

Progress Report One

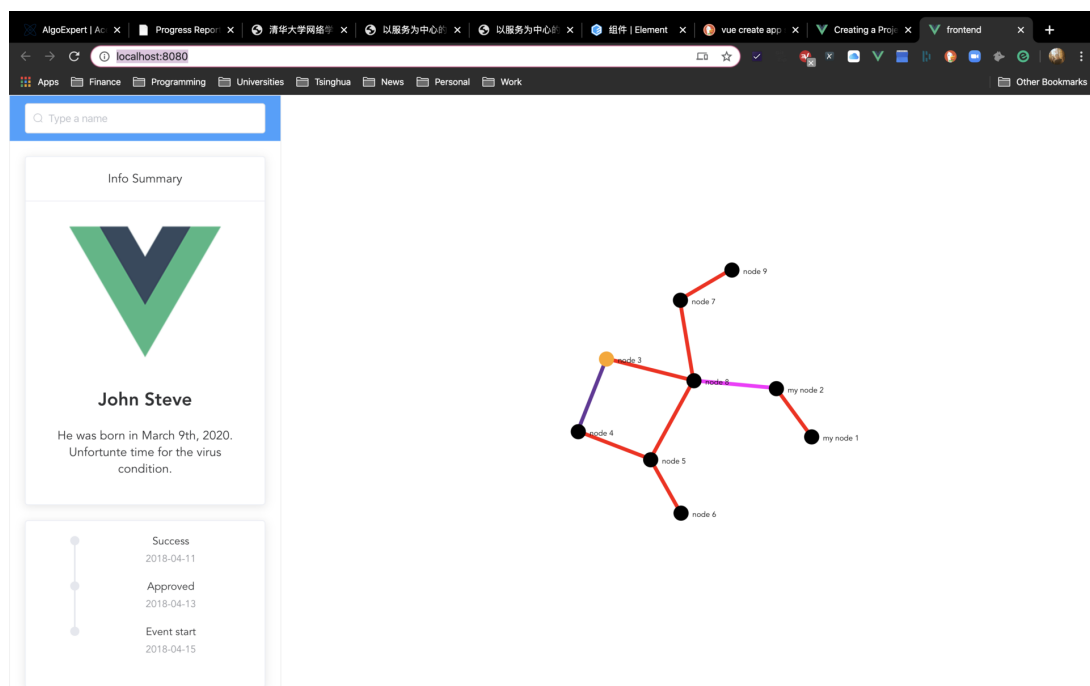
1 当前进展

前端

目标

前端根据计划，是要实现基础的交互功能，整理好少量的后端数据。

实际情况



- 前端基本布局已经实现
 - 左边Panel是显示节点数据，例如某个人的基本信息，照片；分成不同类别的信息以卡片的方式往下延伸、滑动。
- 可以显示后端数据，但还未连接
- 拓扑图可以实现：
 - 拖拽
 - 缩放（通过滚轮zoom）

为了能够在保证前端的性能的同时和给用户提供良好的交互性，使用了vue-d3-network第三方库，通过传入参数来绘制拓扑图。

- 与后端对接的api接口以实现，但出了一些CORS的问题，这个将在下周解决。

后端

根据计划，后端需要整理好小规模数据，并且提供数据获取的接口。目前计划基本完成。

实际情况

- 数据整理

为了找到比较符合我们项目需求的数据，我详细了解了<https://www.aminer.cn/data>网址内所提供的约13个开放数据集。目前选定**Topic-coauthor**数据集作为数据来源建立学术知识图谱，该数据集包含8个topic下的学者合著网络。

数据预处理方面，我先筛选出了**Data Mining** topic下的约20位学者及他们之间的合著关系作为小规模测试（具体数据见项目地址），目前已经整合进提供的接口。

- 数据获取接口

<http://101.200.241.8:3389/authors>

<http://101.200.241.8:3389/links>

以上两个接口分别用于访问所有的author和他们之间的关联关系，格式为**json**。

目前使用了阿里云的ECS服务器（试用期一个月）用作暂时的后端。框架为**Flask**，数据库为**sqlite**，使用**SQLAlchemy**来操作数据库。表单结构较为简单，只有作者信息和关联信息两张表，如下图：

Author表项包含**id**、**name**、**field**等信息

```
class Author(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(30))
    field = db.Column(db.String(40))
```

Link表项包含**id**、**name**、**tid** (target node id)、**sid** (source node id)、**field**属性。

```
class Link(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(20))
    tid = db.Column(db.Integer)
    sid = db.Column(db.Integer)
    field = db.Column(db.String(40))
```

2 苦难和阻碍

前端

1. 前端的知识拓扑图是使用的github上别人上传的npm库，一开始选用的库并不是特别完整。把它当component使用的时候，它的尺寸总是会当前网页的vh和vw，使得不能实现满屏布局，而是加多少component，网页就会被延长多少。后来换了一个更加完整的库之后就好了。但之前的问题整整花了我6-7个小时的时间，当然从中也学到了不少。
2. 不知道为什么，当我把页面里的某些部件当作部件import和直接在App.vue里面声明，结果是不一样的。就比如我在App.vue里面引用左边信息显示部件的时候，它的大小就是不受控制，后来换成直接在App.vue里面定义布局的时候，大小就可以控制了。所以

我认为肯定是我在App.vue里面引用的时候，size的传参有问题，但还是卡了很久。后来换成用element ui来布局，就解决了。

后端

- 1. 由于Topic-coauthor数据集给出的数据比较规整，所以处理起来没有遇到很大困难。部署的时候一开始想用Heroku，但是无奈访问太慢了，折腾了一会儿只好放弃。最后采用免费试用的阿里云服务器。
- 2. 还有一个问题是现在用的这个数据集有点陈旧，规模也比较小。按照助教的建议，可以自己根据学者的论文来计算其领域等相关信息，这点我做了一些尝试，但遇到了问题，下次讨论希望助教可以给一些指导。

3 之后的计划评估

首先，虽然转了很多弯子，到目前为进展还是很可观的，并没有落后于我们原有的计划。理想情况下，这周应该是实现了前后端的对接，但实际情况是还是差了一点，这个问题下周应该就能很快解决。

未来计划：（做了微调）

Breakpoint	Frontend	Backend
3th	学习d3js	开始整理数据
4th: Progress Report 1	1. 前端功能实现一部分 2. 未部署，能在本地查看效果	整理好一小部分的小规模数据
5th: Demo Deployed	1. 前端功能基本实现	1. 整理好一部分不同领域的 数据
	1. 部署上线 2. 前后对接	
6th & 7th	1. 完成剩下的前端功能 2. 优化前端展示	整理并加入各种领域的数据
8th: Midway Report & Demo	前端所有功能都实现 且完成度高	整理好所有的数据
9th: Peer Review	性能优化（暂定）	算法优化（暂定）
10th		
11th		
12th: Progress Report 2		
13th		
14th: 高可用负载均衡热更新		
15th		
16th: Final Presentation & Report	最后审核	最后审核

附录

项目地址: <https://git.tsinghua.edu.cn/cheng-zy17/storms/tree/master>