tools Documentation

发布 *v1.0*

yuanjh

2022 年 07 月 05 日

Contents:

工具类, 脚本等

CHAPTER 1

collect_links module

脚本功能：广度遍历采集链接（友情链接采集工具）

涉及：树的广度遍历（队列），代理模式，多线程，生成者消费者，跨线程数据共享 - 队列（阻塞）

帮助:python collect_links.py -h

使用示例:python collect_links.py -s 'https://hexo.yuanjh.cn' -suf '/links'

程序执行步骤

1,https://hexo.yuanjh.cn => (1,https://hexo.yuanjh.cn/links)

2,(1,https://hexo.yuanjh.cn/links) => [(2,http://xxx.yy.com/links),(2,https://zz.ff.cn/links)]

3,[(2,http://xxx.yy.com/links),(2,https://zz.ff.cn/links)]=> [(3,http://xxx.yy.zz/links),(3,https://zz.ff.zz/links)]

=> 循环此步骤

class collect_links.**CollectLinks**(*seed_url: object*, *suffix: object*, *max_count: object = 100*, *max_depth: object = 10*)

基类：`object`

递归采集链接

变量

- **seed_url** (*str*) – 种子链接
- **suffix** (*str*) – 后缀
- **max_count** (*int*) – 最大采集链接个数
- **max_depth** (*int*) – 最大采集链接深度

`collect()` → None

启动多线程进行网页广度采集任务

`headers = {'Accept': 'application/json, text/javascript, */*; q=0.01', 'Accept-Encoding': 'gzip, d`

`process_queue()` → None

执行任务队列中任务

广度遍历形式解析 url 对应网页中的 url 地址, 并回送到任务队列 1, 从任务队列 unique queue 中获取到 url 2, 下载解析 url, 获取网页中的 url 地址 3, 将 url 地址加入到 unique queue 中

`class collect_links.UniqueQueue`(*maxsize: int = 0*, *key: Callable = None*)

基类: `object`

工具类, 唯一性队列 (Queue), 同一个元素只能入队一次

变量

- **maxsize** (*int*) – 队列最大元素个数（队列为阻塞队列）
- **key** (*callable*) – 可调用函数，作用在 item 上用于产生唯一的 key 来做重复性判别，重复元素仅能入队一次（首次）

`empty()` → bool

判断队列是否为空.

`get()` → Tuple[int, str]

获取队列元素.

`put`(*item: Tuple[int, str]*) → bool

向队列中添加新元素

Param tuple item: item[0] url 的深度,url 链接地址

CHAPTER 2

pkl2csv module

将 pkl 转为 csv 格式 (方便直接导入数据库等)

帮助:python pkl2csv.py -h

使用示例:python pkl2csv.py -f '/home/john/下载/dd_price_vp_20200809_20200818.pkl' -m '{"index": "datetime", "volume": "vol"}'

步骤

1, 依次取得 pkl 文件 minor_xs 轴维度 as df

2,df.reset_index(),df.dropna(),df 拼接为 all_df

3,all_df.rename()

4,all_df.to_csv(index=False)

class pkl2csv.**PKl2Csv**

基类: object

static file_path_split(*filename: str*) → Tuple[str, str, str]

获取文件路径、文件名、后缀

参数 **filename** (*str*) – 文件全路径

Return tuple 文件路径、文件名、后缀

trans(*file_path_pkl: str*, *rename_map: Dict[str, str] = None*) → None

pkl 文件转为 csv 文件

参数

- **file_path_pkl** – pkl 文件路径
- **rename_map** – 字段映射字典 (dict)

CHAPTER 3

rtmp_player module

基于 opencv 的简易流媒体播放器

帮助:python pkl2csv.py -h

使用示例: python rtmp_player.py -u rtmp://58.200.131.2:1935/livetv/hunantv(湖南卫视的 rtmp 地址)

控制逻辑:

p: 暂停

c: 继续

f: 完成 (结束)

class rtmp_player.**RtmpPlayer**(*url: str*)

基类：object

简易播放器

变量 **url** (*str*) – 播放的视频流地址

last_frame_nowait()

非阻塞的方式读取下一帧

Return **array** 最新帧

max_queue_size = 200

read()

阻塞的方式读取下一帧

Return **array** 最新帧

run_status

start_capture() → None
启动帧采集进程

class rtmp_player.RunStatus
基类：enum.Enum

视频状态枚举类

CONTINUE = 3

FINISH = 4

NOT_START = 0

PAUSE = 2

START = 1

get_after_status = <bound method RunStatus.get_after_status of <enum 'RunStatus'>>

CHAPTER 4

valid_rtmp module

寻找过滤有效的 rtmp,or,rtsp 直播地址

帮助:python pkl2csv.py -h

使用示例:python valid_rtmp.py -u https://blog.csdn.net/osle123/article/details/52757886

步骤:

1, 下载页面：https://blog.csdn.net/osle123/article/details/52757886（避免使用 csdn 等，需点击触发显示全部的网页）

2, 正则匹配：rtmp://, rtsp://等地址

3,[rtmp;//xx.com,rtsp://yy.com] 使用 ping+cv2.read() 验证有效性

class valid_rtmp.**ValidRtmp**

基类：object

采集过滤有效的 rtmp,or,rtsp 播放地址

get_rtmp_url(*url: str*) → List[str]

获取 url 网页内容中的 rtsp,rtmp 地址

参数 **url** (*str*) – 待采集的种子 url 地址

Return list 种子 url 中的 rtsp,rtmp 地址

headers = {'Accept': 'application/json, text/javascript, */*; q=0.01', 'Accept-Encoding': 'gzip, d

valid(*url: str*) → str

是否是合法的 rtsp,rtmp 地址

:param str url: 待验证的 url(rtsp or rtmp) 地址，:return str: 如果：是，返回入参的 url，如果：不是，返回空串

CHAPTER 5

vnote2hexo module

CHAPTER 6

vnote_md_format module

CHAPTER 7

vnote2sphinx module

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python 模块索引

索引