

# TP N 1 Web sémantique :

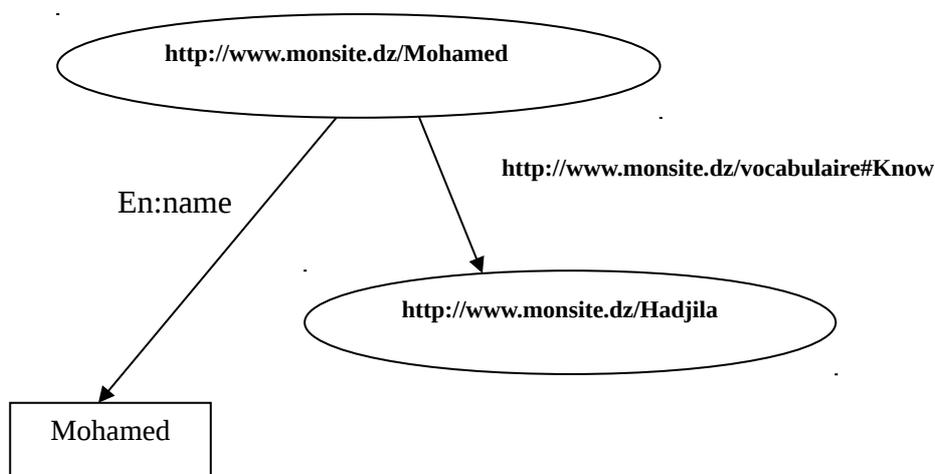
## l'utilisation de la bibliothèque Jena

### Partie I: création d'un fichier RDF

Jena est une API créée par « [The Apache Software Foundation \(ASF\)](http://incubator.apache.org/jena/) », qui permet de manipuler (lecture, écriture et traitement) des fichiers RDF, ainsi que des ontologies RDFs et OWL. Elle fournit aussi une implémentation d'un moteur SPARQL pour faire des requêtes sur des données RDF.

#### I. 1 Exemple 1

Le code suivant présente un code qui permet de créer un fichier RDF/XML du graphe suivant :



1. Créez un nouveau projet : java desktop application.
2. Ajoutez la bibliothèque Jena à votre projet.
3. Créez la classe suivante à votre projet.
4. Instanciez cette classe et appelez la méthode `creat()` dans la méthode « main » de votre projet.

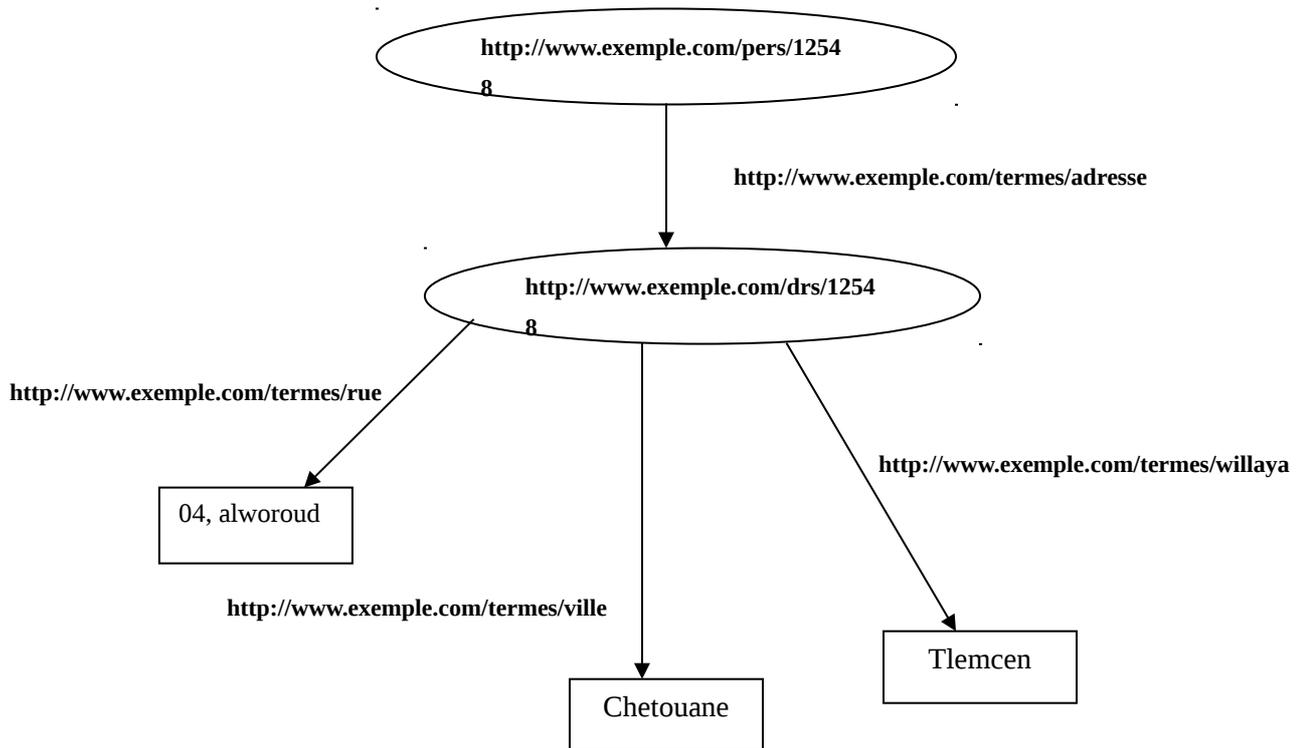
```

public class createRDF {
    public void creat() {
        try{
            String MohURI="http://www.monsite.dz/Mohamed";
            String HadjURI="http://www.monsite.dz/Hadjila";
            String NameMoh="Mohamed";
            //la création d'espace de nom
            String EspaceNom ="http://www.monsite.dz/vocabulaire#";
            //la création d'un modèle
            Model model=ModelFactory.createDefaultModel();
            // association de l'URI EspaceNom au prifixe en
            model.setNsPrefix("en",EspaceNom);
            //La creation d'une resource
            Resource personne1= model.createResource(MohURI);
            // creation d'une propriété (name)
            Property name = model.createProperty(EspaceNom,"name");
            personne1.addProperty(name, NameMoh);
            Resource personne2= model.createResource(HadjURI);
            Property connait = model.createProperty(EspaceNom,"know");
            personne1.addProperty(connait,personne2);
            //sauvgarde sous forme d'un fichier xml
            PrintWriter w = new PrintWriter(new File("C:\\file.rdf"));
            model.write(w,"RDF/XML"); //choix du format "XML"
            model.write(System.out);
        }
        catch(Exception e){
            System.out.println("le fichier n'existe pas");
        }
    }
}
}

```

## I.2 Travail à faire :

Écrivez le programme qui génère le fichier RDF/XML correspondant au graphe RDF suivant :



## Partie II : Lecture d'un fichier RDF

Le code suivant présente un programme qui permet de lire un fichier RDF/XML :

5. Créez un nouveau projet : java application.
6. Ajoutez la bibliothèque Jena à votre projet.
7. Créez la classe suivante dans votre projet.

Instanciez cette classe et appelez la méthode `readRdf('file.rdf')` dans la méthode « main » de votre projet.

### Travail à faire :

- 1- Ecrire le programme qui lit le fichier RDF/XML «exo2.rdf».
- 2- quels sont les sujets, les prédicats et les objets de ce fichier.
- 3- donnez une représentation graphique de ce fichier.

```

public class Rdfreader {
    public void readRdf(String file){
        Model model= ModelFactory.createDefaultModel();
        model.read(file,"RDF/XML");
        StmtIterator iter= model.listStatements();
        while (iter.hasNext()){
            Statement Stmt=iter.nextStatement();
            Resource subject=Stmt.getSubject();
            RDFNode Object = Stmt.getObject();
            String predicate= Stmt.getPredicate().getLocalName();
            System.out.println("+++++++ déclaration ++++++");
            System.out.println("le sujet est: "+subject);
            System.out.println("le prédicat est: "+predicate+" ");
            if (Object instanceof Resource)
                System.out.println("le'objet est: "+Object.toString());
            else
                System.out.println("le'objet est: "+" \'" +
Object.toString() + "\'");}
    }}

```