

Does or will Spring Data Neo4j 4.0 support @Query annotations in node entities ?

5 posts by 4 authors  

Leward

Apr 20



Hello,

With Spring Data Neo4j 3.X I used to have the following in my entities:

```
@NodeEntity
public class Agency {

    @GraphId
    private Long id;

    private String name

    @Query("start n=node({self}) match n<-[:BELONG_TO]-(u:User) return count(u);")
    private int numberOfEmployees;

}
```

However in SDN 4.0.0M1 the @Query annotation only applies to method. It is written in the documentation that this works for repository methods, however nothing is mentioned about using the @Query annotation.

Does SDN support such a thing?

If the answer is no, I guess I can make it work in one query with Cypher (getting the agency and the count of agency members), but then how to map it back to a proper SDN entity?

Kind Regards,
Leward



Sumit Gupta

Apr 20



Hi,

I had a cursory look at the 4.0 docs but there is no where mentioned that @Query is only for methods and cannot be applied to prooerties.

http://docs.spring.io/spring-data/neo4j/docs/4.0.0.M1/reference/html/#reference_programming_model_annotatedQueries

Thanks,
Sumit
- show quoted text -

Leward

Apr 22



Hello,

If you look at the source code for @Query here:<https://github.com/spring-projects/spring-data-neo4j/blob/4.0/spring-data-neo4j/src/main/java/org/springframework/data/neo4j/annotation/Query.java> you will see the following:

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
@QueryAnnotation
@Documented
public @interface Query {
```

It states that the annotation will work only when applied to methods.

However in SDN 4.0.0M1 the @Query annotation only applies to method. It is written in the documentation that this works for repository methods, however nothing is mentioned about using the @Query annotation.

Does SDN support such a thing?

If the answer is no, I guess I can make it work in one query with Cypher (getting the agency and the count of agency members), but then how to map it back to a proper SDN entity?

Kind Regards,
Leward



Luanne

Apr 22



Re: [Neo4j] Re: Does or will Spring Data Neo4j 4.0 support @Query annotations in node entities ?
Other recipients: vi...@graphaware.com

Hi

@Query is currently only available on Repository methods. We discussed adding it to domain entity classes, and decided to defer that for M1, primarily for complexity/performance reasons. (Imagine a node entity with 10 @Query annotations and then a request to load a couple of hundred of those entities in one go). In the worst case (naive) scenario, a single request end up hitting the database thousands of times.

As an alternative, you can do this:

...

```
public class Agency {
```

```
    @Relationship(type="BELONG_TO", direction=Relationship.INCOMING)
```

```
    private List<Employee> employees
```

```
    private int numberOfEmployees() {
```

```
        return employees.size();
```

```
    }
```

```
}
```

...

By default, when you load an entity, its immediate neighbours in the graph will be loaded as well, so you get the Employees information when you make a request for an Agency.

On behalf of Vince Bickers.

- 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>.



vi...@graphaware.com

Apr 22



Hi

@Query is currently only available on Repository methods. We discussed adding it to domain entity classes, and decided to defer that for M1, primarily for complexity/performance reasons. (Imagine a node entity with 10 @Query annotations and then a request to load a couple of hundred of those entities in one go). In the worst case (naive) scenario, a single request end up hitting the database thousands of times.

As an alternative, you can do this:

```
public class Agency {  
  
    @Relationship(type="BELONG_TO", direction=Relationship.INCOMING)  
    private List<Employee> employees  
  
    private int numberOfEmployees() {  
        return employees.size();  
    }  
}
```

By default, when you load an entity, its immediate neighbours in the graph will be loaded as well, so you will get the Employees information when you make a request for an Agency.

Vince