

Neo4j JDBC driver for Neo4j 3.x with BOLT protocol <http://neo4j-contrib.github.io/neo4j-jdbc/>

#neo4j #neo4j-driver #jdbc #integration #etl #business-intelligence #driver #java

🕒 315 commits

🌿 13 branches

📦 11 releases

👥 12 contributors

🔖 Apache-2.0

Branch: 3.1 ▾

New pull request



Create new file


Upload files

Find file

Clone or download ▾


This branch is 141 commits ahead, 1 commit behind larusba:master.

 Pull request  Compare

 jexp removed index.html from /docs

Latest commit a5c1171 on May 15

📁 docs	removed index.html from /docs	6 months ago
📁 neo4j-jdbc-bolt	next version 3.1.1	6 months ago
📁 neo4j-jdbc-driver	next version 3.1.1	6 months ago
📁 neo4j-jdbc-examples	next version 3.1.1	6 months ago
📁 neo4j-jdbc-http	next version 3.1.1	6 months ago
📁 neo4j-jdbc	next version 3.1.1	6 months ago
📄 .gitignore	One jar driver	a year ago
📄 .travis.yml	Running integration and unit test per default	a year ago
📄 README.adoc	fixed readme	6 months ago
📄 docs.sh	update readme sections to not show in the docs	a year ago
📄 index.html	preparing for release 3.1.0 (#99)	6 months ago
📄 license.txt	fixed license.txt	2 years ago
📄 neo4j-jdbc-idea-settings.jar	CHANGED LICENSE TO APACHE 2.0	2 years ago
📄 pom.xml	next version 3.1.1	6 months ago

 README.adoc

Overview: Neo4j JDBC Driver

build passing

You can find the full documentation at: <http://neo4j-contrib.github.io/neo4j-jdbc/>

This is the official JDBC driver for Neo4j.

This driver was mainly developed by [Larus BA, Italy](#), a certified consulting and integration solutions partner for Neo4j. Thank you so much for all your work.

Note

The previous JDBC driver for Neo4j 2.x was moved to the <https://github.com/neo4j-contrib/neo4j-jdbc-2x> repository.

Being a graph database, Neo4j is not serving data in a relational way and the exact purpose of this project is to allow projects that are using the classic JDBC connector in the relational paradigm to interact with Neo4j.

This driver supports various types of database transports:

- through the Bolt protocol (3.1.x and 3.0.x) using `jdbc:neo4j:bolt://<host>:<port>/`
- through the HTTP protocol (2.x+) using `jdbc:neo4j:http://<host>:<port>/`

Going forward there will also be support for:

- direct file connection
- embedded connection

Maven dependency

For the all-in-one module supporting both Bolt and HTTP, you can simply use:

```
<dependency>
  <groupId>org.neo4j</groupId>
  <artifactId>neo4j-jdbc-driver</artifactId>
  <version>{neo4j-jdbc-version}</version>
</dependency>
```

If you only want one of the protocols, you can depend on its module:

```
<dependency>
  <groupId>org.neo4j</groupId>
  <artifactId>neo4j-jdbc-bolt</artifactId>
  <version>{neo4j-jdbc-version}</version>
</dependency>
```

or

```
<dependency>
  <groupId>org.neo4j</groupId>
  <artifactId>neo4j-jdbc-http</artifactId>
  <version>{neo4j-jdbc-version}</version>
</dependency>
```

Minimum viable snippet

Dependency declaration

```
org.neo4j:neo4j-jdbc-driver:3.1.0
```

Run Query

```
// Connecting
try (Connection con = DriverManager.getConnection("jdbc:neo4j:bolt://localhost", 'neo4j', password)) {

  // Querying
  String query = "MATCH (u:User)-[:FRIEND]-(f:User) WHERE u.name = {1} RETURN f.name, f.age";
  try (PreparedStatement stmt = con.prepareStatement(query)) {
    stmt.setString(1, "John");

    try (ResultSet rs = stmt.execute()) {
      while (rs.next()) {
        System.out.println("Friend: "+rs.getString("f.name")+" is "+rs.getInt("f.age"));
      }
    }
  }
}
```

Please note that the example above uses the try-with-resource blocks that automatically closes resources when the `try` block is exited.

Usage with Neo4j Server

- Install a Neo4j 3.1.x server and start it with the Bolt protocol enabled
- Connect with URLs in the form `jdbc:neo4j:bolt://<host>:<port>/` (e.g. `jdbc:neo4j:bolt://localhost/`)
- You can also use additional comma separated parameters in the URL for authentication, debug mode, SSL encryption and flattening e.g. `jdbc:neo4j:bolt://localhost/?user=neo4j,password=xxxx,debug=true,noSsl,flatten=[-1,100,1000]`
- Add the JDBC driver dependency or jar file to your project
- Get a connection from `DriverManager`
- Execute queries and transactions using the Cypher graph query language

Flattening

As most JDBC clients and tools don't support complex objects, the driver can flatten returned nodes and relationships by providing all their properties as individual columns with names like `u.name`, `r.since` if you just return a node `u` or relationship `r`.

This is enabled with the JDBC-URL parameter `flatten=<rows>`, where `<rows>` indicates how many rows are sampled to determine those columns. With `-1` all rows are sampled and with any other value you determine the number of rows being looked at.

Building the driver yourself

First clone [the repository](#).

This project is composed by the following modules:

- [Neo4j JDBC](#) - the core module
- [Neo4j JDBC - Bolt](#) - module supporting the Bolt protocol
- [Neo4j JDBC - HTTP](#) - module supporting the HTTP protocol
- [Neo4j JDBC - Driver packaging](#) - module to package all above modules

Run all unit tests & integration tests

```
mvn clean test
```

Run only integration tests

```
mvn clean test -Pintegration-test
```

Run performance tests

```
mvn clean test -Pperformance-test
```

Note
To run the performance test, you must have a Neo4j Server 3.1.x running with the Bolt protocol enabled on port 7687 (default)

License

Copyright (c) 2017 [Neo4j](#) and [LARUS Business Automation](#)

The "Neo4j JDBC Driver" is licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Feedback

Please provide feedback and report bugs as [GitHub issues](#) or join the [neo4j-users Slack](#) and ask on the #neo4j-jdbc channel. You might also ask on [StackOverflow](#), please tag your question there with `neo4j` and `jdbc`.