

Kochen mit ChatGPT und REWE-Angeboten



Dass ChatGPT mittlerweile mit einem Bild deines Kühlschranks kochen kann, ist bekannt. Aber wie wäre es einmal mit einem Gericht, dessen Zutatenliste nur aus Sonderangeboten deines nächstgelegenen REWE-Markts besteht? In diesem Projekt probierst du genau das aus: **Du fragst die aktuellen REWE-Angebote per API ab und lässt ChatGPT daraus Rezeptideen erstellen** – Kochen mit ChatGPT.

Zum Einsatz kommen hierbei zwei Python-Scripts – das erste, um die ID deines Wunsch-Supermarkts herauszufinden und die zweite für die Abfrage der Angebote und die Erstellung der Rezepte. Wenn du möchtest, kannst du letzteres auch automatisieren: Auf deinem Raspberry Pi könnte ein Cronjob jeden Montag die aktuellen Angebote abrufen und ChatGPT dir daraus sieben Rezepte für die ganze Woche erstellen.

Noch ein Hinweis vorab: Bei diesem Projekt handelt es sich natürlich um eine Spielerei – ob die von ChatGPT vorgeschlagenen Gerichte wirklich schmecken, ist leider nicht so ganz sicher. **Aber bei den ersten Tests kam schon ansprechende Speisen heraus, z.B. Süßkartoffel-Kürbis-Auflauf, Mediterrane Hähnchenschenkel mit frischen Kräutern und Zitronen-Butter oder Rouladen nach Landhaus-Art.** Sowohl die Zutatenliste auch die Anweisungen zur Zubereitungen klangen nicht abwegig. Ein Versuch ist es allemal Wert – und dieser Versuch erweitert auf jeden Fall deine Fähigkeiten.

Finde die ID deines REWE-Markts heraus

Bevor du die Angebote des REWE-Markts deiner Wahl abrufen kannst, benötigst du den **marketCode** dieses Markts. Den kannst du ganz leicht mit einem kleinen Python-Script herausfinden:

```
import requests
import json

# Trage hier deinen Standort ein
search_term = 'Karlsruhe'
url = f'https://www.rewe.de/api/marketsearch?searchTerm={search_term}'

# Stelle die Anfrage an die API.
response = requests.get(url)

# Überprüfe, ob die Anfrage erfolgreich war (HTTP Status Code 200).
if response.status_code == 200:
    # Parse die Antwort als JSON.
    data = response.json()

    # Schön formatierte Ausgabe des JSON.
    print(json.dumps(data, indent=4))
else:
    print('Fehler bei der Anfrage:', response.status_code)
```

Um die Märkte an deinem Standort zu finden, trage diesen in die Variable **search_term** ein – in meinem Fall ist das Karlsruhe. Wenn du das Script nun ausführst, erhältst du die Antwort als JSON. Hier die ersten beiden Einträge einer möglichen Antwort der API:

```
{
    "wwIdent": "831008",
    "isReweDortmund": false,
```

```

        "companyName": "REWE Ponzer GmbH & Co. oHG",
        "contactStreet": "Hans - Sachs - Str. 8",
        "contactZipCode": "76133",
        "contactCity": "Karlsruhe",
        "marketHeadline": "REWE Markt",
        "openingInfo": {
            "willOpen": {
                "onDay": 5,
                "at": "07:00"
            }
        }
    },
    {
        "wwIdent": "840913",
        "isReweDortmund": false,
        "companyName": "REWE Christopher Lannert oHG",
        "contactStreet": "Josef-Schofer-Str. 14",
        "contactZipCode": "76187",
        "contactCity": "Karlsruhe",
        "marketHeadline": "REWE Markt",
        "openingInfo": {
            "willOpen": {
                "onDay": 5,
                "at": "07:00"
            }
        }
    }
},

```

Gleich der jeweils erste Key **wwIdent** ist die ID bzw. marketCode, den du später benötigst. Suche dir also den gewünschten Markt aus der Liste heraus und notiere dir die ID.

Aktuelle Angebote finden

Bevor es mit dem nächsten Python-Script losgehen kann, musst du noch die Bibliothek **cloudscraper** installieren. Diese benötigst du, das die API von REWE eigentlich keine automatisierten Abfragen zulässt. Mit dieser Bibliothek kannst du diese Sperre allerdings umgehen. Achte deshalb bitte

darauf, keinen „Unfug“ anzustellen und z.B. zahllose Anfragen an die API zu stellen.

Um die Bibliothek zu installieren, gib im Terminal bzw. in der Konsole folgenden Befehl ein:

```
pip install cloudscraper
```

Außerdem benötigst du noch die Bibliothek **openai**, mit der du die API von ChatGPT ansteuern kannst. Diese installierst du wie folgt:

```
pip install openai
```

Nun kann es mit dem Script losgehen. Zunächst bindest du die beiden genannten Bibliotheken ein und hinterlegst deine gefundene ID sowie die URL für die Abfrage:

```
import cloudscraper
from openai import OpenAI

# Deine ID, hier ein Beispiel aus Karlsruhe
market_id = '831008'
# Die Basis-URL der API
url =
f'https://mobile-api.rewe.de/api/v3/all-offers?marketCode={market_id}'
```

Anschließend bereitest du **cloudscraper** vor, erstellst einen leeren String, in den die gefundenen Sonderangebote geschrieben werden – und erstellst eine Liste der Angebotskategorien, die überhaupt für Rezepte in Frage kommen. Es gibt natürlich auch Angebote in den Kategorien Haushalt, Tierfutter, Spirituosen etc., mit denen wir aber hier nichts anfangen können.

```
# Erstellen ein Cloudscraper-Objekt
scraper = cloudscraper.create_scraper()
```

```

# Stelle die GET-Anfrage
response = scraper.get(url)

# Initialisiere einen leeren String für die Titel der Angebote
titles_string = ''

# Eine Liste der gewünschten Kategorietitel
desired_categories = {
    'Frische & Kühlung',
    'Süßes & Salziges',
    'Obst & Gemüse',
    'Nahrungsmittel',
    'Tiefkühl'
}

```

Nun fehlt nur noch die eigentliche Abfrage. Hier stellst du sicher, dass in der Variablen **titles** (für die Namen der angebotenen Artikel) nur jene aus den vorgegebenen Kategorien gespeichert werden.

```

# Überprüfe den Statuscode der Antwort
if response.status_code == 200:
    # Wenn der Statuscode 200 ist, war die Anfrage erfolgreich
    # Konvertiere die Antwort in JSON
    data = response.json()
    # Extrahiere die Angebote, die den gewünschten
    Kategorietiteln entsprechen
    titles = [
        offer['title']
        for category in data['categories']
        for offer in category['offers']
        if 'rawValues' in offer and 'categoryTitle' in
offer['rawValues']
        and offer['rawValues']['categoryTitle'] in
desired_categories
    ]
    # Erstelle einen String mit den gefilterten Titeln,
    getrennt durch Kommas
    titles_string = ', '.join(titles)
    # Ausgabe des Strings

```

```
    print("Gefilterte Titel:", titles_string)
else:
    # Wenn der Statuscode nicht 200 ist, gab es ein Problem
    mit der Anfrage
    print(f'Fehler: {response.status_code}')
```

Rezepte erstellen lassen mit ChatGPT

Nun folgt der zweite Teil des Scripts. Hier fragst du ChatGPT über die API nach Rezeptideen, die auf den gefundenen Sonderangeboten basieren. Hierfür benötigst du einen API-Key von OpenAI. [In diesem Tutorial erfährst du, wie du dir den benötigten Key erstellst.](#)

Du hinterlegst im Script also deinen Key und erstellst gleich danach die Abfrage:

```
client = OpenAI(
    api_key="DEIN API-KEY",
)

print("Ich erstelle Gerichte...")
completion = client.chat.completions.create(
    #model="gpt-3.5-turbo",
    model="gpt-4",
    messages=[
        {"role": "system", "content": "Du bist ein KI-Koch, der aus vorgegebenen Zutaten Rezepte für Gerichte entwirft."},
        {"role": "user", "content": "Es folgt eine Liste von Zutaten. Kreiere hieraus 3 Rezepte für leckere Gerichte. Gehe davon aus, dass die Küchen-Basics wie Salz, Pfeffer, Öl, Butter etc. bereits vorhanden sind. Hier die Liste: {}".format(titles_string)}]
    )

print(completion.choices[0].message.content)
```

Du findest im obigen Code zwei Zeilen, mit denen du auswählen kannst, ob du das Modell 3.5 oder die neuere Version 4 verwenden möchtest. **Kommentiere einfach die nicht gewünschte Version aus.**

Als Anweisung (system) an ChatGPT legst du fest, dass die KI als Ideengeber für Rezepte fungieren soll. Mit dem eigentlichen Prompt (user) legst du dann fest, was du möchtest. Im obigen Beispiel sind es drei Rezepte, für die übliche Zutaten wie Salz und Pfeffer enthalten sein dürfen, auch wenn sie nicht in den Angeboten zu finden sind. An diesen Prompt hängst du dann einfach die Liste der Angebote an.

Hier hast du natürlich einiges an Spielraum. So könntest du z.B. nur vegetarische Gerichte entwerfen lassen und ChatGPT Unverträglichkeiten mitteilen.

Die Antwort der künstlichen Intelligenz sieht dann beispielsweise folgendermaßen aus:

Rezept 1: Rinderrouladen mit Apfel-Rotkohl und Maronen

Zutaten:

- Rinder-Rouladen
- Petersilie
- Rote Tafeläpfel
- REWE Beste Wahl Apfelrotkohl
- REWE Bio Gekochte Maronen

1. Die Rinderrouladen flach ausbreiten, mit Salz und Pfeffer würzen und mit gehackter Petersilie bestreuen. Die Rouladen aufrollen und mit Küchengarn befestigen.
2. Die Rouladen in einem großen Topf mit Öl scharf anbraten und anschließend beiseite stellen.
3. Die Äpfel schälen, vierteln, entkernen und klein schneiden. Zusammen mit dem Apfelrotkohl in den Topf geben und ca. 5 Minuten dünsten.
4. Die Rinderrouladen zurück in den Topf geben und bei niedriger Hitze 60 Minuten köcheln lassen.

5. 10 Minuten vor Ende der Kochzeit die gekochten Maronen hinzufügen.
6. Abschmecken und servieren.

Klingt doch halbwegs plausibel, oder? Eine Beilage wäre vielleicht noch gut und dass der Rotkohl in den gleichen Topf wie die Rouladen soll, ist auch nicht ganz einleuchtend. **Auch wenn du die Rezepte nicht 1:1 übernehmen solltest, erhältst du mit diesem Projekt recht brauchbare Ideen für Gerichte, die deinen Geldbeutel schonen.** Ich wünsche dir jedenfalls schon einmal guten Appetit!

Hier nun das vollständige Python-Script:

```
import cloudscraper
from openai import OpenAI

# # Deine ID, hier ein Beispiel aus Karlsruhe
market_id = '831008'
# Die Basis-URL der API
url =
f'https://mobile-api.rewe.de/api/v3/all-offers?marketCode={market_id}'

# Erstellen ein Cloudscraper-Objekt
scraper = cloudscraper.create_scraper()

# Stelle die GET-Anfrage
response = scraper.get(url)

# Initialisiere einen leeren String für die Titel der Angebote
titles_string = ''

# Eine Liste der gewünschten Kategorietitel
desired_categories = {
    'Frische & Kühlung',
    'Süßes & Salziges',
    'Obst & Gemüse',
    'Nahrungsmittel',
    'Tiefkühl'
```



```

}

# Überprüfe den Statuscode der Antwort
if response.status_code == 200:
    # Wenn der Statuscode 200 ist, war die Anfrage erfolgreich
    # Konvertiere die Antwort in JSON
    data = response.json()
    # Extrahiere die Angebote, die den gewünschten
    Kategorietiteln entsprechen
    titles = [
        offer['title']
        for category in data['categories']
        for offer in category['offers']
        if 'rawValues' in offer and 'categoryTitle' in
offer['rawValues']
        and offer['rawValues']['categoryTitle'] in
desired_categories
    ]
    # Erstelle einen String mit den gefilterten Titeln,
    getrennt durch Kommas
    titles_string = ', '.join(titles)
    # Ausgabe des Strings
    print("Gefilterte Titel:", titles_string)
else:
    # Wenn der Statuscode nicht 200 ist, gab es ein Problem
    mit der Anfrage
    print(f'Fehler: {response.status_code}')

# Mit ChatGPT Rezepte erstellen
client = OpenAI(
    api_key="DEIN API-KEY",
)

print("Ich erstelle Gerichte...")
completion = client.chat.completions.create(
    #model="gpt-3.5-turbo",
    model="gpt-4",
    messages=[
        {"role": "system", "content": "Du bist ein KI-Koch, der aus
vorgegebenen Zutaten Rezepte für Gerichte entwirft."},

```

```
{"role": "user", "content": "Es folgt eine Liste von Zutaten.  
Kreiere hieraus 3 Rezepte für leckere Gerichte. Gehe davon  
aus, dass die Küchen-Basics wie Salz, Pfeffer, Öl, Butter etc.  
bereits vorhanden sind. Hier die Liste:  
{}}".format(titles_string)]  
)
```

```
print(completion.choices[0].message.content)
```