

## Getting an AST for a Cypher query?

5 posts by 3 authors  



**Rich Morin**

Apr 25



Ronja looks like a great addition to Neo4j. I'd like to leverage it for my own (exploratory) work.

By way of background, the first stage of Neo4j 2.2's Cypher processing is to:

### **Convert the input query string into an abstract syntax tree (AST)**

The input query string is first tokenized and then parsed into an [AST](#).

Using this AST, we perform semantic checking of the variable types and scoping of variables within the tree. Any errors regarding basic typing information – such as attempting to divide two strings – are caught and returned to the user.

-- Introducing the new Cypher Query Optimizer

<http://neo4j.com/blog/introducing-new-cypher-query-optimizer>

I would like to be able to send a Cypher query to Neo4j and (*without* executing the query) get the resulting AST. For extra credit, I'd like to see some of the data structures (eg, optimized and normalized AST, query graph, logical plan, execution plan) that Ronja generates.

Can anyone point me to (a) information on how I might do this and/or (b) documentation on these data structures?

-r



**Michael Hunger**

Apr 25



I think you would have to write code to access the internal structures in the cypher compiler

Most probably to return the data that's cached in the two execution plan caches

The code is in the cypher-compiler modules for each version

Von meinem iPhone gesendet

- show quoted text -

- show quoted text -

--

You received this message because you are subscribed to the Google Groups "Neo4j" group.

To unsubscribe from this group and stop receiving emails from it, send an email to [neo4j+un...@googlegroups.com](mailto:neo4j+unsubscribe@googlegroups.com).

For more options, visit <https://groups.google.com/d/optout>.



**Rich Morin**

Apr 29



I'd like to minimize the amount of Java hacking I need to do, particularly when it involves experimental, special-purpose code. Fortunately, I think I see a general solution that would serve a number of disparate purposes.

Basically, I'd like to have a way to register callbacks (dynamically) to the Ronja control flow. So, I've written this up as a [Neo4j issue](#) (ducks :-).

-r



**Clark Richey**

Apr 29



I do have to say that being able to access the internal Cypher interpreter / parser would be of great help. I looked at trying to access it with Neo 2.1.6 but simply couldn't figure out how to properly do so (lots of dependencies of things maintained by the neo kernel). I am interested in using it for parsing out cypher that is sent via an API so that I can do things such as add in additional security clauses (match patterns) and look for things that may not be allowed such as deletions, merges, access to certain properties, etc.

We are currently writing our own parser to handle this but it would be SO much more efficient if there was an easy way to ask the neo kernel to parse text into cypher and get access to its constituent parts.

Clark Richey  
[clark....@gmail.com](mailto:clark....@gmail.com)

- show quoted text -  
- show quoted text -

--

You received this message because you are subscribed to the Google Groups "Neo4j" group.  
To unsubscribe from this group and stop receiving emails from it, send an email to [neo4j+un...@googlegroups.com](mailto:neo4j+un...@googlegroups.com).  
For more options, visit <https://groups.google.com/d/optout>.



**Rich Morin**

Apr 29



Chris Leishman indicated (in an [issue response](#)) that this thread is the best place to make a case for the callback registration idea, ask for help,

