

Datenanalyse umwelt.info: Von Rohdaten zu Projekten

github.com/tifa365/mvp_umwelt_info_exploration

Tim Fangmeyer
[linkedin/today/author/timfangmeyer](https://linkedin.com/in/today/author/timfangmeyer)

umwelt.info

Stellt die Daten sowohl über die Suche ...



... als auch über eine Schnittstelle bereit.

Web-UI vs. API-Zugriff

Web-Oberfläche ([umwelt.info](#))

- Intuitives Erkunden und Filtern der Daten
- ABER: “Händische” Arbeit nötig, nicht automatisierbar

API-Zugriff

- Automatisierung möglich (Scripting, regelmäßige Updates)
- Verarbeitung großer Datenmengen
- ABER: Technisches Know-how erforderlich

Wir sind hier: Metadaten-Exploration

1. Katalog durchsuchen
2. Metadaten abrufen
3. Eigentliche Daten herunterladen
4. Daten reinigen, aggregieren, analysieren, visualisieren, etc.

Wichtig: Die folgenden API-Beispiele zeigen nur die *Suche und Beschreibung* der Daten.

API

Programmierschnittstellen (APIs) ermöglichen Dritten den Zugang zu vorher verschlossenen

The screenshot shows the homepage of the umwelt.info website. At the top right, there are links for "Leichte Sprache", "Barriere melden", and "Open CoDE". Below the header, there is a navigation bar with categories: Luft, Wasser, Boden, Lebewesen & Lebensräume, Mensch & Umwelt, Neu hier?, and zur Suche. A red arrow labeled '1' points to the "zur Suche" button. Below the navigation bar is a search bar with the placeholder "Ich suche...". To the right of the search bar is a green magnifying glass icon. Underneath the search bar, there is a section titled "Filter" and "638,373 Ergebnisse". On the left, there is a map of Northern Europe with various regions labeled. On the right, there is a detailed description of a WMS service: "WMS MSRL: D8-Schadstoffe (sh-llur), Mittelwert 2005-2010". It includes information about the service's scope, last update (7.7.2023), license (d1-by-de/2.0), source (Land\Schleswig-Holstein\Umweltportal), and keywords (Chrom, Messstation, Nickel, Kupfer, Arsen). A red arrow labeled '2' points to the "API" link at the top right of this section.

Datenpools.

Swagger-UI

Interaktive Dokumentation für REST-APIs. Mit Swagger UI können Entwickler die verfügbaren API-Endpunkte direkt im Browser erkunden.

[swagger-ui/](#)

Umfang der Schnittstelle

Wichtigste Endpunkte:

- **Volltextsuche** über alle Datensätze (+ Filteroptionen)

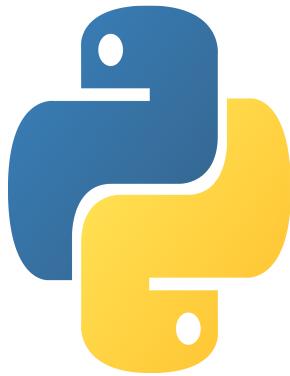
- **Einzeldatensatz-Abruf** detaillierter Metadaten
- **Statistik-Endpunkt** für Nutzungsstatistiken

API: Wichtige CKAN-Endpunkte für die Abfrage von umwelt.info

- `/package_search` Volltextsuche nach Suchbegriff. Die Wildcard "*" ermöglicht die Abfrage des gesamten Metadatenbestandes
- `/package_show` Ermöglicht die Abfrage einzelner Datensätze im CKAN Format.
- `/counts/now`: Aktuelle Gesamtzahl an Datensätzen.

API Beispiele mit Python

Beispielhafte Abfrage der CKAN-API mit Python.



API Endpoint: /counts/now

```
import requests

url = "https://md.umwelt.info/counts/now"
response = requests.get(url)
data = response.json()

# Ausgabe der Antwort
print(data)
```

```
# Beispiel-Output:  
{  
    "datasets": 638373,  
    "sources": 129,  
    "providers": 45,  
    "failed_harvests": 0,  
    "errors": 900  
}
```

API Endpoint: /package_search

Request

```
import requests  
  
# API-Parameter  
base_url = "https://md.umwelt.info/ckan/api/3/action/package_search"  
params = {  
    "q": "Grundwasserpegel", # Suchbegriff  
    "rows": 1                # Anzahl der Ergebnisse  
}  
  
# API-Abfrage durchführen  
response = requests.get(base_url, params=params)  
data = response.json()
```

API Endpoint: /package_search

Result

```
# Beispiel-Output:  
{  
    "help": "/api/3/action/help_show?name=package_search",  
    "success": true,  
    "result": {  
        "count": 81765,  
        "facets": [],  
        "search_facets": [],  
        "sort": "score desc",  
        "results": [  
            {  
                "id": "59444444444444444444444444444444",  
                "name": "Grundwasserpegel",  
                "url": "http://www.umwelt.de/.../Grundwasserpegel",  
                "type": "dataset",  
                "modified": "2018-01-12T12:00:00Z",  
                "created": "2018-01-12T12:00:00Z",  
                "title": "Grundwasserpegel",  
                "description": "Hier finden Sie die Grundwasserpegel der Deutschen Umwelthilfe.",  
                "notes": "Hier finden Sie die Grundwasserpegel der Deutschen Umwelthilfe.",  
                "tags": [{"name": "Grundwasserpegel"}, {"name": "Deutsche Umwelthilfe"}],  
                "resources": [{"name": "Grundwasserpegel", "url": "http://www.umwelt.de/.../Grundwasserpegel", "format": "CSV", "size": 123456789}, {"name": "Grundwasserpegel", "url": "http://www.umwelt.de/.../Grundwasserpegel", "format": "JSON", "size": 123456789}],  
                "extras": {"id": "59444444444444444444444444444444", "name": "Grundwasserpegel", "url": "http://www.umwelt.de/.../Grundwasserpegel", "type": "dataset", "modified": "2018-01-12T12:00:00Z", "created": "2018-01-12T12:00:00Z", "title": "Grundwasserpegel", "description": "Hier finden Sie die Grundwasserpegel der Deutschen Umwelthilfe.", "notes": "Hier finden Sie die Grundwasserpegel der Deutschen Umwelthilfe.", "tags": [{"name": "Grundwasserpegel"}, {"name": "Deutsche Umwelthilfe"}], "resources": [{"name": "Grundwasserpegel", "url": "http://www.umwelt.de/.../Grundwasserpegel", "format": "CSV", "size": 123456789}, {"name": "Grundwasserpegel", "url": "http://www.umwelt.de/.../Grundwasserpegel", "format": "JSON", "size": 123456789}]}  
        ]  
    }  
}
```

```
{
    "id": "**Z292ZGF0YS8xZjM2ZGRhYS0xZGM4LTQ4YWYtOWEyMS1iNmVjZmFlZjExNzk=**",
    "name": "govdata/1f36ddaa-1dc8-48af-9a21-b6ecfaef1179",
    "title": "Grundwasserpegelmessung in Stadt und Landkreis in Osnabrück",
    "private": false,
    "license_url": "https://www.govdata.de/dl-de/by-2-0",
    "license_title": "dl-by-de/2.0"
    ...
}
]
}
```

API Endpoint: /package_show

Request

```
import requests

# API-Parameter
base_url = "https://md.umwelt.info/ckan/api/3/action/package_show"
dataset_id = "Z292ZGF0YS8xZjM2ZGRhYS0xZGM4LTQ4YWYtOWEyMS1iNmVjZmFlZjExNzk="
params = {"id": dataset_id}

# API-Abfrage durchführen
response = requests.get(base_url, params=params)
data = response.json()
```

API Endpoint: /package_show

Result

```
# Beispiel-Output:
{
    "help": "/api/3/action/help_show?name=package_show",
    "success": true,
    "result": {
        "id": "Z292ZGF0YS8xZjM2ZGRhYS0xZGM4LTQ4YWYtOWEyMS1iNmVjZmFlZjExNzk=",
        "name": "govdata/1f36ddaa-1dc8-48af-9a21-b6ecfaef1179",
        "title": "Grundwasserpegelmessung in Stadt und Landkreis in Osnabrück",
```

```

    "private": false,
    "license_url": "https://www.govdata.de/dl-de/by-2-0",
    "license_title": "dl-by-de/2.0",
    "notes": "An mehreren Stellen in Stadt und Landkreis Osnabrück werden Grundwasserpegel
  }
}

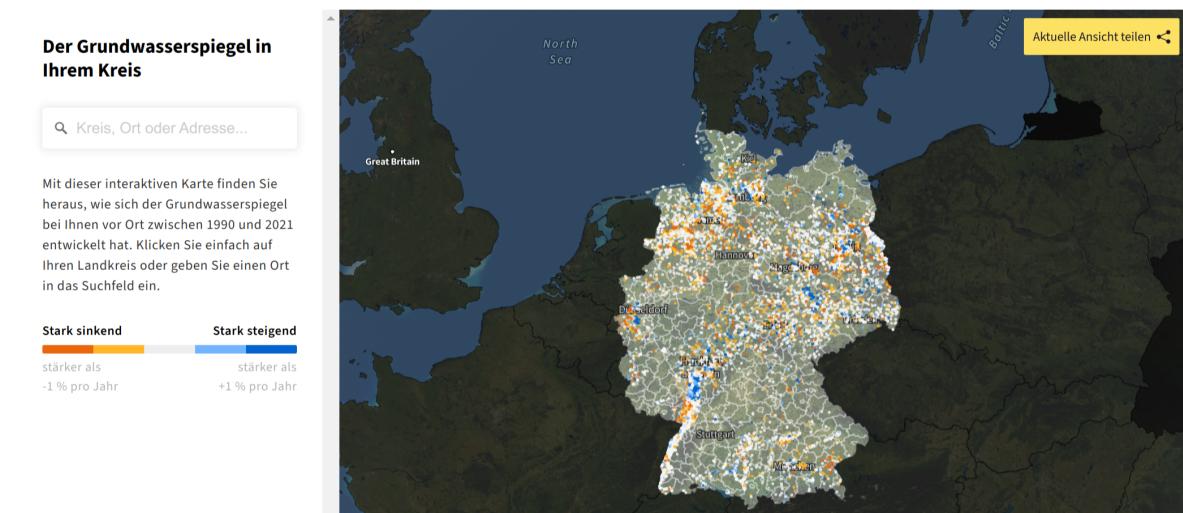
```

Praxisprojekt: Grundwasser-Atlas

Das Journalismuskollektiv CORRECTIV mit einer interaktiven Karte einen Überblick, wo in Deutschland das Grundwasser seit 1990 sinkt, gleich bleibt oder steigt.

Quelle: [CORRECTIV](#)

Interaktive Karte mit Lokalbezug



Quelle: [CORRECTIV](#)

Vorgehen beim Grundwasser-Atlas

- Abruf der Positionen aller Grundwassermessstellen
- Abruf der jeweiligen Pegelstände 1990-2021
- Datenreinigung, Normalisierung, Trendberechnung, Visualisierung

Welche Probleme können bei der Analyse auftreten?

- Daten nicht vorhanden oder nicht unter Open-Data-Lizenz
- Messwerte vorhanden, aber die Position der Messstellen nicht (Anonymisierung)
- Messstellen werden in unterschiedlichen Zeiträumen gemessen

Praxisbeispiel: Abruf Grundwassermessstellen Berlin

- Praktische Demonstration anhand eines Bundeslandes
- Suche über das Portal nach den Berliner Grundwasser-Messstellen

Praxisbeispiel: Abruf Grundwassermessstellen Berlin

The screenshot shows a search interface with a search bar containing the query "Grundwassermessstellen". Below the search bar, there is a "Filter" section with a "zurücksetzen" button. The "Region" filter shows a map of Europe with various regions highlighted in blue. The "Zeitraum" filter allows for date range selection. The "Lizenz" filter is present but not expanded. The main search results area displays two entries for "Grundwassermessstellen" from Hessen and Berlin. Each entry includes a title, a brief description, an "Aktualisiert" (last updated) date, a "Lizenz" (license) type, a "Herkunft" (origin) source, and "Schlagwörter" (keywords). A legend on the right side indicates four license types: F, A, I, and R, each represented by a purple bar.

Region	Aktualisiert	Lizenz	Herkunft	Schlagwörter
Hessen	19.11.2024	F, A, I, R	Bund\GDI-DE, Land\GDI-DE	Messstation, Hessen, Grundwasserentnahme, Rohwasser, INSPIRE-Daten
Berlin	10.10.2024	F, A, I, R	Bund\GovData	Messstation, Berlin, Grundwasser, Klimaschutz, Verkehr

Grundwassermessstellen als WFS-Daten

Web Feature Services sind Webschnittstellen, die den direkten Zugriff auf raumbezogene Daten ermöglichen.

 **Datensatz**

Grundwassermessstellen

Lage der Grundwassermessstellen, die sich im Eigentum der Senatsverwaltung für Mobilität, Verkehr, Klimaschutz und Umwelt und der Berliner Wasserbetriebe befinden.

Details zum Datensatz

Letzte Änderung
10.10.2024

Veröffentlichungsdatum
01.03.2024

Download-Link für Metadaten im RDF/XML-Format
[Download Metadaten](#)

Datenbereitsteller
GDI-DE

Veröffentlichende Stelle
Senatsverwaltung für Mobilität, Verkehr, Klimaschutz und Umwelt Berlin

Kategorien
Regionen und Städte
Umwelt
Verkehr

Schlagwörter

Ressourcen und Datenlinks

Titel und Details	Letzte Änderung	Dateiformat	
> Downloaddienst - Grundwassermessstellen (WFS)	10.10.2024	TEXT/HTML	zur Ressource
> Grundwassermessstellen	10.10.2024	VIEW	zur Ressource
> Grundwassermessstellen	10.10.2024	DOWNLOAD	zur Ressource

Quelle: [GovData](#)

Abruf der WFS-Daten mit Python

Request

```
import requests
import json

url = "https://gdi.berlin.de/services/wfs/gwm"
params = {
```

```

    'SERVICE': 'WFS',
    'VERSION': '2.0.0',
    'REQUEST': 'GetFeature',
    'TYPENAMES': 'gwm:grundwassermessstellen',
    'OUTPUTFORMAT': 'application/json'
}

response = requests.get(url, params=params)
data = response.json()

print(json.dumps(data, indent=2))

```

Abruf der WFS-Daten mit Python

Result

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "grundwassermessstellen.BL101A0060FIL001",
      "geometry": {
        "type": "Point",
        "coordinates": [
          409227.03,
          5812357.89
        ]
      },
      "geometry_name": "geom",
      ...
    }
  ]
}
```

Grundwassermessstellen

mvp.umwelt.info stellt die Daten der Messstellen bereits als Geojson-Datei bereit.



Link: [Grundwasser-Analyse](#)

Projekt-Ideen zum Kennenlernen von mvp.umwelt.info

- Ein weiteres Bundesland auswählen und Grundwasserdaten auf umwelt.info finden
- Metadatenanalyse der Grundwasser-Messstellen fortführen
- Wie entwickelt sich der Wasserstand bei Flüssen, Seen, Teichen?

Link: [Grundwasser-Analyse](#)

Bonus-Content

Auf den folgenden Folien finden sich einige vertiefende Infos zur Präsentation.

Web-UI vs. API-Zugriff Zusammenfassung

- Suche: Zum Erkunden und Verstehen der Daten
- Schnittstelle (API): Für systematische Datenabfragen

CKAN-API: Bereitstellung von Metadaten

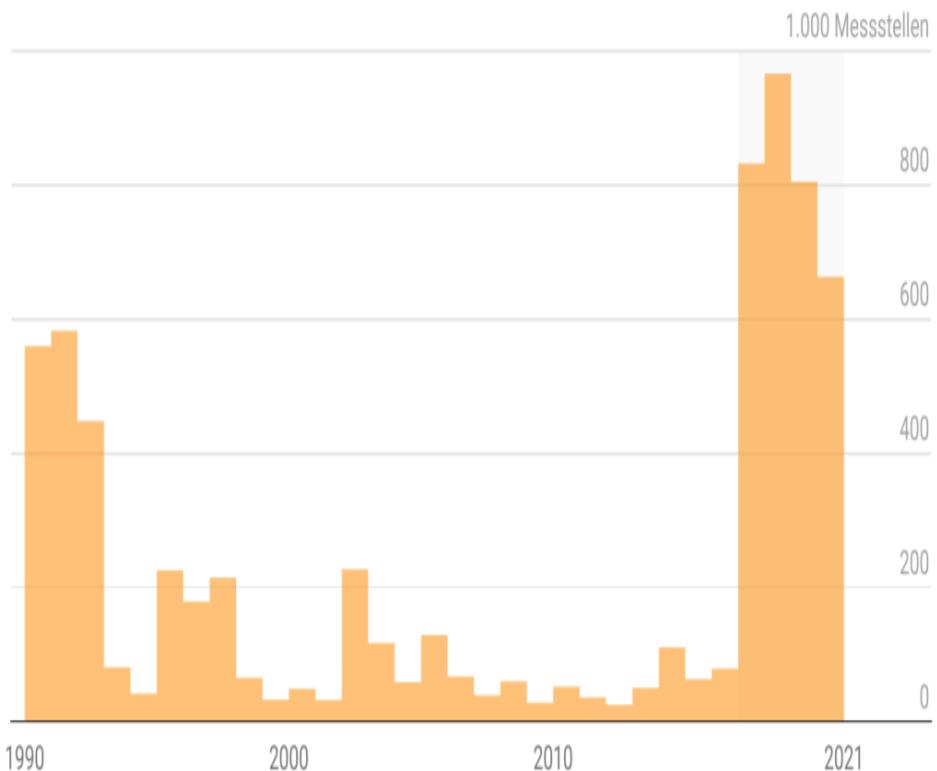
CKAN (Comprehensive Knowledge Archive Network) ist eine Open Source-Software zur Katalogisierung und Bereitstellung von Open Data.



Grundwasser-Atlas: Beispiel-Visualisierung

Anzahl der absoluten Tiefstwerte pro Jahr

In den Dürrejahren zwischen 2018 und 2021 erreichte das Grundwasser an der Hälfte der ausgewerteten Messstellen den tiefsten Stand seit 32 Jahren.



Die Analyse basiert auf einer Auswertung von 6.677 Grundwassermessstellen in 13 Bundesländern durch CORRECTIV.
Einige der Messstellen verzeichnen über mehrere Jahre einen Tiefstand.

Quelle: CORRECTIV • Erstellt mit [Datawrapper](#)

Quelle: [CORRECTIV](#)

Grundwasser-Atlas: Beispielhafte Visualisierung mit Python

Grundwassertrends in Deutschland 1990-2021 (Python-Code verfügbar in .qmd)

```
import plotly.graph_objects as go

# Data
trend_data = [
    {"name": "Stark sinkend", "value": 533, "color": "#E9650E", "percent": 8.0},
    {"name": "Leicht sinkend", "value": 1087, "color": "#FFB727", "percent": 16.3},
    {"name": "Kein starker Trend", "value": 4035, "color": "#eeeeee", "percent": 60.4},
    {"name": "Leicht steigend", "value": 529, "color": "#71B5FE", "percent": 7.9},
    {"name": "Stark steigend", "value": 493, "color": "#0163CB", "percent": 7.4}
]

# Create figure
fig = go.Figure()

# Add bar chart
fig.add_trace(
    go.Bar(
        x=[d["name"] for d in trend_data],
        y=[d["value"] for d in trend_data],
        marker_color=[d["color"] for d in trend_data],
        text=[d["value"] for d in trend_data],
        textposition='auto',
    )
)

# Update layout
fig.update_layout(
    title_text="Grundwassertrends in Deutschland 1990-2021",
    showlegend=False,
    height=500, # Reduced height since we only have one chart now
    paper_bgcolor='black',
    plot_bgcolor='black',
    font=dict(color='white'),
    template='plotly_dark'
    # annotations=[
    #     dict(
    #         text=f"Gesamtzahl der Messstellen: {sum(d['value'] for d in trend_data)}",
    #         xref="paper",
    #         yref="paper",
    #         x=0.5,
```

```
#           y=1.1,
#           showarrow=False,
#           font=dict(size=16, color='white')
#       )
#   ]
)

# Update axes
fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='rgba(128,128,128,0.2)')
fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='rgba(128,128,128,0.2)')

# Make it responsive
config = {
    'responsive': True,
    'displayModeBar': False
}

fig.show(config=config)
```

Unable to display output for mime type(s): text/html

Unable to display output for mime type(s): text/html