deti universidade de aveiro
departamento de electrónica, telecomunicações e informática

Representação do Conhecimento

A *Triplestore* GraphDB

GraphDB

- A GraphDB é uma base de dados orientada a grafos (*triplestore*) que suporta um conjunto significativo de standards e tecnologias que são a base da web semântica.
- Caraterísticas:
  - Elevada eficiência, robusta e escalável;
  - Consegue lidar com carregamentos, pesquisas e inferências, em tempo real, de modo massivo;
  - Implementa e reconhece standards como: RDF, RDFS, OWL, SPARQL;
  - Permite efetuar pesquisas a repositórios federados;
  - Apresenta uma boa integração com ferramentas exteriores de pesquisa como o Lucene, SOLR e Elasticsearch.

GraphDB (ii)

- Esta base dados apresenta-se em três edições:
  - **Free** – comercial; baseada em ficheiros; escalável até dezenas de biliões de triplos RDF; executa num único servidor; apresenta a limitação de 2 *queries* concorrentes. Download em: http://graphdb.ontotext.com/
  - **Standard Edition (SE)** – igual à anterior, mas sem limitação de *queries* concorrentes.
  - **Enterprise Edition (EE)** – baseada num cluster de alta disponibilidade, com uma implementação Master-Slave, para resiliência e alta performance em *queries* paralelas.

GraphDB (iii)

- Workbench Interface – gestão da base de dados a partir de uma interface gráfica.
- Permite:
  - Gerir repositórios;
  - Carregar e exportar dados;
  - Ver e editar recursos RDF
  - Executar queries SPARQL;
  - Fazer monitorização de queries, recursos e utilizadores;
  - Oferece uma API REST para acesso programático;
  - Acesso por defeito em “http://localhost:7200”.

GraphDB – s4api

- Para o desenvolvimento mais fácil de uma aplicação que usa a API REST da GraphDB, é possível usar a biblioteca python “s4api” disponível em: https://pypi.python.org/pypi/s4api
- Esta biblioteca implementa as operações básicas de acesso, via REST, não sendo necessário ter em conta os detalhes técnicos desse acesso.

GraphDB – s4api

- SELECT

```
import json
from s4api.graphdb_api import GraphDBApi
from s4api.swagger import ApiClient

endpoint = "http://localhost:7200"
repo_name = "movies"
client = ApiClient(endpoint=endpoint)
accessor = GraphDBApi(client)
...
```

GraphDB – s4api

- SELECT

```
...
query = """
PREFIX mov:<http://movies.org/pred/>
SELECT ?actor_n
WHERE{
      ?film mov:name "Blade Runner" .
      ?film mov:starring ?actor .
      ?actor mov:name ?actor_n .
}
"""
payload_query = {"query": query}
res = accessor.sparql_select(body=payload_query,
                             repo_name=repo_name)
res = json.loads(res)
for e in res['results']['bindings']:
    print(e['actor_n']['value'])
```

GraphDB – s4api

- UPDATE - INSERT

```
...
update = """
    PREFIX mov:<http://movies.org/pred/>
    PREFIX move: <http://movies.org/>
    INSERT DATA
    {
        move:my_life mov:name "My Life in Hell" .
    }
"""
payload_query = {"update": update}
res = accessor.sparql_update(body=payload_query,
repo_name=repo_name)
```

GraphDB – s4api

- UPDATE - DELETE

```
...
update = """
    PREFIX mov:<http://movies.org/pred/>
    PREFIX move: <http://movies.org/>
    DELETE DATA
    {
        move:my_life mov:name "My Life in Hell" .
    }
"""
payload_query = {"update": update}
res = accessor.sparql_update(body=payload_query,
repo_name=repo_name)
```