**教务在线爬取文档**

一、基本准备工作

1、操作系统：ubuntu16.04LTS（u盘装系统）

2、操作平台：ubuntu终端

3、scrapy框架搭建

4、步骤：

a、ubuntu16.04LTS自带了python2.7,lxml，openssl

b、按照网上的方法，一共有3个方法安装scrapy，选取pip安装

c、在终端里输入：

sudo apt-get install python-pip

sudo apt-get install python-dev

sudo apt-get install libevent-dev

（注：不分顺序）

最后输入：

sudo pip install scrapy（注：都为下载命令）

5、检测

输入： scrapy version

（注：如果不报错则是下载完成。）

6、问题

scrapy是否适用于爬虫所有的情况，scrapy里其他文件的具体用法

二、爬取教务在线

1、操作系统：ubuntu16.04LTS

2、目标网站：重庆邮电大学教务在线网站

http：//jwzx.cqupt.edu.cn (内网）

3、目标：抓取全校学生名单信息：学生姓名、年级、专业、班级

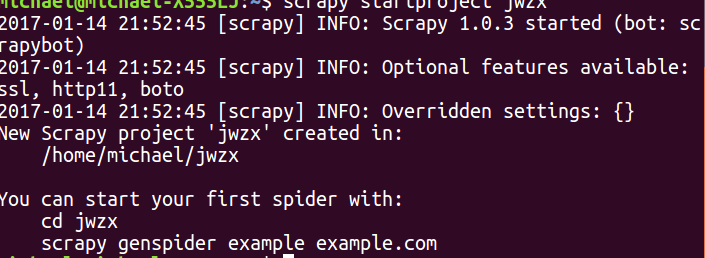
4、步骤：

a、打开终端，输入：

scrapy startproject jwzx

（注：jwzx属于工程名字，我原来写的工程名是start，创建一个爬虫项目）

就创建好了一个工程，会获得如下提醒

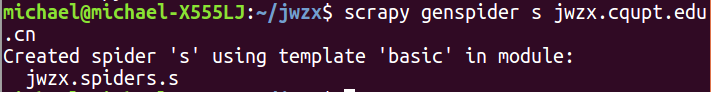


根据提醒可以知道，可用

cd jwzx

scrapy genspider jwzx jwzx.cqupt.edu.cn

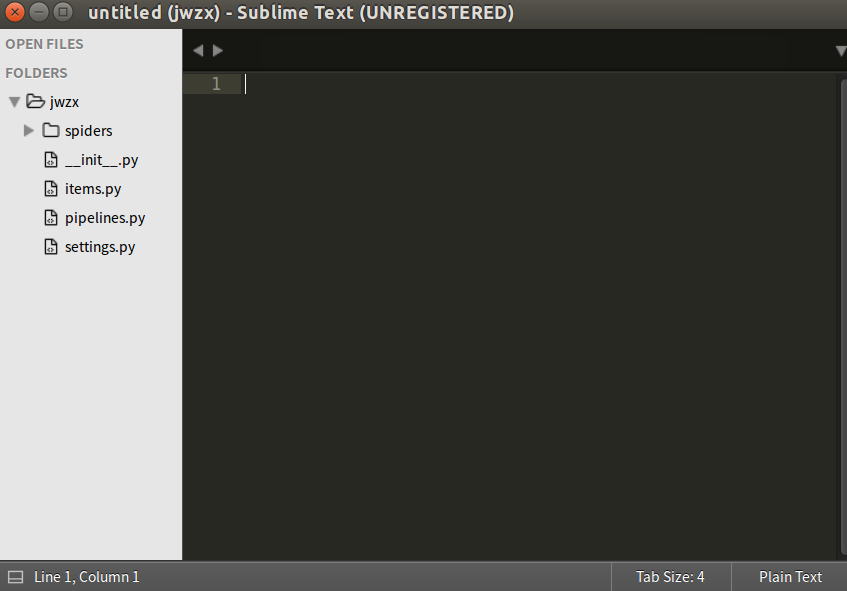
来创建一个新的爬虫



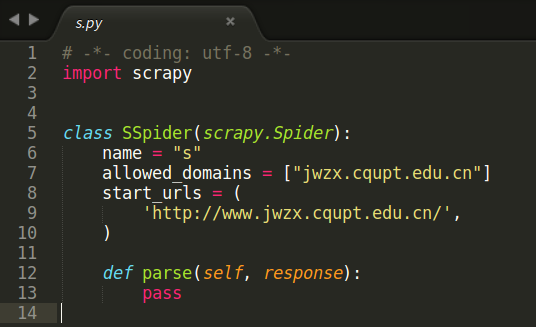
注意：工程名和爬虫名可以相同，但他们不是一样的，一个工程可以有多个爬虫

为区分，创建的爬虫名是s

b、打开sublime ，在file里面 ctrl+o ，再把刚才创建的jwzx工程文件夹打开，会出现：

调节sublime界面的右下角的 Plain Text ，选择Python

c、打开 s.py ，会看见这样的



属性有：爬虫工程名字：s

允许爬取的域名：allowed\_domains：["jwzx.cqupt.edu.cn"]

开始的首页网址：start\_urls(注：这个要修改成学生名单的网址)

['http://jwzx.cqupt.edu.cn/jwzxtmp/pubBjsearch.php?action=bjStu']

在items.py里面修改自己需要的类的元素，如名字，name



然后在s.py里面添加引用

from jwzx.items import JwzxItem

from scrapy.http import Request

import pdb

注：第一个的作用是引用自己添加的类的元素

第二个的作用是引用scrapy的request，因为要用到的xpath来抓取数据，也可以用正则表达式，(注：其中response是对象作为唯一参数传递给函数,提取数据,生成item)

第三个的作用是引用断点（注：不需要断点检测的就不需要引用）

d、去掉pass，打开教务在线的学生名单界面，并 F12。观察网址规律

结论：每一个班级的网址前面都是一样的，为 <http://jwzx.cqupt.edu.cn/jwzxtmp/>

而后缀部份各不相同，所以用拼接的方法来完成



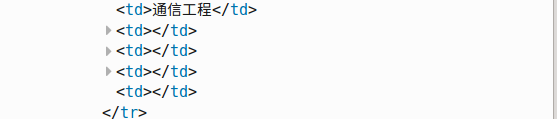












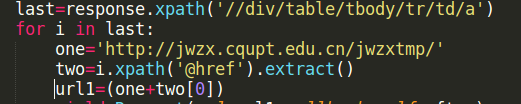


依次点击截图值中的箭头，就会出现所需要的网址后缀，现在可以确定这些网址后缀的路径：

//div/table/tbody/tr/td/a

注：前面的 // 是代表全部选取，并且省略了body的路径

利用以下代码来拼接：



解释：第一句是利用xpath语句（引用的response里面的）对刚才找到的网址后缀路径进行全部提取

第二句是for循环，对截取的网址后缀的href赋值给two，@表示锁定

for循环里面的url1就是拼接，组成一个完整的网址

问题：

extract（）在使用xpath是有时候用，有时候不用，截取数据保留的时候用。目前我的理解是在返回一个list时使用

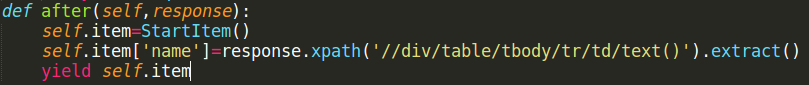
e、返回网址，和进入下一页(注：进入抓取的url,进入next函数,采用回调函数callback赋值给request)：



after是接下来的函数，因为刚才的网址只是把全校的班级取出来了，现在要进去每一个班级里面取提取每一个同学的信息

f、进入下一页函数（after）

其中：self.item=StartItem()为连接item.py文件

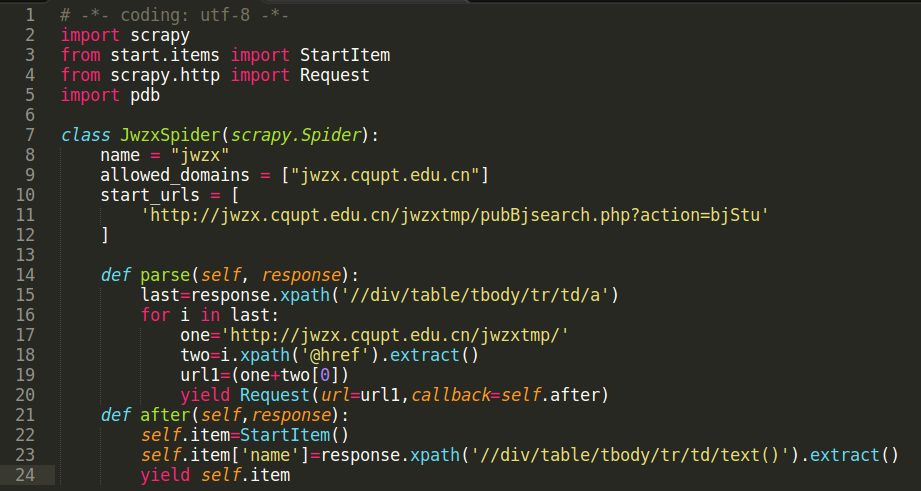


步骤和上面一样，就是先找到要提取的东西的路径



确认路径后，用xpath提取，因为提取的是后面的文档，所以加text（），最后返回提取的东西（注意：设置断点看 haha/B/url1/self.item['name']中有没有数据）

全部程序：



f:保存数据: 进入项目的根目录执行终端中输入scrapy crawl jwzx -o items.json(注：采用json格式对爬取的数据进行序列化,生成items.json文件)