Backend - Dokumentation

Ziel des Backends:

Das Backend ist verantwortlich für die Kommunikation zwischen Datenbank, Frontend und Mikrokontroller. Das Backend stellt eine REST API zur Verfügung über die das Frontend Daten bezieht, darüber hinaus sorgt das Backend für die Kommunikation mit dem ESP32, das Einspeisen der Daten des ESP32 in die Datenbank, sowie eine Websocket Verbindung zum Frontend für das Übertragen von Echtzeitdaten.

Technologie:

Komponente	Technologie / Paket
Programmiersprache	Python 3.x
Web-Framework	Flask
Authentifizierung	JWT (JSON Web Tokens)
WebSocket	Flask-Sock
Datenbank	SQLite
Messaging	MQTT mit paho-mqtt
Rate-Limiting	Flask-Limiter
Sicherheit	Passwort-Hashing mit Werkzeug
CORS-Unterstützung	Flask-CORS
Datenformatierung	JSON
.env-Verwaltung	Python-dotenv

Aufbau:

Das Backend besteht aus dem Hauptteil app.py, welcher für die Bereitstellung der Endpunkte, Websockets sowie Datenbankzugriffe verantwortlich ist.

Darüber hinaus wird die gesamte Kommunikation zwischen Backend und ESP32 über den MQTT_handler.py abgewickelt. Dieser baut eine Verbindung zum MQTT Broker auf, hört auf Anfragen und kann auf vordefinierten Topics Nachrichten versenden.

Sicherheitsaspekte:

Das Backend ist durch unterschiedliche Methoden geschützt:.

- Admin Anmeldung – Stellt sicher, dass sich nur Admin User am Webinterface anmelden können. Die Passwörter in der Datenbank werden lediglich gehasht gespeichert.
- JSON Web Tokens (JWT) sorgen für sichere authentifizierte Sessions und schützen sensible Endpunkte vor nicht autorisierten Zugriffen
- MQTT Broker Passwort – Der MQTT Broker ist mit einem Passwort versehen, da die gesamte Kommunikation und Konfiguration über diese Schnittstelle läuft.
- Rate Limiting für Login – Das Backend begrenzt den Zugriff von Geräten (IP basiert), die sich zu häufig mit einem falschen Kennwort anmelden wollten.

Endpunkte:

Das Backend stellt folgende Endpunkte zur Verfügung:

Login

- **URL:** /login
- **Methode:** POST
- **Beschreibung:** Authentifiziert einen Benutzer und gibt ein JWT-Token zurück
- **Parameter:**
    - username: Benutzername
    - password: Passwort
- **Rückgabe:** JWT-Token bei erfolgreicher Authentifizierung
- **Hinweis:** Der Endpunkt verfügt über einen Ratenbegrenzer (10 Fehlversuche führen zu einer 5-minütigen Sperrung)

Konfiguration abrufen

- **URL:** /getConfig
- **Methode:** GET
- **Beschreibung:** Ruft die aktuelle Systemkonfiguration ab
- **Authentifizierung:** JWT erforderlich
- **Rückgabe:** Aktuelle Konfigurationsdaten

Konfiguration setzen

- **URL:** /setConfig
- **Methode:** POST
- **Beschreibung:** Aktualisiert die Systemkonfiguration
- **Authentifizierung:** JWT erforderlich
- **Parameter:** Konfigurationsdaten als JSON im Request-Body

- **Rückgabe:** Bestätigungsstatus

Zugangs-ID hinzufügen

- **URL:** /addAccessID
- **Methode:** POST
- **Beschreibung:** Fügt eine neue Zugangs-ID zum System hinzu
- **Authentifizierung:** JWT erforderlich
- **Rückgabe:** Bestätigungsstatus

Daten abrufen

- **URL:** /data
- **Methode:** GET
- **Beschreibung:** Ruft Dashboarddaten ab
- **Authentifizierung:** JWT erforderlich
- **Parameter:**
    - start_date (optional): Startdatum für Filterung
    - end_date (optional): Enddatum für Filterung
    - limit (optional, Standard: 100): Maximale Anzahl der zurückgegebenen Datensätze
- **Rückgabe:** Gefilterte Dashboarddaten

Vorfälle abrufen

- **URL:** /incidents
- **Methode:** GET
- **Beschreibung:** Ruft Vorfallsdaten ab
- **Authentifizierung:** JWT erforderlich
- **Parameter:**
    - start_date (optional): Startdatum für Filterung
    - end_date (optional): Enddatum für Filterung
    - type (optional): Vorfallstyp
    - limit (optional, Standard: 100): Maximale Anzahl der zurückgegebenen Datensätze
- **Rückgabe:** Gefilterte Vorfallsdaten

Benutzerprofil abrufen

- **URL:** /user/profile

- **Methode:** GET
- **Beschreibung:** Ruft das Profil des aktuellen Benutzers ab
- **Authentifizierung:** JWT erforderlich
- **Rückgabe:** Benutzerprofildaten

Statusprüfung

- **URL:** /health
- **Methode:** GET
- **Beschreibung:** Überprüft den Systemstatus
- **Rückgabe:** Systemstatusdaten und Zeitstempel

WebSocket

WebSocket-Verbindung

- **URL:** /ws
- **Beschreibung:** WebSocket-Endpunkt für Echtzeitkommunikation
- **Verwendung:** Verbindung über WebSocket-Protokoll