

HEG-796-22-030

TP

Jan Krause-Bilvin

2022-05-02

Exercice 0 : prise en main

Télécharger et extraire zip de l'exercice (par exemple sur le bureau). Ouvrir un terminal anaconda prompt et se rendre dans le répertoire dézippé:

```
cd Desktop\SHACL\
```

Vérifier que le fichier *dossier.ttl* est conforme aux contraintes *shacl.ttl*:

```
python shacl.py dossier.ttl shacl.ttl
```

A rendre: catprue d'écran de la sortie de la commande tp3ex0.jpg

Exercice 1 : ajouter des contraintes sur le titre des dossiers

Copier le fichier *shacl.ttl* dans le fichier *tp3ex2_shacl.ttl*.

Compléter la section de “archvd:DossierShape” du fichier *tp3ex1_shacl.ttl* en y ajoutant la contrainte suivante:

```
sh:property [  
  sh:path rico:title;  
  rdfs:comment "Exacly one title (Name)" ;  
  sh:minCount 1 ;  
  sh:maxCount 1 ;  
] ;
```

Puis compléter cette contrainte *sh:property*, de façon à ce que le titre:

- soit obligatoirement de type *xsd:string*
 - et soit limité à 100 caractères au maximum.
-

En créant des variantes de *dossier.ttl* vérifier que:

- en l'état *dossier.ttl* est valide ;
- en effaçant (temporairement) le titre (*rico:title*) vérifier que si *dossier.ttl* ne contient plus de titre il n'est pas valide ;
- en dupliquant le titre, vérifier que si *dossier.ttl* contient deux titres, il n'est pas valide non plus;
- en allongeant le titre que ci celui-ci fait plus de 100 caractères, *dossier.ttl* n'est pas valide .

A rendre: *tp3ex1_shacl.ttl*

Exercice 2 : les dossiers doivent avoir des dates de début et de fin valides

Copier le fichier *shacl.ttl* dans le fichier *tp3ex2_shacl.ttl*.

Vérifier que le fichier *dossier.ttl* est bien valide:

```
python shacl.py dossier.ttl tp3ex2_shacl.ttl
```

Dans le fichier *tp3ex2_shacl.ttl*, ajouter une contrainte *sh:property* à de façon à s'assurer que les dossiers dans *dossier.ttl* possèdent au moins un champ *rico:hasBeginningDate*.

En pratique, vous pouvez vous inspirer de la contrainte existante sur le champ *rico:hasEndDate*.

Vérifier que,

- si le champ *rico:hasBeginningDate* est supprimé dans *dossier.ttl*, celui-ci n'est pas valide.
- si le champ *rico:hasEndDate* est supprimé dans *dossier.ttl*, celui-ci n'est pas valide.

A rendre: *tp3ex2_shacl.ttl*

Exercice 3 : validation des documents

Copier le fichier *shacl.ttl* dans le fichier *tp3ex3_shacl.ttl*.

Dans le fichier *tp3ex3_shacl.ttl*, ajouter des règles de validation *archvd:DocumentShape* en vous inspirant de *archvd:DossierShape*.

Ces règles doivent cibler les documents appartenant à la classe *archvd:Document*, qui est utilisée pour qualifier les documents dans *document.ttl*. c'est-à-dire être de la forme:

```
archvd:DocumentShape
  a sh:NodeShape ;
  sh:targetClass archvd:Document ;

  sh:property [
    sh:path rico:title;
    !!! A COMPLETER ICI !!!
  ];

  sh:property [
    sh:path rico:scopeAndContent ;
    !!! A COMPLETER ICI !!!
  ].
```

Compéter les *sh:property* de façon à s'assurer que les documents ont bien:

- exactement un titre (*rico:title*)
- que le titre est une chaîne de caractères (utiliser: *xsd:string*)
- que le titre fait au maximum 200 caractères (utiliser: *sh:maxLength*)
- au maximum une description (*rico:scopeAndContent*)
- exactement une date de fin (*rico:hasEndDate*)

Vérifier que le fichier *document.ttl* est bien valide:

```
python shacl.py dossier.ttl tp3ex3_shacl.ttl
```

A rendre: *tp3ex3_shacl.ttl*

Exercice 4 : validation du numéro AVS des agents

Copier le fichier *shacl.ttl* dans le fichier *tp3ex4_shacl.ttl*.

Dans le fichier *tp3ex4_shacl.ttl*, ajouter des règles de validation *archvd:AgentPersonShape* en vous inspirant de *archvd:DossierShape*.

Ces règles doivent cibler les agents appartenant à la classe *archvd:AgentPerson*, qui est utilisée pour qualifier les documents dans *document.ttl*. c'est-à-dire être de la forme:

```
archvd:AgentPersonShape
  a sh:NodeShape ;
  sh:targetClass archvd:AgentPerson ;

  sh:property [
    sh:path rico:identifiant ;
    rdfs:comment "Exacly one identifier based on N-AVS-13." ;
    !!! A COMPLETER ICI !!!
  ] .
```

Compéter le *sh:property* de façon à s'assurer que les agents ont bien:

- un et un seul identifiant (propriété “rico:identifiant”)
 - que cette propriété est de type string (xsd:string)
 - que ce format est conforme avec l'expression régulière “`^{\d{3}}\.\d{4}\.\d{4}\.\d{2}$`” (en utilisant “sh:pattern”).
-

Vérifier que le fichier *document.ttl* est bien valide:

```
python shacl.py document.ttl tp3ex4_shacl.ttl
```

Vérifier encore que si le numéro d'AVS 13 n'a pas une structure conforme, *document.ttl* n'est pas valide.

A rendre: *tp3ex4_shacl.ttl*
