Data processing with PROCESSit module

by Roman Vey

Ukrainian Catholic University
APPLIED SCIENCES FACULTY

INTRODUCTION

Today we can find a lot of modules for scrapping data from page. But it`s hard to find library for scrapping from different sources and even more harder to find something for autonomous work. This module was developed for this purposes. With it you would be able to develop flexible and scalable system, which can group, merge and autodetect sources with a lot of prebuilded functions.

MAIN COMPONENTS

Splitter - split text in list using custom function.
RegexSplitter - split text in list using custom regex.
DataBlock(**DB**) - use **Splitters** to get columns from text, one **DB** could be divided into more **DB**s using same **Spliiters,** you perform operations to all of them at once.
Parser - set of **DB**s and rules to them.
LinkParser - like **Parsers**, but you also need to specify which columns would be sources and paths to save.
Page - represent HTML-page, you need to specify sources where get URLs and **Parser** or **LinkParser** (or both).
File - can operate with multiple files, all previous structures use it inside.
Crawler - set of **Pages** and rules to them.
Scheduler - high-level wrapper over "sched" and "threading" modules, with it you can specify time, when to start, pause, stop crawling and order.
RegexSplitterEnv - utility, which helps you to write regex queries in few times faster.

FEATURES

- Scheduler, you can specify when you want to parse data;
- Split large files into smaller;
- Can work with multiple files like with one;
- Crawler could be started in different threads, because scheduler works with them by default;
- Flexible, because project divided into small logical components;
- Tools for writing correct and fast regex queries;
- Additional post processors for regex like "remove trailing spaces" and "remove tags and content inside" for faster work;
- Logging system for checking progress in real-time;
- Fault tolerant, can handle: empty link file, not valid link, not founded data, unexpected exit etc.;
- Can check for duplicates, so you can run it in loop and module will add only new data;
- Can work with different scenarios: check one page with some interval (for example, get Apple prices every minute), iterate through links to get more links and after that parse them (for example, some sites have pages with links to articles) or iterate through pages and parse them in one step (for example, some pages have links to next pages);

```python
import logging
import logging.config
logging.config.fileConfig('logging.conf')

from core import Scheduler
from custom.crawlers.link_crawler import LinkCrawler
from custom.crawlers.article_crawler import ArticleCrawler

link_crawler = LinkCrawler(timeout=10)
article_crawler = ArticleCrawler(timeout=10)
scheduler = Scheduler()

scheduler.add_at_specific_time("08/01/2019 18:30:00", link_crawler.start,
repeat="00:06:00", name="Crawl links for articles")
scheduler.add_at_specific_time("08/01/2019 18:35:00", article_crawler.start,
repeat="00:06:00", name="Crawl articles themselves")
scheduler.start()
```

GITHUB LINK

https://github.com/romanvey/process_it