



# 포팅 메뉴얼

---

## Backend

### 메인 서버

build.gradle

```
plugins {  
    id 'org.springframework.boot' version '2.7.5'  
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'  
    id 'java'  
}  
  
group = 'com.a705'  
version = '1'  
sourceCompatibility = '11'  
  
configurations {  
    compileOnly {  
        extendsFrom annotationProcessor  
    }  
}  
  
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-security'  
    implementation 'org.springframework.boot:spring-boot-starter-validation'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    compileOnly 'org.projectlombok:lombok'  
    developmentOnly 'org.springframework.boot:spring-boot-devtools'  
    runtimeOnly 'com.mysql:mysql-connector-j'  
    annotationProcessor 'org.projectlombok:lombok'  
}
```

```
testImplementation 'org.springframework.boot:spring-boot-starter-test'
testImplementation 'org.springframework.security:spring-security-test'

implementation 'io.springfox:springfox-boot-starter:3.0.0'
implementation 'io.jsonwebtoken:jjwt:0.9.1'
implementation group: 'com.google.guava', name: 'guava', version: '28.1-jre'
}

tasks.named('test') {
    useJUnitPlatform()
}
```

실행(main\_server 프로젝트를 bootjar를 통해 빌드)



```
nohup java -jar 파일이름.jar &
```

- nohup : 터미널을 나가도 실행 유지
- & : 백그라운드 실행

이미 사용 중인 포트라면?



ps -ef | grep java 를 입력하면 8080 사용 중인 서버 이름이 보인다.  
kill -9 PID(프로세스 id)를 입력해 종료해준다.

## 채팅 서버

build.gradle

```
plugins {
    id 'org.springframework.boot' version '2.7.5'
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'
    id 'java'
}

group = 'com.example'
```

```

version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-mongodb'
    implementation 'org.springframework.boot:spring-boot-starter-data-redis'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    compileOnly 'org.projectlombok:lombok'
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'

    // socket
    implementation 'org.springframework.boot:spring-boot-starter-websocket'
}

tasks.named('test') {
    useJUnitPlatform()
}

```

실행(chat\_server 프로젝트를 bootjar를 통해 빌드)



```
nohup java -jar 파일이름.jar &
```

OS에 mongodb가 설치되어 있어야 함



```
$ sudo apt-get install -y mongodb-org
```

채팅 내역 삭제 방법



mongodb-clients 설치된 상태에서

mongo 입력

use hello\_gifty 로 db 이동

삭제 명령

```
db.chatMessage.deleteMany({});
```

## 이미지 서버

실행(image\_server 프로젝트로 들어가서 아래 명령어 입력)

OS에 tesseract가 설치되어 있어야 함



```
sudo apt install tesseract-ocr
```

django 서버 필요 라이브러리



pip install로 아래 설치

djangorestframework

opencv-python

pytesseract

pillow

이미 사용중인 포트라면?



ps -ef | grep py 를 입력하면 8090 사용 중인 서버 이름이 보인다.

kill -9 PID(프로세스 id)를 입력해 종료해준다.

# Frontend

## React Native 개발 환경 구성(Window, Android)

### 해당 게시물의 지시사항 따라서 개발 환경 구성

Setting up the development environment · React Native

This page will help you install and build your first React Native app. If you are new to mobile development, the easiest way to get started is with Expo Go. Expo is a set of tools and services

 <https://reactnative.dev/docs/environment-setup>



1. Node, JDK11 설치
2. 안드로이드 SDK 설치 후 SDK 플랫폼, SDK 툴 설치
3. 윈도우 시작 → 고급 시스템 설정 보기(제어판), 환경 변수 → 유저 환경 변수  
ANDROID\_HOME 환경 변수에 SDK 의 path 추가
4. 유저 환경 변수에 platform-tool path 추가

### 프로젝트 폴더 받은 후 npm 모듈 설치



npm install

### 개발용 앱을 조작할 Metro Bundler 구동



npx react-native start

## 다른 CMD에서 개발용 앱 실행 명령어 입력



```
npx react-native run-android
```

## APK 추출

### [React-Native] Android APK & AAB 파일 생성하기

며칠 전 앱을 급하게 빌드해서 release 해야했는데, 회사 CI (jenkins) server가 급먹통이 되었다.. :( 수동으로 android 앱을 빌드해서 심사

에 넣어달라고 요청이 왔다. 이번 노트는 android의 apk 파일과 aab



<https://velog.io/@cm961115/Android-apk-aab>



## 1. keystore 생성



```
keytool -genkey -v -keystore your_key_name.keystore -alias  
your_key_alias -keyalg RSA -keysize 2048 -validity 10000
```

키 저장소 비밀번호 입력:  
새 비밀번호 다시 입력:  
이름과 성을 입력하십시오. // optional  
조직 단위 이름을 입력하십시오. // optional  
조직 이름을 입력하십시오. // optional  
구/군/시 이름을 입력하십시오. // optional  
시/도 이름을 입력하십시오. // optional  
이 조직의 두 자리 국가 코드를 입력하십시오. // optional

## 2. keystore 저장

your\_key\_name.keystore 파일을 RN프로젝트의 android/app 디렉토리에 붙여넣기한다.

### 3. Gradle 설정

프로젝트의 android/gradle.properties를 편집해 다음 코드를 추가(비밀번호 입력 시 앞뒤 공백이 없도록 한다.)

```
MYAPP_UPLOAD_STORE_FILE=your_key_name.keystore
MYAPP_UPLOAD_KEY_ALIAS=your_key_alias
MYAPP_UPLOAD_STORE_PASSWORD=1234567 // 키 생성 시 설정한 비밀번호
MYAPP_UPLOAD_KEY_PASSWORD=1234567 // 키 생성 시 설정한 비밀번호
```

#### 3-2. android / app / build.gradle에 다음 코드를 추가한다.

```
signingConfigs {
    debug {
        ...
    }
    release {
        if (project.hasProperty('MYAPP_UPLOAD_STORE_FILE')) {
            storeFile file(MYAPP_UPLOAD_STORE_FILE)
            storePassword MYAPP_UPLOAD_STORE_PASSWORD
            keyAlias MYAPP_UPLOAD_KEY_ALIAS
            keyPassword MYAPP_UPLOAD_KEY_PASSWORD
        }
    }
    ....
}

....
buildTypes {
    debug {
        ...
    }
    release {
        ...
        signingConfig signingConfigs.debug
        signingConfig signingConfigs.release
        ...
    }
}
```

---

## 4. apk 파일 생성

프로젝트 루트 폴더에서 android/ 폴더로 이동 후 해당 명령어 입력하면 android\app\build\outputs\apk\release 폴더에 생성된다.

```
cd android  
./gradlew assembleRelease
```