

SRU



Bitte beachten Sie: Für den GVK sowie einige andere [Datenbanken](#) verwenden Sie bitte ab sofort die [K10plus-SRU-Schnittstelle](#).

Was ist SRU?

SRU (Search/Retrieve via URL) ist ein auf HTTP basierendes Protokoll, um Datenbankabfragen durchzuführen. Suchanfragen können dabei über einen URL in CQL (Contextual Query Language) übermittelt werden. Die zurückgelieferten Ergebnisse können per XML in verschiedenen bibliographischen Formaten ausgeliefert werden.

Weitere Informationen zu SRU gibt es unter [\[1\]](#) und [\[2\]](#).

Angebot der Verbundzentrale

Die VZG bietet unter <http://sru.gbv.de> einen SRU-Service für verschiedene PSI-Systeme als Datenquellen an. Grundsätzlich ist es möglich, hierüber auf die bibliographischen Daten

- des Katalogisierungssystems (CBS)
- des Recherchesystems im WWW (GSO)
- der Lokalen Bibliothekssysteme (LBS)

zuzugreifen. Von den CBS- und LBS-Systemen werden derzeit allerdings nicht alle unterstützt.

Verwendung

Sie erreichen den Service unter sru.gbv.de - so können Sie beispielsweise die Online Contents Aufsatzdatenbank (OLC) über SRU abfragen, indem Sie die Adresse <http://sru.gbv.de/olc> als BaseURL verwenden. Die Authentifizierung erfolgt automatisch per IP-Adresse oder mittels Zugangsdaten, die Sie von der Verbundzentrale erhalten. Welche Ausgabeformate bislang unterstützt werden, erfahren Sie weiter unten.

- Beispiel-URL für eine searchRetrieve-Operation; gesucht wird das Stichwort "entwicklung" im Titindex der Datenbank "olc" mit einem Ergebnis von maximal 10 Datensätzen im Format "mods":

<http://sru.gbv.de/olc?version=1.1&operation=searchRetrieve&query=pica.tif%3Dentwicklung&maximumRecords=10&recordSchema=mods>

- Beispiel für eine Recherchesitzung mittels yaz-client [\[3\]](#); gesucht wird das Stichwort "entwicklung" im Titindex der Datenbank "olc", Anzeige des ersten Treffers im Format MARC-XML:

```

Z> open http://sru.gbv.de/olc
Connecting...OK.
Z> find pica.tit=entwicklung
Received SRW SearchRetrieve Response
Number of hits: 55116
Elapsed: 0.026330
Z> s 1
Received SRW SearchRetrieve Response
Number of hits: 55116
pos=1 schema=
<?xml version="1.0" encoding="UTF-8"?>
<record xmlns="http://www.loc.gov/MARC21/slim">
  <leader>xxxxxnaa a22yyyyy c 4500</leader>
  <controlfield tag="001">2001270054</controlfield>
  <controlfield tag="003">DE-601</controlfield>
  <controlfield tag="005">20190726145930.0</controlfield>
  <controlfield tag="008">190726s2019          000 0 und d</controlfield>
  <datafield tag="016" ind1="7" ind2=" " >
    <subfield code="a">200635--2</subfield>
    <subfield code="2">DE-600</subfield>
  </datafield>
  <datafield tag="024" ind1="8" ind2=" " >
    <subfield code="a">99500424</subfield>
  </datafield>
  <datafield tag="028" ind1="5" ind2="2">
    <subfield code="a">ctib1915</subfield>
  </datafield>
  <datafield tag="040" ind1=" " ind2=" " >
    <subfield code="b">ger</subfield>
    <subfield code="c">GBVCP</subfield>
  </datafield>
  <datafield tag="084" ind1=" " ind2=" " >
    <subfield code="a">wiw</subfield>
    <subfield code="2">olc-ssg</subfield>
  </datafield>
  <datafield tag="245" ind1="0" ind2="0">
    <subfield code="a">Versicherer setzen auf eigene Software-Entwicklung</subfield>
  </datafield>
  <datafield tag="773" ind1="0" ind2="8">
    <subfield code="i">in</subfield>
    <subfield code="t">Zeitschrift für Versicherungswesen : ZfV : unabhängiges Fachorgan für d.
Versicherungspraxis</subfield>
    <subfield code="d">Hamburg : Allg. Fachverl. Mathern</subfield>
    <subfield code="g">Vol. 70, No. 14 (2019), p. 425</subfield>
    <subfield code="q">70:14&lt;425</subfield>
    <subfield code="w">(DE-601)129461199</subfield>
    <subfield code="x">0514-2784</subfield>
  </datafield>
  <datafield tag="900" ind1=" " ind2=" " >
    <subfield code="a">GBV</subfield>
    <subfield code="b">SBB-PK Berlin &lt;l+1A&gt;</subfield>
    <subfield code="d">!2:HB9! HB 9 Ja 640</subfield>
    <subfield code="x">L</subfield>
    <subfield code="z">LC</subfield>
    <subfield code="g">Zsn 28676</subfield>
    <subfield code="c">22.1971 -</subfield>
  </datafield>
  [...]
Elapsed: 0.040522
Z>

```



Neu:

Es ist nun auch möglich, HTTPS für Ihre Anfragen zu verwenden, dazu können Sie das Protokoll http:// in den Beispielen einfach durch https:// austauschen. Bitte beachten Sie, dass Browser sich teilweise merken, dass für die Domain HTTPS verwendet wird ("HSTS"). Falls Sie anschließend dennoch einmal wieder zu HTTP zurück wechseln möchten, gibt es meist die Option, den Browser die Seiteneinstellungen für die Domain "vergessen" zu lassen.

Beispiel für eine Recherche mit Catmandu-SRU, Ausgabe in PICA Plain:

```

catmandu convert SRU --base http://sru.k10plus.de/gvk --query "pica.tit=Nuklearwaffenkonvention" --recordSchema
picaxml --parser picaxml to PICA --type plain

```

Datenbanken

Eine Datenbank wird durch einen alphanumerischen Schlüssel abgekürzt angesprochen. In der Regel verfügen Datenbanken auch über eine öffentliche URL mit einer Suchoberfläche. **Eine Liste der verfügbaren Datenbanken steht unter <http://uri.gbv.de/database/>.** Hier einige Beispiele:

Schlüssel	Datenquelle	URL
olc	OLC - Online Contents	https://gso.gbv.de/DB=2.3/
opac-de-luen4	Gesamtkatalog Universitätsbibliothek Lüneburg	http://hlue.gbv.de/DB=1/LNG=DU/

Als Suchattribute stehen sämtliche Indexschlüssel (IKTs) zur Verfügung, die auch in der jeweiligen Web-Datenbank benutzbar sind. Eine detaillierte Liste der verfügbaren Suchschlüssel erhalten Sie, wenn Sie die gewünschte Datenbank via <http://uri.gbv.de/database/> aufrufen und an den URL die Zeichenkette "XML=1.0/IKTLIST" anfügen, z.B. für Online Contents: <https://gso.gbv.de/DB=2.3/XML=1.0/IKTLIST>

Formate

Ein Format wird durch eine FCV-Tabelle und eine Liste von Filtern definiert und ebenfalls durch einen alphanumerischen Schlüssel abgekürzt angesprochen. Eine Liste der verfügbaren Formate steht unter <http://unapi.gbv.de/about>, wobei für SRU lediglich XML-basierte Formate unterstützt werden. Hier einige Beispiele:

Format	Name	Typ	Beschreibung
dc	Dublin Core	application/xml	Dublin Core
marcxml	MARC-XML	application/xml	MARC-Daten in XML
mods	MODS	application/xml	Metadata Object Description Schema
picaxml	PicaXML	application/xml	PICA+ in XML
turbomarc	Turbomarc	application/xml	Turbomarc, faster XML for MARC records

Sofern Sie Anregungen und Fehler bezüglich der Metadaten-Konvertierung haben, teilen Sie uns diese bitte unter Angabe eines konkreten Anwendungsbeispiels mit.

Sortierung

Als Sortierung der Treffer werden "relevance", "year", "author" und "title" unterstützt, der Standardwert ist "year" (absteigend sortiert). Wird auf die Angabe der Sortierreihenfolge verzichtet, dann ist sie ansonsten immer aufsteigend. Die Sortierung kann entweder über URL-Parameter (v1.1) oder über CQL (v1.2) festgelegt werden:

Beispiel SRU/SRW v1.1

<http://sru.gbv.de/olc?version=1.1&operation=searchRetrieve&query=pica.all%3Dreformation&maximumRecords=5&sortKeys=year,,1>

Hierbei kann die Sortierreihenfolge mittels des dritten Wertes für sortKeys bestimmt werden: 1 = aufsteigend (Standardwert), 0 = absteigend

Beispiel SRU/SRW v1.2

<http://sru.gbv.de/olc?version=1.2&operation=searchRetrieve&query=pica.all%3Dreformation+sortby+year%2Fdescending&maximumRecords=5>

Filter

Beispiel für die Einschränkung der Suchergebnisse in "vd18" auf Titeldatensätze:

<http://sru.gbv.de/vd18!rec=1?version=1.1&operation=searchRetrieve&query=pica.all%3Dhamburg&maximumRecords=10&recordSchema=marcxml>

Beispiel für die Einschränkung der Suchergebnisse in "vd18" auf Normdatensätze:

<http://sru.gbv.de/vd18!rec=2?version=1.1&operation=searchRetrieve&query=pica.all%3Dhamburg&maximumRecords=10&recordSchema=marcxml>

Beispiel zur Beschränkung der Datensatz-Anzeige in "olc" auf die Titelebene (Ausblenden von Lokal- und Exemplardaten):

<http://sru.gbv.de/olc!levels=0?version=1.1&operation=searchRetrieve&query=pica.all%3Dschiffahrt&maximumRecords=10&recordSchema=marcxml>