

## Anexo E: Teste de conexão com o endpoint SparQL que distribui dados do Participa.br

Este script faz buscas simples no endpoint SparQL com os dados do Participa.br, para testar a conexão e o tempo de resposta. Para os fins até agora utilizados, foi suficiente:

- I. Fazer uma busca simples na DBpedia, para testar conexão de onde está sendo executado o script e para testar o funcionamento dos recursos envolvidos.
- II. Uma busca de dados do Participa.br, leve (todos os participantes, com nomes e URIs)
- III. Uma busca pesada de dados do Participa.br (todos os comentários, com texto, título e URIs)

Basta clicar nas células e apertar SHIFT+ENTER para executar os comandos dela.

```
In [3]: from SPARQLWrapper import SPARQLWrapper, JSON
import time
```

```
In [35]: NOW=time.time()
sparql = SPARQLWrapper("http://dbpedia.org/sparql")
sparql.setQuery("""
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    SELECT ?label
    WHERE { <http://dbpedia.org/resource/Love> rdfs:label ?label }
""")
sparql.setReturnFormat(JSON)
results = sparql.query().convert()
print(    "%.2f segundos para consultar a dbpedia os nomes:" +
        " %s"*len(results["results"]["bindings"]) )
% tuple([time.time()-NOW]+[i["label"]["value"]
                        for i in results["results"]["bindings"]]) )
```

0.49 segundos para consultar a dbpedia os nomes: 愛 Liebe Love Amor Amor Amour Amore  
愛 Liefde Miłość Любoвь Kärlek

```
In [36]: PREFIX=""
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ops: <http://purl.org/socialparticipation/ops#>
PREFIX opa: <http://purl.org/socialparticipation/opa#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX tsioc: <http://rdfs.org/sioc/types#>
PREFIX schema: <http://schema.org/>
"""
```

```
In [40]: q2="SELECT ?nome WHERE {?s rdf:type ops:Participant . ?s foaf:name ?nome .}"
NOW=time.time()
sparql3 = SPARQLWrapper("http://localhost:82/participabr/query")
# caso esteja rodando em instancia propria:
#sparql3 = SPARQLWrapper("http://200.144.255.210:8082/participabr/query")
sparql3.setQuery(PREFIX+q2)
sparql3.setReturnFormat(JSON)
results3 = sparql3.query().convert()
print("%.2f segundos para puxar todos os %i nomes dos participantes do Participa.br"
      %(time.time()-NOW,len(results3["results"]["bindings"])))
print( "com os seguintes campos: %s"
      %(str(results3["results"]["bindings"][0].keys())) )
```

0.07 segundos para puxar todos os 2624 nomes dos participantes do Participa.br

0.07 segundos para puxar todos os 3624 nomes dos participantes do Participa.br com os seguintes campos: [u'nome']

```
In [43]: NOW=time.time()
q="""SELECT ?comentario ?titulo ?texto WHERE {?comentario dc:type tsioc:Comment.
OPTIONAL {?comentario dc:title ?titulo . }
OPTIONAL {?comentario schema:text ?texto .}}"""
sparql3.setQuery(PREFIX+q)
sparql3.setReturnFormat(JSON)
results4 = sparql3.query().convert()
print("%.2f segundos para puxar todos os %i comentários do Participa.br"%
      (time.time()-NOW,len(results4["results"]["bindings"])))
print("com campos: "+str(results4["results"]["bindings"][0].keys()))
```

2.44 segundos para puxar todos os 16253 comentários do Participa.br com campos: [u'titulo', u'comentario', u'texto']

|||--- FIM ---|||