

# Programmieren mit Python

## Teil 10: APIs

---

Dr. Aaron Kunert

*[aaron.kunert@salemkolleg.de](mailto:aaron.kunert@salemkolleg.de)*

07. Dezember 2022

# Web APIs

Wie Python im Netz surfen kann

---

## Definition: API

Eine *API* (Application Programming Interface) ist im Allgemeinen eine klar definierte Schnittstelle zwischen Programmen.

Im Kontext des Internets geht es dabei meist um Webanwendungen, die statt einer für menschen lesbaren Seite, ein maschinenlesbares JSON ausliefern. Auf diese Weise ist ein Datenaustausch zwischen Deinem Programm und einer Webanwendung in beide Richtungen möglich.

Eine Web-API besteht meist aus einer (oder mehreren) Webadressen und einer Anleitung, wie sie zu benutzen ist.

## Beispiele für APIs

- **Instagram Api:** Erhalte Infos, wer Dir folgt und welche Reaktionen Du auf Deine Bilder erhältst.
- **GoogleMaps Api:** Sende Koordinaten an Google-Maps und erhalte die Adresse.
- **REST Countries:** Erhalte Daten von Ländern dieser Erde.
- **Stock Market Apis:** Erhalte live Daten zu Börsenkursen.

## Anwendungsbeispiele

- **Instagram:** Schreibe Dir einen Python-Scheduler, wann welche Bilder von Dir veröffentlicht werden sollen.
- **REST Countries:** Nutze die Daten, um ein Quiz zu schreiben.
- **Stock Market Apis:** Trading Apps, SMS-Warnung bei einbrechenden Preisen.

## Wie macht man einen API-Call?

Im Folgenden gehen wir durch die Schritte, die man ausführen muss, um einen lesenden API-Call auszuführen. Wir verwenden dafür die **REST Countries Api**. Die Dokumentation (*Docs*) der Api findet sich unter <https://restcountries.com/>.

## Schritt 1: Finde die richtige URL (d.h. Webadresse)

Zunächst benötigt man eine URL (den sogenannte *Endpoint*). Diese kann völlig statisch sein, oftmals aber sind Teile der URL dynamisch bzw. hängen von Deiner genauen Anfrage ab.

Beispiel:

`https://restcountries.com/v3.1/all`

oder

`https://restcountries.com/v3.1/name/france`

## Schritt 2: Schicke einen Request an den Endpoint

Verwende folgenden Code, um einen Request zu schicken:

---

```
import requests
...

response = requests.get("https://restcountries.com/v3.1/name/france")
data = response.json()
# print(data)
```

---

### Schritt 3: Erweitere den Request ggf. um *Query-Parameter*

---

```
import requests
...

payload = {"fields": ["capital","population","continents"]}
response = requests.get("https://restcountries.com/v3.1/name/france", params=payload)
data = response.json()
# print(data)
```

---



## Schritt 4: Transformiere, filtere und speichere die Daten je nach Bedarf

---

```
import requests
import json
...

payload = {"fields": ["capital","population","continents"]}
response = requests.get("https://restcountries.com/v3.1/name/france", params=payload)
data = response.json()

with open("france.json","w") as my_file:
    json.dump(data, my_file)
```

---

## Daten eines Landes extrahieren

Stelle einen Request zu Italien an die Countries-API. Bereite die zurückgegebenen Daten so auf, dass folgendes auf der Konsole erscheint:

Hauptstadt: Rome

Bevölkerung: 59554023

Kontinent: Europe

## Daten eines Landes extrahieren

---

```
import requests

response = requests.get("https://restcountries.com/v3.1/name/italy")
data = response.json()

data = data[0]
capital = data["capital"][0]
population = data["population"]
continent = data["continents"][0]

print(f"Hauptstadt: {capital}")
print(f"Bevölkerung: {population}")
print(f"Kontinent: {continent}")
```

---

## Hauptstadt-Orakel

Schreibe eine Funktion, die den (englischsprachigen) Namen eines Landes erwartet und den Namen der Hauptstadt zurückgibt.

## Hauptstadt-Orakel

---

```
import requests

def get_capital(country):
    response = requests.get("https://restcountries.com/v3.1/name/" + country)
    data = response.json()
    data = data[0]
    capital = data["capital"][0]
    return capital
```

---

## Datenaufbereitung

Erstelle eine Liste aller Länder. Jedes Land soll dabei als Dictionary der folgenden Form abgespeichert werden:

---

```
{  
  "name": "Italy",  
  "capital": "Rome",  
  "continent": "Europe",  
  "population": 59554023  
}
```

---

Speichere diese Liste als JSON in der Datei `countries.json`.

## Datenaufbereitung

---

```
import requests
import json

response = requests.get("https://restcountries.com/v3.1/all")
data = response.json()

countries = []
for country in data:
    if "capital" in country.keys():
        countries.append({
            "name": country["name"]["common"],
            "capital": country["capital"][0],
            "population": country["population"],
            "continent": country["continents"][0]
        })

with open("countries.json", "w") as my_file:
    json.dump(countries, my_file, indent=4)
```

---