**Dokumentation zur Webanwendung: Verwaltung von Wohnungen und Mietern**

**Inhaltsverzeichnis**

1. [Installationsanleitung](https://chatgpt.com/c/6752b80e-8ae4-8002-9c3f-52317c362c40#installationsanleitung)
2. [API-Beschreibung](https://chatgpt.com/c/6752b80e-8ae4-8002-9c3f-52317c362c40#api-beschreibung)
   * [Wohnungen-Endpunkte](https://chatgpt.com/c/6752b80e-8ae4-8002-9c3f-52317c362c40#wohnungen-endpunkte)
   * [Mieter-Endpunkte](https://chatgpt.com/c/6752b80e-8ae4-8002-9c3f-52317c362c40#mieter-endpunkte)
3. [Technologie-Stack](https://chatgpt.com/c/6752b80e-8ae4-8002-9c3f-52317c362c40#technologie-stack)

**Installationsanleitung**

**1. Voraussetzungen**

* **Python 3.8+**: Für das Backend (Flask)
* **Node.js**: Für das Frontend (React)
* **MySQL oder MariaDB**: Als Datenbank
* **pip**: Python-Paketmanager
* **npm**: Node.js-Paketmanager

**2. Datenbank einrichten**

1. Starte deinen MySQL- oder MariaDB-Server.
2. Erstelle die Datenbank und Tabellen (siehe Quelldatei „wohnungsverwaltung.sql“)

**3. Backend installieren und starten**

1. Klone das Projekt oder lade den Quellcode herunter.
2. Navigiere zum Backend-Verzeichnis:
3. cd backend
4. Installiere die benötigten Python-Bibliotheken:
5. pip install flask flask-sqlalchemy flask-cors pymysql
6. Passe die Datenbankverbindung in app.py an:
7. app.config['SQLALCHEMY\_DATABASE\_URI'] = 'mysql+pymysql://<BENUTZER>:<PASSWORT>@localhost/wohnungsverwaltung'
8. Starte das Backend:
9. python app.py
10. Das Backend läuft unter [http://localhost:5000](http://localhost:5000/).

**4. Frontend installieren und starten**

1. Navigiere zum Frontend-Verzeichnis:
2. cd frontend
3. Installiere die benötigten npm-Pakete:
4. npm install
5. Starte die React-App:
6. npm start
7. Das Frontend läuft unter [http://localhost:3000](http://localhost:3000/).

**API-Beschreibung**

Die API basiert auf REST-Prinzipien und kommuniziert über JSON. Alle Anfragen erfolgen an http://localhost:5000.

**Wohnungen-Endpunkte**

**1. GET /wohnungen**

**Beschreibung**: Liefert alle Wohnungen.

**Beispielantwort**:

[

{

"id": 1,

"adresse": "Musterstraße 10",

"groesse": 75.0,

"mietpreis": 850.0

}

]

**2. POST /wohnungen**

**Beschreibung**: Fügt eine neue Wohnung hinzu.

**Request-Body**:

{

"adresse": "Musterstraße 10",

"groesse": 75.0,

"mietpreis": 850.0

}

**Antwort**:

{

"message": "Wohnung hinzugefügt"

}

**Mieter-Endpunkte**

**1. GET /mieter**

**Beschreibung**: Liefert alle Mieter.

**Beispielantwort**:

[

{

"id": 1,

"name": "Max Mustermann",

"email": "max@example.com",

"wohnung\_id": 1

}

]

**2. POST /mieter**

**Beschreibung**: Fügt einen neuen Mieter hinzu.

**Request-Body**:

{

"name": "Max Mustermann",

"email": "max@example.com",

"wohnung\_id": 1

}

**Antwort**:

{

"message": "Mieter hinzugefügt"

}

**Technologie-Stack**

**1. Frontend**

* **Framework**: React
* **Bibliotheken**: Axios für HTTP-Anfragen, React Hooks für State-Management

**2. Backend**

* **Framework**: Flask
* **Datenbankanbindung**: SQLAlchemy
* **Cross-Origin-Unterstützung**: Flask-CORS

**3. Datenbank**

* **Typ**: MySQL oder MariaDB
* **Verbindung**: Über SQLAlchemy