

# Ontology

A bit about the ontology used to structure the RDF triples in our historical record query tool

## 1. Loading triples

When inserting a set of events into the GraphDB database, our tool goes on to fetch a subset of triples containing the specified year from YAGO's SPARQL API in the following manner:

```
PREFIX yago: <http://yago-knowledge.org/resource/>
SELECT ?s ?p ?o
WHERE {{
    ?s ?p ?o .
    FILTER regex(?o, "{query_param}", "i")
    FILTER langMatches(lang(?o), "{lang}")
}}
LIMIT {limit}
OFFSET {offset}
```

Here, in the WHERE statement of the query we can observe we filter by year and then match the language selected.

Then, we go on to limit the number of response items to a smaller subset using LIMIT. This is done to avoid timeout of the server due to a 60 second timeout placed by YAGO to avoid latency for all its users.

This limit creates an issue for us, where we can only get the first (10-20) items in the list for a year. To avoid this we've set an OFFSET parameter to our query to skip a desired number of triples in the response, so for example, if you already queried the first 10 items, and you'd like to move on to other subsets of the list, you can make the same query with an offset of at least 10.

## 2. Writing triples to the graph database

After already having loaded our results we construct the ontology used in our db

```
for result in results['results']['bindings']:
    s = result['s']['value']
    p = result['p']['value']
    o = result['o']['value']
```

by going through each of the results in the response object and constructing the triple as subject (s), predicate (p) and object (o).

```
# Get the year from the serializer
year = query_param

# Calculate the century from the year
century = (int(year) - 1) // 100 + 1
```

Then, the year and century attributes are assigned and calculated, which makes us ready to build the SPARQL query and add them to the graph.

```
INSERT DATA {{
    <{s}> <{p}> "{o}" .
    <{s}> a <http://example.com/Event> .
    <http://example.com/{year}> a <http://example.com/Year> .
    <http://example.com/{century}> a
<http://example.com/Century> .
    <{s}> <http://example.com/hasYear>
<http://example.com/{year}> .
    <http://example.com/{year}>
<http://example.com/hasCentury> <http://example.com/{century}> .
}}
```

We add the triple, then make the subject a type Event pertaining to a year, of type year pertaining to a century of type century as illustrated in the image below:

