NICE D&R

Nicelib Framework 개발가이드

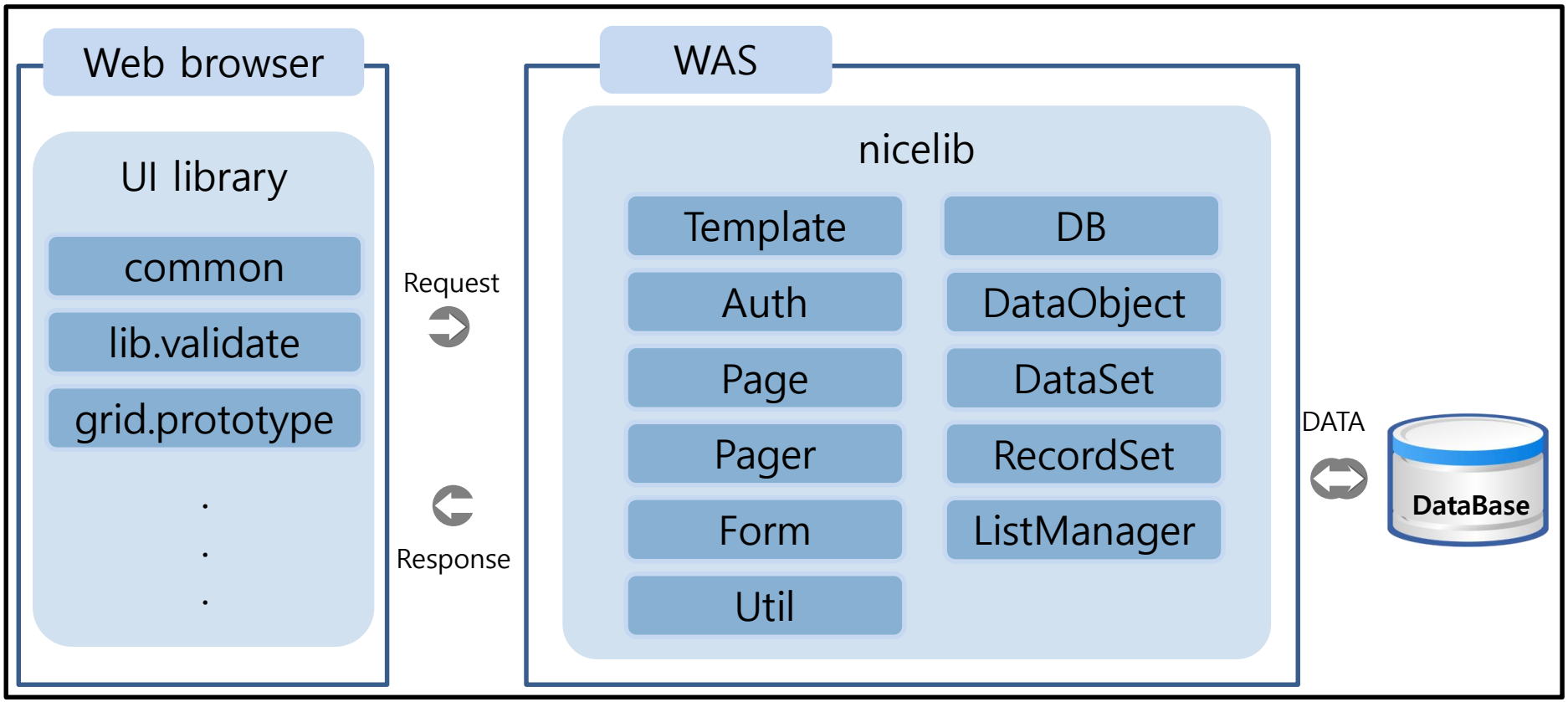
1.Nicelib Framework 소개

원시함수 기반의 Java라이브라리를 이용 하여 중,소 규모의 웹사이트 개발을 쉽고 빠르게 진행 하기 위하여 만들어진 Framework 입니다.

Java기반의 개발 방식 보다는 ASP 또는 PHP와 같은 스크립트 언어 처럼 프로그래밍 할 수 있도록 설계 되었습니다.

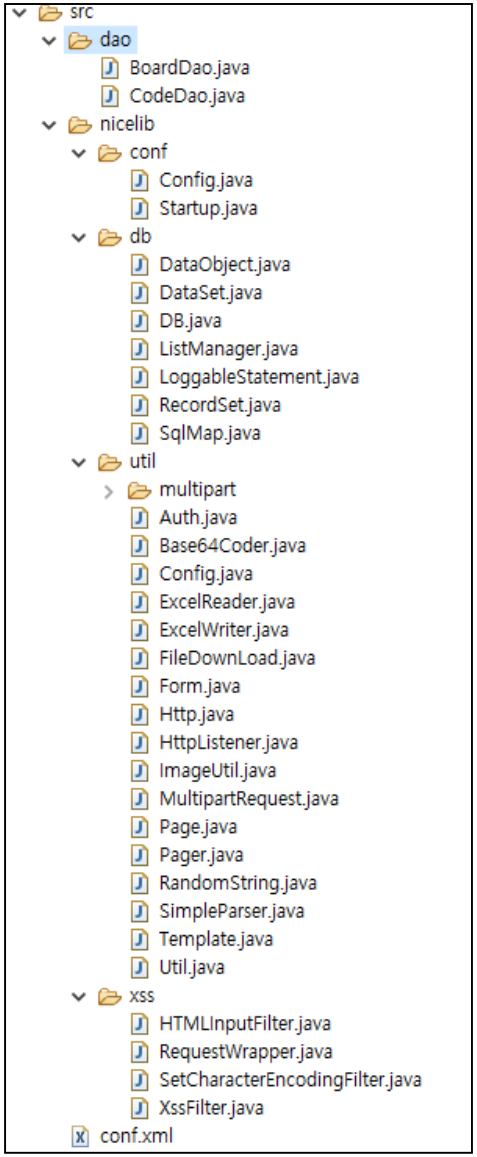
2.Nicelib 구성도

Nicelib의 코어 라이브러리들을 이용 하여 개발을 진행하도록 되어 있습니다.



3.Nicelib Java 패키지 구조

nicelib 패키지 안에 핵심 라이브러리가 위치.
dao 패키지는 각 업무별 공통으로 재사용될 비즈니스 로직이 포함.

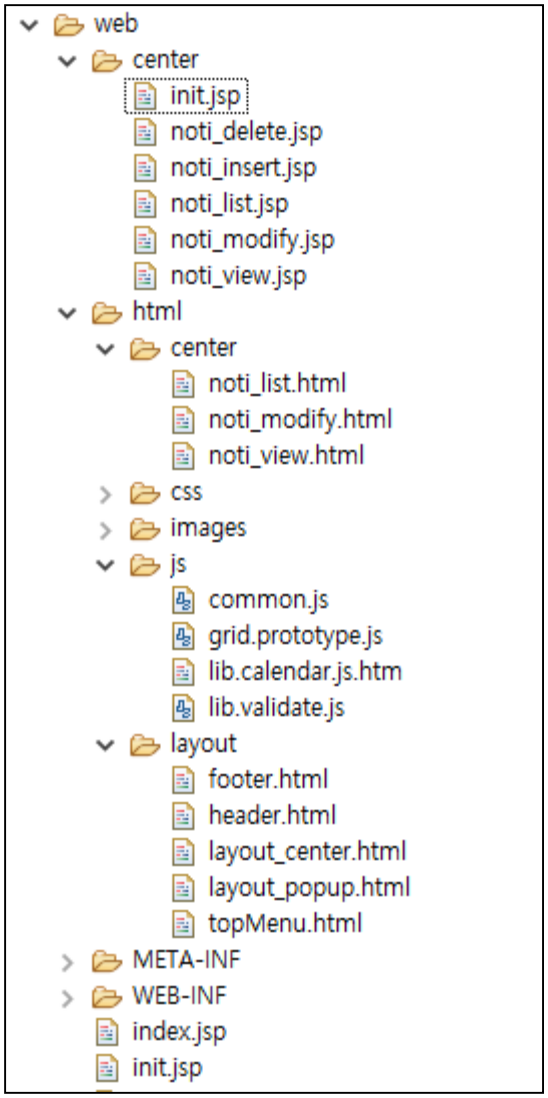


[Java package 구조]

패키지경로		Class명	설명
nicelib	db		DB와 관련되 각종 처리기능을 하는 패키지 이다.
		DB	DB를 액세스 하고 연결하는 Class이다. 모든 DB연결을 관리한다.
		LoggableStatement	PreparedStatement 사용시 query로그에 값을 매핑하고 logging하기 위한 class 이다.
		DataObject	table하나 또는 복수의 Table을 select,insert,update, delete할수 있는 Class이다.
		DataSet	map형태의 Data를 Array형태로 담고 있는 구조의 data 매개변수 역할의 Class 이다.
		RecordSet	DataSet을 상속하여사용 하명 DB에서 가져온 Data의 mata data를 확인하여 DataSet형태로 return한다.
		ListManager	jsp에서 list페이지와 관련괸 각종 처리 함수들이 있는 Class이다.
	util		각종 공통 함수 및 페이지 처리를 위한 모듈이 있는 패키지이다.
		Config.java	모듈과 관련된 각종 환경 변수를 setting하는 class이다.
		Auth	사용자 인증을 위한 쿠키 또는 세션을 컨트롤 하는 class이다.
		Template.java	html과 jsp코딩을 분리 하게 하는 핵심 class로 html을 파싱하는 class이다.
		Page.java	layout html을 사용할수 있게 하는 class이다.
		Pager.java	listManager class와 함계 list페이지를 컨트롤 할때 사용 하는 class이다.
		Form.java	html의 form객체를 컨트롤 할수 있도록 하는 clsass이다.
		Util.java	각종 공통 함수들이 포함된 class이다.
		ExcelReader.java	excel파일을 읽어올수 있도록 하는 class이다.
		ExcelWriter.java	excel파일을 작성할수 있도록 하는 class이다.

[주요 class 소개]

3.Nicelib web 폴더 구조



[web 폴더 구조]

※ 기본 설명
-각 대분류 업무 별로 폴더를 구분한다.
-대분류 업무폴더 안에는 jsp파일이 위치 한다.
- html 폴더 안에는 html, image, js파일 등의 정적인 파일이 위치한다.
  html 폴더 안에는 각 대분류 폴더와 동일 한 구조를 가진 폴더를 생성 하며
  jsp파일에서는 html폴더 안의 파일을 읽어 들여 web page의 화면을 구성 한다.

파일 명명 및 jsp와 html 매칭 규칙

기능	jsp	html	명명규칙
목록	note_list.jsp	noti_list.html	*_list 형식
신규작성	note_insert.jsp	noti_modify.html	*_insert
수정	noti_modify.jsp		*_modify
삭제	noti_delete.jsp	프로세스페이지로 해당없음.	*_delete

※ layout폴더
-html 구성시 매 페이지다 공통으로 표시 되어야 하는 부분을 layout폴더에서 구성한뒤
  각 페이지에서는 layout을 호출 하여 html이 구성되도록 한다.

※ init.jsp
-모든 웹 페이지의 최상위 페이지로 다른 모든 페이지는 init.jsp를 include하고 있다.
- 해당 페이지에서는 자주 사용 되는 Util, Form, Page, 등의 class가 생성/선언 되어 있다.

4.DB핸들링(1/2)

-모든 DB DATA 조작 (select,insert,update,delete)은 DB.java파일을 통해 이루어 진다.
- DataObject.java를 이용 하여 단일 row의 data는 쉽게 핸들링 할 수 있다. DataObject 내부에서 DB.java를 호출 하여 사용 되도록 되어 있다.

• select 예제
DataObject 를 통해 조회된 값은 DataSet 객체로 반환 되며 DataSet 은 Vector Class에 HashTable Class가 순차 적으로 들어 가 있는 구조이다.

```
DataObject boardDao = new DataObject("tcb_board");//조회 대상이 되는 table의 DataObject를 선언 new DataObject("테이블명")

//실행 되는 query : SELECT * FROM tcb_board WHERE 1=1
DataSet board = boardDao.find("1=1");

//비정렬 -> 실행 되는 query : SELECT board_id,title,open_date FROM tcb_board WHERE 1=1
board = boardDao.find("1=1","board_id,title,open_date");

//정렬 -> 실행 되는 query : SELECT board_id,title,open_date FROM tcb_board WHERE 1=1 ORDER BY board_id asc
board = boardDao.find("1=1","board_id,title,open_date","board_id asc");

//query
board = boardDao.query("select * from tcb_board where 1=1");

while(board.next()){
    System.out.println(board.getString("title"));//게시글 제목 출력
    System.out.println(board.getInt("board_id"));//게시글 id 출력
}

//하나의 String 값조회
String boardId = boardDao.getOne(" select board_id from tcb_board where 1=1");
//하나의 int형 값조회
int ibordId = boardDao.getOneInt(" select board_id from tcb_board where 1=1");
```

* insert, update, delete 예제

```
//insert
//실행 되는 query -> INSERT INTO tcb_board (board_id, title) values('1','test입력');
DataObject boardDao = new DataObject("tcb_board");
boardDao.item("board_id", "1");
boardDao.item("title", "test입력");
if(!boardDao.insert()){
    System.out.println("insert 실패");
    return;
}

//update
//실행 되는 query -> UPDATE SET tcb_board SET title = 'test입력- 제목수정' WHERE board_id = '1';
boardDao = new DataObject("tcb_board");
boardDao.item("title", "test입력- 제목수정");
if(!boardDao.update("board_id = '1' ")){
    System.out.println("update 실패");
    return;
}

//delete
//실행 되는 query -> DELETE FROM tcb_board WHERE board_id = '1';
boardDao = new DataObject("tcb_board");
if(!boardDao.delete("board_id = '1' ")){
    System.out.println("delete 실패");
    return;
}
```

4.DB핸들링(2/2)

- DB data 조직시 Transaction 처리를 위해서는 아래 예제를 참고 하여 코딩 할 수 있다.

• Transaction 처리 예제

```
DB db = new DB();
//insert 정보
DataObject boardDao = new DataObject("tcb_board");
boardDao = new DataObject("tcb_board");
boardDao.item("category", "aa");
boardDao.item("board_id", "1");
boardDao.item("title", "test입력");
db.setCommand(boardDao.getInsertQuery(), boardDao.record);

//update정보
boardDao = new DataObject("tcb_board");
boardDao.item("title", "test입력-제목수정");
db.setCommand(boardDao.getUpdateQuery(" board_id ='1' "), boardDao.record);

//delete정보
boardDao = new DataObject("tcb_board");
db.setCommand(boardDao.getDeleteQuery("  board_id ='1' "), boardDao.record);

//DB객체서 각 정보이용 하여 순차적으로 실행
//실행 중 실패 하면 rollback 정상 처리시 commit
if(!db.executeArray()){
    System.out.println("처리 실패");
    return;
}
```

4.Page객체 안내 및 html파싱 안내

- Nicelib framework는 jsp 페이지 호출시 각 비즈니스 로직을 실행 후 html파일을 Page 객체를 통해 파싱하여 사용자의 브라우져의 응답하는 구조임.

•page객체 사용 예시

```
Page p = new Page("/WebRoot경로");
p.setRequest(request);
p.setPageContext(pageContext);

DataObject boardDao = new DataObject("tcb_board");//조회 대상이 되는 table의 DataObject를 선언 new DataObject("테이블명")
DataSet board = boardDao.find("1=1");
if(board.next()){//Vector 객체 포지션 이동
}
p.setBody("center.noti_list");//html파일 경로 /html폴더 아래로 입력된 인지의 순서대로 html파일을 찾아서 읽어 들인다.
//각종 datatype의 변수를 page객체에 넘겨 줄수 있고 넘겨 받은 값은 html 파싱 시 매칭되는 값에 따라 랜더링 된다.
p.setVar("value_string","문자 입력");
p.setVar("value_int",1);
p.setVar("value_boolean", true );
p.setVar("value_dataset", board );// 현제 포지셔닝 된 행의 data만 접근 가능
p.setLoop("value_loop", board ); // loop tag를 통해 전체 dataset의 정보를 표시가능
p.display(out);// 최종적으로 display 함수 실행시 html파싱 및 정보 치환 작업이 일어 난다.
```

-페이지 객체를 통해 html파일을 읽어 사용자에게 응답시 Nicelib의 차제 파서를 통해 html이 파싱되며 이때 Nicelib만의 html Tag가 사용된다.

•NiceLib html tag안내

```
<!-- page 객체에 setVar 된 값의 표시 -->
{{value_string}} => 문자 입력으로 표시
{{value_int}} => 1이 표시
{{value_boolean}} => True 가표시됨.
{{value_dataset.board_id}} => 게시판 글아이디 표시
{{value_dataset.title}} => 게시판 글제목 표시
```

```
<!-- HTML상의 LOOP문 처리 -->
<!-- LOOP START 'value_loop' -->
게시글 아이디 : {{value_loop.board_id}}
    <!-- IF START 'value_loop.title' -->
        게시글 제목 : {{value_loop.title}}
    <!-- IF END 'value_loop.title' -->
    <!-- IFNOT START 'value_loop.title' -->
        게시글 제목이 공백 입니다.
    <!-- IFNOT END 'value_loop.title' -->
<!-- LOOP END 'value_loop' -->
```

```
<!-- HTML상의 if문 처리 -->
//IF 구문과 IFNOT 구문만 존재 하며
//boolean 값 외에도 객체의 존재 여부 또는 값의 존재 여부에 따라 true / false 가 결정 된다.

<!-- IF START 'value_boolean' -->
True 일 경우
<!-- IF END 'value_boolean' -->
<!-- IFNOT START 'value_boolean' -->
false 일 경우
<!-- IFNOT END 'value_boolean' -->

<!-- IF START 'value_string' -->
값이 공백이 아닌경우
<!-- IF END 'value_string' -->
<!-- IFNOT START 'value_string' -->
값이 공백인 경우
<!-- IFNOT END 'value_string' -->

<!-- IF START 'value_dataset.board_id' -->
게시글 아이디가 있는 경우/ 게시글 아이디 : {{value_dataset.board_id}}
<!-- IF END 'value_dataset.board_id' -->
<!-- IFNOT START 'value_dataset.board_id' -->
게시글 아이디가 공백 또는 값이 없습니다.
<!-- IFNOT END 'value_dataset.board_id' -->
```

```
<!-- INCLUDE FILE 'html폴더 이하 html파일경로' -->
해당 파일을 찾아서 함께 렌더링 한다.

<!-- EXECUTE FILE 'WebRoot이하의 jsp파일 경로' -->
실행된 jsp팡리 결과를 함께 렌더링 한다.
<jsp:include>와 동일 하다.
```

5.실제 jsp소스 안내 (목록 Sample)

```jsp
<%@ page contentType="text/html; charset=euc-kr" %><%@ include file="init.jsp" %><%

//form객체를 통해 html상에 <form> 안의 <input>객자를 선언. 초기 값 setting및 server validateion에 사용
f.addElement("s_title",null, null);//검색 조건 값setting

//목록 생성
ListManager list = new ListManager();//pageing처리를 위한 객체 선언
list.setRequest(request);
//list.setDebug(out);
list.setListNum(10);// 한페이지 에 보여질 목록의 수
list.setTable("tcb_board"); // 조회 하고자 하는 table
list.setFields("*"); // 조회 하고자 하는 field
list.addWhere(" category = 'noti' "); // where 조건 추가
list.addWhere("  and open_yn= 'Y' "); // where 조건 추가
list.addSearch("title", f.get("s_title"), "LIKE"); //검색 조건 추가 f.get("s_title")의 값이 있는 경우만 LIKE검색을 한다.
list.setOrderBy(" open_date desc "); // order by 구문 추가

//목록 데이타 수정
DataSet rs = list.getDataSet();// 실재로 query 가 실행 되는 시점

//조회된 data를 formatting 및 변형
while(rs.next()){
    rs.put("open_date",u.getTimeString("yyyy-MM-dd",rs.getString("open_date")));// 날짜 형식 formatting 20170101 = > 2017-01-01
    rs.put("mgr_auth", auth.getString("_MGR_AUTH").equals("10")?true:false); //
}

//html 렌더링을 위해 page객체에 값을 할당
p.setLayout("center");// 함계 렌더링된 layout을 설정
//p.setDebug(out);
p.setBody("center.noti_list");// 메인을 나올 페이지를 설정
p.setVar("nevi","홈 &gt; 고객센터 &gt; 고객센터 &gt; 공지사항");
p.setVar("page_title","공지사항");
p.setLoop("list", rs);
p.setVar("pagerbar", list.getPaging());//listManager를 통해 생성된 pageing처리 html 문자열
p.setVar("query", u.getQueryString()); // request.getQueryString() 과 동일
p.setVar("list_query", u.getQueryString("noti_seq"));// request.getQueryString()중 noti_seq변수를 제거
p.setVar("form_script",f.getScript());// Form객체를 통해 검색값을 유지하거사 validation.js를 통해 validation할 javascript 문자열
p.setVar("member_type",(auth.getString("_MEMBER_TYPE") == null)||auth.getString("_MEMBER_TYPE").equals("02")?false:true);//회원종류
p.display(out);
%>
```

5.실제 jsp소스 안내.(Data입력 페이지 sample)

```jsp
<%@ page contentType="text/html; charset=euc-kr" %><%@ include file="init.jsp" %><%
f.uploadDir=Startup.conf.getString("file.path.noti");//파일이 저장된 경로 지정
f.maxPostSize= 10*1024;//업로드 될 파일 크기 제한

//form객체를 통해 html상에 <form> 안의 <input>객체를 선언. 초기 값 setting및 server validateion에 사용
//Form.java를 통해 optiom값을 확인 하기 바람. lib.validat.js 연동 되어 있음.
f.addElement("title", null, "hname:'제목', required:'Y', maxbyte:'255'");//필수 입력 사항이고 , 255자 이하고 입력 해야 함.
f.addElement("open_date", null, "hname:'공지일자', required:'Y'");
f.addElement("open_yn", null, "hname:'공개여부'");
f.addElement("reg_id", null, "hname:'등록자', required:'Y', maxbyte:'12'");
f.addElement("contents", null, "hname:'공지내용', required:'Y'");

if(u.isPost() && f.validate()){
    // 입력수정
    DB db = new DB();
    DataObject boardDao = new DataObject("tcb_board");
    String id = boardDao.getOne(
        "select nvl(max(board_id),0)+1 board_id "+
        "  from tcb_board where category = 'noti'"
    );
    String file_path = id+"/";

    f.uploadDir= Startup.conf.getString("file.path.noti")+file_path;
    String sContents = f.get("contents");
    boardDao.item("board_id", id);
    boardDao.item("category", "noti");
    boardDao.item("reg_id", f.get("reg_id"));
    boardDao.item("open_date", f.get("open_date").replaceAll("-", ""));
    if(f.get("open_yn").equals("Y"))
        boardDao.item("open_yn", "Y");
    else
        boardDao.item("open_yn", "N");
    boardDao.item("title", f.get("title"));
    boardDao.item("contents", sContents);
    boardDao.item("reg_date", u.getTimeString());
    db.setCommand(boardDao.getInsertQuery(), boardDao.record);

    //파일
    for(int i = 1 ; i <= 3; i ++){
        File file = f.saveFileTime("file_pds_"+i);
        if(file != null){
            String file_name = file.getName();
            DataObject fdao = new DataObject("tcb_board_file");
            fdao.item("board_id", id);
            fdao.item("category", "noti");
            fdao.item("file_seq", i);
            fdao.item("doc_name", f.get("doc_name_"+i));
            fdao.item("file_path", file_path);
            fdao.item("file_name",file_name);
            fdao.item("file_ext", u.getFileExt(file_name));
            fdao.item("file_size", file.length());
            db.setCommand(fdao.getInsertQuery(),fdao.record);
        }
    }

    if(!db.executeArray()){
        u.jsError("처리중 오류가 발생 하였습니다.");//정상적으로 처리 되지 않은 경우 메세지 경고후 history.go(-1)
        return;
    }
    u.jsAlert("정상적으로 저장 되었습니다.");//정상적으로 처리 되지 된경우 메세지
    u.jsReplace("noti_list.jsp");//페리지 이동
    return;
}

p.setLayout("center");
//p.setDebug(out);
p.setBody("center.noti_modify");
p.setVar("nevi","홈 &gt; 고객센터 &gt; 고객센터 &gt; 공지사항");
p.setVar("page_title","공지사항");
p.setVar("modify", false);
p.setVar("form_script", f.getScript());
p.display(out);
%>
```

-get방식으로 요청이 온 경우 페이지 가 조회된다.
- post방식으로 넘어온경우 저장 로직 실행 후 페이지가 이동된다.

-Form.java 및 lib.validate.js 소스를 확인 시 validation시 사용 되는 각종 옵션을 확인 할 수 있다.

5.실제 jsp소스 안내.(프로세스 페이지)

```jsp
<%@ page contentType="text/html; charset=euc-kr" %><%@ include file="init.jsp" %>
<%
String id = u.request("id");// request.getParameter("") 와 동일
if(id.equals("")){
    u.jsError("정상적인 경로로 접근하여 주십시오.");//key값이 없는 경우 경고메세지후 history.go(-1);
    return;
}
DataObject dao = new DataObject("tcb_board");
// 첨부파일 삭제
DataSet ds = dao.find("board_id=" + id + " and category = 'noti'");
if(!ds.next()){
    u.jsError("해당 개시물이 없습니다.");
    return;
}

DB db = new DB();
db.setCommand("delete from tcb_board_file where board_id=" + id + " and category = 'noti' ", null);
db.setCommand("delete from tcb_board where board_id=" + id + " and category = 'noti' ", null);
if(!db.executeArray()){
    u.jsError("처리에 실패 하였습니다.");
    return;
}
u.jsAlertReplace("삭제 되었습니다.", "./noti_list.jsp?"+u.getQueryString("id"));
%>
```