

MQTT Chat App with Expo

A modern MQTT chat application built with Expo and React Native, featuring real-time messaging, topic management, and a clean UI.

Features

- Real-time MQTT messaging
- Separate windows for sent/received messages
- Topic management
- Connection settings configuration
- Dark/Light theme support
- Bottom tab navigation

Quick Start

1. Create a new Expo project: `bash npx create-expo-app my-mqtt-app cd my-mqtt-app`
2. Install dependencies: `bash npx expo install react_native_mqtt @react-native-async-storage/async-storage expo-router@^4.0.17 react-native-reanimated expo-status-bar`
3. Clone this repository and copy the following files:

```
my-mqtt-app/
├── app/
│   ├── (tabs)/
│   │   ├── index.tsx      # Main chat screen
│   │   ├── topics.tsx     # Topic management
│   │   ├── settings.tsx  # MQTT settings
│   │   └── _layout.tsx    # Navigation setup
│   └── components/
│       ├── mqtt/
│       │   ├── MessageInput.tsx
│       │   ├── MessageWindows.tsx
│       │   ├── MQTTStatus.tsx
│       │   └── styles.ts
│       ├── ThemedText.tsx
│       └── ThemedView.tsx
├── hooks/
│   ├── useMQTTConnection.ts
│   └── useColorScheme.ts
├── types/
│   └── react_native_mqtt.d.ts
```

1. Update tsconfig.json: `json { "extends": "expo/tsconfig.base", "compilerOptions": { "module": "esnext", "strict": true, "baseUrl": ".", "paths": { "@": "src" } } }`
2. Start the app: `bash npx expo start`

MQTT Configuration

Default configuration uses HiveMQ's public broker: - Broker: `broker.hivemq.com` - Port: `8000` (WebSocket) - Default topic: `expo/test`

You can modify these settings in the app's settings screen.

Project Structure

```
├── app/                # Main application screens
├── components/         # Reusable components
│   └── mqtt/           # MQTT-specific components
├── hooks/              # Custom React hooks
└── types/              # TypeScript definitions
```

