

Bibliothekarische Schnittstellen

Exemplarische APIs zum Lesezugriff

KIM Workshop, Mannheim, 2023-05-08

Übersicht

Was ist eine Schnittstelle?

Grundlagen (HTTP)

Einfache Abfragen (unAPI)

Komplexere Abfragen (SRU)

Anwendungsspezifische APIs (DAIA)

Authentifizierung (PAIA)

Komplexere Abfragesprachen (SPARQL)

Was ist eine Schnittstelle?

Finden wir es heraus!

Vier Gruppen:

- ▶ Schnittstelle (hat die Datenbank)
- ▶ Transport (Internet)
- ▶ 2x Client-Anwendungen (stellen Anfragen)

*Kommunikation zwischen den Gruppen nur per Daten, d.g.
niemand darf mit den Anderen Gruppen reden!*

Schnittstellen

- ▶ Teil eines Systems welcher der Kommunikation mit anderen Systemen dient.
- ▶ Rest des Systems ist nicht zugänglich (Black-Box)
- ▶ Kommunikation per festgelegten Protokollen und -Formaten

<https://de.wikipedia.org/wiki/Schnittstelle>

Grundlagen (HTTP)

Warum das *Hypertext Transfer Protocol* (HTTP)?

- ▶ Grundlage des WWW (1989)
- ▶ Grundlage der meisten APIs (*Web-Services, REST...*)

Lernziel: *HTTP-Abfragen verstehen und analysieren*

- ▶ Bestandteile einer HTTP-Abfrage
- ▶ Bestandteile einer HTTP-Antwort

Bestandteile einer HTTP-Abfrage

- ▶ Protokoll (HTTP/HTTPS)
- ▶ GET/POST/PUT/DELETE/PATCH
- ▶ URL
 - ▶ Server
 - ▶ ggf. Port (80, 443, ...)
 - ▶ URL-Pfad
 - ▶ Query-Parameter (optional)
- ▶ Header (optional: Cookies, Parameter...)
- ▶ Body (optional)

Bestandteile einer HTTP-Antwort

- ▶ Header
- ▶ Status-Code (200, 404, ...)
- ▶ Body

HTTP-Clients die Jeder kennen sollte

- ▶ Browser
Firefox Netzwerkanalyse (Strg + Shift + E)
- ▶ curl
- ▶ wget

Übung: Wie viele und welche Anfragen macht der Browser bei Aufruf von <http://gso.gbv.de/>?

Darüber hinaus...

- ▶ HTTP ist grundsätzlich zustandslos. *Gegenbeispiel?*
- ▶ HTTP/2, HTTP/3 betreffen lediglich die Verbindungsebene
- ▶ Websockets
 - ▶ initiiert durch HTTP
 - ▶ dann full-duplex Nachrichten zwischen Server und Client
- ▶ P2P-Alternativen: BitTorrent, IPFS, ...
- ▶ *Fragen?...*

Einfache Abfragen (unAPI)

- ▶ Zweck: einzelne Datensätze in verschiedenen Datenformaten
- ▶ Einfache GET-Abfrage mit zwei Query-Parametern
 - ▶ id
 - ▶ format
- ▶ <http://unapi.gbv.de/>
- ▶ <http://unapi.k10plus.de/>

Übung: MARC-Datensätze vergleichen

Komplexere Abfragen (SRU)

Komplexere Abfragen (SRU)

► <http://sru.k10plus.de/>

```
catmandu convert SRU --base http://sru.gbv.de/opac-de-204\  
  --query "pica.bbg=Tdv" --recordSchema picaxml \  
  --parser picaxml to PICA --type plain
```

Anwendungsspezifische APIs (DAIA)

Anwendungsspezifische APIs (DAIA)

...

Authentifizierung (PAIA)

Authentifizierung (PAIA)

- ▶ `http://paia.gbv.de/`
- ▶ `http://gbv.github.io/paia/`

...

Komplexere Abfragesprachen (SPARQL)

- ▶ Anfrage in spezieller Abfragesprache
- ▶ Jede Abfragesprache nimmt eine grundlegende Datenstrukturierungssprache an

Beispiele: SQL, SPARQL, GraphQL

SPARQL

- ▶ Datenstrukturierungssprache: RDF
- ▶ Semantic Web in Bibliotheken (SWIB-Konferenz)
- ▶ SPARQL-Endpunkte
- ▶ Kenntniss der verwendeten Ontologien hilfreich

Mögliche Übung: <https://query.wikidata.org/>

Fragen? Vertiefung?