     Send
853     -결과
854     {
855         "code": true
856     }
857
858 11)페이지에서 [새 글쓰기] 클릭하여 write.jsp를 통해 입력이 성공하는지 테스트한다.
859     -Insert Success alert창이 나오고
860     -http://localhost:8080/board/로 redirect 되면 성공이다.
861     -Database에 가서 입력된 것을 확인한다.
```

862

863

864 8. 목록보기

865 1)BoardController.java 코드 추가

866

```
867     @RequestMapping(value = "/", method = RequestMethod.POST)
868     @ResponseBody
869     public Map list() {
870         Map<String, Object> map = new HashMap<String, Object>();
871         this.boardService.select(map);
872         List<BoardVO> list = (List<BoardVO>) map.get("results");
873         Map<String, Object> data = new HashMap<String, Object>();
874         data.put("code", Boolean.TRUE);
875         data.put("data", list);
876         return data;
877     }
878
```

879

880 2)/WEB-INF/views/index.jsp code 추가

881

```
882     <script>
883         $(document).ready(function(){
884             $.ajax({
885                 url : "/board/",
886                 dataType : "json",
887                 method : "POST",
888                 success : function(data){
889                     display(data);
890                 },
891                 error : function(err){
892                     console.log("Error 발생 : " + err);
893                 }
894             });
895             function display(result){
896                 var data = result.data;
897                 var count = data.length;
898                 $("#count").text(count);
899                 var str = null;
900                 if(count == 0) {
901                     str = "<tr>";
902                     str += "<td colspan='5' align='center'>";
903                     str += "글이 없습니다.</td>";
904                     str += "</tr>";
905                 }else if(count > 0){
906                     for(var i = 0 ; i < count ; i++){
907                         var board = data[i];
908                         var email = board.email;
```



```

909         var writedate = board.writedate.substr(0, 10);
910         var grp = board.grp;
911         str += "<tr>";
912         str += "<td style='text-align:center'>" + grp + "</td>" +
913             "<td style='text-align:center'><a href='mailto:" + email + "'>" + board.name +
914             "</a></td>" +
915             "<td>&nbsp;<a href='/board/" + board.idx + "'>" + board.title + "</a></td>" +
916             "<td style='text-align:center'>" + writedate + "</td>" +
917             "<td style='text-align:center'>" + board.readnum + "</td>";
918         str += "</tr>";
919     }
920     $("#records").html(str);
921 }
922 </script>
923

```

3)BoardServiceImpl.java 코드 추가

```

925
926     @Override
927     public void select(Map map) {
928         this.boardDao.readAll(map);
929     }
930

```

4)BoardDaoImpl.java code 추가

```

931
932
933     @Override
934     public void readAll(Map map) {
935         this.sqlSession.selectList("Board.selectAllSP", map);
936     }
937

```

5)Database에 StoredProcedure 추가

```

938
939
940     CREATE OR REPLACE PROCEDURE sp_board_select_all
941     (
942         rec_board      OUT      SYS_REFCURSOR
943     )
944     AS
945     BEGIN
946         OPEN rec_board FOR
947         SELECT idx, name, email, title, TO_CHAR(writedate, 'yyyy-mm-dd') AS writedate, readnum, grp,
948             lev, step
949         FROM Board ORDER BY grp DESC, step ASC;
950     END;
951

```

6)board-mapper.xml 코드 추가

```

952
953     <resultMap type="boardVo" id="selectResultMap">
954         <result column="idx" property="idx" jdbcType="INTEGER" javaType="Integer"/>
955         <result column="name" property="name" jdbcType="VARCHAR" javaType="String"/>
956         <result column="email" property="email" jdbcType="VARCHAR" javaType="String"/>
957         <result column="title" property="title" jdbcType="VARCHAR" javaType="String"/>
958         <result column="contents" property="contents" jdbcType="VARCHAR" javaType="String"/>
959         <result column="writedate" property="writedate" jdbcType="DATE" javaType="String"/>
960         <result column="readnum" property="readnum" jdbcType="SMALLINT" javaType="Integer"/>
961         <result column="grp" property="grp" jdbcType="INTEGER" javaType="Integer"/>
962         <result column="lev" property="lev" jdbcType="TINYINT" javaType="Integer"/>
963         <result column="step" property="step" jdbcType="TINYINT" javaType="Integer"/>
964     </resultMap>
965

```

```

964     </resultMap>
965
966     <parameterMap type="hashMap" id="selectAllMap">
967         <parameter property="results" javaType="ResultSet" jdbcType="CURSOR"
968             mode="OUT" resultMap="selectResultMap"/>
969     </parameterMap>
970     <select id="selectAllSP" parameterMap="selectAllMap" statementType="CALLABLE">
971         { call sp_board_select_all(?) }
972     </select>
973

```

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

8)POSTMAN 으로 테스트

GET localhost:8080/board/

Send

-결과

```

980     {
981         "code": true,
982         "data": [
983             {
984                 "idx": 2,
985                 "readnum": 0,
986                 "grp": 2,
987                 "lev": 0,
988                 "step": 0,
989                 "name": "한라산",
990                 "email": "javaexpert@nate.com",
991                 "title": "두번째 글",
992                 "contents": null,
993                 "writedate": "2018-06-13 09:53:16.0"
994             },
995             {
996                 "idx": 1,
997                 "readnum": 0,
998                 "grp": 1,
999                 "lev": 0,
1000                 "step": 0,
1001                 "name": "백두산",
1002                 "email": "javaexpert@nate.com",
1003                 "title": "첫 글입니다",
1004                 "contents": null,
1005                 "writedate": "2018-06-13 09:31:43.0"
1006             }
1007         ]
1008     }

```

9)정상적으로 게시글 목록이 나오면 성공

9. 한 게시글 보기

1)BoardController.java 코드 추가

```

1016     @RequestMapping(value =("/{idx}", method = RequestMethod.GET)
1017     @ResponseBody
1018     public ModelAndView read(@PathVariable int idx) {
1019         Map<String, Object> map = new HashMap<String, Object>();
1020         map.put("idx", idx);

```

```

1021     this.boardService.selectBoard(map);
1022     List<BoardVO> list = (List<BoardVO>)map.get("result");
1023     ModelAndView mav = new ModelAndView();
1024     BoardVO board = list.get(0);
1025     String title = board.getTitle();
1026     title = title.replace("&gt;", "<");
1027     title = title.replace("&gt;", ">");
1028     title = title.replace("''", "");
1029     board.setTitle(title);
1030     String contents = board.getContents();
1031     contents = contents.replace("&lt;", "<");
1032     contents = contents.replace("&gt;", ">");
1033     contents = contents.replace("''", "");
1034     contents = contents.replace("<br />", "\r\n");
1035     board.setContents(contents);
1036     mav.addObject("data", board);
1037     mav.setViewName("view");
1038     return mav;
1039 }

```

2)/WEB-INF/views/view.jsp 생성하기

```

1043 <%@ page language="java" contentType="text/html; charset=UTF-8"
1044     pageEncoding="UTF-8"%>
1045 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
1046 <c:set var="board" value="${data}" />
1047 <!DOCTYPE html>
1048 <html>
1049 <head>
1050     <meta charset="UTF-8">
1051     <title>게시판 글 확인</title>
1052     <link rel="stylesheet" href="static/css/bootstrap.min.css"
1053         type="text/css" />
1054     <script src="static/js/.min.js"></script>
1055     <script>
1056         var idx = null;
1057         $(document).ready(function() {
1058             idx = ${board.idx};
1059             $("#txtIdx").val("${board.idx}");
1060             $("#txtWritedate").val("${board.writedate}");
1061             $("#txtName").val("${board.name}");
1062             $("#txtEmail").val("${board.email}");
1063             $("#txtReadnum").val("${board.readnum}");
1064             $("#txtTitle").val("${board.title}");
1065
1066             $("#btnCancel").bind("click", function() {
1067                 location.href = "/board/";
1068             });
1069         });
1070     </script>
1071 </head>
1072 <body>
1073     <h1 class="text-center">게시판 글 확인</h1>
1074     <form class="form-horizontal">
1075         <div class="form-group">
1076             <label class="col-sm-3 control-label">글번호</label>
1077             <div class="col-sm-2">

```

```
1078         <input type="text" class="form-control" id="txtIdx" readonly>
1079     </div>
1080     <label class="col-sm-2 control-label">조회수</label>
1081     <div class="col-sm-2">
1082         <input type="text" class="form-control" id="txtReadnum" readonly>
1083     </div>
1084 </div>
1085 <div class="form-group">
1086     <label class="col-sm-3 control-label">작성날짜</label>
1087     <div class="col-sm-6">
1088         <input type="text" class="form-control" id="txtWritedate" readonly>
1089     </div>
1090 </div>
1091 <div class="form-group">
1092     <label for="txtName" class="col-sm-3 control-label">작성자</label>
1093     <div class="col-sm-6">
1094         <input type="text" class="form-control" id="txtName"
1095             placeholder="Writer" readonly>
1096     </div>
1097 </div>
1098 <div class="form-group">
1099     <label for="txtEmail" class="col-sm-3 control-label">Email</label>
1100     <div class="col-sm-6">
1101         <input type="email" class="form-control" id="txtEmail"
1102             placeholder="Email">
1103     </div>
1104 </div>
1105 <div class="form-group">
1106     <label for="txtTitle" class="col-sm-3 control-label">제목</label>
1107     <div class="col-sm-6">
1108         <input type="text" class="form-control" id="txtTitle"
1109             placeholder="Title">
1110     </div>
1111 </div>
1112 <div class="form-group">
1113     <label for="txtContents" class="col-sm-3 control-label">내용</label>
1114     <div class="col-sm-6">
1115         <textarea class="form-control" rows="5" id="txtContents"
1116             placeholder="Contents">${board.contents}</textarea>
1117         <--여기에 넣지 않으면 <br /> 문제 발생하여 오류발생 -->
1118     </div>
1119 </div>
1120 <div class="form-group">
1121     <div class="text-center">
1122         <button type="button" class="btn btn-primary" id="btnUpdate">
1123             <span class="glyphicon glyphicon-signal" aria-hidden="true"></span>
1124             수정하기
1125         </button>
1126         <button type="button" class="btn btn-danger" id="btnDelete">
1127             <span class="glyphicon glyphicon-trash" aria-hidden="true"></span>
1128             삭제하기
1129         </button>
1130         <button type="button" class="btn btn-warning" id="btnAnswer">
1131             <span class="glyphicon glyphicon-object-align-bottom"
1132                 aria-hidden="true"></span> 답변하기
1133         </button>
1134         <button type="button" class="btn btn-success" id="btnCancel">
```

```
1135         <span class="glyphicon glyphicon-remove" aria-hidden="true"></span>
1136         목록으로
1137     </button>
1138 </div>
1139 </div>
1140 </form>
1141 </body>
1142 </html>
1143
1144 3)BoardServiceImpl.java 코드 추가
1145
1146     @Override
1147     public void selectBoard(Map map) {
1148         this.boardDao.read(map);
1149     }
1150
1151 4)BoardDaoImpl.java code 추가
1152
1153     @Override
1154     public void read(Map map) {
1155         this.sqlSession.selectOne("Board.selectOneSP", map);
1156     }
1157
1158 5)Database에 StoredProcedure 추가
1159
1160     CREATE OR REPLACE PROCEDURE sp_board_select_one
1161     (
1162         v_idx          IN      Board.idx%TYPE,
1163         rec_board      OUT      SYS_REFCURSOR
1164     )
1165     AS
1166     BEGIN
1167         OPEN rec_board FOR
1168         SELECT idx, name, email, title, contents, TO_CHAR(writedate, 'yyyy-mm-dd') AS writedate,
1169         readnum, grp, lev, step
1169         FROM Board
1170         WHERE idx = v_idx;
1171     END;
1172
1173 6)board-mapper.xml 코드 추가
1174
1175     <parameterMap type="hashMap" id="selectOneMap">
1176         <parameter property="idx" javaType="Integer"
1177             jdbcType="INTEGER" mode="IN" />
1178         <parameter property="result" javaType="ResultSet"
1179             jdbcType="CURSOR" mode="OUT" resultMap="selectResultMap" />
1180     </parameterMap>
1181     <select id="selectOneSP" parameterMap="selectOneMap"
1182         statementType="CALLABLE">
1183         { call sp_board_select_one(?,?)}
1184     </select>
1185
1186 7)project > right-click > Run As > Run on Server > Finish
1187
1188 8)POSTMAN 으로 테스트
1189     GET localhost:8080/board/board/2
1190     Send
```

```
1191 -결과
1192 <!DOCTYPE html>
1193 <html>
1194 <head>
1195 <meta charset="UTF-8">
1196 <title>게시판 글 확인</title>
1197 <link rel="stylesheet" href="static/css/bootstrap.min.css" type="text/css" />
1198 <script src="static/js/jquery-1.12.4.min.js"></script>
1199 <script>
1200     var idx = null;
1201     $(document).ready(function() {
1202         idx = 2;
1203         $("#txtIdx").val("2");
1204         $("#txtWritedate").val("2018-12-25");
1205         $("#txtName").val("한라산");
1206         $("#txtEmail").val("bbb@bbb.com");
1207         $("#txtReadnum").val("0");
1208         $("#txtTitle").val("두번째 글");
1209
1210         $("#btnCancel").bind("click", function() {
1211             location.href = "/board/";
1212         });
1213     });
1214 </script>
1215 </head>
1216 <body>
1217 <h1 class="text-center">게시판 글 확인</h1>
1218 <form class="form-horizontal">
1219 <div class="form-group">
1220 <label class="col-sm-3 control-label">글번호</label>
1221 <div class="col-sm-2">
1222 <input type="text" class="form-control" id="txtIdx" readonly>
1223 </div>
1224 <label class="col-sm-2 control-label">조회수</label>
1225 <div class="col-sm-2">
1226 <input type="text" class="form-control" id="txtReadnum" readonly>
1227 </div>
1228 </div>
1229 <div class="form-group">
1230 <label class="col-sm-3 control-label">작성날짜</label>
1231 <div class="col-sm-6">
1232 <input type="text" class="form-control" id="txtWritedate" readonly>
1233 </div>
1234 </div>
1235 <div class="form-group">
1236 <label for="txtName" class="col-sm-3 control-label">작성자</label>
1237 <div class="col-sm-6">
1238 <input type="text" class="form-control" id="txtName" placeholder="Writer"
1239     readonly>
1240 </div>
1241 <div class="form-group">
1242 <label for="txtEmail" class="col-sm-3 control-label">Email</label>
1243 <div class="col-sm-6">
1244 <input type="email" class="form-control" id="txtEmail" placeholder="Email">
1245 </div>
```

```

1246     </div>
1247     <div class="form-group">
1248         <label for="txtTitle" class="col-sm-3 control-label">제목</label>
1249         <div class="col-sm-6">
1250             <input type="text" class="form-control" id="txtTitle" placeholder="Title">
1251         </div>
1252     </div>
1253     <div class="form-group">
1254         <label for="txtContents" class="col-sm-3 control-label">내용</label>
1255         <div class="col-sm-6">
1256             <textarea class="form-control" rows="5" id="txtContents"
1257                 placeholder="Contents">두번째
1258                 글
1259                 입니다.</textarea>
1260         </div>
1261     </div>
1262 </div>
1263 <div class="form-group">
1264     <div class="text-center">
1265         <button type="button" class="btn btn-primary" id="btnUpdate">
1266             <span class="glyphicon glyphicon-signal" aria-hidden="true"></span>수정하기
1267         </button>
1268         <button type="button" class="btn btn-danger" id="btnDelete">
1269             <span class="glyphicon glyphicon-trash" aria-hidden="true"></span>삭제하기
1270         </button>
1271         <button type="button" class="btn btn-warning" id="btnAnswer">
1272             <span class="glyphicon glyphicon-object-align-bottom"
1273                 aria-hidden="true"></span> 답변하기
1274         </button>
1275         <button type="button" class="btn btn-success" id="btnCancel">
1276             <span class="glyphicon glyphicon-remove" aria-hidden="true"></span>목록으로
1277         </button>
1278     </div>
1279 </div>
1280 </form>
1281 </body>
1282 </html>

```

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

10. 글 삭제하기

1)BoardController.java 코드 추가

```

1290 @RequestMapping(value =("/{idx}", method = RequestMethod.DELETE)
1291 @ResponseBody
1292 public Map delete(@PathVariable int idx) {
1293     this.boardService.deleteBoard(idx);
1294     Map<String, Object> map = new HashMap<String, Object>();
1295     map.put("code", Boolean.TRUE);
1296     return map;
1297 }

```

2)/WEB-INF/views/view.jsp 코드 수정하기

```

1301 $("#btnDelete").bind("click", function() {
1302     $.ajax({

```

```
1303         url : "/board/" + idx,  
1304         method : "DELETE",  
1305         success : function(data) {  
1306             if (data.code) {  
1307                 alert("Delete Success");  
1308                 location.href = "/board/";  
1309             }  
1310         },  
1311         error : function(err) {  
1312             console.log("Error 발생 : " + err);  
1313         }  
1314     });  
1315 });
```

1317 3)BoardServiceImpl.java 코드 추가

```
1318  
1319     @Override  
1320     public void deleteBoard(int idx) {  
1321         this.boardDao.delete(idx);  
1322     }  
1323
```

1324 4)BoardDaoImpl.java code 추가

```
1325  
1326     @Override  
1327     public void delete(int idx) {  
1328         this.sqlSession.delete("Board.deleteSP", idx);  
1329     }  
1330
```

1331 5)Database에 StoredProcedure 추가

```
1332  
1333     CREATE OR REPLACE PROCEDURE sp_board_delete  
1334     (  
1335         v_idx      IN      Board.idx%TYPE  
1336     )  
1337     IS  
1338     BEGIN  
1339         DELETE FROM Board  
1340         WHERE idx = v_idx;  
1341  
1342         COMMIT;  
1343     END;  
1344
```

1345 6)board-mapper.xml 코드 추가

```
1346  
1347     <delete id="deleteSP" parameterType="Integer">  
1348         { call sp_board_delete(#{idx}) }  
1349     </delete>  
1350
```

1351 7)project > right-click > Run As > Run on Server > Finish

1353 8)POSTMAN 으로 테스트

```
1354     DELETE localhost:8080/board/2  
1355     Send  
1356     -결과  
1357     {  
1358         "code": true  
1359     }
```



```
1360
1361 9)원하는 글 삭제 성공
1362
1363
1364 11. 글 수정하기
1365 1)BoardController.java 코드 추가
1366
1367 @RequestMapping(value = "/{idx}", method = RequestMethod.PUT)
1368 @ResponseBody
1369 public Map update(@PathVariable int idx, @RequestBody BoardVO board) {
1370     board.setIdx(idx);
1371     this.boardService.updateBoard(board);
1372     Map<String, Object> map = new HashMap<String, Object>();
1373     map.put("code", Boolean.TRUE);
1374     return map;
1375 }
1376
1377 2)/WEB-INF/views/view.jsp 코드 수정하기
1378
1379 $("#btnUpdate").click(function() {
1380     $.ajax({
1381         url : "/board/" + idx,
1382         method : "PUT",
1383         data : JSON.stringify({
1384             "email" : $("#txtEmail").val(),
1385             "title" : $("#txtTitle").val(),
1386             "contents" : $("#txtContents").val()
1387         }),
1388         contentType : "application/json; charset=utf-8",
1389         dataType : "json",
1390         success : function(data) {
1391             if (data.code) {
1392                 alert("Update Success");
1393                 location.href = "/board/";
1394             } else {
1395                 alert("Update Failure");
1396             }
1397         },
1398         error : function(err) {
1399             console.log("Error 발생 : " + err);
1400         }
1401     });
1402 });
1403
1404 3)BoardServiceImpl.java 코드 추가
1405
1406 @Override
1407 public void updateBoard(BoardVO board) {
1408     this.boardDao.update(board);
1409 }
1410
1411 4)BoardDaoImpl.java code 추가
1412
1413 @Override
1414 public void update(BoardVO board) {
1415     this.sqlSession.update("Board.updateSP", board);
1416 }
```

5) Database에 StoredProcedure 추가

```
CREATE OR REPLACE PROCEDURE sp_board_update
(
    v_email      IN      Board.email%TYPE,
    v_title      IN      Board.title%TYPE,
    v_contents   IN      Board.contents%TYPE,
    v_idx        IN      Board.idx%TYPE
)
IS
BEGIN
    UPDATE Board SET email = v_email, 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="email" javaType="String" jdbcType="VARCHAR" />
    <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/board/3

Send

-결과에서 다음을 수정

```
{
    "idx" : 3,
    "email" : "devexpert@dreamwiz.com",
    "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++) {
```

```
1474         BoardVO board = new BoardVO();
1475         board.setName("한지민");
1476         board.setEmail("aaa@aaa.com");
1477         board.setTitle(i + "번째 글입니다.");
1478         board.setContents(i + "번째 글입니다.");
1479         this.boardService.insertBoard(board);
1480     }
1481 }
```

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

-한번에 1000개 입력이 안되기 때문에 계속 F5 click해서 Oracle에 1000개 이상의 데이터 들어가는지 확인.

3)index.jsp 코드 추가 및 수정

```
1488 <script>
1489     var page = null;
1490     $(document).ready(function(){
1491         $.ajax({
1492             url : "/board/1",
1493             dataType : "json",
1494             method : "POST",
1495             success : function(data){
1496                 display(data);
1497             },
1498             error : function(err){
1499                 console.log("Error 발생 : " + err);
1500             }
1501         });
1502     });
1503
1504     function goPage(pge){ //page 이동
1505         page = pge;
1506         $.ajax({
1507             url : "/board/" + pge,
1508             dataType : "json",
1509             method : "POST",
1510             success : function(result){
1511                 display(result);
1512             }
1513         });
1514     }
1515
1516     function display(result){
1517         var startPage = result.startPage;
1518         var endPage = result.endPage;
1519         var totalPage = result.totalPage;
1520         var pageSize = result.pageSize;
1521         page = result.page; //current Page
1522         var data = result.data;
1523         var count = data.length;
1524         $("#count").text(count);
1525         var str = null;
1526         if(count == 0) {
1527             str = "<tr>";
1528             str += "<td colspan='5' align='center'>";
1529             str += "글이 없습니다.</td>";
1530             str += "</tr>";
1531         }
```

Page 28 of 47

```
1588         $("##paging").html(pageStr);
1589     }
1590 </script>
1591
1592 4)BoardController.java에서 아래 코드 수정
1593 @RequestMapping(value =("/{page}", method = RequestMethod.POST)
1594 @ResponseBody
1595 public Map list(@PathVariable int page) {
1596     Map<String, Object> countMap = new HashMap<String, Object>();
1597     this.boardService.getTotalCount(countMap);
1598     int totalCount = (Integer) countMap.get("result");
1599     int pageSize = 10;
1600     int totalPage = (totalCount % pageSize == 0) ? totalCount / pageSize : totalCount / pageSize +
1601     1;
1602     if (totalPage < page)
1603         page = totalPage;
1604     int pageCount = 10; // 한 페이지에 뿌려줄 수 있는 갯수는 10개씩
1605
1606     int start = (page - 1) * pageCount + 1; // 11
1607     int end = start + pageCount - 1; // 20
1608
1609     int startPage = ((page - 1) * pageCount / pageCount) + 1; // 2
1610     int endPage = startPage + pageCount - 1; // 11
1611
1612     if (endPage > totalPage)
1613         endPage = totalPage;
1614
1615     Map<String, Object> results = new HashMap<String, Object>(); // 전체 레코드
1616     results.put("start", start);
1617     results.put("end", end);
1618     this.boardService.select(results);
1619     List<BoardVO> list = (List<BoardVO>) results.get("results");
1620     Map<String, Object> map = new HashMap<String, Object>();
1621     map.put("code", Boolean.TRUE);
1622     map.put("startPage", startPage);
1623     map.put("endPage", endPage);
1624     map.put("totalPage", totalPage);
1625     map.put("pageSize", pageSize);
1626     map.put("page", page);
1627     map.put("data", list);
1628     return map;
1629 }
1630
1631 5)BoardServiceImpl.java 코드 수정
1632 @Override
1633 public void getTotalCount(Map map) {
1634     this.boardDao.getTotalCount(map);
1635 }
1636
1637 6)BoardDaoImpl.java 코드 수정
1638 @Override
1639 public void getTotalCount(Map map) {
1640     this.sqlSession.selectOne("Board.selectTotalCountSP", map);
1641 }
1642
1643
```

1644 7)Database Stored Procedure 추가

```
1645  
1646 CREATE OR REPLACE PROCEDURE sp_board_total_count  
1647 (  
1648     v_count    OUT    Number  
1649 )  
1650 IS  
1651 BEGIN  
1652     SELECT COUNT(*) INTO v_count  
1653     FROM Board;  
1654 END;
```

1655
1656 8)board-mapper.xml 추가

```
1657  
1658 <parameterMap type="hashMap" id="selectCountMap">  
1659     <parameter property="result" javaType="Integer" jdbcType="INTEGER" mode="OUT"/>  
1660 </parameterMap>  
1661 <select id="selectTotalCountSP" parameterMap="selectCountMap" statementType="CALLABLE">  
1662     { call sp_board_total_count(?) }  
1663 </select>
```

1664
1665 9)시작페이지와 마지막페이지를 읽어오기 위한 board-mapper.xml 수정

```
1666  
1667 <parameterMap type="hashMap" id="selectAllMap">  
1668     <parameter property="results" javaType="ResultSet" jdbcType="CURSOR"  
1669         mode="OUT" resultMap="selectResultMap"/>  
1670     <parameter property="start" jdbcType="INTEGER" javaType="Integer" mode="IN"/>  
1671     <parameter property="end" jdbcType="INTEGER" javaType="Integer" mode="IN"/>  
1672 </parameterMap>  
1673 <select id="selectAllSP" parameterMap="selectAllMap" statementType="CALLABLE">  
1674     { call sp_board_select_all(?, ?, ?)}  
1675 </select>
```

1676
1677 10)시작페이지와 마지막페이지를 읽어오기 위한 Stored Procedure 수정

```
1678  
1679 CREATE OR REPLACE PROCEDURE sp_board_select_all  
1680 (  
1681     rec_board    OUT    SYS_REFCURSOR,  
1682     v_start      IN     NUMBER,  
1683     v_end        IN     NUMBER  
1684 )  
1685 AS  
1686 BEGIN  
1687     OPEN rec_board FOR  
1688     SELECT B.idx, grp, name, email, title, writedate, readnum, lev, step  
1689     FROM  
1690         (SELECT rownum AS rnum, A.idx, A.grp, A.name, A.email, A.title, A.writedate, A.readnum,  
1691             A.lev, A.step  
1692             FROM  
1693                 (SELECT idx, grp, name, email, title, writedate, readnum, lev, step  
1694                     FROM Board ORDER BY grp DESC, step ASC) A  
1695                 WHERE rownum <= v_end) B  
1696     WHERE B.rnum >= v_start;  
1697 END;
```

1698 11)project > right-click > Run As > Run on Server > Finish

1699

1700 12)한 글 읽기 수정

1701 -BoardController.java 수정

1702 --page가 들어오기 때문

```
1703
1704 @RequestMapping(value = "/{page}/{idx}", method = RequestMethod.GET)
1705 @ResponseBody
1706 public ModelAndView read(@PathVariable int page, @PathVariable int idx) {
1707     Map<String, Object> map = new HashMap<String, Object>();
1708     map.put("idx", idx);
1709     this.boardService.selectBoard(map);
1710     List<BoardVO> list = (List<BoardVO>)map.get("result");
1711     ModelAndView mav = new ModelAndView();
1712     BoardVO board = list.get(0);
1713     String title = board.getTitle();
1714     title = title.replace("&gt;", "<");
1715     title = title.replace("&gt;", ">");
1716     title = title.replace("'", "");
1717     board.setTitle(title);
1718     String contents = board.getContents();
1719     contents = contents.replace("&lt;", "<");
1720     contents = contents.replace("&gt;", ">");
1721     contents = contents.replace("'", "");
1722     contents = contents.replace("<br />", "\r\n");
1723     board.setContents(contents);
1724     mav.addObject("data", board);
1725     mav.addObject("page", page);
1726     mav.setViewName("view");
1727     return mav;
1728 }
```

1729
1730 -view.jsp 수정

1731 --코드 추가

```
1732
1733 <c:set var="board" value="${data}" />
1734 <c:set var="currentPage" value="${page}" /> <---추가
1735 <!DOCTYPE html>
1736 <html>
1737
1738 --경로 수정
1739 <link rel="stylesheet" href="../static/css/bootstrap.min.css" type="text/css" />
1740 <script src="../static/js/jquery-1.12.4.min.js"></script>
1741
1742 --목록으로 가기 수정
1743 $("#btnCancel").bind("click", function() {
1744     location.href = "/board/${currentPage}";
1745 });
```

1746
1747 13)BoardController.java 수정

1748 -아래 코드 수정

1749 --한 글 보기에서 목록으로 돌아오기 위해서는 GET 방식이어야 함.

```
1750
1751 @RequestMapping(value = "/{pge}", method = RequestMethod.GET)
1752 @ResponseBody
1753 public Map list(@PathVariable String page) {
1754     int page = Integer.parseInt(pge);
1755
1756 -index.jsp 수정
```

```
1757
1758     $(document).ready(function(){
1759         $.ajax({
1760             url : "1",           <-- 여기 수정
1761             dataType : "json",
1762             method : "GET",      <---여기 수정
1763             success : function(data){
1764                 display(data);
1765             },
1766             error : function(err){
1767                 console.log("Error 발생 : " + err);
1768             }
1769         });
1770     });
1771
1772     function goPage(pge){ //page 이동
1773         page = pge;
1774         $.ajax({
1775             url : "/board/" + pge,
1776             dataType : "json",
1777             method : "GET",      <--여기 수정
1778             success : function(result){
1779                 display(result);
1780             }
1781         });
1782     }
1783
1784     -view.jsp 수정
1785
1786     $("#btnCancel").bind("click", function() {
1787         history.back();
1788     });
1789
1790
```

13. 답글달기

1)view.jsp 코드 추가

```
1793
1794     $("#btnAnswer").click(function() {
1795         location.href = "../answer/${currentPage}/" + idx;
1796     });
1797
```

2)com.example.controller.AnswerController.java 추가

```
1798
1799     package com.example.controller;
1800
1801
1802     import java.util.HashMap;
1803     import java.util.List;
1804     import java.util.Map;
1805
1806     import org.springframework.beans.factory.annotation.Autowired;
1807     import org.springframework.stereotype.Controller;
1808     import org.springframework.web.bind.annotation.PathVariable;
1809     import org.springframework.web.bind.annotation.RequestMapping;
1810     import org.springframework.web.bind.annotation.RequestMethod;
1811     import org.springframework.web.bind.annotation.ResponseBody;
1812     import org.springframework.web.servlet.ModelAndView;
1813
```



```

1814 import com.example.service.BoardService;
1815 import com.example.vo.BoardVO;
1816
1817 @Controller
1818 public class AnswerController {
1819     @Autowired
1820     BoardService boardService;
1821
1822     @RequestMapping(value = "answer/{page}/{idx}", method = RequestMethod.GET)
1823     @ResponseBody
1824     public ModelAndView answer(@PathVariable int page, @PathVariable int idx) {
1825         Map<String, Object> map = new HashMap<String, Object>();
1826         map.put("idx", idx);
1827         this.boardService.selectBoard(map);
1828         List<BoardVO> list = (List<BoardVO>)map.get("result");
1829         BoardVO board = list.get(0);
1830         String title = board.getTitle();
1831         title = title.replace("&gt;", "<");
1832         title = title.replace("&gt;", ">");
1833         title = title.replace("'", "");
1834         title = "Re : " + title;
1835         board.setTitle(title);
1836
1837         String str = "\r\n\r\n-----";
1838         str += "\r\n>";
1839         String contents = board.getContents();
1840         contents = contents.replace("&lt;", "<");
1841         contents = contents.replace("&gt;", ">");
1842         contents = contents.replace("'", "");
1843         contents = contents.replace("<br />", "\r\n");
1844         str += contents;
1845         board.setContents(str);
1846
1847         ModelAndView mav = new ModelAndView();
1848         mav.addObject("data", list.get(0));
1849         mav.addObject("page", page);
1850         mav.setViewName("answer"); // static/answer.jsp
1851         return mav;
1852     }
1853
1854     @RequestMapping(value = "answer", method = RequestMethod.POST)
1855     @ResponseBody
1856     public Map answer(@RequestBody BoardVO board) {
1857         board.setLev(board.getLev() + 1);
1858         board.setStep(board.getStep() + 1);
1859         this.boardService.answerBoard(board);
1860         Map<String, Object> result = new HashMap<String, Object>();
1861         result.put("code", Boolean.TRUE);
1862         return result;
1863     }
1864 }
1865
1866 2)src/main/webapp/static/answer.jsp
1867
1868 <%@ page language="java" contentType="text/html; charset=UTF-8"
1869     pageEncoding="UTF-8"%>
1870 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

```

```
1871 <c:set var="board" value="${data}" />
1872 <c:set var="currentPage" value="${page}" />
1873 <!DOCTYPE html>
1874 <html>
1875 <head>
1876 <meta charset="UTF-8">
1877 <title>답변하기</title>
1878 <link rel="stylesheet" href="../../static/css/bootstrap.min.css"
1879 type="text/css" />
1880 <script src="../../static/js/jquery.min.js"></script>
1881 <script>
1882 $(document).ready(function(){
1883 $("#btnCancel").bind("click", function(){
1884 history.back();
1885 });
1886 ////답글쓰기
1887 $("#btnWrite").bind("click", function(){
1888 $.ajax({
1889 url : "../../",
1890 dataType : "json",
1891 method : "POST",
1892 data : JSON.stringify({
1893 "name" : $("#txtName").val(),
1894 "email" : $("#txtEmail").val(),
1895 "title" : $("#txtTitle").val(),
1896 "contents" : $("#txtContents").val(),
1897 "grp" : ${board.grp},
1898 "lev" : ${board.lev},
1899 "step" : ${board.step}
1900 }),
1901 contentType : "application/json; charset=utf-8",
1902 success : function(data){
1903 if(data.code){
1904 alert("Answer Insert Success");
1905 location.href = "../../";
1906 }
1907 },
1908 error : function(err){
1909 console.log("Error 발생 : " + err);
1910 }
1911 });
1912 });
1913 });
1914 </script>
1915 </head>
1916 <body>
1917 <h1 class="text-center">게시판 답글 쓰기</h1>
1918 <form class="form-horizontal">
1919 <div class="form-group">
1920 <label for="txtName" class="col-sm-3 control-label">작성자</label>
1921 <div class="col-sm-6">
1922 <input type="text" class="form-control" id="txtName"
1923 placeholder="Writer">
1924 </div>
1925 </div>
1926 <div class="form-group">
1927 <label for="txtEmail" class="col-sm-3 control-label">Email</label>
```

```

1928         <div class="col-sm-6">
1929             <input type="email" class="form-control" id="txtEmail"
1930                 placeholder="Email">
1931         </div>
1932     </div>
1933     <div class="form-group">
1934         <label for="txtTitle" class="col-sm-3 control-label">제목</label>
1935         <div class="col-sm-6">
1936             <input type="text" class="form-control" id="txtTitle"
1937                 placeholder="Title" value="{board.title}">
1938         </div>
1939     </div>
1940     <div class="form-group">
1941         <label for="txtContents" class="col-sm-3 control-label">내용</label>
1942         <div class="col-sm-6">
1943             <textarea class="form-control" rows="5" id="txtContents"
1944                 placeholder="Contents">{board.contents}</textarea>
1945         </div>
1946     </div>
1947     <div class="form-group">
1948         <div class="text-center">
1949             <button type="button" class="btn btn-primary" id="btnWrite">답글 쓰기</button>
1950             <button type="button" class="btn btn-success" id="btnCancel">취소</button>
1951         </div>
1952     </div>
1953 </form>
1954 </body>
1955 </html>

```

3) BoardServiceImpl.java 코드 추가

```

1956
1957
1958     @Override
1959     public void answerBoard(BoardVO board) {
1960         this.boardDao.answer(board);
1961     }
1962

```

4) BoardDaoImpl.java code 추가

```

1963
1964
1965     @Override
1966     public void answer(BoardVO board) {
1967         this.answer_update(board.getGrp(), board.getStep());
1968         this.sqlSession.insert("Board.answerInsertSP", board);
1969     }
1970
1971     private void answer_update(int grp, int step) {
1972         Map<String, Integer> map = new HashMap<String, Integer>();
1973         map.put("grp", grp);
1974         map.put("step", step);
1975         this.sqlSession.update("Board.answerUpdateSP", map);
1976     }
1977

```

5) Database에 StoredProcedure 추가

```

1978
1979
1980     CREATE OR REPLACE PROCEDURE sp_board_answer_update
1981     (
1982         v_grp          IN          board.grp%TYPE,
1983         v_step         IN          board.step%TYPE

```

```

1984 )
1985 IS
1986 BEGIN
1987     UPDATE Board SET step = step + 1
1988     WHERE grp = v_grp AND step >= v_step;
1989     COMMIT;
1990 END;
1991 /
1992
1993 CREATE OR REPLACE PROCEDURE sp_board_answer_insert
1994 (
1995     v_name      IN    Board.name%TYPE,
1996     v_email     IN    Board.email%TYPE,
1997     v_title     IN    Board.title%TYPE,
1998     v_contents  IN    Board.contents%TYPE,
1999     v_grp       IN    Board.grp%TYPE,
2000     v_lev       IN    Board.lev%TYPE,
2001     v_step      IN    Board.step%TYPE
2002 )
2003 IS
2004 BEGIN
2005     INSERT INTO Board(idx, name, email, title, contents, grp, lev, step)
2006     VALUES(seq_board_idx.NEXTVAL, v_name, v_email, v_title, v_contents, v_grp, v_lev, v_step);
2007     COMMIT;
2008 END;
2009 /

```

6)board-mapper.xml 코드 추가

```

2013 <parameterMap type="boardVO" id="answerMap">
2014     <parameter property="name" mode="IN" />
2015     <parameter property="email" mode="IN" />
2016     <parameter property="title" mode="IN" />
2017     <parameter property="contents" mode="IN" />
2018     <parameter property="grp" mode="IN" />
2019     <parameter property="lev" mode="IN" />
2020     <parameter property="step" mode="IN" />
2021 </parameterMap>
2022 <insert id="answerInsertSP" parameterMap="answerMap">
2023     { call sp_board_answer_insert(?,?,?,?,,?) }
2024 </insert>
2025 <update id="answerUpdateSP" parameterType="hashMap">
2026     { call sp_board_answer_update(#{grp}, #{step}) }
2027 </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

2041 -하나의 게시글은 여러개의 댓글을 가지는 1:N관계

2042

2043 CREATE SEQUENCE seq_reply_idx

2044 MAXVALUE 9999

2045

2046 -댓글 테이블을 생성하고, 댓글 참조키를 설정한다.

2047

2048 CREATE TABLE Reply

2049 (

2050 idx NUMBER(4),

2051 parent NUMBER(4) DEFAULT 0, //FK는 null이 가능하기 때문에

2052 writer VARCHAR2(20) NOT NULL,

2053 text VARCHAR2(2000) NOT NULL,

2054 regdate DATE DEFAULT SYSDATE,

2055 updatedate DATE DEFAULT SYSDATE,

2056 CONSTRAINT reply_idx_pk PRIMARY KEY(idx),

2057 CONSTRAINT reply_idx_fk FOREIGN KEY(parent) REFERENCES Board(idx)

2058)

2059

2060 3)댓글을 위한 VO 객체 생성

2061 -com.example.vo.ReplyVO.java

2062

2063 package com.example.vo;

2064

2065 public class ReplyVO {

2066 private int idx;

2067 private int parent;

2068 private String writer;

2069 private String text;

2070 private String regdate;

2071 private String updatedate;

2072

2073 public int getIdx() {

2074 return idx;

2075 }

2076 public void setIdx(int idx) {

2077 this.idx = idx;

2078 }

2079 public int getParent() {

2080 return parent;

2081 }

2082 public void setParent(int parent) {

2083 this.parent = parent;

2084 }

2085 public String getWriter() {

2086 return writer;

2087 }

2088 public void setWriter(String writer) {

2089 this.writer = writer;

2090 }

2091 public String getText() {

2092 return text;

2093 }

2094 public void setText(String text) {

2095 this.text = text;

2096 }

2097 public String getRegdate() {

```
2098         return regdate;
2099     }
2100     public void setRegdate(String regdate) {
2101         this.regdate = regdate;
2102     }
2103     public String getUpdatedate() {
2104         return updatedate;
2105     }
2106     public void setUpdatedate(String updatedate) {
2107         this.updatedate = updatedate;
2108     }
2109     @Override
2110     public String toString() {
2111         return "ReplyVO [idx=" + idx + ", parent=" + parent + ", writer=" + writer + ", text=" +
2112             text + ", regdate="
2113             + regdate + ", updatedate=" + updatedate + "]\n";
2114     }
2115 }
```

4)ReplyDao interface와 ReplyDaoImpl.java 생성

```
2116 -com.example.dao.ReplyDao.java
2117
2118 package com.example.dao;
2119
2120 import java.util.List;
2121
2122 import com.example.vo.ReplyVO;
2123
2124 public interface ReplyDao {
2125     List<ReplyVO> list(int parent);
2126     void create(ReplyVO replyVO);
2127     void update(ReplyVO replyVO);
2128     void delete(int idx);
2129 }
2130
2131 -com.example.dao.ReplyDaoImpl.java
2132
2133 package com.example.dao;
2134
2135 import java.util.List;
2136
2137 import com.example.vo.ReplyVO;
2138
2139 @Repository("replyDao")
2140 public class ReplyDaoImpl implements ReplyDao {
2141     @Autowired
2142     private SqlSession sqlSession;
2143
2144     @Override
2145     public List<ReplyVO> list(int parent) {
2146         // TODO Auto-generated method stub
2147         return null;
2148     }
2149
2150     @Override
2151     public void create(ReplyVO replyVO) {
2152         // TODO Auto-generated method stub
2153     }
```

```
2154
2155     }
2156
2157     @Override
2158     public void update(ReplyVO replyVO) {
2159         // TODO Auto-generated method stub
2160
2161     }
2162
2163     @Override
2164     public void delete(int idx) {
2165         // TODO Auto-generated method stub
2166
2167     }
2168 }
2169
2170 5)config/mybatis-config.xml 코드 추가
2171
2172     <typeAliases>
2173         <typeAlias type="com.example.vo.BoardVO" alias="boardVO"/>
2174         <typeAlias type="com.example.vo.ReplyVO" alias="replyVO"/> <--추가
2175     </typeAliases>
2176
2177 6)config/reply-mapper.xml 추가
2178     <?xml version="1.0" encoding="UTF-8" ?>
2179     <!DOCTYPE mapper
2180         PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
2181         "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
2182
2183     <mapper namespace="Reply">
2184
2185     </mapper>
2186
2187 7)src/main/webapp/WEB-INF/spring/applicationContext.xml 코드 추가
2188
2189     <property name="mapperLocations">
2190         <list>
2191             <value>classpath:board-mapper.xml</value>
2192             <value>classpath:reply-mapper.xml</value>
2193         </list>
2194     </property>
2195
2196 8)ReplyService interface와 ReplyServiceImpl.java 생성
2197     -com.example.service.ReplyService.java
2198
2199     package com.example.service;
2200
2201     import java.util.List;
2202
2203     import com.example.vo.ReplyVO;
2204
2205     public interface ReplyService {
2206         List<ReplyVO> selectReply(int parent);
2207         void insertReply(ReplyVO replyVO);
2208         void updateReply(ReplyVO replyVO);
2209         void deleteReply(int idx);
2210     }
```

```
2211
2212 -com.example.service.ReplyServiceImpl.java
2213
2214 package com.example.service;
2215
2216 import java.util.List;
2217
2218 import org.springframework.beans.factory.annotation.Autowired;
2219 import org.springframework.stereotype.Service;
2220
2221 import com.example.dao.ReplyDao;
2222 import com.example.vo.ReplyVO;
2223
2224 @Service("replyService")
2225 public class ReplyServiceImpl implements ReplyService {
2226     @Autowired
2227     private ReplyDao replyDao;
2228
2229     @Override
2230     public List<ReplyVO> selectReply(int parent) {
2231         // TODO Auto-generated method stub
2232         return null;
2233     }
2234
2235     @Override
2236     public void insertReply(ReplyVO replyVO) {
2237         // TODO Auto-generated method stub
2238     }
2239
2240
2241     @Override
2242     public void updateReply(ReplyVO replyVO) {
2243         // TODO Auto-generated method stub
2244     }
2245
2246
2247     @Override
2248     public void deleteReply(int idx) {
2249         // TODO Auto-generated method stub
2250     }
2251
2252 }
2253
2254
2255 9)com.example.controller.ReplyController.java 생성
2256
2257 package com.example.controller;
2258
2259 import org.springframework.beans.factory.annotation.Autowired;
2260 import org.springframework.stereotype.Controller;
2261
2262 import com.example.service.ReplyService;
2263
2264 @Controller
2265 public class ReplyController {
2266     @Autowired
2267     private ReplyService replyService;
```



```
2268     }
2269
2270 10)댓글 입력하기
2271     -Database에 StoredProcedure 생성하기
2272
2273     CREATE OR REPLACE PROCEDURE sp_reply_insert
2274     (
2275         v_parent    IN    Reply.parent%TYPE,
2276         v_writer    IN    Reply.writer%TYPE,
2277         v_text      IN    Reply.text%TYPE
2278     )
2279     IS
2280     BEGIN
2281         INSERT INTO Reply (idx, parent, writer, text)
2282         VALUES (seq_reply_idx.NEXTVAL, v_parent, v_writer, v_text);
2283         COMMIT;
2284     END;
2285
2286     -ReplyController.java 코드 추가
2287
2288     package com.example.controller;
2289
2290     import java.util.HashMap;
2291     import java.util.Map;
2292
2293     import org.springframework.beans.factory.annotation.Autowired;
2294     import org.springframework.stereotype.Controller;
2295     import org.springframework.web.bind.annotation.RequestBody;
2296     import org.springframework.web.bind.annotation.RequestMapping;
2297     import org.springframework.web.bind.annotation.RequestMethod;
2298     import org.springframework.web.bind.annotation.ResponseBody;
2299
2300     import com.example.service.ReplyService;
2301     import com.example.vo.ReplyVO;
2302
2303     @Controller
2304     public class ReplyController {
2305         @Autowired
2306         private ReplyService replyService;
2307
2308         @RequestMapping(value = "/reply", method = RequestMethod.POST)
2309         @ResponseBody
2310         public Map register(@RequestBody ReplyVO replyVO) {
2311             Map<String, Object> map = new HashMap<String, Object>();
2312             this.replyService.insertReply(replyVO);
2313             map.put("code", Boolean.TRUE);
2314             return map;
2315         }
2316     }
2317
2318     -view.jsp 코드 추가
2319     --body
2320
2321     <hr />
2322     <div style="text-align:center">
2323         
2324         <span style='font-size: 2em'>댓글 작성</span>
```

```

2325     </div>
2326     <form>
2327         <table style="width:30%;margin-left:auto;margin-right:auto" border='0'>
2328             <tr>
2329                 <td rowspan="2"><textarea id="newReplyText" name="text"
2330                     rows="3" cols="50" placeholder="댓글 내용을
2331 입력해주세요"></textarea></td>
2332                 <td><input id="newReplyWriter" name="writer"
2333                     placeholder="댓글작성자..." size="10"></td>
2334             </tr>
2335             <tr>
2336                 <td><button class="btn btn-info" type="button" id="replyWrite"><span
2337                     class="glyphicon glyphicon-comment"
2338                     aria-hidden="true"></span> 댓글쓰기</button>
2339                 </td>
2340             </tr>
2341         </table>
2342     </form>
2343     <div style="text-align:center">
2344         <ul id="replies" style="list-style-type:none">
2345         </ul>
2346     </div>
2347 --script
2348 $(document).ready(function(){
2349     $("#replyWrite").click(function(){
2350         // 화면으로부터 입력 받은 변수값의 처리
2351         var replyText = $("#newReplyText");
2352         var replyWriter = $("#newReplyWriter");
2353
2354         var replyTextVal = replyText.val();
2355         var replyWriterVal = replyWriter.val();
2356
2357         $.ajax({
2358             method : "POST",
2359             url : "/board/reply",
2360             contentType : "application/json;charset=utf-8",
2361             dataType : "json",
2362             data : JSON.stringify({
2363                 "parent" : parseInt(idx),
2364                 "text" : replyTextVal,
2365                 "writer" : replyWriterVal
2366             }),
2367             success : function (result) {
2368                 if(result.code){
2369                     alert("Insert Success");
2370                 }
2371                 //getReplies(); // 댓글 목록 출력 함수 호출
2372                 replyText.val(""); // 댓글 내용 초기화
2373                 replyWriter.val(""); // 댓글 작성자 초기화
2374             }
2375         });
2376     });
2377
2378 -reply-mapper.xml 코드 추가
2379
2380     <parameterMap type="replyVO" id="replyInsertMap">

```

```

2381     <parameter property="parent" javaType="Integer"
2382         jdbcType="INTEGER" mode="IN" />
2383     <parameter property="writer" javaType="String"
2384         jdbcType="VARCHAR" mode="IN" />
2385     <parameter property="text" javaType="String"
2386         jdbcType="VARCHAR" mode="IN" />
2387 </parameterMap>
2388 <insert id="create" parameterMap="replyInsertMap">
2389     { call sp_reply_insert (?, ?, ?) }
2390 </insert>

```

2391
2392 -Insert를 위한 ReplyDaoImpl.java 수정

```

2393
2394     @Override
2395     public void create(ReplyVO replyVO) {
2396         this.sqlSession.insert("Reply.create", replyVO);
2397     }

```

2398
2399 -Insert를 위한 ReplyServiceImpl.java 수정

```

2400
2401     @Override
2402     public void insertReply(ReplyVO replyVO) {
2403         this.replyDao.create(replyVO);
2404     }

```

2405
2406 11)댓글 목록 보기

2407 -Database에 StoredProcedure 생성하기

```

2408
2409     CREATE OR REPLACE PROCEDURE sp_reply_select
2410     (
2411         v_parent      IN      Reply.parent%TYPE,
2412         v_reply        OUT     SYS_REFCURSOR
2413     )
2414     IS
2415     BEGIN
2416         OPEN v_reply FOR
2417         SELECT idx, parent, writer, text, regdate, updatedate
2418         FROM Reply
2419         WHERE parent = v_parent
2420         ORDER BY idx;
2421     END;

```

2422
2423 -ReplyController.java 코드 추가

```

2424
2425     @RequestMapping(value = "/reply/all/{parent}", method = RequestMethod.GET)
2426     @ResponseBody
2427     public Map list(@PathVariable int parent) {
2428         Map<String, Object> map = new HashMap<String, Object>();
2429         List<ReplyVO> list = this.replyService.selectReply(parent);
2430         map.put("code", Boolean.TRUE);
2431         map.put("data", list);
2432         return map;
2433     }

```

2434
2435 -view.jsp 코드 추가

```

2436     --script

```

2437

```

2438     $(document).ready(function(){
2439         idx = ${board.idx};
2440         getReplies();          <--코드 추가
2441
2442         ...
2443         ...
2444         success : function (result) {
2445             if(result.code){
2446                 alert("Insert Success");
2447             }
2448             getReplies(); //주석 해제
2449             ...
2450             ...
2451         function getReplies() {
2452             $.getJSON("/board/reply/all/" + idx, function (result) {
2453                 var data = result.data;
2454                 var str = "";
2455                 $(data).each(function () {
2456                     str += "<li class='replyLi' id='" + this.idx + "'>"
2457                         + "<p class='text'>" + this.text + "</p>"
2458                         + "<p class='writer'>" + this.writer + "</p>"
2459                         + "<button type='button' class='update'
2460                             style='background-color:green;color:white;font-weight:bold;'>댓글 수정</button>"
2461                         + "<button type='button' class='delete'
2462                             style='background-color:red;color:white;font-weight:bold;'>댓글 삭제</button>"
2463                         + "</li>"
2464                         + "<hr/>";
2465                 });
2466                 $("#replies").html(str);
2467             });
2468         }
2469     });

```

-reply-mapper.xml 코드 추가

```

2470     <resultMap id="replyResultMap" type="replyVO">
2471         <id property="idx" column="idx" />
2472         <result property="parent" column="parent" />
2473         <result property="writer" column="writer" />
2474         <result property="text" column="text" />
2475         <result property="regdate" column="regdate" />
2476         <result property="updatedate" column="updatedate" />
2477     </resultMap>
2478
2479     <parameterMap type="hashMap" id="replySelectMap">
2480         <parameter property="parent" mode="IN"
2481             javaType="Integer" jdbcType="INTEGER"/>
2482         <parameter property="results" mode="OUT"
2483             javaType="ResultSet" jdbcType="CURSOR" resultMap="replyResultMap"/>
2484     </parameterMap>
2485     <select id="list" parameterMap="replySelectMap" statementType="CALLABLE">
2486         { call sp_reply_select(?, ?) }
2487     </select>
2488

```

-Select를 위한 ReplyDaoImpl.java 수정

```

2490
2491     @Override
2492     public List<ReplyVO> list(int parent) {

```

```

2493     Map<String, Object> result = new HashMap<String, Object>();
2494     result.put("parent", parent);
2495     this.sqlSession.selectList("Reply.list", result);
2496     List<ReplyVO> list = (List<ReplyVO>)result.get("results");
2497     return list;
2498 }
2499

```

-Select를 위한 ReplyServiceImpl.java 수정

```

2501
2502 @Override
2503 public List<ReplyVO> selectReply(int parent) {
2504     return this.replyDao.list(parent);
2505 }
2506

```

12)댓글 수정하기

-view.jsp 코드 추가

```

2509 --코드에서 주목해서 봐야할 것은 클릭 이벤트 선택자가 <ul>의 id인 replies로 되어있다는 점이다.
2510 --단순히 클릭이 발생한 수정버튼을 선택자로 지정하면 될 것 같지만 Ajax통신 후에 생기는 요소이기 때문에 이벤트처리가 되
    지 않는다.
2511 --그래서 Ajax통신 이전에 존재하는 <ul>을 이용해서 이벤트를 등록해야만 한다.
2512

```

```

2513 var replyIdx = null;
2514 var replyUpdateFlag = true;
2515 $("#replies").on("click", ".replyLi button", function () {
2516     if(replyUpdateFlag){
2517         var replyLi = $(this).parent();
2518
2519         replyIdx = replyLi.attr("id");
2520         var replyText = replyLi.find(".text").text();
2521         var replyWriter = replyLi.find(".writer").text();
2522
2523         var replyStr = "<textarea id='updateReplyText' rows='3' cols='60'>" + replyText +
            "</textarea>";
2524         replyLi.find(".text").replaceWith(replyStr);
2525         $(this).text("댓글 수정 하기");
2526         replyUpdateFlag = false;
2527     }else{
2528         var replyText = $(this).parent().find("#updateReplyText").val();
2529         $(this).text("댓글 수정");
2530         $.ajax({
2531             url : "/board/reply",
2532             method : "PUT",
2533             dataType : "json",
2534             contentType : "application/json;charset=utf-8",
2535             data : JSON.stringify({
2536                 "idx" : parseInt(replyIdx),
2537                 "text" : replyText
2538             }),
2539             success : function(result){
2540                 getReplies();
2541                 replyUpdateFlag = true;
2542             }
2543         });
2544     }
2545 });
2546

```

-Database StoredProcedure 추가하기

```

2548
2549 CREATE OR REPLACE PROCEDURE sp_reply_update
2550 (
2551     v_text    IN    Reply.text%TYPE,
2552     v_idx     IN    Reply.idx%TYPE
2553 )
2554 IS
2555 BEGIN
2556     UPDATE Reply SET text = v_text, updatedate = SYSDATE
2557     WHERE idx = v_idx;
2558     COMMIT;
2559 END;
2560
2561 -reply-mapper.xml 코드 추가
2562
2563 <parameterMap type="hashMap" id="replyUpdateMap">
2564     <parameter property="text" javaType="String"
2565         jdbcType="VARCHAR" mode="IN"/>
2566     <parameter property="idx" javaType="Integer"
2567         jdbcType="INTEGER" mode="IN"/>
2568 </parameterMap>
2569 <update id="update" parameterMap="replyUpdateMap">
2570     { call sp_reply_update(?, ?) }
2571 </update>
2572
2573 -ReplyDaoImpl.java 코드 수정
2574
2575 @Override
2576 public void update(ReplyVO replyVO) {
2577     Map<String, Object> map = new HashMap<String, Object>();
2578     map.put("text", replyVO.getText());
2579     map.put("idx", replyVO.getIdx());
2580     this.sqlSession.update("Reply.update", map);
2581 }
2582
2583 -ReplyServiceImpl.java 코드 수정
2584
2585 @Override
2586 public void updateReply(ReplyVO replyVO) {
2587     this.replyDao.update(replyVO);
2588 }
2589
2590 -ReplyController.java 코드 추가
2591
2592 @RequestMapping(value = "/reply", method = RequestMethod.PUT)
2593 @ResponseBody
2594 public Map update(@RequestBody ReplyVO replyVO) {
2595     Map<String, Object> map = new HashMap<String, Object>();
2596     this.replyService.updateReply(replyVO);
2597     System.out.println(replyVO);
2598     map.put("code", Boolean.TRUE);
2599     return map;
2600 }
2601
2602 13)댓글 삭제하기
2603 -view.jsp 코드 추가
2604 $("#replies").on("click", ".replyLi .delete", function () {

```

```
2605     var replyIdx = $(this).parent().attr("id");
2606     $.ajax({
2607         url : "/board/reply/" + parseInt(replyIdx),
2608         method : "DELETE",
2609         dataType : "json",
2610         success : function(result){
2611             getReplies();
2612         }
2613     });
2614 });
2615
2616 -Database StoredProcedure 추가하기
2617
2618 CREATE OR REPLACE PROCEDURE sp_reply_delete
2619 (
2620     v_idx IN Reply.idx%TYPE
2621 )
2622 IS
2623 BEGIN
2624     DELETE FROM Reply WHERE idx = v_idx;
2625     COMMIT;
2626 END;
2627
2628 -reply-mapper.xml 코드 추가
2629
2630 <delete id="delete" parameterType="Integer">
2631     { call sp_reply_delete(#{idx}) }
2632 </delete>
2633
2634 -ReplyDaoImpl.java 코드 수정
2635
2636 @Override
2637 public void delete(int idx) {
2638     this.sqlSession.delete("Reply.delete", idx);
2639 }
2640
2641 -ReplyServiceImpl.java 코드 수정
2642
2643 @Override
2644 public void deleteReply(int idx) {
2645     this.replyDao.delete(idx);
2646 }
2647
2648 -ReplyController.java 코드 추가
2649
2650 @RequestMapping(value = "/reply/{idx}", method = RequestMethod.DELETE)
2651 @ResponseBody
2652 public Map delete(@PathVariable int idx) {
2653     Map<String, Object> map = new HashMap<String, Object>();
2654     this.replyService.deleteReply(idx);
2655     map.put("code", Boolean.TRUE);
2656     return map;
2657 }
```