**Job requirements**

Scraping job written in Python using Scrapy

**Requirements**:

\* Uses Python3

\* Integrates with Google Sheets

\* Proxy servers to be sourced by parsing results in https://api.nordvpn.com/server

\* Proxy server login to be provided on request

\* Able to use proxies in parallel with Scrapy middleware

\* Proxies to be sourced dy}namically from <https://api.nordvpn.com/server>

\* Proper retry and proxy validation logic is expected as not all proxy servers will be reliable

\* Custom Scrapy middleware to be developed which can set the cookies to be used by the running thread

\* Delay of 5 seconds per request per thread

\* Source code checked in to Github & shared with <http://github.com/gavinong10>

**Private Information**

Once awarded the job, please request access to the following private documents:

[Proxy Information](https://docs.google.com/document/d/12REMorDK66uG8qhzcmY1iXPfp5uj_h38TfVi7UOjByU/edit)

[AWS Account Login Information](https://docs.google.com/document/d/1f1F1GqpEGVZYonhnxyHyKF2WmrdocB49Zb8RfdAKAIk/edit)

**Phase 1:**

Below outlines the manual process that can be followed in a browser in order to scrape the information we require to be scraped.

From the high level, we want to take search criteria and/or links from <https://docs.google.com/spreadsheets/d/1ItXlYbNKUh9buALC-3WObEFJIrXILlCiRBIUJHvX3AA/edit#gid=0>, go to individual property detail pages from <https://pdonline.brisbane.qld.gov.au/MasterPlan/default.aspx> and record the results to a file.

Step 1: Fetch <https://pdonline.brisbane.qld.gov.au/MasterPlan/default.aspx>

A screenshot of a social media post

Description automatically generated

Step 2: “I Agree” POST’s the form to set cookie to navigate the subsequent site.

NOTE: This step should be performed as Scrapy Middleware so it is parallelizable.

Step 3: Navigate to Property Search

<https://pdonline.brisbane.qld.gov.au/MasterPlan/Modules/Enquirer/PropertySearch.aspx>

`A screenshot of a social media post

Description automatically generated

Step 4: Take input from Google Sheets to either perform a search or navigate to a specific link

<https://docs.google.com/spreadsheets/d/1ItXlYbNKUh9buALC-3WObEFJIrXILlCiRBIUJHvX3AA/edit?usp=sharing>

-> If link was not provided, search would produce a list of matching addresses. This list needs to be scraped

A screenshot of a social media post

Description automatically generated

-> If link IS provided, we can skip directly to this step:

<https://pdonline.brisbane.qld.gov.au/MasterViewUI/Modules/PropertyMaster/default.aspx?page=wrapper&Key=636179>

A screenshot of a social media post

Description automatically generated

**Step 5:**

“Agree” to the Terms & Conditions

**Step 6**

Scrape the entire raw HTML of the page and put it in a directory

/scraped\_results/<date>/property/<key>/raw.html

Where *date* is in format like:

2020\_09\_07\_14\_19\_22

(YYYY\_MM\_DD\_HH\_MM\_SS where the scrape first started)

and *key* is the key specified in the URL (above screenshot is 636179)

**Step 7**

Scrape all “Applications” for a property and save in

/scraped\_results/<date>/application/<key>/raw.html

**Phase 2 -** AWS deployment, S3 storage and notifications

Scrapes need to be run on a schedule, running on an AWS Lambda.

Raw scrape data needs to be saved in Amazon S3 instead of local filesystem

We need a Postgres database (please deploy this on a t2.small instance) which records for each application whenever a change is detected:

* The Decision field
* The timestamp of the scrape run when change is detected
* URL of the application
* Application Number (i.e. the key)

A screenshot of a cell phone

Description automatically generated`

Once a change is detected, a notification needs to be sent with a message containing the new information, including specifying what the change of the Decision field was previously and what it became.

Configuration for the database location, user, password, etc. to be deployed on AWS System Manager Parameter Store and used in the code.}