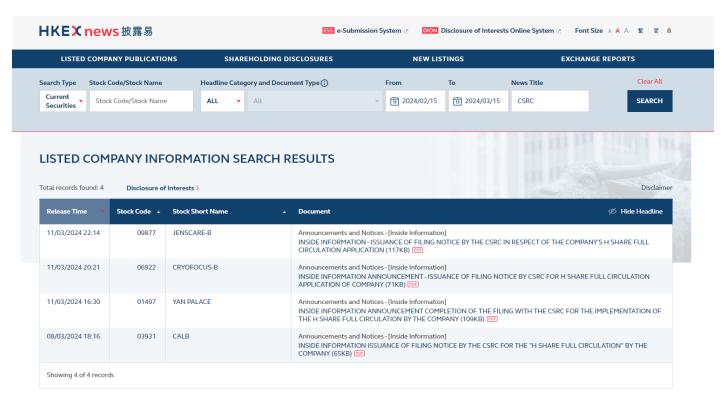# HKEX CCASS Scrape - 15206

Root URL:

https://www1.hkexnews.hk/search/titlesearch.xhtml?lang=en

Description:

Search for "CSRC" in the search bar and always choose a one month date range before today to today.

For example on 2024-03-15:

Output Columns:

Column name	Data type	Example value	Comments
keyword	String	this should always be the value we add into the search box. for example now it is CSRC	
stock_code	String	09877 in screenshot first row	
stock_name	String	JENSCARE-B in screenshot first row	
news_id	String	When you open the link, the unique number of the PDF name. for example the first row has this link: https://www1.hkexnews.hk/listedco/listconews/sehk/2024/0311/2024031101105.pdf	
news_date_time	Timestamp	the release datetime for example first row: 11/03/2024 22:14	
news_link	String	The link in the description: https://www1.hkexnews.hk/listedco/listconews/sehk/2024/0311/2024031101105.pdf	
news_title	String	Text of the document box: Announcements and Notices - [Inside Information] INSIDE INFORMATION - ISSUANCE OF FILING NOTICE BY THE CSRC IN RESPECT OF THE COMPANY'S H SHARE FULL CIRCULATION APPLICATION (117KB)	
news_text	String	Announcements and Notices - [Inside Information] INSIDE INFORMATION - ISSUANCE OF FILING NOTICE BY THE CSRC IN RESPECT OF THE COMPANY'S H SHARE FULL CIRCULATION APPLICATION (117KB)	same as news_title
scrape_datetime	Datetime	scrape datetime of the scrape as usual	

Timeline:

You may complete this job any time and submit any required files to the linked Github repository within one week of accepting the job.

Submission Files:

- Sample.csv for sample data
- A requirement.txt
- scrape/ - containing all of the source code
- Main file: scrape.py that will be run with an output $filename
- OTHER_FILES – e.g. If you require a full historical scrape/anything one off: historical.csv

Job Schema/Output Format:

You should save the output CSV using these settings from a pandas DataFrame:

```
encoding="utf-8"
line_terminator="\n"
quotechar='"'
quoting=csv.QUOTE_ALL
index=False
```

Package Preferences and Scraping Notes:

- It is generally unnecessary to run a full browser, so Selenium is a last resort (only purposeful obfuscation from website owner) and provide explanation
- BeautifulSoup
- Avoid hard codes, references should be relative
- Parse JSON from an API rather than HTML

Runtime Environment:

Your code will be copied from the root to `/usr/src/scrape`

You should feel free to modify the requirements as you need. However, you must keep the `awscli` dependency

You may also upload additional binaries into the repository root and reference them there.

Please do not change the Dockerfile or shell scripts in the repository as this will cause automated test failure.

Execution Command:

```
python scrape.py $filename
```

Page Access Limitations (Max Requests / Day):

If you encounter a captcha during your scrape job, please contact the job poster before continuing.

10% of website traffic max