# Semantic Classical Music REST API

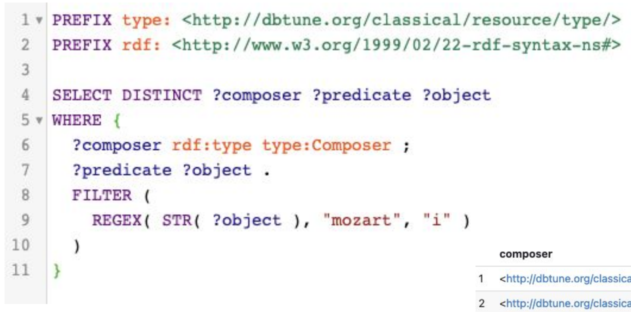
**WSDL, Group A**
João Sousa, up201806613
João Matos, up201703884
Tiago Gomes, up201806658

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

## Recap

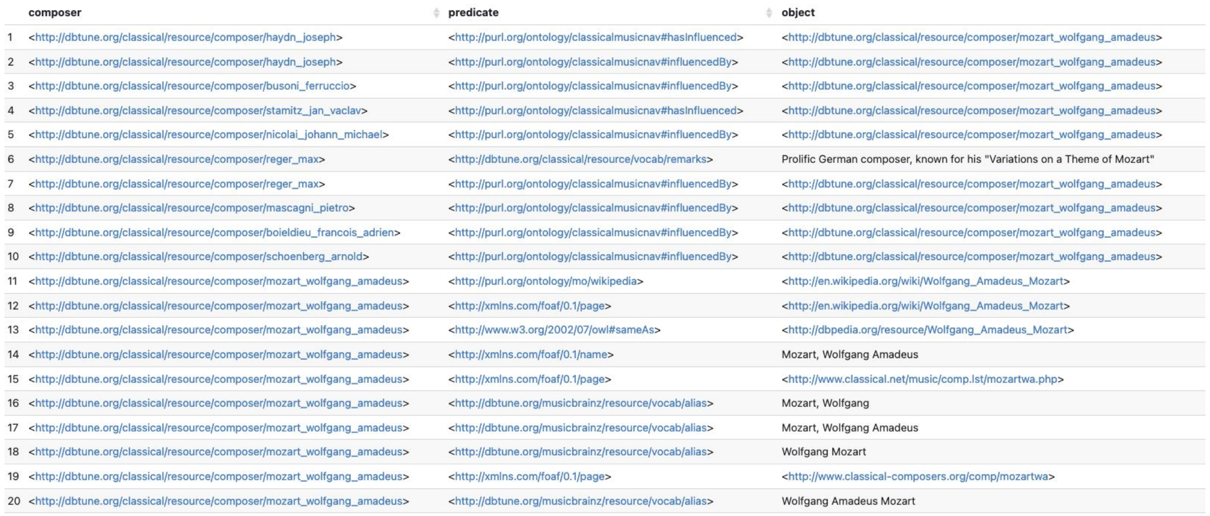
- Classical music API for the Semantic Web
  - CRUD operations on the available resources
  - Search different types of resources using keywords
  - Example queries on our knowledge graph
  - Execute federated queries
- DBtune and DBpedia as knowledge sources
- Apache Jena Fuseki is used as a triplestore
- Spring Boot is the backend framework

# Development Details - Search



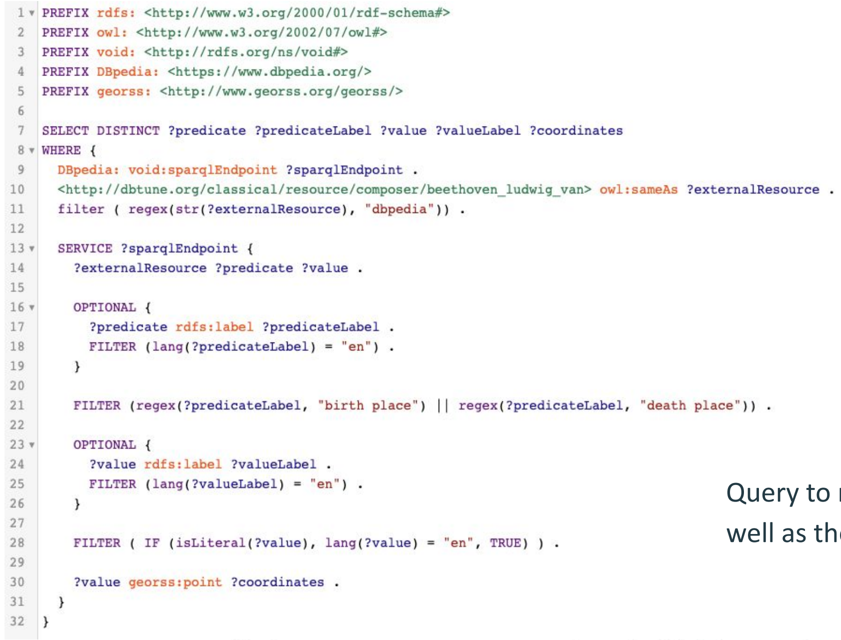
```
PREFIX type: <http://dbtune.org/classical/resource/type/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?composer ?predicate ?object
WHERE {
  ?composer rdf:type type:Composer ;
  ?predicate ?object .
  FILTER (
    REGEX( STR( ?object ), "mozart", "i" )
  )
}
```

- Search all the triples that contain the term "mozart" in the object
- FILTER and REGEX to make a simple search



| | composer | predicate | object |
|---|---|---|---|
| 1 | <http://dbtune.org/classical/resource/composer/haydn_joseph> | <http://purl.org/ontology/classicalmusicnav#hasInfluenced> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 2 | <http://dbtune.org/classical/resource/composer/haydn_joseph> | <http://purl.org/ontology/classicalmusicnav#influencedBy> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 3 | <http://dbtune.org/classical/resource/composer/busoni_ferruccio> | <http://purl.org/ontology/classicalmusicnav#influencedBy> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 4 | <http://dbtune.org/classical/resource/composer/stamitz_jan_vaclav> | <http://purl.org/ontology/classicalmusicnav#hasInfluenced> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 5 | <http://dbtune.org/classical/resource/composer/nicolai_johann_michael> | <http://purl.org/ontology/classicalmusicnav#influencedBy> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 6 | <http://dbtune.org/classical/resource/composer/reger_max> | <http://dbtune.org/classical/resource/vocab/remarks> | Prolific German composer, known for his "Variations on a Theme of Mozart" |
| 7 | <http://dbtune.org/classical/resource/composer/reger_max> | <http://purl.org/ontology/classicalmusicnav#influencedBy> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 8 | <http://dbtune.org/classical/resource/composer/mascagni_pietro> | <http://purl.org/ontology/classicalmusicnav#influencedBy> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 9 | <http://dbtune.org/classical/resource/composer/boieldieu_francois_adrien> | <http://purl.org/ontology/classicalmusicnav#influencedBy> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 10 | <http://dbtune.org/classical/resource/composer/schoenberg_arnold> | <http://purl.org/ontology/classicalmusicnav#influencedBy> | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> |
| 11 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://purl.org/ontology/mo/wikipedia> | <http://en.wikipedia.org/wiki/Wolfgang_Amadeus_Mozart> |
| 12 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://xmlns.com/foaf/0.1/page> | <http://en.wikipedia.org/wiki/Wolfgang_Amadeus_Mozart> |
| 13 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://www.w3.org/2002/07/owl#sameAs> | <http://dbpedia.org/resource/Wolfgang_Amadeus_Mozart> |
| 14 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://xmlns.com/foaf/0.1/name> | Mozart, Wolfgang Amadeus |
| 15 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://xmlns.com/foaf/0.1/page> | <http://www.classical.net/music/comp.lst/mozartwa.php> |
| 16 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://dbtune.org/musicbrainz/resource/vocab/alias> | Mozart, Wolfgang |
| 17 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://dbtune.org/musicbrainz/resource/vocab/alias> | Mozart, Wolfgang Amadeus |
| 18 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://dbtune.org/musicbrainz/resource/vocab/alias> | Wolfgang Mozart |
| 19 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://xmlns.com/foaf/0.1/page> | <http://www.classical-composers.org/comp/mozartwa> |
| 20 | <http://dbtune.org/classical/resource/composer/mozart_wolfgang_amadeus> | <http://dbtune.org/musicbrainz/resource/vocab/alias> | Wolfgang Amadeus Mozart |

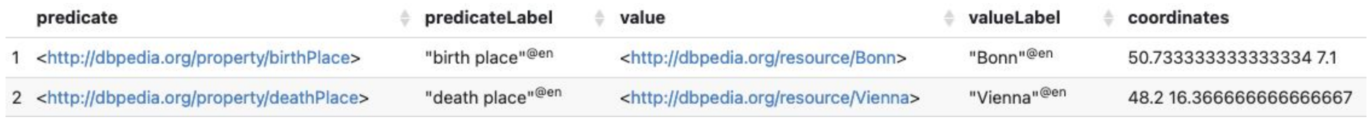
# Development Details - Federated Queries



```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX void: <http://rdfs.org/ns/void#>
PREFIX DBpedia: <https://www.dbpedia.org/>
PREFIX georss: <http://www.georss.org/georss/>

SELECT DISTINCT ?predicate ?predicateLabel ?value ?valueLabel ?coordinates
WHERE {
  DBpedia: void:sparqlEndpoint ?sparqlEndpoint .
  <http://dbtune.org/classical/resource/composer/beethoven_ludwig_van> owl:sameAs ?externalResource .
  filter ( regex(str(?externalResource), "dbpedia")) .

  SERVICE ?sparqlEndpoint {
    ?externalResource ?predicate ?value .

    OPTIONAL {
      ?predicate rdfs:label ?predicateLabel .
      FILTER (lang(?predicateLabel) = "en") .
    }

    FILTER (regex(?predicateLabel, "birth place") || regex(?predicateLabel, "death place")) .

    OPTIONAL {
      ?value rdfs:label ?valueLabel .
      FILTER (lang(?valueLabel) = "en") .
    }

    FILTER ( IF (isLiteral(?value), lang(?value) = "en", TRUE) ) .

    ?value georss:point ?coordinates .
  }
}
```

- void description vocabulary to store meta information about external datasets (DBpedia SPARQL endpoint)
- SERVICE keyword to retrieve information from the external dataset
- OPTIONAL keyword to make sure the query does not fail if there are no labels for the predicate or values

Query to retrieve the birth and death places of a composer, as well as the respective coordinates, from DBpedia



| | predicate | predicateLabel | value | valueLabel | coordinates |
|---|---|---|---|---|---|
| 1 | <http://dbpedia.org/property/birthPlace> | "birth place"@en | <http://dbpedia.org/resource/Bonn> | "Bonn"@en | 50.733333333333334 7.1 |
| 2 | <http://dbpedia.org/property/deathPlace> | "death place"@en | <http://dbpedia.org/resource/Vienna> | "Vienna"@en | 48.2 16.366666666666667 |

# Implemented Endpoints

## CRUD and Search

| HTTP Method | Path | Input Data |
|---|---|---|
| GET | event/<event_id> | The id of the event. |
| DELETE | event/<event_id> | The id of the event. |
| POST | event | URI and triples of the event. |
| GET | event/search/<query> | The search query. |
| GET | composer/<composer_id> | The id of the composer. |
| DELETE | composer/<composer_id> | The id of the composer. |
| POST | composer | URI and triples of the composer. |
| GET | composer/search/<query> | The search query. |
| GET | work/<composer_id>/<work_id> | The id of the work and the composer. |
| DELETE | work/<composer_id>/<work_id> | The id of the work and the composer. |
| POST | work | URI and triples of the work. |
| PUT | work/<composer_id>/<work_id> | The id of the work and the composer, the triples of the work. |
| GET | work/search/<query> | The search query. |
| GET | conductor/<conductor_id> | The id of the conductor. |
| DELETE | conductor/<conductor_id> | The id of the conductor. |
| POST | conductor | URI and triples of the conductor. |
| GET | conductor/search/<query> | The search query. |

## Queries

| HTTP Method | Path | Input Data |
|---|---|---|
| GET | queries/composerWorks?composerId=<composer_id> | The id of the composer. |
| GET | queries/workKeys?key=<key> | The desired key. |
| GET | queries/composersWhoInfluenced?composerId=<composer_id> | The id of the composer. |
| GET | queries/composersWhoWereInfluenced?composerId=<composer_id> | The id of the composer. |
| GET | queries/partsOfWork?composerId=<composer_id>&workId=<work_id> | The id of the composer and the work. |
| GET | queries/compositionsByYear?year=<year> | The desired year. |
| GET | queries/compositionsByTimeRange?year1=<year1>&year2=<year2> | The upper and lower bounds of the range. |
| GET | queries/compositionsByPlace?place=<place> | The desired place. |

## Federation

| HTTP Method | Path | Input Data |
|---|---|---|
| GET | composer/dbpedia-federation/<composer_id> | The id of the composer. |
| GET | queries/composerLocations?composerId=<composer_id> | The id of the composer. |

## Demonstration

## Conclusion & Future Work

- Improvements:
    - Improve the way updates are implemented
    - More queries to cover a wider range of use cases
    - Frontend to allow the user to explore our knowledge graph in a more user friendly way
- API that allows users to interact with a classical music dataset
- Use of other data sources through federated queries
- Technically, requirements for the first star are not met (due to the costs associated with hosting)
    - Requirements for the other 4 stars are met

Questions?