



포팅 메뉴얼

목차

목차

1. 프로젝트 기술 스택

Front-End 기술 스택

1. 개발환경
2. 사용기술
3. 라이브러리

Back-End 기술 스택

1. 개발환경
2. Database
3. 사용기술

Embedded 기술 스택

1. 개발환경
2. 사용 기기
3. 사용 기술
4. 라이브러리

2. 배포 상세

Front-End) Flutter application 배포

1. 개발 환경 Setting

Back-End) Springboot Server 배포

1. Build(Jenkins 빌드 시 사용)
 2. Dockerfile
 3. Docker container run
- 그 외 설정 파일

3. 시연 시나리오

어플리케이션 다운로드

로그인/회원가입

코스터 연결

메인 페이지

리워드 페이지

소셜 페이지

설정 페이지

망우리

1. 프로젝트 기술 스택

Front-End 기술 스택

1. 개발환경

- Flutter
- Dart

2. 사용기술

- Flutter
- Firebase
- GetX

- WebSocket
- BLE (Bluetooth Low Energy)

3. 라이브러리

- step_progress_indicator: ^1.0.2
- sliding_up_panel2: ^3.2.2
- flutter_blue_plus: ^1.4.0
- http: ^0.13.5 // api 통신
- firebase_core: ^2.10.0
- firebase_auth: ^4.4.2
- provider: ^6.0.5
- google_sign_in: ^6.1.0
- fl_chart: ^0.62.0
- get: ^4.6.5 // getx 상태관리
- flutter_secure_storage: ^8.0.0 // 내부 저장소
- intl: ^0.18.1 // 포매팅
- awesome_notifications: ^0.7.4+1
- syncfusion_flutter_charts: ^21.2.4
- image_picker: ^0.8.7+4

Back-End 기술 스택

1. 개발환경

- Java11
- IntelliJ 2022.3
- Spring Boot 2.7.11
- Gradle-Kotlin

2. Database

- MySQL
- Redis

3. 사용기술

- Spring Data JPA
- Spring Security JWT
- Spring Cache Redis
- WebSocket
- Jacoco
- Lombok
- Docker

Embedded 기술 스택

1. 개발환경

- C / C++
- Arduino 2.1.0
- 틸커캐드

2. 사용 기기

- ESP32
- FSR406(압력센서)
- NeoPixel
- 터치센서

3. 사용 기술

- BLE (Bluetooth Low Energy)
- SPIFFS
- 3D 모델링

4. 라이브러리

- BLEDevice.h
- BLEServer.h
- BLEUtils.h
- BLE2902.h
- Adafruit_NeoPixel.h
- SPIFFS.h
- esp_system.h

2. 배포 상세

Front-End) Flutter application 배포

1. 개발 환경 Setting

Chocolatey

Installing Chocolatey

Chocolatey is software management automation for Windows that wraps installers, executables, zips, and scripts into compiled packages. Chocolatey integrates w/SCCM, Puppet, Chef, etc. Chocolatey is trusted by businesses to manage software deployments.

 <https://chocolatey.org/install#install-with-cmdexe>



1. powershell을 관리자 권한으로 실행
2. Get-ExecutionPolicy 입력
3. Restricted가 뜬다면 설치 코드 입력 (공식문서에 있음 Individual 선택)

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::Se
```

4. 설치가 완료되면 choco 입력으로 설치 및 버전 확인
5. powershell에서 chocolatey가 설치된 상태에서, choco install flutter 입력
6. flutter —version으로 버전확인

Flutter 3.7.10 • channel stable • <https://github.com/flutter/flutter.git>
 Framework • revision 4b12645012 (2 weeks ago) • 2023-04-03 17:46:48 -0700
 Engine • revision ec975089ac
 Tools • Dart 2.19.6 • DevTools 2.20.1

7. 환경변수 설정
 시스템 변수에서 Path에 C:\tools\flutter\bin 추가 (flutter\bin 위치 추가)
8. flutter doctor 실행 → 정상적으로 flutter 개발을 할 수 있는 상태인지 검사해주는 명령어

```

관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Windows\system32> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel master, 3.10.0-7.0.pre.16, on Microsoft Windows [Version 10.0.19045.2728], locale ko-KR)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Build Tools 2019 16.11.22)
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.77.3)
[✓] Connected device (4 available)
[✓] Network resources

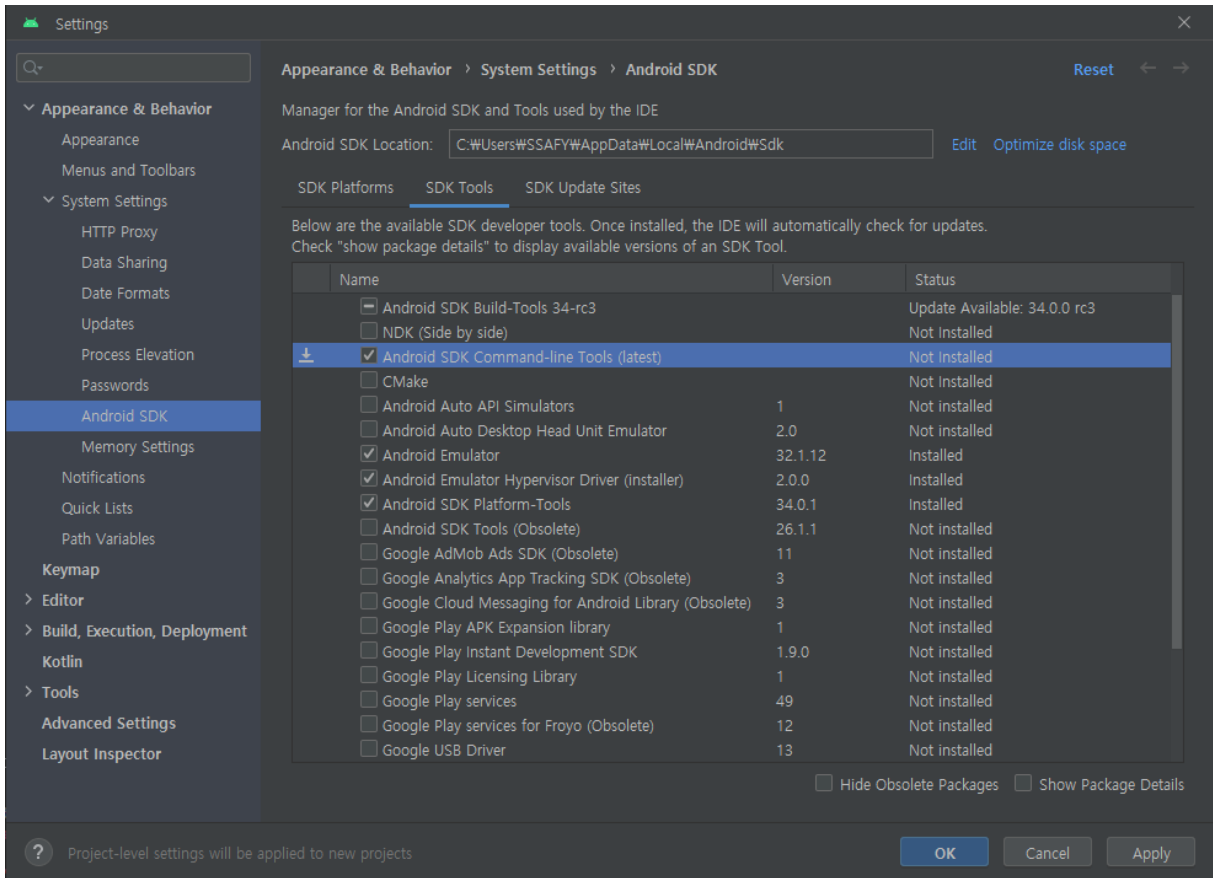
No issues found!
PS C:\Windows\system32>

```

전부 그린체크가 되면 정상적으로 flutter 개발을 할 수 있는 상태이다.

9. Android sdkmanager not found. Update to the latest Android SDK and ensure that the cmdline-tools are installed to resolve this.
10. Android Studio 실행
11. SDK Manager 실행
12. [Android SDK] > [SDK Tools] 에서 오른쪽 하단의 Hide Obsolete Packages 체크박스 해제
13. Android SDK Command-line Tools(latest) 체크 후 OK
14. install 끝나면 재확인

▼ 예시



15. Android Emulator 설치

Download Android Studio & App Tools - Android Developers Android Studio provides app builders with an integrated development environment (IDE) optimized for Android apps. Download Android Studio today. <https://developer.android.com/studio>

23.04.18 시점 최신버전 : Android Studio Flamingo

16. HAXM → 인텔 하드웨어 가속 관리자

안드로이드 에뮬레이터 속도를 향상시키고 성능을 향상시킨다.

Android studio에서 flutter install

빈 프로젝트를 생성 → file → Settings → plugin 검색 후 plugins 선택 → flutter install 후 재실행

17. Android Emulator 기기 설정 → Create Device 선택

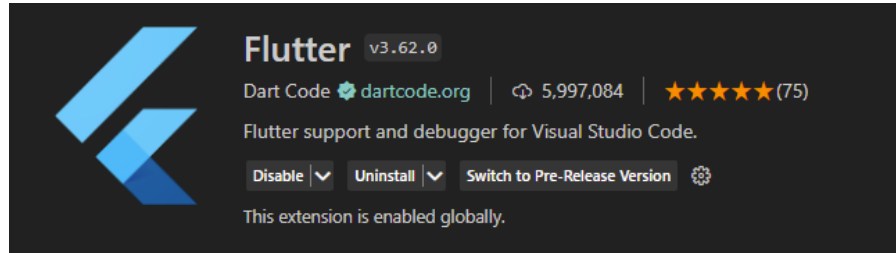
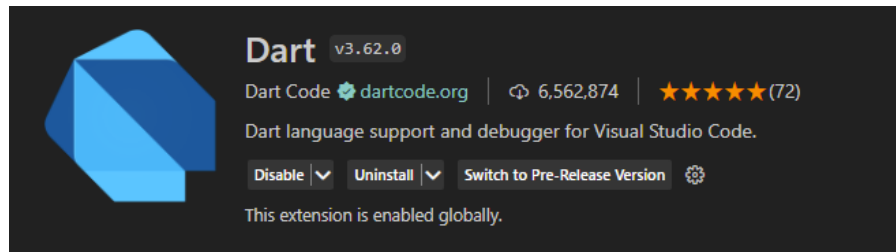
android version 선택 (UpsideDownCake가 23.2.8 최신이라 선택함)

랜더링을 Hardware로 선택 후 Finish로 완료

18. git으로 프로젝트를 클론

19. flutter pub get 실행

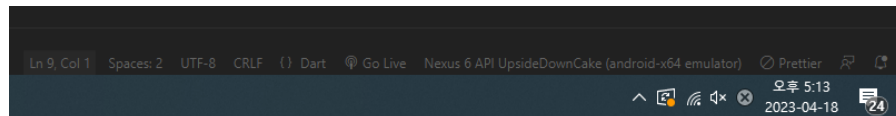
20. vscode에서 flutter 개발 extention



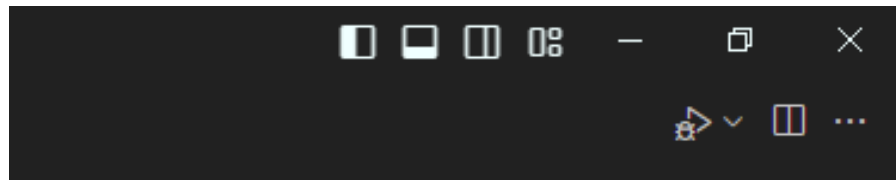
20. 우하단에서 연결 디바이스 선택

android studio에서 만든 device를 골라준다.

(여기서는 Nexus 6 API UpsideDownCake emulator)



21. 우상단의 Start Debugging을 실행한다.



22. 프로젝트 실행 완료

Back-End) Springboot Server 배포

1. Build(Jenkins 빌드 시 사용)

```
cd back/habitat
chmod +x gradlew
./gradlew build
```

2. Dockerfile

```
FROM openjdk:11-jre
COPY build/libs/bora-0.0.1-SNAPSHOT.jar app.jar
ENTRYPOINT ["java", "-jar", "app.jar"]
```

3. Docker container run

```
cd S08P31A704/back/habitat

docker stop habitat-server
docker rm habitat-server
docker rmi habitat-server

docker build -t habitat-server .
docker run -d -p 8081:8080 --name habitat-server -v /etc/localtime:/etc/localtime:ro -e TZ=Asia/Seoul habitat-server
```

그 외 설정 파일

```
# /back/habitat/src/main/resources/application.properties

# DB - MySQL
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://k8a704.p.ssafy.io:3306/S08P31A704?serverTimezone=Asia/Seoul&useUnicode=true&characterEncoding=utf8
spring.datasource.username=S08P31A704
spring.datasource.password={secret}

# DB - Redis
spring.redis.host=k8a704.p.ssafy.io
spring.redis.port=6379
spring.data.redis.repositories.enabled= false

# JPA
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.show-sql=false

# SWAGGER
spring.mvc.pathmatch.matching-strategy=ant_path_matcher

# S3
cloud.aws.stack.auto=false
cloud.aws.region.static=ap-northeast-2
cloud.aws.credentials.access-key={secret}
cloud.aws.credentials.secret-key={secret}
cloud.aws.s3.bucket=your-habitat

# Spring Security
jwt.header= Authorization
jwt.secret= {jwt_secret}

# Redis Cache
spring.cache.type=redis
spring.cache.redis.cache-null-values=true

# Logging
logging.file.name = ./test-log.log
```

3. 시연 시나리오

어플리케이션 다운로드

Google 플레이스토어 다운로드 <https://play.google.com/store/apps/details?id=com.status200.habitat>

로그인/회원가입

구글 로그인

코스터 연결

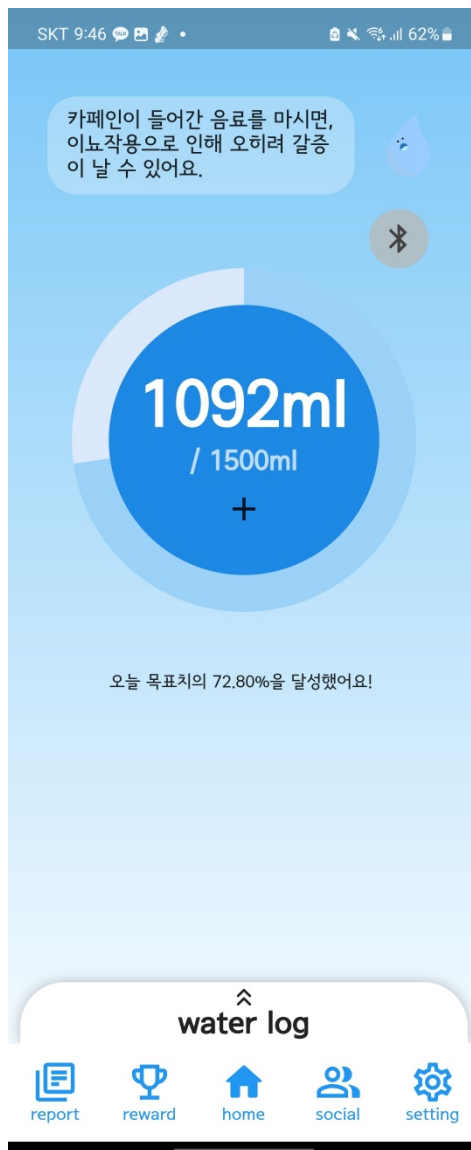
*핸드폰의 블루투스 기능이 켜져 있어야 합니다

설정 페이지 > 코스터 연결 > 블루투스 기기 스캔 > 코스터 선택

메인 페이지

- 중단) 음수량 : 유저의 오늘 실시간 음수량 추적

- 하단) 일일 음수 기록 조회



리워드 페이지

- 상단) 경험치 : 현재 유저가 키우고 있는 꽃의 상태(꽃 종류, 경험치, 레벨) 조회
- 하단) 컬렉션 : 모든 꽃에 대한 유저의 상태(수확/해금/???) 조회



소셜 페이지

- 상단) 나의 친구코드 : 유저의 친구코드 조회 및 복사
- 중단) 친구신청 목록 : 유저에게 도착한 친구신청 목록 조회 및 수락/거절
- 하단) 친구 목록 : 유저의 친구 목록 조회 및 찌르기 기능



설정 페이지

- 상단) 마이프로필 : 닉네임, 프로필 사진 수정
- 중단1) 목표 설정 : 목표 음수량 수정, 목표 음수량 추천받기
- 중단2) : 코스터 연결, 삭제
- 하단) 캐시 삭제

Setting 설정



한겨울의 노란 해바라기



목표 음수량 : 1500ml

목표설정

목표 음수량 설정

목표 음수량 추천받기

코스터 설정

코스터 등록

코스터 삭제

코스터 알림설정

앱 설정

앱 알림설정

앱 캐시 삭제



report



reward



home



social



setting

망우리

- 서비스 이용 팁

