

# TP STTL

## Introduction

Les ressources logicielles se trouvent ici : <http://wimmics.inria.fr/lectures>

Vous y trouverez le serveur corese et un navigateur sttl.

L'exercice consiste à concevoir un navigateur hypertexte sur un graphe RDF. Vous disposez du serveur corese et d'une archive qui contient une version initiale de navigateur qu'il s'agit d'étudier et ... d'améliorer !

Le navigateur est conçu sous forme d'une transformation STTL, embarquée dans un serveur Web, qui engendre dynamiquement une page HTML en exploitant les données du graphe RDF. La transformation affiche soit le contenu du graphe en entier soit une ressource particulière (un URI et ses propriétés).

L'archive demo.tar.gz contient les répertoires suivants :

1. sttl : la transformation initiale
2. format : les canevas des pages HTML à engendrer avec STTL
3. data : les données RDF et un fichier profile.ttl qui définit le service de navigation hypertexte

## Marche à suivre

1. Extraire l'archive dans votre espace de travail, et appelons /path/demo le chemin absolu dans le file system.
2. **Sous Windows, éditez et sauvegardez les fichiers en UTF-8**
3. Editer le fichier /path/demo/data/profile.ttl et remplacer la valeur de st:transform “...” par la valeur “/path/demo/sttl” dans votre file system.  
Sous windows, la syntaxe est : “C:\\path\\demo\\sttl”
4. Editer le fichier /path/demo/sttl/profile.rq et remplacer le corps de la fonction us:format par la valeur de “/path/demo/format/” dans votre file system.  
Sous windows, la syntaxe est : “C:\\path\\demo\\format\\”
5. Lancer le serveur avec la commande :  
java -jar corese-server-3.2.2.jar -pp file:///path/demo/data/profile.ttl  
Sous windows :  
java "-Dfile.encoding=UTF-8" -jar corese-server-3.2.2.jar -pp file:///C:/path/demo/data/profile.ttl
6. Lancer le navigateur hypertexte en saisissant l'URL suivante dans votre navigateur :  
<http://localhost:8080>

puis cliquer dans le menu Misc/Demo

## Exercice

Lisez la transformation `demo/sttl` en commençant par le fichier `start.rq`, lisez également les formats HTML dans le répertoire `demo/format`.

Le TP consiste à adapter et améliorer la transformation STTL et les formats HTML associés.

Ajouter des templates dédiés à des classes de ressources (`h:Woman`, `h:Researcher`, `rdfs:Class`, etc.) en vous inspirant du template `match-man.rq`

Vous pouvez par exemple associer des icones aux classes et pour chaque ressource afficher l'icone de sa classe. Les icones doivent être identifiées par un URL HTTP.

Vous pouvez engendrer un lien hypertexte vers une ressource de la base avec une balise HTML "a" ayant comme valeur d'attribut "href" le résultat de la fonction `st:pplink(?uri)`.

Associer des lieux de DBpedia à certaines ressources et afficher le lieu lors de l'affichage de la ressource en vous inspirant du fichier `data/place.ttl` et du template `match-man.rq` avec l'énoncé :

- `st:call-template(st:locator, ?place)`

Afficher certains types de ressources avec une autre transformation à l'aide des énoncés suivants :

- Format Turtle : `st:apply-templates-with(st:hturtle, ?in)`
- Format RDF/XML : `st:apply-templates-with(st:hrdfxml, ?in)`

Si vous avez un jeu de données RDF préféré, vous pouvez aussi modifier le fichier `demo/data/profile.ttl` pour charger et traiter ces données.