ISSN: 1813-4890

Import and Export Biological News Crawler based on Scrapy

Ziyi Huang[a], Rigui Zhou[b]

Technology Information College, Shanghai Maritime University, Shanghai 200135 China

[a]huangzyi@163.com, [b]rgzhou@shmtu.edu.cn

Abstract

In the customs entry-exit inspection and quarantine, biological species resources are the key protection objects. In order to acquire the latest biological news at home and abroad, an import and export biological news crawler system has been designed and developed. Based on Scrapy crawler framework, this article acquire biological news from six official websites. First, it download all web pages, and use selenium to obtain web pages with forms. Then, filter and clean all data. Next, download images, files and read all the text from attachment. Finally, store them in mysql database asynchronously. After testing, the system can run regularly in the Linux system, crawl biological news, and provide data support for biological early warning and protection of biological species resources.

Keywords

Scrapy; Crawler; Import and Export; Creature.

1. Introduction

The entry-exit inspection and inspection of biological species resources can protect biological species resources and biodiversity, and prevent the loss of biological resources and the invasion of alien species [1]. The latest import and export biological news plays an important role in enhancing the vigilance of customs related personnel to sensitive biological species and intercepting important biological species in real time. There is a lot of websites on the Internet, so it is difficult for customs officials to extract important information from the large amount of information. The system crawls biological-related laws and regulations and the latest notices from government websites such as the general administration of customs, the china center for disease control and prevention, and the china Center for Animal and Epidemiology.

2. Web Crawler

Web crawlers can automatically crawl information in web pages according to certain rules. Generally, it can be divided into three steps: web page download, web page parsing and data storage. The webpage downloader downloads the webpage to the local, the webpage parser extracts the required information from the content of the html webpage, and finally stores the extracted data in a certain format [2].

2.1. Scrapy Frame

Scrapy is a Twisted-based asynchronous processing framework based on Python language, which is mainly used to scrape web pages and extract structured data from them. Scrapy mainly consists of five components: Scheduler, Downloader, Spider, Item Pipeline, and Scrapy Engine [3], see Fig. 1. Spider sends Request to Scheduler through Engine. The Engine sends the Request returned by the Scheduler to the Downloader through Middleware. The Downloader sends the obtained Response to the Engine through Middleware, and the Engine returns it to the Spider. Spider parses the content, sends the parsed content to the Engine, and the Engine sends the parsed Item to the Item pipeline.

ISSN: 1813-4890

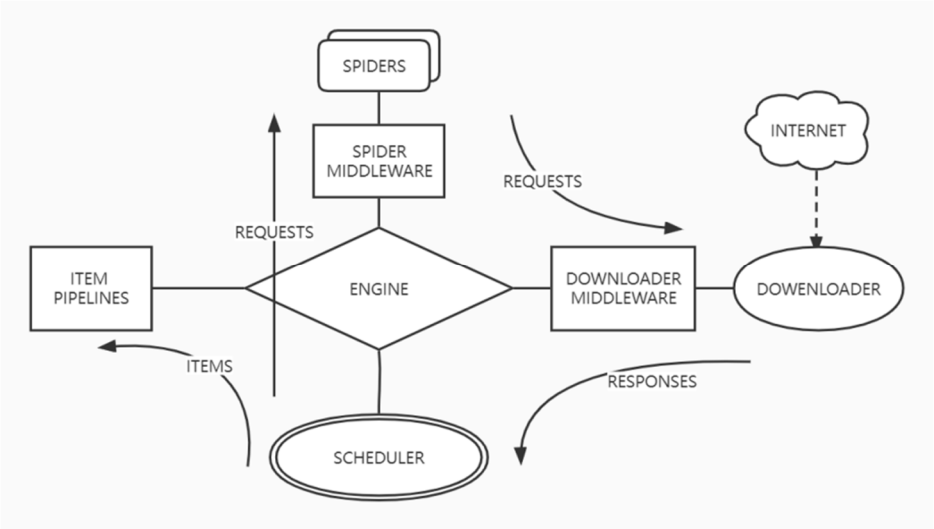

Fig 1. Scrapy architecture diagram

2.2. Selenium

Selenium is a web automation testing framework. Selenium can simulate the user's operation in the browser and supports a variety of browsers [4]. Before using Selenium, you must download the corresponding version of the Selenium driver.

3. Design of Crawler

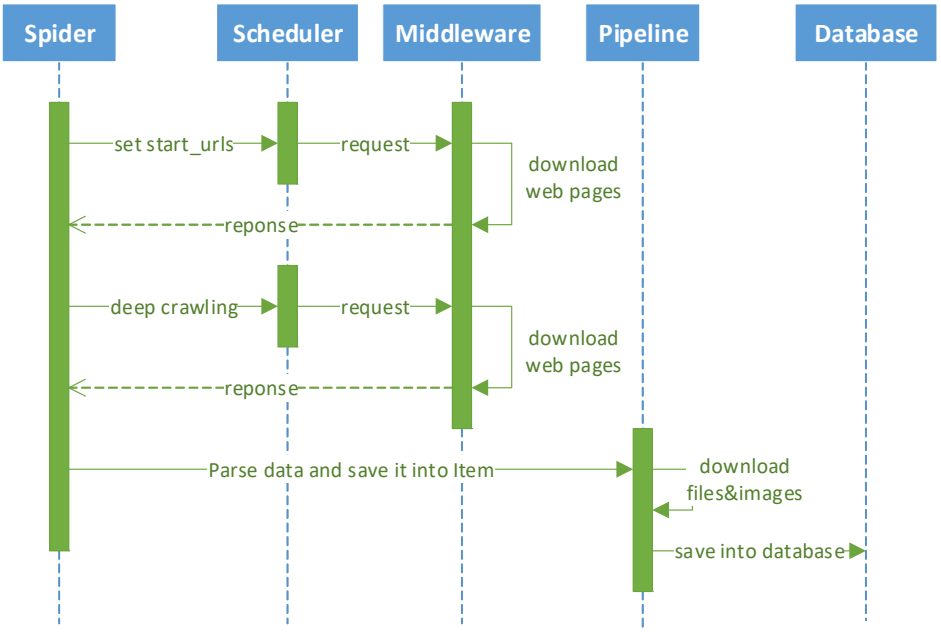

Fig 2. Scrapy timing diagram

The main work of this paper is to crawl news data from websites. First, extract news links from news catalog page. Next, extract information from web pages of the news links. If there are images or files in the webpage, the link of the image or file should be extracted. Then download images and files. Finally, upload the extracted data to the database.

According to the structure of scrapy, firstly set the start urls in spiders. Secondly, set different middleware according to the type of the web page, take each url as a request. and send the

ISSN: 1813-4890

request to the corresponding downloader middleware. Thirdly, package the web content into a response and send it to the spider. Fourthly, in the spiders, web content is parsed and cleaned. First extract new links that need to be crawled. Then repeat the steps of downloading web pages and store the cleaned data in Items. Fifthly, download pictures and attachments in the pipeline. Read the files and stored in Item. Finally, save Items in the database. The scrapy timing diagram is shown in Fig.2.

4. Web Page Download

In each website, the web page structure of the same type of web page is roughly the same. And a spider file is set for each website, which is placed in the same spider for parsing.

The start page of each website is a pageable page with multiple news links. The number of pages can be changed through the parameters in the url, and the news links to be crawled are stored in the start_urls array. To limit crawled domains, set domain names in allowed_domains. Then in the start_requests function, the request is sent through *yield scrapy. Request (url ,callback =self.parse)*, and the returned web page content will be parsed by the parse function. Send the request to the downloader middleware to download the web page content, use different downloader middleware according to whether the web page contains forms, personalize the settings of different spiders in *custom_setting.py*, and then set custom_setting in each spider. For general web pages, you only need to use scrapy's default downloader middleware; for web pages with forms, selenium will be used to solve the problem of anti-crawlers.

In the article, selenium is the main method to solve form problems in customs regulations website. The use of selenium requires pre-downloading the webdriver of the corresponding browser version. In the article, Firefox browser is used. The steps to download a webpage with selenium are as follows:

Step 1 In selenium downloader middleware, set headless mode and open the browser. Setting headless mode can make the browser running in the background.

step 2 Delay and wait until the interface is loaded. If the following steps are performed before the interface is loaded, the content of the webpage cannot be obtained and an error will be reported. This article mainly uses the explicit wait method to wait until the navigation interface appears before continuing to run.

Step 3 Encapsulate the data to be posted in ajax and execute js. There is a search form for conditional selection in the interface of customs regulations, which can narrow the scope. After that the data will be preliminarily filtered. Although simulating clicks through selenium is also feasible, it is really slow.

Step 4 Obtain the page source code. Return the response to the corresponding spider for further analysis.

step 5 Close the browser.

5. Web Page Parsing

The news page is divided into two types: news catalog page and news body page. In this article, the main method of parsing the web pages is the xpath selector and css selector which comes with scrapy. Further parsing and changes of content use beautifulsoup.

5.1. Xpath, cssSelector, beautifulSoup

Xpath selectors locate elements in html through paths [5] while css selectors locate elements through css styles [6]. There are two forms of web page content. One is the response returned directly from the downloader middleware, using *Selector(response=response)* or *response*. The

ISSN: 1813-4890

other is text which use *Selector(text=li)*. Then select element by css function or xapth function. Finally use the extract function or re function to extract.

BeautifulSoup is a python library for extracting information from structured documents [7]. It provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree. The get_text function can easily extract the text content in the tag or in its descendants.

5.2. News Catalog Page Parsing.

The news body page contains multiple news links. It is generally an unordered list, in which each li is a news link. The information of news link includes url, news title and date, which can be directly extracted through the xpath of the url, such as *response.xpath ('// ul [@ class= "conList_ull"]//a/@href').extract()*.

In order to get the latest news and avoid repeated crawling, the number of pages and dates of the news are limited. The limit on the number of news pages is that when adding the urls to start_urls, only limited pages are to be crawled. The restriction on the news date will limit the news after the limited date to be crawled. First, the content in each li will be extracted, and then matches date with regular expression. If it is later than the limited date, the crawling will be continuing. The newly acquired news links will be put into the scheduler for deep crawling. Next each news link is parsed by the custom getInfo function after being downloaded by the downloader middleware.

5.3. News Body Page Parsing

Table 1. Item, database and their explanation

Item	Database	Explaination
title	headline(varvhar)	News title.
url	url(varchar)	News urls.
date	news_date(date), upload_date(date)	The release time of the news is date in item and news_date in database. The upload_date in the database stores the date of the first crawling.
content	message(text)	News text.
source	source(varchar)	News source sites, defined in spider.
Images(image_urls, images, images_path, image_info)	image_urls(text), images_info(text), images_path(varchar)	Image_urls and images is what must be defined. Image_urls stores the image link to be downloaded, and images stores the downloaded image information. The custom images_path stores the location where images are downloaded locally, and image_info stores image links and image notes.
Files(file_url, files, files_path, files_content)	file_urls(text), files_content(text), files_path(varchar)	File_urls and files must be defined. File_urls stores the link of attachment to be downloaded, and files stores the downloaded file information. The customized files_path stores the local location where the file is downloaded, and file_content stores the content.that reads from file.

The information on the news body pages of different websites is different, but the news title, date and text can always be extracted. Then the main job of parsing body page is to parse out these contents and save them into item. For the convenience of later display, the text will be

ISSN: 1813-4890

regularized and formatted. If there are attachments or pictures, the links of the attachments or pictures should be extracted.

The content to be extracted is defined in Items.py, each item is initialized by *scrapy. Field ()*. and the set items are shown in Table 1.

News titles are generally the same as web page titles, so the content in the title tag can be taken out. Some titles will be wrapped in h2 tags, depending on the specific situation. The date of the news is generally placed in the span tag, which can be extracted by xpath or css, or by *re_first ('\d {2,4}-\d {1,2}-\d {1,2 after a limited range}')* matches the date.

The format of the news text is relatively complex, you can first extract the div containing all the text, then use BeautifulSoup for further processing, and finally extract the text content through the get_text method. What's more, additional processing of the body content is required.

5.3.1. Space, Newline, First Line Indentation

Before putting the content into beautifulSoup for deep processing, uniformly handles spaces and line breaks with replace method. For example, replace br tag with line breaks. The p tag makes newline in html, so the extracted content cannot be wrapped, and line breaks needs to be added. Also, some paragraphs have the css styles as *text-indent: 2 attribute*, in which the extracted content has no first line indentation. The solution is matching each paragraph with "text-indent: 2 attribute", and use the insert_before method to add a space before each paragraph.

5.3.2. Image Links and Remark Information Extraction

First find all img tags through find_all function, get the image link by getting the attribute of src, and store it in the item['image_urls'] list. The non-empty center segment after finding the images is the images remark information. Finally, the images remark information and blank lines is deleted by the extract method.

5.3.3. Attachment Links Extraction

Find the paragraph with the "attachment" field, the following tags are all attachments, get their href as attachment links, and store them in item['file_urls'].

Store the extracted information in the item, returns through yield. Finally, the data will enter the pipeline for processing.

6. Image, Attachment Handling

6.1. Image, Attachment Download

The ImagesPipeline that comes with scrapy can easily download images. Also, use the FilesPipeline to download files. The use of Images Pipeline requires the following conditions:

1. Two attributes, image_urls and images, are defined in item. Image_urls stores the link to the file to be downloaded. After the file is downloaded, images stores the downloaded image information.

2. In the configuration file *settings.py*, set IMAGES_STORE as the image download path. Open ImagesPipeline in ITEM_PIPELINES.

The use of Files Pipeline is like ImagesPipeline. Correspondingly, need to define two attributes file_urls and files in item. Set FILES_STORE as the files download path in *settings.py,* and open FilesPipeline.

6.2. Attachment Reading

The downloaded files are mainly doc, docx, xls, rar, pdf, ofd files. Files in different formats are read using different packages and methods. After identifying the file suffix, different methods are called for processing. The specific methods are shown in Table 2.

ISSN: 1813-4890

Table 2. Attachment reading

Filename extension	Lib	Steps
doc	win32com (windows), antiword (Linux)	The Linux subprogram executes the antiword file to obtain the contents of the file. In Windows, save the doc file as a docx file with win32com, and read it by reading the content of the docx file.
docx	Io, BytesIO, zipfile	Read the word document into a binary file object, decompress it with zipfile. Then it turns an xml file and read the textin the w:t tag with beautifulSoup.
xls	xlrd	After opening the excel file, read all worksheets, get the number of rows in the worksheet and read them row by row.
rar	Unrar, rarfile	Get the file names stored in the compressed package and read them one by one after decompression. Many of these files are xsd files.
xsd		It can be regarded as an xml file. After opening, read the text in the xs:documentation tag with beautifulSoup.
pdf	pdfplumber	Open it and use the extract_text method to extract the content page by page.
ofd	Io, BytesIO, zipfile	Similar to docx, compress them into binary files and decompressed with zipfile, then a file list will appear. Read these files as xml file one by one and use beautifulSoup to read the content of the ofd:TextCode tag.

7. Data Storage

The database used in the article is mysql. The data table is shown in Table 1. Most type of data are varchar. The type of news_date and upload_date is date. Also, images and files links, urls, text and file content are set to text. To prevent data duplication, set the unique key of url and the unique key of news_date and headline to prevent the same link and the same content from being crawled.

In the article, the twisted asynchronous IO framework adbapi is used to save data in database after converting the items to strings. The steps are as follows:

Step 1 Set the connection pool and pass in the parameter as *self.dbpool = adbapi. ConnectionPool ('pymysql', **dbparams)* at initialization, where dbparams are database connection parameters, including database address, user name, password, pointer, etc.

Step 2 Execute the insert_item method asynchronously in the connection pool and add an error handling method.

Step 3 Prepare data. Convert the list to a string, non-existing value to an empty string and get get current date.

Step 4 Set the sql statement "insert ignore into...", pass other parameters into params, and finally put it into *cursor.execute(sql,params)* for execution.

8. Multiple Crawlers Run Regularly

All spider files are placed in the spider folder. Get the spider list in the run method of crawlall.py in the commands folder. Run with *self.crawler_process.crawl (spider_name, **opts.__dict__)* and start with *self.crawler_process.start()*. Execute *execute("scrapy crawlall".split())* in main.py to run all spiders in parallel.

This article runs regularly in the Linux system. First of all, in cron.sh file, set command to run the crawler system [8]. Then go to the command crontab -e to set the system environment, set

ISSN: 1813-4890

the specified time to run and send the result file to the mailbox. The running result is shown in fig3:

```
haiguan_spider: 4pages, 44data, costs 30.79s, finished
wto_spider: 3pages, 20data, costs 27.61s, finished
reea_spider: 1pages, 20data, costs 28.16s, finished
cahec_spider: 2pages, 16data, costs 32.64s, finished
linye_spider: 4pages, 71data, costs 30.48s, finished
cadc_spider: 5pages, 81data, costs 190.12s, finished
```

Fig 3. Running result

Acknowledgments

This work is supported by the National Key R&D Plan under Grant No. 2021YFF0601200 and 2021YFF0601204.

References

[1] Dong-Ming L I. The Inspection and Quarantine Work of Inbound and Outbound of Biological Species Resources[J]. Journal of Anhui Agricultural Sciences, 2014.

[2] Abukausar M, Dhaka V S, Singh S K. Web Crawler: A Review[J]. International Journal of Computer Applications, 2013, 63(2):31-36.

[3] Wei S, Xia B. The design of product reviews acquisition system based on the Scrapy framework[J]. Microcomputer & Its Applications, 2017.

[4] Shi Z, Shi M, Lin W. The Implementation of Crawling News Page Based on Incremental Web Crawler [C] // 2016 4th Intl. Conf. on Applied Computing and Information Technology (ACIT), 3rd Intl. Conf. on Computational Science/Intelligence and Applied Informatics (CSII), and 1st Intl. Conf. on Big Data, Cloud Computing, Data Science & Engineering (BCD). IEEE, 2016.

[5] Chan C Y, Felber P, Garofalakis M, et al. Efficient filtering of XML documents with XPath expressions [J]. Vldb Journal, 2002, 11(4):354-379.

[6] Tao L, Qian M O. A NEW EXTRACTION METHOD IN DEEP WEB RESULT PAGES BASED ON CSS SELECTOR[J]. Journal of Beijing Technology and Business University(Natural Science Edition), 2009.

[7] Shen C F, Da-Long M O. Use Skills of Beautifulsoup Library[J]. Computer Knowledge and Technology, 2019.

[8] Xu L, Cao S X, Ma L M. A Strategy of Database Project Task Based on the Crontab[J]. Applied Mechanics & Materials, 2014, 610:611-614.