

Introduction

The Flutter Smart Home Control App is a mobile application that allows users to control various smart home devices in different rooms. Users can add rooms and specify the devices within each room, such as lights, fans, or other appliances. The app provides a user-friendly interface to toggle the state of these devices, and it uses MQTT (Message Queuing Telemetry Transport) to communicate with the smart home devices.

Prerequisites

Before you begin using this Flutter app, ensure you have the following prerequisites in place:

- **Flutter Development Environment:** You need to have Flutter and Dart installed on your development machine. You can follow the [official Flutter installation guide](#) for instructions.
- **Flutter IDE:** You can use popular integrated development environments (IDEs) like Android Studio or Visual Studio Code with the Flutter plugin installed.
- **MQTT Server:** You must have access to an MQTT server for the app to connect to. The app is configured to connect to a specific MQTT server by default. You can modify the server details in the code as needed.

App Structure

The Flutter Smart Home Control App consists of several components:

- **Main App (main.dart):** This file is the entry point of the application and contains the **MyApp** widget, which sets up the overall app structure.
- **Home Screen (home_screen.dart):** The main user interface of the app is implemented in this widget. It displays a list of rooms, each represented as a **RoomCard**.
- **RoomCard (room_card.dart):** This widget represents a single room and its associated devices. It allows users to toggle the state of each device using switches.
- **MQTT Client (mqtt_client_wrapper.dart):** This component handles the MQTT communication with the smart home devices. It connects to the

MQTT server, publishes messages to control devices, and listens for updates.

Usage

Adding a Room

- Launch the app on your mobile device or emulator.
- Click the "+" button in the app's header to add a new room.
- Enter a name for the room in the "Room Name" field.
- Optionally, specify the number of switches in the "Number of Switches" field (default is 2).
- Enter the names of the switches (devices) in the "Switch Names" field, separated by commas (,).
- Click the "Add" button to create the room.

Controlling Devices

- Once a room is created, it will appear on the home screen as a **RoomCard**.
- Each **RoomCard** displays the room's name and a list of switches (devices) with corresponding toggle switches.
- Toggle the switches to control the state of the devices in each room. The app will publish MQTT messages to the configured MQTT server to control the devices with the **Topic** *<RoomName/SwitchName>*.

Deleting a Room

- Swipe a **RoomCard** to the left to reveal a delete button.
- Tap the delete button, and a confirmation dialog will appear.
- Enter the password "9702" to confirm the deletion (this password is hardcoded).
- If the password is correct, the room will be deleted. If not, an error message will be displayed.

MQTT Configuration

The app is pre-configured and the following data are hardcoded for specific broker to connect to the following MQTT server:

- **Broker:** fe9cc652a75744b99492d75f8ba5b6b2.s1.eu.hivemq.cloud

- **Port:** 8883 (secure)
- **Username:** amarkc
- **Password:** Amarkc9702

You can modify these MQTT server details by updating the **broker**, **port**, **username**, and **password** variables in the code as needed.

Dependencies

This Flutter app relies on the following external packages:

- **mqtt_client:** A Dart package for MQTT communication.
- **typed_data:** A Dart package for working with typed data buffers.
- These packages are included in the **pubspec.yaml** file and will be automatically fetched when you run the app

Testing

- **Web Client :** Go to [broker site](#) and login with Gmail [dammar0088@gmail.com](#). Navigate to **WebClient** and login user with **Username:** “amarkc” & **Password:** “Amarkc9702”. You can subscribe to topic you like. For example *beedroom1/light*, *beedroom2/fan*, *kitchen/oven*, etc.
- **CLI:** Install hiveMQ CLI as below :

```
brew tap hivemq/mqtt-cli
brew install mqtt-cli
```

Execute following command for subscription:

```
mqtt sub -h fe9cc652a75744b99492d75f8ba5b6b2.s1.eu.hivemq.cloud -p 8883 -s
-u amarkc -pw Amarkc9702 -t <Topic of subscription>
```

Conclusion

The Flutter Smart Home Control App provides a convenient way to manage and control smart home devices in different rooms. By following the instructions above, you can set up and use the app to control your own smart home devices through MQTT communication. For any further customization or modifications, you can refer to the Flutter documentation and the official packages used in the app.