Flutter Vibe Coding Environment Setup Guide

An automated installation guide for Flutter Vibe Coding development environment.



⚠ IMPORTANT: macOS Only

These scripts are designed exclusively for macOS and will not work on Windows or Linux without modification. Windows users should use WSL2 or manually install Flutter SDK, FVM, Gemini CLI, Firebase MCP, etc.

Table of Contents

- 1. Documentation and Installation Scripts Overview
- 2. Flutter Vibe Coding Tools
- 3. install.sh
- 4. doctor.sh
- 5. Post-Installation Configuration
- 6. Troubleshooting
- 7. Resources

1. Documentation and Installation Scripts Overview

This documentation is distributed with two scripts to streamline Flutter Vibe Coding development environment setup.

- **install.sh** : Automated installation script for Flutter SDK and Vibe Coding related tools
- doctor.sh : Script to verify the Vibe Coding development environment

2. Flutter Vibe Coding Tools

The install.sh script automatically configures the following tools in your development environment:

- 2.1. FVM (Flutter Version Management) Flutter version management tool
- 2.2. Flutter SDK Latest stable version installed via FVM
- 2.3. Dart MCP Server Model Context Protocol server for Dart/Flutter
- 2.4. Firebase MCP Firebase integration via MCP
- 2.5. Gemini CLI Google's command-line based Gemini Al tool

When running the install.sh script, Homebrew, Git, and Node.js (NPM) are automatically installed as additional dependencies for the Vibe Coding development environment configuration.

3. install.sh

3.1 Understanding install.sh

The installation script performs the following steps:

- 3.1.1. Check prerequisites and install Homebrew, Git, Node.js/npm if needed
- 3.1.2. Install FVM via Homebrew
- 3.1.3. **Install Flutter SDK via FVM**: Install latest stable version using FVM. Set as global default version and install to ~/fvm/default
- 3.1.4. **Configure environment**: Add FVM's Flutter to PATH and update .zshrc or .bashrc environment variables
- 3.1.5. **Set up Dart/Flutter MCP Server**: Verify Dart version (requires 3.9+) and enable MCP server functionality
- 3.1.6. **Install Firebase MCP**: Globally install firebase-tools version 13.21.0+ or higher for MCP support
- 3.1.7. Install Gemini CLI: Globally install via npm

3.2 Running install.sh

Grant execution permission to the install.sh script

```
chmod +x install.sh doctor.sh
```

Run the install.sh script to configure the Flutter Vibe Coding environment

```
./install.sh
```

Restart terminal or reload environment variables

```
source ~/.zshrc # for zsh
# or
source ~/.bashrc # for bash
```

4. doctor.sh

4.1 Understanding doctor.sh

The doctor.sh script performs comprehensive checks on the following items:

- 4.1.1. Check prerequisites: Homebrew, Git, Node.js, npm
- 4.1.2. Check Flutter SDK: Verify installation directory, command availability, etc.
- 4.1.3. Check Dart
- 4.1.4. FVM: Check installation status and Flutter versions installed via FVM
- 4.1.5. MCP Servers: Check availability of Dart MCP Server and Firebase MCP Server
- 4.1.6. **Gemini CLI**: Check installation status and configuration state including authentication
- 4.1.7. Run Flutter Doctor: Execute flutter doctor -v to check for common issues

4.2 Running doctor.sh

After running install.sh to configure the Flutter Vibe Coding development environment, run the doctor.sh script to verify the environment

```
./doctor.sh
```

The script displays messages in green, yellow, and red as follows:

- Green check: Everything is working correctly
- **Yellow warning**: Non-critical issues
- × Red error: Critical problems requiring attention

5. Post-Installation Configuration

Complete the following steps after successful installation

- 5.1. Accept Android licenses if planning Android development
- 5.2. Configure Gemini CLI if installed
- 5.3. Install Firebase extension for Gemini CLI Firebase integration
- 5.4. Configure MCP servers

5.1. Accept Android Licenses if Planning Android Development

flutter doctor --android-licenses

5.2. Configure Gemini CLI if Installed

gemini

After launching Gemini CLI, enter /auth to authenticate:

- 1. Get API key from Google Al Studio (https://aistudio.google.com/apikey)
- 2. Enter API key
- 3. Select default model

5.3. Install Firebase Extension for Gemini CLI Firebase Integration

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

5.4. Configure MCP Servers for Use in Your Editor or IDE

- Dart/Flutter MCP: Requires VS Code with Dart Code extension v3.116+
- Firebase MCP: Configure in Claude Desktop, Cursor, or other MCP clients

Example MCP configuration for Gemini CLI (~/.gemini/settings.json):

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

6. Troubleshooting

- 6.1. Flutter command not found
- 6.2. Permission issues
- 6.3. Dart MCP Server not available
- 6.4. Firebase MCP not available
- 6.5. Windows users

6.1. Flutter Command Not Found

If the flutter command is not found after installation:

- 1. Restart your terminal
- 2. Or reload your shell configuration:

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

3. Verify PATH is configured:

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

4. Verify FVM is installed correctly:

```
fvm ——version
fvm list
```

6.2. Permission Issues

If you cannot execute install.sh and doctor.sh due to permission errors

chmod +x install.sh doctor.sh

6.3. Dart MCP Server Not Available

Dart SDK 3.9+ is required. Update Flutter:

```
flutter upgrade
dart --version
```

6.4. Firebase MCP Not Available

Update to latest firebase-tools:

```
npm install -g firebase-tools@latest
```

Or use via npx:

```
npx -y firebase-tools@latest mcp
```

6.5. Windows Users

These scripts are **not natively compatible with Windows**. If you are on Windows, you can try running install.sh and doctor.sh via WSL2, but this has not been verified as we don't have a Windows machine for testing.

7. Resources

- 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

Thanks

Please report issues or suggest improvements.

https://github.com/aiiiiiiden/tools