

모바일메시지서비스

【 기온실치매뉴얼 】

2022.06



모바일메시지서비스

개정이력

버전	개정일자	개정사유	개정내역	작성자	승인자
1.0	2021.08.20	제정	최초 제정	이준희	박재아
1.1	2021.09.10	수정	프로세스 실행중인지 확인 하는 방법 추가	이준희	
1.2	2022.01.06	수정	setting.yaml 내용 최신화	이준희	
1.3	2022.04.29	수정	setting.yaml 내용 최신화	이준희	
1.4	2022.06.21	수정	setting.yaml 내용 최신화	이준희	

For NT, Unix, LINUX with JDBC

설치매뉴얼

version 1.4

Copyright © 2021, NanalSMI Inc. All rights reserved.

이 프로그램은 핵, 항공, 대량 수송 수단, 의료 또는 기타 본질적으로 위험한 상황에서 사용하도록 개발된 것이 아닙니다. 따라서 그런 목적으로 사용된 경우, 사용자는 응용 프로그램의 안전한 사용을 보장하기 위한 모든 적절한 안전 조치, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. 프로그램이 이러한 목적으로 사용되었을 경우 모바일메시지서비스는 이러한 프로그램 사용으로 인한 피해를 책임지지 않습니다.

이 프로그램(소프트웨어와 설명서 포함)은 저작권법, 특허 및 기타 지적재산권 관련 법규에 의해 보호됩니다. 이 프로그램을 리버스 엔지니어링하거나 분해하거나 또는 역 컴파일 하는 것은 금지되어 있습니다. 또한 이 프로그램에서 사용되는 프로토콜을 리버스 엔지니어링으로 분석하는 것도 금지되어 있습니다.

이 문서의 내용은 사전 공지 없이 변경될 수 있습니다. 이 문서의 어떤 부분도 어떠한 목적이든 모바일메시지서비스의 허가 없이 기계적, 전자적 또는 기타 임의의 수단을 통해 배포되거나 복사할 수 없습니다.

1. 개요

■ GAON설치가이드는 RDBMS에 관한 기본적 지식을 가진 사람은 누구나 설치 가능하도록 작성되었습니다. 문서내용을 순서대로 확인하면서 GAON를 설치하시기를 당부드립니다.

■ RDBMS란?

행과 열로 된 2차원의 표로 데이터를 표현하는 데이터베이스관리시스템. 대규모 컴퓨터 시스템을 대상으로 많은 이용자가 대량의 데이터를 다룰 때 데이터베이스를 관리하는 시스템으로서, PC용데이터베이스에 비해 대량의 데이터를 취급할 수 있고 신뢰성 측면에서도 뛰어난 업무시스템을 구축할 수 있다. 구조화 조회 언어(SQL)로 작성된 데이터 조작용 명령에 따라 서버상의 데이터를 조작하며, 수정 등의 명령을 서버로 보낼 수 있는 것이면 어떤 클라이언트에서도 서버의 데이터를 조작할 수 있다.

■ Gaon 모듈이란?

Gaon은 모바일메시지서비스의 앱 기반 메시지 게이트웨이 서비스를 사용하기 위한 이용기관이 이용기관의 시스템과 쉽게 연동할 수 있도록 도와주는 앱 기반 메시지 게이트웨이 도우미로서, 이용기관의 시스템에 설치된다.

■ 설치 가능 OS는?

Gaon 모듈은 JDK 6.0 이상 운영 가능한 시스템에서 설치하실 수 있으며 JDK는 ORACLE사 JDK 시스템을 이용할 수 있습니다.

■ 설치가능 RDBMS는?

지원 RDBMS	지원버전	비고
MYSQL	5.1 이상	
MSSQL	2014 이상	
ORACLE	11g 이상	
TIBERO	8.2.0 이상	
CUBRID	6 이상	
POSTGRESQL	9.6 이상	

1. 개요

■ 설치가능 RDBMS는?

지원 RDBMS	지원버전	비고
ALTIBASE	6.1.1 이상	

1. 개요

■ 참고

1. RDBMS 서버의 위치는 리모트인 경우 RDBMS가 설치된 운영체제(OS)와는 상관이 없습니다.
2. Gaon 설치서버에 각 RDBMS 클라이언트 설치 및 설정이 필요합니다. (JDBC 포함)
3. RDBMS가 Gaon과 같은 시스템에 설치, 운영이 된다면 별도의 클라이언트 프로그램이 필요하지 않을 수도 있습니다. 단, 서버 설치 시 JDBC드라이브 부분이 설치가 되어 있어야 합니다.

■ 설치 전 확인사항

1. DB 설정 및 RDBMS Client 설치 확인
Gaon은 이용기관의 DB에 접속하여 전송, 결과 및 MMS 테이블을 생성합니다.
Gaon을 설치하기 전 Database, DB User를 우선 생성하셔야 합니다. (각 DBMS 매뉴얼 참조)

2. 설치

■ 준비물

1. Gaon 모듈
2. DATABASE

■ 설치 과정

1. 다운 받은 Gaon모듈을 확인 후, 적당한 위치에 압축을 해제합니다.
Gaon모듈 기동 후에 필요한 폴더(Log 폴더 등)는 자동으로 생성됩니다.
설치된 파일들의 사이즈는 Gaon 버전에 따라 다를 수 있습니다.
2. JDBC드라이버 설치 및 DB 설정
Gaon모듈은 JDBC를 사용합니다. 자바 프로그램 내에서 데이터베이스 질의문 즉, SQL을실행하기 위한 자바 API(application programming interface)입니다. Java database connectivity의 약자로 생각하기도 하지만 실제로는 상표 이름입니다. JDBC는 데이터베이스 및 애플리케이션 개발자들을 위한 표준 API를 제공하고 순수 자바API만으로도 데이터베이스 응용 업무를 만들게 해줍니다. JDBC를 사용하면, 어떠한 관계데이터베이스(relational database)에서도 SQL문을 사용하기 쉽습니다. 즉, JDBC API를 사용하면 데이터베이스에 접근하는 프로그램을 따로 만들 필요가 없습니다. 단지 하나의 프로그램을 작성하고 그 프로그램에서 SQL문을 적당한데이터베이스에 전송할 수 있습니다.
또한 응용 업무를 자바로 작성한다면 보유한 기종이나 소프트웨어에 따라 다르게 작성하지 않아도 되기 때문에 자바와 JDBC의 결합은 하나의 프로그램이 어디에서나 동작할 수 있게 해줍니다. 자바는 사용하기에 견고하고 안전하고 쉽고 이해하기 쉬우며 데이터베이스 응용 업무를 만드는데 있어서 최적의 언어라 할 수 있습니다. 단지 필요한 것은 다양한데이터베이스에 연결하는 방법일 것이라고 할 수 있으며, JDBC는 이러한 것을 위한 도구입니다.
3. JDBC 드라이버설치
Gaon모듈에 기본적인 JDBC드라이버가 기본적으로 내장되어 있습니다. DB 접근에 문제가 있을 경우 최신 JDBC드라이버로 업그레이드 한 후 다시 실행해 주시길 바랍니다.
4. 1번에서 다운받은 Gaon모듈의 압축을 풀면 해당 폴더 내에 gaon.jar 파일과 setting.yaml.sample 파일이 존재합니다. setting.yaml.sample파일을 setting.yaml으로 이름을 변경 혹은 복사한 후, 사용자 매뉴얼을 참고하여 이용기관의 시스템에 맞게 config 값을 수정합니다.

2. 설치

```
#####
# MobileMessageService App GateWay Config File
#####
# Host Info(접속정보)
# 입력 예)를 참조 (공백 '_' 포함불가)
#
#=====
#gwlp: GateWay IP Address (목적지 IP : 모바일메시지서비스 G/W 서버 IP)
#gwPort: 30000 운영1, 40000 운영2, 50000 개발
#gwClientId: 고객사 고유ID (시스템,발송 계정)
#gwClientPwd: 고객사 비밀번호(시스템,발송 계정 패스워드)
#=====
# !!!! 현재는 개발서버가 없습니다. * 실제 발송 됩니다. !!!!
# 현재는 IP 한개에 포트를 여러개 설정해서 사용해야 합니다.
gwlp: [10.000.000.170]
gwPort: [30000, 40000]
gwClientId: 계정
# 패스워드에 특수문자가 들어가는 경우 시작과 끝에 쌍따옴표(")를 붙여 주시길 바랍니다.
# 예) "fqw!@3"
gwClientPwd: 패스워드

#####
#[DB설정1] 연결 Connect
#####
# * 중요 : 'dbName' 은 수정금지 항목 입니다.
# 가온모듈에서 어떤 DBMS를 사용하는지를 구분하는 값입니다.
# ORACLE, MSSQL, MYSQL, TIBERO, CUBRID
#=====
# ORACLE
#=====
#dbName: ORACLE
#dbDriverName: oracle.jdbc.driver.OracleDriver
#dbUrl: jdbc:oracle:thin:@192.168.20.77:1521:xe
#dbUser: gaon
#dbPassword: pwd
```


2. 설치

```
#=====
# MSSQL (Microsoft SQL Server 6.5, 7, 2000, 2005, 2008 and 2012)까지 지원
#=====
#dbName: MSSQL
#dbDriverName: net.sourceforge.jtds.jdbc.Driver
##dbDriverName: com.microsoft.sqlserver.jdbc.SQLServerDriver
#dbUrl: jdbc:sqlserver://localhost:1433;database=데이터베이스명
#dbUser: gaon
#dbPassword: pwd
#=====
# MYSQL & mariaDB
#=====
#dbName: MYSQL
#dbDriverName: com.mysql.jdbc.Driver
#dbUrl: jdbc:mysql://서버주소:3306/데이터베이스명?useSSL=false
#dbUser: gaon
#dbPassword: pwd
#=====
# TIBERO
#=====
#dbName: TIBERO
#dbDriverName: com.tmax.tibero.jdbc.TbDriver
#dbUrl: jdbc:tibero:thin:@192.168.20.77:8629:tibero
#dbUser: gaon
#dbPassword: pwd
#=====
# CUBRID
#=====
#dbName: CUBRID
#dbDriverName: cubrid.jdbc.driver.CUBRIDDriver
#dbUrl: jdbc:cubrid:DB서버주소:33000:데이터베이스명::
#dbUser: gaon
#dbPassword: pwd
#=====
#dbName: POSTGRESQL
#=====
#dbDriverName: org.postgresql.Driver
#dbUrl: jdbc:postgresql://아이피:포트/데이터베이스명
#dbUser: gaon
#dbPassword: pwd
#=====
```

2. 설치

```
#=====
# dbName: MYSQL 은 수정금지 항목 입니다.
dbName: MYSQL
#=====
dbDriverName: com.mysql.jdbc.Driver
dbUrl: jdbc:mysql://localhost:3306/board?useSSL=false
dbUser: user
dbPassword: passwd
#=====

#####
#[DB설정2] 데이터베이스 CharSet(db_charset)
#####
# ORACLE, MSSQL, MYSQL, SYBASE, DB2, INFOMIX, CUBRID, TIBERO, ALTIBASE 공
통사용
# DBMS에 설정된 CharSet으로 설정한다.
# DBMS마다 euckr, euc_kr, eucksc 으로 명칭이 틀릴경우 EUC-KR로 사용.
# EUC-KR UTF-8, UTF8, euckr, ISO-8859-1(ISO 8859)
# !!사용하지 않음!!
#dbCharset: EUC-KR
#####

#####
#[DB설정3] 테이블명
#####
# 사용하는 테이블 명을 변경하는 옵션
# 기존에 테이블 명을 사용자가 임의의 테이블 명으로 변경할 수 있다.
# 테이블명 대,소문자 구분 DBMS 에 맞게 설정
# 아래 예와 같이 설정을 하면
# 기본값 : 테이블 설정 (DBMS 마다 대소문자 식별할 경우도 있음)
# ORACLE, TIBERO의 경우 대문자만
# POSTGRESQL의 경우 소문자만
#####
# 알림톡 기본테이블 명
#####
appSendContentsTableName: app_send_contents
appSendDataTableName: app_send_data
appSendDataLogTableName: app_send_data_log
templateCodeTableName: app_template_code
```

2. 설치

```
#####
# 누리테이블 명, 알림톡 실패시 입력할 테이블명
#####
msgDataTableName: nuri_msg_data
msgContentsInfoTableName: nuri_mms_contents_info
#####

#####
# 사용자 정의 컬럼, 인덱스를 추가할수 있는 최대 갯수 (최대값 10)
#####
maxCustomColumnIndexCount: 10

#####
# 사용자 정의 컬럼
#####
#
# 사용할 테이블만 주석을 제거한 후 사용
# {COLUMN NAME} {COLUMN TYPE} 형식으로 작성
# 주석부분 제거 후 사용
#####
# APP_SEND_DATA테이블에 추가할 컬럼
#customColumnForAppSendData:
# - COLUMN1 VARCHAR(100)
# APP_SEND_DATA
customColumnForAppSendData:
- MSG_IDATE datetime

#####
# 사용자 정의 인덱스
# indexName : 인덱스 이름
# columns: 인덱스에 추가할 컬럼 (컬럼이 여러개인경우 콤마(,)로 구분지어서 넣음)
# 주석된 부분 제거후 사용
#####
# APP_SEND_DATA 테이블에 추가할 인덱스
#customIndex:
# - indexName: TEST_INDEX
# columns: SUB_ID DESC, CALL_BACK ASC
# - indexName: test2
# columns: test2 desc, test2 asc
```

2. 설치

```
#####
#[발송기능설정] MSG_DATA 시퀀스, MMS_CONTENTS_INFO
#####
# 알림톡 실패시 SMS/LMS 로 처리하기 위해 사용 필수 설정.
# - 알림톡 메시지 전송 실패시 SMS/LMS 로 재발송 처리
# - 자동 재발송처리를 위한 설정 입니다.
#####
# DEFAULT인 경우 SMS/LMS 발송시 APP_SEND_CONTENTS의 PACK_UNIQUEKEY컬럼
의 값을 NURI_MMS_CONTENTS_INFO의 CONT_SEQ컬럼에 복사해서 삽입하고,
# APP_SEND_DATA의 NURI_MSG_SEQ컬럼의 값을 NURI_MSG_DATA의 MSG_SEQ컬럼
에 복사해서 삽입한다.
#
# 경고) msgContentsInfoSeq가 DEFAULT인 경우 APP_SEND_CONTENTS의 MSG_DATA
컬럼에 변수를 사용할 수 없습니다.
#
# AUTO인 경우 NURI_MSG_DATA의 MSG_SEQ와 NURI_MMS_CONTENTS_INFO의
CONT_SEQ가 AUTO_INCREMENT설정이 되있는 상태로 그냥 삽입함.
# NURI_MSG_DATA, NURI_MMS_CONTENTS_INFO를 위한 시퀀스를 사용하는 경우, 해당
시퀀스 이름을 입력하면
# 해당 시퀀스를 사용해서 삽입한다.
#####
# [설정방법 1]. 기본(누리 시퀀스 계속사용기관)
# =====
# msgDataSeq: DEFAULT
# msgContentsInfoSeq: DEFAULT
# -----
# APP_SEND_CONTENTS.PACK_UNIQUEKEY ▶ (복사, PACK_UNIQUEKEY)
NURI_MMS_CONTENTS_INFO.CONT_SEQ
# APP_SEND_DATA.MSG_SEQ ▶ (복사, NURI_MSG_SEQ )
NURI_MSG_DATA.MSG_SEQ
# =====
# [설정방법 2]. 시퀀스별도 사용, APP 실패시 누리테이블에는 신규 시퀀스로 입력
#
# =====
# =====
# msgDataSeq: NURI_MSG_DATA_SEQ
# msgContentsInfoSeq: NURI_MMS_CONTENTS_INFO_SEQ
# -----
```

2. 설치

```
# APP_SEND_CONTENTS.PACK_UNIQUEKEY ▶(신규입력,
NURI_MMS_CONTENTS_INFO_SEQ.NEXTVAL)
NURI_MMS_CONTENTS_INFO.CONT_SEQ
# APP_SEND_DATA.MSG_SEQ          ▶(신규입력, NURI_MSG_DATA_SEQ.NEXTVAL)
NURI_MSG_DATA.MSG_SEQ
# =====
# [설정방법 3]. AUTO (시퀀스 미사용기관) * Auto_Increment 는 권장하지 않음.
# 시퀀스를 사용하지 않으면 MSG_SEQ를 중복 발생.
# =====
# msgDataSeq: AUTO
# msgContentsInfoSeq: AUTO
# -----
# APP_SEND_CONTENTS.PACK_UNIQUEKEY ▶ (신규입력, Auto_Increment)
NURI_MMS_CONTENTS_INFO.CONT_SEQ
# APP_SEND_DATA.MSG_SEQ(NULL)    ▶ (신규입력, Auto_Increment)
NURI_MSG_DATA.MSG_SEQ
# =====
# MSG_DATA 시퀀스
# =====
msgDataSeq: DEFAULT
#msgDataSeq: AUTO
#msgDataSeq: NURI_MSG_DATA_SEQ
# =====
# NURI_MMS_CONTENTS_INFO 시퀀스
msgContentsInfoSeq: DEFAULT
#msgContentsInfoSeq: AUTO
#msgContentsInfoSeq: NURI_MMS_CONTENTS_INFO_SEQ
#####
#알림톡 실패시 NURI이용 메시지 발송 여부 (yes|no)
#####
useLms: yes

# NURI이용 메시지 발송시 사용자 정의 컬럼데이터도 같이 복사할 것인지 여부
# NURI의 NURI_MSG_DATA테이블에 같은 사용자 정의 컬럼이 만들어져 있어야함
# 모듈 실행시 msgDataTableName에 해당하는 테이블에 사용자 정의 컬럼이 존재 하지 않으면
# 자동으로 생성시도
copyCustomColumnWhenSendLms: ye
#####
```

2. 설치

```
#####
# 테이블 로그 기록
#####
# yes인 경우 서버로 부터 결과를 받을 때마다 로그테이블로 이관시킬 수 있는 데이터가 있는지 확
인 후 로그 테이블로 이동
# no인 경우 5분에 한번씩 로그테이블로 이관시킬 수 있는 데이터가 있는지 확인 후 로그 테이블
로 이동
logRealTime: yes
# LOG 테이블 관리
# LOG_TABLE_YYYYMM 형식으로 여러개 쪼갤 것인지
# 하나의 LOG_TABLE로 만들것인가
# ONE, DEFAULT(YYYYMM)
#makeLogMode: DEFAULT
makeLogMode: ONE

# yes인 경우
# 로그 테이블을 생성
# APP_SEND_DATA 테이블에서 발송 완료 되고 결과까지 다 받은 데이터를 로그테이블로 이관
# no인 경우
# 로그 테이블 생성하지 않음
# APP_SEND_DATA에 계속 데이터 누적
isTransferLog: yes

#####
# 서버에 접속 실패시 재접속시도 관련
#####
# 몇분 간격으로 재접속 시도를 할것인가 (1~)
retryTime: 1
# 재접속 시도를 몇번 할 것인가
# 재접속 시도를 초과하면 클라이언트 종료됨
# 0으로 설정시 무한히 재접속시도
retryCount: 10

#####
# 발송 제한 시간 설정 (해당 시간동안 메시지를 발송하지 않음)
# 24시간제로 설정
# HH:mm:ss 또는 HHmmss
#####
noSendTimeStart: 09:00:00
noSendTimeEnd: 00:00:00
```

2. 설치

```
# retryTime 간격으로 retryCount만큼 재접속 시도를 해도 실패한 경우
# 문자로 보낼 것인지 여부
# yes인 경우 문자발송 후 다시 재접속 시도
# no인 경우 모듈 종료
# retryCount를 0으로 설정해두면 해당 옵션이 yes여도 문자 발송 안함
sendLmsWhenRetryEnd: no
# 서버에 접속 실패시 재시도를 하지 않고 바로 sms/lms로 보낼것인지 (yes | no)
# yes로 설정하면 위의 retryTime과 retryCount 설정 무시됨
sendLmsWhenConnectionFail: yes

# 서버가 살아 있는지 확인하기 위해 접속 시도 할때의 타임아웃
socketConnectionTimeout: 5

#####
# 로그파일 저장 기간
# 일정기간 지난 로그파일을 삭제함 (1 ~ )
# 단위: 일
#####
logSaveTerm: 999

#####
# Log파일을 저장할 위치
# default : ./GaonLog/
#####
logFilePath: ./gaon_log

#####
# DB에서 메시지를 읽어오는 주기 (단위: 초)
# 최소값 1초
#####
tranTerm: 1

#####
# 메시지를 전송 할때 몇시간 전 메시지까지만 보낼 것인가 (단위: 시간)
#####
beforeTime: 3
#####
```


2. 설치

```
# 하루에 한 번호에 보낼 수 있는 최대 건수
# 0은 무제한
#####
duplicatePhoneCount: 0

#####
# 타임아웃 제한 시간
# 해당 시간동안 서버로부터 응답이 없으면 타임아웃(단위: 분)
# 최소값 10분
#####
timeoutLimit: 10

#####
# 동보전송에 보낼 수 있는 최대 연락처 갯수
#####
# 동보 전송에 최대 포함될 수 있는 연락처 갯수
# 범위 10 ~ 1000(MAX)
groupMessageCount: 150

#####
# 서버에 현재 메시지가 대량으로 쌓여있어 발송이 늦어지는 경우
# 서버에서 앱메시지를 보낸 시간과 모듈에서 서버로 발송한 시간 차이가
# 설정한 값만큼 벌어지는 경우
# 이전에 보낸 메시지가 전부 보내질때까지 서버로 발송을 일시정지를 하게됨
# 단위 : 초
# 최소값 : 40
# 2.0.0.000001이상 부터 사용하지 않음
#####
#flowControlTime: 90

##### log 기록 범위 설정
# FATAL : 치명적에러
# ERROR : 수행가능정도의에러
# WARN : 시스템에 문제를 일으킬가능성이 있을때 사용
# INFO : 정보용으로 사용
# DEBUG : 원인분석용 (임시로만사용, 로그양이 많음)
# ex)loggerLevel: FATAL,ERROR,WARN,INFO,DEBUG
# 전송클라이언트로그, 송신클라이언트로그, 정보로그
# DEBUG > INFO > WARN, ERROR, FATAL
loggerLevel: INFO
```


2. 설치

■ 설치 과정

- 윈도우의 cmd창이나 리눅스의 터미널에서 `java -jar gaon.jar`을 입력하면 아래와 같은 화면이 나타납니다. (백그라운드 실행을 권장합니다.)

5.1 백그라운드 실행 방법

- Window의 경우 Cmd혹은 powershell에서

`javaw -jar gaon.jar [setting.yaml파일 경로, 생략시 ./setting.yaml]`

- 리눅스의 경우 터미널에서

`nohup java -jar gaon.jar [setting.yaml파일 경로, 생략시 ./setting.yaml] > dev/null &`

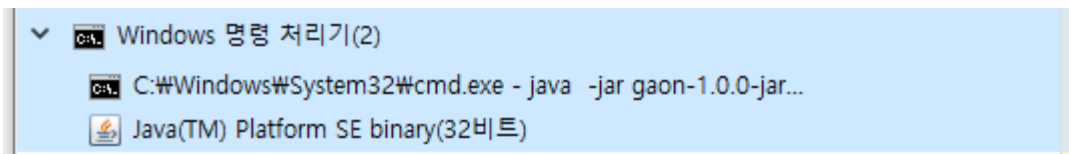
5.2 Window에서 백그라운드로 실행했을 경우 PID 확인하는 방법

cmd: `wmic process where "name like '%gaon.jar%' " get processid,commandline`

powershell: `Get-WmiObject Win32_Process -filter "CommandLine LIKE '%gaon.jar%'"`

```
C:\Users\nana1\2021PC01\Desktop\gaon\target>java -jar gaon.jar
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:298) -버전 : 1.0.0.000001
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:299) -로깅 레벨 : INFO
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:300) -게이트웨이 IP : 
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:301) -게이트웨이 PORT : 
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:303) -게이트웨이 CLIENT ID : 
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:304) -알림록 실패시 LMS사용여부 : true
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:305) -DB_DRIVER_NAME : com.microsoft.sqlserver.jdbc.SQL ServerDriver
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:306) -DB_URL : 
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:308) -DB_USER_NAME : **
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:309) -누리 CONTENTS_INFO 테이블 이름 : NURI_MMS_CONTENTS_INFO
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:310) -누리 CONTENTS_INFO 시퀀스 : AUTO
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:311) -누리 MSG_DATA 테이블 이름 : NURI_MSG_DATA
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:312) -누리 MSG_DATA 시퀀스 : AUTO
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:313) -가온 SEND_CONTENTS 테이블 이름 : APP_SEND_CONTENTS
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:314) -가온 SEND_DATA 테이블 이름 : APP_SEND_DATA
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:315) -가온 ERROR_CODE 테이블 이름 : APP_ERROR_CODE
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:316) -앱메시지 실패시 SMS/LMS 발송 여부 : true
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:317) -SMS만 사용 여부 : false
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:318) -REALTIME 여부 : false
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:319) -LOG_INTERVAL : 30
[INFO] com.nana1.setting.NuriSetting.getSetting(NuriSetting.java:320) -로그테이블로 옮길 시간 : 0시 0분
[INFO] com.nana1.Main.main(Main.java:43) -뉴 누리 모듈 시작
[INFO] com.nana1.Main.main(Main.java:47) -----필요한 테이블 생성 시작-----
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:26) -APP_ERROR_CODE 테이블 생성
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:28) -APP_ERROR_CODE 테이블 생성 완료
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:35) -APP_ERROR_CODE 테이블 생성
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:37) -APP_ERROR_CODE 테이블 생성 완료
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:57) -APP_SEND_CONTENTS 테이블 생성
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:61) -APP_SEND_CONTENTS 테이블 생성 완료
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:68) -APP_SEND_DATA 테이블 생성
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:73) -APP_SEND_DATA 테이블 생성 완료
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:82) -TEMPLATE_CODE 테이블 생성
[INFO] com.nana1.db.TableCreator.Creator.createTableForNewNuri(Creator.java:86) -TEMPLATE_CODE 테이블 생성 완료
[INFO] com.nana1.Main.main(Main.java:50) -----필요한 테이블 생성 완료-----
[INFO] com.nana1.thread.sender.GaonSender.run(GaonSender.java:80) -가온 SENDER 시작
```

- 윈도우의 경우 작업관리자를 열어 프로세스가 실행 중인지 확인할 수 있습니다.



- 리눅스의 경우 `ps -ef | grep gaon` 명령어를 통해 프로세스가 실행 중인지 확인할 수 있습니다.

```
[/home/nana1/gaon] ps -ef | grep gaon
nana1 15046 14037 99 15:01 pts/2 00:00:06 java -jar gaon.jar
```

2. 설치

■ 설치 과정

8. Gaon Log파일 확인
gaon_log폴더에는 Gaon모듈이 실행중에 발생한 로그 기록이 있습니다.
BIZM Gateway접속 오류 및 DB오류가 연속적으로 발생하는 경우 log파일을 확인하여 원인을 파악할 수 있습니다.
 - clientid_system.log 파일에는 시스템의 전반적인 내용이 기록되는 파일입니다.
 - clientid_sender.log 파일에는 sender와 관련된 로그만 따로 기록하는 파일입니다.
 - clientid_receiver.log 파일에는 서버로부터 수신 받은 결과 데이터를 기록하는 파일입니다.
 - clientid_lmssend.log 파일에는 앱메시지 전송에 실패하여 문자로 전송된 데이터를 기록하는 파일입니다.
9. DB Table생성 확인
Gaon 모듈을 실행하면 APP_SEND_CONTENTS, APP_SEND_DATA, TEMPLATE_CODE 테이블이 생성됩니다.
10. 테이블이 정상적으로 생성되었다면 사용자 설명서를 참고하여 전송 테스트를 진행하시면 됩니다.



모바일메시지서비스 이용문의

- 모바일메시지서비스 웹사이트 : <http://www.mgov.go.kr>
- 국가정보자원관리원 운영총괄과 모바일메시지서비스 담당 : 042-250-5333
- 국가정보자원관리원 모바일메시지서비스 운영·기술지원 : 042-250-5537