

# Ontologies et

# Web Sémantique

**SPARQL – Outils et Démonstration**

# Plan du cours

1. DBpedia SPARQL Endpoint
2. Jena Apache ARQ
3. Fuseki Server
4. Python SPARQLWrapper

# Quelques Outils - Démo

## o. DBpedia SPARQL Endpoint

- Onglet 1 - Go to : <http://fr.dbpedia.org/sparql>
- Onglet 2 – Go to : [http://fr.dbpedia.org/page/Bernard\\_Stiegler](http://fr.dbpedia.org/page/Bernard_Stiegler)

SPARQL Query Editor

About

Tables ▼

Conductor

Facet Browser

Permalink

Extensions: cxml save to dav sponge User: SPARQL

Default Data Set Name (Graph IRI)

Query Text

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?phils

WHERE {
  ?phils dbo:influenced dbr:Bernard_Stiegler .
}
```

Results Format

HTML ▼

Execute Query

Reset

## Quelques Outils - Démo

### o. DBPedia SPARQL Endpoint

- Onglet 1 - Go to : <http://fr.dbpedia.org/sparql>
- Onglet 2 – Go to : [http://fr.dbpedia.org/page/Bernard\\_Stiegler](http://fr.dbpedia.org/page/Bernard_Stiegler)
- Exécuter la requête :

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?phils

WHERE {
    ?phils    dbo:influenced    dbr:Bernard_Stiegler .
}
```

## Quelques Outils - Démo

### o. DBPedia SPARQL Endpoint

- Onglet 1 - Go to : <http://fr.dbpedia.org/sparql>
- Onglet 2 – Go to : [http://fr.dbpedia.org/page/Bernard\\_Stiegler](http://fr.dbpedia.org/page/Bernard_Stiegler)
- **Exécuter la requête - Résultat:**

```
PREFIX dbr: <http://fr.dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?phils
WHERE {
    ?phils dbo:influenced dbr:Bernard_Stiegler .
}
```

**phils**

[http://fr.dbpedia.org/resource/Edmund\\_Husserl](http://fr.dbpedia.org/resource/Edmund_Husserl)

[http://fr.dbpedia.org/resource/Gilbert\\_Simondon](http://fr.dbpedia.org/resource/Gilbert_Simondon)

[http://fr.dbpedia.org/resource/Jacques\\_Derrida](http://fr.dbpedia.org/resource/Jacques_Derrida)

## Quelques Outils - Démo

### 1. Jena Apache **ARQ** – SPARQL Processor for Jena

- Pré-requis : Java JDK + Variables d'environnement **PATH**
- Télécharger Apache Jena : <https://archive.apache.org/dist/jena/binaries/>  
[apache-jena-2.10.0.zip](#)
- Dé zipper et extraire le dossier.
- Ajouter la variables d'environnement système **JENAROOT** avec comme valeur le chemin du dossier extrait Jena.
- Ajouter à la variable d'environnement **PATH** le chemin du dossier **bat** de Jena.
- Pour tester la bonne installation : cmd.exe ➔ entrer la commande : sparql --help

## Quelques Outils - Démo

### 1. Jena ARQ – SPARQL Processor for Jena

- Interroger un fichier RDF : Web ou Local

```
> sparql --data rdfFile --query sparqlFile
```

- Exemples :

```
> sparql --data D:\Bernard_Stiegler.ttl --query D:\q0.sparql
```

```
> sparql --data http://fr.dbpedia.org/data/Bernard_Stiegler.rdf  
--query D:\q0.sparql
```

```
> sparql --data D:\secondFile.ttl --data D:\Bernard_Stiegler.ttl  
--query D:\q5.sparql
```

## Quelques Outils - Démo

### 1. Jena ARQ – SPARQL Processor for Jena

- Interroger un fichier RDF : Web ou Local

```
> sparql --data rdfFile --query sparqlFile
```

- Exemples :

```
> sparql --data D:\Bernard_Stiegler.ttl --query D:\q0.sparql
```

```
7 - Files>sparql --data Bernard_Stiegler.ttl --query q0.sparql
-----
| phils |
=====
| dbr:Gilbert_Simondon |
| dbr:Jacques_Derrida |
| dbr:Edmund_Husserl |
-----
```



## Quelques Outils - Démo

### 1. Jena ARQ – SPARQL Processor for Jena

- Interroger un fichier RDF : Web ou Local

```
> sparql --data rdfFile --query sparqlFile
```

- Exemples :

```
> sparql --data http://fr.dbpedia.org/data/Bernard_Stiegler.rdf  
--query D:\q0.sparql
```

```
7 - Files>sparql --data http://fr.dbpedia.org/data/Bernard_Stiegler.rdf --qu  
ery q0.sparql  
-----  
| phils |  
=====  
| dbr:Gilbert_Simondon |  
| dbr:Jacques_Derrida |  
| dbr:Edmund_Husserl |  
-----
```

## Quelques Outils - Démo

### 1. Jena ARQ – SPARQL Processor for Jena

- Interroger un fichier RDF : Web ou Local

```
> sparql --data rdfFile --query sparqlFile
```

- Exemples :

```
> sparql --data D:\secondFile.ttl --data D:\Bernard_Stiegler.ttl  
--query D:\q5.sparql
```

```
> sparql --data Bernard_Stiegler.ttl --data secondFile.ttl --query q5.sparql
```

phils	name
dbr:Jacques_Derrida	"Jacques Derrida"@en
dbr:Bernard_Stiegler	"Bernard Georges Alain Stiegler"@en

## Quelques Outils - Démo

Motif Optionnel - **OPTIONAL { ..... }**

OPTIONAL correspond à une **jointure à gauche**. Les éléments du graphe de gauche sont conservés même s'ils ne répondent pas à la clause présente dans la partie optionnelle (à droite).

Données :

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

_:a rdf:type foaf:Person .
_:a foaf:name "Stiegler" .
_:a foaf:mbox <mailto:stiegler@example.com> .
_:a foaf:mbox <mailto:stiegler@work.example> .

_:b rdf:type foaf:Person .
_:b foaf:name "Simondon" .
```

## Quelques Outils - Démo

Motif Optionnel - **OPTIONAL { ..... }**

Requête :

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE {
    ?res foaf:name ?name .
    OPTIONAL { ?res foaf:mbox ?mbox . }
}
```

Résultat de requête :

name	mbox
"Simondon"	
"Stiegler"	<mailto:stiegler@work.example>
"Stiegler"	<mailto:stiegler@example.com>

## Quelques Outils - Démo

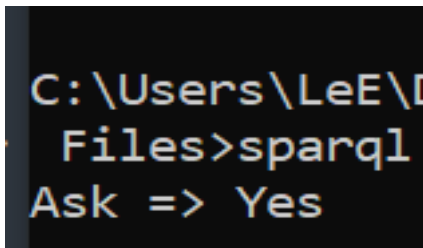
### 1. Jena ARQ – SPARQL Processor for Jena

- Interroger un fichier RDF : Web ou Local

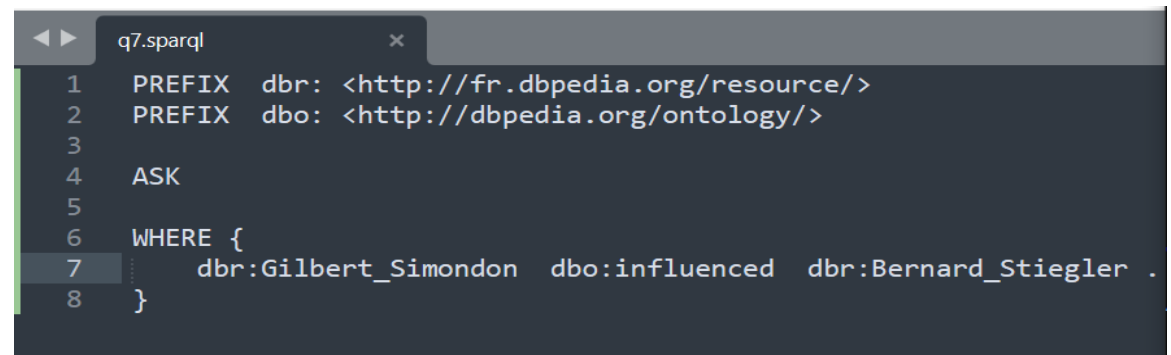
```
> sparql --data rdfFile --query sparqlFile
```

- Exemples :

```
> sparql --data http://fr.dbpedia.org/data/Bernard_Stiegler.rdf  
--query D:\q7.sparql
```



A terminal window showing the execution of a SPARQL query. The prompt is 'C:\Users\LeE\Documents>' and the command is 'sparql'. The output is 'Ask => Yes'.



A terminal window showing the contents of a SPARQL query file named 'q7.sparql'. The query is an ASK query with two prefixes and a WHERE clause.

```
q7.sparql  
1 PREFIX dbr: <http://fr.dbpedia.org/resource/>  
2 PREFIX dbo: <http://dbpedia.org/ontology/>  
3  
4 ASK  
5  
6 WHERE {  
7   dbr:Gilbert_Simondon dbo:influenced dbr:Bernard_Stiegler .  
8 }
```

## Quelques Outils - Démo

### 1. Jena ARQ – SPARQL Processor for Jena

- Interroger différents fichiers RDF avec des graphes nommés - remote

```
> rsparql --service endpointURL --query sparqlFile
```

- Exemple : DBpedia Endpoint

```
> rsparql --service http://fr.dbpedia.org/sparql --query  
D:\q1.sparql
```

```
rsparql --service http://fr.dbpedia.org/sparql --query q1.sparql  
res  
=====  
<http://wikidata.dbpedia.org/resource/Q10441046>  
<http://wikidata.dbpedia.org/resource/Q11949802>  
<http://wikidata.dbpedia.org/resource/Q16623069>  
<http://wikidata.dbpedia.org/resource/Q1669814>  
<http://wikidata.dbpedia.org/resource/Q1684159>  
<http://wikidata.dbpedia.org/resource/Q2238942>  
<http://wikidata.dbpedia.org/resource/Q5600590>  
<http://wikidata.dbpedia.org/resource/Q7948865>  
<http://wikidata.dbpedia.org/resource/Q897815>
```

## Quelques Outils - Démo

### 1. Jena ARQ – SPARQL Processor for Jena

- Interroger différents fichiers RDF avec des graphes nommés - local

```
> sparql --graph loadFileAsDefaultGraph  
--namedGraph loadFileAsNamedGraph --query ...
```

- Exemple : (TD 3 – Exo 3)

```
> sparql --graph fileexo1.ttl --namedGraph newfile.ttl --  
query D:\q4.sparql
```

```
C:\Users\lee\Documents\cours\mes modules\module 6\6.1 - Organized\cours\cours  
> sparql --graph fileexo1.ttl --namedGraph newfile.ttl --query q4.sparql  
-----  
| single      | res      |  
=====
```

dbr:El_Dorado	dbr:El_Dey
	dbr:Freeklane

```
-----
```

## Quelques Outils - Démo

### 1. Jena ARQ – SPARQL Processor for Jena

```
> sparql --graph fileexo1.ttl --namedGraph newfile.ttl --  
query D:\q4.sparql
```

```
SELECT ?single ?res
```

```
FROM <fileexo1.ttl>
```

```
FROM NAMED <newfile.ttl>
```

```
WHERE {
```

```
  { ?single dbo:composer dbr:Steve_Harris }
```

```
  UNION
```

```
  { GRAPH <newfile.ttl> { ?res rdf:type <http://schema.org/MusicGroup> } }
```

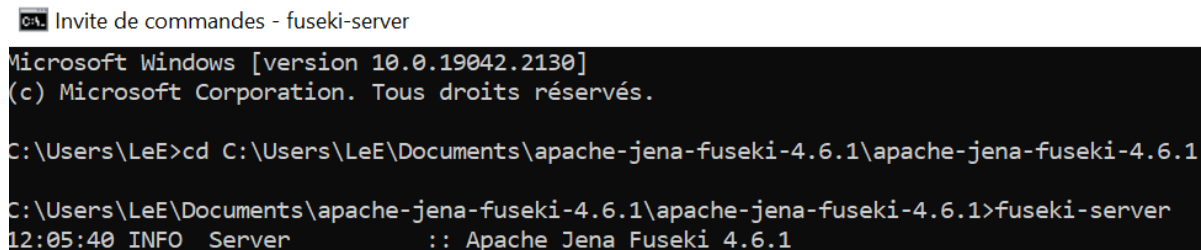
```
}
```



## Quelques Outils - Démo

### 2. Jena Apache Fuseki – SPARQL Endpoint

- SPARQL endpoints provide an access point for receiving and processing SPARQL protocol requests.
- Télécharger le serveur Fuseki: <https://jena.apache.org/download/>
- *apache-jena-fuseki-4.6.1.zip*
- Dézipper et extraire le dossier.
- Démarrer le serveur Fuseki : Dans un CMD, se placer sur le répertoire *apache-jena-fuseki-4.x.x* où vous avez décompressé jena-fuseki. Puis, exécuter la commande : > *fuseki-server*



```
Microsoft Windows [version 10.0.19042.2130]
(c) Microsoft Corporation. Tous droits réservés.

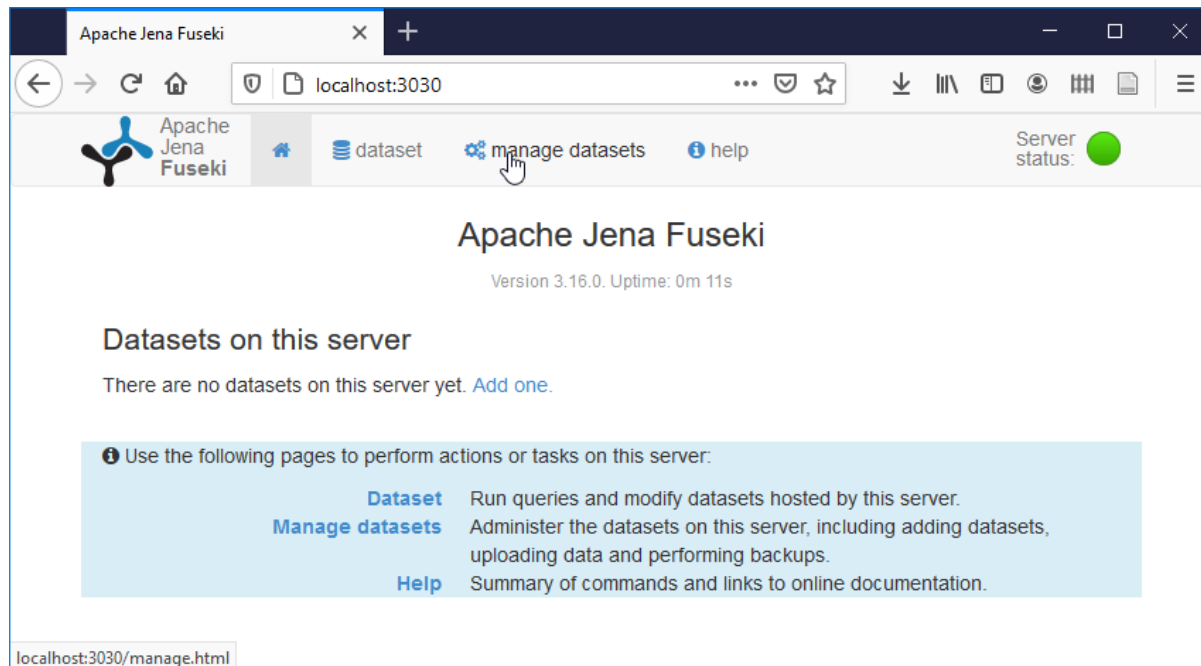
C:\Users\LeE>cd C:\Users\LeE\Documents\apache-jena-fuseki-4.6.1\apache-jena-fuseki-4.6.1

C:\Users\LeE\Documents\apache-jena-fuseki-4.6.1\apache-jena-fuseki-4.6.1>fuseki-server
12:05:40 INFO Server :: Apache Jena Fuseki 4.6.1
```

## Quelques Outils - Démo

### 2. Jena Apache Fuseki – SPARQL Endpoint

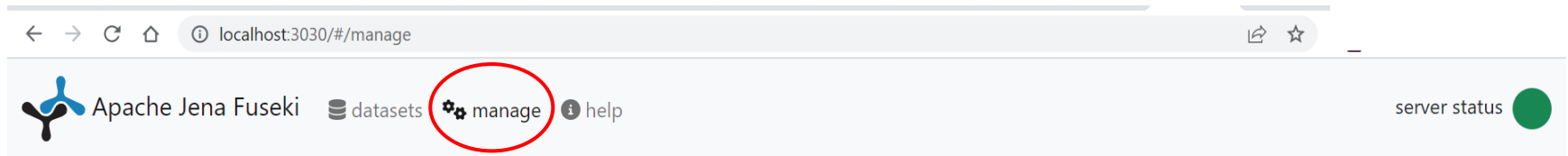
- SPARQL endpoints provide an access point for receiving and processing SPARQL protocol requests.
- Pour vérifier le bon fonctionnement de Fuseki, entrer l'url <http://localhost:3030> dans votre navigateur.



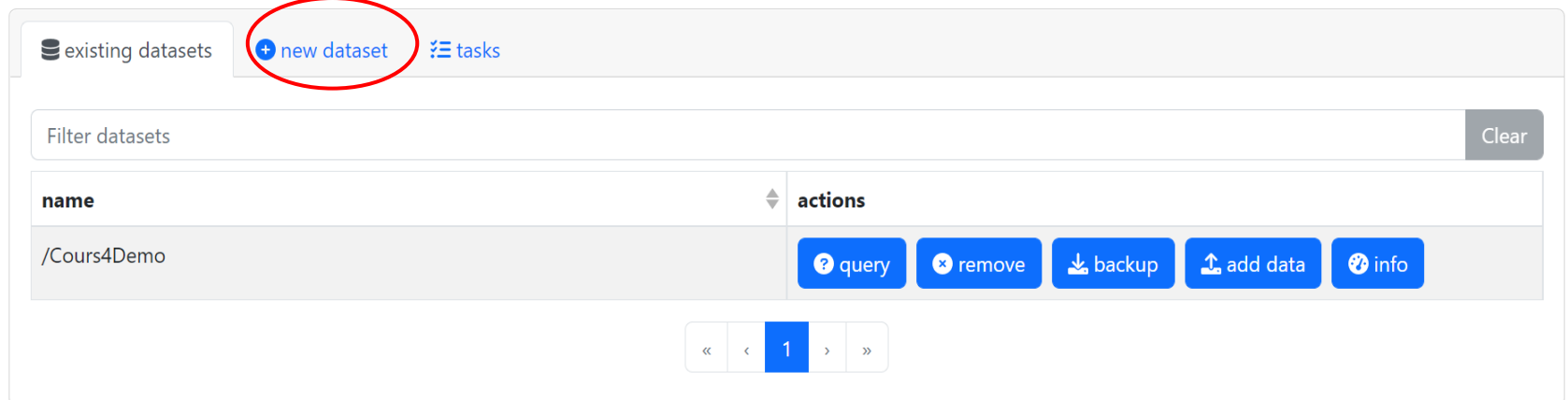
## Quelques Outils - Démo

### 2. Jena Apache Fuseki – SPARQL Endpoint

- Charger un **jeu de données** RDF dans le serveur Fuseki:
- Onglet *manage* datasets > new datasets >



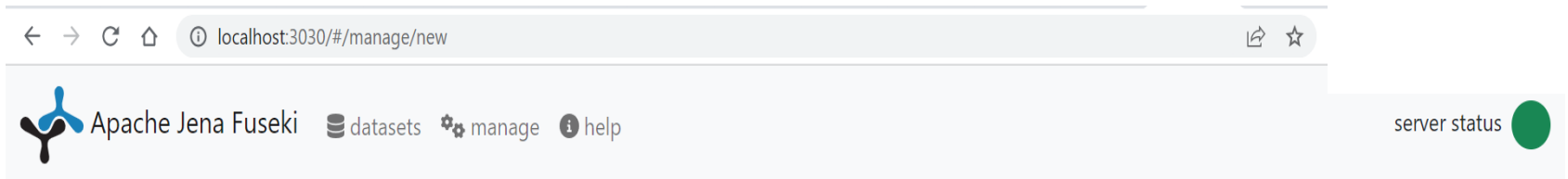
### Manage datasets



## Quelques Outils - Démo

### 2. Jena Apache Fuseki – SPARQL Endpoint

- Charger un jeu de données RDF dans le serveur Fuseki:
- Onglet [manage](#) datasets > new dataset > name dataset > create dataset



### New dataset

[existing datasets](#) **+ new dataset** [tasks](#)

Dataset name

MyRDFDataset

Dataset type

☐ In-memory – dataset will be recreated when Fuseki restarts, but contents will be lost






☒ Persistent (TDB2) – dataset will persist across Fuseki restarts

✓ create dataset

# Quelques Outils - Démo

## 2. Jena Apache Fuseki – SPARQL Endpoint

- Charger un jeu de données RDF dans le serveur Fuseki:
- Onglet [manage](#) > existing datasets > My Dataset > [add data](#) > [select files](#) (XML, Turtle, etc.) > [upload files](#)

 Apache Jena Fuseki  datasets  manage  help server status 

### /MyRDFDataset

[query](#) [add data](#) [edit](#) [info](#)

#### Upload files

Load data into the default graph of the currently selected dataset, or the given named graph. You may upload any RDF format, such as Turtle, RDF/XML or TriG.

Dataset graph name

Files to upload

[+ select files](#) [upload all](#)

name	size	speed	status	actions
No files selected				

# Quelques Outils - Démo

## 2. Jena Apache Fuseki – SPARQL Endpoint

- Charger un jeu de données RDF dans le serveur Fuseki:
- Onglet [manage](#) datasets > [Query](#)

The screenshot shows the Jena Apache Fuseki web interface. At the top, there are tabs for 'query', 'add data', 'edit', and 'info'. The 'query' tab is active. Below the tabs, the title 'SPARQL Query' is displayed, followed by the instruction: 'To try out some SPARQL queries against the selected dataset, enter your query here.' Under 'Example Queries', there are two buttons: 'Selection of triples' (highlighted in blue) and 'Selection of classes'. To the right, under 'Prefixes', there are buttons for 'rdf', 'rdfs', 'owl', and 'xsd'. Below these, there are three input fields: 'SPARQL Endpoint' with the value '/Cours4Demo/', 'Content Type (SELECT)' with a dropdown menu showing 'JSON', and 'Content Type (GRAPH)' with a dropdown menu showing 'Turtle'. A red circle highlights a share icon and a play button icon in the bottom right corner of the query editor area. The query editor contains the following SPARQL query:

```
1 PREFIX dbr: <http://fr.dbpedia.org/resource/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3
4 SELECT ?phils ?name
5
6 WHERE {
7   ?phils dbo:birthName ?name .
8 }
```

At the bottom of the interface, there are tabs for 'Table' and 'Response', and a download icon.

# Quelques Outils - Démo

## 2. Jena Apache Fuseki – SPARQL Endpoint

- Charger un jeu de données RDF dans le serveur Fuseki:
- Onglet **manage** datasets > **Query response**

Table

Response

2 results in 0.019 seconds

Simple view ☐ Ellipse ☒

Page size: 50

phils	name
1 < <a href="http://fr.dbpedia.org/resource/Bernard_Stiegler">http://fr.dbpedia.org/resource/Bernard_Stiegler</a> >	"Bernard Georges Alain Stiegler"@en
2 < <a href="http://fr.dbpedia.org/resource/Jacques_Derrida">http://fr.dbpedia.org/resource/Jacques_Derrida</a> >	"Jacques Derrida"@en

Showing 1 to 2 of 2 entries

< 1 >

## Quelques Outils - Démo

### 2. Python **SPARQLWrapper** – SPARQL Endpoint

- Interroger un endpoint SPARQL en Python.
- Installer le package avec : 

```
> pip install sparqlwrapper
```
- Documentation : <https://sparqlwrapper.readthedocs.io/en/stable/>
- RDFLib/SPARQLWrapper on GitHub : <https://github.com/RDFLib/sparqlwrapper/>
- Coder. Exemple : see `sparql_example.py`



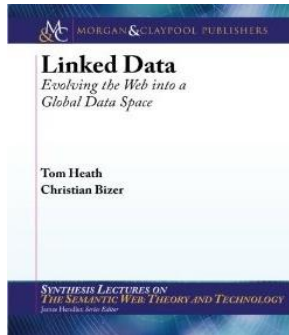


# Quelques Outils - Démo

## 2. Python **SPARQLWrapper** – SPARQL Endpoint

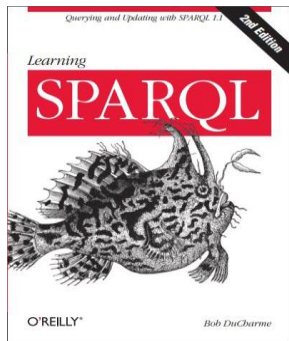
```
File Edit Selection Find View Goto Tools Project Preferences Help
sparql_example.py x
1  from SPARQLWrapper import SPARQLWrapper, JSON
2
3
4  sparql = SPARQLWrapper("http://fr.dbpedia.org/sparql")
5  # sparql = SPARQLWrapper("http://localhost:3030/Cours4Demo/sparql")
6
7  sparql.setQuery("""
8      PREFIX dbr: <http://fr.dbpedia.org/resource/>
9      PREFIX dbo: <http://dbpedia.org/ontology/>
10
11      SELECT DISTINCT ?phils |
12      WHERE {
13          ?phils dbo:influenced dbr:Bernard_Stiegler .
14      }
15
16      """)
17
18  sparql.setReturnFormat(JSON)
19
20  try:
21      ret = sparql.queryAndConvert()
22
23      for r in ret["results"]["bindings"]:
24          print(r['phils']['value'])
25  except Exception as e:
26      print(e)
```

# Références



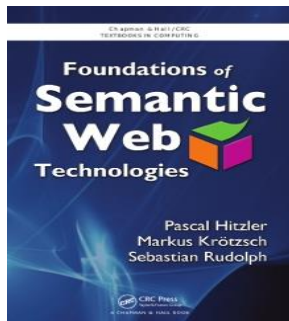
## **Linked Data: Evolving the Web into a Global Data Space**

- ✓ Auteur : Christian Bizer, Tom Heath
- ✓ Éditeur : Morgan & Claypool Publishers
- ✓ Edition : Février 2011 - 136 pages - ISBN 9781608454310



## **Learning SPARQL : Querying and Updating with SPARQL**

- ✓ Auteur : Bob DuCharme
- ✓ Éditeur : O'Reilly Media
- ✓ Edition: Juillet 2013– 386pages -ISBN : 9781449306595



## **Foundations of Semantic Web Technologies**

- ✓ Auteur : Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph
- ✓ Éditeur : CRC Press/Chapman and Hall
- ✓ Edition : 2009 - 455 pages - ISBN : 9781420090505