

Neo4j Data Protocol


Merged

jakewins

merged 1 commit into `neo4j:2.3` from `jakewins:2.3-ndp` Apr 6, 2015

Conversation37Commits1Files changed96

+12,539-1




jakewins

commented Apr 1, 2015


Collaborator

This contains the initial work on a server portion of a lower-overhead data protocol for remote clients.

- 


chrisvest

commented on an outdated diff Apr 1, 2015

Show outdated diff
- 


chrisvest

commented on an outdated diff Apr 1, 2015

Show outdated diff
- 

chrisvest

commented on an outdated diff Apr 1, 2015

Show outdated diff
- 

lutovich

commented on the diff Apr 1, 2015

Labels

2.3server


Milestone

No milestone


Assignee

No one assigned

5 participants



...java/org/neo4j/ndp/messaging/v1/util/BytePrinter.javaView full changes



((14 lines not shown))

14+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15+ * GNU General Public License for more details.
16+ *
17+ * You should have received a copy of the GNU General Public License
18+ * along with this program. If not, see <http://www.gnu.org/licenses/>.
19+ */
20+package org.neo4j.ndp.messaging.v1.util;
21+
22+import java.io.ByteArrayOutputStream;
23+import java.io.PrintStream;
24+import java.io.UnsupportedEncodingException;
25+import java.nio.ByteBuffer;
26+
27+import static java.nio.ByteBuffer.wrap;
28+
29+public class BytePrinter

lutovich

added a note Apr 1, 2015

Collaborator

There exists a similar thing already: `org.neo4j.kernel.impl.util.HexPrinter` . It would probably be good to consolidate these two guys in one.


jakewins

added a note Apr 1, 2015

Collaborator

+1, I'll add a card for doing that


Add a line note



lutovich

commented on the diff Apr 1, 2015

...rc/test/java/org/neo4j/packstream/PackStreamTest.javaView full changes

		((45 lines not shown))
45		+ *
46		+ * This file is part of Neo4j.
47		+ *
48		+ * Neo4j is free software: you can redistribute it and/or modify
49		+ * it under the terms of the GNU General Public License as published by
50		+ * the Free Software Foundation, either version 3 of the License, or
51		+ * (at your option) any later version.
52		+ *
53		+ * This program is distributed in the hope that it will be useful,
54		+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
55		+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
56		+ * GNU General Public License for more details.
57		+ *
58		+ * You should have received a copy of the GNU General Public License
59		+ * along with this program. If not, see <http://www.gnu.org/licenses/>.
60		+ */



lutovich added a note Apr 1, 2015

Collaborator

This license header is redundant.



jakewins added a note Apr 5, 2015

Collaborator

GH is being too clever here, it moved your comment to the first header when I removed the other one.. Anyway removed the duplicate header :)

Add a line note



cleishm commented on an outdated diff Apr 1, 2015




Show outdated diff



cleishm commented on the diff Apr 1, 2015

community/ndp/pom.xml

[View full changes](#)

		((31 lines not shown))
31		+licensed under the GNU GENERAL PUBLIC LICENSE Version 3 to all third
32		+parties and that license is included below.
33		+
34		+However, if you have executed an End User Software License and Services
35		+Agreement or an OEM Software License and Support Services Agreement, or
36		+another commercial license agreement with Neo Technology or one of its
37		+affiliates (each, a "Commercial Agreement"), the terms of the license in
38		+such Commercial Agreement will supersede the GNU GENERAL PUBLIC LICENSE
39		+Version 3 and you may use the Software solely pursuant to the terms of
40		+the relevant Commercial Agreement.
41		+ </comments>
42		+ </license>
43		+ </licenses>
44		+
45		+
46		+ <modules>



cleishm added a note Apr 1, 2015

Owner

Do we need this many modules? A single "remote protocol" module should be sufficient - perhaps a discussion is required on this.



jakewins added a note Apr 2, 2015

Collaborator

Very good question. We've consciously tried an approach with a fine-grained set of modules. We certainly don't *need* them, we could solve the same things with a single module.

The intention with the fine-grained modules is that they *force* you to design without adding circular dependencies between components, vs. doing the same thing with packages where there is nothing stopping someone from introducing bad dependencies across components.

In the other end of the scale are things like the overhead of maintaining this many modules, and of

course that it's trickier to deal with if you get the modules themselves wrong in the design.

I've found it to be much less overhead than what I'd anticipated, and must say I quite like the fine-grained modules. I think that, beyond the architecturally-proper straightjacket, it also helps looking at the code since things are in tiny labelled boxes. All that said, I really don't feel strongly about it, I'd be ok swapping it to a single module (or two) if people feel strongly opposed to strict compartmentalization.



thobe added a note Apr 2, 2015

Owner

I'm very much in favour of small modules for the same reasons. It helps finding code. And it helps finding tests, the test code comes "closer" to the code-under-test, and if you look at for example kernel, finding out where something is tested is extremely hard.



jakewins added a note Apr 2, 2015

Collaborator

The only concern I have is that I, for OCD reasons, would very much like our Maven Central footprint to not contain hundreds of modules. I don't think that's an issue here though, as I expect these modules should not be needed in maven central. I may be wrong there though, but even then we could look into bundling larger modules to maven central to make the public module API easy to digest.



cleishm added a note Apr 2, 2015

Owner

+1 (to both). I like the clear separation of concerns that modules can give. I don't like the plethora of jars that end up on Maven Central and in the product. I don't like the dependency issues that get created, esp. wrt tests.



jakewins added a note Apr 3, 2015

Collaborator

So, on the maven and product issues, I *think* both can be solved by some well planned packaging (eg. uber-jar for the product and simply not putting this part of the code base on maven central for maven, or uber-jar there as well). The test jars is an issue. Like Tobias notes, its awesome to have the tests close to small distinct modules. However, of course, we often have associated test tooling and utility code which we'd prefer to not duplicate across modules.

IIIRC, you sent out a few proposals for options on solving that latter issue just a few days ago :)

Add a line note



lutovich commented on the diff Apr 1, 2015

...test/java/org/neo4j/ndp/transport/http/util/HTTP.java

[View full changes](#)



((36 lines not shown))

```
36 +import org.neo4j.server.rest.domain.JsonParseException;
37 +import org.neo4j.ndp.messaging.v1.util.BytePrinter;
38 +
39 +import static java.util.Collections.unmodifiableMap;
40 +import static org.hamcrest.Matchers.allOf;
41 +import static org.hamcrest.Matchers.anyOf;
42 +import static org.hamcrest.Matchers.containsString;
43 +import static org.hamcrest.Matchers.not;
44 +import static org.junit.Assert.assertThat;
45 +import static org.neo4j.helpers.collection.IteratorUtil.singleOrNull;
46 +import static org.neo4j.helpers.collection.MapUtil.stringMap;
47 +
48 +/**
49 + * A tool for performing REST HTTP requests
50 + */
51 +public class HTTP
```



lutovich added a note Apr 1, 2015

Collaborator

Is it exactly same thing as `org.neo4j.test.server.HTTP` ?



lutovich added a note Apr 1, 2015

Collaborator

No it is not, but these two should definitely be combined.



jakewins added a note Apr 2, 2015

Collaborator

It almost is indeed. Although, it seems we'll drop the regular HTTP transport introduced in this PR anyhow, so we won't need this client in the medium term.



jakewins added a note Apr 5, 2015

Collaborator

I think I should expand this a bit further: I don't want to use the server HTTP client here & now, because it would require adding a dependency on the server, which will become a circular dependency if we pull this into the Neo4j Server. The right thing to do would be to have something like a test-util module for the HTTP stuff, but it seems that's a bit of overhead to add since the plan is to remove this altogether in the next two weeks or so.

Add a line note



cleishm commented on an outdated diff Apr 1, 2015



Show outdated diff



thobe commented Apr 2, 2015

Owner

I would really really really like for there to be a description about each module. Something about the purpose and responsibilities of each module, possibly how it ties in to other things. The `README` files for each module seem like they would be a great place for such a description.

Having a description would help a lot with getting down and dirty with the code.



jakewins commented Apr 2, 2015

Collaborator

@thobe I've removed the u-ext module for now, and added a `README` with an overview in the `ndp` super-module.



cleishm commented on the diff Apr 2, 2015

...neo4j/ndp/messaging/v1/PackStreamMessageFormatV1.java

[View full changes](#)



((36 lines not shown))

```
36 +import org.neo4j.graphdb.PropertyContainer;
37 +import org.neo4j.graphdb.Relationship;
38 +import org.neo4j.helpers.collection.Iterables;
39 +import org.neo4j.ndp.messaging.v1.infrastructure.ValueNode;
40 +import org.neo4j.ndp.messaging.v1.infrastructure.ValuePath;
41 +import org.neo4j.ndp.messaging.v1.infrastructure.ValueRelationship;
42 +import org.neo4j.ndp.messaging.v1.message.Message;
43 +import org.neo4j.packstream.PackStream;
44 +import org.neo4j.packstream.PackType;
45 +import org.neo4j.ndp.runtime.internal.Neo4jError;
46 +import org.neo4j.stream.Record;
47 +
48 +import static org.neo4j.ndp.runtime.internal.Neo4jError.codeFromString;
49 +import static org.neo4j.stream.Records.record;
50 +
51 +public class PackStreamMessageFormatV1 implements MessageFormat
```



cleishm added a note Apr 2, 2015

Owner

This is already in a `v1` package, so probably don't need it in the name also.



jakewins added a note Apr 3, 2015

Collaborator

I actually like having it in both - the package groups the related stuff for that version, and the version in the name clarifies that this is the v1 message format and allows mixing this class with

other versions without fully qualified names.

Add a line note



cleishm commented on an outdated diff Apr 4, 2015



Show outdated diff



cleishm commented on an outdated diff Apr 4, 2015



Show outdated diff



cleishm commented on an outdated diff Apr 4, 2015



Show outdated diff



lutovich commented on an outdated diff Apr 4, 2015



Show outdated diff



jakewins added `server` `2.3` labels Apr 5, 2015



Neo4j Data Protocol



350a1aa



jakewins merged commit `11a1037` into `neo4j:2.3` Apr 6, 2015
1 check passed

View details



jakewins deleted the `jakewins:2.3-ndp` branch Apr 6, 2015



lutovich referenced this pull request Apr 9, 2015

Introducing Java driver core framework #4401

Closed

Sign up for free

to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

