

Descripción de solución que realiza el scrap de páginas realizadas en shopify

La presente solución es capaz de hacer un scraper de páginas realizadas en shopify, además extrae la información y la almacena en archivos temporales, a fin de evitar consumir las páginas de una manera constante y evitar ser baneado.

La revisión se hace con base en las colecciones las cuales albergan todos los productos de la página web, posteriormente realiza una comparación de los productos deseados incluida la talla, en caso de encontrar una nueva talla disponible para el producto dado mandará una alerta a discord a un canal configurable, donde se especificará la talla nueva, se brindará la oportunidad de dar click y proceder a realizar una compra de manera automática.

Componentes

La solución se compone de diversos archivos, los cuales son:

- `kihParser.py`, es el archivo orquestador, tiene el flujo a nivel general y a nivel de pruebas medidores de tiempos
- `common.py`, es el archivo con lógica común, se definieron a este nivel los métodos re utilizables para el monitoreo de las páginas deseadas
- `PROPERTIES.py`, es el archivo de configuración general de la aplicación, donde se definen por ejemplo la página a revisar, el nombre de los archivos temporales, las colecciones a monitorear, la url de discord, la cantidad de objetos a comprar, etc.
- `configuracion.csv`, en este archivo se almacena la información de los productos a monitorear, se encuentra compuesto por tres columnas, la colección, la url a monitorear y las tallas
- `collectionFile.csv`, archivo temporal que almacena los productos asociados a una colección
- `newProducts.csv`, archivo temporal que lleva el registro de tallas nuevas detectadas para los productos a monitorear



Casos de uso

En la primera ejecución, se genera el archivo temporal “collectionFile.csv”, el cual tiene la información de todos los productos asociados a las colecciones a revisar, esto para consumir una única ocasión la página web, en vez de una por producto, esta estrategia se comprobó revisando un producto hasta 20, siendo limitado desde los 7 por minuto.

Posteriormente se llevará a cabo la comparación de los productos definidos en “configuracion.csv” para ver si existe alguna nueva talla para el producto dado.

En caso de existir una nueva talla se manda la alerta a Discord y se almacena este cambio en el archivo temporal “newProducts.csv”

Para la segunda ejecución y subsecuentes, el flujo continúa normalmente pero al detectar una coincidencia de una talla que no existía en el archivo de configuración continúa revisando el archivo “newProducts.csv”, en caso de encontrar la nueva talla asociada al producto deseado no se envía la notificación ya que se envió la primera vez, cuando ese registro se creó en el archivo. En caso de encontrar nuevas tallas para el mismo producto u algún otro se agregará el registro correspondiente en el archivo “newProducts.csv” después de haber hecho el envío a Discord

En caso de no encontrar tallas nuevas disponibles para los productos configurados no se enviará ninguna notificación ni se almacenará en dicho archivo.

Requerimientos

Para poder ejecutar el script únicamente es necesario tener instalado Python3 con las librerías siguientes:

```
csv
json
time
urllib.request
urllib.error
os
discord_webhook
```

Las cuales pueden ser instaladas usando pip

Ejecución

Se debe ejecutar el script siguiendo la ruta de instalación de python3 y el nombre del archivo:

```
/bin/python kithParser.py
```

Internamente tiene las referencias a los archivos de configuración, además tiene la suficiente inteligencia para eliminar y actualizar los archivos temporales.