

Shell scripts para manipular grafos de conhecimento em um servidor Fuseki.

Motivação

Os shell scripts visam automatizar operações. Essa automatização evita que o programador use seu tempo digitando trechos de comandos que podem ser padronizados, ou precise ajustar-se a sintaxes específicas de ferramentas, como o shell script (que precisa "escapar" certos caracteres), ou requisições GET (que precisam substituir caracteres como espaços por %20 e chaves por %7b e %7d).

Numa camada seguinte, uma IDE pode chamar os shell scripts através de atalhos de teclado ou botões na interface, inclusive passando como argumentos, por exemplo, o texto marcado em uma janela de edição de texto, como é possível fazer com o plug-in de ferramentas externas do gedit Ver captura de tela (Captura%20de%20tela%20de%202021-04-28%2018-13-35.png).

SPARQL é uma linguagem de consulta que permite muitas construções, por isso sua sintaxe não é das mais simples (<https://www.w3.org/TR/sparql11-query/> (<https://www.w3.org/TR/sparql11-query/>), <https://www.w3.org/TR/sparql11-update/> (<https://www.w3.org/TR/sparql11-update/>)).

Consultas SPARQL frequentemente são enviadas a servidores na Internet usando o protocolo HTTP. Por exemplo em dbpedia e wikidata. Algumas consultas podem ser enviadas através de requisições GET, e todas elas podem ser enviadas através de requisições POST (checar afirmação).

A ferramenta mais comum usada para acesso à internet é o navegador. Este, sem o auxílio de plugins ou addons (a denominação depende do particular desenvolvedor), não permite ao usuário enviar requisições POST. Com o auxílio de plugins tornam o usuário capaz de enviar requisições POST, mas os recursos de composição de atividades e resultados (ex. macros e scripts) são limitados. Esta dificuldade geralmente é contornada com o uso de javascript.

Outra característica do navegador é a implementação da segurança do sistema e da informações do usuário. Navegadores executam os scripts de cada janela/aba em áreas protegidas (sandboxes), com acesso limitado ao sistema de arquivos. Esta estratégia é muito boa para garantir a segurança do usuário contra sites maliciosos, mas muito ruim para, por exemplo, construir uma IDE executada localmente e dentro do navegador. Seria necessário instalar um servidor web na máquina local, estratégia usada por NodeJS.

Seria muito cômodo deixar a conversão de caracteres RFC3986 (<https://tools.ietf.org/html/rfc3986#page-12>) por conta do navegador, mas esta facilidade não compensa as dificuldades citadas acima.

Comunicação por HTTP pode ser feita em linha de comando, por aplicativos como `wget` e `curl`. O primeiro costuma ser usado para transferência de arquivos através de HTTP, o segundo para envio de requisições e apresentação de resultados como listagem de texto (que pode ser redirecionada para arquivos de acordo com a conveniência). Esta escolha acrescenta dois níveis de conversão de caracteres: um pelo aplicativo (`curl`), outro pela linha de comando (`sh`). Ainda assim, como são tratadas por quem constrói os shell scripts (programadores desta ferramenta), são transparentes aos usuários (programadores de aplicação).

Objetivo

Construir uma coleção de shell scripts para manipulação de grafos de conhecimento.

Resultados

nota: Para SO Linux.

init.sh (<http://init.sh>)

Uso: `source init.sh`

Função: Cria as variáveis de ambiente QUERY e UPDATE, que contém os links para os endpoints SPARQL. *Este script deve ser ajustado para apontar para os seus endpoints.*

nota: há formas de tornar esta linha de comando mais curta.

nota: o script é executado em um 'sub-shell' (em analogia a subclasses). Em princípio as variáveis definidas nele não seriam propagadas para o 'super-shell' (em analogia a superclasses). O comando `source` faz essa propagação. **As variáveis de ambiente são essenciais para o funcionamento dos outros scripts.**

Exemplo:

```
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ source init.sh (http://init.sh)
```

la.sh (<http://la.sh>)

Uso: `sh la.sh`

Função: Lista todas as triplas armazenadas no dataset.

Exemplo:

```

fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh la.sh |more
% Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
           Dload  Upload  Total      Spent    Left     Speed

  0     0    0     0    0     0      0      0  --:--:--  --:--:--  --:--:--    0{
"head": {
  "vars": [ "s" , "p" , "o" ]
} ,
"results": {
  "bindings": [
    {
      "s": { "type": "uri" , "value": "http://www.semanticweb.org/joeds/ontologies/2020/3/swotDom
oticProject" } ,
      "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type" } ,
      "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#Ontology" }
    } ,
    {
      "s": { "type": "uri" , "value": "http://www.semanticweb.org/joeds/ontologies/2020/3/swotDom
oticProject#belongsTo" } ,
      "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type" } ,
      "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#ObjectProperty" }
    } ,
    {
      "s": { "type": "uri" , "value": "http://www.semanticweb.org/joeds/ontologies/2020/3/swotDom
oticProject#belongsTo" } ,
      "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type" } ,
      "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#FunctionalProperty" }
    } ,
    {
      ...

```

up.sh (<http://up.sh>)

Uso: sh up.sh <arquivo>

Função: Executa a operação definida no arquivo contra o endpoint de update.

Operações de inserção de tripla

Arquivo: umaTripla.spq

```
update=
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT DATA {
  <http://example/book1> dc:title "A new book" ;
                        dc:creator "A.N.Other"
  .
}
```

Arquivo: outraTripla.spq

```
update=
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT DATA {
  <http://example/book2> dc:title "A new book2" ;
                        dc:creator "A.N.Other2"
  .
}
```

Exemplo:

```
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh up.sh outraTripla.s
pq
<html>
<head>
</head>
<body>
<h1>Success</h1>
<p>
Update succeeded
</p>
</body>
</html>
```

Operação de carga de ontologia armazenada em servidor de terceiro

Arquivo: carregassn.spq

```
update= LOAD <http://www.w3.org/ns/ssn/>
```

Arquivo: carregasosa.spq

```
update= LOAD <http://www.w3.org/ns/sosa/>
```

Exemplo: ver exemplo de uso de RMALL.sh (README.md#rmall-sh)

qs.sh (<http://qs.sh>)

Uso: sh qs.sh <sujeito>

Função: Lista todas as triplas com o sujeito dado como argumento.

Exemplo:

```
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh ./qs.sh "<http://example/book2 (http://example/book2)>"
{
  "head": {
    "vars": [ "p" , "o" ]
  } ,
  "results": {
    "bindings": [
      {
        "p": { "type": "uri" , "value": "http://purl.org/dc/elements/1.1/title (http://purl.org/dc/elements/1.1/title)" } ,
        "o": { "type": "literal" , "value": "A new book2" }
      } ,
      {
        "p": { "type": "uri" , "value": "http://purl.org/dc/elements/1.1/creator (http://purl.org/dc/elements/1.1/creator)" } ,
        "o": { "type": "literal" , "value": "A.N.Other2" }
      }
    ]
  }
}
```

rms.sh (<http://rms.sh>)

Uso: sh rms.sh <sujeito>

Função: Remove todas as triplas com o sujeito dado como argumento.

Exemplo:

```
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh ./rms.sh "<http://example/book2 (http://example/book2)>"
update=DELETE { <http://example/book2 (http://example/book2)> ?p ?o } WHERE {
<http://example/book2 (http://example/book2)> ?p ?o. }
<html>
<head>
</head>
<body>
<h1>Success</h1>
<p>
Update succeeded
</p>
</body>
</html>
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh ./qs.sh "<http://example/book2 (http://example/book2)>"
{
  "head": {
    "vars": [ "p" , "o" ]
  } ,
  "results": {
    "bindings": [

]

}
}
```

RMALL.sh (<http://RMALL.sh>)

Uso: sh RMALL.sh

Função: Remove todas as triplas.

Exemplo:

Neste exemplo mostra-se a remoção de todo o conteúdo do dataset usando RMALL.sh , a listagem do dataset vazio usando la.sh , a carga de SSN usando up.sh carregassn.spq e a listagem (parcial) do dataset usando la.sh .

```
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh RMALL.sh
update=DELETE { ?s ?p ?o } WHERE { ?s ?p ?o. }
<html>
<head>
</head>
<body>
<h1>Success</h1>
<p>
Update succeeded
</p>
</body>
</html>
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh la.sh
{
  "head": {
    "vars": [ "s" , "p" , "o" ]
  } ,
  "results": {
    "bindings": [

    ]
  }
}
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh up carregassn.spq
sh: 0: Can't open up
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh up.sh carregassn.sp
q
<html>
<head>
</head>
<body>
<h1>Success</h1>
<p>
Update succeeded
</p>
</body>
</html>
fabio@fabio-13Z940-G-BK71P1:~/Documentos/ZZfiles/sparql$ sh la.sh
{
  "head": {
    "vars": [ "s" , "p" , "o" ]
  } ,
  "results": {
    "bindings": [
      {
        "s": { "type": "uri" , "value": "http://purl.org/vocommons/voaf#Vocabul
ary" } ,
        "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-synta
```

```

x-ns#type" } ,
    "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#Class" }
  } ,
  {
    "s": { "type": "uri" , "value": "http://xmlns.com/foaf/0.1/Agent" } ,
    "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-synta
x-ns#type" } ,
    "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#Class" }
  } ,
  {
    "s": { "type": "uri" , "value": "http://xmlns.com/foaf/0.1/name" } ,
    "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-synta
x-ns#type" } ,
    "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#Annotati
onProperty" }
  } ,
  {
    "s": { "type": "uri" , "value": "http://purl.org/dc/terms/title" } ,
    "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-synta
x-ns#type" } ,
    "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#Annotati
onProperty" }
  } ,
  {
    "s": { "type": "uri" , "value": "http://purl.org/dc/terms/description"
} ,
    "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-synta
x-ns#type" } ,
    "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#Annotati
onProperty" }
  } ,
  {
    "s": { "type": "uri" , "value": "http://purl.org/dc/terms/rights" } ,
    "p": { "type": "uri" , "value": "http://www.w3.org/1999/02/22-rdf-synta
x-ns#type" } ,
    "o": { "type": "uri" , "value": "http://www.w3.org/2002/07/owl#Annotati
onProperty" }
  } ,
  {
    ...

```

Próximos passos

- Verificar se alguma consulta SPARQL retorna resposta em RDF/XML, que pode ser armazenada e trabalhada em um *reasoner* Jena. CONSTRUCT pode resolver;
- Criar uma consulta que carregue ontologias em um dataset. ~~Seja em arquivo remoto (como em SPARQL:LOAD)~~ (FEITO, ver up), seja a partir de um arquivo local.

Referências

- <https://stackoverflow.com/questions/7366775/what-does-the-line-bin-sh-mean-in-a-unix-shell-script> (<https://stackoverflow.com/questions/7366775/what-does-the-line-bin-sh-mean-in-a-unix-shell-script>)
- <https://superuser.com/questions/149329/what-is-the-curl-command-line-syntax-to-do-a-post-request> (<https://superuser.com/questions/149329/what-is-the-curl-command-line-syntax-to-do-a-post-request>)
- <https://www.baeldung.com/linux/bash-variables-export> (<https://www.baeldung.com/linux/bash-variables-export>)
- <https://stackoverflow.com/questions/15912924/how-to-send-file-contents-as-body-entity-using-curl> (<https://stackoverflow.com/questions/15912924/how-to-send-file-contents-as-body-entity-using-curl>)
- [http://vos.openlinksw.com/owiki/wiki/VOS/VirtTipsAndTricksSPARQL11Delete#DELETE DATA Examples](http://vos.openlinksw.com/owiki/wiki/VOS/VirtTipsAndTricksSPARQL11Delete#DELETE_DATA_Examples) (<http://vos.openlinksw.com/owiki/wiki/VOS/VirtTipsAndTricksSPARQL11Delete#DELETE%20DATA%20Examples>)
- <https://www.w3.org/TR/sparql11-update/> (<https://www.w3.org/TR/sparql11-update/>)
- <https://www.baeldung.com/linux/use-command-line-arguments-in-bash-script> (<https://www.baeldung.com/linux/use-command-line-arguments-in-bash-script>)
- <https://en.wikipedia.org/wiki/Percent-encoding> (<https://en.wikipedia.org/wiki/Percent-encoding>)
- https://www.w3schools.com/tags/ref_urlencode.ASP (https://www.w3schools.com/tags/ref_urlencode.ASP)

Referências para os próximos passos

- <https://stackoverflow.com/questions/45192163/rdf-format-output-from-fuseki-1-0> (<https://stackoverflow.com/questions/45192163/rdf-format-output-from-fuseki-1-0>)
- <https://www.futurelearn.com/info/courses/linked-data/0/steps/16104> (<https://www.futurelearn.com/info/courses/linked-data/0/steps/16104>)