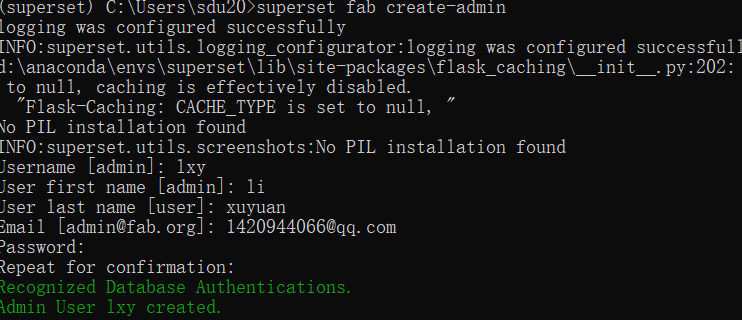
Pip使用国内镜像源并trust ：-i [http://pypi.douban.com/simple --trusted-host pypi.douban.com](http://pypi.douban.com/simple%20--trusted-host%20pypi.douban.com)

Superset启动：superset run

初始化数据库：superset db upgrade （在congif.py文件改过基本url后要执行一次）数据库（database）的名字不能带中文否则会报错

创建管理用户：最新版本的superset只能用 fab create-admin.不能用网上说的旧版的一大长串（要以官网为准）



初始化设置用户权限：superset init

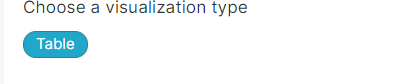
本文的superset的根目录是D:\Anaconda\envs\superset\Lib\site-packages\

创建一个图表，前端选择绿色按钮+Chart 请求url：<http://127.0.0.1:8088/chart/add>

调取static/assets/中的thumbnail\*\*\*\*.png图片作为展示供用户选择不同的Chart

（要学习flask.appbuilder）

1. 添加一个图表首先添加示意图到： superset\static\assets\images\viz\_thumbnails

添加的图片将要显示在

2.

重新进行源码安装：参考：https://blog.csdn.net/qq8241994/article/details/108083990

1. 下载源码，新建anaconda环境并把源码解压后拖到site-packages里面进入superset下面的目录用pip install -r requirements.txt安装需要的模块（cd 进入与当前同级的目录要加/d）。
2. 然后安装 requirements-dev.txt开发依赖模块
3. 先更改coonfig.py文件中的database\_url
4. 再初始化数据库：进行如下操作：



1. 连接数据库的时候url要加上charset=utf8不然下载示例的时候会报错

这个数据保存在连接的数据库的ab\_user表中

Username：lxy

Password：asd134679852

1. 前端打包：

// 进入文件夹

cd superset-frontend

// 安装依赖包

npm install （只有第一次才执行这条语句）

//打包前端给后端服务器

npm run dev（每次修改前端文件都要重新打包）

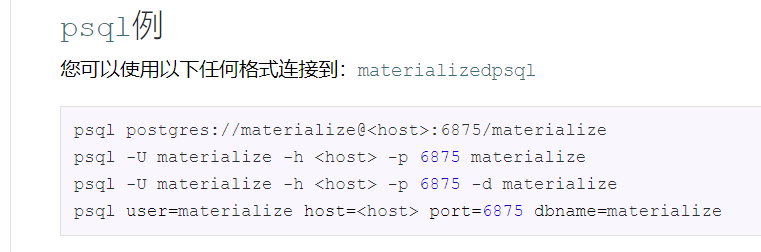
7 运行直接flask run

1. viz.py 这个文件起到一个类似于视图的功能，从前端接受请求后进行一系列处理。一个图例就是一个类。经过分析已有的这些类，这些类大部分是继承了BaseViz这个基类，并且主要重写了query\_obj和get\_data这两个方法。其中query\_obj，是用来构造查询条件的。而查询条件来源于页面左侧的组件，这些组件的信息存储在form\_data的一个字典中，query\_obj方法中将会处理form\_data，构造除自己需要的查询条件并返回，在BaseViz这个基类中，会根据这个查询条件来查询出数据。然后是get\_data方法，get\_data接收一个参数df，也就是query\_obj之后查询出来的数据，也可以结合自己构造的form\_data中的一些条件对df进行处理。处理之后的数据以dict格式返回，这个dict会交给js进行下一步处理，
2. 自定义的类还应该有两个重要的类属性viz\_type和verbose\_name。其中viz\_type是标识当前类，也就是我们自定义图例的名称，注意这个名称应该与后面的js也结合起来。verbose\_name则是相当于别名，展示在前端的。

**流数据库探索**

Druid数据库datasource的name不要出现“-”否则会导致查询报错！！！

Materialized使用psql链接



远程登录本机mysqlIP地址以及用户密码