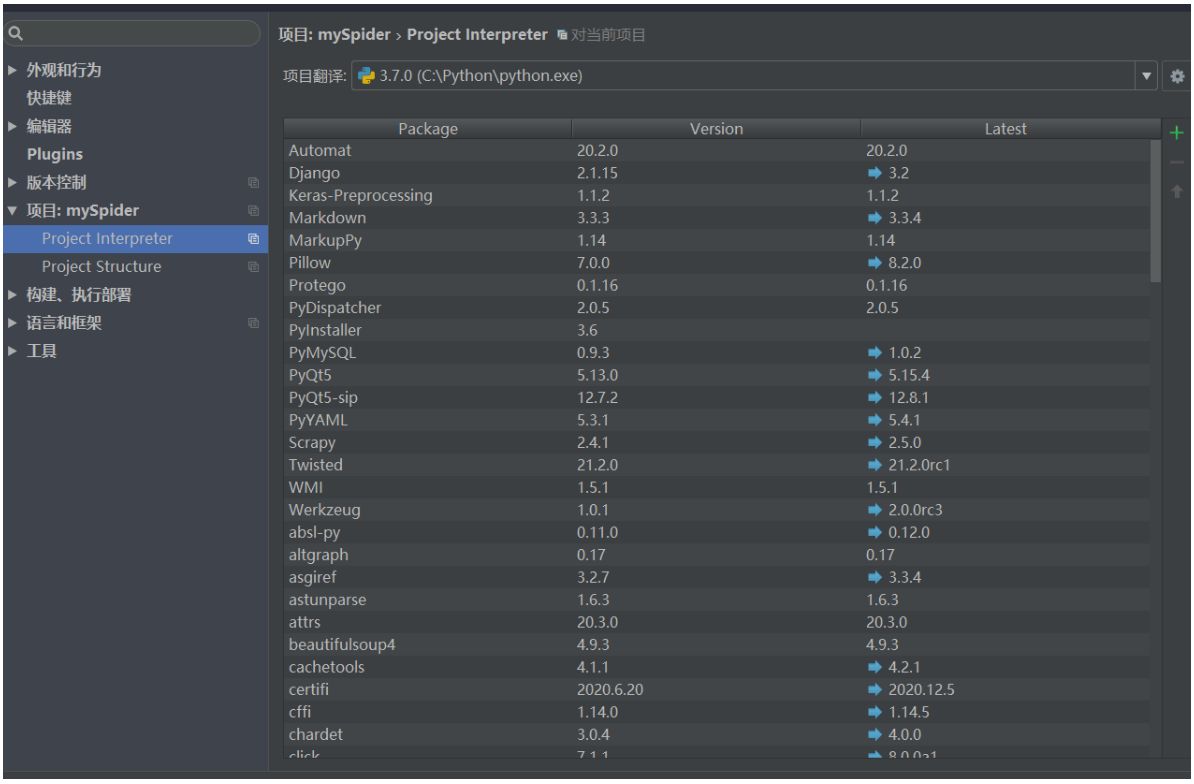
scrapy环境搭建

环境:python3+pycharm

1.

安装scrapy

打开pycharm,工具栏File/settings,在项目(Project,我的是中英一半翻译,表述可能不准确)中找到project interpreter,此时界面如下

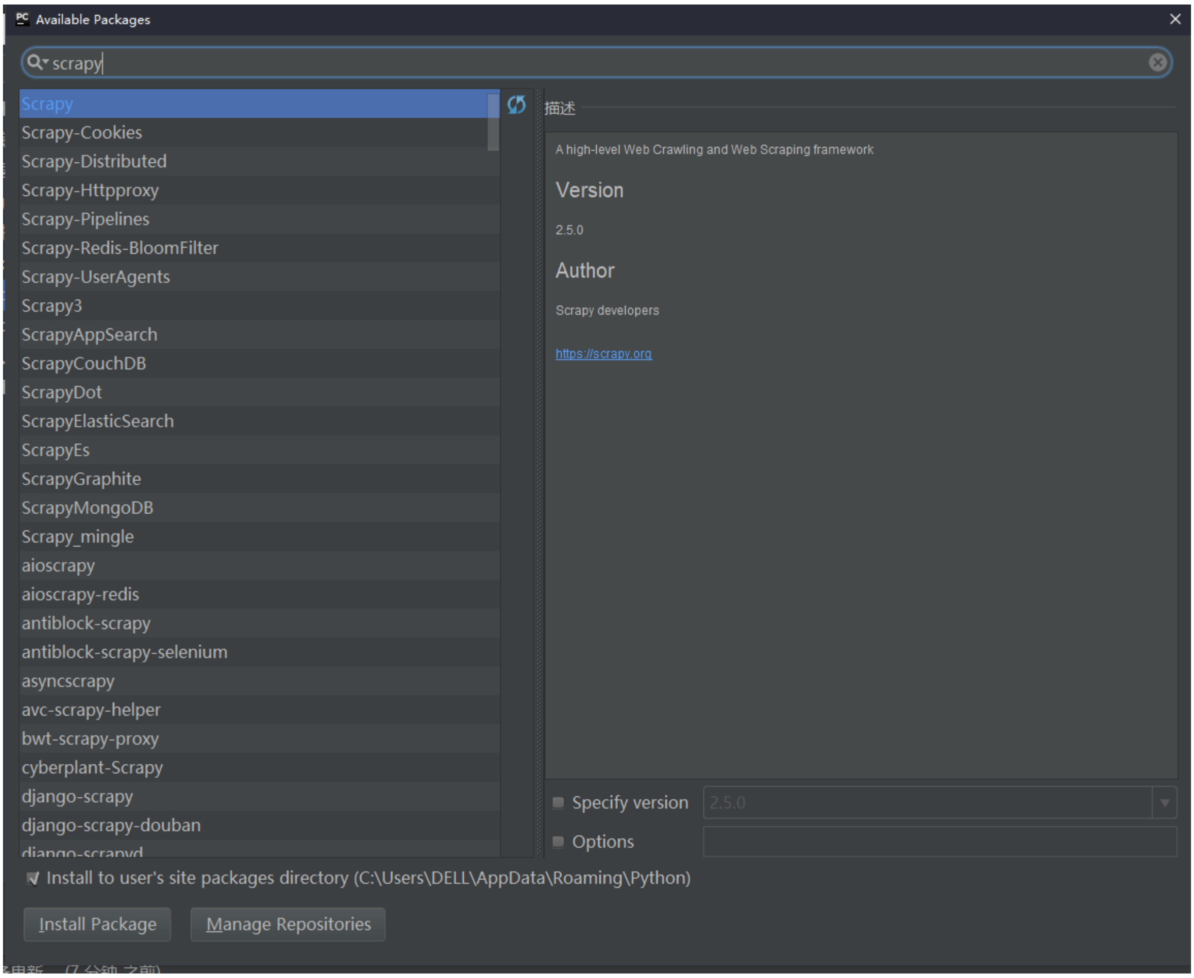


项目: mySpider › Project Interpreter

项目翻译: 3.7.0 (C:\Python\python.exe)

Package	Version	Latest
Automat	20.2.0	20.2.0
Django	2.1.15	➡ 3.2
Keras-Preprocessing	1.1.2	1.1.2
Markdown	3.3.3	➡ 3.3.4
MarkupPy	1.14	1.14
Pillow	7.0.0	➡ 8.2.0
Protego	0.1.16	0.1.16
PyDispatcher	2.0.5	2.0.5
PyInstaller	3.6	
PyMySQL	0.9.3	➡ 1.0.2
PyQt5	5.13.0	➡ 5.15.4
PyQt5-sip	12.7.2	➡ 12.8.1
PyYAML	5.3.1	➡ 5.4.1
Scrapy	2.4.1	➡ 2.5.0
Twisted	21.2.0	➡ 21.2.0rc1
WMI	1.5.1	1.5.1
Werkzeug	1.0.1	➡ 2.0.0rc3
absl-py	0.11.0	➡ 0.12.0
altgraph	0.17	0.17
asgiref	3.2.7	➡ 3.3.4
astunparse	1.6.3	1.6.3
attrs	20.3.0	20.3.0
beautifulsoup4	4.9.3	4.9.3
cachetools	4.1.1	➡ 4.2.1
certifi	2020.6.20	➡ 2020.12.5
cffi	1.14.0	➡ 1.14.5
chardet	3.0.4	➡ 4.0.0
click	7.1.1	➡ 8.0.0a1

右上角的绿色加号点击+



scrapy

Scrapy
Scrapy-Cookies
Scrapy-Distributed
Scrapy-Httpproxy
Scrapy-Pipelines
Scrapy-Redis-BloomFilter
Scrapy-UserAgents
Scrapy3
ScrapyAppSearch
ScrapyCouchDB
ScrapyDot
ScrapyElasticSearch
ScrapyEs
ScrapyGraphite
ScrapyMongoDB
Scrapy_mingle
aioscrapy
aioscrapy-redis
antiblock-scrapy
antiblock-scrapy-selenium
asyncscrapy
avc-scrapy-helper
bwt-scrapy-proxy
cyberplant-Scrapy
django-scrapy
django-scrapy-douban
django-scrapyd

描述

A high-level Web Crawling and Web Scraping framework

Version

2.5.0

Author

Scrapy developers

https://scrapy.org

□ Specify version 2.5.0

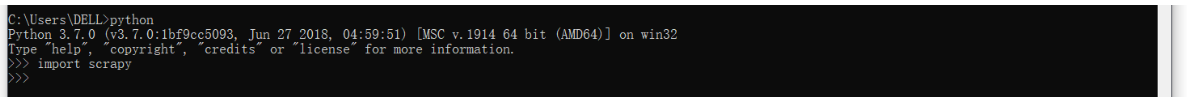
□ Options

☑ Install to user's site packages directory (C:\Users\DELL\AppData\Roaming\Python)

搜索Scrapy安装

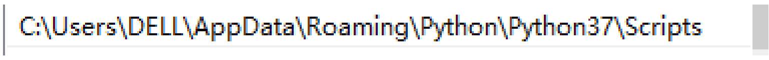
install后应该会显示install sucessfully

然后在terminal检查下scrapy是否安装成功,如下



```
C:\Users\DELL>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import scrapy
>>>
```

如果显示的是找不到模块,就在环境变量里配置一下,以下是我的路径位置供参考



C:\Users\DELL\AppData\Roaming\Python\Python37\Scripts

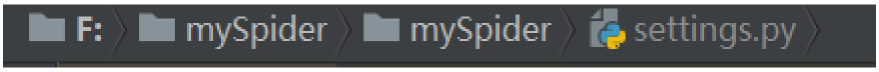
2.

开爬

terminal进入某个文件夹下,我选的是F:,执行命令

```
scrapy startproject mySpider
(mySpider是项目名,自选)
```

然后在F:盘下应该有个项目,用pycharm打开,项目结构如下图(我project侧栏有问题,将就着看)



F:/mySpider/mySpider/settings.py更改为

```
# Scrapy settings for mySpider project
#
```

```
# For simplicity, this file contains only settings considered important or
# commonly used. You can find more settings consulting the documentation:
#
#     https://docs.scrapy.org/en/latest/topics/settings.html
#     https://docs.scrapy.org/en/latest/topics/downloader-middleware.html
#     https://docs.scrapy.org/en/latest/topics/spider-middleware.html

BOT_NAME = 'mySpider'

SPIDER_MODULES = ['mySpider.spiders']
NEWSPIDER_MODULE = 'mySpider.spiders'

# Crawl responsibly by identifying yourself (and your website) on the user-agent
# USER_AGENT = 'mySpider (+http://www.yourdomain.com)'

# Obey robots.txt rules
ROBOTSTXT_OBEY = False

# Configure maximum concurrent requests performed by Scrapy (default: 16)
# CONCURRENT_REQUESTS = 32

# Configure a delay for requests for the same website (default: 0)
# See https://docs.scrapy.org/en/latest/topics/settings.html#download-delay
# See also autothrottle settings and docs
# DOWNLOAD_DELAY = 3
# The download delay setting will honor only one of:
# CONCURRENT_REQUESTS_PER_DOMAIN = 16
# CONCURRENT_REQUESTS_PER_IP = 16

# Disable cookies (enabled by default)
# COOKIES_ENABLED = False

# Disable Telnet Console (enabled by default)
# TELNETCONSOLE_ENABLED = False

# Override the default request headers:
# DEFAULT_REQUEST_HEADERS = {
#   'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
#   'Accept-Language': 'en',
# }

# Enable or disable spider middlewares
# See https://docs.scrapy.org/en/latest/topics/spider-middleware.html
# SPIDER_MIDDLEWARES = {
#    'mySpider.middlewares.MyspiderSpiderMiddleware': 543,
# }

# Enable or disable downloader middlewares
# See https://docs.scrapy.org/en/latest/topics/downloader-middleware.html
# DOWNLOADER_MIDDLEWARES = {
#    'mySpider.middlewares.MyspiderDownloaderMiddleware': 543,
# }

# Enable or disable extensions
# See https://docs.scrapy.org/en/latest/topics/extensions.html
# EXTENSIONS = {
#    'scrapy.extensions.telnet.TelnetConsole': None,
# }
# Configure item pipelines
# See https://docs.scrapy.org/en/latest/topics/item-pipeline.html
ITEM_PIPELINES = {
    #'mySpider.pipelines.MyspiderPipeline': 300,
    # 'mySpider.pipelines.MyspiderPipeline1': 301,
     'mySpider.pipelines.MyspiderPipeline2': 302
}

# Enable and configure the AutoThrottle extension (disabled by default)
# See https://docs.scrapy.org/en/latest/topics/autothrottle.html
# AUTOTHROTTLE_ENABLED = True
# The initial download delay
# AUTOTHROTTLE_START_DELAY = 5
# The maximum download delay to be set in case of high latencies
# AUTOTHROTTLE_MAX_DELAY = 60
# The average number of requests Scrapy should be sending in parallel to
# each remote server
# AUTOTHROTTLE_TARGET_CONCURRENCY = 1.0
# Enable showing throttling stats for every response received:
# AUTOTHROTTLE_DEBUG = False

# Enable and configure HTTP caching (disabled by default)
# See https://docs.scrapy.org/en/latest/topics/downloader-
middleware.html#httpcache-middleware-settings
# HTTPCACHE_ENABLED = True
# HTTPCACHE_EXPIRATION_SECS = 0
# HTTPCACHE_DIR = 'httpcache'
# HTTPCACHE_IGNORE_HTTP_CODES = []
# HTTPCACHE_STORAGE = 'scrapy.extensions.httpcache.FilesystemCacheStorage'

# LOG_LEVEL = "WARNING"

MYSQL_HOT = "127.0.0.1"
MYSQL_DBNAME = "spider_test"
MYSQL_USER = "root"
MYSQL_PASSWORD = "ana"
```

最后四行是我的本地mysql配置,根据自己的写

F:\mySpider\mySpider\items.py改为如下

```
import scrapy
from scrapy.loader import ItemLoader
from scrapy.loader.processors import TakeFirst



class SpiderItem(scrapy.Item):
    name=scrapy.Field()
    title=scrapy.Field()
```

F:\mySpider\mySpider\items.py\pipelines.py改为如下

```
# Define your item pipelines here
#
# Don't forget to add your pipeline to the ITEM_PIPELINES setting
# See: https://docs.scrapy.org/en/latest/topics/item-pipeline.html


# useful for handling different item types with a single interface
from itemadapter import ItemAdapter

#
# class MyspiderPipeline:
#     def process_item(self, item, spider):
#         print(item)
#         return item
#
#
# class MyspiderPipeline1:
#     def process_item(self, item, spider):
#         if spider.name == "spiders":
#             print(item)
#             return item


import pymysql
from twisted.enterprise import adbapi
import pymysql.cursors


class MyspiderPipeline2(object):
    def __init__(self, dbpool):
        self.dbpool = dbpool

    @classmethod
    def from_settings(cls, settings):  # 函数名固定，会被scrapy调用，直接可用settings的值
        """
        数据库建立连接
        :param settings: 配置参数
        :return: 实例化参数
        """
        print(settings['MYSQL_HOST'])
        adbparams = dict(
            host=settings['MYSQL_HOST'],
            db=settings['MYSQL_DBNAME'],
            user=settings['MYSQL_USER'],
            password=settings['MYSQL_PASSWORD'],
            cursorclass=pymysql.cursors.DictCursor  # 指定cursor类型
        )

        # 连接数据池ConnectionPool，使用pymysql或者Mysqldb连接
        dbpool = adbapi.ConnectionPool('pymysql', **adbparams)
        # 返回实例化参数
        return cls(dbpool)

    def process_item(self, item, spider):
        """
        使用twisted将MySQL插入变成异步执行。通过连接池执行具体的sql操作，返回一个对象
        """
        query = self.dbpool.runInteraction(self.do_insert, item)  # 指定操作方法和操作数据
        # 添加异常处理
        query.addCallback(self.handle_error)  # 处理异常

    def do_insert(self, cursor, item):
        # 对数据库进行插入操作，并不需要commit，twisted会自动commit
        insert_sql = """
        insert into teacher(name, title) VALUES (%s,%s)
        """
        cursor.execute(insert_sql, (item['name'], item['title']))

    def handle_error(self, failure):
        if failure:
            # 打印错误信息
            print(failure)
```

接着创建爬虫文件,目录为F:\mySpider\mySpider\spiders\spiders.py(spiders.py为新建文件)

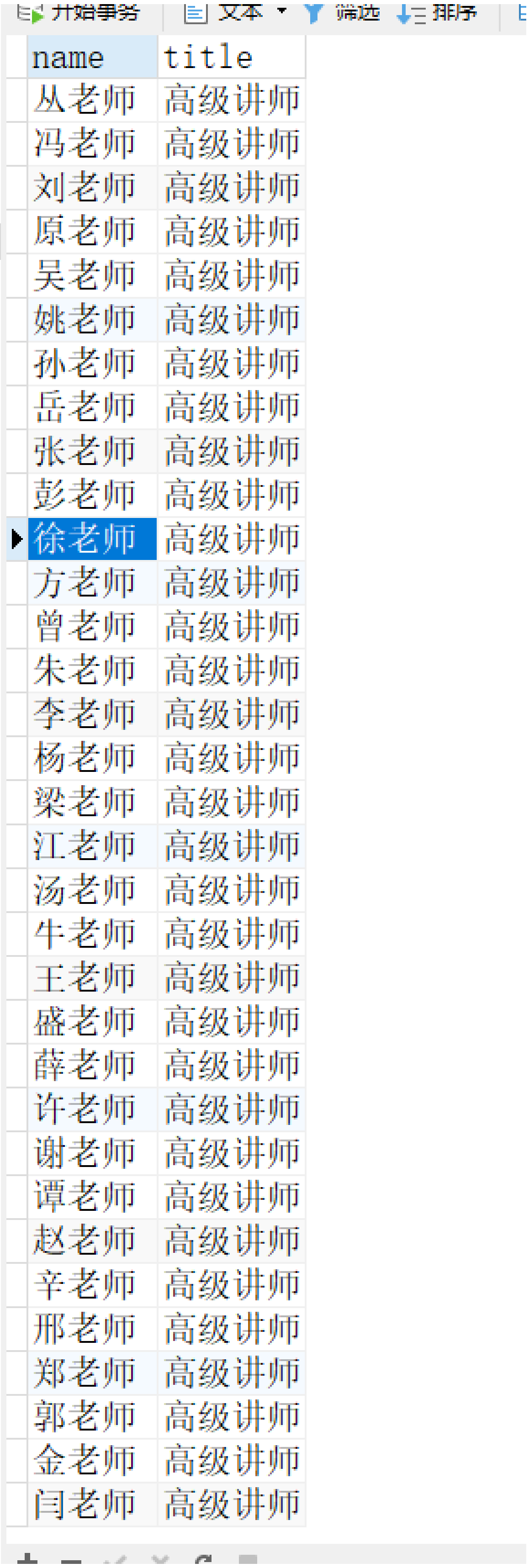
spiders.py代码如下

```
import scrapy


# from items import SpiderItemLoader, SpiderItem

# class collection75Item(scrapy.Item):
#     museumID = scrapy.Field()
#     collectionID = scrapy.Field()
#     collectionName = scrapy.Field()
#     collectionIntroduction = scrapy.Field()
#     collectionImage = scrapy.Field()  # 图片链接


class SpiderItem(scrapy.Item):
    name = scrapy.Field()
    title = scrapy.Field()


class SpidersSpider(scrapy.Spider):
    name = 'spiders'  # 爬虫名
    allowed_domains = ['itcast.cn']  # 允许爬虫的范围
    start_urls = ['http://www.itcast.cn/channel/teacher.shtml']  # 最开始请求的url的地址

    def parse(self, response):
        li_list = response.xpath("//div[@class='tea_con']/div/ul/li")
        print(li_list)
        item = SpiderItem()

        for li in li_list:
            item['name'] = li.xpath(".//h3/text()").extract_first()
            item['title'] = li.xpath(".//h4/text()").extract_first()
            yield item



            # name = 'collection4'
            # allowed_domains = ['jb.mil.cn']
            # start_urls = ['http://www.jb.mil.cn/was/web/search?
token=14.1499419140318.94&channelid=237727']
            #
            # def parse(self, response):
            #     li_list =
response.xpath("//div[@class='relicAppRight']/div[@class='raAppList']/ul/li")
            #     for li in li_list:
            #         item = collection75Item()
            #         item["museumID"] = 4
            #         url = li.xpath("./a/@href").extract_first()
            #         yield scrapy.Request(
            #             url,
            #             callback=self.parse_detail,
            #             meta={"item": item}  # 传递参数
            #         )
            #
            # def parse_detail(self, response):
            #     item = response.meta["item"]
            #     item['collectionName'] =
response.xpath("//div[@class='interContext']/h2/text()").extract_first()
            #     item['collectionImage'] =
'http://www.jb.mil.cn/gcww/wwjs_new/shzysq/201707/' + response.xpath(
            #         "//img[@border='0']/@oldsrc").extract_first()
            #
            #     # 从这以上的代码都是没问题的 都是写好的 Name 和 Image都是爬完的
            #     # 就是下面这个Introduction还没有爬取成功
            #     #
http://www.jb.mil.cn/gcww/wwjs_new/shzysq/201707/t20170705_32875.html  这是关于彭桓
武的那个url
            #
            #     # 这两行代码是我测试<p>能不能被找到
            #     data =
response.xpath("//div[@class='interaction']/div[@class='interContext']/p")
            #     item['collectionIntroduction'] = "这个有点难爬,后面再改"
            #     yield item
```

(为什么spiders.py里又有个SpiderItem,按理说Item应该写在items.py里的,但是我的pycharm不好用,标记为root也引用不了items.py,姑且这样用着)

```
接着在terminal项目目录下执行
scrapy crawl spiders
spiders是spiders/spiders.py里的name,要对应好
```

结果可以在数据库看到如下图



name	title
从老师	高级讲师
冯老师	高级讲师
刘老师	高级讲师
原老师	高级讲师
吴老师	高级讲师
姚老师	高级讲师
孙老师	高级讲师
岳老师	高级讲师
张老师	高级讲师
彭老师	高级讲师
徐老师	高级讲师
方老师	高级讲师
曾老师	高级讲师
朱老师	高级讲师
李老师	高级讲师
杨老师	高级讲师
梁老师	高级讲师
江老师	高级讲师
汤老师	高级讲师
牛老师	高级讲师
王老师	高级讲师
盛老师	高级讲师
薛老师	高级讲师
许老师	高级讲师
谢老师	高级讲师
谭老师	高级讲师
赵老师	高级讲师
辛老师	高级讲师
邢老师	高级讲师
郑老师	高级讲师
郭老师	高级讲师
金老师	高级讲师
闫老师	高级讲师

我的表设置了主键,所以条数会少一点