OPEN AI API

Funktionsweise und Grundprinzipien

Einführung – Wie funktioniert die OpenAI API

Die OpenAI API bietet Zugriff auf Sprachmodelle, Embeddings und andere KI-Funktionen über einfache HTTP-Anfragen.

Sie arbeitet modellunabhängig - ob GPT, Embedding- oder Bildmodelle.

Jede Anfrage enthält Text (Prompt), Optionen und Authentifizierung per API-Key.

Die API antwortet mit JSON-Daten.

Grundprinzip: Anfrage und Antwort

Ein Client sendet eine JSON-Anfrage an den OpenAI-Endpunkt, z. B. /v1/chat/completions.

Diese enthält das Modell, Nachrichten oder Texteingaben und optionale Parameter wie Temperatur oder Max-Tokens.

Das Modell verarbeitet die Eingabe und gibt eine Antwort mit Token-Statistiken und generiertem Text zurück.

Komponenten einer API-Anfrage

Model: definiert, welches Sprach- oder Embedding-Modell verwendet wird.

Messages: enthält den Dialogverlauf (Rollen: system, user, assistant).

Parameters: steuern Verhalten, z. B. Kreativität, Länge, Sampling.

API Key: Authentifizierung für sichere Nutzung.

Der Chat-Workflow

Das Modell arbeitet im Chat-Format: Jede Nachricht besitzt eine Rolle (z. B. „user“, „assistant“, „system“).

Der System-Teil definiert die Persönlichkeit oder das Verhalten des Modells.

Die gesamte Historie dient als Kontext. Das Modell sieht also den gesamten Gesprächsverlauf bei jeder Anfrage. Ohne diesen Gesprächsverlauf weiß das Model (bzw. die Session) nichts davon.

Die Historie ist neben der System-Nachricht(en) eine Abfolge von User-Eingaben und Assistant-Antworten. Es ist üblich, den Verlauf (persistent) zu speichern und bei jeder Anfrage mitzusenden.

Tokenisierung und Kosten

Texte werden in Tokens zerlegt, die das Modell verarbeitet. Kosten und Limits basieren auf der Anzahl dieser Tokens.

Je nach Modell kostet ein Token-Bruchteil eines Cents.

Tokens umfassen Wörter, Satzzeichen und Wortteile. Das Verständnis von Token-Limits ist entscheidend für Performance und Preis.

Temperatur und Sampling

Temperature steuert die Zufälligkeit: niedriger Wert = präzise, hoher Wert = kreativ.

Top-p (Nucleus Sampling) legt fest, wie viel Wahrscheinlichkeitsmasse berücksichtigt wird, das heisst, wie breit das Model bei der Wortauswahl denkt (top_p=1 ist maximale Vielfalt, top_p=0.3 nur 30% der wahrscheinlichsten Worte genutzt).

Kombiniert bestimmen sie, ob das Modell eher konservativ oder explorativ antwortet.

Embeddings über die API

Neben Textgenerierung bietet die API auch Embeddings. Ein Text wird an /v1/embeddings gesendet, das Modell liefert einen Vektor zurück.

Diese Vektoren können für semantische Suche, Clustering oder RAG-Systeme (z. B. mit ChromaDB) verwendet werden.

Sicherheit und Datenschutz

Daten werden verschlüsselt übertragen (HTTPS).

OpenAI speichert Anfragen zeitweise zur Qualitätskontrolle, kann aber auf Wunsch deaktiviert werden.

Für sensible Daten ist Self-Hosting oder ein On-Prem-Ansatz (z. B. mit offenen Modellen) zu empfehlen.

Allgemeine Architekturprinzipien

Die Funktionsweise der OpenAI API gilt auch für andere Anbieter wie Anthropic, Mistral oder Google Gemini. Alle folgen einem ähnlichen Schema:

Client → HTTP-Request → Modell → JSON-Response.

Dadurch sind Implementierungen austauschbar, wenn man sich an Standard-APIs und Prompt-Strukturen hält.