Handbuch zu LOSDDaten von Statistik Stadt Zürich

Matthias Günter, 16.7.2018

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc520808816)

[2 Daten 2](#_Toc520808817)

[2.1 Begriffe zu den historische Daten 2](#_Toc520808818)

[2.2 Begriffe zum RDF Data Cube Modell 2](#_Toc520808819)

[3 Linked Data 4](#_Toc520808820)

[3.1 Zugang mittels SPARQL 4](#_Toc520808821)

[3.2 Beispiel Query: Alle Datensets 4](#_Toc520808822)

[3.3 Beispiel Query: Die ersten Hundert Triples 5](#_Toc520808823)

[3.4 Beispiel Query: Die Anzahl Triple im Triple Store 6](#_Toc520808824)

[3.5 Beispiel: Anzahl Observationen pro Dataset für alle Datesets 6](#_Toc520808825)

[3.6 Beispiel: Suche 7](#_Toc520808826)

[3.7 Beispiel: Suche nach Fläche ohne Wald für ganz Zürich 8](#_Toc520808827)

[3.8 Beispiel: Alle Codes 9](#_Toc520808828)

[3.9 Beispiele: Hierarchien 10](#_Toc520808829)

[3.10 Beispiel: Hierarchie des Raums 11](#_Toc520808830)

[3.11 Beispiel: Alle Arten von Kennzahlen 13](#_Toc520808831)

[3.12 Beispiel eines Datensatzes 14](#_Toc520808832)

[3.13 Beispiel: Verknüpfung von Kennzahlen 17](#_Toc520808833)

[3.14 Beispiel: Umgang mit Zeit 18](#_Toc520808834)

[3.15 Grundlagen SPARQL 18](#_Toc520808835)

[4 Zugang mittels REST-Interface 18](#_Toc520808836)

[4.1 Grundlegendes 18](#_Toc520808837)

[4.2 Search-API 19](#_Toc520808838)

[4.3 Alle Dataset 19](#_Toc520808839)

[4.4 Dataset 20](#_Toc520808840)

[4.5 Slice 21](#_Toc520808841)

[4.6 Shape 25](#_Toc520808842)

[4.7 Darstellung verschiedener Metadaten 26](#_Toc520808843)

[4.8 API-Tests 26](#_Toc520808844)

[5 Zugang mit Hydra-Client 26](#_Toc520808845)

[6 Verwendung von sszvis 27](#_Toc520808846)

[7 OpenSource Projekte zu STIP, LOSD und sszvis 27](#_Toc520808847)

# Einleitung

Statistik Stadt Zürich ersetzt das Jahrbuch mit einer Weblösung (stat.stadt-zuerich.ch). Die Lösung kann direkt online verwendet werden.

Allerdings können die Schnittstellen auch programmatisch verwendet werden. Dazu gibt es die folgenden Zugänge:

* Mittels SPARQL direkt auf die Basisdaten
* Mittels REST-Interface

Dieses Dokument beschreibt, welche Daten wie zur Verfügung stehen und wie diese von interessierten Kreisen abgefragt werden können.

Der einfache Download über stat.stadt-zuerich.ch wird dabei ausgeführt.

# Linked Data

Die Daten werden von Statistik Stadt Zürich über Linked Open Data (LOD) zur Verfügung gestellt. Die Daten können aus dem sogenannten Data Cube mittels SPARQL abgefragt und verarbeitet werden.

Eine Überblick kann man sich mit dem [Linked Data crash course](http://www.linkeddatatools.com/semantic-web-basics) verschaffen (nur in englisch).

## Grundsteine

Linked Data ist ein weitläufiges Thema und eine komplette Abdeckung oder Schulung würde bei weitem den Rahmen dieses Dokuments sprengen. Es seien aber die wichtigsten Begriffe genannt und kurz erklärt oder entsprechend verlinkt.

TODO

* Open Data und Linked Data: Definition und Vorteile
* Linked Data vs. Relationale Datenbank: Graphen vs. Tabellen
* Grundsteine von Linked Data: URIs für die Benennung, RDF für die Beschreibung, SPARQL für die Abfrage von Daten
* Triple Storage und Resource Description Framework (RDF): Subjekt 🡪 Prädikat 🡪 Objekt
* RDF Vokabular: RDFS (RDF Schema), OWL (Web Ontology Language) und viele weitere
* RDF Serialisierung: TURTLE, JSON etc.
* RDF Data Cube Modell (siehe Abschnitt 3.2)
* und schlussendlich, die Abfrage der Daten: SPARQL
* evt. Vergleich zu SQL?

## Hilfreiche Links

TODO

[RDF / OWL basics](https://www.scss.tcd.ie/Owen.Conlan/CS7063/06%20Introduction%20to%20OWL%20(1%20Lecture).ppt.pdf)

[SPARQL cheat-sheet](http://www.iro.umontreal.ca/~lapalme/ift6281/sparql-1_1-cheat-sheet.pdf)

[SPARQL Einführung](http://dbis.informatik.uni-freiburg.de/content/courses/WS1011/Spezialvorlesung/Webbasierte%20Informationssysteme/folien/Vorlesung-SPARQL-Einfuehrung.pdf)

[TURTLE syntax](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwikusXevJrcAhWKlCwKHTv1Cj0QFgg3MAA&url=http%3A%2F%2Finfo.slis.indiana.edu%2F~dingying%2FTeaching%2FZ636%2FSlides%2FTurtleRDF.ppt&usg=AOvVaw1OypRtjGYumIgxw7PM-GqO)

[Offizielle SPARQL Dokumentation](https://www.w3.org/TR/rdf-sparql-query)

# Darstellung der Daten

## Begriffe zu den historische Daten

**Observationen**: Die Datenbasis sind Observationen. Diese haben eine festgelegte Struktur von der Form «ZXXXX2015R00011BEVHER0000SEX0000XXX0000XXX0000XXX0000».

**Kennzahl**: Die Kennzahlen repräsentieren Dimensionen, z.B. Dinge wie Bevölkerung, Tierarten. Die Kennzahl ist die primäre Dimension einer Observation. Z.B. «Wohnbevölkerung nach Herkunft, Geschlecht und Stadtquartier im Jahr 2015» hat «Bevölkerung» (BEV) als Kennzahl.

**Gruppen**: Gruppen sind Dimensionen, die beschreiben, wie eine Kennzahl in Untermengen aufzuteilen ist, z.B. «Geschlecht» oder «Tiergattung». Im Datensatz «Wohnbevölkerung nach Herkunft, Geschlecht und Stadtquartier im Jahr 2015» entsprechen Herkunft (HER) und Geschlecht (SEX) Gruppen. Raum und Zeit entsprechen zwar auch Dimensionen, nehmen aber spezielle Rollen ein bzw. sind in einer Observation immer enthalten. Nicht vorhandene Dimensionen werden mit «XXX» angegeben.

**GruppenCode**: Gruppencodes sind Facetten und immer abhängig von einer Dimension. Sie definieren das aktuelle Subset, das die Dimension beschreibt. Z.B. «0001» für «weiblich». Leere GruppenCodes werden mit «0000» angegeben.

In der folgenden Abbildung sieht man einen Vergleich der statistischen Daten aus dem Jahre 1941 und 2016 vor der Digitalisierung. Solche Daten wurden in einem ersten Schritt in Observationen umgewandelt wie zuvor beschrieben. In einem zweiten Schritt werden die Observationen auf das sogenannte RDF Data Cube Modell abgebildet.

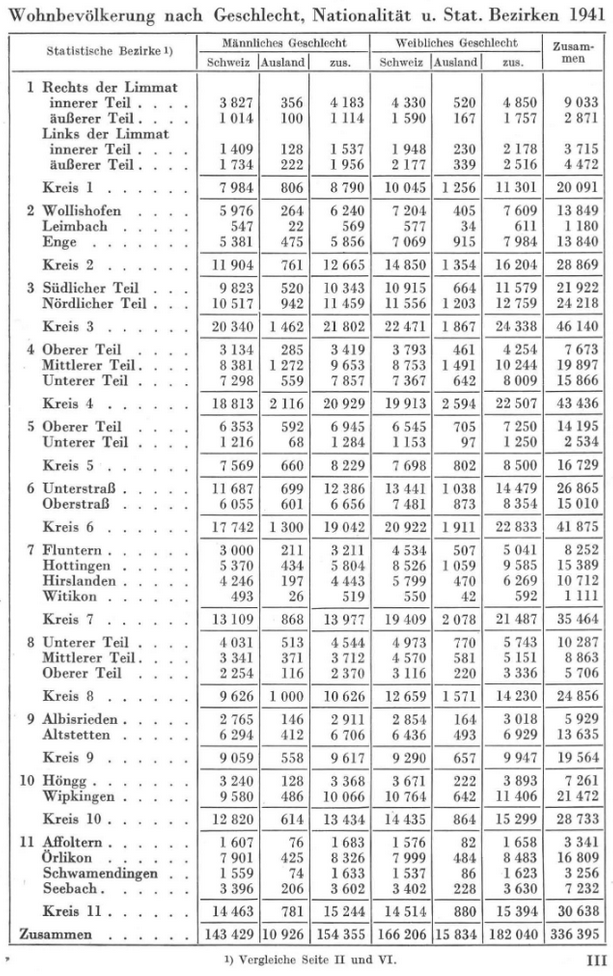
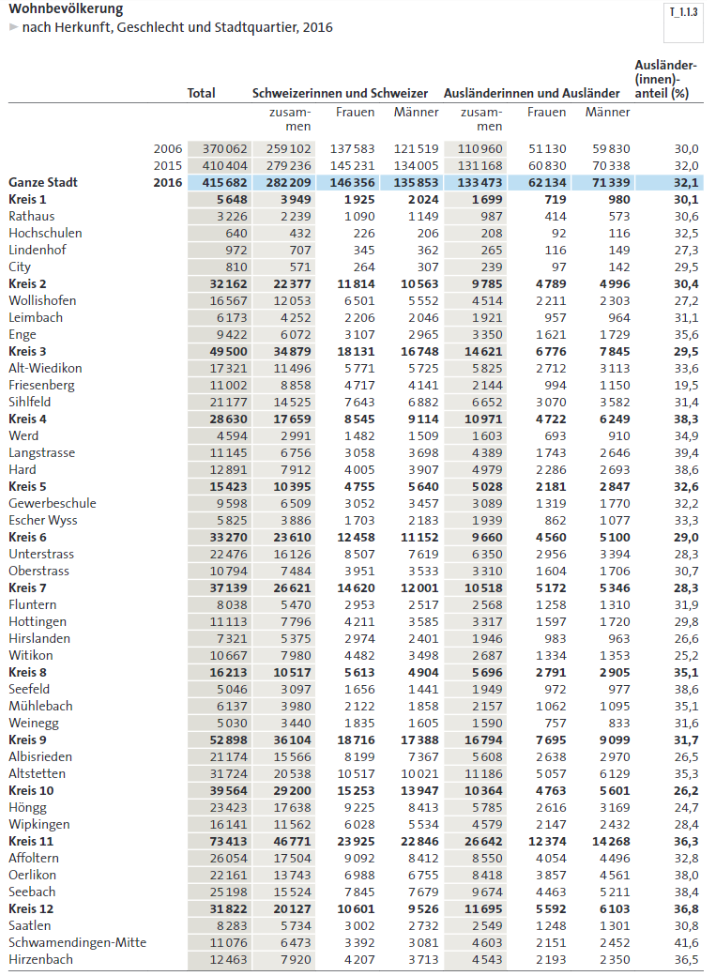


Abbildung : Vergleich zweier Ausschnitte aus dem statistischen Jahrbuch des Jahres 1941 und 2016.

## Begriffe des RDF Data Cube Modells

Das Modell beschreibt einen mehrdimensionalen Datenwürfel mithilfe des Resource Description Frameworks (RDF) und folgenden Grundbegriffen:

**DataSet**: Sammlung von Daten bzw. Observationen, welche zu bestimmten Punkten von bestimmten Dimensionen gemacht wurden (zeitlich, örtlich, nach Geschlecht, nach Einkommen etc.).

**Dimension**: Charakterisiert wie sich die Observationen verändern bzw. nach was sie gemessen wurden (pro Jahr, nach Bezirk, nach Herkunft etc.).

* In den historischen Daten entsprechen die Gruppen der DimensionProperty

**Observation**: Die eigentlichen Daten bzw. gemessenen Werte eines DataSets.

**Measure**: Beschreibt was genau eine Observation misst. Im Observations-Beispiel aus vorigen Kapitel würde dies der Bevölkerung entsprechen (in Anzahl Menschen gemessen).

* In den historischen Daten entsprechen die Kennzahlen der MeasureProperty.

**Attribute**: Attribute beschreiben die observierten Werte und helfen diese zu interpretieren. Sie können z.B. aus Metadaten bestehen (Aktualisierungsdatum etc.) oder die Einheiten der Messwerte spezifizieren, die Skalierung der Daten etc.

**Slice**: Eine Teilmenge von Observationen eines DataSets. Das heisst, für eine oder mehrere Dimensionen wird ein konstanter Wert gesetzt, z.B. alle Observationen eines bestimmen Jahres etc.

**Shapes**: Definieren verschiedene Randbedingungen oder Einschränkungen für die Komponenten eines RDF Graphen, bzw. in unserem Fall eines RDF Data Cubes. Shapes werden in der Shapes Constraint Language (SHACL) definiert und stellen einen Validierungs-Mechanismus dar. Beispiele solcher Randbedingungen sind:

* Ein DataSet muss mehrere Slices enthalten und die Slice Struktur ist nach Jahr
* Observationen müssen mit einem Ort verknüpft sein
* Observationen messen einen float Wert
* Observationen beziehen sich auf einen Ort und ein Jahr
* etc.

Mehr zu Shapes in der [W3C Dokumentation](https://www.w3.org/TR/shacl/).

Eine Übersicht des RDF Data Cube Modells ist in nachfolgender Abbildung zu finden. Neben der [W3C Dokumentation](https://www.w3.org/TR/vocab-data-cube/) gibt es noch [hier](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=11&ved=0ahUKEwjN68WrvprcAhVC8RQKHd50D2M4ChAWCDMwAA&url=https%3A%2F%2Fweb.imsi.athenarc.gr%2Fredmine%2Fattachments%2Fdownload%2F1018%2FData%2520Cube%2520Vocabulary%2520Overview%2520and%2520Example.pdf&usg=AOvVaw3ujILUzk2T1dLd9ebo2hYW) eine gute Einführung.

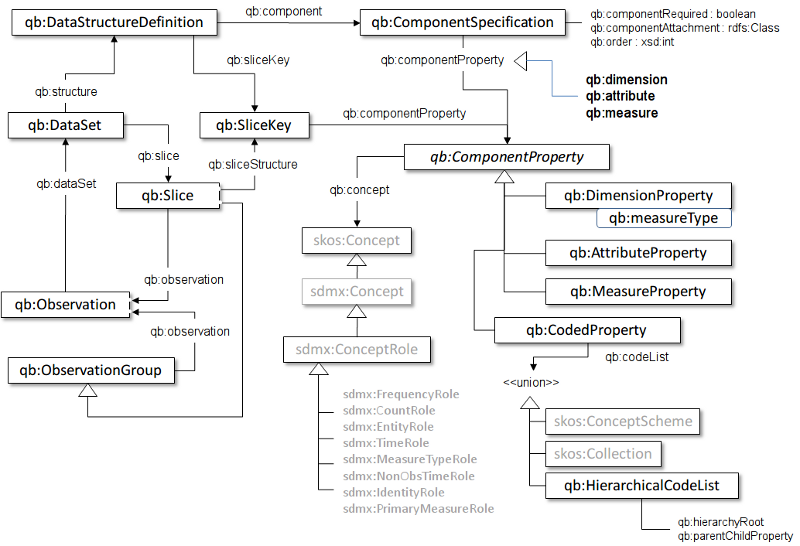


Abbildung 2: Diagramm des RDF Data Cube Modells © 2012-2014 W3C.

# SPARQL

Das SPARQL Interface ist über <http://ld.stadt-zuerich.ch/sparql/> exponiert.

Die effektiven Abfragen lassen sich machen über <http://ld.stadt-zuerich.ch/query>

Die Integrationsumgebung ist auf <http://ld.integ.stadt-zuerich.ch> zugänglich. Es gilt jedoch zu beachten, dass die URI auch in der Integration tendenziell wieder in die Produktion zeigen, da dies im Triple-Store so gemacht wird. Beim Testen ist von Hand oder programmatisch das „.integ“ einzufügen.

## Beispiel Query: Alle DataSets

### Abfrage

|  |
| --- |
| PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  SELECT ?dataset ?label WHERE{  ?dataset a qb:DataSet;  rdfs:label ?label.  } |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

Alle verschiedenen DataSet (gemäss RDF Data Cube Definition) und ihre label werden angezeigt. Diese Abfrage ist hilfreich um sich einen Überblick über die angebotenen Daten zu verschaffen und deren Aufteilung. Man könnte die Abfrage z.B. ergänzen mit einem FILTER um alle möglichen DataSets einer bestimmten Kennzahl (MeasureProperty) oder mit bestimmten Dimensionen (DimensionProperty) zu erhalten.

## Beispiel Query: Bestimmte DataSets

### Abfrage

|  |
| --- |
| PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  SELECT ?dataset ?label WHERE{  ?dataset a qb:DataSet;  rdfs:label ?label.  FILTER REGEX(STR(?label), "^Zuz").  } |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

Hier wurden all diese DataSets abgefragt, welche als Kennzahl bzw. MeasureProperty «Zuzüge» enthalten. Details zu den regular expressions oder REGEX sind [hier](https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference) zu finden. Auf dieselbe Weise kann man auch nach anderen Eigenschaften filtern wie z.B. Dimensionen oder Attribute.

## Beispiel Query: Die ersten Hundert Triples

Wer gerne die Rohdaten im Triplestore anschauen will, kann dies mit der folgenden Abfrage tun.

### Abfrage

|  |
| --- |
| PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  SELECT \* WHERE {  ?sub ?pred ?obj.  }  LIMIT 100 |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Ausschnitt Resultat i JSON

|  |
| --- |
| **{**  "head" **:** **{**  "vars" **:** **[**  "sub"**,**  "pred"**,**  "obj"  **]**  **},**  "results" **:** **{**  "bindings" **:** **[**  **{**  "sub" **:** **{**  "type" **:** "uri"**,**  "value" **:** "http://ld.stadt-zuerich.ch/statistics/property/NAF"  **},**  "pred" **:** **{**  "type" **:** "uri"**,**  "value" **:** "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"  **},**  "obj" **:** **{**  "type" **:** "uri"**,**  "value" **:** "http://purl.org/linked-data/cube#DimensionProperty"  **}**  **},**  **{**  "sub" **:** **{**  "type" **:** "uri"**,**  "value" **:** "http://ld.stadt-zuerich.ch/statistics/property/NAF"  **},**  "pred" **:** **{**  "type" **:** "uri"**,**  "value" **:** "http://www.w3.org/2004/02/skos/core#notation"  **},**  "obj" **:** **{**  "type" **:** "literal"**,**  "value" **:** "NAF"  **}**  **},**  **{**  "sub" **:** **{**  "type" **:** "uri"**,**  "value" **:** "http://ld.stadt-zuerich.ch/statistics/property/NAF"  **},**  "pred" **:** **{**  "type" **:** "uri"**,**  "value" **:** "http://www.w3.org/2000/01/rdf-schema#label"  **},**  "obj" **:** **{**  "type" **:** "literal"**,**  "value" **:** "Namenfolge"  **}**  **},**  **.**  **.**  **.** |

### Erläuterung

In fast jedem praktischen Sinn macht diese Abfrage keinen Sinn, aber sie vermittelt ein Gefühl für die Triples bzw. für das Mapping. Man erkennt z.B., dass «property» aus den statistischen Daten auf «DimensionProperty» des RDF Data Cube Modells abgebildet wurde. Oder, dass «code» aus den statistischen Daten im SKOS Modell einem «Concept» entspricht etc. Das LIMIT statement am Schluss der Abfrage ist selbstverständlich optional und dient nur dazu, die Anzahl Resultate zu beschränken.

## Beispiel Query: Die Anzahl Triple im Triple Store

Wir möchten gerne wissen, wie viele Triple im Store drin sind.

### Abfrage

|  |
| --- |
| PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  SELECT (COUNT(\*) AS ?count) WHERE{  ?sub ?pred ?obj.  } |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

In fast jedem praktischen Sinn macht diese Abfrage keinen Sinn.

## Beispiel: Anzahl Observationen pro Dataset für alle Datesets

Mit der folgenden Abfrage wird festgestellt, welche Datasets wie gross sind.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX qb: <http://purl.org/linked-data/cube#>  SELECT ?dataset (COUNT(DISTINCT ?observationIRI) AS ?observationcount) ?label WHERE{  ***#Alle Observationen eines DataSets***  ?observationIRI a qb:Observation;  qb:dataSet ?datasetIRI.    ?datasetIRI rdfs:label ?label.    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?datasetIRI), STR(<dataset/>)) AS ?dataset).  }  GROUP BY ?dataset ?label  ORDER BY DESC(?observationcount) |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

Die Darstellung zeigt, dass es für viele Anwendungen keine gute Idee ist, das Dataset GEB-RAUM-ZEIT-NAF-NAM-SEX direkt zu laden.

## Beispiel: Suche

Wenn gesucht werden soll, wo welche Begriffe vorkommen, so kann dies mit der untenstehenden Abfrage erfolgen. Wildcards werden mit “\*“ im REGEX statement unterstützt. Zum Suchen wird das Label der jeweiligen Observation verwendet.

### Abfrage

|  |
| --- |
| PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX skos: <http://www.w3.org/2004/02/skos/core#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  SELECT \* WHERE{  GRAPH <https://linked.opendata.swiss/graph/zh/statistics>  {  ?obs rdfs:label ?label.    ***# Wo "\*Tiere\*" steht, Suchbegriff eingeben. \* lassen.***  FILTER REGEX(?label, "Tiere\*")  }  }  LIMIT 100 |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

Es gibt verschiedene Typen von Antworten:

* Topic (z.B. <http://stat.stadt-zuerich.ch/topic/WIR009> Tierhaltung der Landwirtschaftsbetriebe)
* Dataset (z.B. <http://ld.stadt-zuerich.ch/statistics/dataset/AST-RAUM-ZEIT-BTA-TIG/ASTBTA9200TIG9000/slice> Anzahl Ladwirtschaftsbetriebe mit übrigen Tierarten)
* Measure (z.B. <http://ld.stadt-zuerich.ch/statistics/measure/TII> Tierindividuen)
* Code (z.B. <http://ld.stadt-zuerich.ch/statistics/code/PAT1200> Tierpartei Schweiz (TPS))
* Property (z.B. <http://ld.stadt-zuerich.ch/statistics/property/TIG> Tiergattung)
* Application externe Applikationen
* Glossar

## Beispiel: Suche nach Fläche ohne Wald für ganz Zürich

Bei dieser etwas komplexeren Abfrage soll die Enwicklung der Grundstücksfläche ohne Wald grafisch dargestellt werden. Dazu wird das entsprechende Dataset „Grundstückfläche nach Bodenbedeckungsart, Raum und Zeit“ ausgewählt. Dieses hat die Kennzahl bzw. MeasureProperty STF und die Dimensionen RAUM, ZEIT und BBA. Dann werden von den Observationen jeweils die Dimensionen ausgewählt, für welche die Bodenbedeckungsart «ohne Wald» entspricht und RAUM ganz Zürich. Um das ganze sauber in einer Google Chart darzustellen muss noch die Jahreszahl vom Datum extrahiert werden.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX skos: <http://www.w3.org/2004/02/skos/core#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  SELECT ?jahr ?flaeche WHERE{  ***#Das fragliche Dataset auswählen***  ?observation qb:dataSet <dataset/STF-RAUM-ZEIT-BBA>;  <measure/STF> ?flaeche;  <property/ZEIT> ?datum;  <property/BBA>/skos:notation ?bodenbedeckungsart;  <property/RAUM>/skos:notation ?raum.    ***#Datum präparieren für die Darstellung mit Google Chart***  BIND(SUBSTR(STR(?datum),1,4) AS ?jahr).    ***#Filtern nach Landflaeche ohne Wald in ganz Zürich***  FILTER(?raum IN ("R10000")).  FILTER(?bodenbedeckungsart IN ("BBA1000")).  }  ORDER BY ?jahr |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

R10000 – ganz Zürich

BBA: Bodenbedeckungsart (http://ld.stadt-zuerich.ch/statistics/property/BBA )

BBA1000 – Landfläche ohne Wald (http://ld.stadt-zuerich.ch/statistics/code/BBA1000)

Ha – Hektar (Alle Masse werden auf Basisdefinitionen zurückgeführt. In diesem Fall über xxx)

Die Zeit muss jeweils auf saubere Jahre zurückgebunden werden für diesen Fall.

## Beispiel: Alle Codes

Alle möglichen Codes können abgefragt werden.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX skos: <http://www.w3.org/2004/02/skos/core#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  SELECT ?code ?label WHERE{  ***#Die Codes haben den Typ "Concept" des RFD Data Cube Modells***  ?codeIRI a skos:Concept;  rdfs:label ?label.    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?codeIRI), STR(<code/>)) AS ?code).    ***#Für Vergleich mit http://ld.stadt-zuerich.ch/statistics/code/***  ***#FILTER(REGEX(?code, "BTA1742") || REGEX(?code, "VOR1006")).***  }  ORDER BY ?codeIRI |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

## Beispiele: Hierarchien

In den Codes kommen Hierarchien vor. Diese können abgefragt werden.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX skos: <http://www.w3.org/2004/02/skos/core#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  SELECT ?hierarchy ?code ?label WHERE{  ***#Die Codes haben den Typ "Concept" des RFD Data Cube Modells***  ?codeIRI a skos:Concept;  rdfs:label ?label;  skos:broader ?hierarchyIRI.    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?codeIRI), STR(<code/>)) AS ?code).  BIND(STRAFTER(STR(?hierarchyIRI), STR(<code/>)) AS ?hierarchy).  }  GROUP BY ?hierarchy ?code ?label  ORDER BY ?code |

### Ausschnitt Resultat

|  |
| --- |
|  |

## Beispiel: Hierarchie des Raums

Die Codes sind nach Schema aufgeteilt. Das heisst, um z.B. alle Raum Codes zu finden, muss man zusätzlich das Schema eingrenzen.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX skos: <http://www.w3.org/2004/02/skos/core#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  SELECT ?hierarchy ?code ?label WHERE{  ***#Die Codes haben den Typ "Concept" des RFD Data Cube Modells***  ?codeIRI a skos:Concept;  rdfs:label ?label;  skos:broader ?hierarchyIRI.    ***#Alle Raum Codes***  ?codeIRI skos:inScheme <scheme/Raum>.    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?codeIRI), STR(<code/>)) AS ?code).  BIND(STRAFTER(STR(?hierarchyIRI), STR(<code/>)) AS ?hierarchy).    ***#Für Vergleich mit http://ld.stadt-zuerich.ch/statistics/code/R00012***  ***#FILTER REGEX(?code, "R00012").***  }  GROUP BY ?hierarchy ?code ?label  ORDER BY ?code |

### Ausschnitt Resultat

|  |
| --- |
|  |

|  |
| --- |
|  |

### Erläuterung

Raum ist komplex organisiert. Es gibt mehrere Roots

* R10000 – Kreis 1
* R20000 – Stadt nach 1. Eingemeindung 1893
* R30000 - Stadt Zürich

## Beispiel: Alle Arten von Kennzahlen

Mit dieser Abfrage sollen alle möglichen Kennzahlen gefunden werden. Da diese im RDF Data Cube auf MeasureProperties abgebildet wurden, lassen sie entsprechend einfach finden. Zusätzlich werden die Einheiten und Kommentare ausgegeben.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX owl: <http://www.w3.org/2002/07/owl#>  PREFIX qb: <http://purl.org/linked-data/cube#>  SELECT ?measure ?label ?range ?comment WHERE {  ?measureIRI a qb:MeasureProperty;  rdfs:label ?label;  rdfs:range ?rangeIRI;  rdfs:comment ?comment.    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?measureIRI), STR(<measure/>)) AS ?measure).  BIND(STRAFTER(STR(?rangeIRI), STR(<http://www.w3.org/2001/XMLSchema#>)) AS ?range).  }  ORDER BY ?measure |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

## Beispiel eines Datensatzes

Mit folgender Abfrage lassen sich alle Dimensionen, Attribute und Measures eines DataSets anzeigen. Da die DataSets benannt sind nach dem Schema “Kennzahl-Dimension1-Dimension2.. “, liefert diese Abfrage die erwarteten dimensions und das erwartete measure. Die Kennzahl wird ja auf die qb:MeasureProperty des RFD Data Cube Modells abgebildet. Die Attribute sind jeweils Metadaten wie z.B. Fussnote, Quelle und Aktualisierungsdatum.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX owl: <http://www.w3.org/2002/07/owl#>  ***#Ein einzelnes DataSet auswählen mit seinen Eigenschaften***  SELECT ?dataset ?dimension ?attribute ?measure WHERE{  ***#Es soll nur ein DataSets ausgewählt werden von Zeile "OFFSET"***  {  SELECT ?datasetIRI WHERE{  ?datasetIRI a qb:DataSet.  }  ORDER BY ?datasetIRI  OFFSET 1  LIMIT 1  }    ***#Von dem ausgewählten DataSet werden die Eigenschaften gesucht***  ?datasetIRI qb:structure/qb:component [qb:dimension ?dimensionIRI; qb:attribute ?attributeIRI; qb:measure ?measureIRI].    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?datasetIRI), STR(<dataset/>)) AS ?dataset).  BIND(STRAFTER(STR(?dimensionIRI), STR(<property/>)) AS ?dimension).  BIND(STRAFTER(STR(?attributeIRI), STR(<attribute/>)) AS ?attribute).  BIND(STRAFTER(STR(?measureIRI), STR(<measure/>)) AS ?measure).  }  ORDER BY ?dimension |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

## Abfrage der Observationen

Mit dieser Abfrage werden alle möglichen Observationen des DataSets aus letztem Beispiel ausgegeben.

### Abfrage

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX qb: <http://purl.org/linked-data/cube#>  SELECT ?dataset ?observation WHERE{  {  SELECT ?datasetIRI WHERE{  ?datasetIRI a qb:DataSet.  }  ORDER BY ?datasetIRI  OFFSET 1  LIMIT 1  }  ?observationIRI qb:dataSet ?datasetIRI.    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?datasetIRI), STR(<dataset/>)) AS ?dataset).  BIND(STRAFTER(STR(?observationIRI), STR(<observation/>)) AS ?observation).  }  LIMIT 100 |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Abfrage der Slices

|  |
| --- |
| BASE <http://ld.stadt-zuerich.ch/statistics/>  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  PREFIX qb: <http://purl.org/linked-data/cube#>  SELECT ?dataset ?slice WHERE{  {  SELECT ?datasetIRI WHERE{  ?datasetIRI a qb:DataSet.  }  ORDER BY ?datasetIRI  OFFSET 1  LIMIT 1  }  ?datasetIRI qb:slice ?sliceIRI.    ***#Der Anschaulichkeit halber werden die PREFIXes nicht angezeigt***  BIND(STRAFTER(STR(?datasetIRI), STR(<dataset/>)) AS ?dataset).  BIND(STRAFTER(STR(?sliceIRI), STR(<dataset/>)) AS ?slice).  } |

### Ausschnitt Resultat

|  |
| --- |
|  |

## Beispiel: Verknüpfung von Kennzahlen

### Erläuterung

Dank der Datenharmonisierung wird die Verknüpfbarkeit der Daten wesentlich vereinfacht. Als Beispiel soll hier die Differenz aus Geburten und Sterbefällen berechnet werden.

### Abfrage

|  |
| --- |
| PREFIX qb: <http://purl.org/linked-data/cube#>  PREFIX dataset: <http://ld.stadt-zuerich.ch/statistics/dataset/>  PREFIX measure: <http://ld.stadt-zuerich.ch/statistics/measure/>  PREFIX property: <http://ld.stadt-zuerich.ch/statistics/property/>  PREFIX code: <http://ld.stadt-zuerich.ch/statistics/code/>  SELECT ?Jahr (SUM(?Geburten) AS ?Geburten) (SUM(?Sterbefaelle) AS ?Sterbefaelle) (SUM(?Differenz) AS ?Differenz) WHERE {  ?observation a qb:Observation;  property:RAUM ?Raum;  property:ZEIT ?Zeit.    ***#Geburten extrahieren aus entsprechendem Dataset***  {  ?observation qb:dataSet dataset:GEB-RAUM-ZEIT;  measure:GEB ?Geburten.  BIND(?Geburten AS ?Differenz)  }    UNION    ***#Sterbefälle extrahieren aus entsprechendem Dataset***  {  ?observation qb:dataSet dataset:GES-RAUM-ZEIT;  measure:GES ?Sterbefaelle.  BIND(?Sterbefaelle\*-1 AS ?Differenz)  }    ***#RAUM soll ganze Stadt umfassen***  FILTER(?Raum = code:R30000)    ***#Datum präparieren für Google Chart***  BIND(STR(?Zeit) AS ?Jahr)  }  GROUP BY ?Jahr  ORDER BY ?Jahr |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

## Beispiel: Umgang mit Zeit

Es gibt Datensätze, die monatlich sind und solche, die jährlich sind.

## Grundlagen SPARQL

<https://de.slideshare.net/OpenDataSupport/d211-training-module-13-introduction-to-rdf-sparql-v03-de>

# Zugang mittels REST-Interface

## Grundlegendes

Startpunkt ist: <http://stat.stadt-zuerich.ch/>

Selbsterklärung des API: <http://stat.stadt-zuerich.ch/api>

Properties:

<http://ld.stadt-zuerich.ch/statistics/property/ALT>

Observation:

<http://ld.stadt-zuerich.ch/statistics/observation/BEW/R00011/Z31121983/HEL1000/SEX0001>

Alter

## Search-API

<http://stat.stadt-zuerich.ch/api/tags/?query=Be&format=json>

(BE) ist dein Teil des Worts

Antworten:

* Topic 🡨 welche anderen gibt es
* Dimension
* Raum
* Zeit
* (Application)

## Alle Dataset

### Abfrage

|  |
| --- |
| http://ld.stadt-zuerich.ch/statistics/dataset/ |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

Mit dieser Abfrage können alle Dataset geholt werden. Angegeben werden die Datasets und die Slices. Slices sind untenstehend beschrieben.

Das Dataset selbst und das Shape können separat aufgerufen werden (siehe folgende Abschnitte).

## Dataset

### Abfrage

|  |
| --- |
| http://ld.stadt-zuerich.ch/statistics/dataset/BEW-RAUM-ZEIT |

### Ausschnitt Resultat

|  |
| --- |
|  |

### Erläuterung

## Slice

Ein Slice enthält die eigentlichen Daten des Dataset. Es ist stellt dabei eine erlaubte Definition der Kombinationen darf

### Abfrage

|  |
| --- |
| <http://stat.stadt-zuerich.ch/api/dataset/BEW-RAUM-ZEIT/slice?format=jasonld> |

### Ausschnitt Resultat

|  |
| --- |
|  |

|  |
| --- |
| {"@id":"http://stat.stadt-zuerich.ch/api/dataset/BEW-RAUM-ZEIT/slice","@type":"Slice","observation":[{"@id":"ssz-observation:BEW/R30000/Z31121966","@type":"Observation","mBEW":"4.33234E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1966-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121966R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31122005","@type":"Observation","mBEW":"3.66809E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"2005-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31122005R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31122011","@type":"Observation","mBEW":"3.90082E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"2011-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31122011R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121947","@type":"Observation","mBEW":["3.76564E5","3.77087E5"],"dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1947-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121947R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121996","@type":"Observation","mBEW":"3.59648E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1996-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121996R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121961","@type":"Observation","mBEW":"4.42314E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1961-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121961R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121983","@type":"Observation","mBEW":"3.63652E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1983-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121983R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121984","@type":"Observation","mBEW":"3.61493E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1984-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121984R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121954","@type":"Observation","mBEW":"4.14011E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1954-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121954R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121957","@type":"Observation","mBEW":"4.28174E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1957-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121957R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121995","@type":"Observation","mBEW":"3.60826E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1995-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121995R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121949","@type":"Observation","mBEW":["3.86003E5","3.8528E5"],"dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1949-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121949R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121986","@type":"Observation","mBEW":"3.58873E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1986-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121986R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31122010","@type":"Observation","mBEW":"3.85468E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"2010-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31122010R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121985","@type":"Observation","mBEW":"3.59084E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1985-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121985R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121965","@type":"Observation","mBEW":"4.3531E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1965-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121965R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121967","@type":"Observation","mBEW":"4.32547E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1967-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121967R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121981","@type":"Observation","mBEW":"0.0E0","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1981-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121981R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121997","@type":"Observation","mBEW":"3.58594E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1997-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121997R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31121953","@type":"Observation","mBEW":"4.0926E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1953-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121953R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31122004","@type":"Observation","mBEW":"3.64977E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"2004-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31122004R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"},{"@id":"ssz-observation:BEW/R30000/Z31122013","@type":"Observation","mBEW":"3.98575E5","dRAUM":"ssz-code: |

### Erläuterungen

Wichtig sind das Label und die relevante Dimensionskombination.

Beim Beispiel:

* BEW als Unterkombination
* Label: Wirtschaftliche Bevölkerung

Ein Datensatz hat ein DefaultSlice und kann mehrere zusätzliche Slices enthalten.

Im Falle des Dataset BEW-RAUM-ZEIT sind dies:

* /slice – Default
* /BEW/slice

Die folgenden Formate können abgefragt werden:

* Json
* Jsonld
* Csv
* Excel
* odf

Der Inhalt sind einzelne Observationen.

Es gilt zu beachten, dass Jahreszahlen immer mit dem 31.12. angegeben sind.

|  |
| --- |
| {"@id":"ssz-observation:BEW/R30000/Z31121997","@type":"Observation","mBEW":"3.58594E5","dRAUM":"ssz-code:R30000","dZEIT":{"@type":"http://www.w3.org/2001/XMLSchema#date","@value":"1997-12-31"},"dataSet":"ssz-dataset:BEW-RAUM-ZEIT","notation":"Z31121997R30000BEWXXX0000XXX0000XXX0000XXX0000XXX0000"} |

## Shape

Die Metadaten eines Dataset sind im Shape beschrieben.

### Abfrage

<http://ld.stadt-zuerich.ch/statistics/dataset/BEW-RAUM-ZEIT/shape?format=jsonld>

### Ausschnitt des Resultats

|  |
| --- |
| [ {  "@graph" : [ {  "@id" : "http://ld.stadt-zuerich.ch/statistics/dataset/BEW-RAUM-ZEIT/shape",  "@type" : [ "http://www.w3.org/ns/shacl#NodeShape" ],  "http://www.w3.org/2002/07/owl#sameAs" : [ {  "@id" : "http://stat.stadt-zuerich.ch/api/dataset/BEW-RAUM-ZEIT/shape"  } ],  "http://www.w3.org/ns/shacl#property" : [ {  "@id" : "\_:BEW2dRAUM2dZEITRAUM"  }, {  "@id" : "\_:BEW2dRAUM2dZEITZEIT"  }, {  "@id" : "\_:BEW2dRAUM2dZEITBEW"  } ],  "http://www.w3.org/ns/shacl#targetNode" : [ {  "@id" : "http://stat.stadt-zuerich.ch/api/dataset/BEW-RAUM-ZEIT/slice"  } ]  } ],  "@id" : "https://linked.opendata.swiss/graph/zh/statistics"  } ] |

### Erläuterungen

Dieses Shape wird nur für den Hydra-Client verwendet.

## Darstellung verschiedener Metadaten

### Themenbaum

### Raumaufteilung

### Seiteninhalte für statische Seiten

### Tags mit Einschränkungen

### Metadaten der statistischen Daten

* Fussnoten
* Glossar
* Quellenangaben
* Aktualisierungsverlauf
* Erhebungsmethode
* Lizenzbestimmungen

### Verwandte Datensätze eines Datensatzes (basierend auf Themenbaum)

### Anwendungen in der Suche und separat

## API-Tests

API-Tests finden sich auf:

<https://sszvis-components.netlify.com/#/api-test>

# Zugang mit Hydra-Client

LATER

# Verwendung von sszvis

* <https://www.npmjs.com/package/sszvis>
* <https://github.com/statistikstadtzuerich/sszvis-components>
* <https://github.com/statistikstadtzuerich/stip-frontend>
* <http://stip-frontend-docs.interactivethings.io/>
* <http://www.catalog.style/>
* Die Datenstrukturen sind hier definiert: <https://sszvis-components.netlify.com/#/chart-system>

# OpenSource Projekte zu STIP, LOSD und sszvis

LATER