Neo4j介绍

图形数据库也称为图形数据库管理系统或GDBMS。图形数据库是以图形结构的形式存储数据的数据库。 它以节点，关系和属性的形式存储应用程序的数据。 正如RDBMS以表的“行，列”的形式存储数据，GDBMS以“图形”的形式存储数据。

**图数据库特点：**

如果我们使用RDBMS数据库来存储更多连接的数据，那么它们不能提供用于遍历大量数据的适当性能。 在这些情况下，Graph Database提高了应用程序性能。 它将每个配置文件数据作为节点存储在内部，它与相邻节点连接的节点，它们通过关系相互连接。

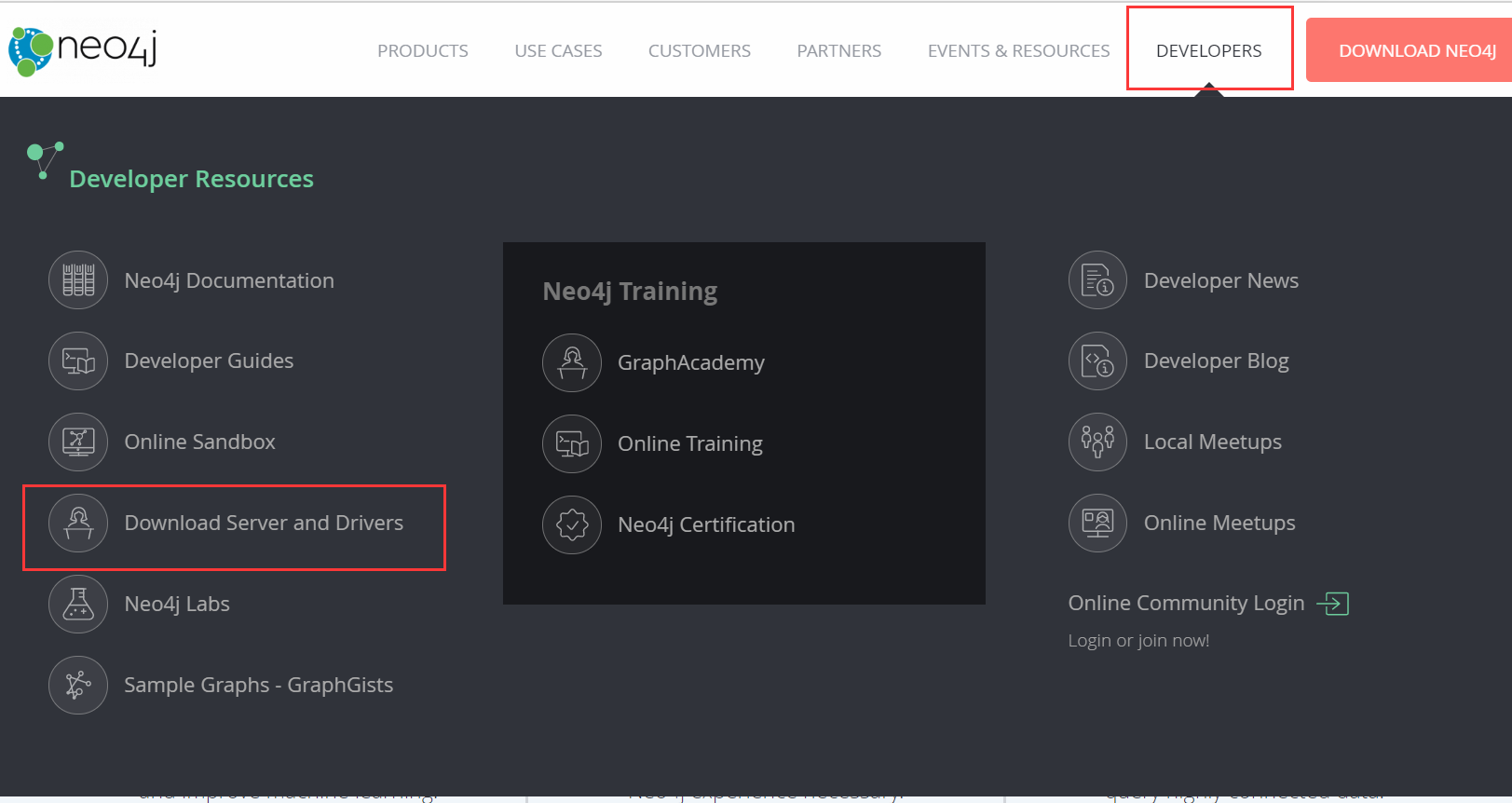
**Neo4j的优点**

* 它很容易表示连接的数据
* 检索/遍历/导航更多的连接数据是非常容易和快速的
* 它非常容易地表示半结构化数据
* Neo4j CQL查询语言命令是人性化的可读格式，非常容易学习
* 使用简单而强大的数据模型
* 它不需要复杂的连接来检索连接的/相关的数据，因为它很容易检索它的相邻节点或关系细节没有连接或索引

## Neo4j下载与安装

Neo4j 官网： <https://neo4j.com/>

### 下载

可根据需要选取所需版本下载。

### 安装

直接解压到安装目录即可。

### 服务器配置

* 修改配置

1. 进入neo4j目录下的conf文件夹，打开neo4j.conf 文件修改如下。

dbms.connectors.default\_listen\_address=0.0.0.0

# Bolt connector

dbms.connector.bolt.enabled=true

#dbms.connector.bolt.tls\_level=OPTIONAL

#dbms.connector.bolt.listen\_address=:7687

dbms.connector.bolt.address=0.0.0.0:7687

# HTTP Connector. There can be zero or one HTTP connectors.

dbms.connector.http.enabled=true

dbms.connector.http.listen\_address=0.0.0.0:7474

# HTTPS Connector. There can be zero or one HTTPS connectors.

dbms.connector.https.enabled=true

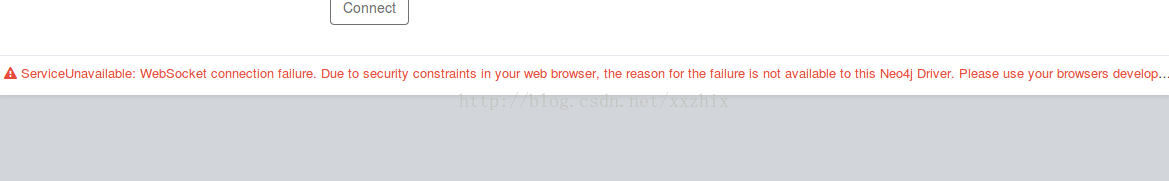
dbms.connector.https.listen\_address=0.0.0.0:7473

1. 开放服务器的7474和7687端口。

firewall-cmd –permanent –add-port=7474/tcp

firewall-cmd –reload

否则登录管理界面会一直出现错误。



* 常用命令

进入bin目录可执行如下命令：

后台启动：neo4j start

前台启动：neo4j console

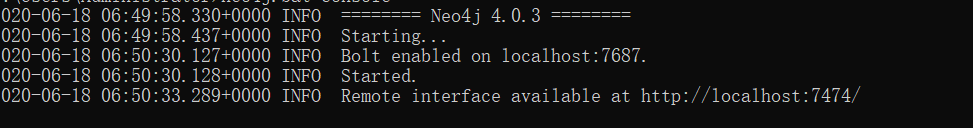
查看状态：neo4j status

停止：neo4j stop

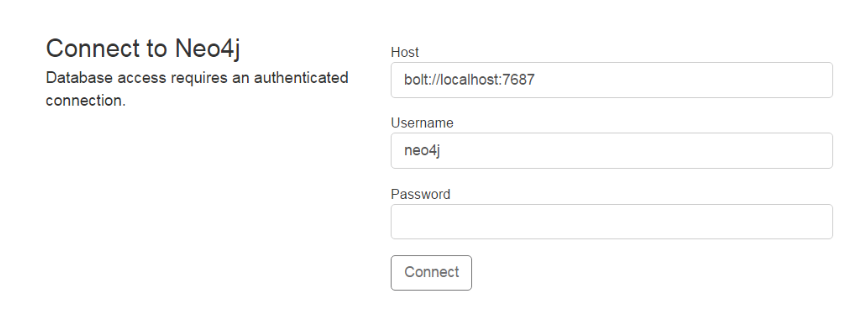
重启：neo4j restart

* 客户端访问

1. 进入bin目录后，执行neo4j console命，我们可以看到如下界面：

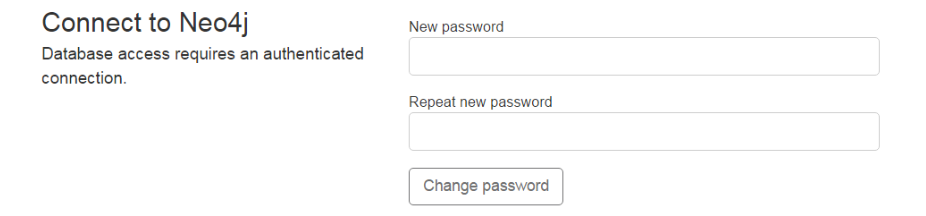


1. 如上图所示，已经开启了NEO4J数据库，可以在浏览器中登录http://localhost:7474网址查看数据库。
2. 首次登录会看到如下界面：



默认的账号和密码均为neo4j。

登陆后如下图：



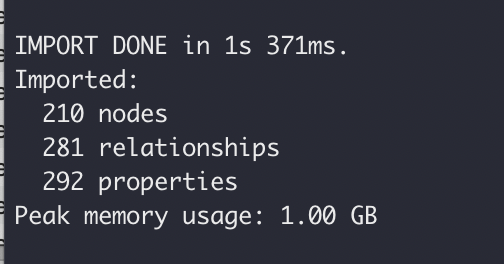
重新设置自己的账户名和密码后，则可进入主页面，至此neo4j配置成功。

## 数据导入

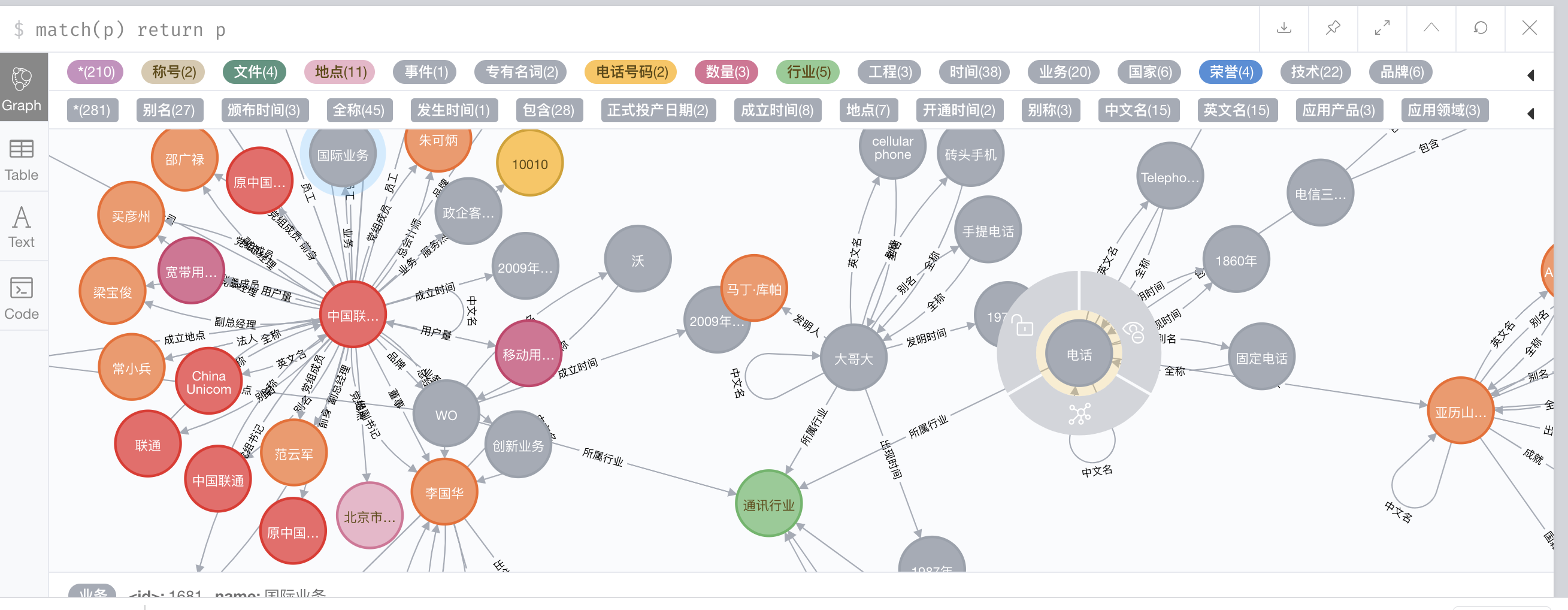
1. 导入工作前，需要关闭neo4j服务。
2. 准备好导入的csv文件，各个类型的实体表，以及通过实体定义的relation表。放到import文件夹下。
3. 删除databases/graph.db下所有文件。

命令格式如下：



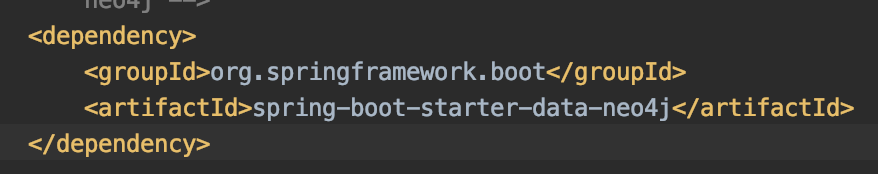


导入后重启服务，登录<http://localhost:7474/browser/>可以看到数据：

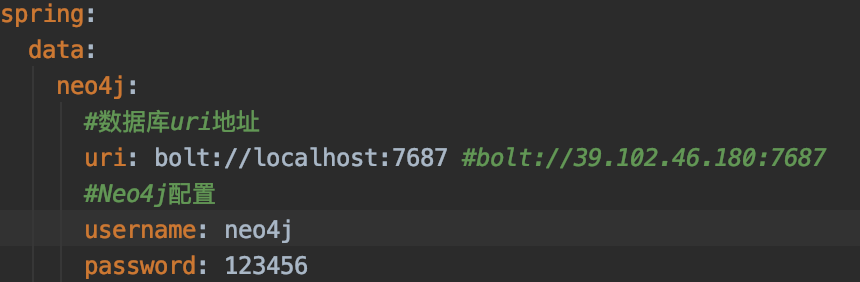


## Neo4j+Springboot

**项目依赖**



**配置**



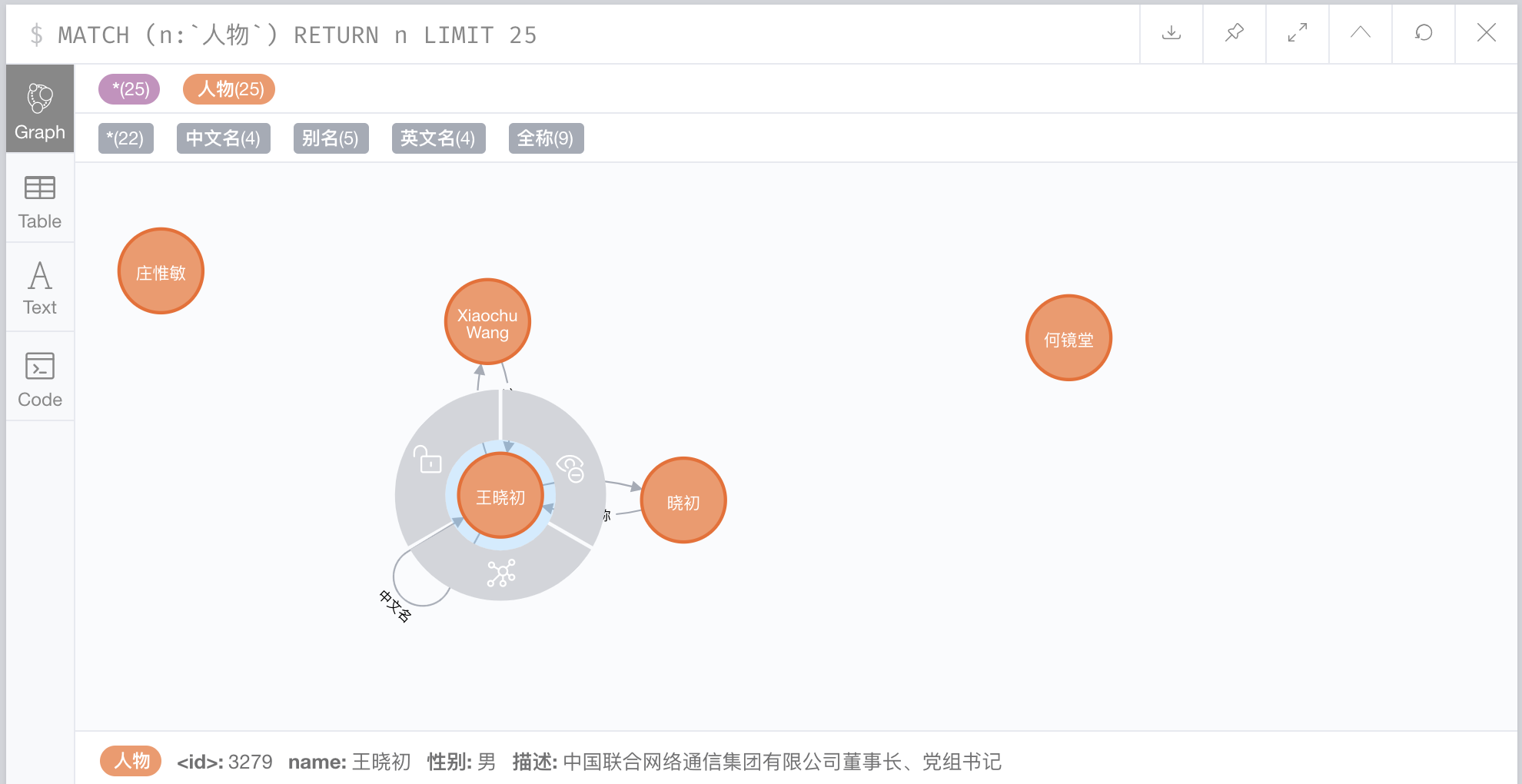
**开启Neo4j服务**

@EnableNeo4jRepositories

在Application入口文件中添加该注解

**基于图形节点进行Java类映射**

**（1）实体类**





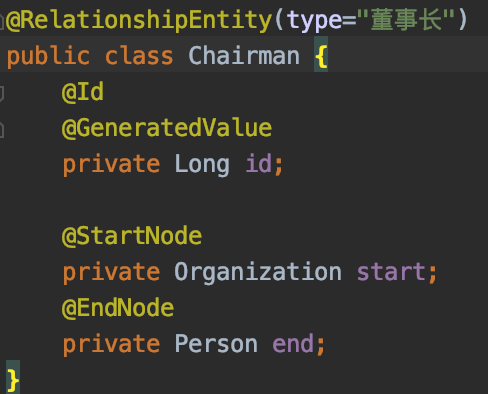
注解@NodeEntity(label=””)：表明该类是一个Neo4j的节点实体类，label必须与neo4j库里名字一致

@Id

@Property(name=””)：节点实体类的属性

注意，Java类中的属性一定要和Neo4j图形数据库中的节点的属性一一对应，否则映射失败！

**（2）关系类**

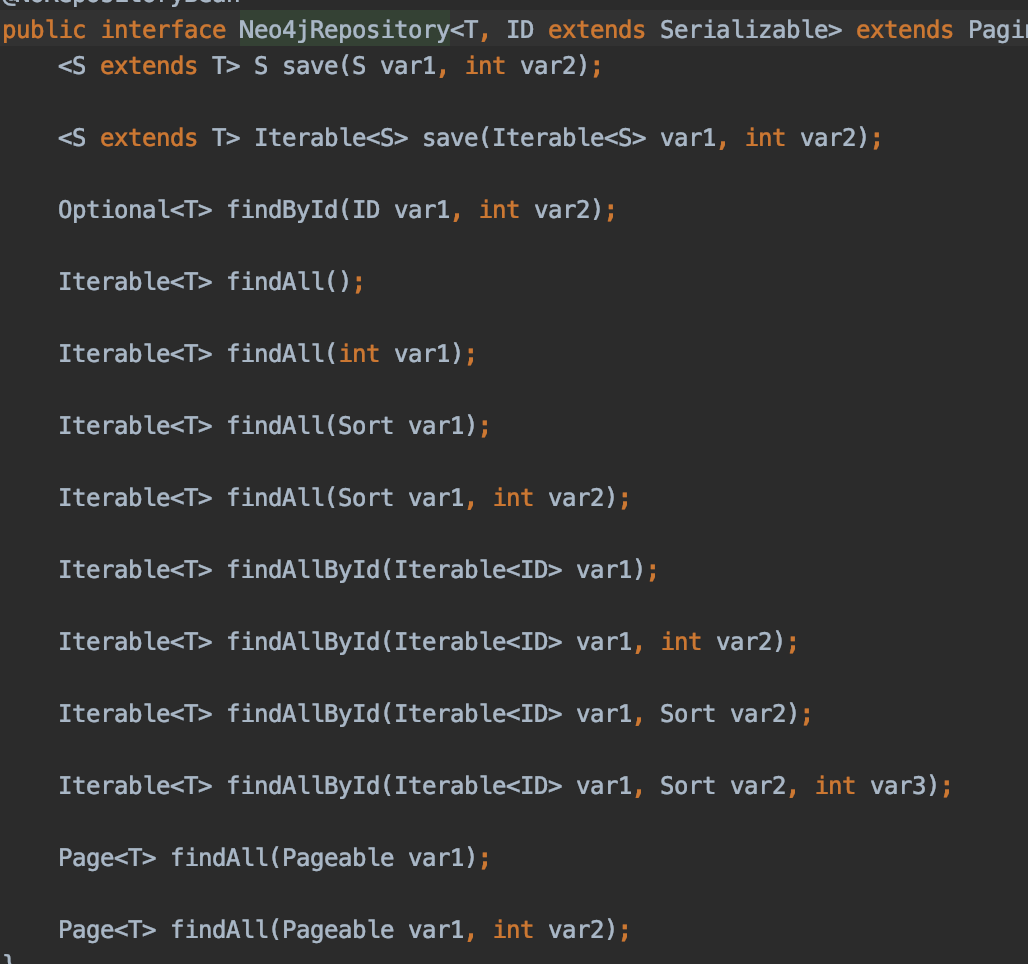


@StartNode和@EndNode分别指向头实体和尾实体。

**持久层，与neo4j数据库进行交互**

**——————spring-data-neo4j接口方式—————————**

继承 Neo4jRepository类实现增删查改，相当于Mybatis的mapper。



继承这些接口基本上对于简单的增删改查操作够用。

如果需要实现自己的接口：

1. 通过xx属性查询：spring-data-neo4j 支持方法命名约定查询 findBy{类的属性名}。例如通过name查询。

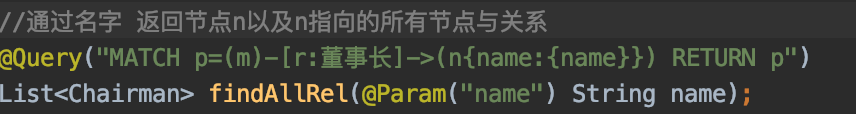
Person findByName(@Param("name") String name);

注意：findBy后面的属性名一定要在节点实体类里存在。

1. 其他：更多复杂的操作可学习Cypher语言+@Query注解即可实现。

例如：

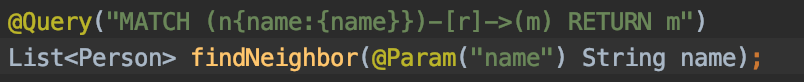
a.关系查询：通过实体name，返回该实体在对应关系下的关系对儿。



(查询name为王晓初的节点在董事长关系下的关系对儿)

[Chairman(id=7, start=Organization(id=3316, name=中国联合网络通信集团有限公司, desc=中国联合网络通信集团有限公司（简称“中国联通”），在国内31个省（自治区、直辖市）和境外多个国家和地区设有分支机构，是中国唯一一家在纽约、香港、上海三地同时上市的电信运营企业，连续多年入选“世界500强企业”。, slogan=创新·改变世界, vision=客户信赖的智慧生活的创造者, mission=联通世界 创享美好智慧生活, value=客为本 团队共进 开放创新 追求卓越, idea=一切为了客户 一切为了一线 一起为了市场), end=Person(id=3279, name=王晓初, desc=中国联合网络通信集团有限公司董事长、党组书记, sex=男))]

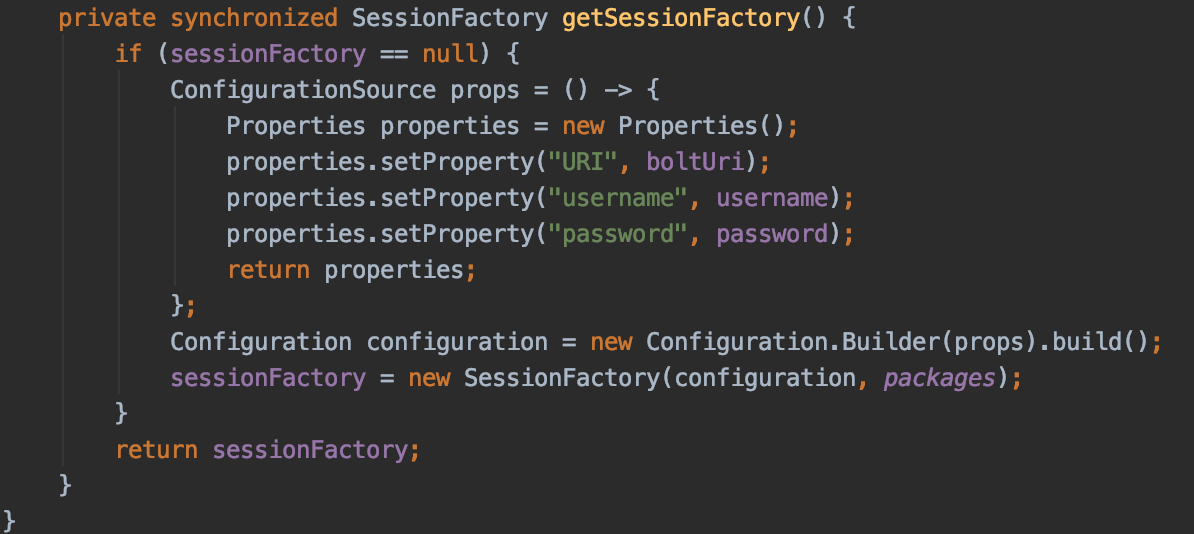
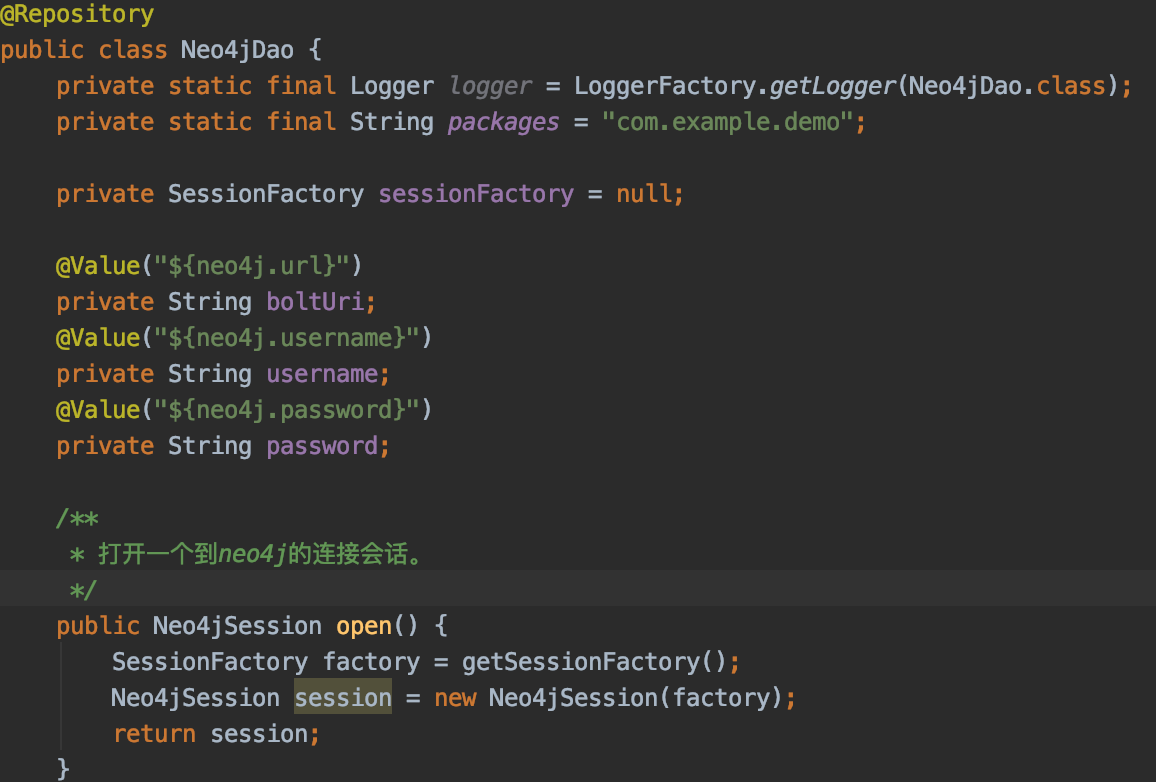
b.邻居节点的查询：通过实体name，返回该实体连接的所有Person类型的邻居节点。

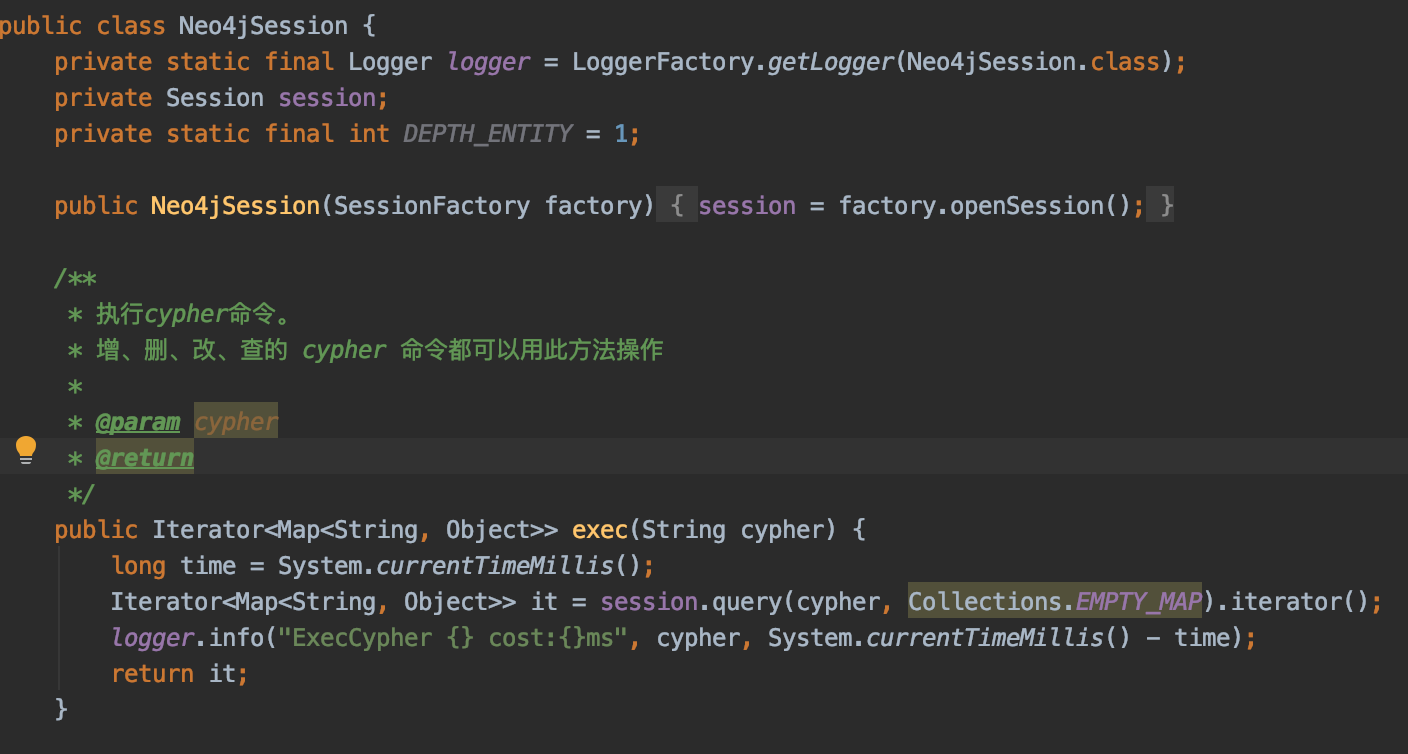


（查询与name为中国联合网络通信集团有限公司实体关联的Person实体）

[Person(id=3314, name=沈 洲, desc=null, sex=男), Person(id=3313, name=高鹏飞, desc=null, sex=男), Person(id=3312, name=陈华林, desc=null, sex=男), Person(id=3311, name=曹俊忠, desc=null, sex=男), Person(id=3310, name=李纪华, desc=null, sex=男), Person(id=3294, name=常小兵, desc=null, sex=男), Person(id=3293, name=朱可炳, desc=null, sex=男), Person(id=3292, name=范云军, desc=null, sex=男), Person(id=3291, name=梁宝俊, desc=null, sex=男), Person(id=3290, name=买彦州, desc=null, sex=男), Person(id=3289, name=邵广禄, desc=null, sex=男), Person(id=3280, name=李国华, desc=中国联合网络通信集团有限公司董事、总经理、党组副书记, sex=男), Person(id=3279, name=王晓初, desc=中国联合网络通信集团有限公司董事长、党组书记, sex=男)]

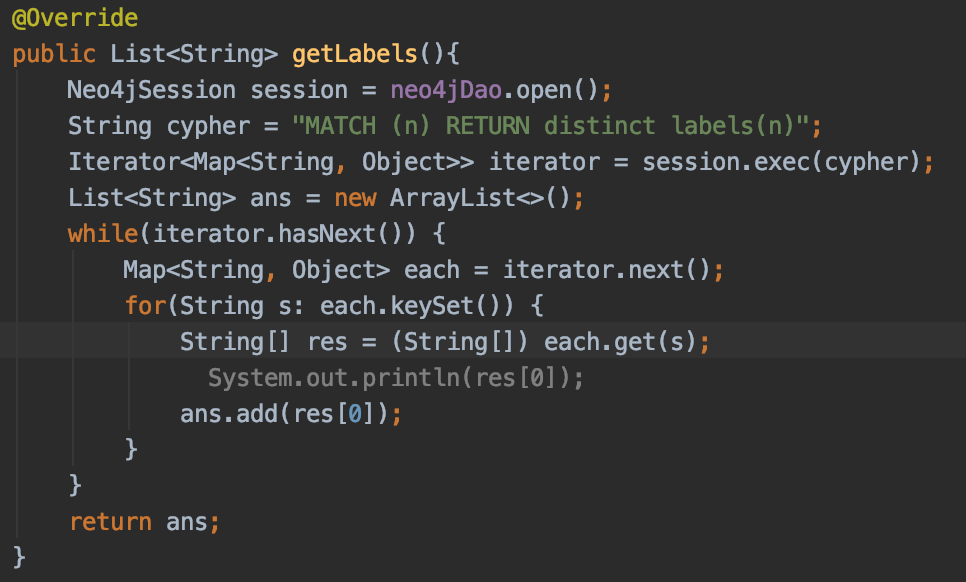
**Neo4jSession方式**





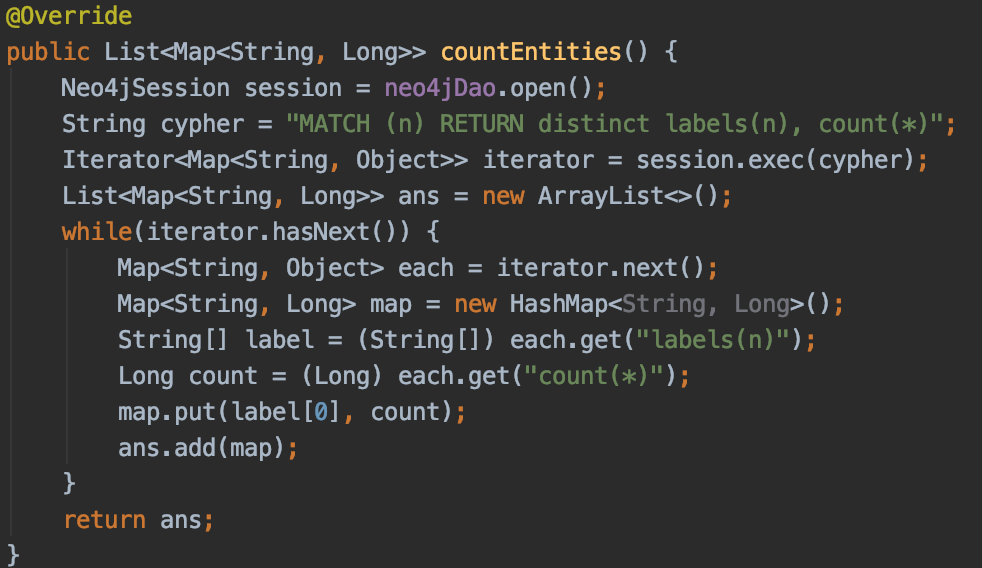


比如说我们想知道图数据库现在有多少个实体标签，通过cypher语言MATCH (n) RETURN distinct labels(n)就可以实现，但是结果并不是实体or关系。所以通过neo4jsession的形式来实现。





想要各类别的实体数量：



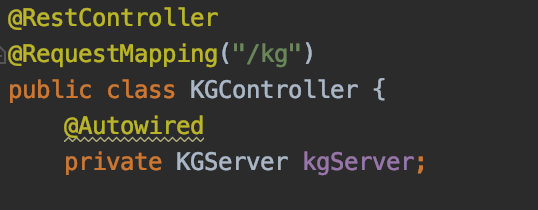


还有关系种类、个数的统计……

**业务层，负责业务逻辑实现，调用持久层的接口。**

****

**控制层，接受前端传来的数据，并返回业务层的数据。**

****

见具体展示。

## Cypher查询语言介绍

可参考资料：<https://www.w3cschool.cn/neo4j/neo4j_cql_introduction.html>

### Neo4j CQL - CREATE命令

* 创建没有属性的节点
* 使用属性创建节点
* 在没有属性的节点之间创建关系
* 使用属性创建节点之间的关系
* 为节点或关系创建单个或多个标签



### Neo4j CQL - MATCH命令

* 从数据库获取有关节点和属性的数据
* 从数据库获取有关节点，关系和属性的数据



### Neo4j CQL - RETURN子句

* 检索节点的某些属性
* 检索节点的所有属性
* 检索节点和关联关系的某些属性
* 检索节点和关联关系的所有属性

