

ATA Agent Ver. 1

사용자 설명서

Version 1.1.0



목차

1. 문서 개요	4
1) 소개	4
2) 용어	4
2. ATA 개요	5
1) 알림톡 / 친구톡 메시지 개요	5
2) 알림톡 / 친구톡 메시지 FLOW	5
3. DBMS 연동	6
1) 사용 TABLE	6
2) STORED PROCEDURE	6
3) 테이블에 특정 필드 추가 방법	7
4) 로그 테이블 고정 하기	9
5) SMS/LMS FAILBACK 처리 방법	10
4. MESSAGING SERVICE 사용	11
1) 카카오톡 알림톡 메시지 전송	11
(1) TEXT형 메시지 전송	11
(2) 버튼형 메시지 전송	12

2) 카카오톡 친구톡 메시지 전송	16
(1) TEXT형 메시지 전송	16
(2) 버튼형 메시지 전송	17
(3) 이미지 메시지 전송	20
3) 전송 메시지 암호화 하기	21
4) 광고성메시지 필수 표기사항	22
5) 우선 순위 적용 하여 전송하기	23
6) 부서코드(참조)필드 지정하기	24
5. 전송차단 및 SPAM	25
1) DB TABLE을 통한 SPAM 체크	25
6. USER LIBRARY	26
1) USER LIBRARY 종류	26
(1) Message Data Callback (BIZAlimCallback.java)	26
(2) Data Transform Callback (BIZMTTCallback.java)	28
(3) User Scheduler Callback (BIZUSCCallback.java)	28
(4) ecurity Callback (BIZDBSCallback.java)	29
2) USER LIBRARY COMPILE 및 동작방법	30
(1) User Library Compile 방법	30
(2) User Library 설정	31



1. 문서 개요

1) 소개

본 문서에는 사용자가 ATA를 이용하여 메시지를 전송, 수신 하는 방법에 대하여 설명한다.
ATA의 설치 및 실행 방법은 “ATA 설치매뉴얼 ~.docx” 을 참조하기 바란다.

참고

ATA Agent란?

비즈톡(주)의 카카오톡 알림톡 / 친구톡 Messaging Service를 이용하기 위하여 고객사의 시스템에 배포되는 클라이언트 프로그램을 말한다.

ATA는 카카오톡 알림톡 / 친구톡 메시지를 전송할 수 있다.

2) 용어

- G/W : Biz Talk Message Gateway를 지칭하는 말. RS/TS로 구성됨
- 단말기 : 메시지를 수신할 수 있는 모든 종류의 기기
- 메시지 : 카카오톡 알림톡 / 친구톡 메시지를 지칭하는 말

2. ATA 개요

이 절에서는 ATA의 개요 및 구성요소 및 아키텍처에 대해 설명을 한다.

1) 알림톡 / 친구톡 메시지 개요

문서 개요에서 언급 했듯이 ATA 는 카카오톡 메시지를 전송할 수 있는 서비스를 제공하며, 카카오톡 옐로우톡 아이디를 통하여 1000자 이하의 카카오톡 메시지를 전송하는 서비스 이다.

- 카카오톡 알림톡 (Kakao Talk Alim Talk) : 정보성 데이터에 사용되며, 템플릿을 등록해야 한다.
- 카카오톡 친구톡 (Kakao Talk Friends Talk) : 광고성 데이터에 사용되며, 템플릿을 등록하지 않는다.

2) 알림톡 / 친구톡 메시지 Flow

ATA는 카카오톡 메시지 전송을 위해 ATA, MMSMTCollector, MTSender, MTRPTReceiver, MTDistributor, LogTransfer 프로세스가 실행된다.

- ① MMSMTCollector가 전송테이블(ata_mmt_tran)을 주기적으로 감시하여 전송할 메시지가 있는지 조회한다.
- ② MMSMTCollector가 전송할 메시지가 있으면 메시지를 큐에 적재하고, **테이블의 메시지 상태정보(msg_status) 필드를 "2"로 업데이트 한다.**
- ③ MTSender가 메시지 전송 큐에서 전송할 메시지를 꺼내온다.
- ④ MTSender가 꺼내온 메시지를 G/W로 전송한다.
- ⑤ MTSender가 메시지 전송 후 메시지 전송에 대한 수신 확인을 받는다.
- ⑥ MTRPTReceiver가 비즈톡 G/W로 전송한 메시지 전송 결과를 G/W 로부터 수신 받는다.
- ⑦ MTRPTReceiver가 수신한 전송 결과를 Report Queue 에 적재한다.
- ⑧ MTDistributor가 Report Queue에서 메시지 전송 결과를 꺼내온다.
- ⑨ MTDistributor가 메시지 전송결과를 DB MMS 전송 테이블에 반영한다. **테이블의 메시지 상태정보(msg_status) 필드가 "3"로 업데이트 된다.**
- ⑩ LogTransfer가 전송테이블에서 전송이 완료되어 메시지 상태가 "3"으로 업데이트된 메시지들을 **월별 이력테이블(ata_mmt_log_yyyyMM)로 옮긴다.** 이력테이블로 옮겨진 메시지들은 전송테이블에서 삭제된다.

3. DBMS 연동

이 절에서는 기본적으로 배포하는 표준 테이블 및 Stored Procedure 에 대해서 설명한다.

1) 사용 Table

아래 테이블 목록은 ATA 배포 시 Stored Procedure 을 통해 생성되는 테이블 스키마 이다.

ATA 테이블 목록

구분	이름	설명
공통	ata_banlist	카카오톡 메시지 전송 시 차단할 수 있는 휴대폰 번호를 등록할 수 있는 테이블
메시지	ata_mmt_tran	카카오톡 메시지 전송을 위한 테이블이다.
	ata_mmt_log_yyyymm	카카오톡 메시지 연별/월별 로그 테이블이다.

2) Stored Procedure

ATA 는 프로그램 내부에서 직접적으로 SQL을 사용하지 않으며, 서버에 저장된 Stored Procedure을 Call하여 DBMS와 인터페이스 한다. 이는 테이블 확장이나 변경이 용이하게 하기 위함이다. ATA 초기 설치 후 Table Schema가 변경되는 경우, 같이 배포된 Stored Procedure도 꼭 변경해 줘야 한다.

공통 Stored Procedure 목록

필드	설명
sp_common_create	ata_banlist 테이블을 생성한다.
sp_common_banlist	전송차단번호 목록을 검색한다.

발송관련 Stored Procedure 목록

필드	설명
sp_mmt_create	카카오톡 메시지 서비스 관련 테이블을 생성한다.
sp_mmt_log_create	카카오톡 메시지 서비스에서 로그 테이블을 생성한다.
sp_mmt_tran_select	카카오톡 메시지 개별 전송 데이터를 검색한다.

sp_mmt_update	카카오톡 메시지를 큐에 적재한 후 적재 결과를 갱신한다.
sp_mmt_tran_rslt_update	카카오톡 메시지 전송 결과를 갱신한다.
sp_mmt_tran_log_move	송수신 완료된 메시지를 로그 테이블로 이동한다.
sp_mmt_tran_log_move_past	유효기간이 지난 메시지를 로그 테이블로 이동한다.
sp_mmt_client_select	카카오톡 메시지 발송 시 현재 사용하지 않음
sp_mmt_client_rslt_update	카카오톡 메시지 발송 시 현재 사용하지 않음
sp_mmt_client_log_move	카카오톡 메시지 발송 시 현재 사용하지 않음
sp_mmt_client_log_move_past	카카오톡 메시지 발송 시 현재 사용하지 않음
sp_mmt_file_select	카카오톡 메시지 발송 시 현재 사용하지 않음
sp_mmt_file_update	카카오톡 메시지 발송 시 현재 사용하지 않음
sp_custom_failback_insert	카카오톡 메시지 발송 실패시 Failback 처리를 한다.

3) 테이블에 특정 필드 추가 방법

이 절에서는 고객이 ATA 테이블에 특정 필드를 추가하는 방법에 대해서 설명한다. ATA는 모듈 수정 없이 프로시저 수정만으로 필드 추가 및 수정이 가능하다. 필드 추가 방법은 각 서비스 별로 비슷하며, 여기서는 Oracle 기준 **TRAN_ETC1 필드를 추가**하는 방법을 설명한다. 각 전체 내용은 일부를 발췌한 것이며, 적색으로 표시된 부분이 변경된 사항이며, 작업 시 Comma(,)에 유의하기 바란다.

프로시저 명	변경 내용
sp_mmt_create	<pre> create table ata_mmt_tran (mt_pr NUMBER(11) NOT NULL, mt_refkey VARCHAR2(20), priority CHAR(2) default "S" NOT NULL, date_client_req TIMESTAMP NOT NULL , ... 중략 ... crypto_yn CHAR(1) default "Y", ata_id CHAR(2) default " ", reg_date TIMESTAMP default sysdate, TRAN_ETC1 VARCHAR2(160), CONSTRAINT pk_ata_mmt_tran PRIMARY KEY (mt_pr) using index tablespace TABLE_SPACE) tablespace TABLE_SPACE'; </pre>

	<pre> create global temporary table ata_mmt_log_temp (mt_pr NUMBER(11) NOT NULL, mt_refkey VARCHAR2(20), priority CHAR(2) default "S" NOT NULL, date_client_req TIMESTAMP NOT NULL , ... 중략 ... crypto_yn CHAR(1) default "Y", ata_id CHAR(2) default " ", reg_date_tran TIMESTAMP, reg_date TIMESTAMP default sysdate, TRAN_ETC1 VARCHAR2(160)) on commit delete rows ; </pre>
sp_mmt_log_create	sp_mmt_create 와 같은 방식으로 적용
sp_mmt_tran_log_move	<pre> INSERT INTO ata_mmt_log_temp SELECT mt_pr, mt_refkey, priority, date_client_req, ... 중략 ... crypto_yn, ata_id, reg_date, sysdate, tran_etc1 FROM ata_mmt_tran WHERE msg_status = '3' AND TO_CHAR (date_client_req, 'yyyymmdd') = v_date_client_req AND ROWNUM <= 2000; sql_string := ' INSERT INTO ata_mmt_log_' v_log_table ' SELECT mt_pr, mt_refkey, </pre>

	priority, date_client_req, ... 중략 ... crypto_yn, ata_id, reg_date_tran, sysdate tran_etc1 FROM ata_mmt_log_temp ';
sp_mmt_tran_log_move_past	sp_mmt_tran_log_move 와 같은 방식으로 적용

4) 로그 테이블 고정 하기

이 절에서는 ATA 로그 테이블을 고정 또는 연별 로그테이블로 적용하는 방법에 대해 설명한다. 필드 추가 방법은 각 서비스 별로 비슷하며, 여기서는 MySQL 기준 **ata_mmt_log_table** 으로 고정하여 사용하는 방법을 설명한다. 적색으로 표시된 부분이 변경된 사항이며, 작업 시 Comma(,)에 유의하기 바란다.

프로시저 명	변경 내용
sp_mmt_log_create	<pre> /* create ata_mmt_log_table table */ IF p_log_table <> " THEN SET @sql = CONCAT(" CREATE TABLE IF NOT EXISTS ata_mmt_log_table (mt_pr int(11) NOT NULL, mt_refkey varchar(20), priority char(2) NOT NULL default 'S', date_client_req datetime NOT NULL default '1970-01-01 00:00:00', content varchar(255) NOT NULL, callback varchar(25) NOT NULL, </pre>
sp_mmt_tran_log_move	<pre> INSERT INTO ata_mmt_log_temp SELECT mt_pr, mt_refkey, priority, date_client_req, content, </pre>

	<pre> SET @sql_insert = CONCAT(" INSERT INTO ata_mmt_log_table SELECT mt_pr, mt_refkey,</pre>
sp_mmt_tran_log_move_past	sp_mmt_tran_log_move 와 같은 방식으로 적용

5) SMS/LMS FAILBACK 처리 방법

이 절에서는 ATA 에서 카카오톡으로 발신이 실패할 경우(결과 코드가 1000이 아닌 경우) 고객센터에서 사용중인 SMS/LMS 업체를 통하여 Failback 처리를 할 수 있는 방법에 대해 설명한다. 단 고객센터에서 SMS/LMS를 알림톡처럼 Database에 Insert하여 사용할 경우에만 적용 가능하다.

여기서는 MySQL을 기준으로 설명한다.

프로시저 명	변경 내용
sp_custom_failback_insert	<pre> DECLARE v_len INT; DECLARE v_mt_pr INT(11); DECLARE v_mt_refkey VARCHAR(20); DECLARE v_subject VARCHAR(40); DECLARE v_content VARCHAR(4000); DECLARE v_callback VARCHAR(25); DECLARE v_recipient_num VARCHAR(25); IF(p_mt_report_code_ib <> '1000') THEN SELECT LENGTH(content) , mt_pr , mt_refkey , subject , content , callback , recipient_num INTO v_len , v_mt_pr , v_mt_refkey , v_subject , v_content , v_callback , v_recipient_num FROM ATA_MMT_TRAN WHERE mt_pr = p_client_msg_key; IF(v_len > 80) THEN -- LMS 테이블에 값 삽입 ELSEIF(v_len <= 80) THEN -- SMS 테이블에 값 삽입 END IF; END IF;</pre>

4. Messaging Service 사용

1) 카카오톡 알림톡 메시지 전송

(1) TEXT형 메시지 전송

ATA 사용자는 메시지를 발송하기 위해서 사전에 비즈톡(주)을 통하여 발신 프로필 키를 발급받고, 템플릿을 등록해야 한다. 등록되지 않은 템플릿은 발송 시 실패처리 된다.

이 절에서는 Text형 알림톡 메시지를 발송하는 방법에 대해 설명한다.

예시 데이터)

- 프로필 키: "aaaaa22222bbbbbb33333ccccc44444ddddd55555"
- 템플릿 메시지: "Test Message 입니다."
- 템플릿 코드: "A000_00"

입력 SQL

▶ Oracle의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다', ' ', '1', '1008', '01000000000',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_00');
```

▶ MSSQL의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code)
values(getdate(), ' ', 'Test Message 입니다', ' ', '1', '01000000000', '1008',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_00');
```

▶ MySQL/MariaDB의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code)
values(sysdate(), ' ', 'Test Message 입니다', ' ', '1', '01000000000', '1008',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_00');
```

(2) 버튼형 메시지 전송

ATA 사용자는 메시지를 발송하기 위해서 사전에 비즈톡(주)을 통하여 발신 프로필 키를 발급받고, 템플릿을 등록해야 한다. 버튼형 메시지를 발송하기 위해서는 템플릿 등록 시 버튼의 type과 버튼을 클릭하였을 때 이동할 메시지의 URL을 등록하여야 한다. 등록되지 않은 템플릿은 발송 시 실패처리 된다.

다음은 생성타입 / 버튼타입(5가지) 버튼 템플릿을 발송하는 방법에 대해 설명한다.

→ 버튼템플릿은 **3가지 생성타입**으로 전송할 수 있다. 세가지 버튼 생성타입 중 사용하기 편한 타입을 선택해 사용하면 된다.

1. Format String

2. JSON

3. XML

→ 버튼템플릿에는 **5가지 버튼타입**이 있다.

버튼타입	표기	기능	주의사항
배송조회	DS	메시지 내용에서 택배사명, 송장번호를 파싱하여 카카오검색 배송조회 페이지로 이동	-필수값&순서 버튼명+버튼타입
			-지원 택배사 KG로지스,우체국택배, 로젠택배, CJ대한통운, 일양로지스, 현대택배, GTX 로지스, FedEx, 한진택배,경동택배, 합동택배,롯데택배
웹링크	WL	버튼 클릭 이벤트 발생시 인앱 브라우저에서 웹페이지 실행	-필수값&순서 버튼명+버튼타입 +버튼링크
			-템플릿등록 시 이동할 Mobile/PC URL 설정

앱링크	AL	버튼 클릭 이벤트 발생시 유저OS환경에 따라 앱 커스텀스킴 실행	-필수값&순서 버튼명+버튼타입+버튼 링크
			-템플릿등록 시 Android/iOS 환경에서 실행할 URL 설정
봇키워드	BK	봇 또는 상담원에게 유저가 수신한 '버튼명'을 전달	-필수값&순서 버튼명 + 버튼타입
메시지전달	MD	봇 또는 상담원에게 유저가 수신한 '버튼명+메시지본문'을 전달	-필수값&순서 버튼명 + 버튼타입

→ 버튼형 메시지 전송시에는 TEXT형 메시지 전송에 두 컬럼을 추가한다.

kko_btn_type	kko_btn_info
Format String - 해당 컬럼값 '1' 기입 JSON - 해당 컬럼값 '2' 기입 XML - 해당 컬럼값 '3' 기입	kko_btn_type에서 지정한 생성타입으로 버튼명과 버튼타입, 버튼링크를 기입

→ 다음은 **버튼형 메시지 전송 예시**이다.

< 예시 데이터 >

- 프로필 키 : "aaaaa22222bbbbbb33333ccccc44444ddddd55555"
- 템플릿 메시지 : "Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다."
- 템플릿 코드 : "A000_01"
- 버튼명(버튼타입) : 배송조회하기(DS), 주문내역 보러가기(WL), 모바일에서 보기(AL)
- mobile_url/pc_url : <http://www.biztalk.co.kr>
- scheme_ios/scheme_android : bizapps://open

입력 SQL

▶ Oracle ▶ Format String 의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code , kko_btn_typ,kko_btn_info)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합
니다.', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01', '1', '배송조회
하기^DS');
```

▶ Oracle ▶ JSON 의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code , kko_btn_typ,kko_btn_info)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합
니다.', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01', '2',
 '{"button":[{"name":"배송조회하기","type":"DS"}]}');
```

▶ Oracle ▶ XML 의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code , kko_btn_typ,kko_btn_info)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합
니다.', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01', '3', '<?xml
version="1.0"?> <button_info> <button> <name> 배송조회하기
</name> <type>DS</type> </button> </button_info>');
```

▶ MSSQL ▶ Format String 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type,kko_btn_info)
values(getdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01','1', 주문내역 보러가기
^WL^http://www.biztalk.co.kr^http://www.biztalk.co.kr');
```

▶ MSSQL ▶ JSON 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type,kko_btn_info)
values(getdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
```

```
'01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01';2', '{"button":[{"name":"주문내역 보러가기","type":"WL","url_pc":"http://www.biztalk.co.kr","url_mobile":"http://www.biztalk.co.kr"}]'});
```

▶ MSSQL ▶ XML 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type, sender_key, template_code , kko_btn_type, kko_btn_info)
values(getdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01';3', '<?xml
version="1.0"?><button_info><button><name>주문내역 보러가기
</name><type>WL</type><mobile_url>http://www.biztalk.co.kr</mobile_url><pc_url>http://www.biztalk.co.kr
</pc_url></button></button_info>');
```

▶ MySQL/MariaDB ▶ Format String 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type, sender_key, template_code , kko_btn_type, kko_btn_info)
values(sysdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01' ;1', '모바일에서 보기
^AL^biztalkapps://open^biztalkapps://open');
```

▶ MySQL/MariaDB ▶ JSON 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type, sender_key, template_code , kko_btn_type, kko_btn_info)
values(sysdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01' ;2', '{"button":[{"name":"모바일에서 보기","type":"AL","scheme_ios":"biztalkapps://open","scheme_android":"biztalkapps://open"}]'});
```

▶ MySQL/MariaDB ▶ XML 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type, sender_key, template_code , kko_btn_type, kko_btn_info)
values(sysdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01' ;3', '<?xml
version="1.0"?><button_info><button><name>모바일에서 보기
</name><type>AL</type><scheme_android>biztalkapps://open</scheme_android><scheme_ios>biztalkapps:
//open</scheme_ios></button></button_info> ');
```

2) 카카오톡 친구톡 메시지 전송

(1) TEXT형 메시지 전송

ATA 사용자는 친구톡 메시지를 발송하기 위해서 사전에 비즈톡(주)을 통하여 발신 프로필 키를 발급받아야 한다. 친구톡의 경우, 템플릿은 사용하지 않는다.

이 절에서는 친구톡 TEXT형 메시지를 발송하는 방법에 대해 설명한다.

예시 데이터)

- 프로필 키: "aaaaa22222bbbbbb33333ccccc44444ddddd55555"

입력 SQL

▶ Oracle의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다', ' ', '1', '1009', '01000000000',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555');
```

▶ MSSQL의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key)
values(getdate(), ' ', 'Test Message 입니다', ' ', '1', '01000000000', '1009',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555');
```

▶ MySQL/MariaDB의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key)
values(sysdate(), ' ', 'Test Message 입니다', ' ', '1', '01000000000', '1009',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555');
```

(2) 버튼형 메시지 전송

ATA 사용자는 메시지를 발송하기 위해서 사전에 비즈톡(ㄴ)을 통하여 발신 프로필 키를 발급받고, 템플릿을 등록해야 한다. 버튼형 메시지를 발송하기 위해서는 템플릿 등록 시 버튼의 type과 버튼을 클릭하였을 때 이동할 메시지의 URL을 등록하여야 한다. 등록되지 않은 템플릿은 발송 시 실패처리 된다.

이 절에서는 생성타입과 4가지 버튼타입, 버튼 템플릿을 발송하는 방법에 대해 설명한다.

버튼템플릿은 **3가지 생성타입**으로 전송할 수 있다. 세가지 생성타입 중 사용하기 편한 타입을 선택해 사용하면 된다.

1. Format String

2. JSON

3. XML

버튼템플릿에는 **4가지 버튼타입**이 있다.

***알림톡 버튼 5가지, 친구톡 버튼 4가지(배송조회버튼이 지원되지 않는다)**

버튼타입	표기	기능	주의사항
웹링크	WL	버튼 클릭 이벤트 발생시 인앱 브라우저에서 웹페이지 실행	-필수값&순서 버튼명+버튼타입 +버튼링크
			-템플릿등록 시 이동할 Mobile/PC URL 설정
앱링크	AL	버튼 클릭 이벤트 발생시 유저OS환경에 따라 앱 커스텀스킴 실행	-필수값&순서 버튼명+버튼타입 +버튼링크
			-템플릿등록 시 Android/iOS 환경에서 실행할 URL 설정
봇키워드	BK	봇 또는 상담원에게 유저가 수신한 '버튼명'을 전달	-필수값&순서 버튼명 + 버튼타입

메시지전달	MD	봇 또는 상담원에게 유저가 수신한 '버튼명+메시지본문'을 전달	-필수값&순서 버튼명 + 버튼타입
-------	----	---	-----------------------

버튼형 메시지 전송시에는 TEXT형 메시지 전송에 두 컬럼을 추가한다.

kko_btn_type	kko_btn_info
Format String - 해당 컬럼값 '1' 기입 JSON - 해당 컬럼값 '2' 기입 XML - 해당 컬럼값 '3' 기입	kko_btn_type에서 지정한 생성타입으로 버튼명과 버튼타입, 버튼링크를 기입

다음은 **버튼형 메시지 전송 예시**이다

< 예시 데이터 >

- 프로필 키 : "aaaaa22222bbbbbb33333ccccc44444ddddd55555"
- 템플릿 메시지 : "Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다."
- 템플릿 코드 : "A000_01"
- 버튼명(버튼타입) : 상담하기(MD), 주문내역 보러가기(WL), 모바일에서 보기(AL)
- mobile_url/pc_url : <http://www.biztalk.co.kr>
- scheme_ios/scheme_android : bizapps://open

입력 SQL

▶ Oracle ▶ Format String 의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code , kko_btn_typ,kko_btn_info)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합
니다.', ' ', '1', '01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01', '1', '상담하기
^MD');
```

▶ Oracle ▶ JSON 의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code , kko_btn_typ,kko_btn_info)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합
니다.', ' ', '1', '01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01', '2',
{"button":[{"name":"상담하기","type":"MD"}]});
```

▶ Oracle ▶ XML 의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code , kko_btn_type,kko_btn_info)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합
니다.', ' ', '1', '01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01', '3', '<?xml
version="1.0"?><button_info><button><name>상담하기
</name><type>MD</type></button></button_info>');
```

▶ MSSQL ▶ Format String 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type,kko_btn_info)
values(getdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01', '1','주문내역 보러가기
^WL^http://www.biztalk.co.kr^http://www.biztalk.co.kr');
```

▶ MSSQL ▶ JSON 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type,kko_btn_info)
values(getdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01','2', '{"button":[{"name":"주문
내역 보러가기","type":"WL","url_pc":"http://www.biztalk.co.kr","url_mobile":"http://www.biztalk.co.kr"}]}');
```

▶ MSSQL ▶ XML 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type,kko_btn_info)
values(getdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01','3', '<?xml
version="1.0"?><button_info><button><name>주문내역 보러가기
</name><type>WL</type><mobile_url>http://www.biztalk.co.kr</mobile_url><pc_url>http://www.biztalk.co.kr
</pc_url></button></button_info>');
```

▶ MySQL/MariaDB ▶ Format String 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type, kko_btn_info)
```

```
values(sysdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01' , '1', '모바일에서 보기
^AL^biztalkapps://open^biztalkapps://open');
```

▶ MySQL/MariaDB ▶ JSON 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type, kko_btn_info)
values(sysdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01' , '2', '{"button":[{"name":"모바
일에서 보기","type":"AL","scheme_ios":"biztalkapps://open","scheme_android":"biztalkapps://open"}]}');
```

▶ MySQL/MariaDB ▶ XML 의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, template_code , kko_btn_type, kko_btn_info)
values(sysdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_01' , '3',
'<?xml version="1.0"?><button_info><button><name>모바일에서 보기
</name><type>AL</type><scheme_android>biztalkapps://open</scheme_android><scheme_ios>biztalkapps:
//open</scheme_ios></button></button_info>');
```

(3) 이미지 메시지 전송

이 절에서는 친구톡 이미지형 메시지를 발송하는 방법에 대해 설명한다.
 전송하고자하는 이미지는 비즈톡을 통하여 카카오에 선 등록절차가 필요하다.
 카카오에 이미지를 선 등록할 경우 해당 이미지의 URL을 받을 수 있다.

예시 데이터)

- 프로필 키: "aaaaa22222bbbbbb33333ccccc44444ddddd55555"

-Image URL :

http://mud-kage.kakao.com/dn/eB7O8/btqeLiTBa68/iawhBWNVO8znTWKftdFm01/img_l.jpg

(카카오에 등록된 이미지의 URL)

- Image Link : <http://www.biztalk.co.kr>(이미지 클릭시 이동할 페이지의 URL)

입력 SQL

▶ Oracle의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key , img_url , img_link)
```

```
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다', ' ', '1', '1009', '01000000000',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'http://mud-
kage.kakao.com/dn/eB7O8/btqeLiTBa68/iawhBWNVO8znTWKftdFm01/img_l.jpg', 'http://www.biztalk.co.kr');
```

▶ MSSQL의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key , img_url , img_link)
values(getdate(), ' ', 'Test Message 입니다', ' ', '1', '01000000000', '1009',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'http://mud-
kage.kakao.com/dn/eB7O8/btqeLiTBa68/iawhBWNVO8znTWKftdFm01/img_l.jpg', 'http://www.biztalk.co.kr');
```

▶ MySQL/MariaDB의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key , img_url , img_link)
values(sysdate(), ' ', 'Test Message 입니다', ' ', '1', '01000000000', '1009',
'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'http://mud-
kage.kakao.com/dn/eB7O8/btqeLiTBa68/iawhBWNVO8znTWKftdFm01/img_l.jpg', 'http://www.biztalk.co.kr');
```

3) 전송 메시지 암호화 하기

ATA는 인증서버 인증하는 과정에서 ATA가 생성하여 교환하는 대칭키로 메시지를 암호화하여 전송하는 기능을 제공한다. 대칭키 암호화 알고리즘은 비즈톡(Biz Talk)과 Messaging Service 계약 시 인증서버에 설정할 수 있다. 인증서버에 설정 이후 전송테이블의 암호화 여부(crypto_yn)을 'Y'로 설정(default : Y)하면, AES 또는 Seed 알고리즘으로 암호화하여 송수신 한다.

입력 SQL

▶ Oracle

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code, crypto_yn) values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test
Message 입니다', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbbbb33333ccccc44444ddddd55555', 'A000_00',
'Y');
```

4) 광고성 메시지 필수 표기사항

친구톡 메시지의 경우(알림톡 제외), ad_flag 파라미터로 광고성 메시지 필수 표기사항을 지원한다.
(ad_flag 기본값은 Y 로 설정되어 있다.)

이는 전송 시, 기존 프로필영역이 (광고)로 표시되며, 메시지 하단에 광고성메시지 수신거부 방법이 안내된다.

추가적인 설정(ad_flag 파라미터를 N 으로 설정)으로 기존 프로필영역에 프로필명으로 표시되며, 메시지 하단에 광고성메시지 수신거부 방법이 노출되지 않는다.

아래는 광고성메시지 수신거부 방법 노출이 되지 않는 예시이다.

입력 SQL

▶ Oracle의 경우

```
insert into ata_mmt_tran (mt_pr, date_client_req, subject, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code , kko_btn_typ,kko_btn_info,ad_flag)
values(sq_ata_mmt_tran_01.nextval, sysdate, ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동
합니다.', ' ', '1', '01000000000', '1009', 'aaaaa22222bbbb33333cccc44444ddddd55555', 'A000_01', '1', '상답
하기^MD','N');
```

▶ MSSQL의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, kko_btn_type,kko_btn_info,ad_flag)
values(getdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbb33333cccc44444ddddd55555', '1','주문내역 보러가기
^WL^http://www.biztalk.co.kr^http://www.biztalk.co.kr','N');
```

▶ MySQL/MariaDB의 경우

```
insert into ata_mmt_tran (date_client_req, subject, content, callback, msg_status, recipient_num, msg_type,
sender_key, kko_btn_type, kko_btn_info , ad_flag)
values(sysdate(), ' ', 'Test Message 입니다. 아래 버튼을 누르면 홈페이지로 이동합니다.', ' ', '1',
'01000000000', '1009', 'aaaaa22222bbbb33333cccc44444ddddd55555', '1', '모바일에서 보기
^AL^biztalkapps://open^biztalkapps://open', 'N');
```

5) 우선 순위 적용 하여 전송하기

ATA는 메시지 전송 시 기본적으로 3종류(Slow/Fast/Very Fast)의 우선순위를 적용하여 발송할 수 있다. 우선순위는 메시지 전송테이블 우선순위(priority) 필드를 'S', 'F', 'VF' 로 설정하여 전송하면 된다. VF는 가장 최우선으로 보내는 메시지이며, F, S의 순서대로 전송하게 된다. 우선순위를 적용하지 않고 일반적으로 보낼 때는 기본값인 'S' 으로 설정하여 보내면 된다.

입력 SQL

▶ Very Fast

```
insert into ata_mmt_tran (mt_pr, priority, date_client_req, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code) values(sq_ata_mmt_tran_01.nextval, 'VF', sysdate, ' ', 'Test Message 입니
다', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbb33333cccc44444ddddd55555', 'A000_00');
```

▶ Fast

```
insert into ata_mmt_tran (mt_pr, priority, date_client_req, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code) values(sq_ata_mmt_tran_01.nextval, 'F', sysdate, ' ', 'Test Message 입니
다', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbb33333cccc44444ddddd55555', 'A000_00');
```

▶ Slow (기본설정)

```
insert into ata_mmt_tran (mt_pr, priority, date_client_req, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code) values(sq_ata_mmt_tran_01.nextval, 'S', sysdate, ' ', 'Test Message 입니
다', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbb33333cccc44444ddddd55555', 'A000_00');
```

중요

ata.cf 파일에서의 priority 관련 설정

우선순위 사용여부 기본값은 false 이다. 우선순위를 적용하고자 한다면 ata.cf에서 아래 값을 변경 해 줘야 한다.

VF 사용 : priority.use.mms.veryfast = 1

F 사용 : priority.use.mms.fast = 1

발송비율

VF, F, S 발송비율 설정 (2:1:1로 설정할 경우)

priority.cnt.mms.veryfast = 2

priority.cnt.mms.fast = 1

priority.cnt.mms.normal = 1

6) 부서코드(참조)필드 지정하기

ATA는 메시지 발송 시 부서별 메시지 구분을 위한 참조 필드를 지정할 수 있다. 만약 참조 키 필드로 정산이 필요할 경우 **mt_refkey** 필드를 지정하여 발송 한다.

입력 SQL

▶ 부서별 정산을 위한 발송(Oracle)

```
insert into ata_mmt_tran (mt_pr, mt_refkey, date_client_req, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code) values(sq_ata_mmt_tran_01.nextval, 'deptA', sysdate, ' ', 'Test
Message 입니다', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbb33333ccccc44444ddddd55555', 'A000_00');
```

```
insert into ata_mmt_tran (mt_pr, mt_refkey, date_client_req, content, callback, msg_status, recipient_num,
msg_type, sender_key, template_code) values(sq_ata_mmt_tran_01.nextval, 'deptB', sysdate, ' ', 'Test
Message 입니다', ' ', '1', '01000000000', '1008', 'aaaaa22222bbbb33333ccccc44444ddddd55555', 'A000_00');
```


5. 전송차단 및 SPAM

ATA 는 고객 SPAM Table을 통하여 메시지 발송 시 전송 차단 및 SPAM 체크 기능을 제공한다. 이 절에서는 SPAM 체크 기능에 대하여 설명한다. 전송차단 및 SPAM 기능을 사용하지 않는다면 이 절을 건너뛰어도 상관 없다.

1) DB Table을 통한 SPAM 체크

차단번호 리스트테이블에 등록된 데이터에 대해 서비스 별 수신번호, 메시지 내용에 대해 전송 차단 기능을 제공한다.

ata_banlist 차단 예시

▶ 수신번호 차단

```
insert into ata_banlist (service_type, ban_seq, ban_type, content, send_yn, allow_yn, ban_status_yn, reg_date,
update_date) values('3','1','R','01000000000','N','Y','Y',sysdate(),sysdate());
```

▶ message 정규식 체크

```
insert into ata_banlist (service_type, ban_seq, ban_type, content, send_yn, allow_yn, ban_status_yn, reg_date,
update_date) values('3','3','T','bibirace.com','N','Y','Y',sysdate(),sysdate());
```

중요

수신번호 차단인 경우 전송메시지 검색 Stored Procedure에서 Outer Join 으로 정보를 가져 오므로, ATA 을 재 기동 안 해도 자동으로 반영된다. content체크에 대해서는 ATA 을 재 기동 해야 한다.

6. User Library

ATA 는 각 서비스 별로 특정 프로세스 수행 후 고객이 작성한 User Library 를 호출하는 기능을 제공한다. 이 절에서는 User Library를 개발, 사용법에 대해 설명한다. User Library Call 기능을 사용하지 않는다면 이 절을 건너뛰어도 상관 없다.

중요

ATA는 User Library 기능을 위한 Interface를 정의하여 고객은 이 Interface를 구현하는 Class를 작성하면 된다. 각 클래스 Sample은 ATA 배포 시 \$ATA_HOME/src 폴더에 같이 배포된다.

1) User Library 종류

ATA에서 제공 되는 User Library는 아래와 같다.

(1) Message Data Callback (BIZAlimCallback.java)

메시지 전송 과정에서 메시지 전송 또는 전송결과 수신 후 수행 후 호출 되는 프로세스를 정의한 Interface이다. 주로 결과 수신 후 특정 Job을 수행하고자 할 경우 사용된다.

- **Void doAfterEnqueue (**
 int dbIndex,
 String key,
 int seq,
 String recipientNum,
 String callback,
 String msg
)

설명	• DB 테이블에서 조회하여 전송 큐에 enqueue 후 호출되는 메소드
Parameter	<ul style="list-style-type: none"> • dbIndex : 연결할 Database Index. 하나의 ATA가 2개 이상의 DB 연결시 사용함. • key : 메시지 키 (mt_pr). • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할 때 순번으로 사용함. • recipientNum : 수신자 번호 • callback : 콜백 번호 (발신자 번호) • msg : 송신 메시지
Return	• void

- **Void doAfterSend (**
 String key,
 String code
)

설명 • 전송 큐에서 **dequeue** 하여 비즈톡 G/W 송신 후 호출되는 메소드

Parameter • key : 메시지 키 (mt_pr).
 • code : 비즈톡 송신 결과 코드

Return • void

- **Void doAfterReceive (**
 String key,
 int seq,
 String recipientNum,
 String code,
 int carrier,
 long date
)

설명 • 전송 결과를 수신하여 리포트 큐에 **enqueue** 후 호출되는 메소드

Parameter • key : 메시지 키 (mt_pr).
 • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할 때 순번으로 사용함.
 • recipientNum : 수신자 번호
 • code : 이통사 결과코드
 • carrier : 최종 착신망 정보 (KAKAO)
 • date : 단말기 도착 시간

Return • void

- **Void doAfterDequeue (**
 int dbIndex,
 String key,
 int seq,
 String recipientNum,
 String code,
 int carrier,
 long date
)

설명 • 전송 결과를 리포트 큐에서 **dequeue** 후 호출되는 메소드
(모든 전송 과정이 끝난 후 호출되는 메소드)

Parameter • dbIndex : 연결할 Database Index. 하나의 ATA가 2개 이상의 DB에 연결시 사용함.
• key : 메시지 키 (mt_pr).
• seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할 때 순번으로 사용함.
• recipientNum : 수신자 번호
• code : 이통사 결과코드
• carrier : 최종 착신망 정보 (KAKAO)
• date : 단말기 도착 시간

Return • void

(2) Data Transform Callback (BIZMTTCallback.java)

메시지 전송을 위해 DB에서 Record 검색 후 특정 필드의 값을 가공하기 위해 호출되는 프로세스를 정의한 Interface이다. 주로 휴대번호, 메시지가 암호화하여 사용하거나, 값을 변경하여 전송하고자 할 경우 사용된다.

• **Boolean doMmtTransform (**
 RefObject refObj
)

설명 • 데이터 조회 후 변환된 데이터를 **return** 받기 위해 호출되는 메소드

Parameter • refObj : 메시지 관련 참조 개체. 수신번호, 회신번호, 메시지 등을 포함하고 있다.
그 외에 여분의 필드 4개가 있다.

Return • 변환 성공 : true
• 변환 실패 : false

(3) User Scheduler Callback (BIZUSCCallback.java)

사용자 정의 Scheduler 프로세스를 정의한 Interface이다. 주로 특정 주기로 Job을 수행하고자 할 경우 사용된다.

• **Void run (**
 int dbIndex,
 int dbCount
)

설명 • 설정된 시간마다 동작하게 되어있는 사용자 콜백 함수

Parameter • dbIndex : 연결할 Database Index (기본적으로 0)
• dbCount : 전체 Database Count (기본적으로 1)

Return	• void
---------------	--------

(4) security Callback (BIZDBSCallback.java)

JDBC 드라이버가 DB 연결 시도할 때, 사용할 DB 사용자에게 대한 비밀번호를 가공하기 위해 사용하기 위한 Interface이다. 기본적으로 db.cf의 db0.password의 값으로 DB 연결을 시도하나 이 Interface를 사용하면 고객사의 필요에 따라 DB 패스워드에 대한 보안 관리를 수행할 수 있다.

- **String getDBPassword (**
int dbIndex,
String driver,
String url,
String dbUserId
)

설명	• DB 패스워드를 리턴하는 콜백 함수
-----------	-----------------------

Parameter	<ul style="list-style-type: none"> • dbIndex : 연결할 Database Index (기본적으로 0) • driver : DB Driver class 명 • url : JDBC URL 주소 • dbUserId : DB 접근 사용자 ID
------------------	--

Return	<ul style="list-style-type: none"> • String : ATA의 특정 dbIndex에 대한 DB 패스워드를 리턴. <p style="text-align: center;">Null을 리턴할 경우, db.cf의 db[index].password를 사용함.</p>
---------------	--

2) User Library Compile 및 동작방법

각 서비스 별 User Library 컴파일 방법 및 동작 설정 방법은 유사하다.

(1) User Library Compile 방법

예제로 제공되는 \$ATA_HOME/src/biz/ata/callback/SampleAlimCallback.java을 클래스 이름 및 모듈을 고객의 환경에 맞게 수정한 후 javac 명령어를 이용하여 컴파일 한다. 아래 예제는 메시지 전송 후 결과 수신 시 특정 테이블을 업데이트 하는 로직을 수행하는 User Callback 이다.

참고

나머지 User Library 적용 방법은 동일하며, Method 파라미터 및 처리 방법은 실제 예제를 참조 하기 바란다.

샘플 Source 수정

```
public void doAfterDequeue(int dbIndex, String key, int seq,
    String recipientNum, String code, int carrier, long date) {

    Connection conn = null;
    PreparedStatement pstmt = null;
    String strQuery = null;
    strQuery = "insert into user_table"
        + "( mt_pr, msg_status, result_code )"
        + "values( ?, '3', ? );";

    try {
        DBConnector dbConnector = new DBConnector();
        conn = dbConnector.getConnection(0);

        pstmt = conn.prepareStatement(strQuery);
        pstmt.setString(1, key);
        pstmt.setInt(2, carrier);
        int ret = pstmt.executeUpdate();
        logger.info("USER_TABLE INSERT: {}/{}/{}", new Object[]{ret, key, carrier });

    } catch (SQLException e) {
        logger.error("SQLException occurred when inserting user_table", e);
    }
}
```

```

    } catch (Exception e) {
        logger.error("Exception occurred when inserting user_table", e);

    } finally {
        try { pstmt.close(); } catch (Exception e) { }
        try { conn.close(); } catch (Exception e) { }
    }
}

```

컴파일

▶ java version 확인

```

biztalk@biztalk:~/Biztalk/ATA$ java -version
java version "1.6.0_12"
Java(TM) SE Runtime Environment (build 1.6.0_12-b04)
Java HotSpot(TM) Server VM (build 11.2-b01, mixed mode)

```

▶ Compile

```

biztalk@biztalk:~/Biztalk/ATA$ cd src
biztalk@biztalk:~/Biztalk/ATA/src$ javac -classpath ../lib/ata-1.0.1.jar../lib/slf4j-api-1.5.6.jar
biz/ata/callback/SampleAlimCallback.java

```

(2) User Library 설정

6.2.1에서 생성된 클래스를 \$ATA_HOME/classes/biz/ata/callback 밑에 복사 후 ata.cf 를 설정한 후 ATA 을 재 기동 하여 테스트 한다.

적용 예

▶ 컴파일 된 클래스 복사

```

biztalk@biztalk:~/Biztalk/ATA$ cp SampleAlimCallback.class $ATA_HOME/classes/biz/ata/callback/

```

▶ ata.cf 설정

```

callback.use.alim = 1
callback.class.alim = biz.ata.callback.SampleAlimCallback

```

▶ ata restart

```

biztalk@biztalk:~/Biztalk/ATA$ ./atasvc stop; ./atasvc start

```
