Flutter 바이브코딩 개발 환경 설정 가이드

Flutter 바이브코딩을 위한 개발 환경 자동 설치 가이드문서입니다.



⚠ 중요: macOS 전용

이 스크립트들은 macOS 전용으로 설계되었으며, 수정 없이는 Windows나 Linux에서 작동하 지 않습니다. Windows 사용자는 WSL2를 사용하거나 Flutter SDK, FVM, Gemini CLI, Firebase MCP 등을 수동으로 설치하세요.

문서 개요

- 1. 문서와 설치 스크립트 안내
- 2. Flutter 바이브코딩 도구
- 3. install.sh
- 4. doctor.sh
- 5. 환경 구성 후 추가 설정
- 6. 문제 해결
- 7. 참고 자료

1. 문서와 설치 스크립트 안내

이 문서는 Flutter 바이브코딩 개발 환경 설정을 간소화하는 두 가지 스크립트와 함께 배포됩니다.

- install.sh : Flutter SDK 및 바이브코딩 관련 도구 자동 설치 스크립트
- doctor.sh : 바이브코딩 개발 환경을 검증하는 스크립트

2. Flutter 바이브코딩 도구

Flutter 바이브코딩 환경을 자동으로 설치하는 install.sh 스크립트는 아래 도구를 개발 환경에 구성합니다. 구성되는 도구는 아래와 같습니다.

- 2.1. FVM (Flutter Version Management) Flutter 버전 관리 도구
- 2.2. Flutter SDK FVM을 통해 최신 stable 버전 설치
- 2.3. Dart MCP Server Dart/Flutter용 Model Context Protocol 서버
- 2.4. Firebase MCP MCP를 통한 Firebase 통합
- 2.5. **Gemini CLI** Google의 커맨드라인 가반 Gemini AI 도구

install.sh 스크립트 실행 시 바이브코딩 개발 환경 구성에 사용되는 도구가 Homebrew, Git, Node.js(NPM)에 대해 추가 의존성이 있어 자동으로 설치됩니다.

3. install.sh

3.1 install.sh 살펴보기

설치 스크립트는 다음 단계를 수행합니다:

- 3.1.1. **사전 요구사항 확인** 필요시 Homebrew, Git, Node.js/npm 설치
- 3.1.2. Homebrew로 FVM 설치
- 3.1.3. FVM으로 Flutter SDK 설치: FVM을 사용하여 최신 stable 버전 설치. 전역 기본 버전으로 설정 후 ~/fvm/default 에 설치
- 3.1.4. **환경 구성**: PATH에 FVM의 Flutter 추가 및 .zshrc 또는 .bashrc 환경변수 업데이트
- 3.1.5. Dart/Flutter MCP 서버 설정: Dart 버전 확인 (3.9+ 필요) 후 MCP 서버 기능 활성화
- 3.1.6. **Firebase MCP 설치**: MCP 지원을 위해 버전 firebase-tools 13.21.0+ 이상의 버젼을 전역 설치
- 3.1.7. **Gemini CLI 설치**: npm을 통해 전역 설치

3.2 install.sh 실행하기

install.sh 스크립트 실행 권한 부여

chmod +x install.sh doctor.sh

Flutter 바이브코딩 환경 구성을 위해 install.sh 스크립트 실행

./install.sh

터미널을 재시작하거나 환경변수 다시 로드

```
source ~/.zshrc # zsh의 경우
# 또는
source ~/.bashrc # bash의 경우
```

4. doctor.sh

4.1 doctor.sh 살펴보기

doctor.sh 스크립트는 다음 항목을 종합적으로 확인합니다:

- 4.1.1. 사전 요구사항 확인: Homebrew, Git, Node.js, npm
- 4.1.2. Flutter SDK 확인: 설치 디렉토리 검사, 명령어 작동 여부 확인 등 수행
- 4.1.3. Dart 확인
- 4.1.4. FVM: 설치 여부 및 FVM을 통해 설치된 Flutter 버전 확인
- 4.1.5. MCP 서버: Dart MCP 서버와 Firebase MCP 서버 사용 가능 여부
- 4.1.6. Gemini CLI: 설치 여부와 인증 등 구성 상태 확인
- 4.1.7. Flutter Doctor 실행: flutter doctor -v 실행해 일반적인 문제 확인

4.2 doctor.sh 실행하기

install.sh 실행해 플러터 바이브코딩 개발 환경을 구성한 후 환경 검증을 위해 doctor.sh 스 크립트를 실행하세요

./doctor.sh

스크립트는 아래와 같이 초록색, 노란색, 빨간색으로 메세지를 표시합니다:

- 🗸 초록색 체크: 모든 것이 올바르게 작동
- 1 노란색 경고: 중요하지 않은 문제
- × **빨간색 오류**: 주의가 필요한 중요한 문제

5. 환경 구성 후 추가 설정

성공적인 설치 후 다음 단계를 완료하세요

- 5.1. Android 개발을 계획하는 경우 Android 라이선스 수락
- 5.2. Gemini CLI를 설치한 경우 Gemini CLI 구성
- 5.3. Gemini CLI Firebase 통합을 위해 Firebase 확장 프로그램 설치
- 5.4. MCP 서버 구성

5.1. Android 개발을 계획하는 경우 Android 라이선스 수락

flutter doctor --android-licenses

5.2. Gemini CLI를 설치한 경우 Gemini CLI 구성

gemini

Gemini CLI가 실행되면 /auth 를 입력하여 인증을 진행합니다:

- 1. Google Al Studio에서 API 키 발급 (https://aistudio.google.com/apikey)
- 2. API 키 입력
- 3. 기본 모델 선택

5.3. Gemini CLI Firebase 통합을 위해 Firebase 확장 프로그램 설치

gemini extensions install https://github.com/gemini-cli-extensions/firebase/

5.4. 편집기 또는 IDE에서 MCP 서버를 사용을 위해 MCP 서버 구성

- Dart/Flutter MCP: VS Code with Dart Code 확장 프로그램 v3.116+ 필요
- Firebase MCP: Claude Desktop, Cursor 또는 기타 MCP 클라이언트에서 구성

Gemini CLI용 MCP 구성 예시 (~/.gemini/settings.json):

```
"mcpServers": {
  "dart": {
    "command": "dart",
    "args": ["mcp-server"]
  },
  "firebase": {
    "command": "npx",
    "args": ["-y", "firebase-tools@latest", "mcp"]
```

6. 문제 해결

- 6.1. flutter 명령을 찾을 수 없음
- 6.2. 권한 문제
- 6.3. Dart MCP 서버를 사용할 수 없음
- 6.4. Firebase MCP를 사용할 수 없음
- 6.5. Windows 사용자

6.1. flutter 명령을 찾을 수 없음

설치 후 flutter 명령을 찾을 수 없는 경우:

- 1. 터미널 재시작
- 2. 또는 셸 설정 다시 로드:

```
source ~/.zshrc # 또는 ~/.bashrc
```

3. PATH가 구성되었는지 확인:

```
echo $PATH | grep fvm
```

4. FVM이 올바르게 설치되었는지 확인:

```
fvm --version
fvm list
```

6.2. 권한 문제

권한 오류로 install.sh 와 doctor.sh 를 실행할 수 없는 경우

chmod +x install.sh doctor.sh

6.3. Dart MCP 서버를 사용할 수 없음

Dart SDK 3.9+가 필요합니다. Flutter 업데이트:

flutter upgrade
dart --version

6.4. Firebase MCP를 사용할 수 없음

최신 firebase-tools로 업데이트:

npm install -g firebase-tools@latest

또는 npx를 통해 사용:

npx -y firebase-tools@latest mcp

6.5. Windows 사용자

이 스크립트들은 Windows에서 기본적으로 호환되지 않습니다. Windows를 사용하는 경우 WSL2를 통해 install.sh 과 doctor.sh 를 실행해 볼 수 있겠으나 윈도우 머신이 없어 검증하지 못했습니다.

7. 참고 자료

- Flutter: https://docs.flutter.dev
- FVM: https://fvm.app
- Dart MCP: https://dart.dev/tools/mcp-server
- Firebase MCP: https://firebase.google.com/docs/cli/mcp-server
- Gemini CLI: https://github.com/google-gemini/gemini-cli

감사합니다.

문제를 보고하거나 개선 사항이 있다면 이슈를 등록해주세요.

https://github.com/aiiiiiiden/tools