Scrapy 爬取实验楼用户数据

简介

本节内容运用前俩节学到的知识，爬取实验楼的用户数据，主要是为了练习、巩固前面学习到了知识。

知识点

Scrapy 项目框架

分析网页元素字段

SQLAlchemy 定义数据模型

创建 Item

解析数据

要爬取的内容

下面是一个用户主页的截图，箭头指的是我们要爬取的内容：

要爬取的内容和字段名称定义：

用户名（name）

类型：普通用户／会员／高级会员（type）

加入实验楼的时间（join\_date）

楼层数（level）

状态：在职／学生（status）

学校（如果用户是学生）（school）

职位（如果用户在职）(job）

学习记录（study\_record）

定义数据模型

决定好了要爬取的内容，就可以使用 SQLAlchemy 定义数据模型了，在 models.py 中的 Course 后面定义 User 模型：

# User 表用到新类型要引入

from sqlalchemy import Date

class User(Base):

\_\_tablename\_\_ = 'users'

id = Column(Integer, primary\_key=True)

name = Column(String(64), index=True)

type = Column(String(64))

status = Column(String(64), index=True)

school = Column(String(64))

job = Column(String(64))

level = Column(Integer, index=True)

join\_date = Column(Date)

learn\_courses\_num = Column(Integer)

现在可以运行程序创建 users 表了：

python3 models.py

SQLAlchemy 默认不会重新创建已经存在的表，所以不用担心 create\_all 会重现创建 couses 表造成数据丢失。

定义数据模型操作视频：

您还没有安装flash播放器，请点击这里安装

安装后重启浏览器即可播放视频

Play Video

创建 Item

在 items.py 中创建 UserItem，为每个要爬取的字段声明一个 Field。

class UserItem(scrapy.Item):

name = scrapy.Field()

type = scrapy.Field()

status = scrapy.Field()

job = scrapy.Field()

school = scrapy.Field()

level = scrapy.Field()

join\_date = scrapy.Field()

learn\_courses\_num = scrapy.Field()

创建爬虫

使用 genspider 命令创建 users 爬虫：

scrapy genspider users shiyanlou.com

scrapy 为我们在 spiders 下面创建 users.py 爬虫，将它修改如下：

import scrapy

class UsersSpider(scrapy.Spider):

name = 'users'

@property

def start\_urls(self):

""" 实验楼注册的用户数目前大约50几万，为了

爬虫的效率，取 id 在 524,800~525,000 之间的

新用户，每间隔 10 取一个，最后大概爬取 20 个

用户的数据

"""

return ('https://www.shiyanlou.com/user/{}/'.format(i) for i in range(525000, 524800, -10))

爬虫及Item实现视频：

您还没有安装flash播放器，请点击这里安装

安装后重启浏览器即可播放视频

Play Video

解析数据

解析数据主要是编写 parse 函数。在实际编写前，最好是用 scrapy shell 对某一个用户先测，将正确的提取代码复制到 parse 函数中。

下面的几个例子是要提取的数据在文档中的结构和对应的提取器（提取器有很多种写法，可以写你自己的）

name:

# response.css('span.username::text')

<span class="username">aiden0z</span>

type:

# response.css('a.member-icon img.user-icon::attr(title)').extract\_first(default='普通用户')

# 用户头像右下角有一个会员标志，用它可以判断用户类型，对于会员用户

# 可以从会员图标图片的 title 获取 type

# 非会员用户没用会员图标，可以为提取器返回一个默认值表示

<a class="member-icon" href="/vip" target="\_blank">

<img class="user-icon" src="

https://static.shiyanlou.com/img/plus-vip-icon.png

" title="标准会员">

</a>

join\_date：

# response.css('span.join-date::text').extract\_first()

# 这样提取的值是 “2014-08-27 加入实验楼”，需要在 pipeline 中将

# 它处理为 date 对象

<span class="join-date">2014-08-27 加入实验楼</span>

依次在 scrapy shell 测试每个要爬取的数据，最后将代码整合进 users.py 中如下：

# -\*- coding: utf-8 -\*-

import scrapy

from shiyanlou.items import UserItem

class UsersSpider(scrapy.Spider):

name = 'users'

start\_urls = ['']

@property

def start\_urls(self):

return ('https://www.shiyanlou.com/user/{}/'.format(i) for i in range(525000, 524800, -10))

def parse(self, response):

yield UserItem({

'name': response.css('span.username::text').extract\_first(),

'type': response.css('a.member-icon img.user-icon::attr(title)').extract\_first(default='普通用户'),

'status': response.xpath('//div[@class="userinfo-banner-status"]/span[1]/text()').extract\_first(),

'job': response.xpath('//div[@class="userinfo-banner-status"]/span[2]/text()').extract\_first(),

'school': response.xpath('//div[@class="userinfo-banner-status"]/a/text()').extract\_first(),

'join\_date': response.css('span.join-date::text').extract\_first(),

'level': response.css('span.user-level::text').extract\_first(),

'learn\_courses\_num': response.css('span.latest-learn-num::text').extract\_first()

})

数据解析操作视频：

您还没有安装flash播放器，请点击这里安装

安装后重启浏览器即可播放视频

Play Video

pipeline

因为 pipeline 会作用在每个 item 上，当和课程爬虫共存时候，需要根据 item 类型使用不同的处理函数。

from datetime import datetime

from sqlalchemy.orm import sessionmaker

from shiyanlou.models import Course, User, engine

from shiyanlou.items import CourseItem, UserItem

class ShiyanlouPipeline(object):

def process\_item(self, item, spider):

""" 对不同的 item 使用不同的处理函数

"""

if isinstance(item, CourseItem):

self.\_process\_course\_item(item)

else:

self.\_process\_user\_item(item)

return item

def \_process\_course\_item(self, item):

item['students'] = int(item['students'])

self.session.add(Course(\*\*item))

def \_process\_user\_item(self, item):

# 抓取到的数据类似 'L100'，需要去掉 'L' 然后转化为 int

item['level'] = int(item['level'][1:])

# 抓去到的数据类似 '2017-01-01 加入实验楼'

# 其中的把日期字符串转换为 date 对象

item['join\_date'] = datetime.strptime(item['join\_date'].split()[0], '%Y-%m-%d').date()

# 学习课程数目转化为 int

item['learn\_courses\_num'] = int(item['learn\_courses\_num'])

# 添加到 session

self.session.add(User(\*\*item))

def open\_spider(self, spider):

Session = sessionmaker(bind=engine)

self.session = Session()

def close\_spider(self, spider):

self.session.commit()

self.session.close()

pipeline 实现视频：

您还没有安装flash播放器，请点击这里安装

安装后重启浏览器即可播放视频

Play Video

运行

使用 `crawl｀ 命令启动爬虫：

scrapy crawl users

爬虫运行及结果查看视频：

您还没有安装flash播放器，请点击这里安装

安装后重启浏览器即可播放视频

Play Video

总结

实验设计了一个新的实例，爬取实验楼的用户页面，在这个页面中首先需要分析页面中的各种元素，从而设计爬虫中数据提取的方式。然后把需要的数据内容通过 Scrapy 项目中的代码获取得到并解析出来，存储到数据库中。

本节实验包含以下的知识点：

Scrapy 项目框架

分析网页元素字段

SQLAlchemy 定义数据模型

创建 Item

解析数据

来源: 实验楼

链接: https://www.shiyanlou.com/courses/996

本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播