gpt-4o

Endpunkt

Ziel-URI
https:...

Key

Bereitstellung

Name
gpt-4o

Bereitstellungsstatus
Erfolgreich

Bereitstellungstyp
Globaler Standard

Erstellt am
2025-10-25T20:14:58.704092Z

Erstellt von
service@meinedokbox.de

Geändert am
Oct 25, 2025 10:14 PM

Geändert von
service@meinedokbox.de

Versionsupgraderichtlinie
Sobald eine neue Standardversion verfügbar ist

Ratenlimit (Token pro Minute)
50.000

Ratenlimit (Anforderungen pro Minute)
300

Modellname
gpt-4o

Modellversion
2024-11-20

Lebenszyklusstatus
GenerallyAvailable

Erstellungsdatum
Dec 3, 2024 1:00 AM

Updatedatum
Dec 3, 2024 1:00 AM

Modelleinstellungsdatum
Mar 1, 2026 1:00 AM

Überwachung

Inhaltsfilter
DefaultV2

Sprache
Python

SDK
Azure OpenAI SDK

Authentifizierungstyp
Key Authentication

Erste Schritte

Nachfolgend finden Sie Beispielcodeausschnitte für einige Anwendungsfälle. Weite Informationen zum Azure OpenAI SDK finden Sie in der vollständigen Dokumentation und in Beispielen.

1. Authentifizierung mit API-Schlüssel

Stellen Sie für OpenAI-API-Endpunkte das Modell bereit, um die Endpunkt-URL und eine API-Schlüssel zur Authentifizierung beim Dienst generieren. In diesem Beispiel sind der Endpunkt und der Schlüssel Zeichenfolgen, die di Endpunkt-URL und den API-Schlüssel enthalten.

Die API-Endpunkt-URL und de API-Schlüssel finden Sie auf de Seite „Bereitstellungen + Endpunkt“, sobald das Modell bereitgestellt wurde.

Um einen Client mit dem OpenAI SDK unter Verwendung eines API-Schlüssels zu erstelle initialisieren Sie den Client, indem Sie Ihren API-Schlüssel a die SDK-Konfiguration übergeben. Dadurch können S sich authentifizieren und nahtl mit den Diensten von OpenAI interagieren.

```
import os
from openai import Az

client = AzureOpenAI(
    api_version="2024
    azure_endpoint="h
    api_key=subscript
)
```

2. Abhängigkeiten installieren

Installieren Sie das Azure Open AI-SDK mit pip (Erforder Python >=3.8):

```
pip install openai
```

3. Einfachen Beispielcode ausführen

Dieses Beispiel zeigt einen grundlegenden Aufruf der Chatabschluss-API. Der Aufruf synchron.

```
import os
from openai import Az

endpoint = "https://c
model_name = "gpt-4o"
deployment = "gpt-4o"

subscription_key = "<
api_version = "2024-1

client = AzureOpenAI(
    api_version=api_v
    azure_endpoint=en
    api_key=subscript
)

response = client.cha
    messages=[
        {
            "role": "
            "content"
        },
        {
            "role": "
            "content"
        }
    ],
    max_tokens=4096,
    temperature=1.0,
    top_p=1.0,
    model=deployment
)

print(response.choice
```

4. Weitere Beispiele erkunden

Ausführen einer mehrteiligen Unterhaltung

Dieses Beispiel zeigt eine Unterhaltung mit mehreren Gesprächen mit der Chatvervollständigungs-API. Wenn Sie das Modell für eine Chatanwendung verwenden, müssen Sie den Verlauf dieser Unterhaltung verwalten und di neuesten Nachrichten an das Modell senden.

```
import os
from openai import Az

endpoint = "https://c
model_name = "gpt-4o"
deployment = "gpt-4o"

subscription_key = "<
api_version = "2024-1

client = AzureOpenAI(
    api_version=api_v
    azure_endpoint=en
    api_key=subscript
)

response = client.cha
    messages=[
        {
            "role": "
            "content"
        },
        {
            "role": "
            "content"
        },
        {
            "role": "
            "content"
        },
        {
            "role": "
            "content"
        }
    ],
    max_tokens=4096,
    temperature=1.0,
    top_p=1.0,
    model=deployment
)

print(response.choice
```

Streamen der Ausgabe

Für eine bessere Benutzererfahrung sollten Sie die Antwort des Modells streamen, damit das erste Toke frühzeitig angezeigt wird und S nicht auf lange Antworten warten müssen.

```
import os
from openai import Az

endpoint = "https://c
model_name = "gpt-4o"
deployment = "gpt-4o"

subscription_key = "<
api_version = "2024-1

client = AzureOpenAI(
    api_version=api_v
    azure_endpoint=en
    api_key=subscript
)

response = client.cha
    stream=True,
    messages=[
        {
            "role": "
            "content"
        },
        {
            "role": "
            "content"
        }
    ],
    max_tokens=4096,
    temperature=1.0,
    top_p=1.0,
    model=deployment,
)

for update in respons
    if update.choices
        print(update.

client.close()
```