

Codestral und Claude 3.5 Sonnet in VS Code verwenden



Vielleicht kennst du schon den GitHub Copilot – ein Programmier-Assistent, den du z.B. in Visual Studio Code verwenden kannst. Eine Alternative bietet das französische KI-Unternehmen Mistral AI: **Codestral – ein Sprachmodell, das auf Programmiersprachen spezialisiert ist.**

Codestral kannst du entweder ähnlich wie den Copilot verwenden oder damit chatten, wie du es von ChatGPT gewohnt bist. Aktuell (Juni '24) bietet Mistral für das Sprachmodell eine Testphase an, sodass du es kostenlos verwenden kannst. Aber auch nach Ende dieser Phase dürften die Preise der API anfallen, die nach Anzahl der tatsächlich verwendeten Token abgerechnet wird – im Gegensatz zu den pauschalen 10\$ des GitHub Copilots.

In diesem Tutorial erfährst du, wie du mit Codestral Code erstellen kannst und es in Visual Studio Code einbettest, damit das Modell dir dort bei deinen Projekten zur Seite steht.

Update zu Claude 3.5 Sonnet

Kurz nach der Veröffentlichung erschien das Modell **Claude 3.5 Sonnet** von Anthropic. Dieses Sprachmodell erzielt meiner Meinung nach bessere Ergebnisse. Wie du Claude 3.5 in VS Code einbettest, [erfährst du weiter unten in diesem Tutorial](#).

Das ist Codestral

Ende Mai 2024 hat Mistral AI ihr neuestes Sprachmodell vorgestellt, das auf die Erstellung von Code spezialisiert ist. Codestral beherrscht laut Angaben des Entwicklers über 80 Programmiersprachen, darunter **Python, C++, Java und JavaScript**. Mit 22 Mrd. Parametern und einen Kontextfenster von 32.000 Token soll es alle anderen bis zur Veröffentlichung verfügbaren Sprachmodelle in vielen Benchmarks übertreffen.

	HumanEval Python	HumanEval C++	HumanEval bash	HumanEval Java	HumanEval PHP	HumanEval Typescript	HumanEval C#	HumanEval
Codestral 22B	81.1%	65.2%	42.4%	63.3%	68.9%	68.6%	43.7%	61.5%
CodeLlama 70B	67.1%	54.0%	28.5%	57.6%	62.1%	60.4%	36.1%	51.9%
DeepSeek Coder 33B	77.4%	67.1%	39.9%	60.1%	54.7%	62.9%	43.7%	57.6%
Llama 3 70B	76.2%	67.7%	39.2%	60.8%	66.5%	74.2%	46.2%	61.2%
								Average

Auch wenn es hierbei um keine Quantensprünge im Vergleich zum in den einzelnen Sprachen jeweils dahinter liegenden LLM handelt, ist Codestral doch offensichtlich ein sehr ernstzunehmender Mitspieler.

Das Unternehmen Mistral AI wurde 2023 in Paris gegründet und versteht sich selbst als europäische Alternative zu amerikanischen Anbietern wie OpenAI, Meta oder Google. Die Qualität ihrer Arbeit kann dabei überzeugen – neben dem hier vorgestellten Sprachmodell hat es auch **Mixtral 8x7B** entwickelt, eins der besten derzeit verfügbaren Large Language Models.

Eine Besonderheit ist dabei die Veröffentlichung als Open Source. Du kannst die Modelle von Mistral AI deshalb zu nichtkommerziellen Zwecken also [über Ollama lokal auf deinem eigenen Rechner installieren und nutzen](#) – falls dieser stark genug dafür ist.

Chatten mit Codestral

Um die Fähigkeiten von Codestral zu testen, brauchst du nichts weiter als ein kostenloses Nutzerkonto. Um eines anzulegen, besuche die [Webseite von Mistral AI](#), klicke auf **Log in** und dann auf **Registrieren**. Nachdem du deine verwendete E-Mail-Adresse verifiziert hast, musst du einen Workspace anlegen. Vergib einen Namen deiner Wahl und wähle **Einzelnutzer** (es sei denn du registriert dich als Unternehmen).

Richten Sie Ihren Workspace ein

Workspaces ermöglichen es Ihnen, Ihre Projekte an einem zentralen Ort zu verwalten.

Name des Workspace *

Dieser Name ist nicht öffentlich und muss nicht zwangsläufig eindeutig sein.

Workspace-Team

	Ich bin ein Einzelnutzer Ich möchte ein Konto für mich persönlich einrichten	<input checked="" type="radio"/>
	Ich bin Teil eines Unternehmens Ich möchte ein Konto für ein Unternehmen einrichten	<input type="radio"/>

☒


Ich akzeptiere die Mistral AI **Nutzungsbedingungen** und die **Datenschutzbestimmungen**.


Workspace erstellen

Lies und akzeptiere die Nutzungsbedingungen und klicke anschließend auf **Workspace erstellen** – fertig. Im Menü oben siehst du nun den Punkt **Le Chat**. Klicke hierauf und es öffnet sich ein Bereich, den du so ähnlich vermutlich bereits von ChatGPT kennst.

Rechts findest du ein Eingabefeld, in das du deine Frage bzw. Anweisung eintragen kannst. Darunter befindet sich ein Dropdown-Menü, in dem du das Sprachmodell einstellst. Wähle hier **Codestral** aus.



Frag etwas! 

Model: Codestral 

Die Antworten können Ungenauigkeiten enthalten.

- Large**
Überlegene Denkfähigkeiten
- Next**
Prototyp-Modell für erhöhte Kürze
- Small**
Schnell und kosteneffektiv
- ☒ **Codestral**
Schnell und auf Code spezialisiert

Anschließend kann es gleich mit einem Test weitergehen:

Schreibe einen Sketch für den Arduino, der im Setup "Hello world" im Seriellen Monitor ausgibt.

Und hier das Ergebnis in Le Chat:

```
void setup() {
```

```
// Öffne die serielle Verbindung mit einem Baudrate von 9600
Serial.begin(9600);

// Schreibe "Hello world" in den Seriellen Monitor
Serial.println("Hello world");
}

void loop() {
    // Hier ist nichts, da wir nur im Setup eine Nachricht
    // ausgeben wollen
}
```

Diese einfache Aufgabe hat Codestral also gemeistert. Den Prompt habe ich auf Deutsch geschrieben. In der Dokumentation zum Sprachmodell ist Englisch als Eingabesprache vermerkt – möglicherweise erhältst du auf Englisch bessere Ergebnisse bei komplizierteren Anweisungen, da dein deutschsprachiger Prompt nicht erst übersetzt werden muss.

So weit, so gut. Falls du möchtest, teste hier doch ein paar härtere Nüsse. **Beeindruckend ist jedenfalls die Geschwindigkeit, mit der Antworten generiert werden** – die scheint mir tatsächlich sogar etwas höher als jene des momentan aktuellen OpenAI-Modells GPT-4o zu sein.

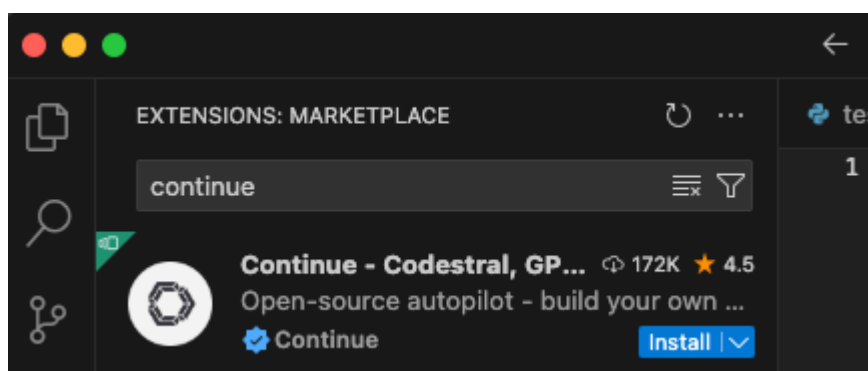
Codestral in Visual Studio Code verwenden

Nun zu einer weiteren Anwendung von Codestral. Du kannst das Modell ähnlich wie den Copilot in Visual Studio Code integrieren, damit es dir dort beim Programmieren zur Seite steht. Das bedeutet, dass du einerseits einen Chat zur Verfügung hast, ähnlich wie jenen in **Le Chat** auf der Webseite von Mistral AI. **Andererseits kannst du direkt mit deinem Sketch oder Script interagieren, indem du Codestral ums Debugging oder eine Ergänzung deines Codes bittest.**

Um Codestral in VS Code verwenden zu können, benötigst du zunächst eine Extension bzw. Plugin sowie einen API-Schlüssel von Mistral AI. Wie eingangs erwähnt, ist letzterer aktuell (Juni '24) innerhalb der Testphase kostenlos. Und die Extension ist natürlich auch kostenlos.

Continue installieren

Öffne zunächst die Bibliothek für die Extensions und suche nach **Continue** und installiere die Erweiterung mit einem Klick auf **Install**.



Öffne nun das Plugin und wähle im ersten Screen **Use your API key**. Wie vorhin erwähnt, könntest du Codestral auch lokal auf deinem Rechner ausführen, sofern er leistungsstark genug ist. Auf der nächsten Seite kannst du nun im ersten Dropdown-Menü die **Mistral API**, und im zweiten dann **Codestral** auswählen.



Welcome to Continue

Let's find the setup that works best for you

✨ Use your API key

Enter an OpenAI or other API key for the best experience. Continue will use the best available commercial models to index code. Code is only ever stored locally.

Chat: Whichever model you choose


Embeddings: Voyage Code 2





Autocomplete: Starcoder 7B via Fireworks AI

🔒 Local models


No code will leave your computer, but less powerful models are used. Works with Ollama, LM Studio and others.

Continue









Model Setup

To get started, choose a model and enter your API key. You can always reconfigure later by clicking the  icon in the bottom right.

- 1. Select a provider**

 Mistral API 
- 2. Select a model**

 Codestral 
- 3. Paste your API key**

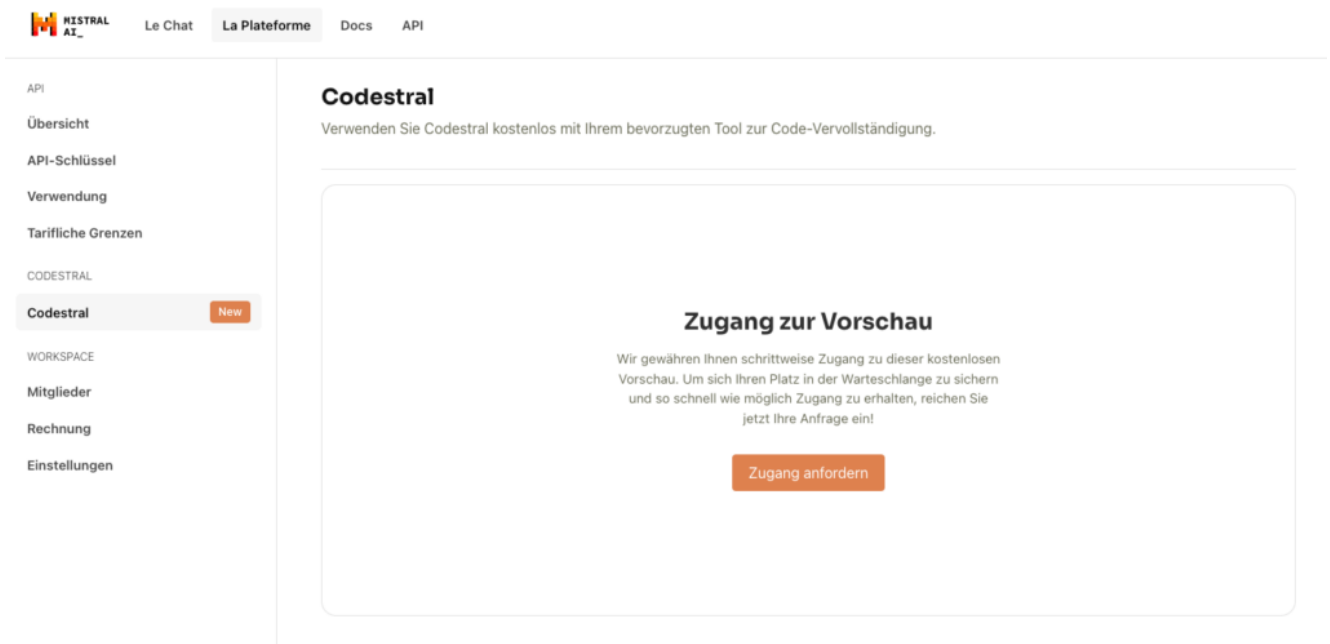
Note: Codestral requires a different API key from other Mistral models

Get API Key

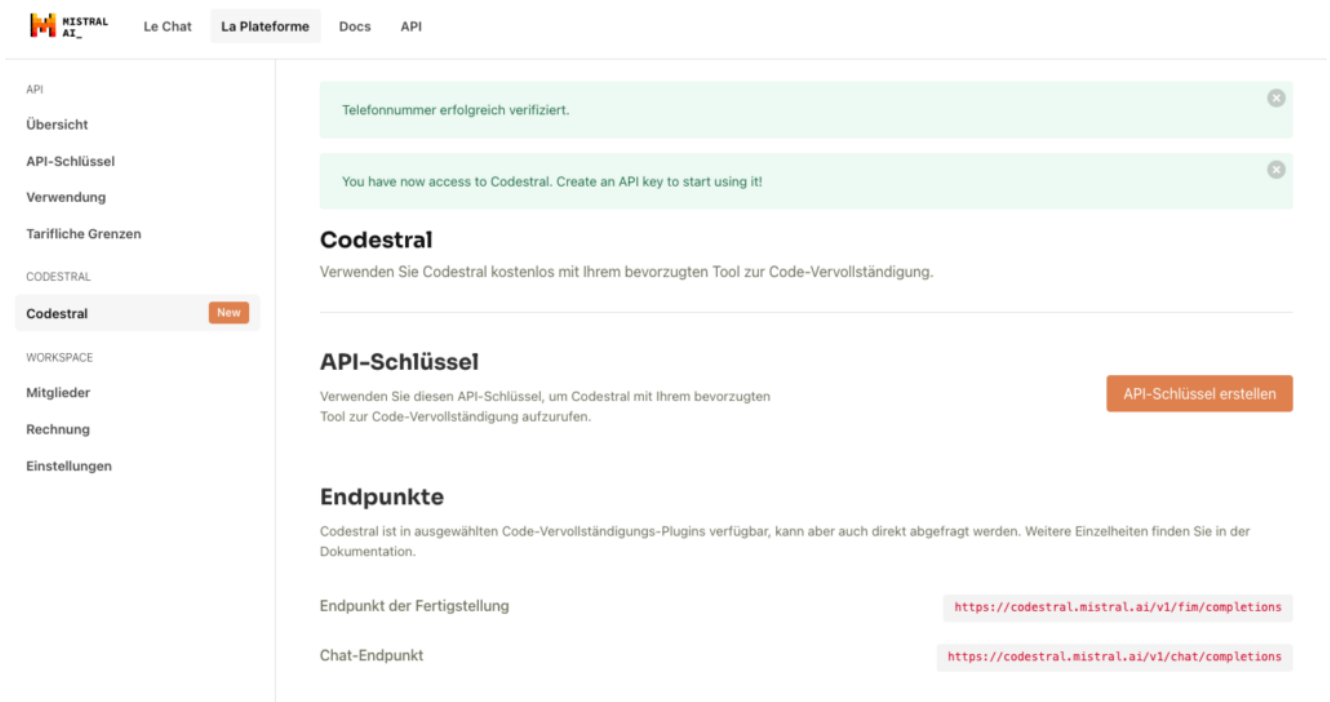
Enter API Key

Add ModelDone

Anschließend benötigst du den API-Key von Mistral, den du vermutlich noch nicht hast. Mit einem Klick auf **Get API Key** gelangst du auf die entsprechende Webseite von Mistral:



Wähle im Menü links **Codestral** aus und klicke dann rechts auf den Button **Zugang anfordern**. Anschließend benötigst du eine Handynummer, auf die du einen Code zur Verifizierung zugesendet bekommst. Anschließend kannst du nun den API-Key erstellen, mit einem Klick auf den Button **API-Schlüssel erstellen**.



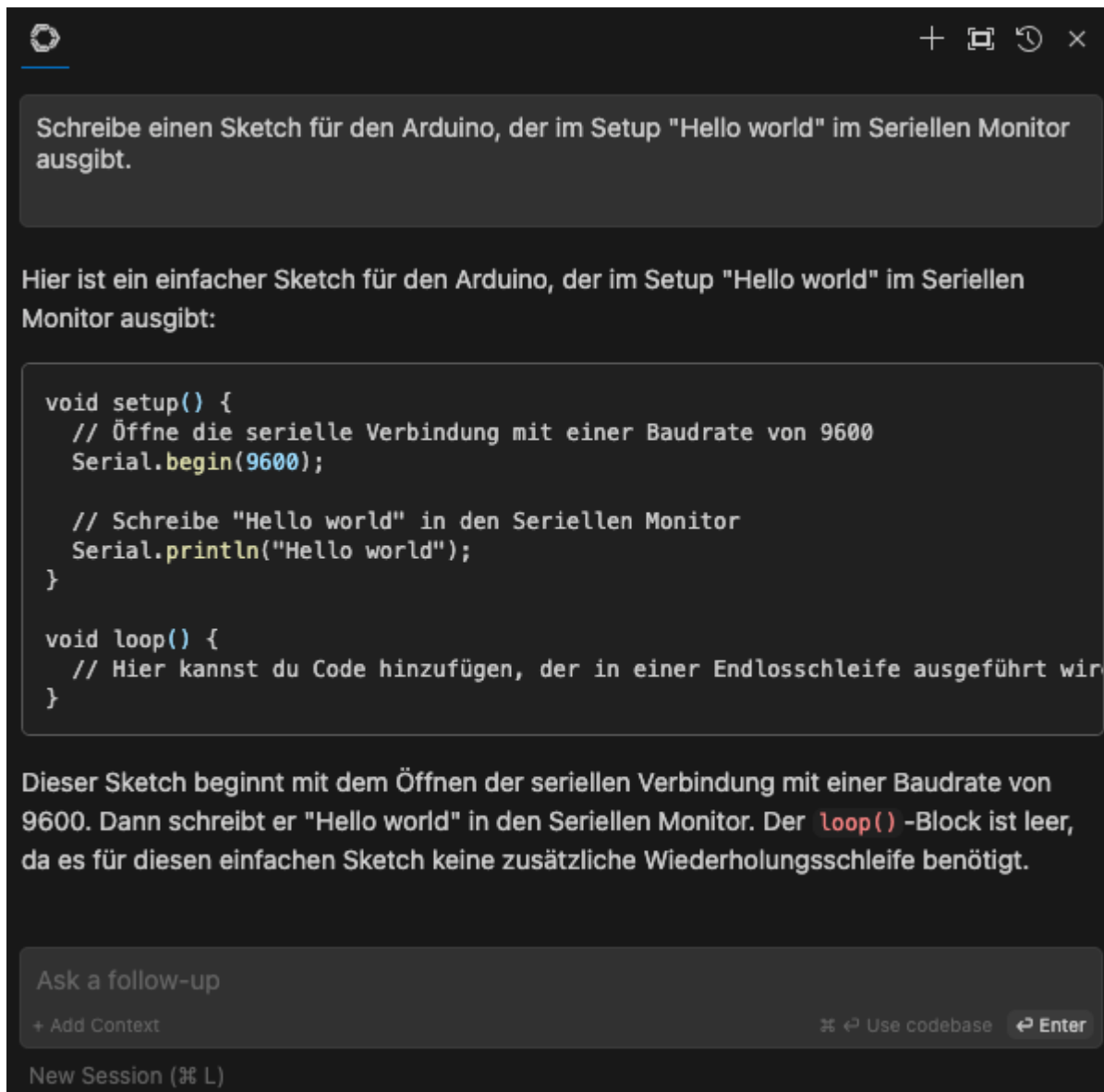
Kopiere anschließend den Key und trage ihn in Visual Studio

Code in das entsprechende Feld ein. Mit einem Klick auf **Done** gelangst du auf den letzten Screen der Einrichtung. Wähle dort **Free Trial (Codestral)** aus und melde dich mit deinem GitHub-Konto an. Falls du noch keines hast, kannst du es dir [kostenlos einrichten](#).

Codestral in Visual Studio Code verwenden

Das wäre nun also geschafft. Zeit für ein paar erste Tests. Für eine bessere Bedienbarkeit von Continue ziehe das Logo in VS Code vom Menü links in die rechte Sidebar. So hast du Zugriff auf den Chat, ohne dass du deinen Code im Hauptbereich verlassen musst.

Probiere es gleich aus, zum Beispiel mit dem Prompt, der auch schon in **Le Chat** zum Einsatz kam:



Das funktioniert schon mal. Aber das ist natürlich nicht alles: Du solltest einen Tab namens **continue_tutorial.py** sehen, das dir eine Einführung in das Plugin gibt.

Darin siehst du verschiedene Code-Abschnitte und Anweisungen, die dich mit wichtigen Tastenkombinationen und den Möglichkeiten von Continue und Codestral vertraut machen.

```

1 """
2 _____
3 _ / \ _____ / \ ____
4 _ / \ _ \ \ \ \ / \ _ \ \ \ \ \
5 _ / \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
6 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
7
8 Welcome to Continue! This is a 2 minute tutorial.
9
10 You can click the gray buttons, starting with "Highlight the function" to follow along.
11 """
12
13 # region ----- Part 1: Ask a question about code [Cmd+L] -----
14
15 Highlight the function
16 """Step 1: Highlight the function below"""
17
18
19 def mysterious_function(x):
20     for i in range(len(x)):
21         for j in range(len(x) - 1):
22             if x[j] > x[j + 1]:
23                 x[j], x[j + 1] = x[j + 1], x[j]
24
25     return x
26
27 Cmd+L
28 """Step 2: Use the keyboard shortcut [Cmd+L] to
29 select the code and toggle the Continue input box"""
30
31 "what does this code do?" | "what is an alternative to this?"
32 """Step 3: Ask a question and press Enter"""
33 # endregion

```

So kannst du zum Beispiel Code markieren (im Tutorial die abgebildete Funktion) und mit der Kombination **Cmd + L** (Mac) wieder den Bereich rechts öffnen und eine Frage dazu stellen – also zum Beispiel was mit ihr angestellt wird. Die Kombination **Cmd + I** ermöglicht es dir, Code nach deinen Wünschen ändern zu lassen.

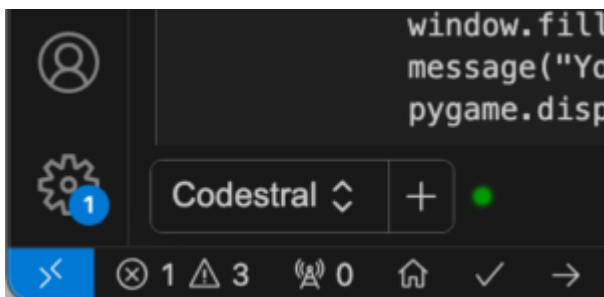
Noch praktischer ist das Debugging. Das Tutorial fordert dich im dritten Teil dazu auf, es auszuführen, was zu einem Fehler führt. Mit **Cmd + Shift + R** startest du die Fehlersuche, die du wieder im rechten Teil sehen kannst. Die verbesserte Funktion kannst du dann mit einem Klick in das Script einfügen – Problem gelöst.

Claude 3.5 Sonnet in VS Code einbinden

Neben Codestral eignet sich auch das LLM Claude 3.5 Sonnet von Anthropic hervorragend, um dir in Visual Studio Code zur Seite zu stehen. Um dieses Modell zu verwenden, kannst du ebenfalls Continue einsetzen – du musst lediglich deinen API Key von Anthropic und das gewünschte Modell hinterlegen.

Apropos API Key: Falls du noch keines hast, erstelle zunächst einen Account bei [Anthropic](#). Anschließend kannst du einen Key [hier erstellen](#). Hierfür benötigst du jedoch etwas Guthaben, **denn deine Kommunikation mit Claude 3.5 wird (wie bei der API von OpenAI) nach den verwendeten Tokens abgerechnet**. Das Guthaben von mindestens 5\$ kannst du in deinem Konto einzahlen. Auf der Webseite von Anthropic findest du die [aktuelle Preisliste für die API](#).

Um Claude 3.5 in Continue (und damit in VS Code) einzubinden, klicke unten links auf das Plus neben dem aktuell gewählten Modell:



Anschließend siehst du eine Auswahl von KI-Anbietern. Wähle dort **Anthropic** aus. Anschließend siehst du die folgende Eingabemaske:

← Configure Model

Anthropic

Requires API Key

To get started with Anthropic models, you first need to sign up for the open beta [here](#) to obtain an API key.

Enter required parameters

apiKey

► Advanced (optional)

Select a model preset

Claude 3 Opus

Anthropic's most capable model, beating GPT-4 on many benchmarks

Claude 3 Sonnet

The second most capable model in the Claude 3 series: ideal balance of intelligence and speed

Claude 3 Haiku

The third most capable model in the Claude 3 series: fastest and most compact model for near-instant responsiveness

OR choose from other providers / models by editing config.json.

Open config.json

Trage in das erste Feld deinen API Key von Anthropic ein und wähle darunter irgendein Modell. Aktuell (Juni 24) gibt es dort nur die Modelle der Version 3 zur Auswahl, das wird sich natürlich später noch ändern. **Solltest du keine Auswahl für das Modell Claude 3.5 Sonnet sehen, ist das kein Problem** – in der config.json wirst du gleich das richtige Modell manuell hinterlegen.

Um diese zu öffnen, klicke auf den Button unten. Dort siehst du nun das neu hinzugefügte Claude-Modell, dass du wie folgt anpassen musst:

```
"model": "claude-3-5-sonnet-20240620",  
"title": "Claude 3.5 Sonnet",  
"apiKey": "sk-[REDACTED]",  
"completionOptions": {},  
"provider": "anthropic"  
}
```

Hier noch einmal das momentan aktuelle Modell zum Herauskopieren:

claude-3-5-sonnet-20240620

Neben „title“ kannst du den korrekten Namen des Modells eintragen. Und das war es auch schon – wähle im Auswahl-Dropdown von Continue das neue Modell aus und probiere Claude 3.5 Sonnet aus.

Fazit

Codestral und Claude 3.5 Sonnet bieten eine wirkliche Alternative zum GitHub Copilot – sowohl in Sachen Funktionalität, Zuverlässigkeit als auch im Preis. In der Testphase von Codestral sowieso, aber auch danach und für

Claude 3.5 sollten die Gebühren beim hobbymäßigen Gebrauch unterhalb der monatlichen 10\$ von GitHub liegen. Nähere Informationen zu den Kosten erhältst du in der [Preisliste von Mistral](#) und jener von [Anthropic](#).