



# 포팅메뉴얼

## FrontEnd

```
flutter 3.3.7
Dart 2.18.4
DevTools 2.15.0

environment:
  sdk: '>=2.18.2 <3.0.0'

dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.2
  shared_preferences: ^2.0.15
  kakao_flutter_sdk_user: ^1.2.2
  flutter_screenutil: ^5.6.0
  fluttertoast: ^8.1.1
  geolocator: ^9.0.2
  flutter_native_splash: ^2.2.14
  pull_to_refresh: ^2.0.0
  faker: ^2.0.0
  flutter_secure_storage: ^6.0.0
  jwt_decode: ^0.3.1
  http: ^0.13.5
  protobuf: ^2.1.0
  flutter_typeahead: ^4.1.1
  intl: ^0.17.0
  sqflite: ^2.2.0+3
  just_audio: ^0.9.30
  carousel_slider: ^4.1.1

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_lints: ^2.0.0
  flutter_launcher_icons: ^0.11.0
```

# BackEnd

```
environment :
  spring 2.7.5
  Java 1.8
  Kafka 3.1.2
  MongoDB 4.4.17
  Redis 5.0.7

command :
[Redis]
$ redis-cli -h 52.79.215.19 -p 6379 -a password

[Kafka]
$ kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic chats --from-beginning

[Mongo]
$ mongo -u o2a4 -p password
$ show databases

port :
api-gateway 80:8000
eureka-server 8761

chat-server 8101/8201(http) 8102/8202(tcp)
service-server 8082
auth-server 8081
redis 6379
kafka 8892:9092
mongo 27017

dependencies :
[api-gateway]
{
  implementation 'org.springframework.cloud:spring-cloud-starter-gateway'
  implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'
  implementation 'io.jsonwebtoken:jjwt-api:0.11.5'
  implementation 'io.jsonwebtoken:jjwt-impl:0.11.5'
  implementation 'io.jsonwebtoken:jjwt-jackson:0.11.5'
  compileOnly 'org.projectlombok:lombok'
  annotationProcessor 'org.projectlombok:lombok'
  testImplementation 'org.springframework.boot:spring-boot-starter-test'
  implementation 'org.springframework.boot:spring-boot-starter-security'
}

[chat-tcp]
{
  implementation 'org.springframework.boot:spring-boot-starter-data-mongodb-reactive'
  implementation 'org.springframework.boot:spring-boot-starter-data-mongodb'
  implementation 'org.springframework.boot:spring-boot-starter-data-redis-reactive'
  implementation 'org.springframework.boot:spring-boot-starter-webflux'
  implementation 'com.google.protobuf:protobuf-java:3.21.8'
  implementation 'org.apache.commons:commons-collections4:4.4'
  implementation 'io.projectreactor.kafka:reactor-kafka:1.3.13'
```

```

        implementation 'org.apache.kafka:kafka-streams'
        implementation 'org.springframework.kafka:spring-kafka'
        implementation 'org.springframework.kafka:spring-kafka-dist:2.9.0'
        implementation 'com.fasterxml.jackson.core:jackson-databind:2.14.0'
        implementation 'com.googlecode.json-simple:json-simple:1.1.1'
        implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'
        implementation 'com.auth0:java-jwt:4.2.1'

        compileOnly 'org.projectlombok:lombok'
        developmentOnly 'org.springframework.boot:spring-boot-devtools'
        annotationProcessor 'org.projectlombok:lombok'
        testImplementation 'org.springframework.boot:spring-boot-starter-test'
        testImplementation 'io.projectreactor:reactor-test'
    }

    [eureka-setting]
    {
        implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-server'
        testImplementation 'org.springframework.boot:spring-boot-starter-test'
    }

    [module-auth]
    {
        implementation 'org.springframework.boot:spring-boot-starter-security'
        implementation 'org.springframework.boot:spring-boot-starter-web'
        implementation 'org.springframework.boot:spring-boot-starter-data-mongodb'
        implementation 'org.apache.httpcomponents:httpclient'
        implementation 'com.fasterxml.jackson.core:jackson-databind:2.13.4.2'
        implementation 'io.jsonwebtoken:jjwt-api:0.11.5'
        implementation 'io.jsonwebtoken:jjwt-impl:0.11.5'
        implementation 'io.jsonwebtoken:jjwt-jackson:0.11.5'
        implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'
        compileOnly 'org.projectlombok:lombok'
        annotationProcessor 'org.projectlombok:lombok'
        developmentOnly 'org.springframework.boot:spring-boot-devtools'
    }

    [module-service]
    {
        implementation 'org.springframework.boot:spring-boot-starter-data-mongodb'
        implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
        implementation 'org.springframework.boot:spring-boot-starter-web'
        implementation 'org.apache.httpcomponents:httpclient'
        implementation 'com.fasterxml.jackson.core:jackson-databind:2.13.4.2'
        implementation 'org.springframework.boot:spring-boot-starter-data-redis:2.7.5'
        implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'
        compileOnly 'org.projectlombok:lombok'
        annotationProcessor 'org.projectlombok:lombok'
        developmentOnly 'org.springframework.boot:spring-boot-devtools'
    }
}

```

## Redis

## 1. 저장소 업데이트 및 업그레이드로 최신 패키지 업데이트

```
sudo apt-get update  
sudo apt-get upgrade
```

## 2. Redis server 설치

```
sudo apt-get install redis-server
```

## 3. 설치된 redis version 확인

```
redis-server --version
```

```
ubuntu@ip-172-26-10-110:~$ redis-server --version  
Redis server v=5.0.7 sha=00000000:0 malloc=jemalloc-5.2.1 bits=64 build=66bd629f924ac924
```

## 4. Redis 메모리 할당을 위해 인스턴스 서버 메모리 확인

```
vmstat -s
```

```
ubuntu@ip-172-26-10-110:~$ vmstat -s  
16396052 K total memory  
4181152 K used memory  
6123592 K active memory  
4085792 K inactive memory  
5587156 K free memory  
67716 K buffer memory  
6560028 K swap cache  
0 K total swap  
0 K used swap  
0 K free swap  
201204 non-nice user cpu ticks  
144243 nice user cpu ticks  
69166 system cpu ticks  
282574911 idle cpu ticks  
8308 IO-wait cpu ticks  
0 IRQ cpu ticks  
1149 softirq cpu ticks  
46720 stolen cpu ticks  
804235 pages paged in  
52406652 pages paged out  
0 pages swapped in  
0 pages swapped out  
128902770 interrupts  
193675619 CPU context switches  
1666571708 boot time  
167290 forks
```

16기가이기 때문에 절반은 서버용으로 두고 8기기로 설정

## 5. 설정을 위한 conf 파일 접근

```
sudo vim /etc/redis/redis.conf
```

### a. maxmemory 8기가 ⇒ 556번째 줄 8gb 지정

```
#  
# In short ... if you have replicas attached it is suggested that you set a lower  
# limit for maxmemory so that there is some free RAM on the system for replica  
# output buffers (but this is not needed if the policy is 'noeviction').  
#  
maxmemory 8gb
```

### b. 597번째 줄 ⇒ 최대 메모리 관리용 키 삭제 방식 LRU 알고리즘(제일 오랫동안 사용 하지 않은 데이터를 제거하는 알고리즘) 적용 [원래는 noeviction : 캐시를 지우지 않는 정책]

```
# sinter sinterstore sunion sunionstore sdiff sdiffstore zadd zincrby  
# zunionstore zinterstore hset hsetnx hmset hincrby incrby decrby  
# getset mset msetnx exec sort  
#  
# The default is:  
#  
maxmemory-policy allkeys-lr
```

### c. bind 0.0.0.0으로 설정

### d. 새로 세팅했으니 재시작하고 실행 여부 확인

```
sudo systemctl restart redis-server.service  
systemctl status redis-server.service
```

```
ubuntu@ip-172-26-10-110:~$ sudo systemctl restart redis-server.service  
ubuntu@ip-172-26-10-110:~$ systemctl status redis-server.service  
● redis-server.service - Advanced key-value store  
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: enabled)  
   Active: active (running) since Tue 2022-11-01 05:49:22 UTC; 4s ago  
     Docs: http://redis.io/documentation,  
           man:redis-server(1)  
   Process: 167347 ExecStart=/usr/bin/redis-server /etc/redis/redis.conf (code=exited, status=0/SUCCESS)  
    Main PID: 167362 (redis-server)  
      Tasks: 4 (limit: 19204)  
     Memory: 2.1M  
    CGroup: /system.slice/redis-server.service  
            └─167362 /usr/bin/redis-server 70.120.240.1:6379
```

### e. redis에 접근하는 비밀번호 설정 (redis.conf)

```
# people do not need auth (e.g. they run their own servers).
#
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
requirepass o2a4redis
```

#### f. 레디스 서버 접속권한 확인

```
netstat -nlpt
```

```
ubuntu@ip-172-26-10-110:~$ netstat -nlpt
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:9090            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6010         0.0.0.0:*               LISTEN      -
tcp6       0      0 :::9090                :::*                    LISTEN      -
tcp6       0      0 :::1:6379              :::*                    LISTEN      -
tcp6       0      0 :::22                  :::*                    LISTEN      -
tcp6       0      0 :::1:6010              :::*                    LISTEN      -
```