



빌드 / 배포 가이드

해당 가이드는 로컬에서 Git 소스 클론 후 빌드 및 배포 기준으로 작성합니다

또한 해당 가이드는 아래 [버전정보](#) 에 기재된 사항들이 모두 설치되었다는 가정하에 작성합니다



버전 정보

React Native	0.68.1
Node	14.15.4
npm	6.14.10
yarn	1.22.18
Android - minSDK	26
Android - targetSDK	30
Android - compileSDK	30
iOS	11
MySQL	8.0.28
JDK	11
Gradle	7.4.1
SpringBoot	2.5.10
lombok	1.18.20
AWS EC2	ubuntu 20.04 LTS

▼ ios 기본 개발환경 설치

- homebrew 설치
- yarn 설치 `brew install yarn`
- node 설치(8.3 버전 이상, 프로젝트에서는 14.15.4 버전 사용) → `nvm` 사용하여 설치
- watchman 설치 (파일시스템의 변화를 확인하는 Facebook 도구) `brew install watchman`
- React Native command line interface (React Native CLI) 설치
- xcode, xcode CLI Tool 설정, 시뮬레이터 설치 (iOS 13이상 사용하려면 최신버전의 xcode 설치 필수)
- ios폴더에 cocoapods 설치 `sudo gem install cocoapods`
- 기타

```
$ brew tap AdoptOpenJDK/openjdk
$ brew cask install adoptopenjdk8
```



버튼을 누르면 목차를 확인할 수 있습니다.

▼ 목차

- [1. Git Clone](#)
- [2. DB](#)
- [3. Backend](#)
 - [3-1. Project import \(IntelliJ 기준\)](#)
 - [3-2. Build Gradle](#)
 - [3-3. Project Run](#)
 - [3-4. Run on Localhost](#)
- [4. Frontend](#)
 - [4-1. module install](#)
 - [4-2. 환경변수 설정](#)
 - [4-3. 실행](#)
- [5. 빌드 및 배포](#)
- [6. 앱 배포](#)
- [7. 기타 설정](#)

1. Git Clone

```
$ git clone https://lab.ssafy.com/s06-final/S06P31A205.git
```

2. DB

DB는 ec2-ubuntu의 Docker에 올려서 원격 DB를 사용했습니다
AWS ec2 서버(k6a205.p.ssafy.io:5050)가 살아있으면
아래 계정 정보로 DB에 접근할 수 있습니다

서버 주소	k6a205.p.ssafy.io
접속 포트	5050
계정	root
비밀번호	Soorim_chicken_205!

로컬 DB와 연결해 사용할 수 있습니다.

테이블은 JPA를 통해 자동 생성되기 때문에

로컬 DB에 `plomeetDB` 라는 스키마를 만들었다는 가정하에 `application.properties` 에 다음과 같이 설정을 하면 됩니다

```
# application.properties
spring.datasource.url=jdbc:mysql://{서버주소}:{접속포트}/plomeetDB?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTim
spring.datasource.hikari.username={계정명}
spring.datasource.hikari.password={비밀번호}
```

3. Backend

3-1. Project import (IntelliJ 기준)

File > New > Project from Existing Sources... > 프로젝트 폴더의 `backend` 폴더 선택

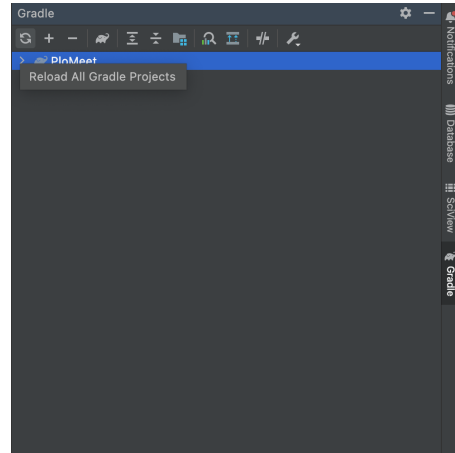
import project from external model > Gradle 선택 > Create

Project SDK, Module SDK가 버전 11인지 확인!

3-2. Build Gradle

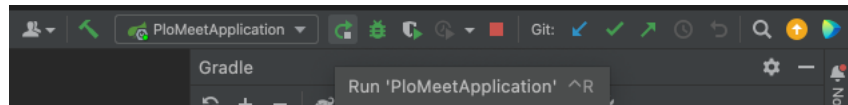
IntelliJ 창의 우측 사이드에

Gradle 탭 열기 > 탭의 좌측 상단의 **Reload All Gradle Projects** 버튼 클릭



3-3. Project Run

Run 'PloMeetApplication'



3-4. Run on Localhost

해당 프로젝트는 ec2 서버에서 배포를 하기 위해 설정 파일이 작성되었기 때문에 서버 주소가 지정되어있지 않습니다.

로컬에서 서버를 run하고 싶다면 application.properties 파일에 `server.address=localhost` 를 추가해주세요

4. Frontend

4-1. module install

```
$ npm install
$ yarn
# ios의 경우
$ cd ios && pod install && cd ..
```

4-2. 환경변수 설정

최상단 폴더에 `.env` 파일 생성

```
IDENTITYPOOLID ="ap-northeast-2:76f0b390-bd70-4a39-a1f8-5cead58a9e7f"
SERVICEKEY_WEATHER = "Yw3zPCyzMoX2VB0yMPfZgip2qIHGaFLGT5RuJ9gtVFGvzjbuNNZa5qB5DFum%2BNMe%2B0kHhUWAYIH1j0BK%2Fdj6MQ%3D%3D"
```

4-3. 실행

에뮬레이터(시뮬레이터)에 실행하는 명령어는 다음과 같습니다.

```
$ react-native run-android #안드로이드 에뮬레이터 실행
$ npx react-native run-android #안드로이드 에뮬레이터 실행
$ npx react-native start #안드로이드 에뮬레이터 실행
$ react-native run-ios #ios 시뮬레이터 실행
```

5. 빌드 및 배포

플로잇은 앱 서비스이기때문에 frontend에 해당하는 어플리케이션은 앱스토어에 배포되었고, backend만 빌드 및 배포하였습니다.

EC2-ubuntu 인스턴스에서 jenkins를 통해 git clone 후 docker 환경에서 빌드 및 배포되었습니다.

다음은 Jenkins에 환경설정 내용입니다.

- 빌드 시 Gradle script

```
clean
bootJar
```

▼ 참고 사항

프로젝트의 배포하려는 백엔드는 프로젝트폴더/backend 내부에 있기때문에

jenkins에서 Build File을 찾을 수 있도록 고급설정이 필요함 (`backend/build.gradle`)

Build File ?

Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)

- Gradle script 실행 후 shell script

```
docker stop ub # 첫 실행시 사용x
docker rm ub # 첫 실행시 사용x
docker rmi uts-backend # 첫 실행시 사용x
docker build -t uts-backend /var/lib/jenkins/workspace/plomeet_webhook/backend
docker run -d --name ub -p 8000:8000 uts-backend
```

- Dockerfile (참고용)

빌드 전에 Dockerfile을 작성해야합니다 (git clone을 받았다면 프로젝트에 이미 작성이 되어있기 때문에 추가적으로 작성할 필요는 없습니다)

```
FROM openjdk:11


ARG JAR_FILE=build/libs/*.jar

COPY ${JAR_FILE} app.jar

ENTRYPOINT ["java", "-jar", "/app.jar"]
```

6. 앱 배포

android 앱 배포는 다음 파일을 참고

 [앱 배포 방법](#)

7. 기타 설정

Android와 ios 환경에서 따로 설정 및 설치 작업이 필요한 기능들은 다음과 같습니다.

- firebase ([참고](#))
- naver map ([참고](#))
- kakao login ([참고](#))
- 릴리즈 버전 가져오기 ([참고](#))
- 위치권한 ([참고](#))
- 아이콘 셋팅 ([참고](#))