haute école ingénierie
neuchâtel berne jura www.he-arc.ch

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Mini-Projet

Introduction au Web des données

Julien Tscherrig

Table of Contents

1 Introduction

1.1 Objectifs

Le mini-projet a pour objectif les points ci-dessous :

- Mise en pratique dans un contexte libre des méthodologie, techniques et outils vus durant le cours autour d'une thématique (soumise à validation)
- Entrainement pour l'examen du 02.06.2020

1.2 Réalisation

Les séances suivantes seront consacrées à la réalisation du mini-projet :

- 12.05.2020 *(20h-21h30)*
- 19.05.2020 *(20h-21h30)*
- 26.05.2020 *(20h-21h30)*

Lors de la séance du 19.05.2020 (20h-21h30), votre présence est obligatoire. Je passerai un moment avec chacun d'entre vous pour faire un pointage sur l'avancement du mini-projet.

Je reste à la disposition de chacun durant les horaires du cours habituels et par email julien.tscherrig@hefr.ch ou sur Microsoft Teams le reste du temps.

1.3 Rendu

Deux rendus sont attendus pour ce cours

- Rapport
- Code

1.3.1 Rapport

En plus des exercices, le contenu attendu du rapport et le suivant :

- Page de titre (au minimum : nom du cours, nom, prénom, date du rendu, lien GitHub)
- Table des matières
- Conclusion sur le Mini-Projet
- Conclusion sur le cours

Le document doit être rendu au format PDF

1.3.2 Code

Tout le code est à rendre sur GitHub (uniquement les fichier java et owl)

1.3.3 Date du rendu

Le travail est à rendre au plus tard par email le lundi 01.06.2020 à 18h

Objet du mail : CAS DAR / MAS-RAD 4 | WEB – Rendu Mini-Projet

1.4 Évaluation

L'évaluation de votre travail tiendra compte des mesures suivantes : *Respect des consignes, créativité, qualité du rendu et justesse des informations, qualité du code et de ses commentaires.*

Ce travail compte à hauteur de 30% de la note du cours

2 Conception et manipulation des données sémantique

Cette première partie concerne la modélisation et la manipulation d'une ontologie et de son contenu.

2.1 Conception de votre Ontologie

Pour ce travail vous êtes libre dans le choix de la thématique que vous souhaitez modéliser avec une ontologie que vous souhaitez modéliser (soumise à validation). La première étape consiste à choisir votre thématique.

2.1.1 RDF Schema (structure)

Partie 1 - Rapport

Présenter succinctement le contexte de votre ontologie.

Représentez, sous la forme **d'un graphe** (visuel) et selon la syntaxe vue dans la théorie, une ontologie répondant aux critères suivants.

Autour de thématique retenue, proposez une ontologie contenant au minimum :

- 8 Classes (Classes), chacune des 8 classes doit contenir au minimum 2 Data Properties (Propriété donnée)
- 8 Object Properties (Propriétés objet)

En plus de cela, ajouter :

- 2 Sub-class (relation Sous-classe)
- 1 Sub-property (relation sous-propriété)

Décrivez brièvement à l'aide d'une phrase chaque élément présent dans votre ontologie (classe, propriété, etc.). Pour chaque Object Propety indiquez uniquement sous forme textuelle le RDFS:domain et RDFS:Range.

Partie 2- Rapport

Proposez un concept d'inférence intéressant pour votre ontologie. Expliquez clairement où ce concept d'inférence serait utilisé dans votre ontologie puis expliquez quel serait son impact sur les données présentes dans l'ontologie.

Remarque : n'hésitez pas à mettre plus d'éléments si nécessaire (classe, propriété, etc.).

Attention : l'ontologie que vous proposez doit être cohérente et consistante avec la thématique que vous choisissez.

2.1.2 RDF (données)

Ajouter des données dans votre Ontologie.

Partie 3 - Rapport

En vous basant sur le schéma réalisé dans la *partie 1*, pour chaque classe *(non abstraite)* représentez, sous la forme **d'un graphe** (visuel) et selon la syntaxe vue dans la théorie, 2 individus avec ses propriétés data et objet.

Faites également **apparaitre *(uniquement)* les classes** réalisées dans la *partie 1* et les relations RDFS de « type ».

2.1.3 Modélisation avec Protégé

Partie 4 - Code

Réalisez avec Protégé l'Ontologie *(Partie 1)* et les données *(Partie 2)*.

Générez ensuite le OWL et nommez-le : ontology.owl

Rendu : Le ontology.owl est à rendre sur GITHub

2.2 Implémentation de votre « Ontologie »

Cette partie consiste à implémenter l'ontologie modélisée au point précédent (RDFS + RDFS) à l'aide de la libraire RDF4J utilisée lors du TP2. Vous trouverez au point 2.4, un pseudo-code présentant le résultat attendu.

Partie 5 - Code

Réaliser les points ci-dessous à l'aide de la libraire RDF4J.

À l'aide des outils vus dans la théorie, implémentez votre ontologie en JAVA :

```
static void buildOntology(Repository rep)
```

Cette méthode doit exister dans votre code en tant que tel et doit permettre la création de votre structure complète (RDFS)

Implémentez les individus (instances) :

```
static void createIndividuals(Repository rep)
```

Cette méthode doit exister dans votre code en tant que tel et doit permettre la création de tous les individus (RDF). Elle devra contenir uniquement des méthodes similaires à celle présente ci-dessous.

Chaque type d'individu (instance) doit être réalisé dans une méthode distincte :

```
static void createIndividualsPerson(Repository rep, String name, IRI
city, ...)
```

Ci-dessus, un exemple d'entête de méthode pour la création d'un élément de type ***Person***

2.3 Récupération de données contenues dans votre Ontologie

Maintenant que vous disposez d'une structure de données (RDFS) et de données (RDF), cette partie s'intéresse à la récupération de vos données dans votre structure.

2.3.1 Conception des requêtes

Partie 6 - Rapport

Proposez 6 requêtes en SPARQL permettant de récupérer du contenu dans votre ontologie. Pour chaque requête que vous proposez, réalisez les points ci-dessous :

1) Expliquer brièvement votre requête SPARQL (ce qu'elle est censée faire, son utilité, etc.)

2) Expliquer la façon dont vous la modélisez (construction de votre requête)

3) Dessinez son graphe selon la façon vue en cours

4) Indiquez votre requête en langage SPARQL ainsi que le résultat obtenu avec vos données

Contraintes :

- Toutes les requêtes doivent utiliser des PREFIX
- Toutes les requêtes doivent contenir au minimum 2 STATEMENTS dans la clause WHERE
- Au minimum 1 requête doit posséder un FILTER
- Au minimum 1 requête doit posséder un OPTIONAL
- Au minimum 1 requête doit posséder un UNION
- Au minimum 1 requête doit posséder un MODIFIER

2.3.2 Implémentation des requêtes

Partie 7 - Code

Implémentez dans votre code chacune de ces requêtes proposées dans la partie 5.

Contraintes :

Chaque requête SPARQL doit être implémentée dans une méthode distincte prenant en paramètre l'objet Repository. La méthode doit contenir la requête SPARQL ainsi que le code permettant d'afficher le nom des colonnes retournées et son résultat dans la console.

```
static void execQueryGetPeople(Repository rep)
```

Ci-dessus, un exemple d'entête de méthode pour l'appel de la requête nommée ***GetPeople***

2.4 Code final

Proposition de pseudo-code final attendu. Vous êtes libres d'ajouter des méthodes ou classes.

```
static void buildOntology(Repository rep) {
  // TODO
}

static void createIndividualsPerson(Repository rep, String name, IRI
city, ...) {
  // TODO
}

static void createIndividuals(Repository rep) {
  // Examples
  createIndividualsPerson(...);
  createIndividualsPerson(...);
}

static void execQueryGetPeople(Repository rep) {
  // TODO
}

static public static void main(String[] args) {
  // TODO
  buildOntology(rep);
  createIndividuals(rep);

  execQueryGetPeople(rep);
  // TODO
}
```

3 Analyse et compréhension RDFa

Cette dernière partie consiste à analyser le code source, en particulier l'utilisation des balises RDFa, présent dans des pages indexées par le moteur de recherche Google.

Partie 8 - Rapport

Recherche **3 sites** contenant des balises RDFa (ne possédant pas le même schéma). Pour chaque site sélectionné :

- Présentez ces sites et expliquez l'implication du RDFa dans la page
- Quel est le graphe RDF qui se cache derrière (selon le site https://schema.org)
- Dessinez le graphe représenté par ce RDFa (vous pouvez utiliser des outils online)
- Comment Google indexe-t-il ces pages ? Comment Google les présente-t-il à l'utilisateur ?