[VO]

```java
package rK.smetsysesor.hye.VO;

import java.util.Date;

public class BaseBoardVO {

	private int idx; // -- IDX
	private int num; // --  재정렬  위한  컬럼
	private String title; // -- TITLE
	private String content; // -- CONTENT
	private String writer; // -- WRTIER
	private Date regDate; // -- REG_DATE
	private Date modDate; // -- MOD_DATE
	private int view; // -- VIEWCOUNT(조회수)

	public int getIdx() { return idx; }
	public void setIdx(int idx) { this.idx = idx; }

	public int getNum() { return num; }
	public void setNum(int num) { this.num = num; }

	public String getTitle() { return title; }
	public void setTitle(String title) { this.title = title; }

	public String getContent() { return content; }
	public void setContent(String content) { this.content = content; }

	public String getWriter() { return writer; }
	public void setWriter(String writer) { this.writer = writer; }

	public Date getRegDate() { return regDate; }
	public void setRegDate(Date regDate) { this.regDate = regDate; }

	public Date getModDate() { return modDate; }
	public void setModDate(Date modDate) { this.modDate = modDate; }

	public int getView() { return view; }
	public void setView(int view) { this.view = view; }

}
```

[1. 게시판 만들기]

```
<mapper namespace="rK.smetsysesor.hye.mapper.BaseMapper">

<!-- public ArrayList<BaseBoardVO> getList(); -->
<select id="getList" resultType="rK.smetsysesor.hye.VO.BaseBoardVO">
	SELECT TITLE, WRITER, REGDATE FROM BASEBOARD ORDER BY IDX DESC;
</select>
```

```
	@Override
	// public ArrayList<BaseBoardVO> getList();
	public ArrayList<BaseBoardVO> getList() {
		return baseMapper.getList();
	}
	// 글 목록
	@RequestMapping(value="/", method= RequestMethod.GET)
	public String list(Model model) {
		model.addAttribute("boardList", baseService.getList());
		return "list";
	}
```

```
<!-- public void write(BaseBoardVO baseBoardVO); -->
<insert id="write">
	INSERT INTO BASEBOARD (TITLE, CONTENT, WRITER, REGDATE)
	VALUES (#{title}, #{content}, "ADMIN", now())
</insert>
```

```
	@Override
	// public void write(BaseBoardVO baseBoardVO);
	public void write(BaseBoardVO baseBoardVO) {
		baseMapper.write(baseBoardVO);
	}
	// 글 작성
	@RequestMapping(value="/write", method= RequestMethod.GET)
	public void write() {}
	// 글 작성 후
	@RequestMapping(value="/write", method= RequestMethod.POST)
	public String write(BaseBoardVO baseBoardVO, RedirectAttributes rttr) {
		baseService.write(baseBoardVO);
		// rttr.addFlashAttribute("result", baseBoardVO.getBoardInx());
		return "redirect:/";
	}
```

```
<!-- public BaseBoardVO read(int idx); -->
<select id="view" resultType="rK.smetsysesor.hye.VO.BaseBoardVO">
	SELECT IDX, TITLE, CONTENT FROM BASEBOARD WHERE IDX = #{idx}
</select>

	@Override
	// public BaseBoardVO view(int idx);
	public BaseBoardVO view(int idx) {
		return baseMapper.view(idx);
	}

@RequestMapping(value= {"/view", "/update"}, method=RequestMethod.GET)
public void view(@RequestParam("idx") int idx, Model model) {
	// BaseBoardVO print = baseService.read(idx);
	// System.out.println(print.getContent());
	model.addAttribute("board1", baseService.view(idx));
}

<!-- public void delete(int idx); -->
<delete id="delete">
	DELETE FROM BASEBOARD WHERE IDX = #{idx}
</delete>

	@Override
	// public void delete(int idx);
	public void delete(int idx) {
		baseMapper.delete(idx);
	}

@RequestMapping(value="/delete", method=RequestMethod.POST)
public String delete(@RequestParam("idx") int idx, RedirectAttributes rttr) {
	baseService.delete(idx);
	return "redirect:/";
}
```

```
<!-- public int update(BaseBoardVO baseBoardVO); -->
<update id="update" >
	UPDATE BASEBOARD
	SET TITLE = #{title}, CONTENT = #{content}, MODDATE = now()
	WHERE IDX = #{idx}
</update>

	@Override
	// public boolean update(BaseBoardVO baseBoardVO);
	public boolean update(BaseBoardVO baseBoardVO) {
		return baseMapper.update(baseBoardVO) == 1;
	}

	@RequestMapping(value="/update", method=RequestMethod.POST)
	public String update(BaseBoardVO baseBoardVO) {
		baseService.update(baseBoardVO);
	return "redirect:/";
	}
```

[PagingBase]

```
package rK.smetsysesor.hye.VO;

public class PagingBase {
	private int page; // 페이지번호
	private int perPageNum; // 페이지당 보여지는 게시물 수
	private int pageStart; // 시작 번호 1, 11, 21

	public PagingBase() { this.page = 1;
				this.perPageNum = 10; }
	public int getPage() { return page; }
	public void setPage(int page) {
		if (page <= 0) { this.page = 1; return; }
		this.page = page;
	}
	public int getPageStart() { //시작번호 = (페이지번호 - 1)*페이지당 보여지는 개수
		pageStart = (this.page - 1) * perPageNum;
		return pageStart;
	}
	public void setPageStart(int pageStart) { this.pageStart = pageStart; }
	public int getPerPageNum() { return this.perPageNum; }
	public void setPerPageNum(int perPageNum) {
		if (perPageNum <= 0 || perPageNum > 100) {
			this.perPageNum = 10; return;
		}
		this.perPageNum = perPageNum;
	}
	@Override
	public String toString() {
		return "Criteria [page=" + page + ", perPageNum=" + perPageNum + ",
			pageStart=" + pageStart + "]";
	}

}
```

[PagingCalc]

```
package rK.smetsysesor.hye.VO;

import org.springframework.web.util.UriComponents;
import org.springframework.web.util.UriComponentsBuilder;

public class PagingCalc {
	private PagingBase pagingBase;
	private int totalCount;
	private int startPage;
	private int endPage;
	private boolean prev;
	private boolean next;
	private int displayPageNum = 10; // 화면에 보여지는 페이지 번호의 숫자.

	public PagingBase getPagingBase() { return pagingBase; }
	public void setPagingBase(PagingBase pagingBase) {
		this.pagingBase = pagingBase;
	}
	public int getTotalCount() { return totalCount; }
	public void setTotalCount(int totalCount) {
		this.totalCount = totalCount;
		calcData(); // 한 라인에 보여줄 displayPAgeNum 계산 (1-10, 11-20, 21-30)
	}
	public int getStartPage() { return startPage; }
	public void setStartPage(int startPage) { this.startPage = startPage; }

	public int getEndPage() { return endPage; }
	public void setEndPage(int endPage) { this.endPage = endPage; }

	public boolean isPrev() { return prev; }

	public boolean isNext() { return next; }

	public int getDisplayPageNum() { return displayPageNum; }
```

```
// 한 라인에 보여줄 displayPAgeNum 계산 (1-10, 11-20, 21-30)
private void calcData() {
// 페이지 나열자의 10번째 번호 =현재페이지/보여줄 페이지수*보여줄 페이지수
        endPage = (int) (Math.ceil(pagingBase.getPage() / (double) displayPageNum)
                        * displayPageNum);
        startPage = (endPage - displayPageNum) + 1;
                // endPage가 10 단위가 아닐 때
        int tempEndPage = (int) (Math.ceil(totalCount
                                / (double) pagingBase.getPerPageNum()));

        if (endPage > tempEndPage) {
                endPage = tempEndPage;
        }

        prev = ( startPage == 1 ) ? false : true;
        next = ( endPage * pagingBase.getPerPageNum() >= totalCount ) ? false : true;
}

public String makeQuery(int page) {
        UriComponents uriComponents
        = UriComponentsBuilder.newInstance()
          .queryParam("page", page)
          .queryParam("perPageNum", pagingBase.getPerPageNum()).build();

        String strTemp = uriComponents.toUriString();

        return uriComponents.toUriString();
        }
}
```

[2. 페이징]

```
<!-- public ArrayList<BaseBoardVO> listCriteria(BaseBoardVO baseBoardVO); -->
<select id="listCriteria" resultType="rK.smetsysesor.hye.VO.BaseBoardVO">
	SELECT IDX, TITLE, CONTENT, REGDATE
	FROM BASEBOARD
	ORDER BY IDX DESC
	LIMIT #{pageStart}, #{perPageNum}
</select>

	@Override
	// public ArrayList<BaseBoardVO> listCriteria(BaseBoardVO baseBoardVO);
	public ArrayList<BaseBoardVO> listCriteria(PagingBase pagingBase) {
		return baseMapper.listCriteria(pagingBase);
	}

// 글 목록 + 게시글 10개 처리
// public ArrayList<BaseBoardVO> listCriteria(BaseBoardVO baseBoardVO);
@RequestMapping(value="/listCriteria", method= RequestMethod.GET)
public String list(PagingBase pagingBase, Model model) {
	model.addAttribute("boardList", baseService.listCriteria(pagingBase));
	return "list";


<!-- public int countIdx(PagingBase pagingBase); -->
<select id="countIdx" resultType="int">
	SELECT COUNT(IDX) FROM BASEBOARD
</select>

	@Override
	// public int countBno(BaseBoardVO baseBoardVO);
	public int countIdx(PagingBase pagingBase) {
		return baseMapper.countIdx(pagingBase);
	}
```

```
<!-- public ArrayList<BaseBoardVO> listPage(int page); -->
<select id="listPage" resultType="rK.smetsysesor.hye.VO.BaseBoardVO">
        SELECT * FROM BASEBOARD
          ORDER BY IDX DESC
          LIMIT #{page}, 10
</select>


        @Override
        // public ArrayList<BaseBoardVO> listPage(int page);
        public ArrayList<BaseBoardVO> listPage(int page) {
                return baseMapper.listPage(page);
        }


// 글 목록 + 페이징 처리
// public String list(Model model) { ... } -- 주석처리 필요
// public ArrayList<BaseBoardVO> listPage(int page);
@RequestMapping(value="/", method=RequestMethod.GET)
public String listPaging(PagingBase pagingBase, Model model) {
        model.addAttribute("boardList", baseService.listCriteria(pagingBase));
        PagingCalc pagingCalc = new PagingCalc();
        pagingCalc.setPagingBase(pagingBase);

        int totalCount = baseService.countIdx(pagingBase);
        pagingCalc.setTotalCount(totalCount);

        model.addAttribute("pagingCalc", pagingCalc);
        return "list";
}
</mapper>
```