Distributed Programming II

A.Y. 2017/18

Assignment n. 2 – Neo4JSimpleXML REST API Documentation

Neo4JSimpleXML is a REST web service that provides the possibility to create simple graphs made of nodes connected by relationships, and to compute simple reachability questions on a graph.

The service is based on NEO4J (a graph-oriented DB): it stores the created graphs on NEO4J. Each node can have properties (i.e. name-value pairs) and labels (i.e. strings), while each relationship has a type (which is a string). The service provides the possibility to query for the nodes that are reachable from a given source node via certain types of relationships, in the graph that has been created.

Getting Started with the Service

The service provider can be started on your own local machine (under Linux) by first starting the NEO4J database service and the Tomcat servlet container, and, then, by deploying the Neo4JSimpleXML service onto Tomcat.

The NEO4J database can be configured and started by the following procedure:

1) Let us call `[neo4j]` the NEO4J installation directory. Modify the default authentication settings of NEO4J by uncommenting part of the NEO4J configuration file `[neo4j]/conf/neo4j.conf` (i.e, `dbms.security.auth_enabled=false`) so that it will not require authentication; 2) Run the NEO4J service by issuing the following command (from the `[neo4j]/bin/` directory):

```
$ ./neo4j console
```

Note that steps 1)-3) are not necessary at Labinf, where the software is already available under /opt.

An alternative, easier way, to start NEO4J is to run the ant target `start-neo4j` or `clear-and-restart-neo4j` (which first clears the NEO4J DB and then starts the server).

Tomcat 8 can be configured by the following procedure:

1) Load the JAX-RS jar files and their dependencies into the folder `/opt/dp2/shared/lib` (these files can be found in the same location at Labinf or can be downloaded from the course web site, in the "information about the assignments" page); 2) Let `[catalina_home]` be the Tomcat 8 installation directory; edit the `[catalina_home]/conf/catalina.properties` configuration file and set the `shared.loader` property as follows

```
shared.loader=/opt/dp2/shared/lib/*.jar
```

2) Edit the `[catalina_home]/conf/tomcat-users.xml` file and add the following lines of code under the `tomcat-user` tag:

```
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="root" password="root" roles="tomcat,manager-gui"/>
<user username="both" password="tomcat" roles="tomcat,role1"/>
<user username="role1" password="tomcat" roles="role1"/>
```

Note that this configuration is already done on the Labinf machines.

Tomcat 8 can be started by issuing the following command (from the `[catalina_home]/bin` folder):

```
$ ./startup.sh
```

A similar script (`shutdown.sh`) is available for stopping Tomcat.

The *Neo4JSimpleXML* service can be deployed into the Tomcat 8 servlet container and started simply by copying the `Neo4JSimpleXML.war` file (that can be found in `[root]/war`) to `[catalina_home]/webapps`.

You can also deploy/un-deploy/start/stop web-services through the Tomcat web interface at http://localhost:8080/manager/html, which requires authentication with username and password (with the above configuration you can use "root" "root")

One last possibility to start/stop tomcat and deploy/un-deploy/start/stop the service is to use the targets in the ant script provided with the zip file of the Assignment. Note that these scripts also require the aforementioned configuration of Tomcat unless you are at Labinf.

Service Description and Organization

Neo4JSimpleXML offers a subset of the REST API provided by NEO4J itself, but it exchanges data in XML format rather than in JSON format. In particular, the web service follows the same organization and naming of resources of the NEO4J REST API (which is documented in https://neo4j.com/docs/rest-docs/current). The base URL of the service, which in NEO4J is http://localhost:7474/db/data, here is http://localhost:8080/Neo4JSimpleXML/webapi/data. Apart from this difference, the resources available in Neo4JSimpleXML are also available with the same meaning in NEO4J, with the only exception of the resource for getting reachable nodes, which is different. So, for most of the resources, you can get additional information by looking at the corresponding resources in the NEO4J REST API.

The resources available in Neo4JSimpleXML are documented by means of Swagger (https://swagger.io/). After having deployed the service, the Swagger documentation will be available as a web application (the link can be found in the index page of the service available at http://localhost:8080/Neo4JSimpleXML).

Neo4JSimpleXML also provides an XML schema for the data types it uses, and a WADL description of the service itself. They are available, respectively, at http://localhost:8080/Neo4JSimpleXML/Neo4JSimpleXML.xsd and at http://localhost:8080/Neo4JSimpleXML/webapi/application.wadl.

The current status of the underlying NEO4J DB can be monitored by means of the NEO4J web application available at http://localhost:7474/browser.

The rest of this document summarizes the resources available in the service and gives examples of how to use them.

PART1: RESOURCES

Neo4JXML REST API includes the following resources:

localhost:8080/Neo4JSimpleXML/webapi/application.wadl	WADL description of the service
localhost:8080/Neo4JSimpleXML/webapi/data/	the base URL of the service
localhost:8080/Neo4JSimpleXML/webapi/data/node	the set of all nodes
localhost:8080/Neo4JSimpleXML/webapi/data/node/id	the node identified by id
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/labels	the set of labels of node id
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/labels/lab	the label of node id named lab
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/properties	the set of properties of node id
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/properties/pro	the property of node id named pro
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/relationships	the relationships of node id
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/relationships/all	all relationships of node id
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/relationships/in	the incoming relationships of node id
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/relationships/out	the outgoing relationships of node id
localhost:8080/Neo4JSimpleXML/webapi/data/relationship	the set of all relationships
localhost:8080/Neo4JSimpleXML/webapi/data/relationship/id	the relationship identified by id
localhost:8080/Neo4JSimpleXML/webapi/data/node/id/reachableNodes	the nodes reachable from node id

PART2: OPERATIONS WITH EXAMPLES

Create Node with properties

POST localhost:8080/Neo4JSimpleXML/webapi/data/node/

***Content-Type*:** application/xml

***Responses**: 2*01 (Created), 400(Bad request), 500(Internal Server Error)

Example request

POST localhost:8080/Neo4JSimpleXML/webapi/data/node/

BODY:

```
<node xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
    <properties>
      <property name="name" value="Client0"/>
    </properties>
</node>
```

Example response

201: Created

BODY:

```
<node xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML"id="0">
    <properties>
      <property name="name" value="Client0"/>
    </properties>
</node>
```

Get Node by ID

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}

***Content-Type*:** application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/0

Example response

200: OK

BODY:

```
<node xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML"id="0">
    <properties>
      <property name="name" value="Client0"/>
    </properties>
</node>
```

Delete Node by ID

DELETE localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}

***Content-Type*:** application/xml

***Responses**: 2*04 (No Content), 404(Not Found), 500(Internal Server Error)

Example request

DELETE localhost:8080/Neo4JSimpleXML/webapi/data/node/0

Example response

204: NO CONTENT

Update Properties of Node

PUT localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ properties

***Content-Type*:** application/xml

***Responses**: 2*04 (No Content), 400(Bad request), 404(Not Found), 500(Internal Server Error)

Example request

PUT localhost:8080/Neo4JSimpleXML/webapi/data/node/0/ properties

BODY:

```
<properties xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
    <property name="name" value="Client0"/>
    <property name="type" value="Client"/>
</properties>
```

Example response

204: No Content

Get Properties by Node ID

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ properties

***Content-Type*:** application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/0/ properties

Example response

200: OK

BODY:

```
<properties xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
    <property name="name" value="Client0"/>
    <property name="type" value="Client"/>
</properties>
```

Delete Properties by Node ID

Delete localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ properties

***Content-Type*:** application/xml

***Responses**: 2*04 (No Content), 404(Not Found), 500(Internal Server Error)

Example request

Delete localhost:8080/Neo4JSimpleXML/webapi/data/node/0/properties

Example response

204: No Content

Update Property of Node by Name

PUT localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ properties/{pro_name}

***Content-Type*:** application/xml

***Responses**: 2*04 (No Content), 400(Bad request), 404(Not Found), 500(Internal Server Error)

Example request

PUT localhost:8080/Neo4JSimpleXML/webapi/data/node/0/ properties/type

BODY:

```
<property xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML"value="Linux_Client"/>
```

Example response

204: No Content

Get Property by of Node by Name

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ properties/{pro_name}

***Content-Type*:** application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/0/ properties/type

Example response

200: OK

BODY:

```
<property xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML"name="type"
value="Linux_Client"/>
```

Delete Property by of Node by Name

DELETE localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ properties/{pro_name}

***Content-Type*:** application/xml

***Responses**: 2*04 (No Content), 404(Not Found), 500(Internal Server Error)

Example request

DELETE localhost:8080/Neo4JSimpleXML/webapi/data/node/0/ properties/type

Example response

204: No Content

Create Label for a Node

POST localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/labels

Content-Type: application/xml

***Responses**:2*04 (No Content), 400(Bed request), 404(Not Found), 500(Internal Server Error)

Example request

POST localhost:8080/Neo4JSimpleXML/webapi/data/node/1/labels

BODY:

```
<labels xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
   <label>HP_Server</label>
</labels>
```

Example response

204: No Content

Get Labels by Node ID

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/labels

Content-Type: application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/15/labels

Example response

200: OK

BODY:

```
<labels xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
   <label>HP_Server</label>
</labels>
```

Update Labels

PUT localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/labels

Content-Type: application/xml

***Responses**: 2*04 (No Content), 400(Bed request), 404(Not Found), 500(Internal Server Error)

Example request

PUT localhost:8080/Neo4JSimpleXML/webapi/data/node/0/labels

BODY:

```
<labels xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
   <label>HP_Tower_Server</label>
</labels>
```

Example response

204: No Content

Delete Labels by ID (Name)

DELETE localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/labels/{label_name}

Content-Type: text/plain

***Responses**: 2*04 (No Content), 404(Not Found), 500(Internal Server Error)

Example request

DELETE localhost:8080/Neo4JSimpleXML/webapi/data/node/1/labels/HP_Server

Example response

204: NO CONTENT

Create Relationship connecting a Node to another Node

POST localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ relationships

Content-Type: application/xml

***Responses**:2*01 (Created), 400(Bed request), 404(Not Found), 500(Internal Server Error)

Example request

POST localhost:8080/Neo4JSimpleXML/webapi/data/node/0/ relationships

BODY:

```
<relationship xmlns=http://pad.polito.it/dp2/Neo4JSimpleXML dstNode="1"
type="Link" />
```

Example response

201: Created

BODY:

```
<relationship xmlns=http://pad.polito.it/dp2/Neo4JSimpleXML dstNode="1"
type="Link" srcNode="0" id="0"/>
```

Get Relationship by ID

GET localhost:8080/Neo4JSimpleXML/webapi/data/relationship/{relationship_id}

Content-Type: application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/relationship/0

Example response

- **200:** OK

```
<relationship xmlns=http://pad.polito.it/dp2/Neo4JSimpleXML dstNode="1"
type="Link" srcNode="0" id="0"/>
```

Delete Relationship by ID

DELETE localhost:8080/Neo4JSimpleXML/webapi/data/relationship/{relationship_id}

Content-Type: application/xml

***Responses**: 2*04 (OK), 404(Not Found), 500(Internal Server Error)

Example request

- **DELETE** localhost:8080/Neo4JSimpleXML/webapi/data/relationship/3

Example response

- **204:** NO CONTENT

Get All Relationship by Node ID

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{nodeid}/relationships/all

Content-Type: application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/0/relationships/all

Example response

- **200:** OK

```
<relationships xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
  <relationship dstNode="1" type="Link" srcNode="1" id="0"/>
  <relationship dstNode="0" type="Link" srcNode="0" id="1"/>
</relationships>
```

Get IN Relationship by Node ID

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{nodeid}/relationships/in

Content-Type: application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/0/relationships/in

Example response

- **200:** OK

```
<relationships xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
  <relationship dstNode="0" type="Link" srcNode="0" id="1"/>
</relationships>
```

Get OUT Relationship by Node ID

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{nodeid}/relationships/out

Content-Type: application/xml

***Responses**: 2*00 (OK), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/0/relationships/out

Example response

- **200:** OK

```
<relationships xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
  <relationship dstNode="1" type="Link" srcNode="1" id="0"/>
</relationships>
```

Get nodes reachable from node

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/{node_id}/ reachableNodes?[relationshipTypes={rel_id1|rel_id2}]&[nodeLabel={label }]

Content-Type: application/xml

***Responses**: 2*00 (OK), 400(Bad request), 404(Not Found), 500(Internal Server Error)

Example request

GET localhost:8080/Neo4JSimpleXML/webapi/data/node/0/reachableNodes?relationshipTypes=Link|link&nodeLabel=Server

Example response

200: OK

BODY:

```
<nodes xmlns="http://pad.polito.it/dp2/Neo4JSimpleXML">
  <node "id="1">
    <properties>
      <property name="name" value="Server0"/>
      <property name="type" value="Server"/>
    </properties>
    <labels>
      <label>HP_Server</label>
    </labels>
  </node>
</nodes>
```