

		개발표준안			
작성자 : 블루베리즈		작성일 : 2020.09.07		문서번호 : 2	
				승인일 :	

1. 개발 환경

1. 개발툴

- Eclipse JAVA EE IDE 2020-3 Version 을 기본 개발툴로 사용한다.
- Oracle Database11gExpress Edition 을 Database 관리툴로 사용한다.
- MariaDB10.5.5 Version 을 Database 관리툴로 사용한다.

2. 버전관리계획

- Git 및 GitHub 를 이용하여 단위 개발 및 Version 관리를 한다.
- 테스트 서버의 관리 및 작업 내용 반영은 최종 개발 책임자가 작업한다.
- 개발 진행중인 작업 내용의 경우 개인 단위의 백업관리가 이루어지며, 개발 완료 작업 내용과 일(daily)단위의 백업은 최종 개발 책임자가 맡는다.

3. 데이터베이스 접근 계획

- 모든 개발자는 하나의 데이터베이스 서버를 공유하여 사용한다.
- 데이터베이스의 접근은 WAS 의 자체 Database Pooling 모듈을 사용하며, DB 리소스는 독립적으로 획득할 수 없다.

2. 명명 규칙

1. 디렉토리 명명 규칙

- 디렉토리 명칭은 영문으로 사용한다.
- <context root>하위에 메뉴의 깊이를 기준으로 디렉토리를 생성하며, 약어를 사용하지 않는다.

2. JSP 파일 명명 규칙

- 해당 디렉토리 명칭을 말머리로 갖고 파일의 주요 기능을 표기한다

		개발표준안			
작성자 : 블루베리즈	작성일 : 2020.09.07	문서번호 : 2	승인일 :		

XXXXX_YYYYY.jsp

a

b

c

a	디렉토리 명칭	메뉴명에 대한 약어
b	구분자	' '
c	주요 기능	1)게시판 : insert(글작성), list(목록), answer(댓글작성), update(수정), delete(삭제) search(검색), paging(페이징) 2)채팅 : openchat(오픈채팅) 3)멤버관리 : signup(회원가입), login(로그인), logout(로그아웃), signout(탈퇴) 4)캘린더 : calendar(일정관리) 5)지도 : map(지도)

구분자를 제외한 모든 섹션은 영문자로 시작하며, 대문자를 사용하지 않으며
가급적 의미를 알 수 없는 약어를 사용하지 않는다.

사용되는 확장자는 아래와 같다.

파일 형식	파일 확장자
Java Servlet Page(JSP)	.jsp
Cascading Style Sheet(CSS)	.css
JavaScript	.js
Hyper Text Markup Language	.html
Web Resource	.gif, .jpg
Flash	.swf
Extensible markup language	.xml
JAVA	.java

		개발표준안			
작성자 : 블루베리즈	작성일 : 2020.09.07	문서번호 : 2		승인일 :	

3. 패키지 명명 규칙 및 자바 파일 명명 규칙

- 패키지명은 3 자리 이상으로 정의한다.
 - ❖ com.bb.controller : Controller 파일 정의
 - ❖ com.bb.model : Service 및 DAO 파일 정의
 - ❖ com.bb.dto : dto 파일 정의
 - ❖ com.bb.db : DB 파일 정의
- 자바 파일은 3 자리 이상의 package 명 하위 단에 작성한다.

4. 자바스크립트 파일 명명 규칙

- 파일이 사용되는 기능명으로 명명한다
 - ❖ board : 게시판
 - ❖ calendar : 달력
 - ❖ login : 로그인
 - ❖ chat : 채팅
 - ❖ map : 지도
- 예) board.js : 게시판 기능에 사용되는 자바스크립트 파일

5. 클래스 명명 규칙

- DAO, DTO 의 경우 해당 데이터베이스 테이블명칭을 그대로 갖는 클래스 이름을 사용하며, com.bb.dto, com.bb.model.dao 을 상위 패키지로 갖는다.
 - 예 1) com.bb.dto.MemberDto.class
 - 예 2) com.bb.model.dao.MemberDao.class
- Jar 파일의 경우 WEB-INF/lib 폴더에 위치 시킨다.

6. 객체 및 변수 명명 규칙

		개발표준안			
작성자 : 블루베리즈	작성일 : 2020.09.07	문서번호 : 2		승인일 :	

- 객체의 경우 해당 객체가 페이지 내에서 유일할 경우 첫글자를 소문자로 바꾼 클래스명을 그대로 사용한다. 만약 해당 객체가 페이지 내에서 유일하지 않은 경우에는 해당 객체의 역할을 상세히 나타낼 수 있는 명칭을 말머리로 하여 선언한다.
 - 예 1) 클래스 MemeberDao 의 객체 *memberDao*
 - 예 2) 클래스 MemberDto 의 객체 *firstMemberDto, lastMemberDto*
- 변수의 경우 단순 증감이며 for 문 등의 제어문 내에서 선언되는 경우 <i,j,k> 등의 간단한 소문자를 사용한다.('p'는 예약된 변수로써, PreparedStatement 의 index 에서 사용한다.)
 - 예) *for(int i=0;i<10;i++)*
- 특정 정보를 담고 있는 변수인 경우, 해당 정보의 성격을 나타낼 수 있는 명칭을 사용한다
 - 예) String *cal_title* = "일정제목";
 - HTML 내의 <input> 태그로 사용되는 필드 값을 전달받을 경우 동일한 이름의 변수에 할당한다.

7. 데이터베이스 명명 규칙

- 테이블 명칭 : 테이블의 용도를 알 수 있도록 명명한다.
 - 예) 캘린더테이블 : CALENDAR
- 필드 명칭 : 외부참조키가 아닌 경우 <TABLE NAME>+'_'를 붙이는 형태로 사용하며, 외부참조키(FK)인 경우는 해당 FK 의 원본 필드 명칭을 그대로 사용한다.
 - 예 1) MEMBER_NO
- 시퀀스 명칭 : 말머리로 <필드명>+NO 를 붙이는 형태로 사용한다
 - 예 1) MEMBER_NO

		개발표준안			
작성자 : 블루베리즈		작성일 : 2020.09.07		문서번호 : 2	
				승인일 :	

3. 개발 구조

1. 디렉토리 구조

- 경로는 클래스 및 인터페이스 파일의 경우 java resource 폴더를, 그 외 파일들은 <context root>를 상위로 갖는다.
- Directory 이름은 의미있는 하나의 영문 단어로 구성하고 전부 소문자로 표기하도록 함. 또한 혼동을 피하기 위해 가급적 약어를 사용하지 않도록 함.
- 다음은 디렉토리 구성 규칙의 예임.
 - ❖ JSP 파일 : <context root> + /*.jsp'
 - ❖ JAVA Class 파일 : <java resource> + <package>
 - ❖ JAVA Source 파일 : <java resource> + <package>
 - ❖ XML 파일 : <context root>'/WEB-INF/xml'
 - <context root> + '/WEB-INF/views/'
 - ❖ 이미지 파일 디렉토리 : <context root> + '/images'
 - ❖ JavaScript 디렉토리 : <context root> + '/js'
 - ❖ CSS 디렉터리 : <context root> + '/css'

		개발표준안	
작성자 : 블루베리즈	작성일 : 2020.09.07	문서번호 : 2	승인일 :

4. 코드 구조

1. JAVA 코드 구조

- DAO 구조 예제

```
package com.kitri.isl.model;

import java.util.Date;

@Repository
public class ABookDao implements IBookDao {

    private final String NAME="abook.";
    private static final Logger logger = LoggerFactory.getLogger(ABookDao.class);
    @Autowired
    private SqlMapClientTemplate manager;

    @Override
    public int getTotalBook() {
        // TODO Auto-generated method stub
        logger.info("getTotalBook {}", new Date());
        return (int) manager.queryForObject(NAME+"getTotalBook");
    }

    @SuppressWarnings("unchecked")
    @Override
    public List<ABookDto> getAdminBookList(int start, int end) {
        // TODO Auto-generated method stub
        logger.info("getAdminBookList {}", new Date());
        Map<String, Integer> map = new HashMap<String, Integer>();
        map.put("start", start);
        map.put("end", end);
        return manager.queryForList(NAME + "getBookList", map);
    }

    @Override
    public boolean insertBook(ABookDto adto) {
        // TODO Auto-generated method stub
        int returnVal = (Integer)manager.insert(NAME + "insertBook", adto);
        return returnVal>0?true:false;
    }
}
```

- DTO 구조는 일반적인 BEAN 의 형태를 갖는다

		개발표준안	
작성자 : 블루베리즈	작성일 : 2020.09.07	문서번호 : 2	승인일 :

2. JSP 및 HTML 코드 구조

- 예제

```
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<fmt:requestEncoding value="UTF-8" />
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" href="./css/content_booklist.css">
</head>
<script type="text/javascript">
    function opt(){
        var getValue = new Function("name", "return document.getElementsByName(name).item(0).value");
        var op = getValue("optType");
        if(getValue("srctext") == ""){
            alert('검색어를 입력하세요');
        }
        if(op == "title"){
            location.href="./booktitlesearchlist.do?titlesrc="+getValue("srctext");
        }else if(op == "writer"){
            location.href="./bookwritersearchlist.do?writersrc="+getValue("srctext");
        }
    }

    function com(){
        if(event.keyCode == 13){
            opt();
        }
    }
</script>
<body>
    <!-- 로그인 여부 판단 -->
    <c:if test="${sessionScope.userInfoSession == null}">
        <script>
            location.href="./index.do";
        </script>
    </c:if>
</body>
</html>
```