API Dokumentation

Diese API wurde mit FastAPI entwickelt. Die API basiert auf einer Neo4j-Graphdatenbank, in der das Straßennetz, relevante Knotenpunkte sowie Adressen und Points of Interest modelliert sind. Die Kommunikation zwischen der FastAPI-Anwendung und der Datenbank erfolgt über den offiziellen Neo4j-Python-Treiber. Die gesamte Logik ist modular aufgebaut und in mehrere Dateien untergliedert, wodurch eine klare Trennung zwischen API-Endpunkten, Routing-Logik, Datenbankzugriff und Koordinatenverarbeitung gewährleistet wird.

Beim Start der Anwendung wird eine FastAPI-Instanz initialisiert, welche verschiedene HTTP-GET-Endpunkte bereitstellt. Ein einfacher Root-Endpunkt dient dabei als Health-Check und ermöglicht eine schnelle Überprüfung, ob die API erreichbar ist. Darüber hinaus stehen mehrere funktionale Endpunkte zur Verfügung, die entweder der Adresssuche oder der Routenberechnung dienen.

Ein zentraler Bestandteil der API ist die Adress- und POI-Suche. Hierbei wird eine vom Nutzer eingegebene Adresse mit in der Datenbank gespeicherten Adressen abgeglichen. Da Nutzereingaben häufig unvollständig oder leicht fehlerhaft sind, erfolgt dieser Abgleich nicht als exakte Suche, sondern mithilfe eines Fuzzy-Matching-Verfahrens. Dazu werden sowohl die gespeicherten Adressen als auch die Nutzereingabe bereinigt und normalisiert. Anschließend wird eine Ähnlichkeitsmetrik auf Basis der Levenshtein-Distanz berechnet. Nur Treffer, die eine ausreichend hohe Ähnlichkeit aufweisen, werden berücksichtigt, wobei stets die am besten passende Adresse ausgewählt wird. Auf diese Weise erhöht sich die Robustheit der API gegenüber Tippfehlern und variierenden Schreibweisen.

Die API ermöglicht Suche nach Start-und Zieladressen. Dies ist insbesondere für die Routenberechnung relevant, da sowohl Start- als auch Zielpunkt eindeutig identifiziert werden müssen, bevor eine Route berechnet werden kann. Werden eine oder beide Adressen nicht gefunden, gibt die API eine entsprechende Fehlermeldung zurück.

Zur Routenberechnung nutzt die API KISS-Plugin. Die API stellt flexibles Routing bereit, bei dem verschiedene Präferenzen dynamisch kombiniert werden können. Der Nutzer kann dabei angeben, ob bestimmte Oberflächen vermieden werden sollen, ob Treppen problematisch sind, ob Steigungen berücksichtigt werden müssen und ob unterstützende Infrastrukturelemente wie Sitzbänke, Toiletten oder Unterstände entlang der Route wichtig sind.

Ein weiterer wichtiger Aspekt der API ist die Koordinatenverarbeitung. Die in der Neo4j-Datenbank gespeicherten Knotenpunkte liegen im UTM-Koordinatensystem vor, welches für präzise Distanzberechnungen im Routing besonders geeignet ist. Für die Darstellung der Route in gängigen Kartenanwendungen müssen diese Koordinaten jedoch in das WGS84-System (Breiten- und Längengrade) umgerechnet werden. Diese Umrechnung erfolgt automatisiert nach Abschluss der Routenberechnung. Die API gibt die Route schließlich als geordnete Liste von geografischen Koordinaten zurück, die direkt in Web- oder Mobilkarten visualisiert werden können.

Zusätzlich zur eigentlichen Route liefert die API Metainformationen wie die Gesamtstrecke, qualitative Bewertungen der Route sowie detaillierte Informationen zu gefundenen Points of Interest. Dadurch können Frontend-Anwendungen nicht nur eine Linie auf der Karte darstellen, sondern dem Nutzer auch erklären, warum eine bestimmte Route gewählt wurde und welche Unterstützungsangebote entlang des Weges verfügbar sind.

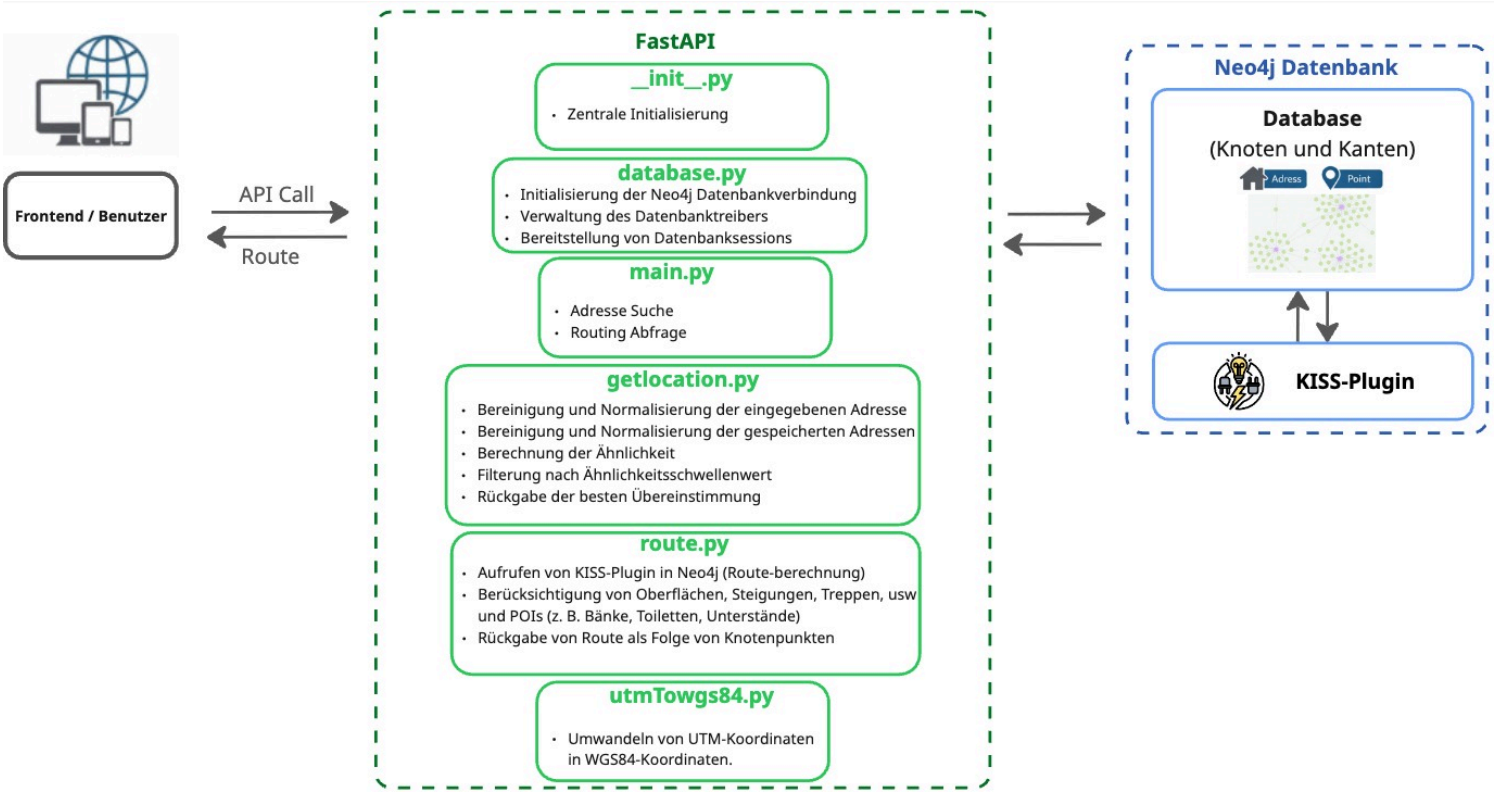

Fig. 1. API Diagram