

TP

RDF & Turtle & Jena

Travail à rendre : Le code source et fichier générés

API Jena

Jena est une API qui permet de manipuler des fichiers RDF, ainsi que des ontologies RDFS et OWL. Elle fournit aussi une implémentation d'un moteur SPARQL pour réaliser des requêtes sur des données RDF.

Installation

La première étape consiste à installer tous les outils dont on aura besoin pour réaliser ce qui est demandé.

1. Installer **JDK (Java)** et **Eclipse** et n'oublier pas de mettre à jour les variables d'environnements (`JAVA_HOME="C:\Program Files\Java\jdk-x.x.x"`, `PATH=PATH + "C:\Program Files\Java\jdk-x.x.x\bin"`).
2. Télécharger ensuite Jena ([apache-jena-4.1.0.zip](#)).
3. Décompresser le fichier `apache-jena-4.1.0.zip` dans le répertoire "`C:\Jena`" (à créer)
4. Définissez la variable d'environnement `JENA_HOME` à l'emplacement où vous avez décompressé **Jena** :
`JENA_HOME = C:\jena\apache-jena-4.1.0`
5. Ajouter le chemin "`C:\jena\apache-jena-4.1.0\bat`" à la variable d'environnement `PATH`

Création d'un projet Eclipse et ajout des librairies Jena

Une fois toutes les *installations* (**JDK**, **Eclipse** et **Jena**) et les *configurations* (variables d'environnements) nécessaires sont faites correctement, la deuxième étape consiste à créer un **nouveau projet java** dans Eclipse et d'y **intégrer Jena** en tant que bibliothèque :

1. Créer un nouveau projet java et donner lui le nom **TpJena** :
Sous **JRE**, sélectionnez **Utiliser l'environnement par défaut (Use Default JRE)**
2. Ajouter ensuite les bibliothèques Jena dans votre projet :
 - 2.1. Exécuter la commande **Fenêtre->Préférences (windows->Preferences)**.
 - 2.2. Dans la boîte de dialogue **Préférences**, choisissez : **Java + Chemin de Compilation + Bibliothèque utilisateur (Java + Build Path + User Libraries)**; puis cliquez sur **Nouveau (New)**.
 - 2.3. Entrer le nom **BibJena** ; puis cliquer **OK**
 - 2.4. Ajouter les fichiers **JAR** de Jena en cliquant sur **Ajout des fichiers JAR externes (Add External JARs)**.
 - 2.5. Naviguer jusqu'au répertoire lib de Jena (`C:\jena\apache-jena-4.1.0\lib`).
 - 2.6. Sélectionner l'ensemble des fichiers JAR ; puis cliquez **Ouvrir**
3. Associer le code source
 - 3.1. Défilez la barre déroulante et localisez le fichier `jena-core-4.1.0.jar`.
 - 3.2. Sélectionnez **Connexion de la source (Source Attachment)**; puis cliquez **Modifier (Edit)**

- 3.3. Sélectionnez **External Location** ; puis cliquez **Fichier externe (External File)**.
- 3.4. Dans le répertoire **C:\Jena\apache-jena-4.1.0\lib-src** sélectionnez le fichier **jena-core-4.1.0-sources.jar** ; puis cliquez **Ouvrir**
- 3.5. Refaire la même opération pour les fichiers : **jena-arq-4.1.0-sources.jar**, **jena-cmds-4.1.0-sources.jar**, **jena-rdfconnection-4.1.0-sources.jar**, **jena-shacl-4.1.0-sources.jar**, **jena-tdb-4.1.0-sources.jar**.
- 3.6. Puis cliquez **ok (Apply an close)**
4. Attacher **BibJena** au projet. Pour ce faire :
 - 4.1. Avec le bouton de droite de la souris, Cliquer sur le projet **TpJena**.
 - 4.2. Choisir l'élément de menu **Chemin de génération->Ajouter des bibliothèques (build Path->Configure build Path)**
 - 4.3. Dans l'onglet **Libraries**, cliquer **Add Library**
 - 4.4. Choisir **Bibliothèque utilisateur (User Library)**; puis **Suivant**
 - 4.5. Sélectionnez **BibJena** ; puis **Terminer**

Jena est une API Java qui peut être utilisée pour créer et manipuler des graphes RDF. Jena possède des classes pour représenter des graphes, des ressources, des propriétés et des littéraux. Les interfaces représentant les ressources, les propriétés et les littéraux sont respectivement nommées **Resource**, **Property** et **Literal**. Dans Jena, un graphe est appelé un modèle et est représenté par l'interface **Model**.

5. Pour votre premier test :
 - 5.1. Créer une nouvelle classe **TestJava** sous un Paquetage : **tpjena** et modifier la de telle sorte à correspondre au code suivant :

```
package tpjena;

public class TestJena {
    public static void main(String[] args) {
        Model monModele = ModelFactory.createDefaultModel();
        // Créer une ressource MESSAGE
        Resource message = monModele.createResource("http://webtutoriel/message");
        // Créer une propriété VALEUR
        Property valeurProp = monModele.createProperty("http://webtutoriel/valeur");
        // Assigner une valeur à la ressource
        message.addProperty(valeurProp, "Bonjour à tous");
        // Imprimer le contenu du modèle dans la syntaxe TTL
        System.out.println("Graphe en Rurtle:\n");
        monModele.write( System.out , "TTL");
        // Imprimer le contenu du modèle dans la syntaxe RDF/XML
        System.out.println("\nGraphe en RDF/XML:\n");
        monModele.write( System.out);
    }
}
```

- 5.2. Corriger les erreurs produites dans le code. Choisissez par exemple le correctif **importer ModelFactory** pour corriger l'erreur à **ModelFactory**

En d'autres termes compléter votre code par l'ajout de des lignes suivantes :

```
import org.apache.jena.rdf.model.Model;
import org.apache.jena.rdf.model.ModelFactory;
import org.apache.jena.rdf.model.Property;
import org.apache.jena.rdf.model.Resource;
```

- 5.3. Compléter et exécuter ensuite ce code

Exercice 1

1. Écrire un programme basé sur Jena qui implémente le graphe de la **figure 1**
2. **Générer** un fichier **Tp3Jena.xml** contenant le code **RDF/XML** du graphe construit par la question 1
3. **Générer** un fichier **Tp3Jena.ttl** contenant le code **TURTLE** du graphe construit par la question 1
4. Écrire un programme basé sur Jena qui **charge** un document **RDF/XML** et qui l’affiche sur la sortie standard sous forme d’un graphe RDF en **TURTLE**.

Remarque

Adresses utiles :

- <https://jena.apache.org/download/index.cgi>
- https://jena.apache.org/tutorials/rdf_api.html
- <https://jena.apache.org/documentation/javadoc/jena/>
- <https://michel-heon.developpez.com/tutoriels/jena/developpement-eclipse-applications-semantiques/>

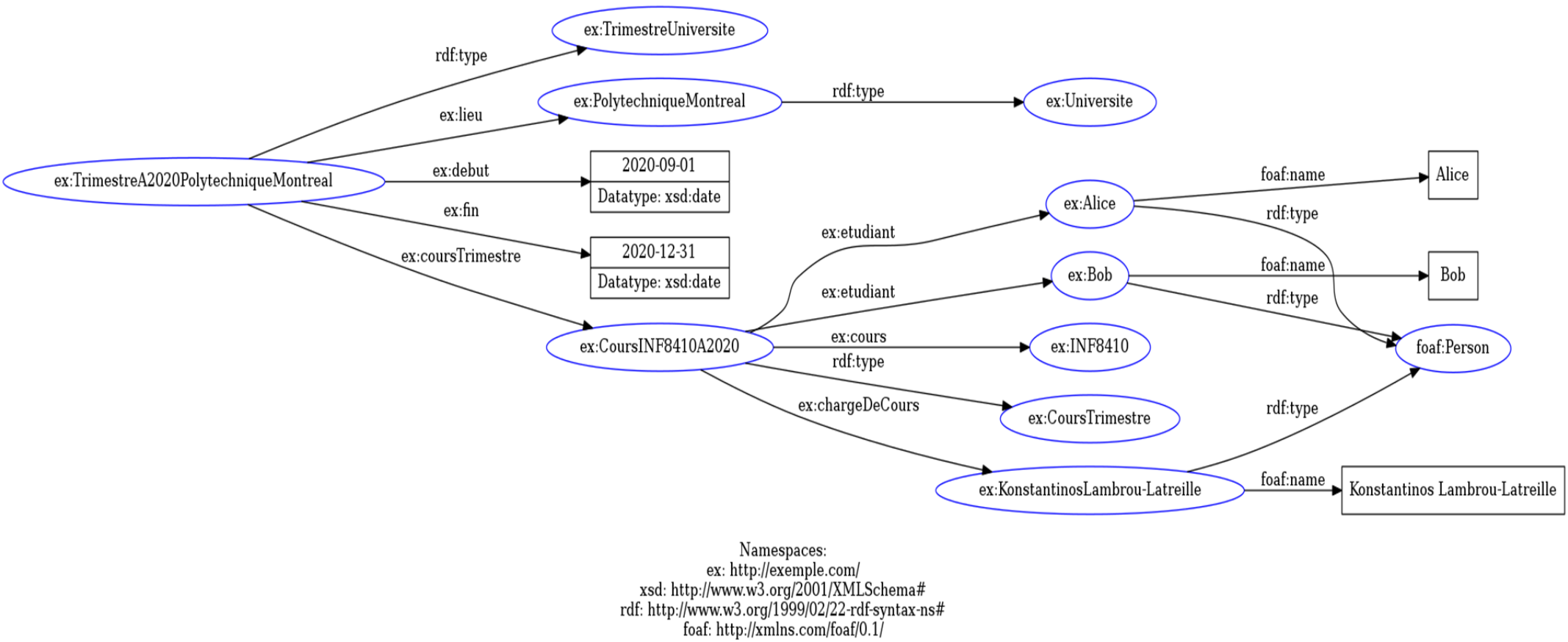


Figure 1