Neo4j instruction

Danchen Zhang

Catalog

- Installation
- Cypher Query Language (CQL)
- Connect to Java and Python

Tutorial materials

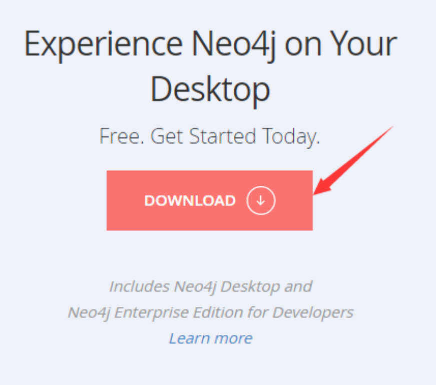
[1] https://neo4j.com/developer/get-started/

[2] https://neo4j.com/blog/neo4j-video-tutorials/

[3] https://www.tutorialspoint.com/neo4j/index.htm

[4] https://www.quackit.com/neo4j/tutorial/

Installation & Start



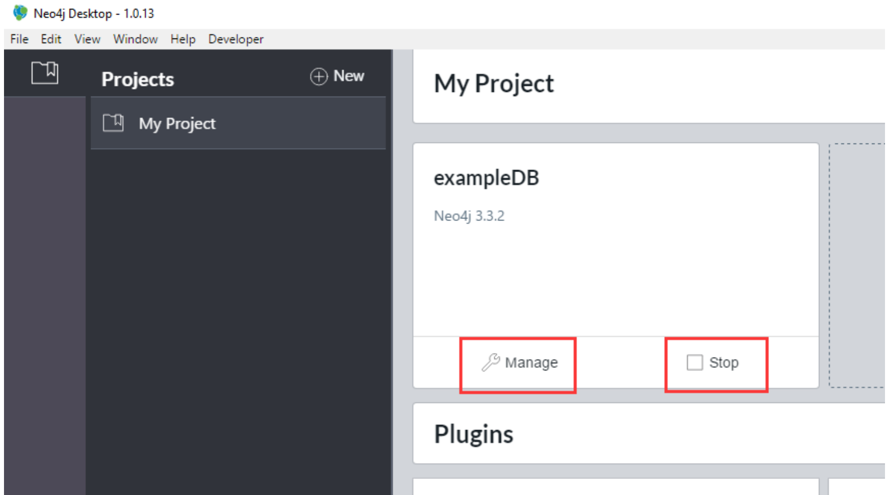
Experience Neo4j on Your Desktop

Free. Get Started Today.

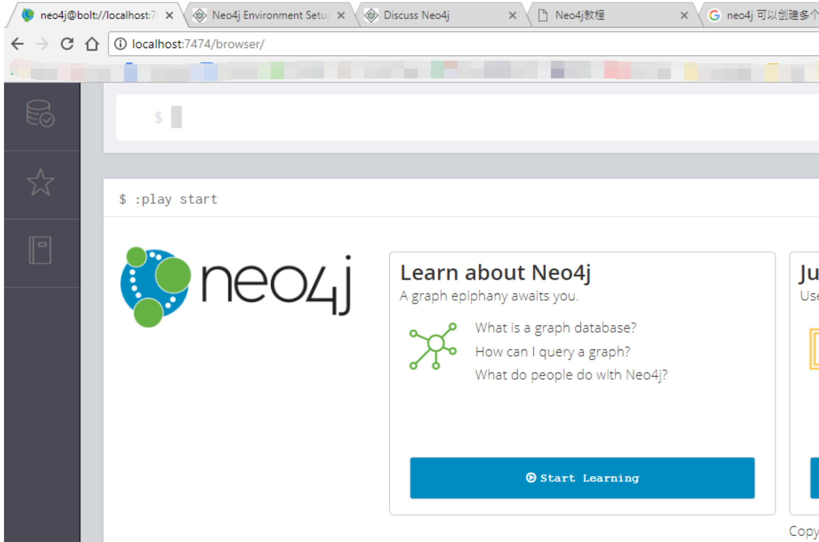
Includes Neo4j Desktop and Neo4j Enterprise Edition for Developers

- You can obtain .exe and .dmg in https://neo4j.com/download/

- Localhost:7474
- Default username: neo4j
- Default password: neo4j



Neo4j Desktop - 1.0.13



```
$ :play start
```

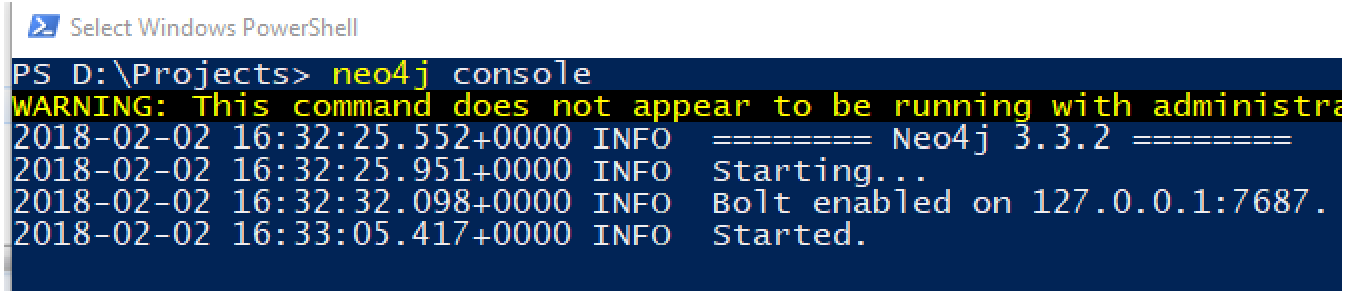
Learn about Neo4j

A graph epiphany awaits you.

What is a graph database?
How can I query a graph?
What do people do with Neo4j?

Installation & Start 2

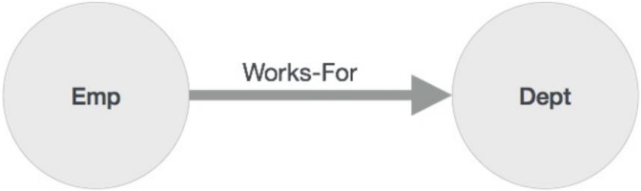
- https://www.tutorialspoint.com/neo4j/neo4j_environment_setup.htm
- Remember to setup the system path:
  - NEO4J_HOME =*******\neo4j-community-3.3.2
  - PATH = *******\neo4j-community-3.3.2\bin



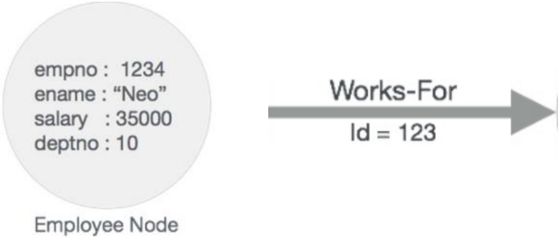
```
PS D:\Projects> neo4j console
WARNING: This command does not appear to be running with administra
2018-02-02 16:32:25.552+0000 INFO  ======== Neo4j 3.3.2 ========
2018-02-02 16:32:25.951+0000 INFO  Starting...
2018-02-02 16:32:32.098+0000 INFO  Bolt enabled on 127.0.0.1:7687.
2018-02-02 16:33:05.417+0000 INFO  Started.
```

Neo4j basic elements

- Node & Relationship
- Labels



- Properties



Neo4j Cypher Query Language (CQL)

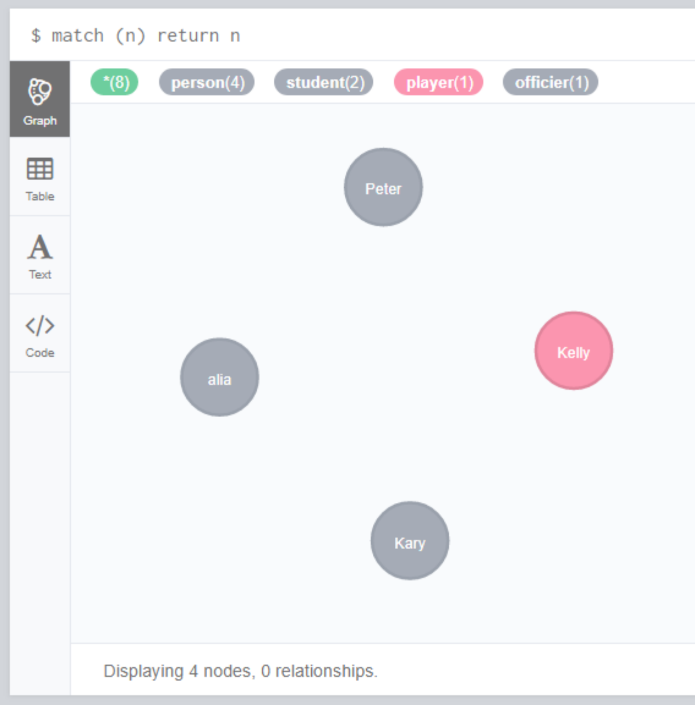
- Ref: https://www.tutorialspoint.com/neo4j/neo4j_cql_introduction.htm
- Frequent CQL operations:
    - Create (unique), delete, merge
    - match, return, where + order by, limit, skip, Count, substring
    - Set, remove
    - foreach

Set 1: create nodes & relationships

- CREATE (node_name:label {key1:value1, key2:value2,…})
- Use “match (n) return n” to check results
- CREATE CONSTRAINT ON (n:label) ASSERT n.property IS UNIQUE

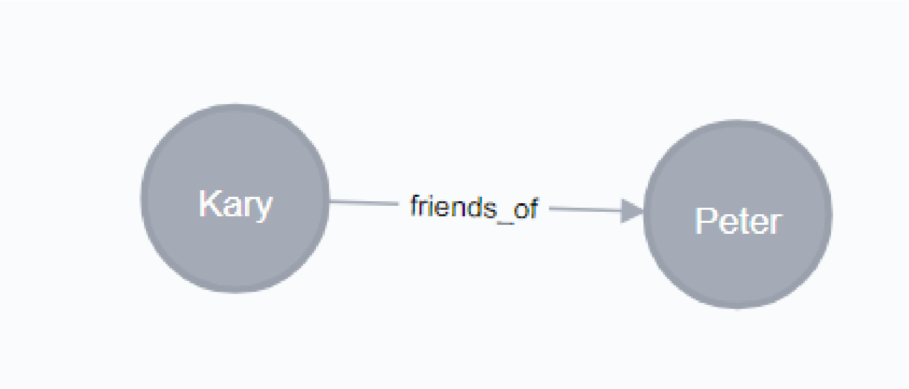
- MATCH (a:student), (b:student)
-    WHERE a.name = "Kary" AND b.name = "Peter"
- CREATE (a)-[rel:friends_of]->(b)
- RETURN a,b

- CREATE (a:person:student{name: "Kary", gender:"female", age:"25"})
- CREATE (b:person:player{name: "Kelly", gender:"female", age:"20"})
- CREATE (c:person:student{name: "Peter", gender:"male", age:"25"})
- CREATE (d:person:officier{name: "alia", gender:"female", age:"23"})



```
$ match (n) return n
```

Displaying 4 nodes, 0 relationships.

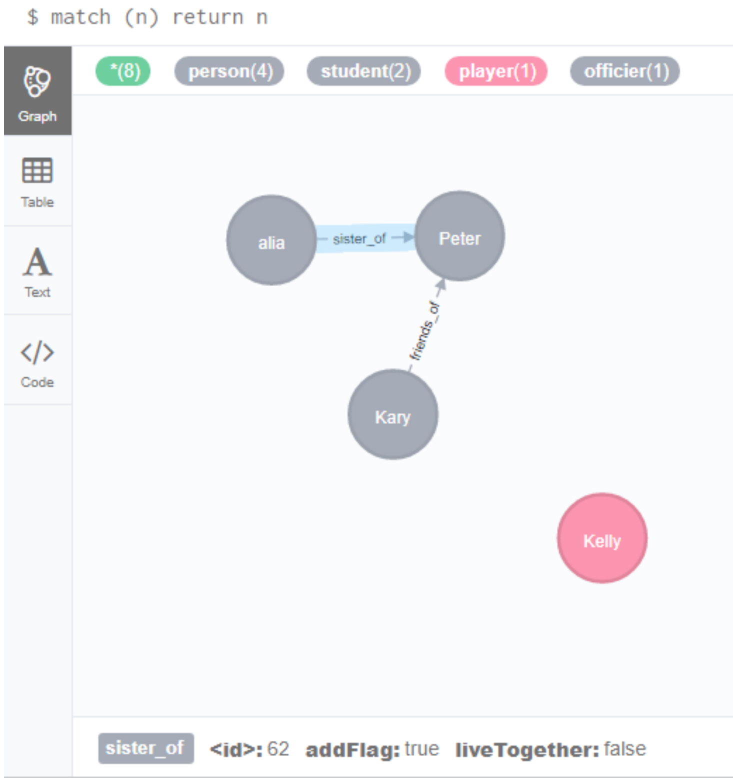


Set 1: Merge

- Match first, if the node is not found, create the node.
- On Create and On Match can help with extra data operation.
- Merge both nodes & relationships.

- MERGE (node:label) RETURN node
- MERGE (node a)-[rel:relationship]->(node b)

- MATCH (a:person), (b:person)
-   WHERE a.name = "alia" AND b.name = "Peter"
-   MERGE (a)-[relationship:sister_of{liveTogether:False}]->(b)
-   on match set relationship.addFlag=False
-   on create set relationship.addFlag=True
- RETURN a, b



```
$ match (n) return n
```

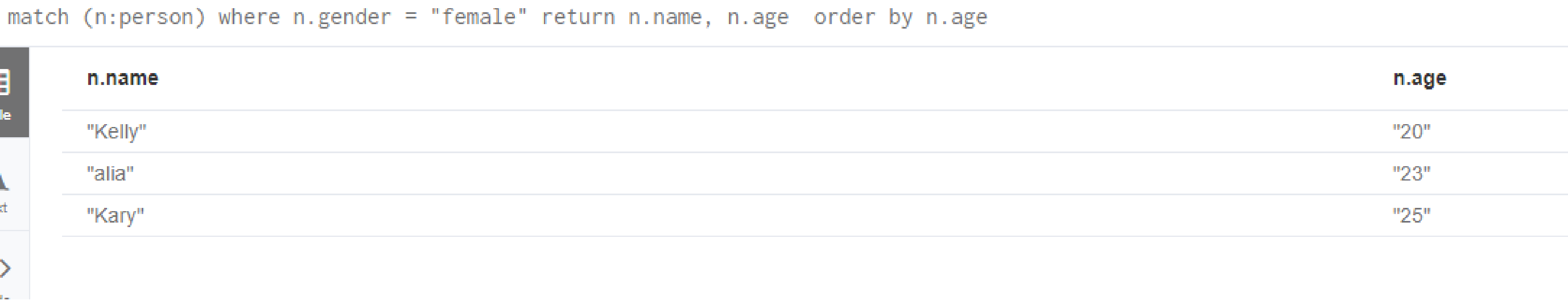
sister_of <id>: 62 addFlag: true liveTogether: false

Set 1: delete

- Delete selected nodes:
    - Merge(testNode:random)
    - MATCH (node:random)
    - DETACH DELETE node
- Delete all nodes & relationships in the database.
    - MATCH (n) DETACH DELETE n

Set 2: match where return + orderby

- match (n:person)
- where n.gender = "female"
- return n.name, n.age
- order by n.age



```
match (n:person) where n.gender = "female" return n.name, n.age  order by n.age
```

n.name	n.age
"Kelly"	"20"
"alia"	"23"
"Kary"	"25"

Set 2: limit, skip, Count, substring

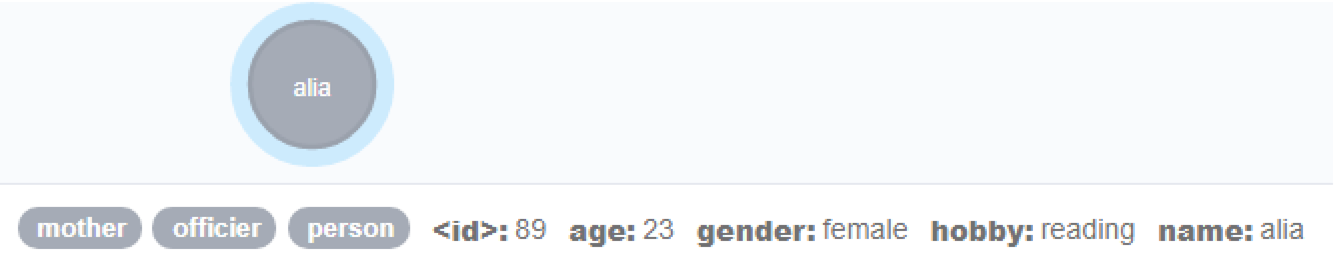
- match (n:person) return n.name, n.age, n.gender limit 2
- match (n:person) return n.name, n.age, n.gender order by n.age skip 2
- match (n:person) return n.gender, count(*)
- match (n:person) return n.name, n.age,substring(n.gender, 0, 1)

Set3: SET + REMOVE

- SET:
  - Add a/multiple property
  - Add a/multiple labels on a node
  - Remove a property = REMOVE

- match(n)
- where n.name="alia"
- set n.hobby="reading" or null
- return n

- match(n)
- where n.name="alia"
- set n:mother
- return n

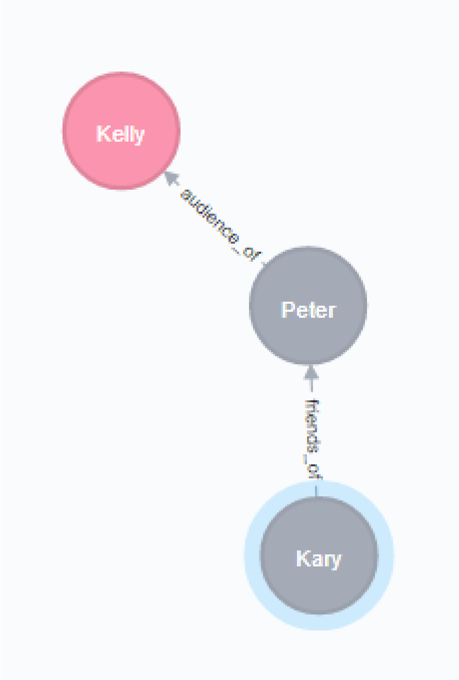


mother officier person **<id>:** 89 **age:** 23 **gender:** female **hobby:** reading **name:** alia

Set 4: foreach

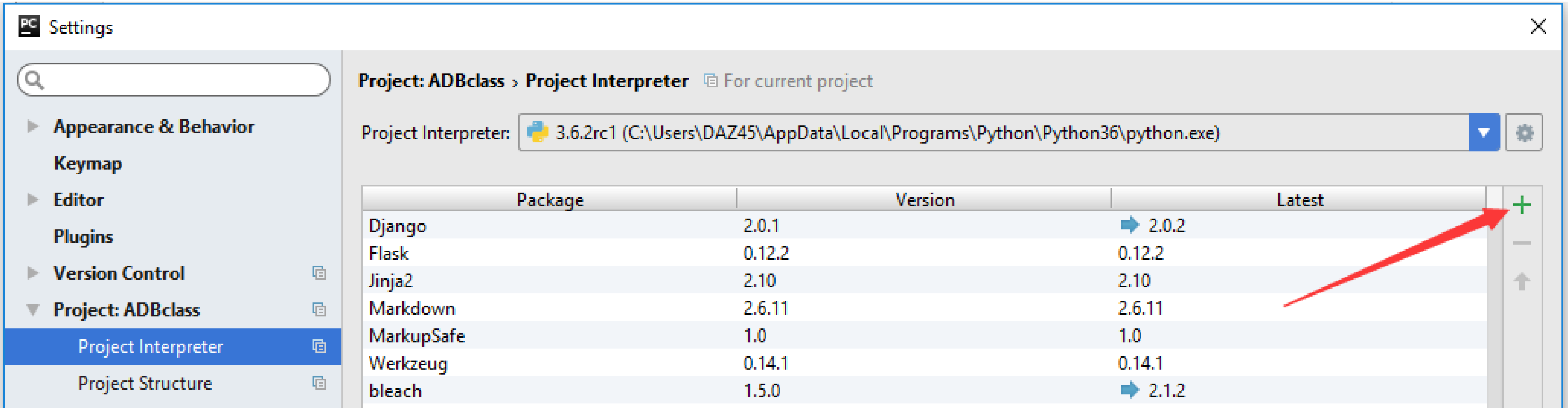
- Traverse the nodes in the path or a result of aggregation.

- MATCH p = (a:person)-[*]->(b:player)
- WHERE a.name = "Kary" AND b.name = "Kelly"
- FOREACH (n IN nodes(p)| SET n.marked = TRUE)
- return p



Control Neo4j with python

- You need py2neo
  - $ pip install py2neo
  - Pycharm=>File=>Settings=>Project:**=>Project Interpreter=>



Project: ADBclass › Project Interpreter For current project

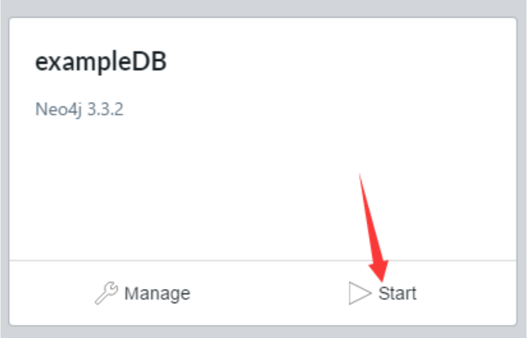
Project Interpreter: 3.6.2rc1 (C:\Users\DAZ45\AppData\Local\Programs\Python\Python36\python.exe)

Package	Version	Latest
Django	2.0.1	➡ 2.0.2
Flask	0.12.2	0.12.2
Jinja2	2.10	2.10
Markdown	2.6.11	2.6.11
MarkupSafe	1.0	1.0
Werkzeug	0.14.1	0.14.1
bleach	1.5.0	➡ 2.1.2

Connection to Neo4j with python

- graph = Graph("localhost:7474", username = "neo4j", password = "112358")

- First, build a database, and turn on the server.
- Second, code and run it.

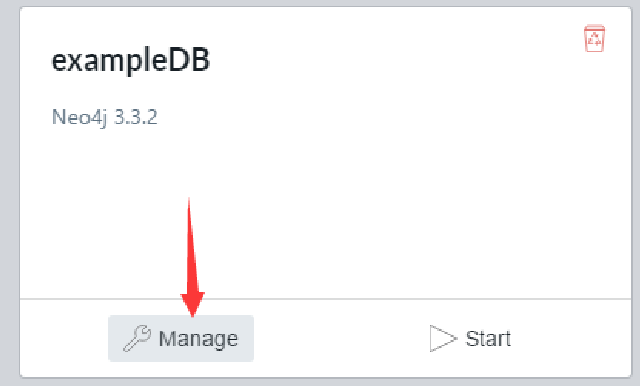


exampleDB

Neo4j 3.3.2

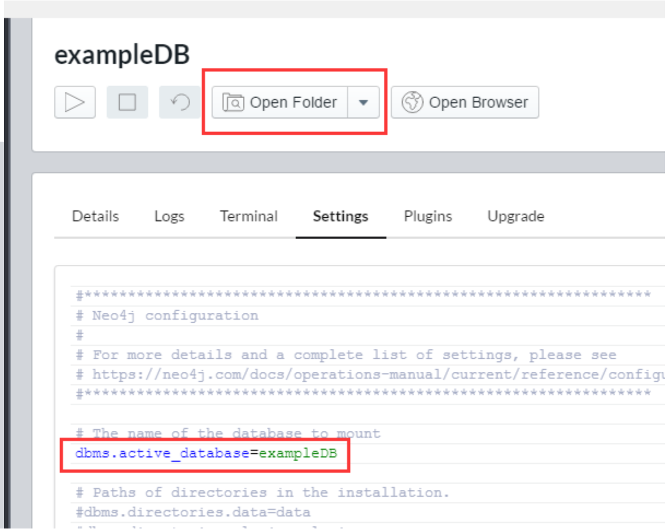
Import jar file for Java

- Right click on your project in eclipse => Build Path => Add external jars => import all jar files in the lib folder.
- Close the neo4j server before running Java.



exampleDB

Neo4j 3.3.2

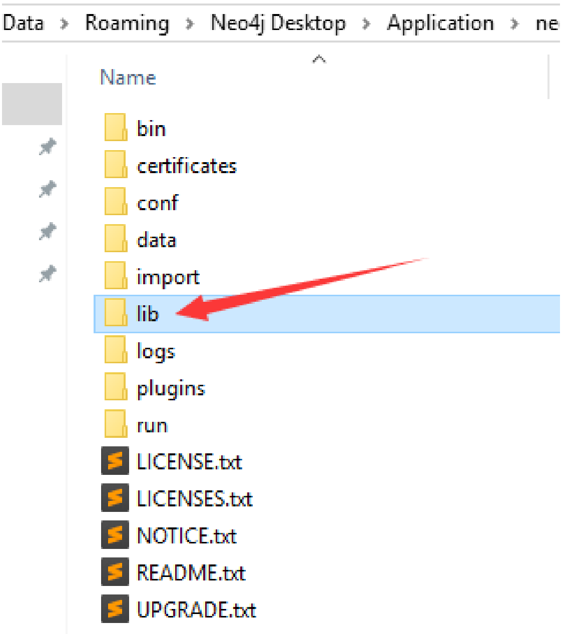


exampleDB

```
#*****************************************************************
# Neo4j configuration
#
# For more details and a complete list of settings, please see
# https://neo4j.com/docs/operations-manual/current/reference/configu
#*****************************************************************

# The name of the database to mount
dbms.active_database=exampleDB

# Paths of directories in the installation.
#dbms.directories.data=data
```



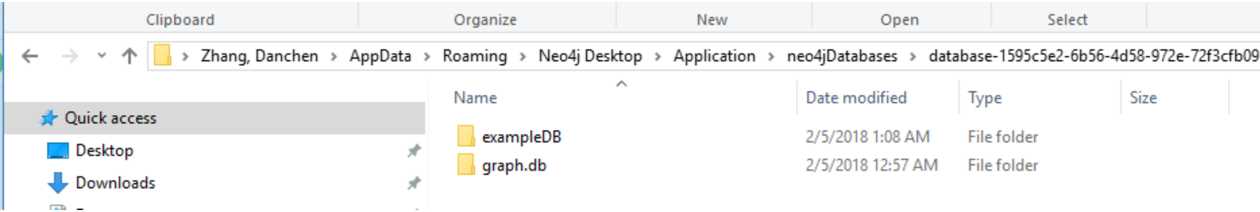
Data › Roaming › Neo4j Desktop › Application › ne

Name

- bin
- certificates
- conf
- data
- import
- lib
- logs
- plugins
- run
- LICENSE.txt
- LICENSES.txt
- NOTICE.txt
- README.txt
- UPGRADE.txt

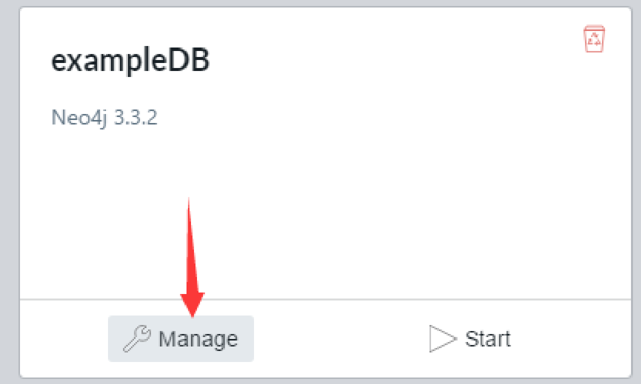
Connect Neo4j with Java

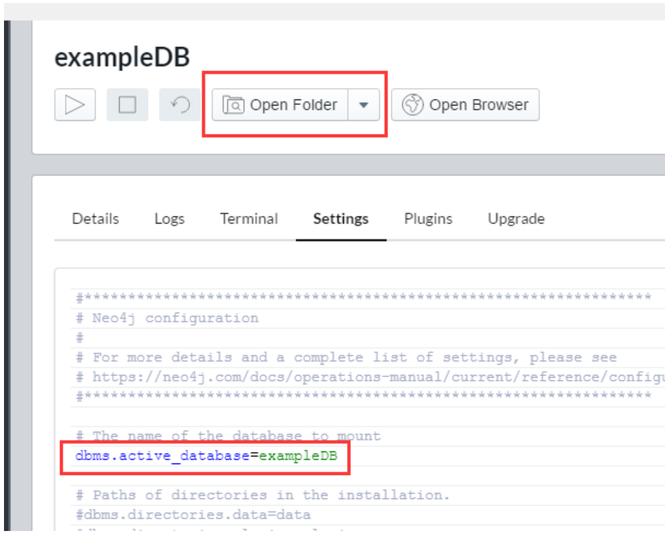
- GraphDatabaseFactory dbFactory = new GraphDatabaseFactory();
- GraphDatabaseService db = dbFactory.newEmbeddedDatabase(new File("C:\\Users\\.........\\installation-3.3.2\\data\\databases\\exampleDB"));



› Zhang, Danchen › AppData › Roaming › Neo4j Desktop › Application › neo4jDatabases › database-1595c5e2-6b56-4d58-972e-72f3cfb09153 › installation-3.3.2 › data › databases

Name	Date modified	Type	Size
exampleDB	2/5/2018 1:08 AM	File folder	
graph.db	2/5/2018 12:57 AM	File folder	





```
#*****************************************************************
# Neo4j configuration
#
# For more details and a complete list of settings, please see
# https://neo4j.com/docs/operations-manual/current/reference/configu
#*****************************************************************

# The name of the database to mount
dbms.active_database=exampleDB

# Paths of directories in the installation.
#dbms.directories.data=data
```

Any Q?