总结、分享、收获

实验室主页 (//dblab.xmu.edu.cn)

Spark+Kafka构建实时分析Dashboard案例（2022年9月V2.0版）——步骤一：实验环境准备

刘, 浩然 2022年9月3日 (updated: 2022年9月24日) 31000



(//dblab.xmu.edu.cn/post/bigdataroadmap/)

返回本案例首页 (http://dblab.xmu.edu.cn/post/spark-kafka-dashboard/)

《Spark+Kafka构建实时分析Dashboard案例——步骤一：实验环境准备》

开发团队：厦门大学数据库实验室 联系人：林子雨老师 ziyulin@xmu.edu.cn

版权声明：版权归厦门大学数据库实验室所有，请勿用于商业用途；未经授权，其他网站请勿转载

本教程介绍大数据课程实验案例“Spark+Kafka构建实时分析Dashboard案例”的第一个步骤，实验环境准备工作，有些软件的安装在相应的章节还会介绍。

预备知识

Linux系统命令使用、了解如何安装Python库。

训练技能

熟悉Linux基本操作、Pycharm的安装、Spark安装，Kafka安装，PyCharm安装。

任务清单

1. Spark安装
2. Kafka安装
3. Python安装
4. Python依赖库
5. PyCharm安装

实验系统和软件要求

```
Ubuntu: 16.04
Spark: 3.2.0
kafka: 2.8.0
Python: 3.7.13
Flask: 1.1.2
Flask-SocketIO: 5.1.0
kafka-python:  2.0.2
```

系统和软件的安装

Spark安装

Spark的安装可以参考Spark系列教程，地址为Spark2.1.0入门：Spark的安装和使用 (https://dblab.xmu.edu.cn/blog/1307-2/)；

Kafka安装

kafka的安装可以参考博客Kafka的安装和简单实例测试 (https://dblab.xmu.edu.cn/blog/1096-2/)；

Python安装

Ubuntu16.04系统自带Python2.7和Python3.5，本案例使用Anaconda中创建的python3.7.13环境，Anaconda的安装使用可以参考博客Anaconda的下载和使用方法 (https://dblab.xmu.edu.cn/blog/2853/)；

Python依赖库

本案例主要使用了两个Python库，Flask和Flask-SocketIO，这两个库的安装非常简单，请启动进入Ubuntu系统，打开一个命令行终端（可以使用快捷键Ctrl+Alt+T），然后开启Anaconda环境：

Shell 命令

```
$ conda activate env_name #这里env_name应替换为创建的conda环境名称
```

Anaconda之所以强大，其中一个原因是其便捷的第三方库管理。可以使用如下Shell命令完成Flask和Flask-SocketIO这两个Python第三方库的安装以及与Kafka相关的Python库的安装：

Shell 命令

```
$ conda install flask
$ conda install flask-socketio
$ conda install kafka-python
```

这些安装好的库在我们的程序文件的开头可以直接用来引用。比如下面的例子。

Python

```
1. from flask import Flask
2. from flask_socketio import SocketIO
3. from kafka import KafkaConsumer
```

from import 跟直接import的区别举个例子来说明。

import socket的话,要用socket.AF_INET,因为AF_INET这个值在socket的名称空间下。

from socket import* 是把socket下的所有名字引入当前名称空间。

PyCharm安装

PyCharm是一款Python开发IDE，可以极大方便工程管理以及程序开发。请在Ubuntu系统中打开自带的火狐浏览器，前往PyCharm官网(https://www.jetbrains.com/pycharm/download/#section=linux)下载免费的Community版本，下载后默认会被保存到当前登录用户的主目录下的“下载”目录中。比如，如果当前使用hadoop用户名登录了Ubuntu系统，那么，就会被保存在“/home/hadoop/下载”这个目录下。然后执行如下命令对安装文件进行解压缩：

Shell 命令

```
$ cd ~  #进入当前hadoop用户的主目录
$ sudo tar -zxvf ~/下载/pycharm-community-2016.3.2.tar.gz -C /usr/local  #把pycharm解压缩到/usr/local目录下
$ cd /usr/local
$ sudo mv pycharm-community-2016.3.2 pycharm  #重命名
$ sudo chown -R hadoop ./pycharm  #把pycharm目录权限赋予给当前登录Ubuntu系统的hadoop用户
```

然后，执行如下命令启动PyCharm：

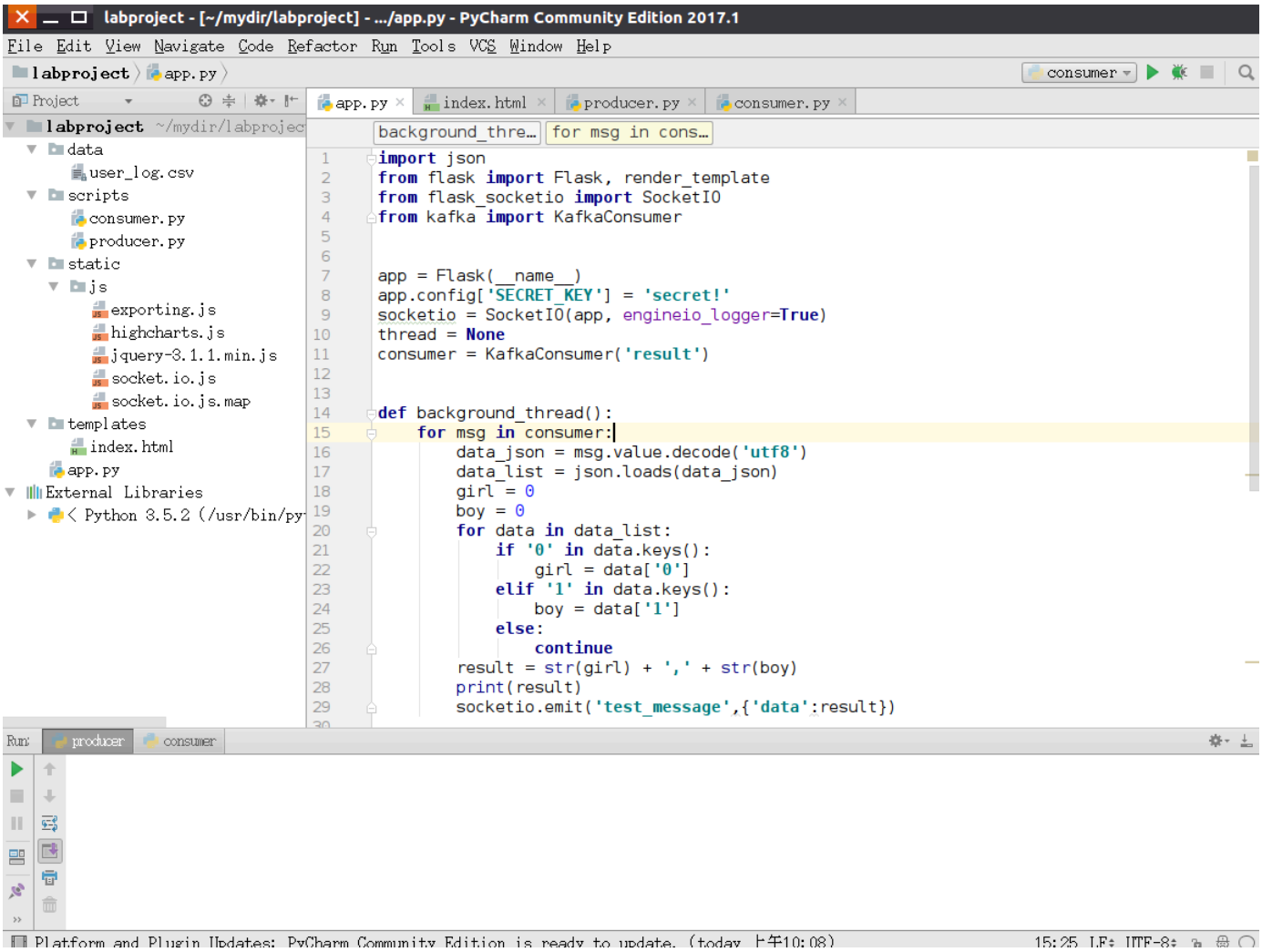
Shell 命令

```
$ cd /usr/local/pycharm
$ ./bin/pycharm.sh  #启动PyCharm
```

执行上述命令之后，即可开启PyCharm。

Python工程目录结构

这里先给出本案例Python工程的目录结构，后续的操作可以根据这个目录进行操作。

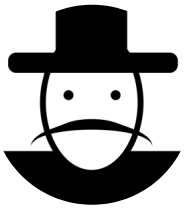


1. data目录存放的是用户日志数据；
2. scripts目录存放的是Kafka生产者和消费者；
3. static/js目录存放的是前端所需要的js框架；
4. templates目录存放的是html页面；
5. app.py为web服务器，接收Structed Streaming处理后的结果，并推送实时数据给浏览器；
6. External Libraries是本项目所依赖的Python库，是PyCharm自动生成。

至此，本案例需要的开发环境就介绍完毕，顺带说一句，Spark自带Scala，因此如果是开发Spark应用程序，则没必要单独安装Scala。

下篇文章链接为Spark+Kafka构建实时分析Dashboard案例——步骤二：数据处理和Python操作Kafka (https://dblab.xmu.edu.cn/blog/2022090001-4/)

本文作者



刘, 浩然 (//dblab.xmu.edu.cn/blog/author/liuhaoran/)

1609126475@qq.com

https://dblab.xmu.edu.cn/blog/3212/ (https://dblab.xmu.edu.cn/blog/3212/)

大数据 (https://dblab.xmu.edu.cn/blog/category/big-data/)

© 2014 厦大数据库实验室 (//dblab.xmu.edu.cn)