Université de Paris **Bases de données spécialisées**
M2 Info et Maths-Info **Année 2025-2026**

TP Web Sémantique - Séance 1

Manipulation RDF avec Jena

1 Environnement de travail

Apache Jena est un framework Java gratuit et open source pour les développement d'applications Semantic Web et Linked Data. Il consiste en plusieurs API qui interagissent pour le traitement des données RDF. Il vient avec un moteur pour le stockage des triplets RDF, et fournit également des outils en ligne de commande pour la gestion et interrogation des triplets.

Principales composantes Jena :

RDF API : le core des API Jena. Fournit les packages Java pour la representation et manipulation des triplets RDF

Moteurs de stockage : Supportent le stockage des triplets RDF aussi bien en mémoire, que dans une base de données SQL (moteur SQL DB), que dans une triple store persistent natif (moteur TDB). Ils peuvent être utilisés depuis Java via la "Store API".

ARQ (SPARQL) : un query processor SPARQL pour interroger et modifier des graphes RDF. Peut exécuter des requêtes sur des fichiers RDF chargés depuis la machine locale ou depuis le Web, ou sur des SPARQL endpoints externes comme DBpedia. Il peut être utilisé depuis Java par la SPARQL API. Il peut être utilisé également par l'outil en ligne de commande `sparql`.

Fuseki : fournit un serveur SPARQL qui utilise TDB pour le stockage persistent des triplets et permet l'interrogation de ces données via HTTP (SPARQL endpoint). À utiliser pour publier des données ou partager un dataset parmi plusieurs applications.

Inference subsystem : Le système de raisonnement de Jena. Permet la dérivation d'assertions RDF additionnelles impliquées par une base RDF et une ontologie optionnelle, par les règles d'inférence propre au moteur d'inférence utilisé ("reasoner"). Les langages d'ontologies nativement supportés sont RDFS et OWL, mais d'autres "reasoners" externes peuvent être branchés sur le système de raisonnement Jena. Accessible depuis Java par les "Inférence API" et "Ontology API".

Parsers/Writers : Pour la lecture et écriture de graphes RDF. Implémentent la technologie "RDF IO Technology" (RIOT). Supportent plusieurs formats de sérialisation (Turtle RDF/XML, N-Triples, RDF/JSON, etc.)

Outils en ligne de commande : Pour la lecture, interrogation et manipulation de données RDF.

Pour plus de détails vous êtes invités à consulter la documentation Jena :
`https://jena.apache.org/documentation/`

2 Installation

1. Si Java n'est pas encore installé, le télécharger et l'installer depuis `www.java.com`.

2. Télécharger la dernière release Jena depuis `http://jena.apache.org/` (Binary distribution sans Fuseki)
3. Extraire l'archive téléchagé dans un répertoire, par exemple :
   — Windows : `C:\Program Files\Jena`
   — OS X : `/Applications/Jena`
   — Linux : `/usr/local/Jena`
4. JENAROOT : Ajouter une variable d'environnement [1] appelée JENAROOT dont la valeur est le nom du répertoire d'extraction.
5. PATH : Ajouter le répertoire ci dessous à la variable d'environnement PATH [1]
   — Windows : `C:\Program Files\Jena\bat`
   — OS X : `/Applications/Jena/bin`
   — Linux : `/usr/local/Jena/bin`
6. Nouvelle fenêtre shell : Le parametrage ci-dessus prend effet à l'ouverture d'une nouvelle fenêtre shell. Vérifier que ARQ est correctement installé en tapant à l'invite du shell : `sparql --help`

3 Transformation RDF

Les commandes `rapper` et `riot` permettent de transformer des fichiers RDF d'un format à un autre :

3.1 riot

La commande `riot` parse (et valide) un fichier RDF d'input en reconnaissant la syntaxe à partir de l'extension du fichier :
— N3 : format N3/Turtle (extension .ttl)
— N-TRIPLE : format triplets (extension .nt)
— RDF/XML : format XML (extension .rdf)
— RDF/XML-ABBREV : format XML affiché de facon lisible (extension .xml)
Le format de l'output peut être choisi avec l'option `-out`
Exemple :

```
1 riot -out=RDF/XML-ABBREV chemin/fichier.ttl
```

L'output est affiché sur le terminal, le rediriger dans un fichier permet de le sauvegarder :

```
1 riot -out=RDF/XML-ABBREV chemin/fichier.ttl > chemin/fichier.xml
```

Taper `riot -help` pour connaître plus d'options.

Exercice 1 Ouvrir avec un éditeur de texte le fichier `uris.rdf`.

1. Deduire les URIs complets de chaque ressource dans ce fichier.
2. Utiliser la commande `riot` pour convertir `uris.rdf` en format Turtle, cela explicitera les URI des ressources, verifier que cela correspond à ce que vous avez trouvé au point 1.

Exercice 2 Refaire l'Exercice 1 mais en partant du fichier `uris1.ttl` pour le convertir ensuite en format RDF/XML.

1. chercher sur le Web comment faire pour votre OS

3.2 rapper

La commande `rapper` (Raptor RDF syntax parsing and serializing utility `http://librdf.org/raptor/` convertit un fichier d'un format à l'autre. Elle doit être installée indépendamment de Jena. [2]

formats d'input :
— `rdfxml` RDF/XML (default)
— `ntriples` N-Triples
— `turtle` Turtle Terse RDF Triple Language
— `trig` TriG - Turtle with Named Graphs
— `rss-tag-soup` RSS Tag Soup
— `grddl` Gleaning Resource Descriptions from Dialects of Languages
— `guess` Pick the parser to use using content type and URI
— `rdfa` RDF/A via librdfa
— `json` RDF/JSON (either Triples or Resource-Centric)
— `nquads` N-Quads

formats d'output :
— `ntriples` N-Triples (default)
— `turtle` Turtle Terse RDF Triple Language
— `rdfxml-xmp` RDF/XML (XMP Profile)
— `rdfxml-abbrev` RDF/XML (Abbreviated)
— `rdfxml` RDF/XML
— `rss-1.0` RSS 1.0
— `atom` Atom 1.0
— `dot` GraphViz DOT format
— `json-triples` RDF/JSON Triples
— `json` RDF/JSON Resource-Centric
— `html` HTML Table
— `nquads` N-Quads

Par exemple la commande :

```
1 rapper -i rdfxml -o turtle data/rois.xml > gendata/rois.ttl
```

convertit le fichier `rois.xml` du format `rdfxml` au format `turtle`.

Exercice 3 Testez la commande `rapper` ci-dessus sur le fichier `rois.xml`. Ouvrez le fichier généré `rois.ttl` et analysez-le pour bien comprendre la correspondence entre les triplets et les elements XML d'origine. Remarquez ensuite que dans `rois.xml` les attributs `rdf:resource` ont valeur `'rois#ri'`, est-ce possible de changer ces valeurs d'attributs sans changer l'URI des ressources qu'ils représentent ? Faites une proposition et testez (c'est à dire changez les URIs et re-faites la conversion Turtle, le résultat doit être identique).

3.3 Transformation en graphe

La commande `rapper` permet en particulier de transformer un fichier RDF en graphe dans le format `dot` pour la visualisation.

Transformation d'un graphe RDF (Turtle) en format dot :

```
rapper -i rdfxml -o dot data/rois.xml > gendata/rois.dot
```

2. une source utile pour trouver rapidement comment faire pour votre OS : `https://command-not-found.com/rapper`

On peut ensuite utiliser le software Graphviz pour convertir le format dot en SVG. Pour ce faire installer Graphviz depuis `https://graphviz.org/download/`. Puis

```
dot -Tsvg gendata/rois.dot > gendata/rois.svg
```

Ou de facon equivalente, pour une conversion directe de RDF/XML à SVG composer :

```
rapper -i rdfxml -o dot data/rois.xml | dot -Tsvg > gendata/rois.svg
```

Le fichier `rois.svg` peut être visualisé avec un navigateur (e.g. Firefox).

3.4 Transformations en différents formats

Exercice 4 Transformez les deux documents `data/roisgraphe.rdf` et `data/roisarbre.rdf` dans différents formats et comparez les documents générés dans le répertoire `gendata` :

Est-ce que les deux documents représentent le même graphe RDF ? Affichez les graphes générés depuis `data/roisgraphe.rdf` et `data/roisarbre.rdf`.

Exercice 5 Modifiez le document `roisgraphe.rdf` en le renommant `roisgraphe-anonyme.rdf`. Dans le nouveau document le rois "r4" sera remplacé par un noeud anonyme. Vérifiez que vous avez bien fait la modification en transformant le document en format N3/Turtle et en N-TRIPLE. Comparez par ailleurs l'encodage du noeud anonyme dans ces deux formats.

4 Stockage RDF

Jena fournit un système de stockage pour des données RDF appelé Jena TDB [3]. Les commandes disponibles sont :

— Pour créer (si nécessaire) une base de données RDF et y charger des triplets depuis un fichier (de préférence en format Turtle) :

```
tdbloader --loc=<db> <fichiers>
```

— Pour afficher le contenu d'une base de données :

```
tdbdump --loc=<db>
```

— Pour afficher des statistiques :

```
tdbstats --loc=<db>
```

— Pour effacer la base de données :

```
rm -rf <db>
```

Exercice 6 Créez une base RDF `db/rois.tdb` et chargez les graphes `data/rois.xml` et `data/rois_schema.rdfs`.

Exercice 7 Afficher le contenu et les statistiques du (des) graphe(s) RDF dans `db/rois.tdb`. Combien de triples, de reines (ressources de type Reine) et de rois (ressources de type Roi) sont dans la base ?

Exercice 8 Rechargez les deux fichiers `data/rois.xml` et `data/rois_schema.rdfs` une deuxième fois et affichez les statistiques. Qu'est-ce que vous observez ?

3. http ://jena.apache.org/documentation/tdb/

Exercice 9 Créez une base RDF `db/roisblanc.tdb` et chargez le graphe `roisgraphe-anonyme.rdf` Affichez le contenu et les statistiques de `db/roisblanc.tdb`. Combien de rois il y a dans `db/roisblanc.tdb` ?

Rechargez le fichier `data/roisgraphe-anonyme.rdf` une deuxième fois et affichez les statistiques. Qu'est-ce que vous observez ?

5 RDFLib

RDFLib est un package Python pur pour travailler avec RDF. Il contient :

Analyse et sérialisation

— RDF/XML, N3, NTriples, N-Quads, Turtle, TriX, JSON-LD, HexTuples, RDFa, Microdata

Stockage

— Stockages en mémoire
— Stockages persistants, sur disque, utilisant des bases de données telles que `BerkeleyDB`
— Points d'accès SPARQL distants

Implémentation de SPARQL 1.1

— requêtes et mises à jour

Notez que RDFLib ne supporte pas directement l'inférence. Il peut toutefois être couplé avec Jena ou avec des reasoners externes

Exercice 10 S'assurer d'avoir python installé sur sa machine. Se renseigner sur RDFLib à l'adresse `https://rdflib.readthedocs.io/en/stable/`. Refaire les exercices 1, 2 et 4 en utilisant RDFlib à la place de riot / jena.