Inferencing-SDKs in Foundry Local integrieren

24.06.2025

> **ⓘ Wichtig**
>
> - Foundry Local ist in der Vorschau verfügbar. Öffentliche Vorschauversionen bieten frühzeitigen Zugriff auf Features, die sich in der aktiven Bereitstellung befinden.
> - Features, Ansätze und Prozesse können sich vor der allgemeinen Verfügbarkeit (General Availability, GA) noch ändern oder eine eingeschränkte Funktionalität aufweisen.

Foundry Local integriert sich in verschiedene inferencing SDKs - wie OpenAI, Azure OpenAI, Langchain usw. In diesem Handbuch erfahren Sie, wie Sie Ihre Anwendungen mit lokal ausgeführten KI-Modellen mithilfe beliebter SDKs verbinden.

Voraussetzungen

- Foundry Local installiert. Anweisungen zur Installation finden Sie im Artikel " Erste Schritte mit Foundry Local ".

Node.js Pakete installieren

Sie müssen die folgenden Node.js Pakete installieren:

Bash

```
npm install openai
npm install foundry-local-sdk
```

Mit dem Foundry Local SDK können Sie den lokalen Foundry-Dienst und die Lokalen Modelle verwalten.

Verwenden Sie das OpenAI SDK mit Foundry Local

Im folgenden Beispiel wird die Verwendung des OpenAI SDK mit Foundry Local veranschaulicht. Der Code initialisiert den lokalen Foundry-Dienst, lädt ein Modell und generiert eine Antwort mit dem OpenAI SDK.

Kopieren Sie den folgenden Code, und fügen Sie ihn in eine JavaScript-Datei mit dem Namen `app.js` ein:

JavaScript

```javascript
import { OpenAI } from "openai";
import { FoundryLocalManager } from "foundry-local-sdk";

// By using an alias, the most suitable model will be downloaded
// to your end-user's device.
// TIP: You can find a list of available models by running the
// following command in your terminal: `foundry model list`.
const alias = "phi-3.5-mini";

// Create a FoundryLocalManager instance. This will start the Foundry
// Local service if it is not already running.
const foundryLocalManager = new FoundryLocalManager()

// Initialize the manager with a model. This will download the model
// if it is not already present on the user's device.
const modelInfo = await foundryLocalManager.init(alias)
console.log("Model Info:", modelInfo)

const openai = new OpenAI({
  baseURL: foundryLocalManager.endpoint,
  apiKey: foundryLocalManager.apiKey,
});

async function generateText() {
  const response = await openai.chat.completions.create({
    model: modelInfo.id,
    messages: [
      {
        role: "user",
        content: "What is the golden ratio?",
      },
    ],
  });

  console.log(response.choices[0].message.content);
}

generateText();
```

Führen Sie den Code mit dem folgenden Befehl aus:

Bash

```bash
node app.js
```

Streamingantworten

Wenn Sie Streamingantworten empfangen möchten, können Sie den Code wie folgt ändern:

JavaScript

```
import { OpenAI } from "openai";
import { FoundryLocalManager } from "foundry-local-sdk";

// By using an alias, the most suitable model will be downloaded
// to your end-user's device.
// TIP: You can find a list of available models by running the
// following command in your terminal: `foundry model list`.
const alias = "phi-3.5-mini";

// Create a FoundryLocalManager instance. This will start the Foundry
// Local service if it is not already running.
const foundryLocalManager = new FoundryLocalManager()

// Initialize the manager with a model. This will download the model
// if it is not already present on the user's device.
const modelInfo = await foundryLocalManager.init(alias)
console.log("Model Info:", modelInfo)

const openai = new OpenAI({
  baseURL: foundryLocalManager.endpoint,
  apiKey: foundryLocalManager.apiKey,
});

async function streamCompletion() {
    const stream = await openai.chat.completions.create({
      model: modelInfo.id,
      messages: [{ role: "user", content: "What is the golden ratio?" }],
      stream: true,
    });

    for await (const chunk of stream) {
      if (chunk.choices[0]?.delta?.content) {
        process.stdout.write(chunk.choices[0].delta.content);
      }
    }
}

streamCompletion();
```

Führen Sie den Code mit dem folgenden Befehl aus:

Bash

```
node app.js
```