Share VDE - Backend

Andrea Gazzarini

www.share-vde.org
info@share-vde.org
Overview
Search API: Overview

Users

Apps

Data

Search API

- StructQL
- SVDEQL
- TermsQL

Data management

- PostgreSQL
- Solr Cloud
- Stardog
- Hadoop
How can we interact with API?
Query Languages

- Interface Level
  - GraphQL
  - REST

- Query Language Level
  - StructQL
  - SVDEQL
  - TermsQL

- Search Level
  - Full text
  - Typeahead
TermsQL

- Available in **GraphQL** and **REST** endpoints
- It isn’t a query language itself
- The **query string** is composed only by **terms**
- When used, it triggers a **typeahead** search (see [here](#) for a detailed functional description)
- The response contains
  - matching entities with highlighting snippets
  - (optionally*) a “correctlySpelled” flag which indicates if some correction has been applied to the original terms
  - (optionally*) matching entities in other languages

*Only if the feature has been enabled and in case of zero results (using original terms + requestor language)*
SVDEQL / StructQL

- **Simple**: we don’t want to build another query language
- **Dedicated grammar** for expressing and validating the “advanced search” syntax
- **Easy** and **human-readable** way to express an **information need**
- **Communication protocol** between **frontend** and **backend**
- Could be potentially **exposed** to **advanced users** in the future
- Only a “**format**” difference between the two query languages

\[
\text{people whose name contains Pastorius AND birth date is in range from 1942 to 1956}
\]
StructQL vs SVDEQL

**StructQL**
- Only available in the GraphQL endpoint
- JSON-like structure
- Grammar enforced by GraphQL schema

**Example**
```json
{
  q: [
    { name: { p: CONTAINS, o: "Carroll"}, op: AND },
    { name: { p: CONTAINS, o: "Lewis"} }
  ]
}
```

**SVDEQL**
- Available in the GraphQL and REST endpoints
- Pseudo-natural syntax
- Grammar enforced by JAVACC

**Example**
```json
agents whose
  name contains Carroll AND
  name contains Lewis
```
What do we get in output?
Hybrid Response: Realtime and Near Realtime

Search API

- GraphQL
- { REST }
- StructQL
- SVDEQL
- TermsQL

Search API: Search (query, filters, sort)

1. Search (query, filters, sort)

2. Response

Response: Resources Collection, Pagination, Facets, Resource
Query Languages

Interface Level

Query Language Level

Search Level

GraphQL

REST

StructQL

SVDEQL

TermsQL

Full text

Typeahead
RESTful API

- **Hypermedia As The Engine Of Application State (HATEOAS)**
- Share VDE resources provide information dynamically through hypermedia
- client actions are discovered within resource representations returned from the server
- linked resources are expressed through URIs, in dedicated sections of the resource representation
RESTFul API: Resource (Person)

```
{  
    "heading": "Carroll, Lewis",
    ...
    "_links": {  
        "self": [
            {  
                "href": "https://share-vde.org/people/201"
            },
            {  
                "href": "https://share-vde.org/agents/201"
            },
            {  
                "href": "http://isni.org/isni/000000012137136X", "type": "ISNI"
            },
            {  
                "href": "https://viaf.org/viaf/66462036", "type": "VIAF"
            }
        ],
        "photo": {  
            "href": "https://commons.wikimedia.org/wiki/lc_1863.jpg"
        },
        "birth_place": {  
            "href": "https://share-vde.org/places/7295222"
        }
    },
    "alternate_headings": [  
        "Dodgson, Charles Lutwidge", "Karol, Luis"
    ],
    "birth_date": 1832,
    "death_date": 1898
}
```
RESTful API: Collection (People)

{ 
  "_embedded": { 
    "agentList": [ 
      { 
        "heading": "Carroll, Lewis",
        "_links": { 
          ...
        },
      },
      { 
        "heading": "Dodgson, Campbell",
        "alternate_headings": [ "Dodgson, C." ],
        "birth_date": 1867,
        "death_date": 1948
      },
      ...
    ]
  }
}
...  
"facets": {  
  "facet_fields": {  
    "location": {  
      { "https://share-vde.org/places//5387877": 3,  
      "https://share-vde.org/places/2650225": 2,  
      ...  
    },  
    "type": {  
      "https://share-vde.org/agentTypes/Meeting": 6,  
      "https://share-vde.org/agentTypes/Person": 4,  
      "https://share-vde.org/agentTypes/Family": 3,  
      "https://share-vde.org/agentTypes/Organisation": 1  
    }  
  }  
}...
GraphQL

**GraphQL (Query) Request**
- executes
- defines

**GraphQL Endpoint**
- produces

**Response**
- conforms to
  - contains
    - 0...1: **Pagination**
    - 1: **Resources**
    - 0...*: **Facets**

**Response Shape**
- conforms to

**JSON-like structure**
- which is

**StructQL**
- defines

**SVDEQL**
- defines

**TermsQL**
- defines

**contains a query using**
For each work return the language, specifically the URI and the heading

For each opus return the child works

For each person return the opuses he contributed

For each person return these literal attributes

For each opus return these literal attributes

Main query

```graphql
{ people(svdeql: {q: "people whose name contains meyer AND birthDate is in range from 1950 to 1988 ")} { resources { uri preferredHeading alternateHeadings birthDate } opuses { resources { uri preferredHeading } works { resources { uri language { uri preferredHeading } } } } } } 
```
GraphQL: Example Response (extract)

```json
{
  people(svdeql: {q: "people whose name contains meyer AND birthDate is in range from 1950 to 1988 ")}) {
    resources {
      uri
      preferredHeading
      alternateHeadings
      birthDate
      opuses {
        resources {
          uri
          preferredHeading
          works {
            resources {
              uri
              language {
                uri
                preferredHeading
              }
            }
          }
        }
      }
    }
    "uri": "https://share-vde.org/agents/211631289194108",
    "preferredHeading": "Meyer, Han",
    "alternateHeadings": [
      "Meyer, Valentin Johannes",
      "Meyer, Han"
    ],
    "opuses": {
      "resources": [
        {
          "uri": "https://share-vde.org/opuses/181631290147810",
          "preferredHeading": "Stad en de haven",
          "works": {
            "resources": [
              {
                "uri": "https://share-vde.org/opuses/961631362475595",
                "preferredHeading": "Delta urbanism",
                "works": {
                  "uri": "https://share-vde.org/languages/21631288253353",
                  "preferredHeading": "Inglese"
                }
              }
            ]
          }
        }
      ]
    }
  }
}
(Other people follow)
```
Search Features
FullText Search

Simple (terms/phrases) Search

Barnaby Potter

All results (2501)  Agents (1819)  Original works (682)

Results for Barnaby and Potter
2501 results found

- Person
  Barnaby Potter

- Original work
  Barnaby

Advanced Search

Typeahead Search
Exact Match Suggestions

A search which targets only entities whose headings or identifiers exactly match the query, partially or totally.

```json
exactMatches(q: "tchaikovsky") {
  resources {
    ... (other)
  }
}

exactMatches(q: "0000000113258500 ...") {
  resources {
    ... (other)
  }
}
```
A virtual explanation entity, associated to a core entity (e.g. Agent, Opus) which provides insights about the reason why a given resource has been included in search results.

"tell me what the agent 981631362140359 has to do with the term populonia"

Request: 
/meetings/981631362140359/explanation?terms=populonia

Response: 
"meta": {
"aut": {
"type": "Role",
"language": "eng",
"label": "author"
}
},
"aut": [
{
"title": "Corsica e <b>Populonia</b>"
}
].

"The term Populonia occurs in the title of an Opus where the Meeting contributed as an author"
Thank you!

Share VDE - Backend

Andrea Gazzarini

www.share-vde.org
info@share-vde.org