GovPredict Interview: Scraping Foreign Principals

The Project

Your task is to extract all active foreign principals from FARA. you can find the list here:

https://www.fara.gov/quick-search.html (Click “Active Principals”)

Each data object you return should look something like this:

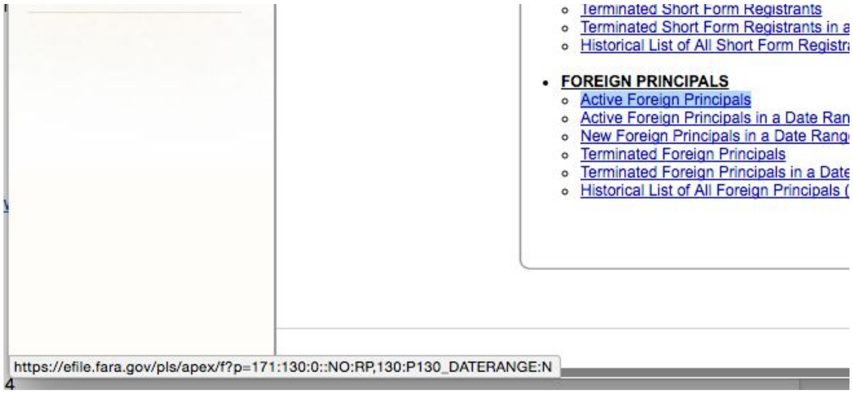
```
{ "url" :
"https://efile.fara.gov/pls/apex/f?p=171:200:::NO:RP,200:P200_REG_NUMBER,P200_DOC_TYPE,P200_CO
UNTRY:2310,Exhibit%20AB,BAHAMAS", "country" : "BAHAMAS",  "state" : null, "reg_num" : "2310",
"address": "Nassau", "foreign_principal" : "Bahamas Ministry of Tourism", "date" :
ISODate("1972-01-27T00:00:00Z"), "registrant" : "Bahamas Tourist Office", "exhibit_url" :
"http://www.fara.gov/docs/2310-Exhibit-AB-19720101-DBBMB702.pdf" }
```

Requirements:

- Code must be written in Python, using Scrapy
- Write unit and/or integration tests for this
- The scraper must be fully automated. This means that we should be able to execute it via one command line task to get all documents
- Provide instructions/Readme to get your project running

Solution

This project and readme its in github: https://github.com/guilhermetavares/myscrapy
In the url https://www.fara.gov/quick-search.html, click on "Active Foreign Principals":



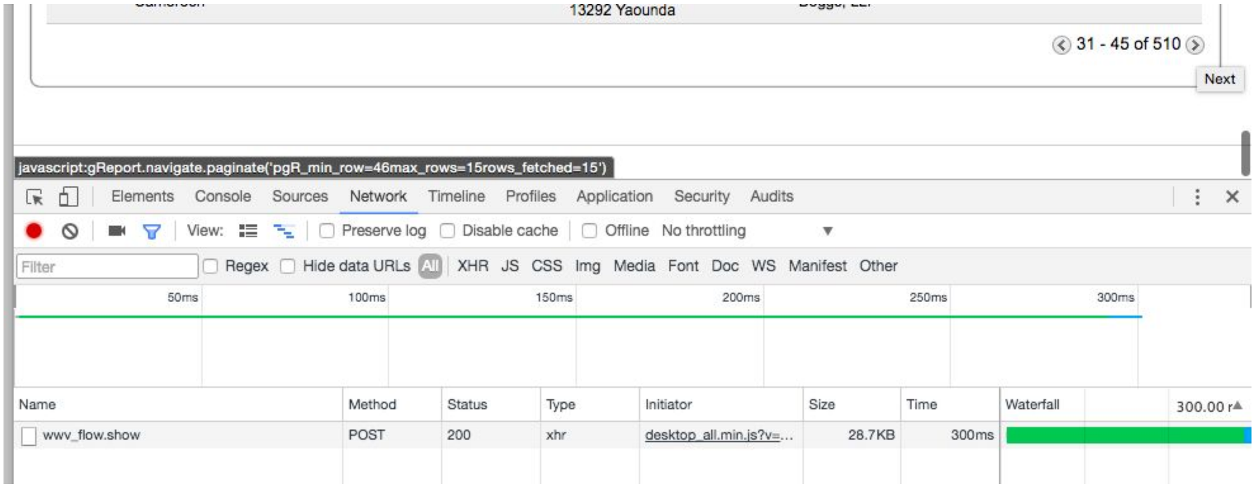
- Terminated Short Form Registrants
- Terminated Short Form Registrants in a
- Historical List of All Short Form Registr

- **FOREIGN PRINCIPALS**
  - Active Foreign Principals
  - Active Foreign Principals in a Date Ran
  - New Foreign Principals in a Date Rang
  - Terminated Foreign Principals
  - Terminated Foreign Principals in a Date
  - Historical List of All Foreign Principals (

https://efile.fara.gov/pls/apex/f?p=171:130:0::NO:RP,130:P130_DATERANGE:N

This url "https://efile.fara.gov/pls/apex/f?p=171:130:0::NO:RP,130:P130_DATERANGE:N" is the starts url in **Scrapy**.

With the startup url set, i inspect the page and she loads a **POST** in javascript for pagination the results, and this **POST** is the navigations pages for **Scrapy**.



```
javascript:gReport.navigate.paginate('pgR_min_row=46max_rows=15rows_fetched=15')
```

Name	Method	Status	Type	Initiator	Size	Time	Waterfall	300.00 r
☐ wwv_flow.show	POST	200	xhr	desktop_all.min.js?v=…	28.7KB	300ms		

The **POST** url "https://efile.fara.gov/pls/apex/wwv_flow.show" navigates on all data pages avaible.

Requirements

Python >= 3.4.3
Scrapy==1.3.1
requests==2.12.4

Scrapy

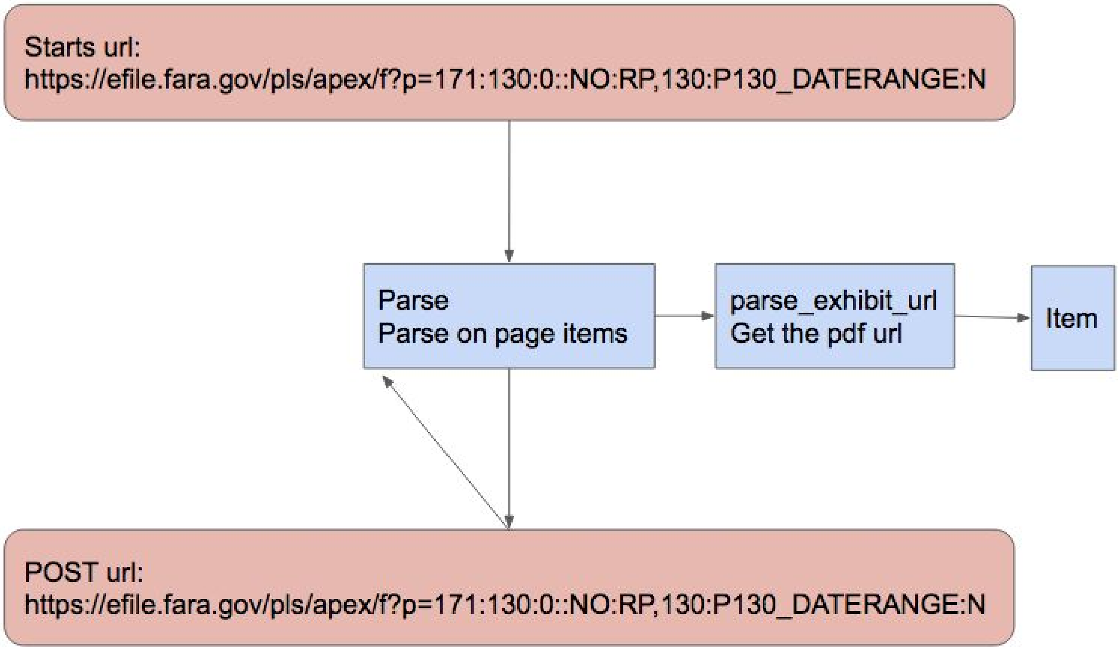
This is a Simplified design off a "**scrapy startproject",** and the class is a simple "**scray.Spider**".
The scrapy is divided in 3 sessions: parse the post data, parse the item data and parse the exhibit_url.

For the item data, see the method "**parse**", and extract from html all principals data ands call for the "**create_item**" and in the end of the list, verify if exists a url available to POST for the next page.

The item has the struct:

```
{
 "address":"  ",
  "country_name":"UNITED ARAB EMIRATES",
  "date":"01/25/2012",
  "exhibit_url":"http://www.fara.gov/docs/6144-Exhibit-AB-20121210-1.pdf",
  "foreign_principal":"Princess Haya Bint Al Hussein",
  "registration":"Hill and Knowlton Strategies, LLC",
  "registration_date":"11/10/1981",
  "registration_number":"3301",
  "state":"",
"url":
"https://efile.fara.gov/pls/apex/f?p=171:200:16319220096257::NO:RP,200:P200_REG_NUMBE
R,P200_DOC_TYPE,P200_COUNTRY:3301,Exhibit%20AB,UNITED%20ARAB%20EMIRATES
"
}
```

In the **create_item** method, for get the **exhibit_url** the scrapy calls the **url** and update the item data with **pdf** document url. The callback update and return the **item.**
After all the data has scraped, the post data has defined in **get_post_data** and if exists redirect for the next data page.


Starts url:
https://efile.fara.gov/pls/apex/f?p=171:130:0::NO:RP,130:P130_DATERANGE:N
Parse
Parse on page items
parse_exhibit_url
Get the pdf url
Item
POST url:
https://efile.fara.gov/pls/apex/f?p=171:130:0::NO:RP,130:P130_DATERANGE:N

Running

For running the project, create a virtualenv with **Python >= 3.4.3.**

Download or clone the project from **github.**
Install the dependencies from **requirements.txt**.
In the path /**myscrapy/faragov/,** run the command **scrapy crawl fara -o faragov.json**
All the data is saved in the **JSON** file.
For running the tests, in the path **/myscrapy/faragov/** run **python3 tests.py.**

For future

Try to run javascript command direct on **scrapy** with **selenium**, **phantom** ou **slash**. To run from the https://www.fara.gov/quick-search.html and the click link in pagination.