

Ten. Million. Questions. Let's celebrate [all we've done together](#).

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free.

Take the 2-minute tour x

## Auto increment property in Neo4j



### Work at SmartSheet

Bellevue, WA



Bruce Leban  
Software Development Manag..



Kim Brandl  
Developer Experience & Part...

[asp.net](#) [jquery](#) [css3](#)

As far as I understand it the IDs given by Neo4j ( `ID(node)` ) are unstable and behave somewhat like row numbers in SQL. Since IDs are mostly used for relations in SQL and these are easily modeled in Neo4j, there doesn't seem to be much use for IDs, but then how do you solve retrieval of specific nodes? Having a REST API which is supposed to have unique routes for each node (e.g. `/api/concept/23` ) seems like a pretty standard case for web applications. But despite it being so fundamental, the only viable way I found were either via

- language specific frameworks
- as an unconnected node which maintains the increments:

```
// get unique id
MERGE (id:UniqueId{name:'Person'})
ON CREATE SET id.count = 1
ON MATCH SET id.count = id.count + 1
WITH id.count AS uid
// create Person node
CREATE (p:Person{id:uid,firstName:'Gabriel',lastName:'Smith'})
RETURN p AS person
```

Source: <http://www.neo4j.org/graphgist?8012859>

Is there really not a simpler way and if not, is there a particular reason for it? Is my approach an anti-pattern in the context of Neo4j?

[neo4j](#) [language-agnostic](#)

share improve this question

add a comment

asked Apr 3 at 14:24



Brewer Gorge  
272 • 1 • 15

1 Answer

active oldest votes

Neo4j internal ids are a bit more stable than sql row id's as they will never change during a transaction for e.g.

And indeed exposing them for external usage is not recommended. I know there are some intentions at Neo internals to implement such a feature.

Basically people tend to use two solutions for this :

1. Using a UUID generator at the application level like for PHP : <https://packagist.org/packages/rhumsaa/uuid> and add a label/uuid unique constraint on all nodes.
2. Using a very handful Neo4j plugin like <https://github.com/graphaware/neo4j-uuid> that will add uuid properties on the fly, so it remove you the burden to handle it at the application level and it is easier to manage the persistence state of your node objects

share improve this answer

add a comment

answered Apr 3 at 16:17



Christophe Willemssen  
5,115 • 1 • 4 • 17

asked 4 months ago

viewed 181 times

active 4 months ago

### Blog

[Why Stack Overflow is a Good Workplace for Women](#)



### SmartSheet

based in Bellevue, WA

### Who you'll work with...



Bruce Leban  
Software Development Ma



Kim Brandl  
Developer Experience & P...  
[asp.net](#) [jquery](#) [css3](#)



Kevin Tao

[view company page](#)

### Linked

- 1 [Reliable \(auto\)incrementing identifiers for all nodes/relationships in Neo4j](#)

### Related

- 1 [Auto Increment in Neo4j](#)
- 0 [optimistic concurrency with neo4j using the REST API](#)
- 1 [multiple loads in neo4j](#)
- 0 [neo4j rebuild auto index](#)
- 2 [Neo4j incremental backup strategy](#)
- 0 [Neo4j Embedded - Auto Index Multiple Properties](#)

