

TD3 Requêtes SPARQL

1. Enoncé

Geonames (<http://www.geonames.org/>) est un référentiel géographique défini au travers des langages du web de données, qui rend accessible la description de millions de lieux géoréférencés dans le monde (pays, villes, villages, lieux-dits, monuments, lacs, parcs, ...). Des requêtes portant sur l'ontologie terminologique geonames (http://www.geonames.org/ontology/ontology_v3.1.rdf) que vous prendrez le soin de télécharger et de rendre accessible au sein de votre projet Java, vous sont données. Geonames permet de typer des places spatialisées (nommées features **Feature**) à l'aide de classes (**Class** au sens geonames) et de codes (**Code**). Dans Geonames, les classes sont des instances de schémas de concepts SKOS (`skos:ConceptScheme`) et les codes sont des instances de concepts SKOS (`skos:Concept`).

Parmi les classes geonames, nous pouvons trouver :

- **A** pour country, state, region (Administrative Boundary Features)
- **H** pour water, bodies (Hydrographic Features)
- **P** pour city, village (Populated Place Features)
- **S** pour spot, building, farm (Spot Features) ...

Par exemple, pour la classe S, nous pouvons trouver les codes **S.UNIV** dont le terme préféré est university et **S.MNMT** pour monument (voir sur <http://www.geonames.org/export/codes.html>).

1. Q1 : Donner toutes les classes Geonames. Ajouter à cette énumération des classes, les commentaires en anglais (étiquette `rdfs:comment`) les décrivant.
2. Q2 : Donner pour tous les codes venant typer les features (classe Code), leur définition, leur label préféré en anglais et leur classe d'appartenance (propriétés skos).
3. Q3 : Lister pour la classe geonames V (Vegetation), les types de feature associés (code) et leur définition.
4. Q4 : Renvoyer pour chaque classe geonames, le nombre de types de feature associés
5. Q5 : Renvoyer les codes et définitions de type de feature ayant "ice" dans leur définition

Vous exprimerez ces requêtes dans le langage SPARQL, en vous aidant pour ce faire du moteur de requêtes SPARQL de Jena nommé ARQ. Vous pouvez soit définir une classe Java par requête construite, soit factoriser les aspects génériques du requêtage dans une classe de service et ainsi préférer exprimer chaque requête sous forme de méthode.