

“轻言轻语” 社交平台说明

“轻言轻语”，一款轻量级的社交平台。它支持发布动态、动态评论，好友管理、好友推荐等基本功能。

项目地址：<http://demo.ljj.top:52001>

前端：https://github.com/hellolijj/neo4j_front

后端：<https://github.com/ccdongyu/neo4j-backend/>

Neo4j 数据库：<http://112.74.36.158:7474>

一、功能与演示

1.1 好友动态

首页面展示用户及用户好友的动态。可以对动态进行评论、点赞。



图 1-1 好友动态

1.2 动态评论

用户可以多可见的动态进行评论互动。



图 1-2 动态评论

1.3 我的好友

可以查看到我的好友



图 1-3 我的好友

1.4 好友推荐

好友推荐，系统会推荐可能认识的人（好友的好友）

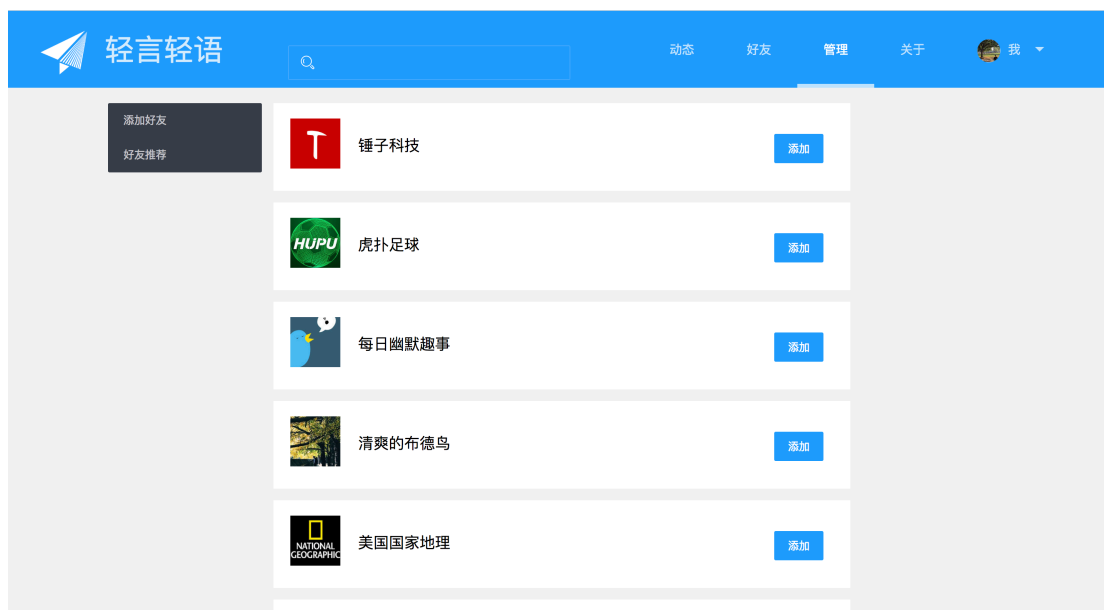


图 1-4 好友推荐

二、前端设计

2.1 功能模块

功能模块分为四大功能模块 11 个小功能



图 2-1 项目模块

2.2 页面设计

由于前端页面过多，此处只罗列首页和好友推荐两个页面的设计文稿。

首页动态列表



图 2-2 好友动态设计图

好友推荐

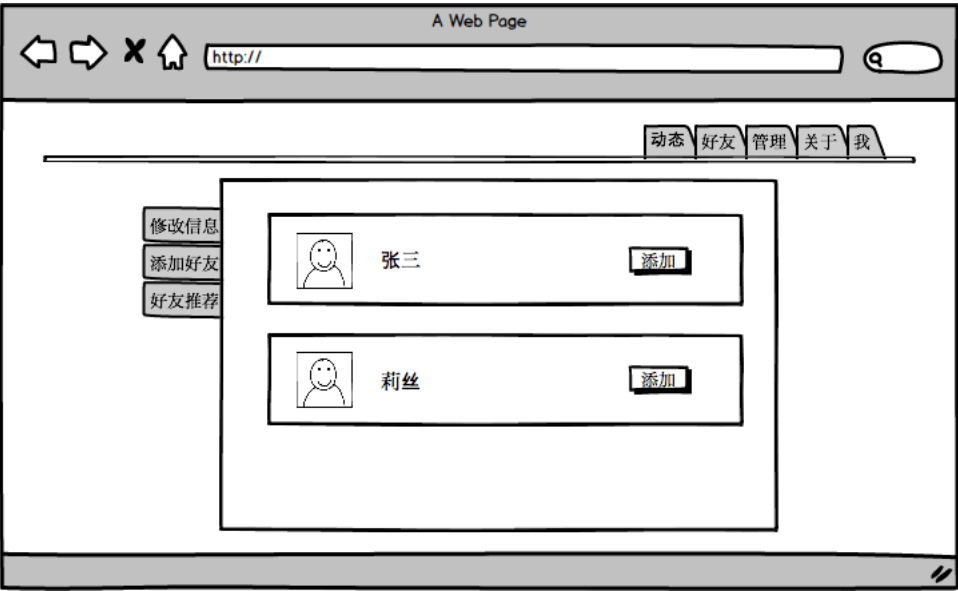


图 2-3 我的好友设计图

2.3 接口设计与说明

接口设计模块内容较多，本文档只罗列注册、登录模块的接口设计。本项目其他的接口设计文档详见：

https://github.com/hellolijj/neo4j_front/blob/master/README.md#%E6%8E%A5%E5%8F%A3%E6%96%87%E6%A1%A3

【登录模块】

GET: /User/login

request body:

```
{
  "userid": "hello_lijj@qq.com",
  "password": "123456",
}
```

field	is_require	type	description
userid	yes	string	用户登录账号
password	yes	string	用户密码

response 返回成功

```
{
  "status": 200
  "message": ""
  "time":
  "data": {
    "userid": "hello_lijj@qq.com",
    "avatar": ""
    "username": "软院小哥哥",
    "sex": "男",
    "desc": "我是软院的小哥哥"
  }
}
```

如果有错误，错误 message 将会提示用户

```
{
  "status": 405
  "message": "erron info"
}
```

【注册模块】

请求接口：

POST: /User/register

request body:

```
{
  "userid": "hello_lijj@qq.com",
  "password": "123456",
  "password_again": "123456",
  "username": "软院小哥哥",
  "sex": "男",
  "desc": "我是软院的小哥哥"
}
```

field	is_require	type	description
userid	yes	string	用户登录账号
password	yes	string	用户密码
password_again	yes	string	重复用户密码
username	yes	string	用户名，用于交友时显示的用户名称
sex		string	男 or 女
desc		string	个人说明

response 返回成功

```
{
  "status": 200
  "message": ""
}
```

如果有错误，错误 message 将会提示用户

```
{
  "status": 405
}
```

```
"message": "erron info"
}
```

三、后端设计

本小节主要阐述项目的后端设计情况,主要分为两部分:后端架构与目录结构两个部分。

3.1 后端架构

后端采用 Spring Boot 技术框架。如图 3-1 所示,本项目主要分为 5 层:控制层,服务层,仓储层,数据层以及领域层。

控制层:主要负责请求的处理。具体包括接收前端发来的请求,调用服务层处理请求然后向前端返回数据。

服务层:主要负责业务逻辑的实现。受控制层调用,处理相关业务逻辑。

仓储层:主要负责提供数据的访问接口。Spring Boot 依据 JPA (Java Persistence API) 相关规范,为我们实现相关的数据访问操作。

数据层:主要负责数据的存储。在本项目中,数据层主要指的是 Neo4j 图形数据库。

领域层:主要负责领域对象的建模。在本项目中,主要是指用户(Person),动态(Dynamic)以及评论(Comment)等对象的建模。

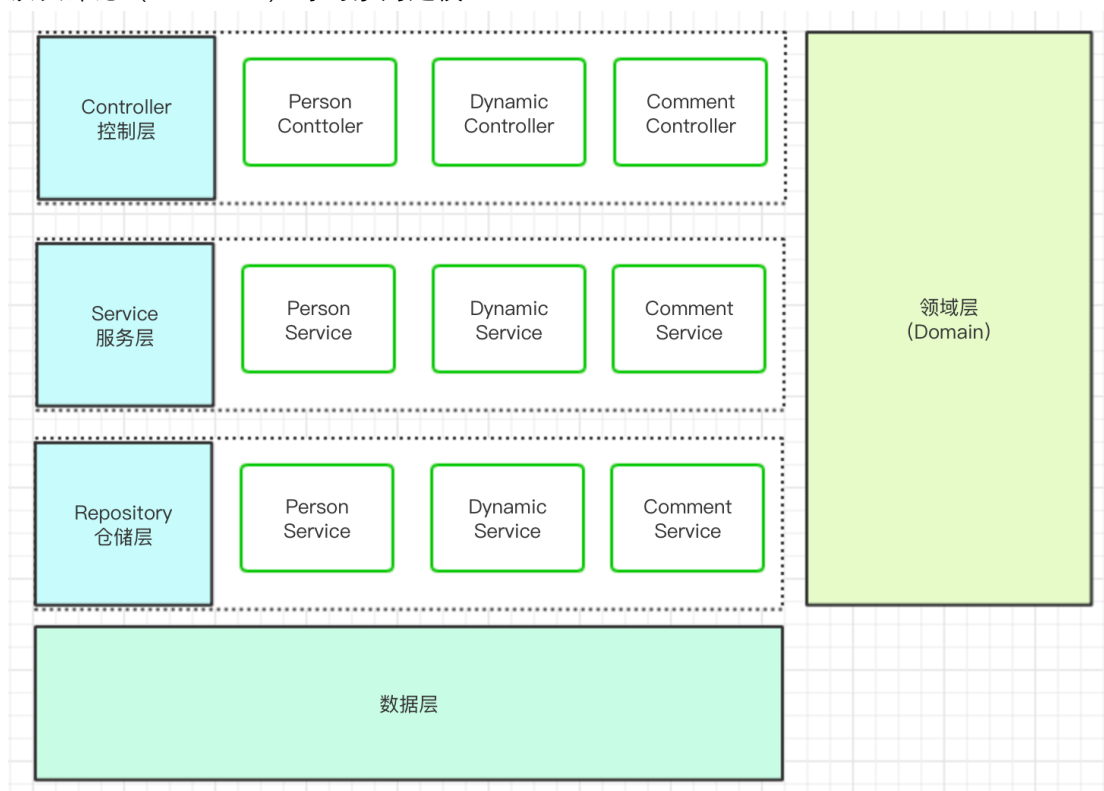


图 3-1 系统后端架构图

3.2 目录结构

后端目录结构如图 3-2 所示。

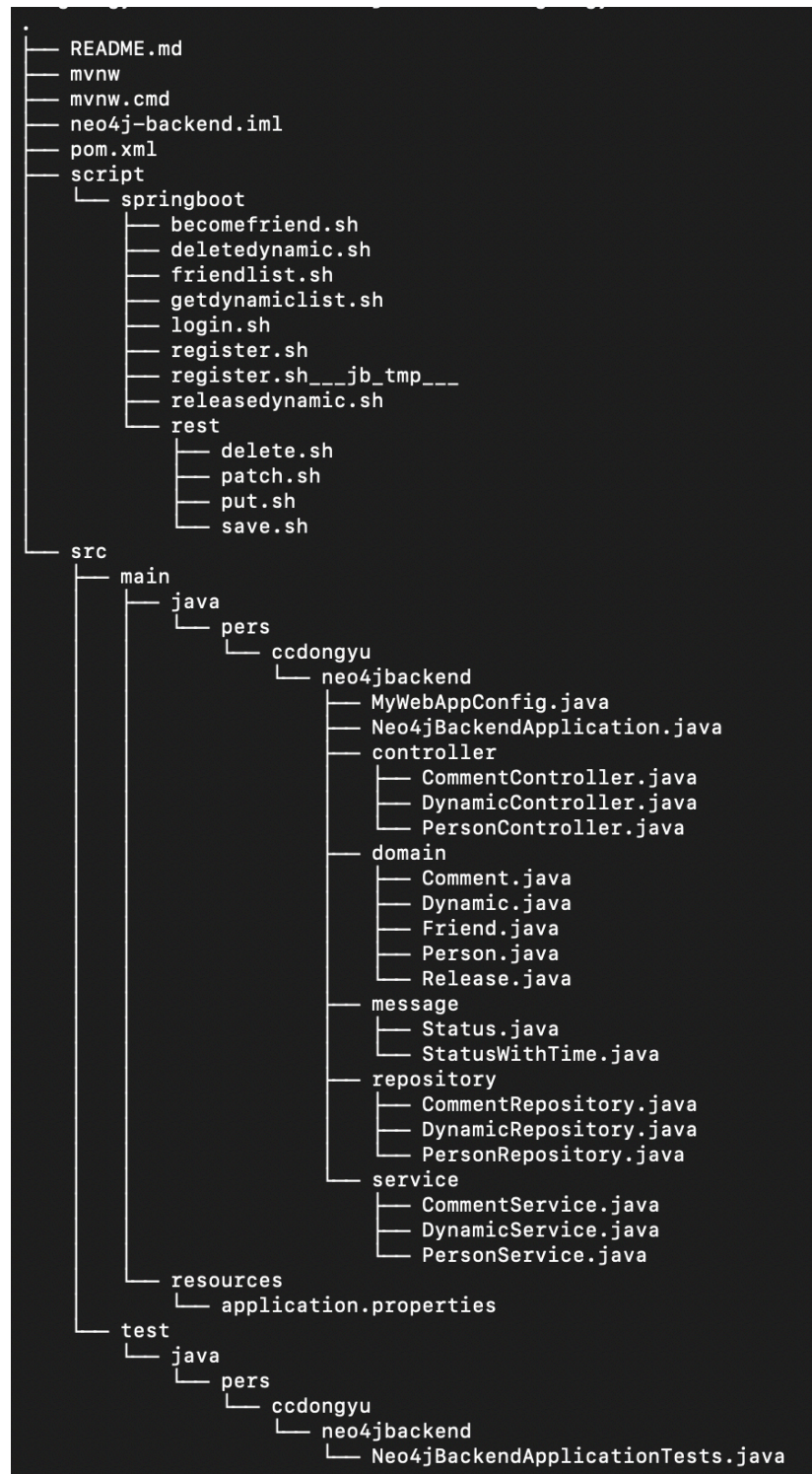


图 3-2 后端目录结构图

如图所示，除去相关的配置，项目主要可以分为两个部分：script 目录以及 src 目录。

Script 目录主要实现了一些基于 shell 的自动化测试脚本，测试的内容基本覆盖了对用户（Person）对象的相关操作。在 Script 目录下有一个名为 rest 的子目录，其中包括了对后端 REST 接口的相关访问测试。

Src 目录主要包含了本项目的相关源代码。src.main.java.pers.ccdongyu.neo4jbackend 路径下的就是本项目的主要源码。不难看出，项目的目录结构和项目的架构设计基本上是相互对应的。

四、数据库设计

4.1 设计理论步骤

需要说明的是，以下设计理论步骤，参考自 neo4j 官网推荐的由 Ian Robinson,Jim Webber 和 Emil Eifrem 共同编著的《Graph Database》。

在图数据库设计中，早期的分析节点，我们要做的工作和关系型的方法其实很相似：利用粗放的方法比如白板草图，来描述业务领域。在这之后，和关系型的方法开始有所不同。不再是将业务域类图模型转化成表，而是去丰富这个类图模型，使得它能够更好地表达我们想在业务中实现的业务域的相关显著特点。也就是说，对域中的每个实体，我们需要确保我们已经捕获了支持业务应用目标所需的属性和与相邻实体的连接。

接下来的图数据库专有设计时期，我们首先要做的是确保每一个节点有适当的参数，这样它可以履行其以数据为中心的特定的业务域职责。我们还需要通过在节点间建立命名的和有向的关系确保每一个节点有正确的语义上下文来描述业务域的结构。

首先，我们知道，我们的业务功能大致可以分成用户之间可以成为好友，用户可以发布动态，用户可以查看好友动态，用户针对动态和评论进行评论，用户可以点赞，根据我们的业务功能，动态、用户为图节点，用户节点和动态节点，用户节点与用户节点存在联系，我们用粗放的方法得到相关业务域类图模型如图 4-1：

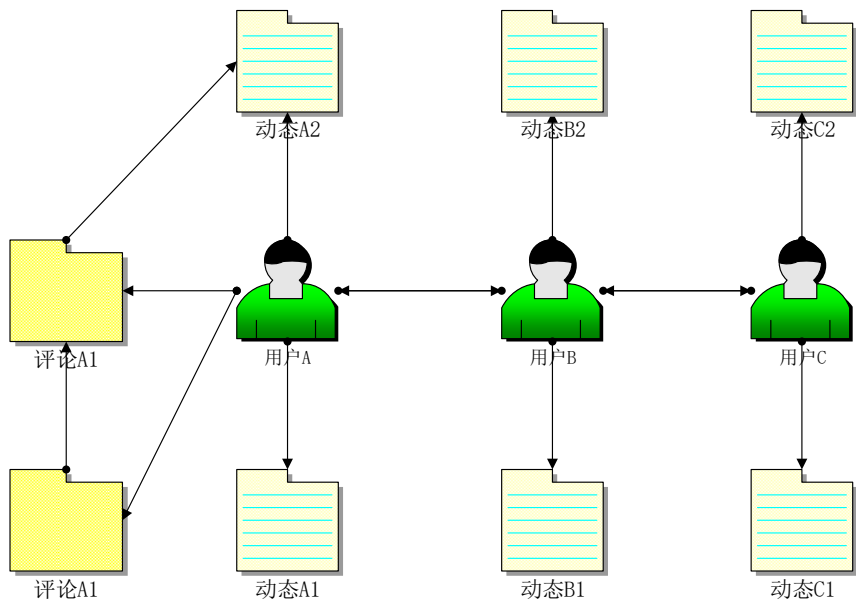


图 4-5 业务域类图模型

在之前的功能上继续丰富，考虑如果用户之间存在联系，则必然是好友联系，我们可以

用单向的好友关系,表示一个人想和另一个人成为好友,这在实际中可以体现成为好友请求,只有在被请求者也想和请求者建立联系,这时才建立好友关系。用户可以发布动态,发布可以是一个关系,动态是一个节点。发布关系中包含属性发布时间。结合用户的属性与节点的属性,我们得到的生成的图模型如图 4-2 :

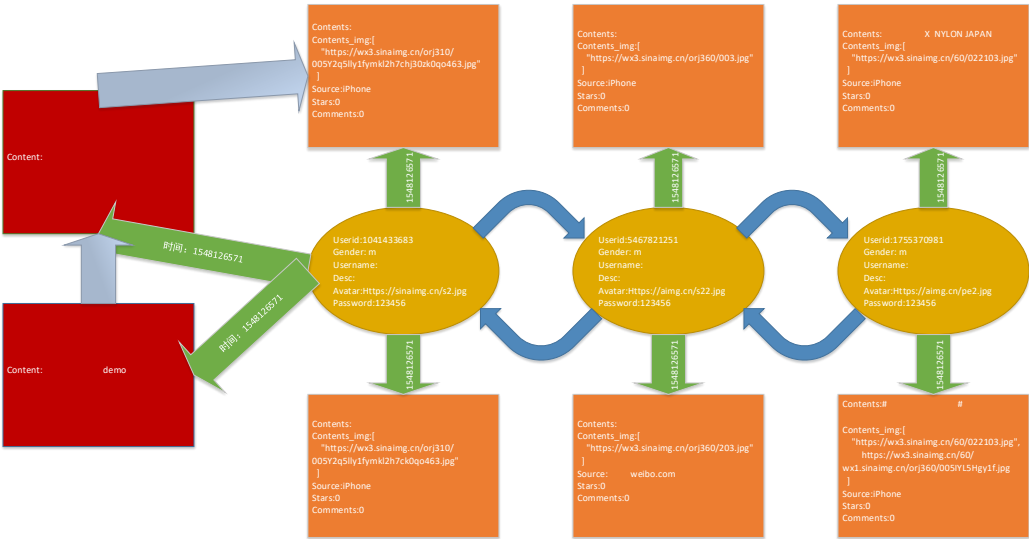


图 4-6 丰富后的图模型

4.2 数据库说明

4.2.1 节点说明

用户：

属性参数：

1. User_id:用户登录标识名
2. Username:用户昵称
3. sex:性别（“男” / “女”）
4. avatar:用户头像地址
5. Desc:用户个人简介
6. Password:密码

动态：

属性参数：

1. Contents:动态内容
2. Content_img:动态图片地址(没有则为空，有的话为字符串数组)
3. Source:动态来源
4. Comments:评论数
5. Stars:点赞数

评论：

属性参数：

1.content:评论内容

4.2.2.关系说明

好友关系（用户—用户）

属性参数：

1. creat_at:建立时间

发布关系（用户—动态/评论）

属性参数：

1. creat_at:发布时间

评论关系（评论—动态/评论）

属性参数：

无

4.3 数据库说明

我们在这里爬取微博 API 的微博内容，作为我们这里的动态节点内容，微博的用户(爬取时设置爬取数限制)作为我们这里的用户节点，为了方便，微博的 follow 关系在这里我们将这种单向的 follow 关系变成双向的 Friend 关系，并且为了去处重复，我们在用户和微博上都维护了一个 set，最终我们一共生成了 4705 个节点，其中 4425 个动态节点，279 个用户节点，创建的关系数 4815，其中 Friend 关系数 390，submit 关系数 4425 条。爬取过程中，我们想将数据存储在 MongoDB 中，MongoDB 中的 document 视图见图 4-3 和图 4-4：

```
{
  "_id": "1041433683",
  "avatar": "https://tva2.sinaimg.cn/crop.0.0.664.664.180/3e130453jw8f4f9ku5j20ig0igjs2.jpg",
  "cover": "https://tva4.sinaimg.cn/crop.0.0.640.640.640/6ce2240djwle81ktk4ohij20hs0hsmz6.jpg",
  "description": "生活真的很艰辛，你会失去爱人，你会经历心碎，有时你会生病，有时你会牙疼得要命，有时你会很饿。但另一方面，你会拥有最美的经历。",
  "fans_count": 185,
  "follows_count": 184,
  "gender": "m",
  "name": "布鲁克林的达芬奇先生",
  "verified": false,
  "verified_reason": null,
  "verified_type": -1,
  "weibos_count": 131,
  "fans": [
    {
      "_id": "6662807836",
      "name": "蔡爱_QiuLandym"
    },
    {
      "_id": "6884457332",
      "name": "情蛊入髓\_cg"
    },
    {
      "_id": "6244630438",
      "name": "云铃union"
    }
  ]
}
```

图 4-7 user collections 部分字段

```

{
  "_id" : "4318300940793449",
  "attitudes_count" : 1,
  "comments_count" : 0,
  "created_at" : "2018-12-17",
  "picture" : "http://wx4.sinaimg.cn/large/3e1304531y1fy9z392m6zj20k00zk41l.jpg",
  "pictures" : [
    {
      "pid" : "3e1304531y1fy9z392m6zj20k00zk41l",
      "url" : "https://wx4.sinaimg.cn/orj360/3e1304531y1fy9z392m6zj20k00zk41l.jpg",
      "size" : "orj360",
      "geo" : {
        "width" : 360,
        "height" : 640,
        "cropped" : false
      },
      "large" : {
        "size" : "large",
        "url" : "https://wx4.sinaimg.cn/large/3e1304531y1fy9z392m6zj20k00zk41l.jpg",
        "geo" : {
          "width" : 720,
          "height" : 1280,
          "cropped" : false
        }
      }
    }
  ]
}

```

图 4-8weibos 部分字段

然后针对 Mongdb 中字段的重命名与筛选以及对类似“7 小时前”的创建时间字段做解析转换操作，得到我们的数据源最后根据导入 Neo4j 数据库，Neo4j 部分数据截图如下：

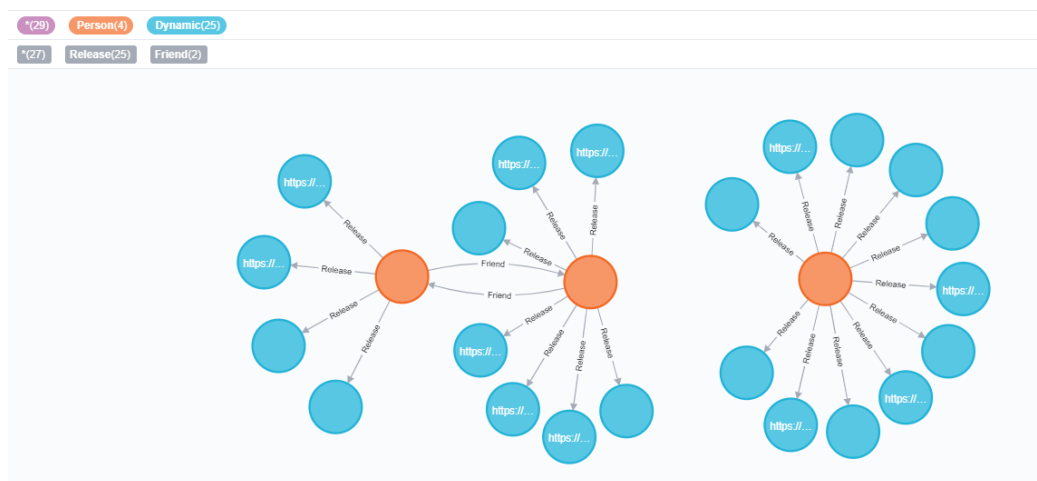


图 4-5 Neo4j 部分数据截图

五、项目分工

本项目由李俊君、邵迪龙、陈礼昌、唐冬雨、孙超、张雁宇、徐凌、鲁家南 8 位小伙伴合作共同完成。具体项目分工如下：

【前端】

21851172 李俊君：负责前端框架的构建，前端 模块的设计及实现，共同参与数据库设计

21851024 邵迪龙：负责好友关系模块的设计及实现，共同参与数据库设计

21851448 陈礼昌：负责登录注册模块的设计及实现，共同参与数据库设计

【后端】

21851034 唐冬雨：后端框架的搭建，好友管理模块的设计及实现，共同参与数据库设计

21851412 孙超：好友推荐模块的设计及实现，共同参与数据库设计

21851030 张雁宇：动态模块的设计及实现，共同参与数据库设计

21851460 徐凌：信息修改模块的设计及实现，共同参与数据库设计

【数据库】

21851154 鲁家南：微博数据的爬取，数据的转换，Neo4j 数据的导入，共同参与数据库设计