Html

명명 규칙

○ html이 위치하는 논리적 위치는 [web-root]/html/novframework/[system]/[sub-system]을 기본으로 한다.

○ html 파일의 명명규칙은 Nov [용어사전].html로 한다

○ 용어사전의 단어를 사용하는 경우 첫 글자만 대문자로 한다.

○ iFrame 사용 시 window if\_ 사용 시 명은 [윈도우위치]를 사용한다 윈도우 위치는 top, bottom,

left, right, main, hidden을 사용하며 여러 개가 필요한 경우 if\_[윈도우위치]\_#를 사용한다. (ex. if\_top\_1, if\_bottom\_9)

○ pop-up 사용시 window명은 p\_[pop-up명]을 사용한다.

○ pop-up은 스크롤이 없이 주소가 나타나지 않도록 하는 것이 원칙이나 보안 설정에

의해서 주소가 나타나더라도 상하 scroll이 나타나지 않도록 한다.

Coding Style

○ 스타일은 css를 사용한다.

○ css이 위치하는 논리적 위치는 [web-root]/css/novframework을 기본으로 한다

○ css 파일의 명명규칙은 Nov[용어사전].css로 한다

○ size는 상대값(%)을 사용하지 않고, 절대값을 사용한다.

Java script

명명규칙

○ js가 위치하는 논리적 위치는 [web-root]/js/novframework/[system]/[sub-system]을 기본으로 한다.

○ js 파일의 명명규칙은 [용어사전].js로 한다

○ 용어사전의 단어를 사용하는 경우 첫 글자만 대문자로 한다. (ex. NovCheckValid.js)

○ function 명은 fb\_nov[동사]\_[명사]를 사용한다. (ex. fn\_nov\_check\_valid(), 동사 및 명사는 용어사전을 기준으로 한다.)

○ 변수명은 소문자로 시작하고 새로운 단어의 시작은 대문자를 사용한다. (ex. reqMsg)Coding Style

○ js 내에서 다른 include는 원칙적으로 사용하지 않다.

○ 다른 js에서 공통으로 include하는 js는 [web-root]/js/novframework/cmm에 위치한다.

○ 여러 sub-system에서 include 하는 js는 [web-root]/js/novframework/[system]/cmm

에 위치한다.

자바

공통

java class file 은 필요 시 사업을 위한 wrapper를 생성하여 사용함으로써 객체에n대한 추상화와 구현 시의 유연성을 확보한다. Java class 작성 시에 해당 package, file 및nMethod의 명명 규칙과 coding style을 준수하여 일관성 있는 개발이 되도록 한다.

명명 규칙

Java package

○기본 패키지 명명규칙은 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

java class

○ java file의 명명규칙은 Nov[용어사전][postfix].java로 한다.

○용어사전의 단어를 사용하는 경우 첫 글자만 대문자로 한다.

○postfix는 framework에서 정의하는 class요소에 대한 구분으로 해당 java class의 성격을 나타낸다. (posfix: service, servicelmpl 등)

Interface class

○ Interface는 java 클래스명과 동일한 방법으로 부여한다.

Java method

○ java method의 명명규칙은 [동사][용어사전]로 한다.

※ 클래스 명명은 필수로 하되 메소드 명명규칙은 용어사전을 최대한 준용하도록 한다.

○ 첫 글자는 소문자를 사용하며 이후 용어의 첫 글자만 대문자를 사용하며 숫자 및 특수문자는 사용하지 않는다.

○ method에서 사용하는 동사는 다음 표와 같이 명명한다.

테이블이(가) 표시된 사진

자동 생성된 설명

Database

공통

SQL은 개발자간 기술의 차이가 크고 Framework을 사용하여 개발하는 application에 비해서 표준화가 어려우므로 Coding 표준으로 제정된 내용을 준수하여 품질의 일관성을 확보한다.

명명규칙

○데이터베이스의 개체들은 각각의 역할과 의미를 잘 반영하는 이름을 사용한다.

예를 들어, 테이블 이름은 복수형 명사를, 뷰 이름은 동사를, 컬럼 이름은 단수형 명사를 사용.

○데이터베이스의 개체들은 약어나 축약어를 사용하지 않고, 영문자와 숫자만을 사용한다.

○데이터베이스의 개체들은 소문자로 작성하고, 데이터 타입을 이름으로 작성하지 않는다

○단어 사이는 underscores(\_)로 구분하고 전체 단어보다는 통용되는 약어를 사용한다.

○데이터베이스의 개체들은 표준화된 접두사나 접미사를 사용해야 합니다.

예시) ○뷰- 맨 마지막에 \_v를 추가

○트리거 - 맨 앞에 tg\_를 추가

○프로시저 - 맨 앞에 sp\_를 추가

○패키지 - 맨 앞에 pkg\_를 추가

○제약조건 - 맨 앞에 ct\_ 를 추가

○외래키명 - 맨 앞에 fk\_를 추가

○명명규칙은 다음과 같다.

○Table



○cmm\_user

테이블이(가) 표시된 사진

자동 생성된 설명

○cmm\_sys\_cd\_grp테이블이(가) 표시된 사진

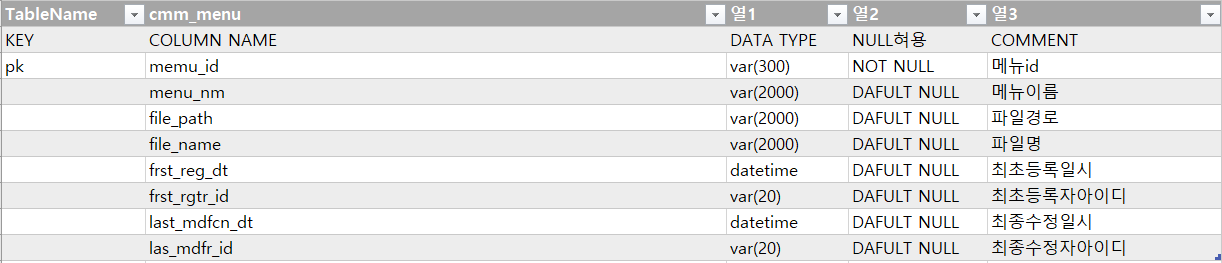
자동 생성된 설명

○cmm\_sys\_cd

테이블이(가) 표시된 사진

자동 생성된 설명

○cmm\_menu



○wrt\_novel

테이블이(가) 표시된 사진

자동 생성된 설명

Coding Style

○ SQL 예약어는 소문자를 사용한다

○ SELECT, UPDATE시 컬럼 나열은 1줄당 1개씩 소문자로 작성을 기본으로 하며 컬럼이 많은 경우에 한하여서 1줄에 3개 컬럼까지 쓰는 것을 예외로 허용한다.

○ SELECT의 들여쓰기는 SELECT를 기준으로 FROM, WHERE, INTO 등의 시작점을 맞춥니다 단 GROUP BY나 ORDER BY는 SELECT(6) 자 보다 더 긴 관계로 BY 이후 스페이스 1칸을 띄우고 관련내용을 기술한다.

○ FROM 절의 테이블명은 1줄당 1개씩 소문자로 작성하는 것을 기본으로 하며 테이블이 많은 경우에 한하여서 1줄당 2개의 테이블까지 적는 것을 예외로 허용한다.

○ WHERE절의 조건은 1줄당 1개의 조건씩 작성한다

○ INSERT문장의 경우 전체 컬럼에 대해서 insert를 하더라도 컬럼명을 반드시 적어준다.

○ DML의 들여쓰기는 UPDATE, DELETE, INSERT를 기준으로 SET, FROM, WHERE 등의 시작점을 맞추고, 컬럼 구분은 가장 길이가 긴 컬럼의 쉼표 나 등호 등을 기준으로 다른 컬럼을 맞춘다.

○ 괄호를 사용할 경우 즉 인라인 뷰 스칼라 서브쿼리 등을 사용할 경우 괄호가 존재하는, 라인은 라인을 변경하지 않는다.