

Representação e Processamento de Conhecimento (RPC)
--

1. ... explorar a “Linked Data” via “SPARQL-endpoint”

- a) Analise e execute `b03_SPARQLendpoint_BBC.py`. Este código trata a hipótese do serviço não estar disponível (que está a acontecer com o “endpoint” da BBC). Integre este código (i.e., o tratamento dessa exceção) na aplicação que implementou na aula (prática) anterior.

2. Explorar a “Linked Data” combinando informação local e remota

Considere `c01_combinar_local_e_global_rkbexplorer.py`.

- a) Execute o código e analise o grafo gerado. Vamos considerar esse grafo (gerado após execução deste código) como sendo um “grafo local” com “informação parcial”. Ou seja, é um grafo que vamos pretender estender com informação disponível na “Linked Data”.
- b) Elimine os comentários de modo a interrogar o “grafo local” obtendo os dois nomes aí registados.
- c) Aceda a `http://dblp.rkbexplorer.com/sparql/` e copie, para a caixa de texto (dessa página), a diretiva SPARQL (presente no ficheiro acima) e altere de modo a projetar os sujeitos, `?s`, do tipo `akt:Person` e “ligados” via `akt:full-name` ao objeto `?nome` (limite 10 respostas).
- d) Elimine comentários (do código) de modo a interrogar, programaticamente, o SPARQL-endpoint `http://dblp.rkbexplorer.com/sparql/` para obter informação sobre cada publicação (variável `pub` no código) e título da publicação (variável `title` no código) dos autores que estão registados no “grafo local” com “informação parcial” (gerado em alínea anterior).
- e) Elimine comentários de modo a estender o “grafo local” com a resposta da interrogação anterior.

3. Explorar a “Linked Data” combinando informação local e remota

Objetivo: rever o que desenvolveu na questão anterior, construir uma solução sua e aproveitar para reanalisar e melhorar (e.g., estruturar e generalizar) o código que lhe serviu de base.

- a) Gere um ficheiro `z_info_base.rdf` com tripletos identificando 4 autores. *Sugestão:* adapte o código explorado na alínea anterior.
- b) Gere um ficheiro `z_info_estendido.rdf` com informação adicional sobre os 4 autores (alínea anterior) que seja obtida de modo automático via acesso a “SPARQL-endpoints”.
- c) Publique a informação em `z_info_estendido.rdf` no RDF4J. *Sugestão:* recorde o processo que desenvolveu na anterior aula prática (onde se trabalhou com o RDF4J).

Representação e Processamento de Conhecimento (RPC)
--

4. Instalar um Servidor Web de Base Python – CherryPy

“CherryPy is a pythonic, object-oriented Web framework.

CherryPy allows developers to build web applications in much the same way they would build any other object-oriented Python program. This results in smaller source code developed in less time.

CherryPy is now more than ten years old and it has proven to be fast and reliable. It is being used in production by many sites, from the simplest to the most demanding.

[...] Above all, have fun with your application!”. [cf., <http://docs.cherrypy.org/en/latest/>].

a) Ver <http://cherrypy.org/>.

b) Use CherryPy-14.0.1.zip de “distribuicao_04”.
ou descarregue-o de <https://pypi.python.org/pypi/CherryPy>

Para instalar o “CherryPy” descompactar, apontar um ambiente de consola para a pasta CherryPy-14.0.1 e fazer `python setup.py install` (cf., indicações em README).

Atenção: Houve uma execução em que o processo acima deu erro por identificar dependências não instaladas. Para ultrapassar instalei usando a ferramenta “pip” (instaladora de módulos Python que a partir do Python 3.4 está incorporada na instalação do Python). Usei então o “pip” fazendo: `pip install cherrypy`. Atenção que este processo exige ligação à Internet.

Instalado em [...] \Python364-32\Lib\site-packages\CherryPy-14.0.1-py3.6.egg.
Aí, na pasta `cherrypy\tutorial`, tem vários exemplos de utilização do CherryPy.

Nota: a instalação e utilização do CherryPy foi testada no Windows10 e MacOS (v10.11.6).

5. Usar CherryPy (Web Server de base Python)

a) Iniciar uma janela de consola, ir até à pasta da aula prática e executar:

```
python d01_cherryPy_olaMundo.py
```

notar configuração serviço em `d00_cherrypy.conf` (o serviço é iniciado)

Atenção: no Windows10, Python 3.6.4 a 32bits, CherryPy-14.0.1 (minha instalação) o servidor não iniciou devolveu o erro: “... module 'socket' has no attribute 'AF_UNIX’”. Este erro deriva na implementação de `socket.py` assumir que “If the AF_UNIX constant is not defined then this protocol is undefined” (cf., <https://docs.python.org/3/library/socket.html>) e a implementação de `server.py` não testar a existência dessa definição.

Assim, caso tenha o problema acima descrito pode usar o ficheiro `server.py` fornecido nesta aula prática que tem um ligeiro ajuste (feito por mim!) que permite colocar o servidor em funcionamento. Para isso deve seguir as instruções em `server.py_README.txt`. pode também analisar os ajustes pesquisando, no ficheiro, por “PTS:fix”.

b) No Browser apontar para:

Representação e Processamento de Conhecimento (RPC)
--

- `http://localhost:8580/`
executa o método “index”
- c) No Browser apontar para:
- `http://localhost:8580/saudacao?nome=um+nome`
executa o método “saudacao” e passa-lhe o parâmetro “nome”
- d) Altere o código para apresentar outro texto (e.g., Olá em vez de Ola). Note que quando grava alterações ao código o servidor atualiza-se com essa nova versão de código. Para notar isto grave o ficheiro de código e repare nas mensagens na janela de consola onde lançou o servidor. Para ver a alteração refletir-se na página Web deve fazer “refresh” (no Browser).

6. Usar e contribuir para a “Linked Data” via CherryPy

Analise e execute `d02_cherryPy_navegadorSPARQLendpoint.py`

i.e., em janela de consola fazer `python d02_cherryPy_navegadorSPARQLendpoint.py`

- a) Analise o resultado deste novo servidor (não esquecer fazer “refresh” ao Browser!).
- b) Copie a interrogação executada no código, aceda ao URL (endpoint) e execute-a aí diretamente.
- c) Copie a outra interrogação (ainda não executada), acede a esse endpoint (DBPedia) e execute.
- d) Configure a aplicação para apresentar a resposta como uma “string” sem qualquer formatação (em vez de apresentar em “html”). *Sugestão:* note, no código o parâmetro atual “formato” na invocação em “`cherrypy.quickstart(Main(..., formato=...)`”.
- e) Ajuste o código (i.e., comente e descomente) para executar agora a interrogação sobre DBPedia.
- f) Configure para apresentar de novo em formato “html” e estenda o código para apresentar (em “html”) as duas variáveis projetadas na interrogação sobre a DBPedia.
- g) Generalize o código para apresentar, em “html”, um qualquer número de variáveis.

7. ... outras interrogações

Considere as interrogações em `x_DBPedia_interrogacoes.txt`.

- a) Obtenha outro(s) género(s) musicais e volte a obter as bandas e membros desse(s) género(s); note que um género musical é uma instância de “`dbo:MusicGenre`”.
- b) Qual a distância (linha recta) entre New York e London?
- c) ... e entre Lisbon e London?

Representação e Processamento de Conhecimento (RPC)
--

8. “Linked Data” e CherryPy – com Grafo Local e Remoto

- a) Estenda o código usado na questão anterior (6) de modo a apresentar a informação gerada na questão 2 (acima, nesta mesma aula prática). Ou seja, apresente numa página Web o resultado da combinação de um grafo local com um gafo remoto.
- b) Se pretender evoluir e construir uma apresentação mais complexa (realista) considere instalar uma “templating language”. *Sugestão:* linguagem “mako” (<http://www.makotemplates.org/>) que permite construir “Python Server Pages” (PSP) de modo muito simples.