Scraper https://www.metrocuadrado.com/

Requerimientos

Requerimiento	Versión
Python	3.7.7
Pip	20.1.1
Google Chrome	85

Configuración

● Navegar desde la línea de comandos hacia el directorio que contiene el proyecto ○

```
~$ cd FIVERR/fiver/Data\ minig/camiloalbarraci/FO61760D15F37/
```

● Instalar los requerimientos

○ `$ pip -r install requirements.txt`

● Archivos

○ Archivo **./logs/urls.json** (vacío)

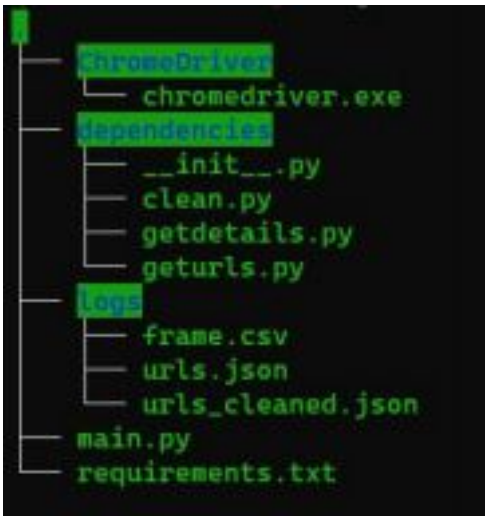
○ Archivo **../logs/urls_cleaned.json** (vacío)

○ Archivo **./logs/frame.csv**

■ Contenido

● [**Codigo Web,Titulo,Barrio,Barrio Catastral,Precio venta,Precio arriendo,Area construida,Area privada,Estrato,Habitaciones,Parqueaderos,Banos,Antiguedad,Interior,Exterior,Zona,Sector,Long,Lat,Direccion,Url**]

Estructura de directorios



● **./ChromeDriver** guarda el driver necesario para controlar Chrome desde python

● **./dependencies** guarda todos los scripts a los cuales el archivo **main.py** hace referencia

  ○ **./dependencies/geturls.py** parte del código específicamente dedicada a extraer las urls específicas de todas las publicaciones en la página dada

  ○ **./dependencies/clean.py** parte del código específicamente dedicada a remover datos duplicados e inservibles que han sido retornados por la anterior parte

  ○ **./dependencies/getdetails.py** parte del código específicamente dedicada a retornar los detalles de una publicación específica y añadirlos al archivo csv en la carpeta **./logs**

● **./logs** guarda los archivos con datos necesarios para hacer el scraping

  ○ **./logs/frame.csv** archivo csv en blanco donde se guardaran todos los datos de las publicaciones.

  ○ **./logs/urls.json** archivo json que guarda todas las urls a publicaciones específicas

    - **./logs/urls_cleaned.json** archivo json que guarda todos los datos de **./logs/urls.json** sin ningún elemento duplicado.

Funciones

- get
    - crea el archivo **urls.json** conteniendo todas las urls de la página web que se le da como parámetro a esta función.
- clean
    - filtra el archivo **urls.json** con el fin de remover todos resultados duplicados, esta función creará el archivo **urls_cleaned.json**
- scrape
    - extrae todos los datos de los registros en el archivo **urls_cleaned.json**, y los añade al archivo csv **frame.csv**

Uso

- Extraer la url deseada de www.metrocuadrado.com/
    - Seleccionar los dos primeros criterios de búsqueda de los diferentes menús desplegables, **escribir** el tercer criterio y hacer click en el botón "**Buscar**"
    - 

    - Copiar la url en la barra buscadora del navegador
    - 

- Función **get**
    - Desde el directorio del proyecto ejecutar el siguiente comando:
        - Estructura
            - `python main.py get [url_copiada]`
        - Ejemplo

- `python main.py get www.metrocuadrado.com/apartamentos-casas/venta/bogota/`

  - Al finalizar este proceso, se verá reflejada la información en el archivo **./logs/urls.json**

- Función **clean**

  - Desde el directorio del proyecto ejecutar el siguiente comando:
    - `python main.py clean`

  - Al finalizar este proceso, se verá reflejada la información en el archivo **./logs/urls_cleaned.json**

- Función **scrape**

  - Desde el directorio del proyecto ejecutar el siguiente comando:
    - `python main.py scrape [true/false]`
      - Los parámetros **true / false** hacen referencia al uso de la API de geocodificación de google

  - Es posible que al iniciar este proceso aparezca siguente error en la línea de comandos
    - Error
      - `Resource blocked, change your internet protocol or try again later`
    - Soluciones
      - Cambiar el protocolo de internet (IP)
        - VPN
        - Cambio de red wifi
      - Esperar un tiempo e intentar de nuevo

  - Al finalizar este proceso, se verá reflejada la información en el archivo **./logs/frame.csv**