使用 ImagesPipeline 下载图片

ImagesPipeline 简介

Scrapy 用 ImagesPipeline 类提供一种方便的方式来下载和存储图片。

可以将下载图片转换成通用的 JPG 和 RGB 格式

ImagesPipeline 重写的方法

需要在自定义的 ImagePipeline 类中重写的方法：

get_media_requests(self,item, info)：

Pipeline 将从 item 中获取图片的 URLs 并下载它们，并返回一个 Request 对象

所以必须重载 get_media_requests

```
def get_media_requests(self, item, info):
    for image_url in item['image_urls']:
        yield Request(image_url)
```

item_completed(self, results, item, info)：

当完成下载后，结果将以元组形式发送到 item_completed 方法，图片下载完毕后，处理

结果会以元组的方式返回给 item_completed()函数。这个元组定义如下：

(success, image_info_or_failure)

其中，第一个元素表示图片是否下载成功；第二个元素是一个字典。

如果 success=true，表示成功下载，image_info_or_error 词典包含以下键值对：

* url：原始 URL

* path：本地存储路径
* checksum：校验码

如果 success=false，表示下载失败，image_info_or_error 则包含一些出错信息。

```
def item_completed(self, results, item, info):
        image_paths = [x['path'] for ok, x in results if ok]
        if not image_paths:
            raise DropItem("Item contains no images")
        item['image_paths'] = image_paths
        return item
```

生成图片缩略图，添加设置

```
IMAGES_THUMBS = {
'small':(50,50),
'big':(270,270),
}
```

案例：爬取斗鱼直播颜值主播

1、创建 scrapy 项目

scrapy startproject douyuspider

2、items.py

```
import scrapy

class DouyuspiderItem(scrapy.Item):
    name = scrapy.Field()  # 存储照片的名字
    imagesUrls = scrapy.Field()  # 照片的 url 路径
    imagesPath = scrapy.Field()  # 照片保存在本地的路径
```

3、创建爬虫

scrapy genspider douyu http://capi.douyucdn.cn

4、spiders/douyu.py

```
import scrapy
import json
from douyuspider.items import DouyuspiderItem

class DouyuSpider(scrapy.Spider):
    name = 'douyu'
    allowed_domains = ['http://capi.douyucdn.cn']
    offset = 0
    url = "http://capi.douyucdn.cn/api/v1/getVerticalRoom?limit=20&offset="
    start_urls = [url + str(offset)]

    def parse(self, response):
        # 返回从 json 里获取 data 段数据集合
        data = json.loads(response.text)["data"]

        for each in data:
            item = DouyuspiderItem()
            item["name"] = each["nickname"]
            item["imagesUrls"] = each["vertical_src"]

            yield item

        self.offset += 20
        yield scrapy.Request(self.url + str(self.offset), callback=self.parse)
```

5、设置 setting.py

```
ITEM_PIPELINES = {'douyuSpider.pipelines.ImagesPipeline': 1}

# Images 的存放位置，之后会在 pipelines.py 里调用
IMAGES_STORE = "/Users/Power/lesson_python/douyuSpider/Images"

# user-agent
USER_AGENT = 'DYZB/2.290 (iPhone; iOS 9.3.4; Scale/2.00)'
```

生成缩略图设置

```
IMAGES_THUMBS = {
```

```
'small':(50,50),
'big':(270,270),
}
```

6、pipelines.py

```
import scrapy
import os
from scrapy.pipelines.images import ImagesPipeline
from scrapy.utils.project import get_project_settings

class DouyuspiderPipeline(ImagesPipeline):
    # def process_item(self, item, spider):
    #         return item

    IMAGES_STORE = get_project_settings().get("IMAGES_STORE")

    def get_media_requests(self, item, info):
        image_url = item["imagesUrls"]
        yield scrapy.Request(image_url)

    def item_completed(self, results, item, info):
        # 固定写法，获取图片路径，同时判断这个路径是否正确，
        # 如果正确，就放到 image_path 里，ImagesPipeline 源码剖析可见
        print('item_completed:',info)
        if results[0][0] ==True :
            image_path = results[0][1]["path"]
            print("2:",image_path)


        os.rename(self.IMAGES_STORE + "/" + image_path[0], self.IMAGES_STORE + "/" +
item["name"] + ".jpg")
        item["imagesPath"] = self.IMAGES_STORE + "/" + item["name"]

        return item
```

7、爬虫调试

创建 run.py 文件，和 setting.py 同级目录:

```
from scrapy import cmdline
```

```
name = 'douyu'
cmd = 'scrapy crawl {0}'.format(name)

cmdline.execute(cmd.split())
```

作业

图行天下海报爬虫　爬取海报图片保存到 images 目录下

http://so.photophoto.cn/tag/%E6%B5%B7%E6%8A%A5