

# Recherche nach Musik-Events

Semantic Web - Projektdokumentation

Phil Taubert | 68246

HTWK-Leipzig | 16MIM

SS 207 | 31.07.2017

### Recherchefragestellung:

Welche Festivals und Konzerte mit Interpreten aus dem Genre Alternative Rock und einer maximalen Anfahrtszeit von zwei Stunden ab Leipzig finden in den Semesterferien nach dem Sommersemester 2017 (30.07.2017 bis 24.09.2017) statt?

## Inhaltliche Interpretation der Fragestellung

Im Modul Semantic Web im Sommersemester 2017 soll eine frei wählbare Fragestellung durch die Kombination von Informationen aus verschiedenen Datenquellen aus dem Internet beantwortet werden.

Im Rahmen dieses Projekts wird nach Veranstaltungen mit diversen Kriterien gesucht. Diese Bedingungen lauten:

- Art: Musikveranstaltungen (Konzerte und Festivals)
- Zeitraum: Semesterferien nach dem Sommersemester 2017  
(30.07.2017 bis 24.09.2017)
- Genre: Interpreten aus dem Bereich "Alternative Rock"
- Ort: Deutschland, maximale Anfahrtszeit von zwei Stunden  
(ab Leipzig)

Als Ergebnis wird eine Auflistung von interessanten und lohnenswerten Musik-Events erwartet.

# Datenquellen

Im Folgenden werden alle zur Beantwortung der Fragestellung verwendeten Datenquellen aufgelistet und beschrieben

## Eventful

Eventful ist ein Online-Service zur Suche nach Veranstaltungen, welcher vom amerikanischen Massenmedienunternehmen CBS Cooperation angeboten wird.<sup>1</sup> Dieser stellt eine API zur Verfügung, worüber sämtliche Events, gefiltert nach bestimmten Kriterien werden können. Eine Beschreibung der Schnittstelle kann unter *api.eventful.com* eingesehen werden.

## Last.fm

Last.fm ist ein soziales Netzwerk mit Ausrichtung auf Musikthemen. Eigentümer ist ebenfalls die CBS Corporation.<sup>2</sup> Auch Last.fm bietet eine Programmierschnittstelle, über welche die gespeicherten Informationen zu allen vorhandenen Interpreten abgerufen werden können. Die Beschreibung der Schnittstelle findet man unter *last.fm/api*.

## Google Directions

Google bietet mit der Directions API eine Schnittstelle, über die Wegbeschreibungen und weitere Navigationsinformationen abgerufen werden. Weitere Informationen erhält man unter *developers.google.com/maps/documentation/directions*.

---

<sup>1</sup> <https://en.wikipedia.org/wiki/Eventful>

<sup>2</sup> <https://de.wikipedia.org/wiki/Last.fm>

# Abruf der Daten

## Events

Zuerst sollten alle Daten sämtlicher Events von Eventful abgerufen werden. Für sämtliche Abfragen der API benötigt man einen gültigen Application Key. Unter <http://api.eventful.com/keys/> kann man einen neuen Key beantragen, indem man seine Kontaktinformationen und eine Beschreibung der Anwendung, welche den Key benutzt, hinterlegt. Danach erhält man direkt einen Application Key, mit welchem man im angegebenen Rahmen unbegrenzt Daten abfragen kann.

Für die Abfrage der Events mit Locations und Künstlern wurde die Suchfunktion der API verwendet. Diese kann über den Pfad `/events/search` erreicht werden. Hier müssen neben dem Application Key weitere Filterkriterien angegebenen werden. Im Rahmen dieses Projekts wurde nach Events in der Kategorie "Musik" im oben angegebenen Zeitraum in Deutschland gesucht. Eine erste Abfrage der Schnittstelle lieferte eine Gesamtzahl von 3003 Events sowie die erste Seite der Suchergebnisse im XML-Format.

Bei der maximalen Anzahl von 250 Einträgen pro Seite müssen also 13 Seiten abgerufen werden. Die Ergebnisse wurden über die Schleife in folgendem Windows Batch-Script in einen Ordner gespeichert:

```
for /l %%x in (1, 1, 13) do (
    echo %%x
    powershell -Command "Invoke-WebRequest
    http://api.eventful.com/rest/events/search
        ?app_key=cDjpmDPHhGGft89j '
        &'location=Germany'
        &'date=2017073000-2017092400 '
        &'category=music'
        &'page_size=250'
        &'page_number=%%x
    -OutFile
    D:\Benutzer\philt\Documents\HTWK\SemanticWeb\data\eventf
    ul\page%%x.xml"
)
```

Die resultierenden XML-Dateien beinhalten sämtliche Musikevents mit Austragungsorten und auftretenden Künstlern in der folgenden Struktur (gekürzt):

```
<total_items>3003</total_items>
<page_size>250</page_size>
<page_count>13</page_count>
<page_number>1</page_number>
<search_time>0.167</search_time>
<events>
  <event id="E0-001-096844043-1">
<title>Glenn Miller Orchestra - directed by Wil
  Salden</title>
    <description/>
    <start_time>2017-08-05 19:30:00</start_time>
    <venue_name>Haus des Gastes Höchenschwand</venue_name>
    <venue_address>Dr.-Rudolf-Eberle-Straße 3</venue_address>
    <city_name>Höchenschwand</city_name>
    <region_name>Baden-Württemberg</region_name>
    <region_abbr>BW</region_abbr>
    <postal_code/>
    <country_name>Germany</country_name>
    <country_abbr2>DE</country_abbr2>
    <country_abbr>DEU</country_abbr>
    <latitude>47.7353</latitude>
    <longitude>8.16585</longitude>
    ...
    <performers>
      <performer>
        <id>P0-001-000115875-6</id>
        <name>Glenn Miller Orchestra</name>
        <short_bio>Big Band</short_bio>
      </performer>
    </performers>
  </event>
  ...
```

## Anfahrtszeiten

In den oben beschriebenen XML Dateien sind zu jedem Event Daten zum entsprechenden Veranstaltungsort hinterlegt. Zu allen vorkommenden Orten sollen nun die Anfahrtszeiten ab Leipzig von der Google Directions API abgerufen werden.

Über die URL <https://maps.googleapis.com/maps/api/directions/xml> liefert die API neben der Wegbeschreibung und der Entfernung auch die geschätzte Anfahrtszeit im XML-Format. Hierbei müssen verschiedene Parameter übergeben werden. Diese sind im Einzelnen:

- Startort (*origin*)
- Zielort (*destination*)
- API-Schlüssel (*key*)

Für dieses Projekt wird der Startort "Leipzig" übergeben.

Als Zielort muss die jeweilige Location aller Events übergeben werden. Hierfür wurde aus den XML-Dateien der Name sowie die Stadt des Events zusammengesetzt und an die API übergeben. Die API suchte dann beispielsweise nach "Täubchenthal Leipzig" und fand in fast allen Fällen die richtige Location.

Unter <https://developers.google.com/maps/documentation/directions/get-api-key> kann man einen API-Schlüssel für ein Projekt anfordern. Das kostenlose Kontingent umfasst 2.500 Anforderungen pro Tag. Insgesamt wurden 413 Veranstaltungsorte extrahiert, sodass diese Grenze nicht überschritten wurde.

Über ein Java-Programm wurden die Orte mit Namen und Stadt aus den Dateien von eventful entnommen.

```
String venue =
    event.getElementsByTagName("venue_name")
        .item(0)
        .getTextContent()
    + " " +
    event.getElementsByTagName("city_name")
        .item(0)
        .getTextContent();
```

Um doppelte Abfragen und Speicherungen zu vermeiden, wurde die resultierende Zeichenkette in Kleinbuchstaben umgewandelt und in einer Liste gespeichert. Nun wird die URL für die API-Abfrage zusammengesetzt und das zurückgegebene Dokument als XML-Datei gespeichert.

```
URL url = new
URL("https://maps.googleapis.com/maps/api/directions/xml
    ?origin=Leipzig
    &destination=" + name.replace(" ", "%20") + "
    &key=[API-Key]");
URLConnection conn = url.openConnection();
conn.connect();
```

Der oben zusammengesetzte Name wurde als Dateiname der XML-Dateien gewählt. Da in den gespeicherten XML-Dateien jedoch nur die Adresse der Orte angegeben war, wurde dieser für eine einfache Verarbeitung in den XML-Baum eingefügt.

```
NodeList leg = doc.getElementsByTagName("leg");
Element performerName = doc.createElement("location");
performerName.setTextContent(file.getName().replace(".xml", ""));
leg.item(0).appendChild(performerName);
```

Zudem wurden die einzelnen Schritte der Wegbeschreibungen aus den Dateien entfernt, da diese so übersichtlicher werden und einfacher verarbeitet werden können. Folgende XML-Struktur wurde nun abgespeichert:

```
...
<duration>
  <value>672</value>
  <text>11 mins</text>
</duration>
<distance>
  <value>4024</value>
  <text>4.0 km</text>
</distance>
<start_location>
  <lat>51.3397681</lat>
  <lng>12.3730411</lng>
</start_location>
<end_location>
  <lat>51.3248045</lat>
  <lng>12.3302090</lng>
</end_location>
<start_address>Leipzig, Germany</start_address>
<end_address>Wachsmuthstraße 1, 04229 Leipzig,
Germany</end_address>
<location>Täubchental</location>
...
```



## Genre der Interpreten

Schließlich müssen noch die Genres der Interpreten abgerufen werden. Bei LastFM sind dies Stichwörter ("Tags"), die von Nutzern gepflegt werden. Aus diesem Grund werden hier für sehr viele Künstler sehr viele Tags gefunden, jedoch können somit auch einige weniger sinnvolle Tags angegeben sein.

Bei der Abfrage der API wurde ähnlich vorgegangen wie bei den Anfahrtszeiten. Mit Java wurden die Interpreten aus den Eventdaten extrahiert.

```
Element performers = (Element)
event.getElementsByTagName("performers").item(0);
NodeList performerList =
    performers.getElementsByTagName("performer");
for (int i = 0; i < performerList.getLength(); i++) {
    Element performer = (Element) performerList.item(i);
    String name =
        performer.getElementsByTagName("name").item(0).getTextContent
        ();
    ...
}
```

Auch die Interpretennamen wurden in Kleinbuchstaben umgewandelt und zwischengespeichert, um Dopplungen zu vermeiden. Nachdem man unter <http://www.last.fm/api/account/create> einen API-Account mit der Beschreibung des Projekts angelegt hat, erhält man einen gültigen API-Key, der in diesem Rahmen für sämtliche Anfragen verwendet werden kann. Nun wurde die folgende Methode der API mit dem Namen des Interpreten und dem API-Schlüssel abgefragt.

```
URL url = new URL("http://ws.audioscrobbler.com/2.0/
    ?method=artist.gettoptags
    &artist=" + name.replace(" ", "%20") + "
    &api_key=[API-Key]");
URLConnection conn = url.openConnection();
conn.connect();
```

Bei der Abfrage fiel auf, dass die Daten der Events unvollständig sind. Nur bei geschätzt ungefähr einem Viertel aller Veranstaltungen waren unter dem Tag *performers* Interpreten angegeben. Aus diesem Grund müssen diese nachträglich so gut wie möglich vervollständigt werden. Hier wurde der Titel der Veranstaltung verwendet, da in diesem häufig der Name des Künstlers in der Form "Thomas Stieler - Think Open Air 2017" stand.

Bei allen Events, bei denen der Künstler nicht eingetragen war, wurde die API mit dem Titel bis zum ersten Bindestrich abgefragt. Wurden Tags zurückgegeben, wurde dieser Künstler in die Event-Datei an die richtige Stelle im XML-Baum geschrieben.

```
Document doc = dBuilder.parse(saveFile);
doc.getDocumentElement().normalize();
NodeList tagList = doc.getElementsByTagName("tag");
if (tagList.getLength() > 0) {
    Element performer = doc.createElement("performer");
    Element performerName = doc.createElement("name");
    performerName.setTextContent(name);
    performer.appendChild(performerName);
    performers.appendChild(performer);
}
```

So wurden zu insgesamt 570 Interpreten Tags abgerufen und gespeichert. Zu jedem Tag ist auch ein Wert "count" angegeben, der als Relevanz interpretiert werden kann, sodass Tags mit einer geringen Relevanz später herausgefiltert werden können.

```
<toptags artist="ABBA">
  <tag>
    <count>100</count>
    <name>pop</name>
    <url>https://www.last.fm/tag/pop</url>
  </tag>
  <tag>
    <count>74</count>
    <name>Disco</name>
    <url>https://www.last.fm/tag/Disco</url>
  </tag>
  ...
</toptags>
```

## Verarbeitung zu RDF

Die zwischengespeicherten XML-Dateien aus den Datenquellen müssen nun aufbereitet und in ein RDF-Format umgewandelt werden. Hierfür wurde Google Refine in der Version 2.5 verwendet, da spätere Versionen diverse Bugs enthalten. Für die Umwandlung in RDF wurde das Refine RDF-Plugin verwendet.<sup>3</sup>

In Refine können sehr einfach XML-Dateien Importiert werden. Nach dem Start des Programms wird ein neues Projekt angelegt, sämtliche Dateien einer Datenquelle ausgewählt und das erste zu parsende Objekt ausgewählt. Im Falle der Events ist dies der erste Block mit einem <event> Tag, da hier mehrere Events in einer Datei gespeichert sind. Für die Anfahrtszeiten und Tags wurden jeweils die kompletten Dateien ausgewählt, da pro Eintrag eine Datei existierte.

Die extrahierten Einträge wurden aufbereitet und für den Export vorbereitet, sodass folgende Tabellen entstanden:

| ▼ Titel   | ▼ Stadt    | ▼ Location            | ▼ Beginn            | ▼ Bundesland        | ▼ Adresse                      | ▼ Interpret                  | ▼ PLZ |
|---|------------|-----------------------|---------------------|---------------------|--------------------------------|------------------------------|-------|
| HILDEGARD VON BINGE DRINKING                                  | Hamburg    | frau hedis tanzkaffee | 2017-08-02 19:30:00 | Hamburg             | 10 Sankt Pauli-Landungsbrücken | hildegard von binge drinking |       |
| Symphonic Rock in Concert - Neue Philharmonie Frankfurt HANAU | Hanau      | amphitheater hanau    | 2017-08-20 19:30:00 | Hessen              | Schloss Philippsruhe           |                              | 63454 |
| Santiano  | Eschweiler | zentrum - marktplatz  | 2017-08-25 00:00:00 | Nordrhein-Westfalen | Marktplatz                     | santiano                     |       |

---

<sup>3</sup> <http://refine.deri.ie/>

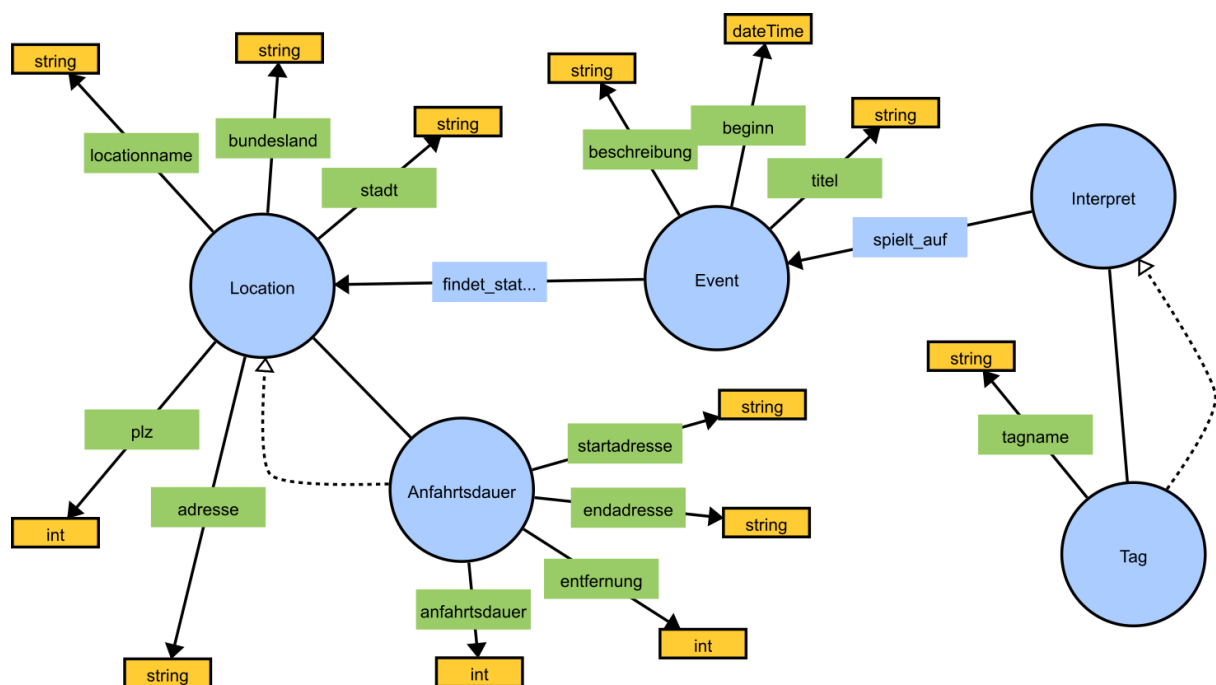
Die Anfahrsdauer ist in Sekunden angegeben. Zusätzlich wird als interessante Information noch die Entfernung in Metern exportiert

| ▼ anfahrtsdauer | ▼ entfernung | ▼ endadresse                                  | ▼ startadresse   | ▼ Location              |
|-----------------|--------------|---|------------------|-------------------------|
| 672             | 4024         | Wachsmuthstraße 1, 04229 Leipzig, Germany     | Leipzig, Germany | täubchental             |
| 740             | 4600         | Klingenstraße 20, 04229 Leipzig, Germany      | Leipzig, Germany | damenhandschuhfabrik    |
| 1015            | 7886         | Riesaer Str. 56 - 64, 04328 Leipzig, Germany  | Leipzig, Germany | sky club                |
| 1003            | 8706         | Gustav-Esche-Straße 4, 04159 Leipzig, Germany | Leipzig, Germany | haus auensee            |
| 3480            | 52178        | 04860 Torgau, Germany                         | Leipzig, Germany | endless summer open air |

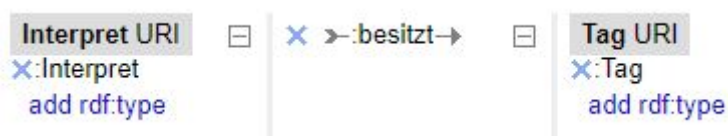
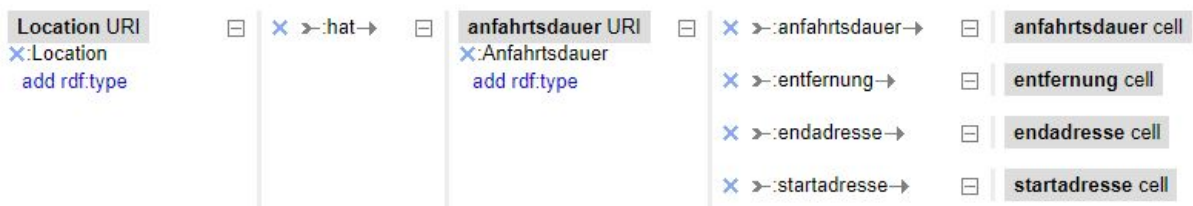
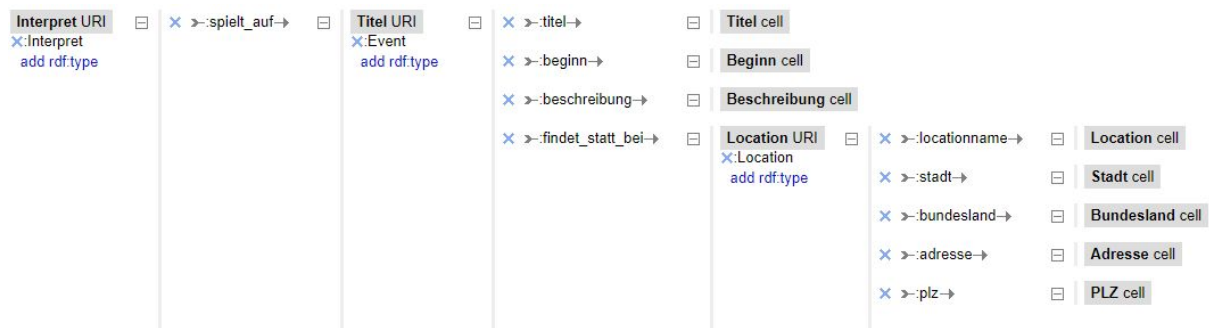
Alle Tags mit einer Relevanz unter 5 wurden herausgefiltert.

| ▼ Interpret         | ▼ Relevanz | ▼ Tag                |
|---------------------|------------|----------------------|
| chris forsyth       | 100        | psychedelic rock     |
| chris forsyth       | 100        | neoclassic rock      |
| 3a                  | 100        | german               |
| 3a                  | 67         | rock                 |
| 3a                  | 67         | pop rock             |
| 3a                  | 34         | pop                  |
| 3a                  | 34         | germany              |
| elbtonal percussion | 100        | nu-jazz              |
| elbtonal percussion | 100        | percussions          |
| elbtonal percussion | 100        | acoustic electronica |

In Protegé wurde nun ein Vokabular erstellt, um das Erstellen des RDF-Skeletts für den Export als RDF-Datei zu vereinfachen. Dabei wurden die benötigten Daten, welche später abgefragt werden sollen, als Klassen deklariert. So werden die Events, Locations, Anfahrtszeiten, Interpreten und Tags später als Objekte gespeichert. Diese werden als Object Properties verbunden: Jeder Interpret spielt auf einem oder mehreren Events, welche bei einer Location stattfinden. Jede Location wiederum hat eine Anfahrtszeit und jeder Interpret besitzt mindestens einen Tag. Zusätzlich wurden noch Data Properties definiert, welche zusätzliche Informationen wie der Beginn eines Events speichern. Diese können dann im Ergebnis der Abfrage mit angezeigt werden.



Nun wurde unter “Edit RDF Skeleton” die Struktur der zu exportierenden RDF-Datei konfiguriert. Dabei wurde mit “add prefix” das aus Protegé abgespeicherte Vokabular importiert, um die Typen der Properties einfach auswählen zu können. Als Base URI wurde die URL des GitLab-Projekts gewählt. In den nachfolgenden Screenshots kann man die RDF-Skeletons sehen:

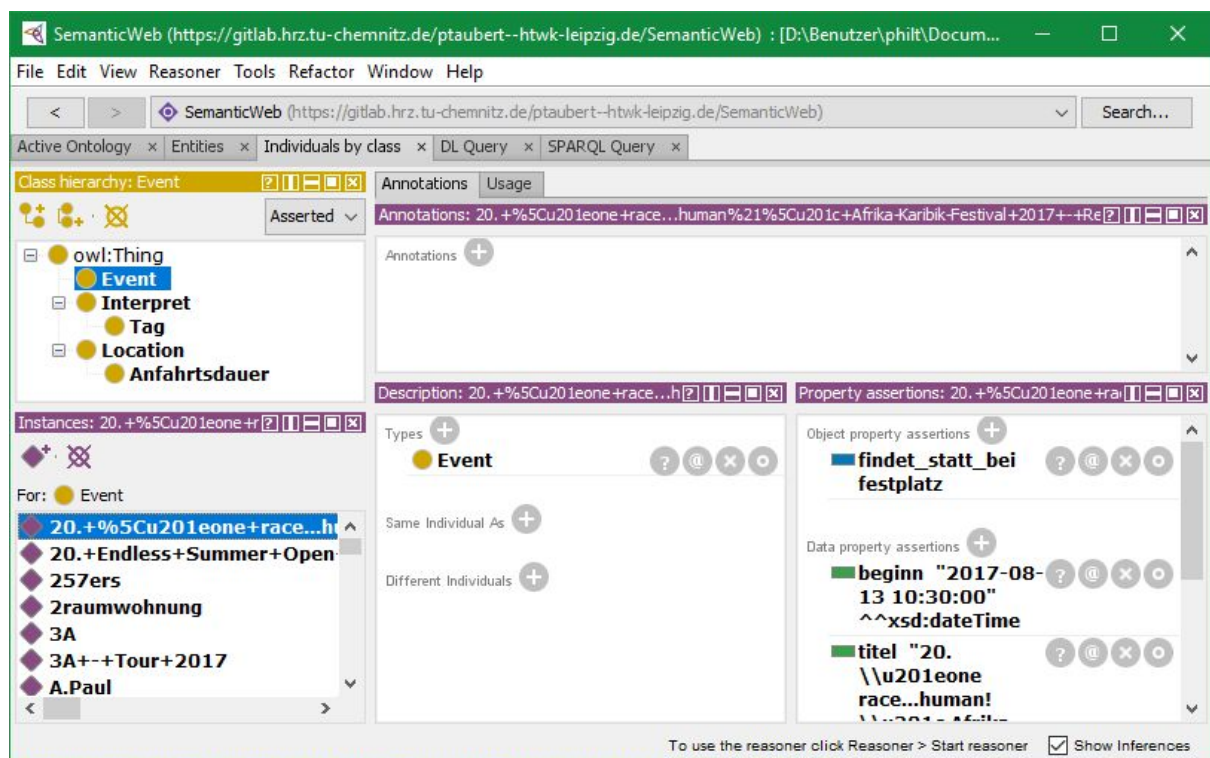


Unter Export → RDF as Turtle werden die Daten nun als ttl-Dateien Exportiert, sodass diese in Protegé importiert werden können.

# Abfrage mit SPARQL

Die exportierten Turtles beinhalten sämtliche relevante Daten aus allen Datenquellen im RDF-Format, welche nun eingeschränkt auf die zu Beginn beschriebenen Kriterien abgefragt werden sollen. Hierfür werden diese in Protegé importiert, worin dann SPARQL-Abfragen ausgeführt werden können.

In Protegé können im Reiter “Active Ontology” unter “Direct Import” mehrere Dateien importiert werden. Mit einem Klick auf das Plus werden alle drei Dateien nacheinander importiert. Nun kann man im Reiter “Individuals by Class” sämtliche Objekte der Klassen mit allen Properties durchsuchen.



Im Reiter “SPARQL Query” können nun Abfragen mit der SPARQL Protocol And RDF Query Language ausgeführt werden. Das Ergebnis wird in einer Tabelle angezeigt. Nun wird eine SPARQL-Abfrage entwickelt, welche sämtliche Elemente miteinander verknüpft und auf die gesuchten Kriterien filtert. Der Interpret muss einen Tag “Alternative Rock” besitzen und die Anfahrtszeit soll kleiner gleich zwei Stunden, also 7200 Sekunden sein. In der Ergebnistabelle sollen der Name des Events, der Interpret, die Location, die Adresse, die Anfahrtsdauer sowie die Entfernung angezeigt werden. Somit entsteht folgende SPARQL-Abfrage:

```
PREFIX sw:
    <https://gitlab.hrz.tu-chemnitz.de/ptaubert--htwk-leipzig.de/
    SemanticWeb#>
SELECT ?event ?beginn ?interpret ?location ?adresse ?seconds
    ?entfernung
WHERE {
    ?interpret sw:besitzt
        <https://gitlab.hrz.tu-chemnitz.de/ptaubert--htwk-
        leipzig.de/alternative+rock> ;
    sw:spielt_auf ?event .
    ?event sw:findet_statt_bei ?location ;
        sw:beginn ?beginn .
    ?location sw:hat ?anfahrtsdauer .
    ?anfahrtsdauer sw:anfahrtsdauer ?seconds ;
        sw:endadresse ?adresse ;
        sw:entfernung ?entfernung .
    FILTER (?seconds <= 7200)
}
```

Das Ergebnis ist genau ein Event, welches alle Kriterien erfüllt.

| event           | beginn                | interpret       | location                | adresse  | seconds | entfernung |
|-----------------|-----------------------|-----------------|-------------------------|--|---------|------------|
| Jimmy+Eat+World | "2017-08-27 00:00:00" | jimmy+eat+world | kulturzentrum+strasse+e | "Werner-Hartmann-Straße 2, 01099 Dresden, Germany" | "4793"  | "125332"   |



Der Interpret "Jimmy Eat World" spielt ein Konzert am 27.08. im Kulturzentrum Strasse E in Dresden. Ab Leipzig benötigt man zu dieser Location mit dem Auto rund 80 Minuten. Trotz der Unvollständigkeit der Event-Datenquelle wurde eine interessante Veranstaltung gefunden. Dieses Ergebnis ist ausreichend und schont den Geldbeutel.

Insgesamt sind 944 Veranstaltungen ohne Filter vorhanden. Sucht man allgemeiner nach Interpreten mit dem Tag "Rock" findet man neben der oben bereits gefundenen Veranstaltung 25 Weitere.

| event                      | beginn                | interpret            | location                | adresse                                       | seconds | entfernung |
|----------------------------|-----------------------|----------------------|-------------------------|---|---------|------------|
| Jimmy+Eat+World            | "2017-08-27 00:00:00" | jimmy+eat+world      | kulturzentrum+strasse+e | "Werner-Hartmann-Straße 2, 01099 Dresden,"    | "4793"  | "125332"   |
| Silly++Open+Air            | "2017-08-18 00:00:00" | silly                | geraer+hofwiesenpark    | "Am Sommerbad 64X, 07548 Gera, Germany"@      | "3883"  | "102694"   |
| The+Sisters+of+Mercy       | "2017-09-18 00:00:00" | the+sisters+of+mercy | alter+schlachthof       | "Gothaer Str. 11, 01097 Dresden, Germany"@    | "4690"  | "119665"   |
| The+Sisters+of+Mercy       | "2017-09-07 20:00:00" | the+sisters+of+mercy | alter+schlachthof       | "Gothaer Str. 11, 01097 Dresden, Germany"@    | "4690"  | "119665"   |
| The+Sisters+of+Mercy       | "2017-09-17 20:00:00" | the+sisters+of+mercy | alter+schlachthof       | "Gothaer Str. 11, 01097 Dresden, Germany"@    | "4690"  | "119665"   |
| The+Sisters+of+Mercy       | "2017-09-24 20:00:00" | the+sisters+of+mercy | alter+schlachthof       | "Gothaer Str. 11, 01097 Dresden, Germany"@    | "4690"  | "119665"   |
| Procol+Harum               | "2017-09-13 00:00:00" | procol+harum         | alter+schlachthof       | "Gothaer Str. 11, 01097 Dresden, Germany"@    | "4690"  | "119665"   |
| Procol+Harum               | "2017-09-16 20:00:00" | procol+harum         | alter+schlachthof       | "Gothaer Str. 11, 01097 Dresden, Germany"@    | "4690"  | "119665"   |
| Heinz+Rudolf+Kunze++Solo+% | "2017-08-27 00:00:00" | heinz+rudolf+kunze   | schloss+freudenstein    | "Schlosspl. 4, 09599 Freiberg, Germany"@      | "4512"  | "108030"   |
| Heinz+Rudolf+Kunze         | "2017-08-27 20:00:00" | heinz+rudolf+kunze   | schloss+freudenstein    | "Schlosspl. 4, 09599 Freiberg, Germany"@      | "4512"  | "108030"   |
| STEVE+HILL++%22Dangerous%  | "2017-09-15 20:00:00" | steve+hill           | museums Keller          | "Juri-Gagarin-Ring 140A, 99084 Erfurt, Germar | "5534"  | "145399"   |
| STEVE+HILL++%22Dangerous%  | "2017-09-17 20:00:00" | steve+hill           | museums Keller          | "Juri-Gagarin-Ring 140A, 99084 Erfurt, Germar | "5534"  | "145399"   |
| CLAYMORE++Sumosluts        | "2017-09-22 21:00:00" | claymore             | oberer+gasthof+gruena   | "Chemnitzer Str. 178, 09224 Chemnitz, Germz   | "3404"  | "84423"    |
| Silbermond                 | "2017-09-03 00:00:00" | silbermond           | filmn%C3%A4chte+am+elbu | "Königsufer, 01097 Dresden, Germany"@         | "4869"  | "121203"   |
| Silbermond                 | "2017-09-09 19:30:00" | silbermond           | filmn%C3%A4chte+am+elbu | "Königsufer, 01097 Dresden, Germany"@         | "4869"  | "121203"   |
| Silbermond                 | "2017-08-25 20:00:00" | silbermond           | filmn%C3%A4chte+am+elbu | "Königsufer, 01097 Dresden, Germany"@         | "4869"  | "121203"   |
| Silbermond                 | "2017-09-03 00:00:00" | silbermond           | ifa+sommergarten++messe | "Summer Garden, Berlin Fair, 14055 Berlin, Ge | "6642"  | "181051"   |
| Silbermond                 | "2017-09-03 19:30:00" | silbermond           | ifa+sommergarten++messe | "Summer Garden, Berlin Fair, 14055 Berlin, Ge | "6642"  | "181051"   |

Von allen gespeicherten Events spielt auf 24 davon ein Interpret aus dem Genre Alternative Rock. Lässt man sich alle Events ausgeben, die 2 Stunden oder weniger entfernt von Leipzig stattfinden, findet man 124.

Durch kleine Änderungen an der Abfrage lassen sich also für verschiedene Musikrichtungen Veranstaltungen im Zeitraum der Semesterferien finden, welche für Bewohner Leipzigs ansprechend sein könnten.

Da der Account im GitLab der TU Chemnitz aus nicht ersichtlichen Gründen gesperrt wurde, wurden sämtliche Daten zu GitHub hochgeladen. Diese können unter <https://github.com/Tapematch/SemanticWeb> abgerufen werden.