jena构建知识图谱

maven依赖

```
<dependency>
  <groupId>org.apache.jena</groupId>
  <artifactId>apache-jena-libs</artifactId>
  <type>pom</type>
  <version>3.7.0</version>
</dependency>
<dependency>
  <groupId>org.apache.jena</groupId>
  <artifactId>jena-sdb</artifactId>
  <version>3.7.0</version>
</dependency>
<dependency>
  <groupId>org.apache.jena</groupId>
  <artifactId>jena-base</artifactId>
  <version>3.7.0</version>
</dependency>
<dependency>
  <groupId>org.apache.jena</groupId>
  <artifactId>jena-fuseki-embedded</artifactId>
  <version>3.7.0</version> <!-- Set the version -->
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.jena/jena-arq -->
<dependency>
  <groupId>org.apache.jena</groupId>
  <artifactId>jena-arq</artifactId>
  <version>3.7.0</version>
</dependency>
```

jena RDF API

三元组 -> Statement

- subject：实体
- predicate：属性
- object：值

创建Model

```
// URI 定义
static String personURI    = "http://somewhere/JohnSmith";
static String fullName     = "John Smith";
static String familyName = "Smith";
static String givenName = "John";

public static Model model;

// 创建一个空模型（KG）
model = ModelFactory.createDefaultModel();

// 创建一个resource（一个subject）
Resource johnSmith = model.createResource(personURI);

//链式API,为resource添加多个Property
johnSmith.addProperty(VCARD.FN, fullName)
  .addProperty(VCARD.N, model.createResource()
               .addProperty(VCARD.Given, givenName)
               .addProperty(VCARD.Family, familyName));

// 添加属性，这里的value是一个literals（文本）
johnSmith.addProperty(VCARD.FN, fullName);
```

遍历Model

```
// list the statements in the Model
StmtIterator iter = model.listStatements();

// print out the predicate, subject and object of each statement
while (iter.hasNext()) {
  Statement stmt      = iter.nextStatement();  // get next statement
  Resource  subject   = stmt.getSubject();     // get the subject
  Property  predicate = stmt.getPredicate();   // get the predicate
  RDFNode   object    = stmt.getObject();      // get the object

  System.out.print(subject.toString());
  System.out.print(" " + predicate.toString() + " ");
```

```
  if (object instanceof Resource) {
    System.out.print(object.toString());
  } else {
    // object is a literal
    System.out.print(" \"" + object.toString() + "\"");
  }
  System.out.println(" .");
}
```

保存为RFD文件

- xml格式

```
model.write(System.out);
```

- ttl格式

```
model.write(System.out, "TURTLE");
```

- ttl格式 add prefix功能

```
model.setNsPrefix( "vCard", "http://www.w3.org/2001/vcard-rdf/3.0#" );
model.setNsPrefix( "rdf", "http://www.w3.org/1999/02/22-rdf-syntax-ns#" );
try {
  model.write(System.out, "TURTLE");
  //            model.write(new FileOutputStream("1.rdf"),"TURTLE");
//write to a file
} catch (Exception e) {
  e.printStackTrace();
}
```

读取RDF文件

```
// create an empty model
model = ModelFactory.createDefaultModel();

InputStream in = FileManager.get().open( "1.rdf" );
if (in == null) {
  throw new IllegalArgumentException( "File: " + "1.rdf"  + " not found");
}

// read the RDF/XML file
model.read(in, "","TURTLE");

// write it to standard out
model.write(System.out);
```

解析RDF文件

- 一个resouce都有一个唯一的URI，我们可以通过URI来获取对应的Resouce
- 获取到Resouce后，通过getRequiredProperty获取属性，如果一个属性包含多个值，可以使用listProperties获取

```
// retrieve the Adam Smith vcard resource from the model
Resource vcard = model.getResource(personURI);

// retrieve the value of the N property
Resource name = (Resource) vcard.getRequiredProperty(VCARD.N)
  .getObject();
// retrieve the given name property
String fullName = vcard.getRequiredProperty(VCARD.FN)
  .getString();
// add two nick name properties to vcard
vcard.addProperty(VCARD.NICKNAME, "Smithy")
  .addProperty(VCARD.NICKNAME, "Adman");

// set up the output
System.out.println("The nicknames of \"" + fullName + "\" are:");
// list the nicknames
StmtIterator iter = vcard.listProperties(VCARD.NICKNAME);
while (iter.hasNext()) {
  System.out.println("    " + iter.nextStatement().getObject()
                     .toString());
}

try {
  model.write(System.out);
} catch (Exception e) {
  e.printStackTrace();
}
```

查询Model

- 查询Property: listResourcesWithProperty

```
ResIterator iter = model.listResourcesWithProperty(VCARD.FN);
if (iter.hasNext()) {
  System.out.println("The database contains vcards for:");
  while (iter.hasNext()) {
    System.out.println("  " + iter.nextResource()
                       .getRequiredProperty(VCARD.FN)
                       .getString() );
  }
} else {
  System.out.println("No vcards were found in the database");
}
```

- 查询Statement: listStatements(SimpleSelector)

```
// select all the resources with a VCARD.FN property
// whose value ends with "Smith"
StmtIterator iter = model.listStatements(
  new
  SimpleSelector(null, VCARD.FN, (RDFNode) null) {
    @Override
    public boolean selects(Statement s) {
      return s.getString().endsWith("Smith");
    }
  });
if (iter.hasNext()) {
  System.out.println("The database contains vcards for:");
  while (iter.hasNext()) {
    System.out.println("  " + iter.nextStatement()
                       .getString());
  }
} else {
  System.out.println("No Smith's were found in the database");
}
```