

## Funktionen der GEOROC API

- **Direkter Datenbezug:** APIs ermöglichen den direkten Zugriff auf Datenquellen, was den Prozess der Datensammlung erheblich beschleunigt und automatisiert.
- **Aktuelle Daten:** Forscher können die neuesten verfügbaren Daten in Echtzeit abrufen, was besonders in schnelllebigem Forschungsbereichen oder technischen Projekten wichtig ist.
- **Integration in bestehende Systeme:** Die API kann in bestehende Datenverwaltungssysteme integriert werden, was die Effizienz und Effektivität der Datenverarbeitung erhöht. Häufig verwendet im wissenschaftlichen Kontext ist Python, R, Julia und Golang. Ferner kann über die Befehlszeile oder Bash-Scripting das Programm 'Curl' für Abfragen genutzt werden.
- **Skriptbasierte Analyse:** Mit Python, R und anderen Programmiersprachen können Forscher Skripte erstellen, die Daten automatisch abrufen, verarbeiten und analysieren.
- **Flexible Abfrageoptionen:** Die API bietet vielfältige Endpunkte für spezifische Anfragen, was eine hohe Anpassungsfähigkeit für verschiedene Forschungsbedürfnisse ermöglicht.
- **Geodaten-Visualisierung:** Möglichkeit, Standortdaten in GeoJSON-Format zu erhalten, was die Integration in GIS-Software und die Visualisierung geologischer Daten erleichtert.
- **Standardisierte Formate:** Die API liefert Daten in standardisierten Formaten wie JSON und XML, die leicht in Datenanalyse-Tools integriert werden können.

### Zugriff auf Geodaten:

- GET /geodata/samplesclustered: Abruf von Proben-IDs, gefiltert und geclustert nach verschiedenen Kriterien.
- GET /geodata/sites: Abruf von Standortdaten im GeoJSON-Format.

### Allgemeine Anfragen:

- GET /ping: Eine einfache Anfrage zur Überprüfung der API-Erreichbarkeit.

### Personenbezogene Daten:

- GET /queries/authors: Abruf von Autoreninformationen.
- GET /queries/authors/{personID}: Abruf von Autoreninformationen anhand der Person-ID.

**Zitationsdaten:**

- GET /queries/citations: Abruf von Zitationsdaten.
- GET /queries/citations/{citationID}: Abruf von Zitationsdaten anhand der Zitations-ID.

**Vollständige Datensätze:**

- GET /queries/fulldata: Abruf vollständiger Datensätze anhand einer Liste von Proben-IDs.
- GET /queries/fulldata/{samplingfeatureid}: Abruf eines vollständigen Datensatzes anhand einer Proben-ID.

**Standortdaten:**

- GET /queries/locations/l1, /l2, /l3: Abruf von Standortdaten verschiedener Ebenen.

**Proben- und Materialdaten:**

- Verschiedene Endpunkte zum Abruf von Informationen über chemische Elemente, Gesteinsklassen, Materialien, Mineralien, Probenahmetechniken und mehr.

**Standortbezogene Anfragen:**

- GET /queries/sites: Abruf aller Standorte.
- GET /queries/sites/settings: Abruf geologischer Einstellungen.
- GET /queries/sites/{samplingfeatureID}: Abruf von Standorten anhand der Proben-ID.

**Statistikdaten:**

- GET /queries/statistics: Abruf von Datenstatistiken.

---

**Grenzen der API:**

- **Sicherheitsprotokolle:** Die API erfordert bestimmte Sicherheits- und Autorisierungsprotokolle.
- **Einhaltung der Lizenzbedingungen:** Nutzer müssen die CC BY-SA 4.0 Lizenzbedingungen verstehen und einhalten, was die Nutzung der Daten in kommerziellen oder proprietären Projekten einschränken könnte.

- **Technische Beschränkungen:** Einschränkungen in der Abfragestruktur, wie das Verbot von Semikolons.
- **Festgelegte API-Endpunkte:** Die Interaktion mit der API ist auf die vorgegebenen Endpunkte beschränkt.
- **Komplexität der API-Integration:** Die Integration und Nutzung der API kann für Nutzer ohne technischen Hintergrund herausfordernd sein.
- **Datenformat und -struktur:** Nutzer müssen sich mit den spezifischen Datenformaten und -strukturen vertraut machen, die die API verwendet.
- **API-Rate-Limits und Zugriffsbeschränkungen:** Es gibt Beschränkungen hinsichtlich der Anzahl der Anfragen, die ein Nutzer innerhalb eines bestimmten Zeitraums stellen kann.

---

### Zusätzliche Ressourcen:

Die API-Dokumentation bietet Links zu weiteren Ressourcen wie der DIGIS-Projektwebsite und Informationen zur Datenlizenzierung unter CC BY-SA 4.0.

Diese erweiterte Übersicht bietet einen umfassenderen Einblick in die Funktionen und Einschränkungen der GeoROC API, die für die Verwaltung und den Abruf geowissenschaftlicher Daten konzipiert ist.

### Filter-DSL

Die Endpunkte `/geodata/sampleclustered` und `/queries/sample` der GeoROC API bieten flexible Filteroptionen, um spezifische `samplingfeatureIDs` basierend auf verschiedenen Kriterien abzurufen. Hier ist eine detaillierte Beschreibung der Filter-DSL (Domain Specific Language) Syntax und ihrer Funktionsweise:

**Filter-DSL Syntax:** Die Syntax für die Filterung folgt dem Muster **FIELD=OPERATOR:VALUE**. Dabei gilt:

**FIELD:** Eines der akzeptierten Abfrageparameter.

**OPERATOR:** Einer der folgenden Operatoren:

**lt** für “kleiner als” (`<`)

**gt** für “größer als” (`>`)

**eq** für “gleich” (`=`)

**in** für “IN” (eine Liste von Werten)

**lk** für “LIKE” (ähnlich, mit Platzhaltern)

**btw** für “BETWEEN” (zwischen zwei Werten)

**VALUE:** Ein unzitierter String, eine ganze Zahl oder eine Dezimalzahl.

**Besonderheiten der Operatoren: Mehrere Werte für “in”:** Bei Verwendung des in-Operators können mehrere, durch Kommas getrennte Werte angegeben werden. Diese werden als disjunktiver Filter interpretiert (d.h., jeder der Werte kann zutreffen).

**Numerische Werte:** Die Operatoren lt, gt und btw sind nur auf numerische Werte anwendbar.

**String-Werte:** Der Operator lk ist nur auf String-Werte anwendbar und unterstützt Wildcards wie \* (null oder mehr Zeichen) und ? (ein Zeichen).

**Zwischenwerte für “btw”:** Der Operator btw akzeptiert zwei durch Kommas getrennte Werte als inklusive untere und obere Grenze. Fehlende Werte werden als 0 bzw. 9999999 angenommen.

**Standardoperator:** Wenn kein Operator angegeben ist, wird eq als Standardoperator angenommen.

**Example of use** Angenommen, Sie wollen Proben finden, die aus einer bestimmten Region stammen, ein bestimmtes Alter haben und einer bestimmten Gesteinsart entsprechen. Alter haben und einer bestimmten Gesteinsart entsprechen. Sie könnten dann die folgenden Filter verwenden:

```
def get_filtered_samples(  
    limit=None,  
    offset=None,  
    setting=None,  
    location1=None,  
    location2=None,  
    location3=None,  
    latitude=None,  
    longitude=None,  
    rocktype=None,  
    rockclass=None,  
    mineral=None,  
    material=None,  
    inclusiontype=None,  
    sampletech=None,  
    chemistry=None,  
    title=None,  
    publicationyear=None,  
    doi=None,  
    firstname=None,  
    lastname=None,  
    agemin=None,  
    agemax=None,
```

```

        geoage=None,
        geoageprefix=None,
        lab=None,
        polygon=None,
        addcoordinates=None
    ):

        endpoint = "queries/samples"

        filters = {
            key: value
            for key, value in locals().items()
            if key not in ["endpoint"] and value is not None
        }

        data = api_query(endpoint, params=filters)
        return data

# Used DSL-Filter
filtered_samples_combined = get_filtered_samples(
    location1="eq:CENTRAL INDIAN TECTONIC ZONE",
    agemin="gt:150",
    rocktype="eq:Granite"
)

```

---

**Konjunktive Auswertung:** Die Filter werden konjunktiv ausgewertet, d.h., alle Bedingungen müssen erfüllt sein.

**Leistungshinweise:** Mehr Filter können die Abfrage verlangsamen, da mehr Tabellen in der Auswertung berücksichtigt werden müssen.

Diese Filteroptionen bieten eine leistungsstarke Möglichkeit, Daten aus der GeoROC-Datenbank gezielt abzufragen, wobei die Nutzer die Flexibilität haben, ihre Abfragen nach spezifischen Bedürfnissen anzupassen.