Neo4j and Cypher using Py2Neo

Python 2.x 2.7

Examples

Adding Nodes to Neo4j Graph

```
results = News.objects.todays_news()
for r in results:
    article = graph.merge_one("NewsArticle", "news_id", r)
    article.properties["title"] = results[r]['news_title']
    article.properties["timestamp"] = results[r]['news_timestamp']
    article.push()
    [...]
```

Adding nodes to the graph is pretty simple, `graph.merge_one` is important as it prevents duplicate items. (If you run the script twice, then the second time it would update the title and not create new nodes for the same articles)

`timestamp` should be an integer and not a date string as neo4j doesnt really have a date datatype. This causes sorting issues when you store date as '05-06-1989'

`article.push()` is an the call that actually commits the operation into neo4j. Dont forget this step.

Adding Relationships to Neo4j Graph

```
results = News.objects.todays_news()
for r in results:
    article = graph.merge_one("NewsArticle", "news_id", r)
    if 'LOCATION' in results[r].keys():
        for loc in results[r]['LOCATION']:
            loc = graph.merge_one("Location", "name", loc)
            try:
                rel = graph.create_unique(Relationship(article, "about_place", loc))
            except Exception, e:
                print e
```

`create_unique` is important for avoiding duplicates. But otherwise its a pretty straightforward operation. The relationship name is also important as you would use it in advanced cases.

Cypher Query Samples

Count articles connected to a particular person over time

```
MATCH (n)-[]->(l)
where l.name='Donald Trump'
RETURN n.date,count(*) order by n.date
```

Search for other People / Locations connected to the same news articles as Trump with at least 5 total relationship nodes.

```
MATCH (n:NewsArticle)-[]->(l)
where l.name='Donald Trump'
MATCH (n:NewsArticle)-[]->(m)
with m,count(n) as num where num>5
return labels(m)[0],(m.name), num order by num desc limit 10
```

Importing and Authenticating

```
from py2neo import authenticate, Graph, Node, Relationship
authenticate("localhost:7474", "neo4j", "<pass>")
graph = Graph()
```

You have to make sure your Neo4j Database exists at localhost:7474 with the appropriate credentials.

the `graph` object is your interface to the neo4j instance in the rest of your python code. Rather thank making this a global variable, you should keep it in a class's `__init__` method.

Query 1 : Autocomplete on News Titles

```
def get_autocomplete(text):
    query = """
    start n = node(*) where n.name =~ '(?i)%s.*' return n.name,labels(n) limit 10;
    """
    query = query % (text)
    obj = []
    for res in graph.cypher.execute(query):
        # print res[0],res[1]
        obj.append({'name':res[0],'entity_type':res[1]})
    return res
```

This is a sample cypher query to get all nodes with the property `name` that starts with the argument `text` .

Query 2 : Get News Articles by Location on a particular date

```
def search_news_by_entity(location,timestamp):
    query = """
    MATCH (n)-[]->(l)
    where l.name='%s' and n.timestamp='%s'
    RETURN n.news_id limit 10
    """

    query = query % (location,timestamp)

    news_ids = []
    for res in graph.cypher.execute(query):
        news_ids.append(str(res[0]))

    return news_ids
```

You can use this query to find all news articles `(n)` connected to a location `(l)` by a relationship.

Syntax

Parameters

Remarks