

[URACLE DB Pitcher] 사용자 가이드

2025 년 02 월

(주) Uracle

목차

1. 개요	4
1.1. 목적	4
1.2. 호출 방법	4
1.3. 연계 DB 포맷	4
2. UMS DB Pitcher 설정 및 구동 가이드	7
2.1. DB Pitcher 설정	7
2.2. PitcherManager 구동	7
2.3. Pitcher 연계 가이드	8
2.3.1. Pitcher Configuration 가이드	8
2.3.1. Pitcher 수신 Customizing 가이드	10
2.3.1. Pitcher 송신 Customizing 가이드	10
3. DBPitcher를 이용하여 대량발송 처리 방법(NC)	11
3.1. CSV 파일 이용	11
3.2. 대상자 테이블(T_UMS_PITCHER_CSV_USER) 이용	12

개정 이력

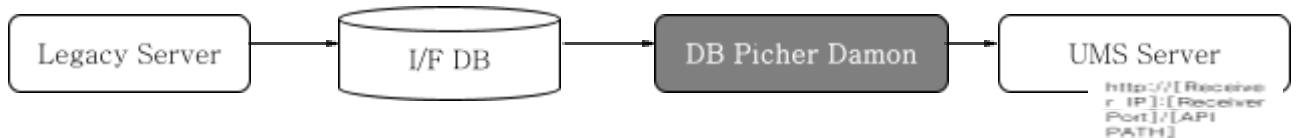
버전	Page	개요	구분	구분
1.0	전체	최초 작성	2025-01-24	유라클
	전체	예시 설명 내용 수정	2025-02-28	유라클

1. 개요

1.1. 목적

본 문서는 UMS 서비스의 발송 요청 연계를 위하여 연계 DB에 저장된 요청 정보를 읽어 UMS 발송 서버로 전송하는 DB Pitcher를 사용하는 방법을 설명한다.

1.2. 호출 방법



I/F DB에 발송 요청 정보가 입력되면 DB Pitcher Daemon은 요청 정보를 UMS API 호출 객체에 Mapping하여 HTTP (POST) API URL 을 호출하고 UMS 서버로 발송한 결과는 T_UMS_PITCHER_LOG 테이블에 기록된다.

※ DB 문자셋은 이모지콘등의 지원 범용성을 위하여 UTF-8(mysql utf8mb4)을 권장하고 있다.

1.3. 연계 DB 포맷

UMS DB Pitcher의 표준 연계 테이블 T_UMS_PITCHER_SEND를 제공하고 있으며 포맷은 UMS API 전문 호출을 위한 정보로 구성되어 있으며 일부 DB 연계를 위한 필드가 포함되어 있다.

※ 세부 항목은 사이트 별 발송 채널 지원 여부와 적용 시스템 전문 포맷에 따라 조정이 필요하다.

[Header 공통]

Column 영문 명	Column 한글 명	Mapping Rule	비고
PITCHER_SEQNO	DB발송요청 시퀀스 번호	자동 생성	
SENDGROUPCODE	발송자그룹코드		
SENDERID	발송자아이디		
TARGET_USER_TYPE	발송대상 유형	기본값 : 'NM' NM : 단일/핸드폰번호필수, NC : CSV	타시스템에서 인터페이스하여 발송시에는 단건 발송일 경우 NM, CSV 발송일 경우 NC
SEND_MACRO_CODE	채널별 발송처리 방법 정의 매크로코드	기본값 MACRO_001	

REPLACE_VARS	개인화 치환변수	ex : {“#{고객명}”:”홍길동”, “#{날짜}”:”2022년 06월 14일”}	대량 CSV 발송 시 생략 가능 메시지 내의 #{고객명} 이 '홍길동'과 매칭됨
TRANS_TYPE	전송 구분	- REAL(실시간) - BATCH(배치)	필수
MSG_TYPE	메세지 발송타입	'I'	필수
RESERVEDATE	예약발송일	YYYY-MM-DD HH24:MM	ex) 2022-07-21 14:50
SERVICECODE	서비스코드	기본값 : 'ALL'	대량 CSV 발송 시 생략 가능
SPLIT_MSG_CNT	분할발송 카운트	단위 : 초	ex) 10 -> 10초 단위로
DELAY_SECOND	분할발송 지연시간(초)	단위 : 건수	ex) 10000 -> 10000건 씩 발송
CUST_TRANSGROUPKEY	고객 그룹 거래식별고유번호.	발송 요청 시스템 구분 코드로 활용	* 요청 시스템별 통계 산출 시 필요함.
CUST_TRANSKEY	고객 거래식별고유번호		연계 시스템에서 식별하기 위한 요청 번호 ex) CMP202208001
MIN_START_TIME	최소 발송 유효시간	24HH:MM	현재 시간이 MIN_START_TIME ~ MAX_END_TIME 사이이면 발송함.
MAX_END_TIME	최대 발송 유효시간	24HH:MM	
FATIGUE_YN	피로도 사용 여부	기본값 : 'N' - Y : 피로도 체크 - N : 피로도 체크 안함	
CALLBACK_NUM	회신전화번호		
CSV_FILE	발송대상자 CSV파일	대량 발송요청 대상 및 치환값 정보 파일	대량 CSV 발송 시 필수
CUID	수신자 아이디		단건 발송시 필수
CNAME	수신자 이름		
MOBILE_NUM	수신자 핸드폰번호		
VAR1	사용자 정의 필드1		
VAR2	사용자 정의 필드2		
VAR3	사용자 정의 필드3		
VAR4	사용자 정의 필드4		
VAR5	사용자 정의 필드5		
VAR6	사용자 정의 필드6		
VAR7	사용자 정의 필드7		
VAR8	사용자 정의 필드8		
VAR9	시스템 활용 JSON맵		

[참고] UMS API 정의서 “2. UMS API 발송 요청 규칙” 참고

[Detail 문자발송]

Column 영문 명	Column 한글 명	Mapping Rule	비고
SMS_TEMPL_ID	SMS 템플릿 아이디		
SMS_TITLE	SMS 제목		
SMS_MSG	SMS 메세지		필수값으로 받음
MMS_IMGURL	MMS 이미지 URL		- 이미지첨부시 MMS로 전송함. [img1, img2] array 형식으로 여러 개 첨부 가능(최대 3개) - 대량 이미지 치환 URL 입력 '대량 이미지 URL 가이드' Sheet 참조

[Detail 카카오킬송]

Column 영문 명	Column 한글 명	Mapping Rule
KKOALT_SVCID	알림톡 서비스아이디	
ALLIMTOLK_TEMPLCODE	알림톡 템플릿코드	
ALLIMTALK_MSG		
KKO_BTNS	카카오 버튼	알림톡, 친구톡 공통

[Detail Push 발송]

Column 영문 명	Column 한글 명	Mapping Rule
APP_ID	앱아이디	
TITLE	제목	
PUSH_MSG	푸시메세지	
PUSH_TEMPL_ID	푸시 템플릿 아이디	
EXT	푸시커스텀확장정보	
SOUNDFILE	사운드파일	기본값 : alert.aif
BADGENO	배지카운트	기본값 : '0' - UPMC 배지 자동 관리 설정 모드 (0: 기능 OFF 1: 기능 ON) - UPMC 배지 자동 관리 설정 모드 (0~999 숫자 입력 가능)
DB_IN	DB저장여부	기본값 : 'Y'

2. UMS DB Pitcher 설정 및 구동 가이드

1.4. DB Pitcher 설정

- DB Pitcher 설정은 설치 디렉토리 conf/config.xml 에 존재한다.

umsPitcher	DB Pitcher 발송 처리 Main 쓰레드의 구동 쓰레드 수, 구동 주기 등 Daemon 쓰레드 설정 정보
ums	DB 요청 정보를 API Object에 맵핑 완료 후 발송하기 위한 UMS Core 발송 엔진 서버 호출 설정 정보
ums_db	DB 요청 건 추출을 위한 DB 설정 정보

FILEDB

발송 큐 정보 관리를 위한 파일 DB 설정 정보

1.5. PitcherManager 구동

Main Class	DBPitcherMain	DB Pitcher 구동 클래스로 설정 로딩 후 MasterRollChecker(요청 데이터 처리), UmsCompleatManager(발송 후 결과 처리)쓰레드를 구동 시키고 TcpAliveConManager(UMS 발송 접속정보 관리) 초기화 한다.
Configuration	LoadConfig	구동 시 변수로 받은 Configuration 파일 위치에서 파일을 읽어 설정을 로딩 한다.
Work Scheduler	MasterRollChecker	주기적으로 DB Pitcher 가 정상 구동 중인지 체크하고 상태 정보를 업데이트 하며 Master 체크 플래그가 True인 경우 masterRollExec를 실행함. - masterRollExec를 호출하여 isMaster가 True인 경우 UmsDBFetcherThread를 실행시켜 DB에 적재된 발송 요청 건을 큐에 적재하며 False인 경우 UmsDBFetcherThread를 종료한다.
DB Fetcher Thread Class	UmsDBFetcherThread	- T_UMS_PITCHER_SEND 테이블에 적재된 발송 요청 건을 UMS_DBP.UQ 파일 큐에 적재 후에 T_UMS_PITCHER_SEND 삭제 Work ConcurrentLinkedQueue에 적재 한다.
발송 Worker Manger	UmsWorkerManager	DBPitcherMain.startDaemon/startRedisWorkers 로 구동되며 내부 큐 파일 DB 또는 Redis에 등록된 일감을 처리 할 수 있는 Worker구동 정보를 설정한다.
발송 Thread Class	UmsWorkerThread	- UMS_DBP.UQ 파일 큐에 적재된 발송 요청건을umsWorkerManager.takeWork하여 UMSWorkBean 객체의 발송 요청 정보를 받아서 CommonService.umsHttpSend에서 UMS Core 엔진 서버에 발송 API 호출 한다. - 발송 성공/실패 결과 처리를 위하여 UmsCompleatManager ConcurrentLinkedQueue에 결과 Object를 적재한다.
종료 Worker Manager	UmsCompleatManager	- UmsCompleatWorkThread.start 호출하여 inPitcherCompletedMsg 큐리를 실행하여 발송 요청 완료 로그를 적재한다. - takeWork로 pitcherWorkQueue에 적재된 일감을 반환한다.

1.6. Pitcher 연계 가이드

2.1.1. Pitcher Configuration가이드

```
<?xml version="1.0" encoding="UTF-8"?>

<configuration>
<umsPitcher>
  <!--
    MasterRoll 구동 여부 확인과 모니터링에 사용될 ID로 구동 시키는 DB Pitcher 별로
    <serverID>를 설정하여야 한다.
  -->
  <serverID>UMS-DBPitcher02</serverID>
</ums>
<ums>
  <useYN>Y</useYN>
  <umsRequestThreadCnt>5</umsRequestThreadCnt>
  <taskWorkerThreadCnt>10</taskWorkerThreadCnt>
  <dbWatchMilliTime>5000</dbWatchMilliTime>
  <maxQueueSize>50000</maxQueueSize>
  <expireSendMsgMinute>90</expireSendMsgMinute>
  <!--발송처리 결과 DB테이블에 저장 여부. 모든 성공 실패는 로그로 남기는게 기본... Default : N 이유는 너무 많은
  ROW가 쌓임.-->
  <completWorkerCnt>5</completWorkerCnt>
  <compleatDataDbSave>Y</compleatDataDbSave>
</ums>
```

```
<monitoring>
  <ip>211.241.199.139</ip>
  <port>28888</port>
</monitoring>
</umsPitcher>
... ..
<!--
  DB Fetcher 쓰레드가 요청 정보를 가져올 DB를 셋팅한다.
-->
<ums_db>
  <dbtype>mysql</dbtype>
  <dbcp_enc>N</dbcp_enc>
  <jdbc>
    <driverClassName>com.mysql.jdbc.Driver</driverClassName>

<url>jdbc:mysql://xxx.xxx.xxx.xxx:3306/UMS30?characterEncoding=UTF-8&zeroDateTimeBehavior=convertToNull</url>

    <username>root</username>
    <password>xxxx</password>

    <minConnectionsPerPartition>0</minConnectionsPerPartition>
    <maxConnectionsPerPartition>50</maxConnectionsPerPartition>
    <idleMaxAgeInSeconds>3600</idleMaxAgeInSeconds>
    <idleConnectionTestPeriodInMinutes>5</idleConnectionTestPeriodInMinutes>
    <connectionTestStatement>select 1</connectionTestStatement>
  </jdbc>
</ums_db>

<!--
#####
# 파일 DB설정
#####
-->
<!--
  DB Pitcher는 기본적으로 Active/Standby 구조로 File DB를 사용하도록 되어 있음.
  File DB 파일명은 구동 시키는 DB Pitcher별로 상이하게 설정하여야 한다.
-->
<FILEDB>
  <USEYN>Y</USEYN>
  <PATH>/Users/bvungheevoo/project/ums/ums-dbpitcher/3.0.x/trunk/data2</PATH>
</FILEDB>
... ..
```

1.1.1. Pitcher 수신 Customizing 가이드

수신 요청의 Customizing이 필요할 경우 UmsDBFetcherThread.run() DB Select 이후 객체 할당 부분의 변경이 필요하며 현재 UnderscoreToCamelCase 자동 맵핑되고 있으며 필요 시 resultMap 재정의가 필요할 수 있음.

```
List<UmsWorkBean> pitcherWorkBeanList = umsPitcherDao.selPitcherMsg(LAST_GET_PITCHER_SEQNO);
... ..
UmsWorkerManager.getInstance().putWork(umsWorkBean);
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<!-- ORACLE 주의 사항 : 오라클의 selPitcherMsg2000 쿼리의 REGDATE는 초까지의 TimeStamp 값이 등록된다. 다른 DB에
적용 할 때 반드시 확인 해야 한다.-->
<mapper namespace="mybatis.umsPitchermsg">
  <select id="selPitcherMsg500" parameterType="hashmap"
resultType="kr.uracle.ums.dbpitcher.worker.ums.UmsWorkBean">
```

```
<![CDATA[
select
  PITCHER_SEQNO ... ..
```

1.1.1. Pitcher 송신 Customizing 가이드

File DB에서 가져온 발송 요청 건을 UMS Core 발송 엔진으로 발송하는 부분은 CommonService umsHttpSend() Method이며 각 API 항목 별로 Customizing이 필요한 부분의 수정이 가능함.

```
public void umsHttpSend(UmsWorkBean umsWorkBean){    ///  
    try {    ///  
        ///  
        ///  
        ///  
        UmsPotalParamVo umsPotalParamVo = new UmsPotalParamVo();    ///  
        ///  
        String ums_host_url = TcpAliveConManager.getInstance().getConHostName();    ///  
        umsPotalParamVo.setUMS_URL(ums_host_url);    ///  
        umsPotalParamVo.setREQ_TRAN_TYPE(TransType.valueOf(umsWorkBean.getTRANS_TYPE()));    ///  
        ///  
        umsPotalParamVo.setSENDGROUPCODE(umsWorkBean.getSENDGROUPCODE());    ///  
        ///  
        umsPotalParamVo.setSENDERID(umsWorkBean.getSENDERID());    ///  
        ///  
        ... ..    ///  
    }    ///  
}
```

2. DBPitcher를 이용하여 대량발송 처리 방법(NC)

메시지를 발송하여 같은 메시지 대상자 중 성공, 실패 통계 내역을 알고싶은 경우 이 방법을 권장한다.

예시) 1건의 메시지 요청(UMS 원장에 1건이 요청된다.) - 100명의 대상자 - 90명 성공 / 10명 실패 등의 결과가 T_UMS_SEND_COUNT에 적재됨

반면, NC 타입이 아닐경우(CSV 이용이 아닐경우) 1건의 메시지 당 1명의 대상자 요청이기 때문에 원장에 1건 성공 or 1건 실패 여부만 확인할 수 있다.

그리고, 개별발송(NM, 1개의 메시지당 1명의 대상자) API 호출 부하를 줄일 수 있으므로 대상자의 건수가 많은 경우(대량발송)에 NC 타입으로 발송하면 유용하다.

2.1. CSV 파일 이용

만들어진 발송대상자/개별치환변수 CSV 파일을 세팅하여 발송하는 방법이다.

step1) T_UMS_PITCHER_SEND 테이블 내에 CSV_FILE 컬럼 내에 만들어진 csv 파일의 절대경로를 세팅한다. (주의: 해당 경로는 DBPitcher가 접근할 수 있는 경로여야 한다.)

step2) TARGET_USER_TYPE 컬럼 내에 "NC" 라고 세팅한다.

step3) 발송할 채널의 메시지를 작성한 뒤 발송 요청한다.

<CSV 파일 예제>

```
testSendReplace2.csv
1  #{아이디}, #{이름}, #{핸드폰번호}, #{날짜}, #{회차}
2  "user1", "홍길동1", "01000011111", "2019-05-01", "1회"
3  "user2", "홍길동1", "01000011111", "2019-05-02", "2회"
4  "user3", "홍길동1", "01000011111", "2019-05-03", "3회"
5  "user4", "홍길동1", "01000011111", "2019-05-04", "4회"
6  "user5", "홍길동1", "01000011111", "2019-05-05", "5회"
7  "user6", "홍길동1", "01000011111", "2019-05-06", "6회"
8  "user7", "홍길동1", "01000011111", "2019-05-07", "7회"
9  "user8", "홍길동1", "01000011111", "2019-05-08", "8회"
10 "user9", "홍길동1", "01000011111", "2019-05-09", "9회"
11 "user10", "홍길동1", "01000011111", "2019-05-10", "10회"
```

2.2. 대상자 테이블(T_UMS_PITCHER_CSV_USER) 이용

대상자 테이블에 대상자/개별치환변수 정보를 insert하여 발송하는 방법이다.

step1) T_UMS_PITCHER_CSV_USER 테이블에 발송대상자/개별치환 정보를 insert 한다.

주의: CUST_TRANSGROUPKEY 에 발송대상자를 그룹핑하는 값을 입력한다. (이 값은 요청할 발송메시지의 발송 대상자 정보와 매칭된다.)

step2) T_UMS_PITCHER_SEND에 발송할 메시지 정보 세팅 및, TARGET_USER_TYPE 에는 “NC”를, CSV_FILE 컬럼에는 빈값을 세팅하여야 한다. 그리고 앞서 T_UMS_PITCHER_CSV_USER에 세팅한 CUST_TRANSGROUPKEY값도 동일 컬럼명에 세팅한다.

step3) 발송 요청

< T_UMS_PITCHER_CSV_USER 테이블 데이터 예제 >

T_UMS_PITCHER_CSV_USER 1 x										
SELECT * FROM T_UMS_PITCHER_CSV_USER Enter a SQL expression to filter results (use Ctrl+Space)										
	SEQNO	CUST_TRANSGROUPKEY	CUST_TRANSKEY	CUID	CNAME	MOBILE_NUM	REPLACE_VARS	REG_DATE	MOD_DATE	
	시퀀스 고유번호	발송메세지 구분 그룹키	발송메세지 고유키	아이디	이름	모바일번호	개별화치환값. JsonObject형식.	등록일	수정일	
1	1	MSG001	000000001	P000000001	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.150655		[NULL]
2	2	MSG001	000000002	P000000002	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152566		[NULL]
3	3	MSG001	000000003	P000000003	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152608		[NULL]
4	4	MSG001	000000004	P000000004	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152626		[NULL]
5	5	MSG001	000000005	P000000005	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152640		[NULL]
6	6	MSG001	000000006	P000000006	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152653		[NULL]
7	7	MSG001	000000007	P000000007	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152666		[NULL]
8	8	MSG001	000000008	P000000008	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152719		[NULL]
9	9	MSG001	000000009	P000000009	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152737		[NULL]
10	10	MSG001	000000010	P000000010	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152752		[NULL]
11	11	MSG001	000000011	P000000011	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152766		[NULL]
12	12	MSG001	000000012	P000000012	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152778		[NULL]
13	13	MSG001	000000013	P000000013	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152790		[NULL]
14	14	MSG001	000000014	P000000014	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152802		[NULL]
15	15	MSG001	000000015	P000000015	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152814		[NULL]
16	16	MSG001	000000016	P000000016	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152826		[NULL]
17	17	MSG001	000000017	P000000017	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152838		[NULL]
18	18	MSG001	000000018	P000000018	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152852		[NULL]
19	19	MSG001	000000019	P000000019	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152866		[NULL]
20	20	MSG001	000000020	P000000020	테스트	01012345678	{"#(가격):"1000","#(날짜):"20241112"}	2024-12-31-01.34.11.152895		[NULL]