

Praktikum CPS

Versuch 3: „Wetterdaten Extraktion“

1. Microsoft Azure: Portal and DevOps	3
2. Open-Weather-Map	4
3. Vorbereitung von ETL Pipeline	7
4. Wetter Daten Extraktion	8

1. Microsoft Azure: Portal and DevOps

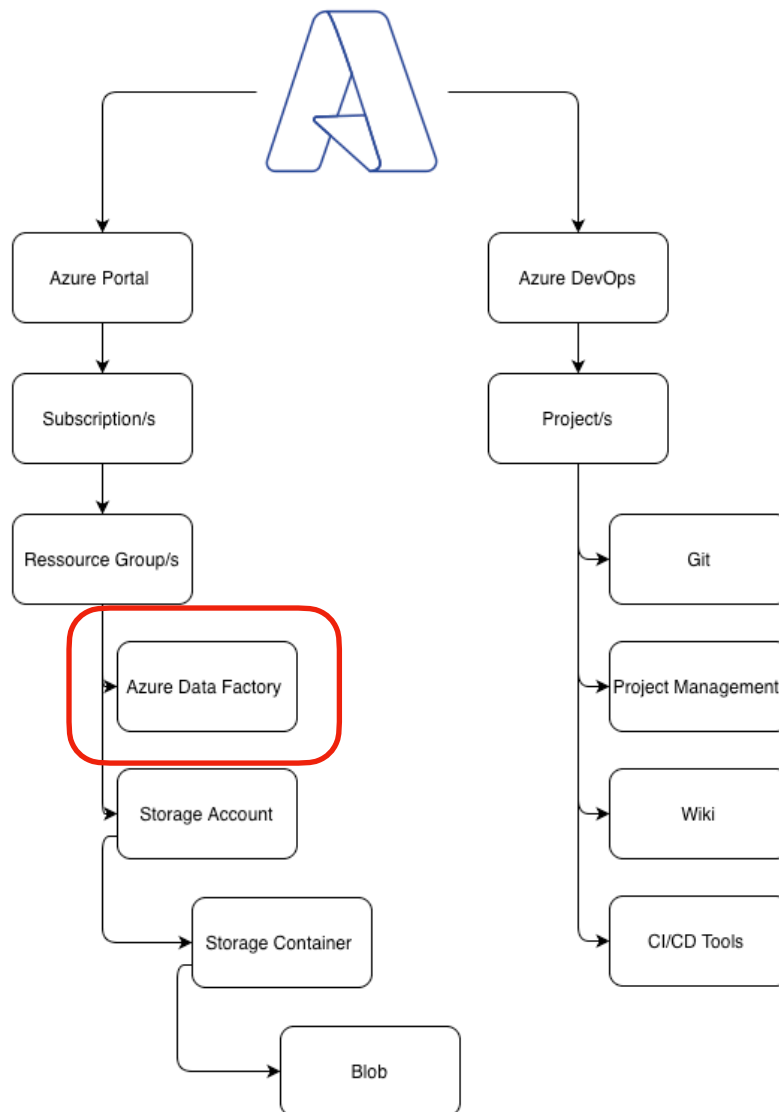


Abbildung 1 - Azure Cloud structure

2. Open-Weather-Map

Open-Weather-Map ist ein webbasierter Dienst, der Wetterdaten über API-Abfragen zur Verfügung stellt. Dies ermöglicht es Entwicklern und Anwendungen, meteorologische Informationen wie aktuelle Wetterbedingungen, Vorhersagen, Temperaturen, Niederschlagsdaten und anderen möglicherweise interessante Daten abzurufen.

2. Erstellen Sie zuerst ein Konto auf <https://openweathermap.org/>.

2. Unter „**My API keys**“ können Sie einen API-Schlüssel generieren.

Ein API Schlüssel ist eine Art von Authentifizierungsmethode, die verwendet wird, um den Zugriff auf bestimmte APIs(**Application Programming Interfaces**) zu steuern. APIs sind Schnittstellen, sie es Anwendungen zu ermöglichen, miteinander zu kommunizieren und Daten auszutauschen.

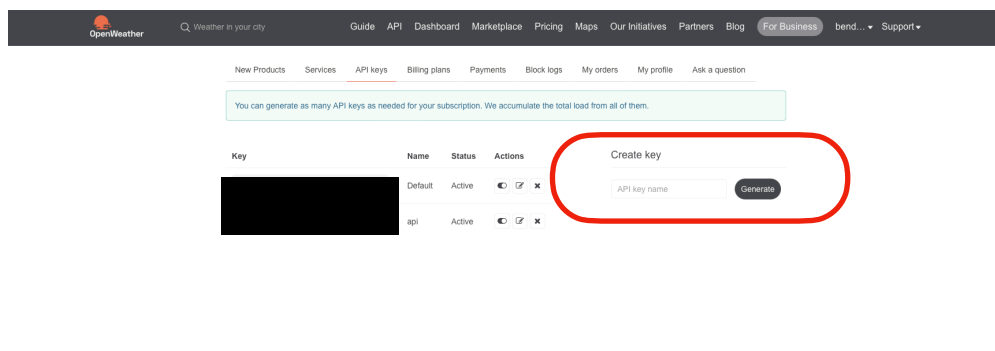


Abbildung 2 - API Key

In unserem Fall werden wir die Wetterdaten abfragen.

3. Suchen Sie nach dem „**Current Weather Data**“ und klicken Sie auf „**API doc**“.

Dort finden Sie, wie ein API-Abfrage aussehen muss.

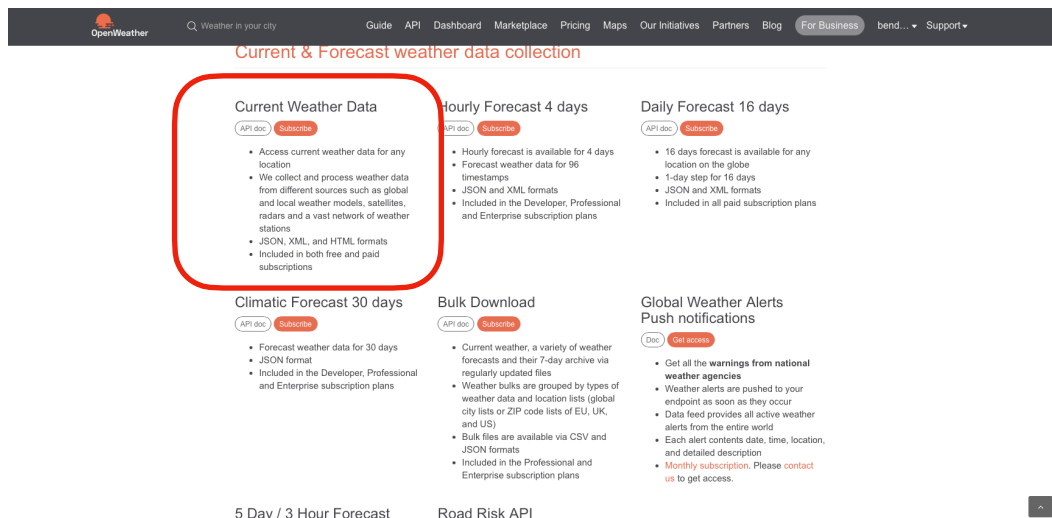


Abbildung 3 - Open-Weather-Map

<https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}>

An dieser Stelle müssen drei Parameter Ihrer API-Abfrage angepasst werden: **lat**, **lon** und **API key**.

Lat und **lon** stehen für latitude and longitude. Ganz allgemein: Breiten- und Längengrade sind geografische Koordinaten, die verwendet werden, um einen bestimmten Punkt auf der Erdoberfläche zu bestimmen.

Sie können einfach googeln, um die Koordinaten für Bochum zu finden und Ihre API key haben Sie bereits generiert.

```
https://api.openweathermap.org/data/2.5/
weather?
lat=*.*****&lon=*.*****&appid=*****
*****
```

Ob Ihre API -Abfrage funktioniert, können Sie ganz einfach überprüfen, indem Sie sie in die Adressleiste Ihres Browsers eingeben und Enter drücken. Als erfolgreiche Abfrage erhalten Sie folgendes JSON-Objekt zurück:

```
{
  "coord": {
    "lon": 7.2222,
    "lat": 51.4817
  },
  "weather": [
    {
      "id": 801,
      "main": "Clouds",
      "description": "few clouds",
      "icon": "02d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 293.24,
    "feels_like": 292.76,
    "temp_min": 291.4,
    "temp_max": 294.64,
    "pressure": 1021,
    "humidity": 55,
    "visibility": 10000,
    "wind": {
      "speed": 1.79,
      "deg": 247,
      "gust": 4.47
    },
    "clouds": {
      "all": 12
    },
    "dt": 1695559524,
    "sys": {
      "type": 2,
      "id": 2035566,
      "country": "DE",
      "sunrise": 1695532779,
      "sunset": 1695576418,
      "timezone": 7200,
      "id": 2947416,
      "name": "Rochum",
      "cod": 200
    }
  }
}
```

Abbildung 4 - JSON File als Antwort auf API Request

3. Vorbereitung von ETL Pipeline

1. Erstellen Sie einen neuen Linked Service und wählen Sie als Data store **,REST‘**(steht für Rest API Abfrage) aus.
2. Als Namen geben Sie **,LS_REST‘** ein
3. Als Base URL verwenden Sie API Request, die Sie in vorherigen Schritt herausgefunden haben.
4. Als User name und Password verwenden Sie Ihr Passwort und Username von Ihrem Open-Weather-Map Konto.
5. Überprüfen Sie, ob die Verbindung funktioniert.
6. Erstellen Sie auch ein neuen Dataset und als Data store wählen Sie wieder **,REST‘** und halten Sie sich an die Namenskonvention, indem Sie es **,DS_REST‘** nennen.
7. Erstellen Sie einen weiteren Dataset **DS_BS_Sink_JSON**.
 - 7.1 Als Data store wählen Sie Blob storage und als format data type wählen Sie JSON.
 - 7.2 In File Path geben Sie als Container **,sinkcontcps‘** an.

4. Wetter Daten Extraktion

1. Erstellen Sie eine neue Pipeline mit Copy data activities und geben Sie ihr den Namen **PL_DS_REST_Wetterdaten_to_DS_BS**
2. Als Source dataset wählen Sie **DS_REST**. An dieser Stelle kann wieder geprüft werden, ob soweit alles funktioniert, indem Sie auf **„Preview data“** klicken. Die Antwort sollte folgendermaßen aussehen:

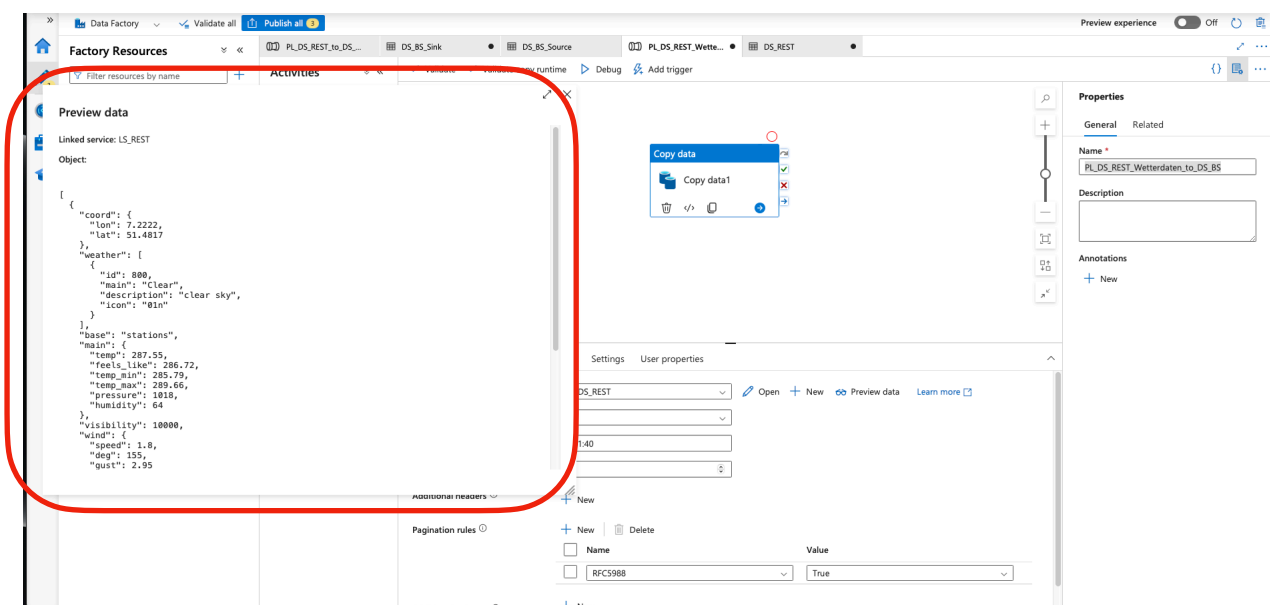


Abbildung 5 - Preview data

3. Als Sink dataset wählen Sie **DS_BS_Sink** aus
4. Alle Änderungen sollen validiert und gepublisht werden
5. Führen Sie die Pipeline aus.
6. In Portal gehen Sie in Sink-Container und überprüfen, ob da die JSON-Datei dort gespeichert ist.

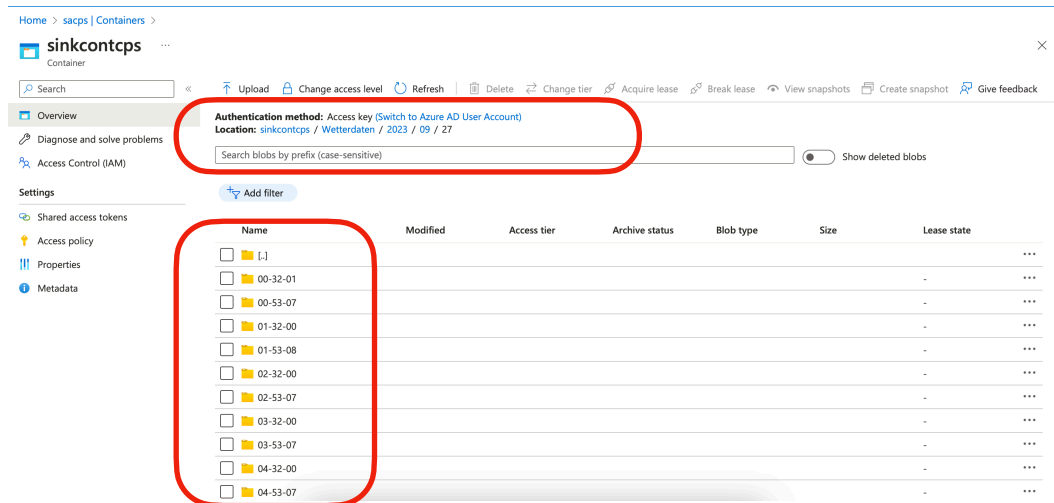


Abbildung 6 - Pfadstruktur

Beachten Sie folgendes:

- Namenskonvention
- Pfad.

Der Pfad sollte folgendermaßen aussehen:

„Containername/Ordnername/yyyy/MM/dd/hh-mm-ss „

Als Beispiel:

„sinkcontcps/Wetterdaten/2023/09/27/00-32-01/(JSON Datei)“

- Gehen Sie zur Author-Page und klicken Sie auf „**Trigger**“ -> „New/Edit“->“Choose trigger“->“New“
- Erstellen Sie einen neuen Trigger.
 - Name - **TR_HOURLY**
 - Type - **ScheduleTrigger**
 - Time zone - **UTC+1**
 - Recurrence - **Every 1 hour**
- Setzen Sie diesen Trigger für die Pipeline „**PL_DS_REST_Wetterdaten_to_DS_BS**“. Die Änderungen sollten gespeichert, validiert und gepubliziert werden.
- Stellen Sie sicher, dass Pipeline stündlich ausgeführt wird und die Daten ständig extrahiert werden. Für den nächsten Versuch werden Sie die Daten benötigen.

Interessante Links:

- <https://learn.microsoft.com/en-us/azure/data-factory/connector-rest?tabs=data-factory>
- <https://learn.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers>