|  |
| --- |
| **『 산출물 관리 시스템 』** |
| **운영자매뉴얼** |
| **문서번호**  **Version 1.0**  **2024.04.11** |

****

제/개정 이력

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | 버전 | 변경일 | 구분 | 변경 내용 | 작성자 | 소속 | 직위 |
| 1 | V1.0 | 2024.04.11 | 작성 | 초기 작성 | 황유진 | 인젠트 | 매니저 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **※ 구분 : 작성/검토/승인 → 추가/수정/삭제 중 선택 기입**  **※ 변경내용 : 변경된 위치와 변경내용을 기록** | | | | | | | |

목 차

[1. 개요 1](#_Toc166054870)

[1.1. 목적 1](#_Toc166054871)

[1.2. 대상 1](#_Toc166054872)

[2. 시스템 개요 2](#_Toc166054873)

[2.1. 시스템 구성 2](#_Toc166054874)

[2.1.1. 시스템 디렉토리 구성 3](#_Toc166054875)

[2.1.2. 개발 서버 시스템 정보 3](#_Toc166054876)

[2.1.3. DB 정보 5](#_Toc166054877)

[2.1.4. GIT 정보 5](#_Toc166054878)

[2.2. 구동 및 종료 방법 6](#_Toc166054879)

[2.2.3. 구동 6](#_Toc166054880)

[2.2.4. 중지 6](#_Toc166054881)

[3. 시스템 운영 7](#_Toc166054882)

[3.1. CommonMethod 7](#_Toc166054883)

[3.2. Config 7](#_Toc166054884)

[3.1.1. application.properties 7](#_Toc166054885)

[3.1.2. MyBatis 8](#_Toc166054886)

[3.1.3. LOG 8](#_Toc166054887)

[3.2. 로그 9](#_Toc166054888)

[3.3. Interceptor 10](#_Toc166054889)

[3.4. Tiles 11](#_Toc166054890)

[3.5. 배포 12](#_Toc166054891)

[3.5.1 . 인텔리제이 환경 배포 방법 12](#_Toc166054892)

[3.5.2 . 이클립스 환경 배포 방법 12](#_Toc166054893)

[4. COMMON 13](#_Toc166054894)

[4.1. API\_INFO 13](#_Toc166054895)

[4.2. USER\_AUTHORITY 13](#_Toc166054896)

[4.3. MENU 14](#_Toc166054897)

[4.3.1 . MENU 14](#_Toc166054898)

[4.3.2 . MENU\_AUTHORITY 14](#_Toc166054899)

[4.4. COMMON\_CODE 15](#_Toc166054900)

[4.5. MAIL 16](#_Toc166054901)

[4.5.1. MAIL\_INFO 16](#_Toc166054902)

[4.5.2 . MAIL\_HISTORY 17](#_Toc166054903)

[4.6. 산출물 파일 다운로드 17](#_Toc166054904)

[4.6.1. Zip 파일 생성 17](#_Toc166054905)

[4.6.2. 산출물 파일 다운로드 17](#_Toc166054906)

그림 목차

[[그림 2‑1] 시스템 구성도 2](#_Toc114241350)

[[그림 3‑1] 인텔리제이 배포 방법 12](#_Toc114241350)

[[그림 3‑2] 이클립스 환경 배포 방법 12](#_Toc114241350)

표 목차

[[표 2‑1] 시스템 디렉토리 구성 3](#_Toc114241397)

[[표 2 2] 개발 서버 deliverablesUI 시스템 정보 3](#_Toc114241398)

[[표 2‑3] 개발 서버 commonAPI시스템 정보 4](#_Toc114241398)

[[표 2‑4] 개발 서버 projectAPI 시스템 정보 4](#_Toc114241398)

[[표 2‑5] 개발 서버 deliverablesAPI 시스템 정보 4](#_Toc114241398)

[[표 2-6] DB접속 정보 5](#_Toc114241398)

[[표 2-7] GIT Branch정보 5](#_Toc114241398)

[[표 2-8] 실행 파일 위치 6](#_Toc114241398)

[[표 2-9] 종료 파일 위치 6](#_Toc114241398)

[[표 3‑1] commonMethod 구조 7](#_Toc114241398)

[[표 3‑2] application.properties 구조 7](#_Toc114241398)

[[표 3-3] mybatis-config 구조 8](#_Toc114241398)

[[표 3-4] logback-spring 구조 5](#_Toc114241398)

[[표 3-5] interceptor 종류 10](#_Toc114241398)

[[표 3-6] Tiles 구조 11](#_Toc114241398)

[[표 4-1] API\_INFO 테이블 13](#_Toc114241398)

[[표 4-2] USER\_AUTHORITY 테이블 13](#_Toc114241398)

[[표 4-3] MENU테이블 14](file:///C:\Users\USER\Desktop\산출물관리시스템\최종문서\산출물관리시스템_운영자매뉴얼.docx#_Toc114241398)

[[표 4-4] MENU\_AUTHORITY 테이블 14](#_Toc114241398)

[[표 4-5] COMMON\_CODE 테이블 15](#_Toc114241398)

[[표 4-5] COMMON\_CODE 테이블 15](#_Toc114241398)

[[표 4-6] COMMON\_CODE\_GROUP 테이블 15](#_Toc114241398)

[[표 4-7] COMMON\_CODE\_GROUP ID 명명규칙 15](#_Toc114241398)

[[표 4-8] MAIL\_INFO 테이블 16](#_Toc114241398)

[[표 4-9] MAIL\_INFO 최신데이터 16](#_Toc114241398)

[[표 4-10] MAIL\_HISTORY 테이블 17](#_Toc114241398)

# 개요

## 목적

본 문서는 산출물관리 시스템의 환경 구성과 주요 기능들을 중점적으로 기술함으로써, 산출물관리 시스템의 운영자의 작업 효율성을 높이고 서비스의 품질을 제고하기 위함을 목적으로 한다.

## 대상

본 문서는 다음의 관련자 또는 관련팀을 대상으로 한다.

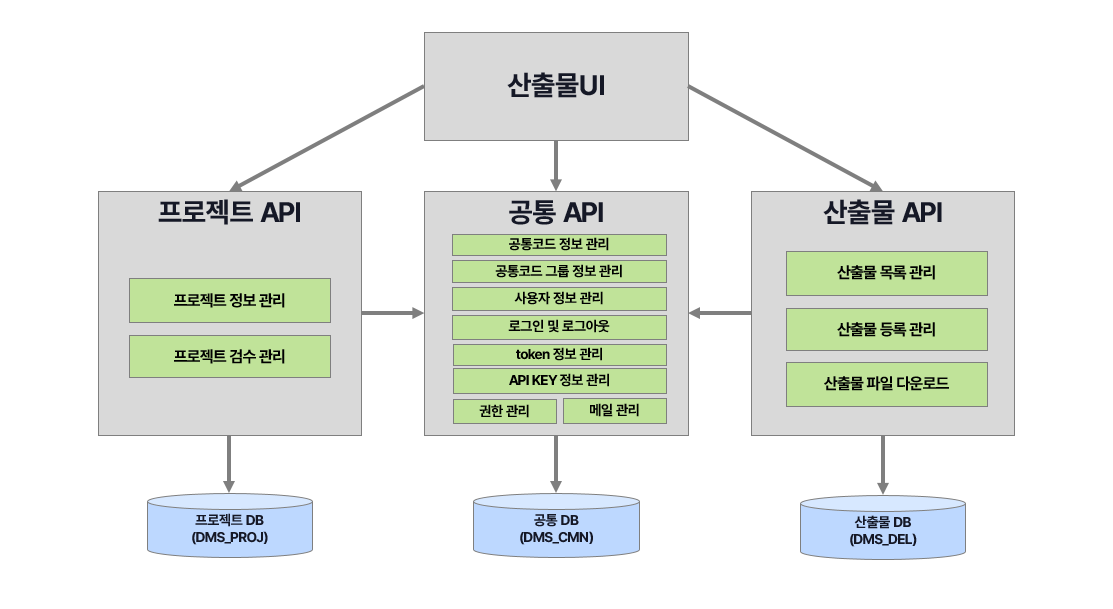
* 산출물관리 시스템 운영자
* 산출물관리 시스템 및 운영팀

# 시스템 개요

## 시스템 구성

산출물 관리 시스템은 MSA 형태로 DeliverablesUI, CommonAPI, ProjectAPI, DeliverablesAPI

UI 시스템 1개, API 시스템 1개로 구성되어있다.



[그림 2‑1] 시스템 구성도

### 시스템 디렉토리 구성

산출물 관리 시스템의 디렉토리 구성은 다음과 같다.

| 구 분 | 기본경로 | 파일경로 | 주요파일명 | 설 명 |
| --- | --- | --- | --- | --- |
| DeliverablesUI | /home/dms/dms\_dev/deliverablesUI | / | deliverablesUI-1.0.0.jar | 실행 파일 |
| /log | deliverablesUI.log | 로그 파일 |
| /ssl | Keystore.p12 | 인증서파일 |
| CommonAPI | /home/dms/dms\_dev/commonAPI | / | commonAPI-1.0.0.jar | 실행 파일 |
| /log | commonAPI.log | 로그 파일 |
| /ssl | Keystore.p12 | 인증서파일 |
| ProjectAPI | /home/dms/dms\_dev/projectAPI | / | projectAPI -1.0.0.jar | 실행 파일 |
| /log | projectAPI.log | 로그 파일 |
| /ssl | Keystore.p12 | 인증서파일 |
| DeliverablesAPI | /home/dms/dms\_dev/deliverablesAPI | / | deliverablesAPI-1.0.0.jar | 실행 파일 |
| /log | deliverablesAPI.log | 로그 파일 |
| /ssl | Keystore.p12 | 인증서파일 |

[표 2‑1] 시스템 디렉토리 구성

### 개발 서버 시스템 정보

|  |  |  |
| --- | --- | --- |
| 구 분 | 항목 | 설정 |
| deliverablesUI 서비스 정보 | IP | 183.111.104.104 |
| HTTP Service port | 6443 |
| deliverablesUI  설치 정보 | 설치 경로 | /dms\_dev/deliverablesUI/ |
| 로그 경로 | /dms\_dev/deliverablesUI/log |
| 인증서 경로 | /dms\_dev/ deliverablesUI /ssl |
| WAS | SpringBoot 내장 Tomcat |

[표 2‑2] 개발 서버 deliverablesUI 시스템 정보

|  |  |  |
| --- | --- | --- |
| 구 분 | 항목 | 설정 |
| commonAPI  서비스 정보 | IP | 183.111.104.104 |
| HTTP Service port | 1443 |
| commonAPI  설치 정보 | 설치 경로 | /dms\_dev/commonAPI/ |
| 로그 경로 | /dms\_dev/commonAPI/log |
| 인증서 경로 | /dms\_dev/commonAPI/ssl |
| WAS | SpringBoot 내장 Tomcat |

[표 2‑3] 개발 서버 commonAPI시스템 정보

|  |  |  |
| --- | --- | --- |
| 구 분 | 항목 | 설정 |
| projectAPI  서비스 정보 | IP | 183.111.104.104 |
| HTTP Service port | 2443 |
| projectAPI  설치 정보 | 설치 경로 | /dms\_dev/projectAPI/ |
| 로그 경로 | /dms\_dev/projectAPI/log |
| 인증서 경로 | /dms\_dev/ projectAPI /ssl |
| WAS | SpringBoot 내장 Tomcat |

[표 2‑4] 개발 서버 projectAPI 시스템 정보

|  |  |  |
| --- | --- | --- |
| 구 분 | 항목 | 설정 |
| deliverablesAPI 서비스 정보 | IP | 183.111.104.104 |
| HTTP Service port | 3443 |
| deliverablesAPI 설치 정보 | 설치 경로 | /dms\_dev/deliverablesAPI/ |
| 로그 경로 | /dms\_dev/deliverablesAPI/log |
| 인증서 경로 | /dms\_dev/ deliverablesAPI /ssl |
| WAS | SpringBoot 내장 Tomcat |

[표 2‑5] 개발 서버 deliverablesAPI 시스템 정보

### DB 정보

DB 접속정보 는 아래 표와 같다.

* 서버 - 183.111.104.100
* Port - 51521
* DataBase – xe

|  |  |  |  |
| --- | --- | --- | --- |
| 서버 | URL | 아이디 | 패스워드 |
| CommonAPI | 183.111.104.100: 51521 | dms\_cmn | dms\_cmn |
| ProjectAPI | 183.111.104.100: 51521 | dms\_proj | dms\_proj |
| DeliverablesAPI | 183.111.104.100: 51521 | dms\_del | dms\_del |

[표 2‑6] DB접속 정보

### GIT 정보

GIT 접속 정보 는 아래와 같다.

* GIT URL - http://183.111.104.117:8081/ecross-team/dms\_dev.git
* Branch 정보

|  |  |
| --- | --- |
| Branch | 설명 |
| Master | 모든 시스템이 통합된 Branch |
| Dev | 개발 단계에서 모든 시스템이 통합된 Branch |
| DeliverablesUI | DeliverablesUI 개발용 Branch |
| CommonAPI | CommonAPI 개발용 Branch |
| ProjectAPI | ProjectAPI 개발용 Branch |
| DeliverablesAPI | DeliverablesAPI 개발용 Branch |

[표 2‑7] GIT Branch정보

## 구동 및 종료 방법

* 구동 스크립트와, 종료 스크립트 실행을 통해 구동/종료 할 수 있다.
* ‘ ./[파일 이름].sh ‘로 실행

### 구동

* 실행 예시 - ./allstartdms.sh

|  |  |
| --- | --- |
| **SYSTEM** | **실행파일 위치** |
| DeliverablesUI | [설치폴더 경로] /dms\_dev/deliverablesUI/startdeliverablesUI.sh |
| CommonAPI | [설치폴더 경로] /dms\_dev/commonAPI/startcommonAPI.sh |
| ProjectAPI | [설치폴더 경로] /dms\_dev/projectAPI/startprojectAPI.sh |
| DeliverablesAPI | [설치폴더경로]/dms\_dev/deliverablesAPI/startdeliverablesAPI |
| 전체 | [설치폴더 경로]/dms\_dev/allstartdms.sh |

[표 2‑8] 실행 파일 위치

### 중지

* 중지 예시 - ./allstopdms.sh

|  |  |
| --- | --- |
| **SYSTEM** | **종료파일 위치** |
| DeliverablesUI | [설치폴더 경로]/dms\_dev/deliverablesUI/stopdeliverablesUI.sh |
| CommonAPI | [설치폴더 경로]/dms\_dev/commonAPI/stopcommonAPI.sh |
| ProjectAPI | [설치폴더 경로]/dms\_dev/projectAPI/stopprojectAPI.sh |
| DeliverablesAPI | [설치폴더 경로]/dms\_dev/deliverablesAPI/stopdeliverablesAPI |
| 전체 | [설치폴더 경로]/dms\_dev/allstopdms.sh |

[표 2‑9] 종료 파일 위치

# 시스템 운영

## CommonMethod

모든 시스템에서 공통적으로 사용하는 코드들을 모아놓은 라이브러리로 구조는 아래 표와 같다

* commonMethod 구조

|  |  |  |
| --- | --- | --- |
| **폴더** | **파일명** | **설명** |
| common | Consts | 공통 변수들을 정의해 놓은 소스 |
| HttpSender | http 통신을 하기 위한 소스 |
| Json | Object 와 Json 간의 변환을 위한 소스 |
| Pagination | 페이징 처리를 위한 소스 |
| RequestMessage | http 통신시 요청 메세지 규격화를 위한 소스 |
| ResponseMessage | http 통신시 응답 메시지 규격화를 위한 소스 |
| Interceptor | CommonApiInterceptor | Header 정보에 있는 API\_KEY 가 일치하는지 확인하는 소스 |
| ConfirmInterceptor | 로그인한 사용자의 권한 및 토큰을 확인하는 소스 |

[표 3‑1] commonMethod 구조

## Config

DMS 시스템을 만들기 위해 설정한 여러 설정 파일들은 아래와 같다

### application.properties

Spring Boot어플리케이션을 구성하는 데 사용하는 공통 어플리케이션 속성을 정의한 파일이다.

* Application.properties 정의 내용

|  |  |  |  |
| --- | --- | --- | --- |
| 속성 | | **값** | **설명** |
| Mybatis. | mapper-locations | classpath:/mapper/\*.xml | myBatis와 관련된 파일의 위치를 선언 |
| config-location | classpath:/mybatis-config.xml |
| Server.port | | [ 각 서비스 포트 ] | Spring application 설정 포트 |
| spring.datasource. | driver-class-name | oracle.jdbc.OracleDriver | Spring application과 DB를 연결하기 위한 선언 |
| url | jdbc:oracle:thin:@183.111.104.100:51521:xe |
| username | dms\_del |
| password | dms\_del |
| spring.mvc.view | prefix | /WEB-INF/views | Model And View 파일 위치, 확장자를 선언 |
| sufix | .jsp |
| API\_KEY | | [ 각 서비스 API KEY 값 ] | 각 API에 해당하는 KEY 값 선언 |
| API\_URL | | https://cmn.inzent.kr:1443 | API\_KEY 확인을 위한 호출 URl 선언 |
| FILEPATH | | /home/dms/dms\_dev/dmsFile/ | 파일을 저장하기 위한 위치 선언 |
| logging.level.root | | Info | Log 레벨설정 |
| server.ssl | Enabled | True | SSL 설정을 위한 값 선언 |
| key-store | ssl /keystore.p12 ( 인증키 파일명 ) |
| key-store-type | PKCS12 |
| key-store-password | dms2024 |
| spring.servlet.multipart | maxFileSize | 100MB | 파일 전송 최대 사이즈 선언 |
| maxRequestSize | 100MB | 파일 전송 사이즈 선언 |

[표 3‑2] application.properties 구조

### MyBatis

* mybatis-config.xml

mybatis에서 사용될 DB를 연동하기 위한 설정값들과 mapper.xml을 등록하기 위한 xml 이다.

* mybatis-config 정의 내용

|  |  |  |
| --- | --- | --- |
| 속성 | **값** | **설명** |
| callSettersOnNulls | true | 가져온 값이 null일 때 메소드를 호출할지 여부 정의 |
| returnInstanceForEmptyRow | true | Select 결과 null이 존재 할 때, null 값 허용 |
| mapUnderscoreToCamelCase | true | 변수의 카멜케이스 적용 |

[표 3‑3] mybatis-config 구조

### LOG

* logback-spring.xml

로그가 저장될 위치 및 파일 이름, 백업 형식 등을 지정하기 위한 xml 이다.

* logback-spring 정의 내용

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 속성 | | | **값** | **설명** |
| LOGS\_PATH | | | ./log/ | 로그가 저장될 위치 지전 |
| FILE | Encoder.Pattern | | %d{HH:mm} %-5level %logger{36} - %msg%n | 인코딩 패턴 설정 |
| file | | ${LOGS\_PATH}/deliverablesAPI.log | 로그 파일 위치 및 이름 지정 |
| rollingPolicy | fileNamePattern | /log/application.log.%d{yyyy-MM-dd}.gz | 백업 파일 위치 및 이름 지정 |
| maxHistory | 30 | 해당 일자가 지나면 오래된순으로 로그를 삭제 |
| totalSizeCap | 5GB | 파일 분할 용량 |
| CONSOLE | Encoder.Pattern | | %d{HH:mm} %-5level %logger{36} - %msg%n | 인코딩 패턴 설정 |
| ROOT | INFO | appender-ref | FILE | 로그를 파일에 출력함 |
| CONSOLE | 로그를 콘솔에 출력함 |
| DEBUG | appender-ref | FILE | 로그를 파일에 출력함 |
| CONSOLE | 로그를 콘솔에 출력함 |

[표 3‑4] logback-spring 구조

## 로그

Spring AOP (Aspect Oriented Programming) 을 사용하여 Controller 호출 시 마다 Debug 레벨의 로그를 호출하게 하였다.

* Maven dependency

|  |
| --- |
| <!-- Logging -->  <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-slf4j-impl</artifactId>  <version>2.10.0</version>  </dependency>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  <version>1.18.30</version>  </dependency> |

* Log.java

|  |
| --- |
| @Around -> 적용될 위치 선언@Component  @Aspect  public class Log {  private Logger logger = LoggerFactory.getLogger(this.getClass());  @Around("within(com.inzent.[ 각 서비스 API 명 ].controller.\*)")  public Object saveLogging(ProceedingJoinPoint pjp) throws Throwable {  Object result = pjp.proceed();  logger.debug("[ CONTROLLER RESULT ] : {}", result);  return result;  }  } |

## Interceptor

API 권한 확인 및 유저 권한, 토큰 유효성을 확인하기 위해 Spring boot에서 제공하는 Interceptor기능을 이용하여 구현하였다.

|  |  |  |
| --- | --- | --- |
| **파일명** | **설명** | **비고** |
| ApiInterceptor | Header 정보에 있는 API\_KEY 가 일치하는지 확인하는 소스 | CommonAPI 에서 사용 |
| AuthorityInterceptor | 로그인한 사용자의 권한을 확인하는 소스 | CommonAPI 에서 사용 |
| TokenInterceptor | 로그인한 사용자의 토큰을 확인하는 소스 | CommonAPI 에서 사용 |
| CommonApiInterceptor | Header 정보에 있는 API\_KEY 가 일치하는지 확인하는 소스 | CommonAPI를 제외한 API에서 사용 |
| ConfirmInterceptor | 로그인한 사용자의 권한 및 토큰을 확인하는 소스 | CommonAPI를 제외한 API에서 사용 |

[표 3‑5] interceptor 종류

* WebMvcConfiguration

\*인터셉터 추가시 WebMvcConfiguration에interceptorRegistry.addInterceptor() 메소드를 이용하여 등록해주어야한다.

|  |
| --- |
| @Override  public void addInterceptors(InterceptorRegistry interceptorRegistry) {  interceptorRegistry.addInterceptor(new ApiInterceptor(commonMapper)).addPathPatterns("/\*\*").order(1);  } |

## Tiles

UI에서 공통된 코드를 줄이기 위해 기본 layout 설정을 해주는 파일

* 작성 예시

|  |
| --- |
| <definition name="base.definition" template="/WEB-INF/tiles/layout.jsp">  <put-attribute name="header" value="/WEB-INF/tiles/header.jsp" />  <put-attribute name="menu" value="/WEB-INF/tiles/menu.jsp" />  <put-attribute name="commonModal" value="/WEB-INF/tiles/commonModal.jsp" />  </definition>  <definition name="login" template="/WEB-INF/views/login.jsp"/>  <definition name="index" extends="base.definition">  <put-attribute name="content" value="/WEB-INF/views/project.jsp" />  </definition> |

|  |  |  |
| --- | --- | --- |
| 속성 | | 설명 |
| definition | Name | 파일 변수명 |
| Extends | 해당 파일을 포함시킬 파일 변수명 |
| put-attribute | name | 해당 파일을 포함시킬 위치 변수명 |
| value | JSP 위치 |

[표 3‑6] Tiles 구조

## 배포

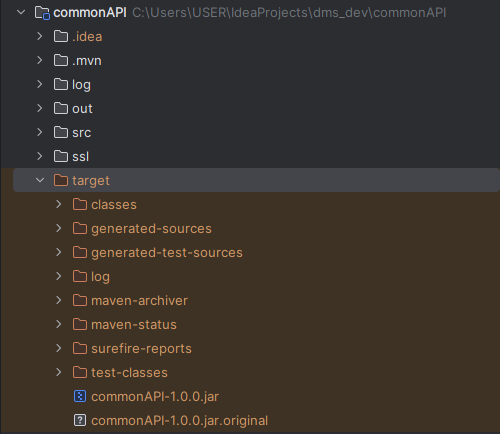
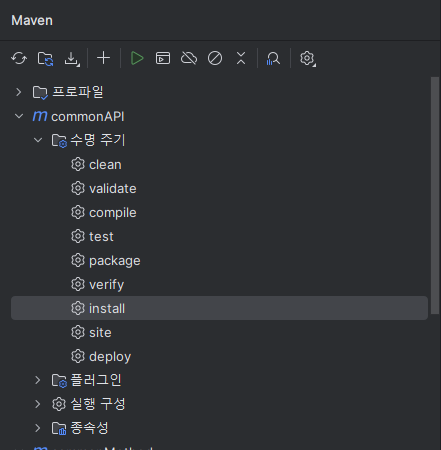
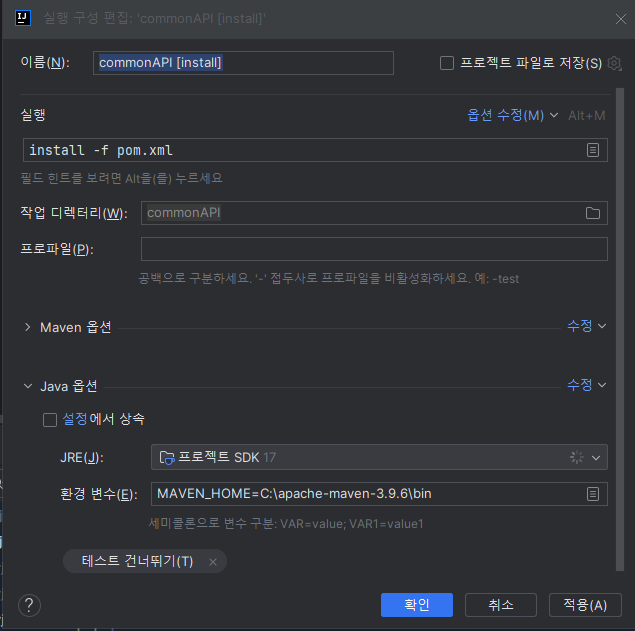
Maven Build 를 통해 .jar / .war 파일을 생성하여 서버에 배포한다.

\* commonMethod 에 변경된 파일이 있다면, commonMethod 파일들을 먼저 빌드 후 다음 시스템을 빌드한다.

### 3.5.1 . 인텔리제이 환경 배포 방법

maven 생명주기 중 install로 선택 후 , 빌드 옵션 중 Java 옵션에서 테스트 건너뛰기 옵션을 선택하여 빌드한다.

해당 과정을 진행하면 소스 폴더 내 ‘target’ 폴더내에 빌드 된 파일을 확인 할 수 있다.

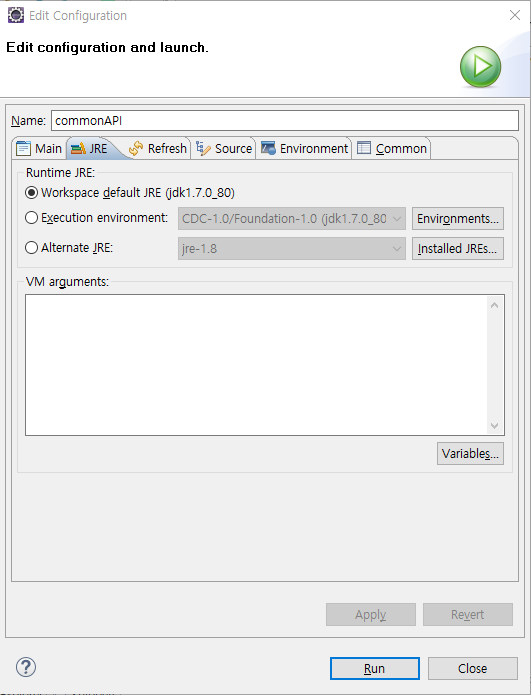
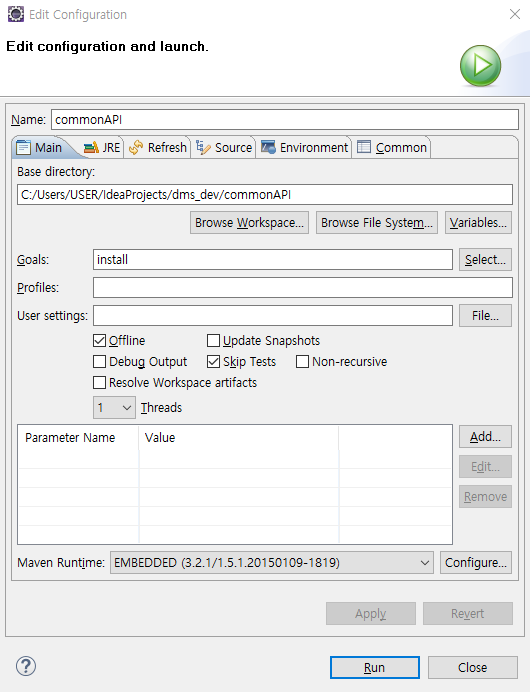
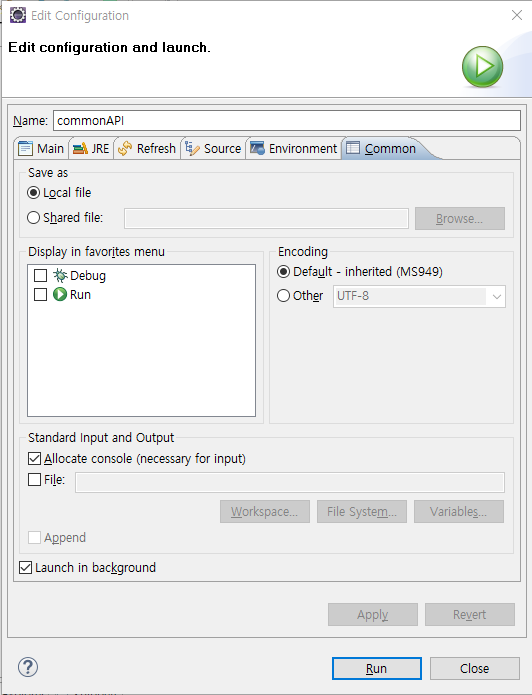


[그림 3‑1] 인텔리제이 배포 방법

### 3.5.2 . 이클립스 환경 배포 방법

Main 탭의 빌드 옵션 중 Goals 에 install입력 후 , Offline, Skip Tests옵션을 선택한다   
JRE 탭에서 1.7.0 버전을 선택한다. Common 탭에서 인코딩 UTF-8로 설정해주고 실행시킨다.

해당 과정을 진행하면 소스 폴더 내 ‘target’ 폴더내에 빌드 된 파일을 확인 할 수 있다.



[그림 3‑2] 이클립스 환경 배포 방법

# COMMON

## API\_INFO

API 가 존재하는지 확인하기위한 테이블

\* API시스템 추가시 업데이트 해야한다.

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| API\_ID | API\_ID |
| API\_KEY | 각 시스템의 고유 API\_KEY |
| API\_NAME | API 명 |
| API\_URL | 각 시스템의 URL |
| API\_CODE | API CODE |

[표 4‑1] API\_INFO 테이블

## USER\_AUTHORITY

사용자 권한을 설정하는 테이블.

\* 권한을 추가시 업데이트 해야한다.

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| AUTHORITY\_ID | AUTHORITY\_ID |
| AUTHORITY\_NAME | 권한 명 |
| SUB\_SYSTEM | 사용할 서브 시스템 API ID (API\_INFO의 API\_ID) / ALL – 전체 |

[표 4‑2] USER\_AUTHORITY 테이블

## MENU

### 4.3.1 . MENU

메뉴 정보를 담고 있는 테이블

\* 메뉴 추가시에 업데이트 해야한다.

* MENU 테이블

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| MENU\_ID | MENU\_ID |
| MENU\_NAME | 메뉴에 보여질 메뉴명 |
| TOP\_MENU\_ID | 상위 MENU\_ID ( 없을시 null) |
| SUB\_SYSTEM | 사용할 서브 시스템 API ID (API\_INFO의 API\_ID) / ALL – 전체 |
| MAPPING\_URL | 해당 메뉴의 메인 API\_URL |
| ICON | 메뉴에 보여질 아이콘 |

[표 4‑3] MENU 테이블

### 4.3.2 . MENU\_AUTHORITY

각 메뉴에 사용자 권한에 따라 CRUD 권한을 설정하는 테이블

\* 권한 , 메뉴, API URL 추가시 업데이트 해야한다.

* MENU\_AUTHORITY 테이블

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| MAPPING\_URL | 권한을 부여할 API URL |
| SUB\_SYSTEM | MAPPING\_URl 을 가지고 있는 시스템 API ID |
| METHOD\_NAME | HTTP TYPE |
| AUTHORITY\_ID | 권한을 줄 사용자 권한ID (USER\_AUTHORITY의 AUTHORITY\_ID) |

[표 4‑4] MENU\_AUTHORITY 테이블

## COMMON\_CODE

시스템상 공통적으로 쓰이는 코드들을 지정해놓은 테이블.

* commonCode 테이블

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| CODE\_ID | CODE\_ID |
| CODE\_NAME | 공통코드명 |
| CODE\_GROUP\_ID | CODE\_GROUP\_ID |
| CODE\_TOP\_ID | 상위 공통코드 ID |

[표 4‑5] COMMON\_CODE 테이블

* commonCodeGroup 테이블

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| CODE\_GROUP\_ID | 공통코드 그룹 아이디 |
| CODE\_GROUP\_NAME | 공통코드 그룹명 |
| SUB\_SYSTEM | 사용할 서브 시스템 API ID (API\_INFO의 API\_ID) / ALL – 전체 |

[표 4‑6] COMMON\_CODE\_GROUP 테이블

* 공통코드그룹ID 명명규칙  
  \* 영어 한글자 + 숫자의 조합으로 지정한다.

|  |  |
| --- | --- |
| CODE\_GROUP\_ID | 설명 |
| T + 숫자 | 소속 관련 코드 |
| P + 숫자 | 프로젝트 관련 코드 |
| I + 숫자 | 산출물 관련 코드 |

[표 4‑7] COMMON\_CODE\_GROUP ID 명명규칙

## MAIL

### MAIL\_INFO

메일 정보를 담은 테이블

\* 메일의 형식이 추가 될 때 마다 업데이트 해야한다

* MENU 테이블

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| MAIL\_INFO\_ID | MAIL\_INFO\_ID |
| MAIL\_TEMPLATE | 메일 형식 |
| SUB\_SYSTEM | 사용할 서브 시스템 API ID (API\_INFO의 API\_ID) / ALL – 전체 |
| MAIL\_TITLE | 보내지는 메일 내용 |

[표 4‑8] MAIL\_INFO 테이블

* MAIL\_INFO 최신 데이터

\* 데이터 추가 시 해당 표를 업데이트 해야한다.

|  |  |  |
| --- | --- | --- |
| MAIL\_TITLE | 사용 상황 | 권한 |
| [인젠트 개발자 센터(IDC)]  프로젝트 검수 상태 변경 | 프로젝트 검수 완료 시 프로젝트 담당자에게 메일을 보낸다 | System Manager, DIRECTOR,  MANAGER (해당 프로젝트 담당자) , DEVELOPER (해당 프로젝트 담당자) |
| [인젠트 개발자 센터(IDC)]  사용자 추가 정보 | 유저 추가시에 추가된 메일을 보낸다. | ALL (본인) |
| [인젠트 개발자 센터(IDC)] 비밀번호 초기화 | 유저 추가시 혹은 비밀번호 초기화시 해당 유저에게 메일을 보낸다 | ALL (본인) |
| [인젠트 개발자 센터(IDC)]  검수 요청 | 프로젝트 검수 완료 시 프로젝트 DIRECTOR 에게 메일을 보낸다 | System Manager, DIRECTOR,  MANAGER (해당 프로젝트 담당자) , DEVELOPER (해당 프로젝트 담당자) |

[표 4‑9] MAIL\_INFO 최신데이터

### 4.5.2 . MAIL\_HISTORY

메일을 보낸 이력을 남가는 테이블

* MAIL\_HISTORY 테이블

|  |  |
| --- | --- |
| **컬럼명** | **설명** |
| MAIL\_HISTORY\_ID | MAIL\_HISTORY\_ID |
| MAIL\_TITLE | 메일 제목 |
| MAIL\_RECIEVER | 메일 수신자 |
| MAIL\_CONTENT | 메일 내용 |
| MAIL\_STATUS | 메일 전송 성공여부 |
| MAIL\_SEND\_DATE | 메일을 보낸 날짜,시간 |

[표 4‑10] MAIL\_HISTORY테이블

## 산출물 파일 다운로드

### 4.6.1. Zip 파일 생성

산출물을 정상적으로 등록 후 검수요청 시 Home디렉토리/Downloads 디렉토리에 프로젝트 명으로 Zip 파일이 생성된다.

### 4.6.2. 산출물 파일 다운로드

산출물 다운로드 버튼 클릭 시 검수요청 시 생성했던Home디렉토리/Downloads 디렉토리의 프로젝트 명.zip 파일이 다운로드 된다.

* 산출물 다운로드 가능한 화면
* 프로젝트 조회 화면의 산출물 다운로드 버튼
* 프로젝트 상세 조회 화면의 산출물 다운로드 버튼
* 프로젝트 검수 상세 조회 화면의 산출물 다운로드 버튼