

## Feuille de T. P. 5 : Interroger et ajouter des règles à une ontologie en java

Se connecter sous Linux

S'assurer que la version de java est 1.8 (sinon l'installer)

Télécharger les fichiers comprimés dans AMeTICE :

Section **POUR-TP-ING-WEBSEM**

Celle-ci contient :

- un fichier `pizza_1.owl`
- un dossier comprimé **library-OWL-API-2020** : qui contient de les librairies utiles pour le TP
- un tutoriel OWL-API

Créer un nouveau projet java (java version 1.8) sous Eclipse

Sauvegarder :

- un tutoriel OWL-API
- toutes les librairies du dossier `library-OWL-API -2020`
- la documentation OWL-API : <http://owlapi.sourceforge.net/documentation.html> qui contient un tutoriel sur *OWL – API* et des exemples de code java.
- le tutoriel : [http://owlapi.sourceforge.net/owled2011\\\_tutorial.pdf](http://owlapi.sourceforge.net/owled2011\_tutorial.pdf)
- exemples de codes : <https://github.com/phillord/owl-api/blob/master/contract/src/test/java/org/code/owlapi/examples/Examples.java>

ATTENTION : lorsqu'on travaille avec un fichier owl crée avec Protégé sauvegardé dans un répertoire de l'utilisateur, vérifier que l'IRI (sous Linux) du fichier est bien : `file:///Users/chemin_jusqu'au_fichier/nom_fichier.owl`

## Interrogation avec SPARQL

On va utiliser un protocole HTTP client /serveur pour interroger l'ontologie via le serveur sparql fuseki sur lequel l'ontologie `pizza_1.owl` est chargée. Le endpoint est l'url : `http://139.124.41.79:8080/fuseki/pizza_1/query`

**documentation**

<https://www.w3.org/TR/sparql11-http-rdf-update/>

[http://linkedbuildingdata.net/ldac2019/summerschool/files/07\\_Bonduel\\_triplestores\\_SPARQL\\_endpoints.pdf](http://linkedbuildingdata.net/ldac2019/summerschool/files/07_Bonduel_triplestores_SPARQL_endpoints.pdf)

<https://jena.apache.org>  
<https://jena.apache.org/documentation/fuseki2/index.html>

## Interrogation avec SQWRL

Reprendre la classe java *OWLAPI\_StepByStep* des TP 3, et 4.  
Rappel : la classe *OWLAPI\_StepByStep* contient une classe java *PIZZA* qui contient les constantes : nom de l'ontologie, IRI, espace de noms.

Rappel : La philosophie de *OWL – API* : une ontologie (ou une OWL ontologie) est une interface de type *OWLontology* qui modélise des axiomes (*OWLAxioms*). Une ontologie a un *IRI* et des méthodes qui lui sont associées. Dans la suite *ontology* désigne une variable de type *OWLontology*. Lire attentivement les tutoriels avant de commencer à écrire les méthodes suivantes.

### documentation

<https://github.com/protegeproject/swrlapi/wiki/SWRLRuleEngineAPI>

<https://github.com/protegeproject/swrlapi>

- Charger l'ontologie des pizzas : fichier *pizza\_1.owl*
- Tester la cohérence de l'ontologie (voir TP2)
- Ecrire une méthode *printAllAxiomWithPellet(OntoData.ontology, OntoData.manager)* qui affiche la base de connaissances
- Ecrire une méthode *sqwrlRequest(OntoData.ontology, OntoData.manager)* qui utilise des requêtes SQWRL :
  - écrire une requête SQWRL : quel est le résultat de  $(2+3)*4$  ?
  - écrire une requête SQWRL : combien de classes dans l'ontologie ?
  - écrire une requête SQWRL : combien y a-t-il d'instances de la *LSISPizza* ?
  - écrire une requête SQWRL : quelles sont les object propriétés qui ont pour domaine *LSISPizza* ?
  - écrire une requête SQWRL : quelles sont les valeurs de la Data property *hasDiameter* ?
  - écrire une requête SQWRL : quelles sont les Data property qui ont une valeur booléenne (Vrai ou Faux) ?
  - écrire une requête SQWRL : quelles sont les classes inférées qui sont des super classes de la classe *NamedPizza* ?

- créer une règle pour définir une grosse pizza : une grosse pizza est une pizza d'un diamètre  $> 30.0$ .
- écrire une requête SQWRL pour compter le nombre de grosses pizzas.
- Ecrire une fonction *main* pour tester la méthode avec successivement les requêtes les unes après les autres.