

Einführung in STAC & STAC API

Matthias Mohr

Matthias Mohr

E-Mail
matthias@mohr.ws

Homepage
mohr.ws

GitHub
[m-mohr](https://github.com/m-mohr)

- Selbständig in Münster
- Softwareentwicklung, Architektur und Beratung
 - STAC
 - openEO
 - Cloud-Native Geospatial
- Hauptautor der openEO-Spezifikation und PSC-Vorsitz
- Mitautor der STAC-Spezifikationen und PSC-Mitglied
 - Dadurch auch aktiv in einige OGC-Arbeitsgruppen
- Entwickler vieler Open-Source-Anwendungen, bspw.
 - JavaScript Ökosystem für STAC & openEO
 - STAC Browser
 - openEO Web Editor
 - ol-stac für OpenLayers / stac-layer für Leaflet

Warum STAC?


```

2926     colorCorrections:
2927       $ref: '#/definitions/ColorCorrection'
2928 OutputExportDefinition:
2929   type: object
2930   properties:
2931     rasterSize:
2932       type: integer
2933       description: desired tiff size after export
2934     crs:
2935       type: integer
2936       description: epsg projection code
2937     stitch:
2938       type: boolean
2939       description: stitch tiles into a single geotiff if possible
2940     crop:
2941       type: boolean
2942       description: crop stitched raster by provided mask if possible
2943     source:
2944       type: string
2945       format: uuid
2946       description: output export path
2947     render:
2948       $ref: '#/definitions/Render'
2949     dropboxCredential:
2950       type: string
2951       description: Dropbox token
2952 Render:
2953   type: object
2954   properties:
2955     operation:
2956       type: string
2957       description: currently useless Field
2958     bands:
2959       type: array
2960       description: bands in the exported geotiffs
2961     items:
2962       type: integer
  
```

organization	UUID for organization to filter by
string(\$uuid)	
<i>(query)</i>	
maxCloudCover	Only return results with cloud cover less than this number
number	
<i>(query)</i>	
minCloudCover	Only return results with cloud cover greater than this number
number	
<i>(query)</i>	
minAcquisitionDatetime	Only return results acquired after this datetime
string(\$datetime)	
<i>(query)</i>	
maxAcquisitionDatetime	Only return results acquired before this datetime
string(\$datetime)	
<i>(query)</i>	
minCreateDatetime	Only return results created after this datetime
string(\$datetime)	
<i>(query)</i>	
maxCreateDatetime	Only return results created before this datetime
string(\$datetime)	
<i>(query)</i>	
datasource	Only return results belonging to this datasource
array[string]	
<i>(query)</i>	
month	Only return results from this month

Was ist STAC?

“SpatioTemporal Asset Catalog”

STAC ist eine Sammlung von
Spezifikationen und Software

Die STAC-Spezifikation
beschreibt geografische
und/oder zeitliche Daten
(durch miteinander verlinkte statische
JSON-Dateien)

Die STAC API-Spezifikation fügt
dynamische Elemente hinzu, wie
Suche, Aggregation und
Transaktion

Was ist es?

- Spezifikationen
 - Fokus: Suchen und auffinden von Daten
 - Einfach und erweiterbar
 - statische STAC-Spezifikation mit Erweiterungen
 - dynamische STAC API-Spezifikation mit Erweiterungen
 - *als OGC Community Standard eingereicht*
- Ökosystem
 - Daten(!)
 - Software-Anwendungen, viele davon Open Source
 - Klienten in verschiedensten Programmiersprachen
 - Server-Anwendungen
 - sonstige Hilfsmittel, wie bspw. Validatoren
 - Lernmaterialien

Was ist es NICHT unbedingt?

- Vollwertiger Metadaten standard
- Einzige Quelle für Metadaten
- Deckt nicht alle Arten von Daten ab
- Ersatz für ISO 19115, OGC CSW/Records, ...

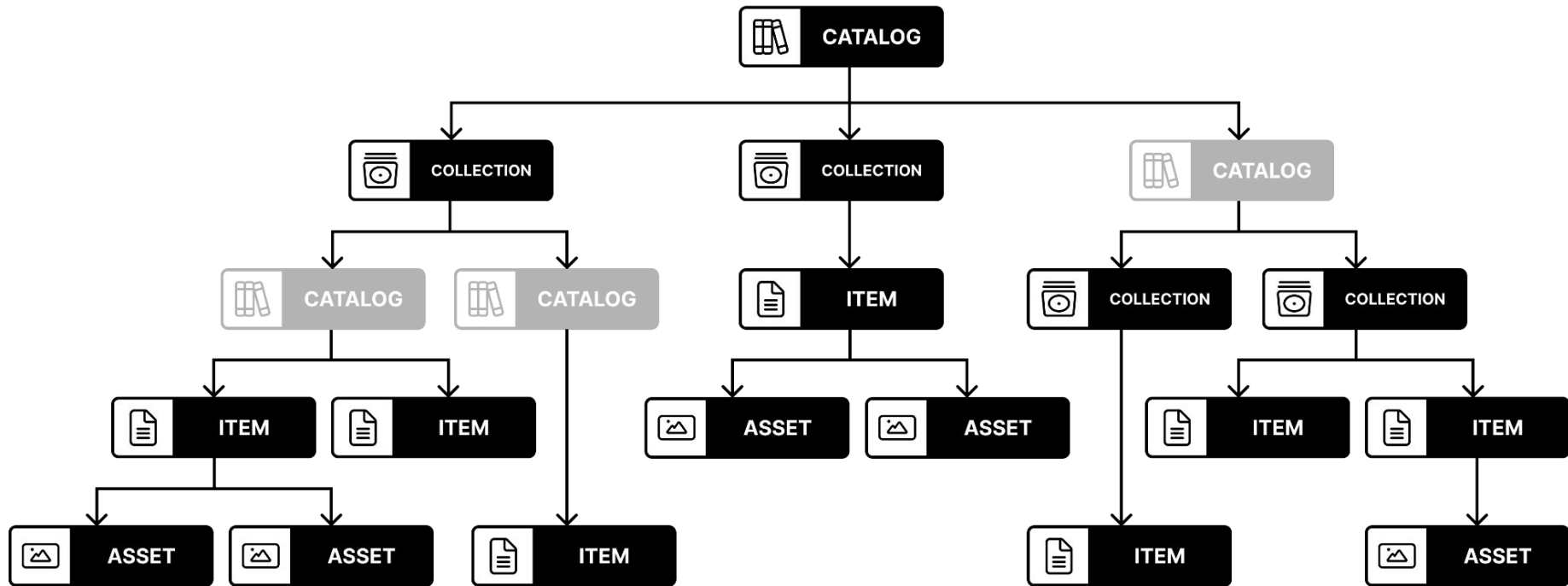
STAC-Spezifikation

Wichtige Entitäten in STAC (I)

- **Catalog**
 - Einfach, primär für statische Kataloge (quasi virtuelle Ordner)
 - Gruppierung für Items, Catalogs und Collections (anstelle von Seiten in einer API)
 - *Beispiel: Alle Sentinel-2-Bilder für einen bestimmten Tag*
- **Collection**
 - Im Kern ein Katalog mit mehr Metadaten (Ausdehnung, Lizenz, Provider, Schlagwörter, ...)
 - *Beispiel: Alle Sentinel-2-Bilder*
- **Item**
 - Dateien für eine Aufnahme zu einem bestimmten Datum + Zeit
 - erweitert GeoJSON
 - *Beispiel: Alle Bänder einer einzelnen Sentinel-2-Aufnahme*

Wichtige Entitäten in STAC (II)

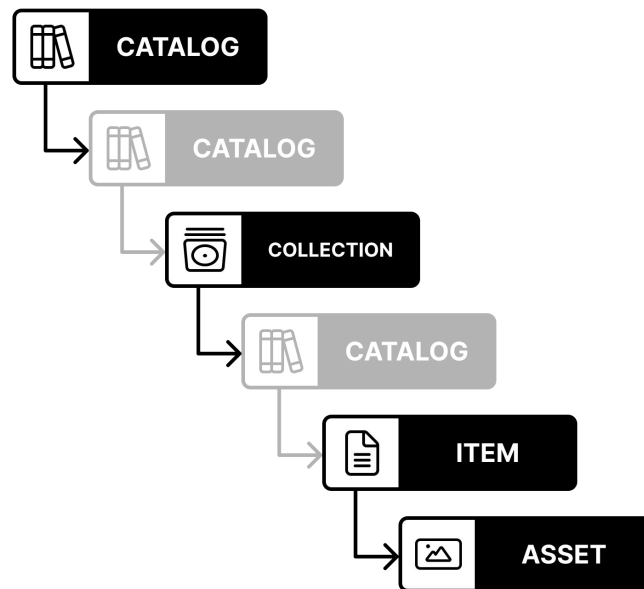
- **Asset** (innerhalb: Item, Collection)
 - Eine einzelne Datei einer Aufnahme
 - *Beispiel: Grünes Band einer Sentinel-2-Aufnahme als GeoTiff*
- **Link** (innerhalb: Catalog, Collection, Item)
 - In Zusammenhang stehende Daten
 - *Beispiel: Sentinel-Dokumentation, Verbindung zwischen STAC-Entitäten*
- **Erweiterungen**
 - zum Beispiel **Multi-spektral, SAR, Punktwolken, Wettervorhersagen, Mehrsprachigkeit, ...**



STAC-API-Spezifikation

STAC API

- HTTP(S) und JSON
- Mehrere Teile:
 - Core
 - Suche nach Items
 - Collections / Features
 - Erweiterungen
 - Aggregationen
 - Suche nach Collections
 - Filter
 - Mehrsprachigkeit
 - Transaktionen
- Kürzlich erst final veröffentlicht als 1.0.0 🎉



STAC API vs OGC API - Features

- Parallel gestartet, später möglichst angeglichen
- STAC API basiert auf OGC API - Features
 - Das Metadaten-Modell ist größtenteils abgeglichen mit OGC API - Records
 - Transaktionen basieren auf Features - Teil 4
 - Filter-Erweiterung basiert auf Features und CQL2
- In der STAC API sind die Features die Metadaten!
- Vektordaten fein-granular verfügbar machen: OGC API - Features
- Rasterdaten oder gebündelte Vektordaten verfügbar machen: STAC API

Ökosystem & Implementierungen

=> siehe stacindex.org

Hilfreiche Links

(aber alle in Englisch)

- [Offizielle Webseite](#)
 - [STAC-Spezifikation](#)
 - [STAC-API-Spezifikation](#)
 - [STAC Index: Liste mit Katalogen, Anwendungen und Tutorials](#)
 - [STAC-Erweiterungen](#)
 - [STAC-API-Erweiterungen](#)
-
- [STAC community call](#)
alle 2 Wochen, Montags um 17:00
 - [STAC working session](#)
jeden ersten Dienstag im Monat um 17:00
 - [Google-Gruppe für Einladungen](#)

Q&A