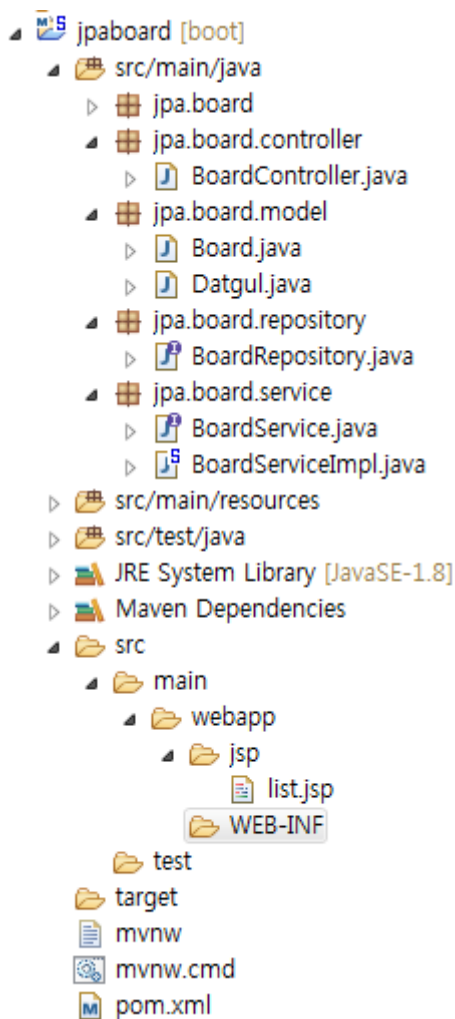


## 게시판 리스트보기(Repository, Service, Controller, 부트메인, JSP, 페이징 적용, Bootstrap 적용)

<http://ojc.asia>, <http://ojcedu.com>

- Spring JDBC 또는 MyBatis로 만들 때 보다 쉽고 빠르게 작성할 수 있다.
- 프로젝트 전체 구조는 다음과 같다.



### [BoardRepository.java]-컨트롤러

- 기본적으로 제공하는 JpaRepository를 상속받아 만들면 된다.
- JpaRepository는 PagingAndSortingRepository를 상속받았고 PagingAndSortingRepository는 CrudRepository(기본적인 CRUD 기능을 제공한다)를 상속 받았으며 다시 CrudRepository는 Repository 인터페이스를 상속받았다.

- 그러므로 JpaRepository는 모든 기능을 다 사용할 수 있고 추가적으로 영속성 컨텍스트에 flush하거나 엔티티를 배치로 삭제할 수 있다.
- 기본 기능만으로도 게시판 기능을 구현할 수 있으므로 JpaRepository를 상속 받자.

```
package jpa.board.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import jpa.board.model.Board;

public interface BoardRepository extends JpaRepository<Board, Integer> {

}
```

- 서비스쪽 클래스를 작성하자.

#### [BoardService.java]

```
package jpa.board.service;

import org.springframework.data.domain.Page;

import jpa.board.model.Board;

public interface BoardService {

    //게시판 리스트 보기
    public Page<Board> findAll(Integer curPage);

}
```

- JpaRepository의 findAll() 메소드를 호출만 하면 되는데 페이징 기능을 구현하기 위해 PageRequest를 만들고 이를 findAll() 메소드의 인자로 넣어주면 된다.
- Board 테이블에서 정렬한 칼럼이 두개 이상이므로 Sort 를 new하면서 Order를 필요한 만큼 생성해주면 되고 ASC는 오름차순, DESC는 내림차순, 그뒤의 문자열은 Board 엔티티의 속성이다.

#### [BoardServiceImpl.java]

```
package jpa.board.service;
```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;
import org.springframework.data.domain.Sort;
import org.springframework.data.domain.Sort.Direction;
import org.springframework.data.domain.Sort.Order;
import org.springframework.stereotype.Service;

import jpa.board.model.Board;
import jpa.board.repository.BoardRepository;

@Service
public class BoardServiceImpl implements BoardService {

    @Autowired
    BoardRepository boardRepository;

    @Override
    //-----
    // 게시판 리스트 보기, 한페이지에 5개씩
    // curPage:요청하는 페이지, 첫페이지는 0
    //-----
    public Page<Board> findAll(Integer curPage) {
        PageRequest pr = new PageRequest(curPage, 5,
            new Sort(
                new Order(Direction.DISC, "reply"),
                new Order(Direction.ASC, "replystep")));
        return boardRepository.findAll(pr);
    }
}

```

- 컨트롤러를 작성하자.

#### [BoardController.java]

```

package jpa.board.controller;
import org.springframework.beans.factory.annotation.Autowired;

```

```

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import jpa.board.model.Board;
import jpa.board.service.BoardService;

@Controller
@RequestMapping("/board")
public class BoardController {

    @Autowired
    BoardService boardService;

    //-----
    // 루트요청(localhost:8080/board/)시 리스트 보기로
    //-----
    @RequestMapping(value="/", method = RequestMethod.GET)
    public String index() {
        return "redirect:/board/list/0";
    }

    //-----
    // 게시판 리스트 보기
    //-----
    @RequestMapping(value="/list/{curPage}", method = RequestMethod.GET)
    public String findAll(Model model, Pageable pageable, @PathVariable Integer curPage) {

        Page<Board> page = boardService.findAll(curPage);
        model.addAttribute("page", page);
        return "list";
    }
}

```

- src/main/webapp/jsp 폴더를 만들고 menu.jsp, header.jsp, list.jsp를 만들자.

### [list.jsp]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Spring JPD 게시판</title>
<style type="text/css">
.tooltip {
    position: absolute;
    z-index: 3;
    background: #E8E8E8;
    border: 1px solid #434343;
    padding: 3px;
}
</style>
</head>
<body>
    <div style="width: 500px;">
        <div style="float: center;">
            <H3 align="center">
                Spring Data JPA 게시판<br>(글리스트보기)
            </H3>
            <h5>총 ${page.totalElements}건</h5>
            <table width="600" border="1" align="left">
                <tr align="left">
                    <td width="7%" align="center">순번</td>
                    <td width="7%" align="center">글번호</td>
                    <td width="41%" align="center">제목</td>
                    <td width="15%" align="center">글쓴이</td>
                    <td width="20%" align="center">등록일</td>
                    <td width="10%" align="center">조회수</td>
                </tr>
                <c:forEach          var="board"          items="${page.content}"
varStatus="loop">
                    <tr>
                        <td align="center">${loop.index + 1}</td>
```

```

<td align="center"><a
href="view?id=${board.id }">${board.id }</a></td>
<td align="left"><c:forEach var="i" begin="1"
end="${board.replylevel}" step="1">
<!-- 레벨의 수만큼 글을 뒤
로 민다 -->
<br><br>
</c:forEach>
<a
href="view?id=${board.id}">${board.title }</a></td>
<td align="center">${board.name}</td>
<td align="center">
<fmt:formatDate
value="${board.regdate}" type="date" dateStyle="default" />
</td>
<td align="center">${board.readcount}</td>
</tr>
</c:forEach>
</table>
<div>
<c:if test="${!page.first}">
<a href="/board/list/${page.number-
1}">Previous</a>
</c:if>
<c:if test="${!page.last}">
<a
href="/board/list/${page.number+1}">Next</a>
</c:if>
<br><br>
<input type="button" value="글쓰기"
onclick="location.href='/write'"></td>
</div>
</div>
</div>
</body>
</html>

```

## [JpaboardApplication.java] - 스프링 부트 메인

```
package jpa.board;
```

```

import java.util.Date;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import jpa.board.model.Board;
import jpa.board.repository.BoardRepository;

@SpringBootApplication
public class JpaboardApplication implements CommandLineRunner{

    @Autowired
    BoardRepository boardRepository;

    public static void main(String[] args) {
        SpringApplication.run(JpaboardApplication.class, args);
    }

    public void run(String...args) {

        //테스트를 위해 글6개 입력
        Board b1 = new Board();
        b1.setContent("JPA강좌 추천해 주세요1~");
        b1.setName("홍길동1");           b1.setPasswd("1111");
        b1.setReadcount(0);              b1.setRegdate(new Date());
        b1.setReply(1);                  b1.setReplylevel(0);
        b1.setReplystep(0);              b1.setTitle("강좌추천요망1");

        boardRepository.save(b1);

        Board b2 = new Board();
        b2.setContent("JPA강좌 추천해 주세요2~");
        b2.setName("홍길동2");           b2.setPasswd("1111");
        b2.setReadcount(0);              b2.setRegdate(new Date());
        b2.setReply(2);                  b2.setReplylevel(0);
        b2.setReplystep(0);              b2.setTitle("강좌추천요망2");
    }
}

```

```
boardRepository.save(b2);
```

```
Board b3 = new Board();
```

```
b3.setContent("JPA강좌 추천해 주세요3~");
```

```
b3.setName("홍길동3");          b3.setPasswd("1111");
```

```
b3.setReadcount(0);          b3.setRegdate(new Date());
```

```
b3.setReply(3);          b3.setReplylevel(0);
```

```
b3.setReplystep(0);          b3.setTitle("강좌추천요망3");
```

```
boardRepository.save(b3);
```

```
Board b4 = new Board();
```

```
b4.setContent("OJC로 가세요...");
```

```
b4.setName("홍길동4");          b4.setPasswd("1111");
```

```
b4.setReadcount(0);          b4.setRegdate(new Date());
```

```
b4.setReply(6);          b4.setReplylevel(1);
```

```
b4.setReplystep(1);          b4.setTitle("[답변]강좌추천요망6");
```

```
boardRepository.save(b4);
```

```
Board b5 = new Board();
```

```
b5.setContent("OJC로 가세요...");
```

```
b5.setName("홍길동5");          b5.setPasswd("1111");
```

```
b5.setReadcount(0);          b5.setRegdate(new Date());
```

```
b5.setReply(2);          b5.setReplylevel(1);
```

```
b5.setReplystep(1);          b5.setTitle("[답변]강좌추천요망2");
```

```
boardRepository.save(b5);
```

```
Board b6 = new Board();
```

```
b6.setContent("JPA강좌 추천해 주세요6~");
```

```
b6.setName("홍길동6");          b6.setPasswd("1111");
```

```
b6.setReadcount(0);          b6.setRegdate(new Date());
```

```
b6.setReply(6);          b6.setReplylevel(0);
```

```
b6.setReplystep(0);          b6.setTitle("강좌추천요망6");
```

```
boardRepository.save(b6);
```



```
}
}
```

- 실행(<http://localhost/board/>) - 리스트 보기에는 페이지 기능이 적용되어 있다.



Spring JPA 게시판

localhost:8080/board/list/0

### Spring Data JPA 게시판 (글리스트보기)

총 6건

순번	글번호	제목	글쓴이	등록일	조회수
1	<a href="#">6</a>	<a href="#">강좌추천요망6</a>	홍길동6	2016. 3. 11	0
2	<a href="#">4</a>	<a href="#">[답변]강좌추천요망6</a>	홍길동4	2016. 3. 11	0
3	<a href="#">3</a>	<a href="#">강좌추천요망3</a>	홍길동3	2016. 3. 11	0
4	<a href="#">2</a>	<a href="#">강좌추천요망2</a>	홍길동2	2016. 3. 11	0
5	<a href="#">5</a>	<a href="#">[답변]강좌추천요망2</a>	홍길동5	2016. 3. 11	0

[Next](#)

[글쓰기](#)