Automated Sneaker Buyer Bot Component Research

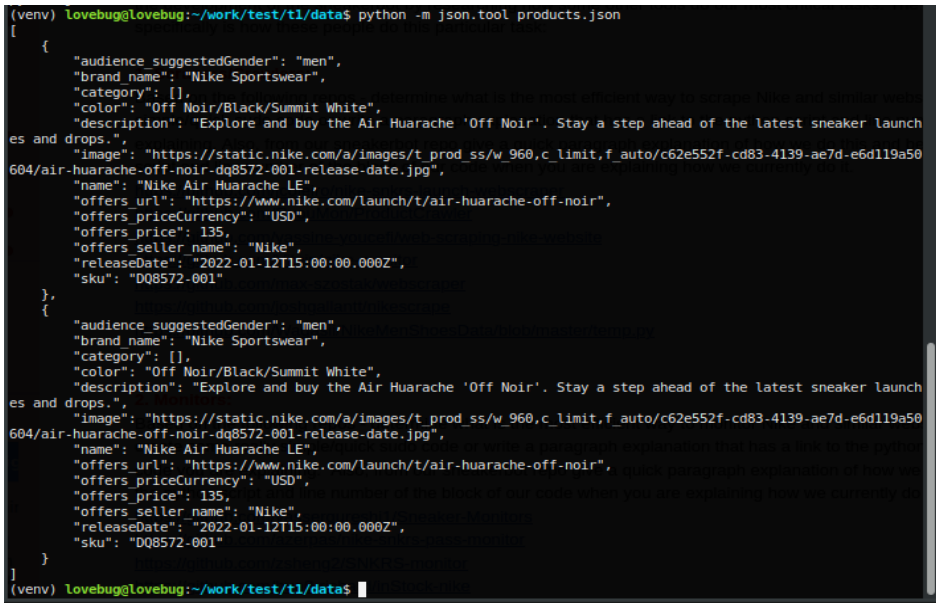
Document is separated by component type.
Author: csr13

---

1) Scraper/Crawler Components

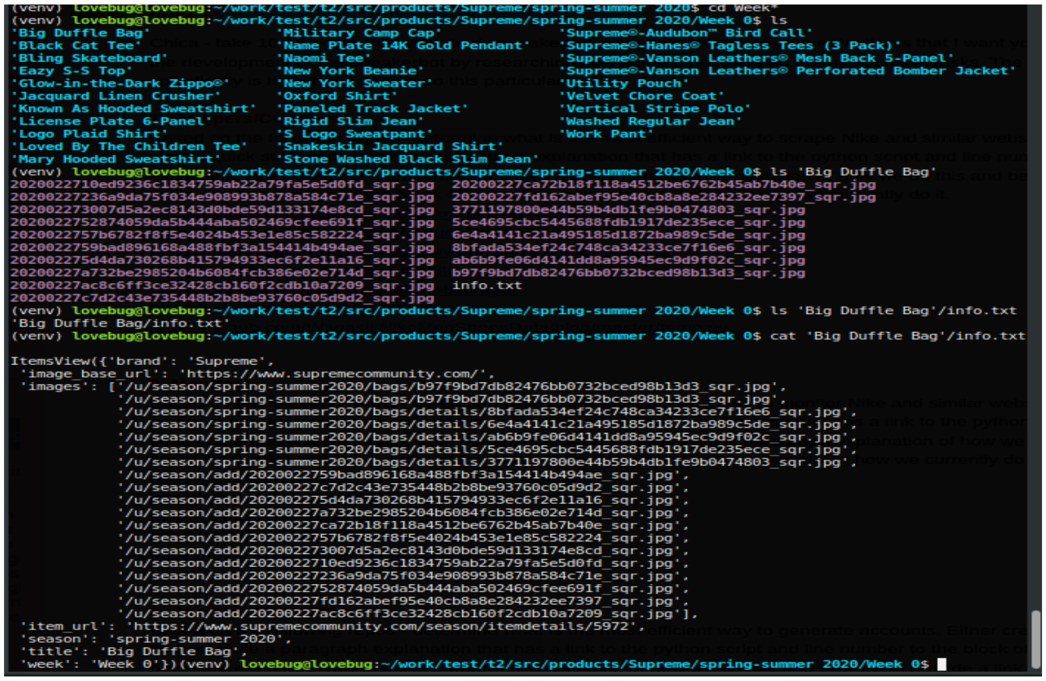
Repositories for scraper/crawler research.

1. https://github.com/jcoleiro/nike-snkrs-launch-webscraper
   1.1. The tool writes output to two files in two different formats, csv and json, json is preferred.
   1.2. This tool only returns a few items in (**feature launch**) this means we need to obtain the actual url for the product to pass to the bot after the release date has been issued (the urls change on some occasions)
   1.3. Example of json output generated by this tool.



```
(venv) lovebug@lovebug:~/work/test/t1/data$ python -m json.tool products.json
[
    {
        "audience_suggestedGender": "men",
        "brand_name": "Nike Sportswear",
        "category": [],
        "color": "Off Noir/Black/Summit White",
        "description": "Explore and buy the Air Huarache 'Off Noir'. Stay a step ahead of the latest sneaker launch
es and drops.",
        "image": "https://static.nike.com/a/images/t_prod_ss/w_960,c_limit,f_auto/c62e552f-cd83-4139-ae7d-e6d119a50
604/air-huarache-off-noir-dq8572-001-release-date.jpg",
        "name": "Nike Air Huarache LE",
        "offers_url": "https://www.nike.com/launch/t/air-huarache-off-noir",
        "offers_priceCurrency": "USD",
        "offers_price": 135,
        "offers_seller_name": "Nike",
        "releaseDate": "2022-01-12T15:00:00.000Z",
        "sku": "DQ8572-001"
    },
    {
        "audience_suggestedGender": "men",
        "brand_name": "Nike Sportswear",
        "category": [],
        "color": "Off Noir/Black/Summit White",
        "description": "Explore and buy the Air Huarache 'Off Noir'. Stay a step ahead of the latest sneaker launch
es and drops.",
        "image": "https://static.nike.com/a/images/t_prod_ss/w_960,c_limit,f_auto/c62e552f-cd83-4139-ae7d-e6d119a50
604/air-huarache-off-noir-dq8572-001-release-date.jpg",
        "name": "Nike Air Huarache LE",
        "offers_url": "https://www.nike.com/launch/t/air-huarache-off-noir",
        "offers_priceCurrency": "USD",
        "offers_price": 135,
        "offers_seller_name": "Nike",
        "releaseDate": "2022-01-12T15:00:00.000Z",
        "sku": "DQ8572-001"
    }
]
(venv) lovebug@lovebug:~/work/test/t1/data$
```

2. https://github.com/RyouMon/ProductCrawler
   2.1. This is a scrapy project, with several 'spiders' this is a list of the following spiders – there are more stores but currently the only ones that carry sneakers are these two:
      2.1.1. Nike
      2.1.2. Supreme
   2.2. Output of the tool

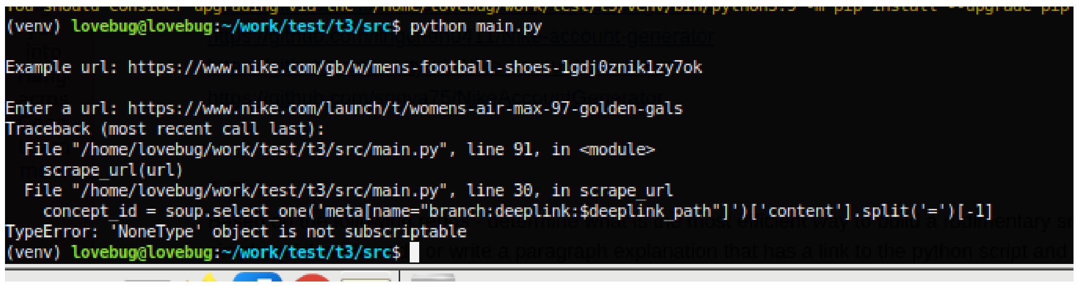
2.2.1. 

```
(venv) lovebug@lovebug:~/work/test/t2/src/products/Supreme/spring-summer 2020$ cd Week*
(venv) lovebug@lovebug:~/work/test/t2/src/products/Supreme/spring-summer 2020/Week 0$ ls
'Big Duffle Bag'                 'Military Camp Cap'             'Supreme®-Audubon™ Bird Call'
'Black Cat Tee'                  'Name Plate 14K Gold Pendant'   'Supreme®-Hanes® Tagless Tees (3 Pack)'
'Bling Skateboard'               'Naomi Tee'                     'Supreme®-Vanson Leathers® Mesh Back 5-Panel'
'Eazy S-S Top'                   'New York Beanie'               'Supreme®-Vanson Leathers® Perforated Bomber Jacket'
'Glow-in-the-Dark Zippo®'        'New York Sweater'              'Utility Pouch'
'Jacquard Linen Crusher'         'Oxford Shirt'                  'Velvet Chore Coat'
'Known As Hooded Sweatshirt'     'Paneled Track Jacket'          'Vertical Stripe Polo'
'License Plate 6-Panel'          'Rigid Slim Jean'               'Washed Regular Jean'
'Logo Plaid Shirt'               'S Logo Sweatpant'              'Work Pant'
'Loved By The Children Tee'      'Snakeskin Jacquard Shirt'
'Mary Hooded Sweatshirt'         'Stone Washed Black Slim Jean'
(venv) lovebug@lovebug:~/work/test/t2/src/products/Supreme/spring-summer 2020/Week 0$ ls 'Big Duffle Bag'
2020022710ed9236c1834759ab22a79fa5e5d0fd_sqr.jpg  20200227ca72b18f118a4512be6762b45ab7b40e_sqr.jpg
20200227236a9da75f034e908993b878a584c71e_sqr.jpg  20200227fd162abef95e40cb8a8e284232ee7397_sqr.jpg
202002273007d5a2ec8143d0bde59d133174e8cd_sqr.jpg  3771197800e44b59b4db1fe9b0474803_sqr.jpg
2020022752874059da5b444aba502469cfee691f_sqr.jpg  5ce4695cbc5445688fdb1917de235ece_sqr.jpg
2020022757b6782f8f5e4024b453e1e85c582224_sqr.jpg  6e4a4141c21a495185d1872ba989c5de_sqr.jpg
2020022759bad896168a488fbf3a154414b494ae_sqr.jpg  8bfada534ef24c748ca34233ce7f16e6_sqr.jpg
202002275d4da730268b415794933ec6f2e11a16_sqr.jpg  ab6b9fe06d4141dd8a95945ec9d9f02c_sqr.jpg
20200227a732be2985204b6084fcb386e02e714d_sqr.jpg  b97f9bd7db82476bb0732bced98b13d3_sqr.jpg
20200227ac8c6ff3ce32428cb160f2cdb10a7209_sqr.jpg  info.txt
20200227c7d2c43e735448b2b8be93760c05d9d2_sqr.jpg
(venv) lovebug@lovebug:~/work/test/t2/src/products/Supreme/spring-summer 2020/Week 0$ ls 'Big Duffle Bag'/info.txt
'Big Duffle Bag/info.txt'
(venv) lovebug@lovebug:~/work/test/t2/src/products/Supreme/spring-summer 2020/Week 0$ cat 'Big Duffle Bag'/info.txt

ItemsView({'brand': 'Supreme',
 'image_base_url': 'https://www.supremecommunity.com/',
 'images': ['/u/season/spring-summer2020/bags/b97f9bd7db82476bb0732bced98b13d3_sqr.jpg',
            '/u/season/spring-summer2020/bags/b97f9bd7db82476bb0732bced98b13d3_sqr.jpg',
            '/u/season/spring-summer2020/bags/details/8bfada534ef24c748ca34233ce7f16e6_sqr.jpg',
            '/u/season/spring-summer2020/bags/details/6e4a4141c21a495185d1872ba989c5de_sqr.jpg',
            '/u/season/spring-summer2020/bags/details/ab6b9fe06d4141dd8a95945ec9d9f02c_sqr.jpg',
            '/u/season/spring-summer2020/bags/details/5ce4695cbc5445688fdb1917de235ece_sqr.jpg',
            '/u/season/spring-summer2020/bags/details/3771197800e44b59b4db1fe9b0474803_sqr.jpg',
            '/u/season/add/2020022759bad896168a488fbf3a154414b494ae_sqr.jpg',
            '/u/season/add/20200227c7d2c43e735448b2b8be93760c05d9d2_sqr.jpg',
            '/u/season/add/202002275d4da730268b415794933ec6f2e11a16_sqr.jpg',
            '/u/season/add/20200227a732be2985204b6084fcb386e02e714d_sqr.jpg',
            '/u/season/add/20200227ca72b18f118a4512be6762b45ab7b40e_sqr.jpg',
            '/u/season/add/2020022757b6782f8f5e4024b453e1e85c582224_sqr.jpg',
            '/u/season/add/202002273007d5a2ec8143d0bde59d133174e8cd_sqr.jpg',
            '/u/season/add/2020022710ed9236c1834759ab22a79fa5e5d0fd_sqr.jpg',
            '/u/season/add/20200227236a9da75f034e908993b878a584c71e_sqr.jpg',
            '/u/season/add/2020022752874059da5b444aba502469cfee691f_sqr.jpg',
            '/u/season/add/20200227fd162abef95e40cb8a8e284232ee7397_sqr.jpg',
            '/u/season/add/20200227ac8c6ff3ce32428cb160f2cdb10a7209_sqr.jpg'],
 'item_url': 'https://www.supremecommunity.com/season/itemdetails/5972',
 'season': 'spring-summer 2020',
 'title': 'Big Duffle Bag',
 'week': 'Week 0'})(venv) lovebug@lovebug:~/work/test/t2/src/products/Supreme/spring-summer 2020/Week 0$
```

The benefit of using this is the amount of images we can get for the product.

3. https://github.com/yassine-youcefi/web-scraping-nike-website
    3.1. Seems like a huge project, but cannot run it at the moment, have to come back to it, did not like that the devs included the entire site packages (requirements.txt) dependencies.
4. https://github.com/Vitnin/nike-monitor
    4.1. Interesting structure/concept, but it is used for nike.br (brazil)
5. https://github.com/max-szostak/webscraper
    5.1. Runs with selenium.
    5.2. 0 stars, no forks (pending review)
6. https://github.com/joshgallantt/nikescrape
    6.1. Single script, It requires manual input of the url. Fails at execution.



```
(venv) lovebug@lovebug:~/work/test/t3/src$ python main.py

Example url: https://www.nike.com/gb/w/mens-football-shoes-1gdj0zniklzy7ok

Enter a url: https://www.nike.com/launch/t/womens-air-max-97-golden-gals
Traceback (most recent call last):
  File "/home/lovebug/work/test/t3/src/main.py", line 91, in <module>
    scrape_url(url)
  File "/home/lovebug/work/test/t3/src/main.py", line 30, in scrape_url
    concept_id = soup.select_one('meta[name="branch:deeplink:$deeplink_path"]')['content'].split('=')[-1]
TypeError: 'NoneType' object is not subscriptable
(venv) lovebug@lovebug:~/work/test/t3/src$
```

7. https://github.com/Waqasii/NikeMenShoesData/blob/master/temp.py
    7.1. Not valuable for us.

Best way to integrate, salvage, or make use of the scraper repositories.

Decide which ones to use, I personally think we can get some value (data wise) on repository 1 and repository 2. We can call these scripts from a cron job, have the application pick these up from a predefined location, parse them and store them in the database for displaying.

---

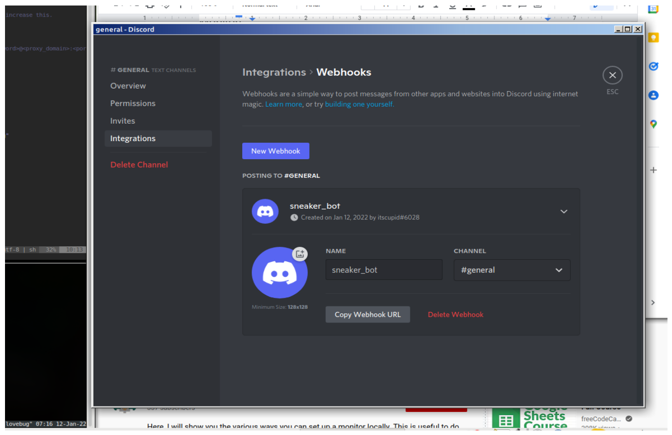
2) Monitors

How we do it

We moved away from that concept, and instead relied on cron batches of the scrapper.

Repositories for the monitor component.

1. https://github.com/yasserqureshi1/Sneaker-Monitors - FAILS
   1.1. Requires a discord webhook url - https://discord.com/api/webhooks/930812046017187850/AnlQTfwqQbnm89cfM3P8ITsc89N2Xaq3oz8Z-phb1D75FjrF-lPk161HHk-ab5qdKdVo



general - Discord

# GENERAL TEXT CHANNELS

Integrations > Webhooks

Webhooks are a simple way to post messages from other apps and websites into Discord using internet magic. Learn more, or try building one yourself.

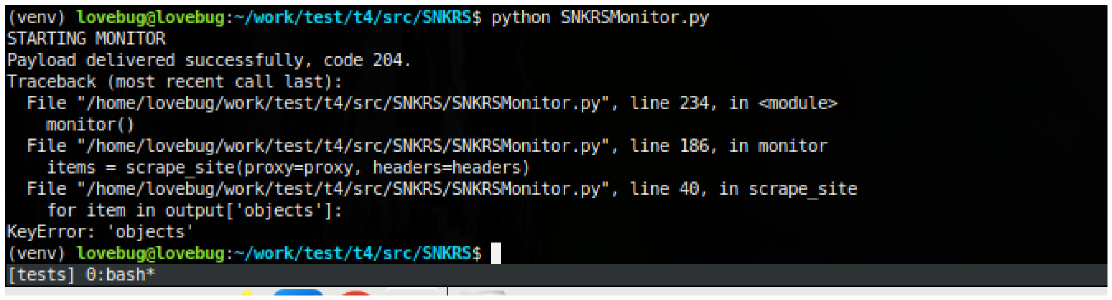
POSTING TO #GENERAL

sneaker_bot
Created on Jan 12, 2022 by itscupid#6028

NAME: sneaker_bot
CHANNEL: #general

Minimum Size: 128x128

   1.2. Only reports to the discord server, needs to stay running so it can keep reporting information about new releases.
   1.3. The execution of the monitor fails with the US site for Nike.

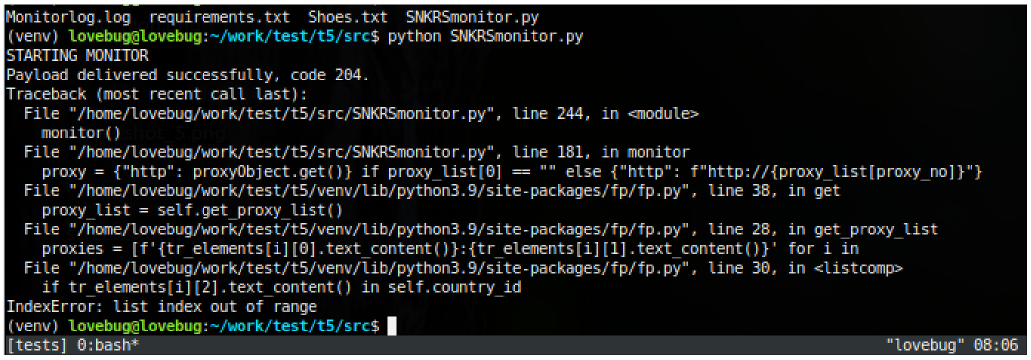


```
(venv) lovebug@lovebug:~/work/test/t4/src/SNKRS$ python SNKRSMonitor.py
STARTING MONITOR
Payload delivered successfully, code 204.
Traceback (most recent call last):
  File "/home/lovebug/work/test/t4/src/SNKRS/SNKRSMonitor.py", line 234, in <module>
    monitor()
  File "/home/lovebug/work/test/t4/src/SNKRS/SNKRSMonitor.py", line 186, in monitor
    items = scrape_site(proxy=proxy, headers=headers)
  File "/home/lovebug/work/test/t4/src/SNKRS/SNKRSMonitor.py", line 40, in scrape_site
    for item in output['objects']:
KeyError: 'objects'
(venv) lovebug@lovebug:~/work/test/t4/src/SNKRS$
[tests] 0:bash*
```

2. https://github.com/azerpas/nike-snkrs-pass-monitor

    2.1. AWS Lambda code, will return to it later.

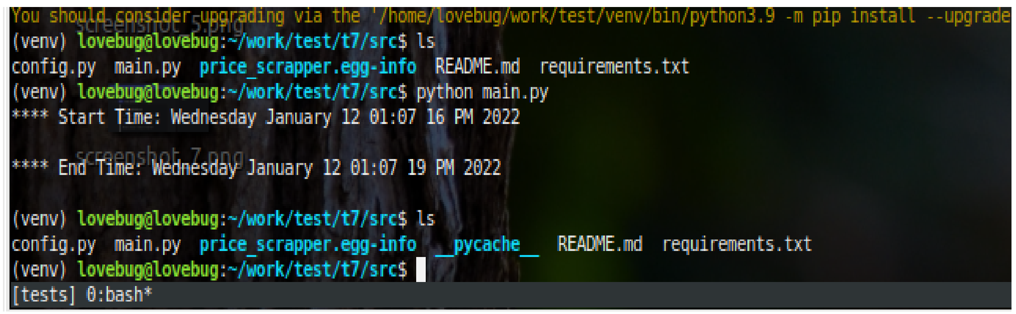
3. https://github.com/zsheng2/SNKRS-monitor
    3.1. Single script, looked well on code, failed at execution.



```
Monitorlog.log  requirements.txt  Shoes.txt  SNKRSmonitor.py
(venv) lovebug@lovebug:~/work/test/t5/src$ python SNKRSmonitor.py
STARTING MONITOR
Payload delivered successfully, code 204.
Traceback (most recent call last):
  File "/home/lovebug/work/test/t5/src/SNKRSmonitor.py", line 244, in <module>
    monitor()
  File "/home/lovebug/work/test/t5/src/SNKRSmonitor.py", line 181, in monitor
    proxy = {"http": proxyObject.get()} if proxy_list[0] == "" else {"http": f"http://{proxy_list[proxy_no]}"}
  File "/home/lovebug/work/test/t5/venv/lib/python3.9/site-packages/fp/fp.py", line 38, in get
    proxy_list = self.get_proxy_list()
  File "/home/lovebug/work/test/t5/venv/lib/python3.9/site-packages/fp/fp.py", line 28, in get_proxy_list
    proxies = [f'{tr_elements[i][0].text_content()}:{tr_elements[i][1].text_content()}' for i in
  File "/home/lovebug/work/test/t5/venv/lib/python3.9/site-packages/fp/fp.py", line 30, in <listcomp>
    if tr_elements[i][2].text_content() in self.country_id
IndexError: list index out of range
(venv) lovebug@lovebug:~/work/test/t5/src$
[tests] 0:bash*                                                            "lovebug" 08:06
```

4. https://github.com/hua0-richard/inStock-nike
    4.1. Not something we could use, at this moment

5. https://github.com/qodetinker/price-scraper
    5.1. Able to run this repository, it runs with a config.py file of a list of urls that point to the specific product, in practice we are already storing url + price of the shoe. Still wanted to test it but the script yielded no output.



```
You should consider upgrading via the '/home/lovebug/work/test/venv/bin/python3.9 -m pip install --upgrade
(venv) lovebug@lovebug:~/work/test/t7/src$ ls
config.py  main.py  price_scrapper.egg-info  README.md  requirements.txt
(venv) lovebug@lovebug:~/work/test/t7/src$ python main.py
**** Start Time: Wednesday January 12 01:07 16 PM 2022

**** End Time: Wednesday January 12 01:07 19 PM 2022

(venv) lovebug@lovebug:~/work/test/t7/src$ ls
config.py  main.py  price_scrapper.egg-info  __pycache__  README.md  requirements.txt
(venv) lovebug@lovebug:~/work/test/t7/src$
[tests] 0:bash*
```

Best way to integrate, salvage, or make use of the monitoring repositories.

We could use the repository 1 with the discord webhook and extend it to become the trigger notification for some functionality that is needed later, user notification related or something similar, I just don't see how the other would help us.

---

3) Automated Account Generation – Nike.

How we do it.

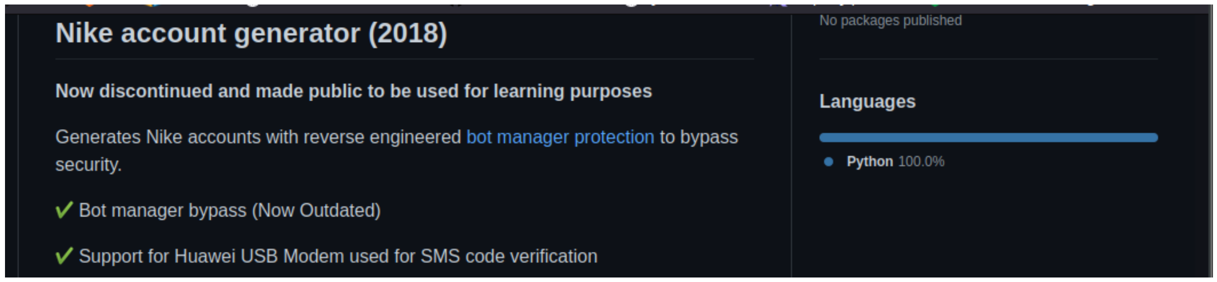
We initially had some boiler code, it did not function properly and we spent 3 + months on it. Then it was decided we could buy accounts, and since there was going to be manual involvement in the process of the workflow of the application, this subject was left open to research.

How other products do it

The users input their nike credentials into the platform.

Repositories for Nike Account Generation.

1. https://github.com/junhuang21484/Nike-Account-Generator
    1.1. This is a cli tool that prompts the user to execute tasks, it uses playwright, some type of selenium on steroids - it requires some extra configuration, returning to it because it seems usable, just not sure about
2. https://github.com/niewiemczego/nike-snkrs-atc-generator
    2.1. This is something related to atc generation not account generation, but it seems like it serves a purpose: it claims to help the purchase action be faster because it generates the early purchase url. The input is manual, if the claims are right it can be modified to receive information input via code.
3. https://github.com/lingchen0411/Nike-account-generator
    3.1. This one looks promising, it has some functionality to bypass headless detection. I will be setting it up on my machine and running it
    3.2. Requires selenium.
4. https://github.com/martyurb/Nike-account-generator
    4.1. Now outdated for bypassing bot protection.



Nike account generator (2018)

Now discontinued and made public to be used for learning purposes

Generates Nike accounts with reverse engineered bot manager protection to bypass security.

✓ Bot manager bypass (Now Outdated)

✓ Support for Huawei USB Modem used for SMS code verification

    4.2. Most of the code itself is very similar to most bots, using selenium.
5. https://github.com/sonya75/NikeAccountGenerator
    5.1. We have inspected, tested and concluded on this repository, it is very creative, the best one for generating accounts, unfortunately it is no longer of value to us, the nike account generation rules and detection software has changed since this script landed on github.

Best way to integrate, salvage, or make use of the monitoring repositories.

---

4) Bots.

How we do it.

We have a repository on keybase, you have access to it, it includes the code for the purchase bot, we also have it on the aws snapshot of the proxy because this is the ‘bot’ and inside the proxy aws instance we are currently using a modified version of this bot https://github.com/samc621/SneakerBot

Repositories for bots.

1. https://github.com/Daniel-Hinds/Sneaker-Bot-UK
   1.1. This repository is a single selenium script with a link collector for twitter, in order to get the shoe url, and only for the UK store, we are currently using a modified version of this bot https://github.com/samc621/SneakerBot

Best way to integrate, salvage, or make use of the bot repositories.

We have to update the current bot because it looks like there have been some updates to the code, and we have also modified it. I cannot reference it because we need a repo for it, then I can drop it there and reference this better.

---

5) Captcha Bypass.

How we do it.

For captcha bypass I believe we use 2captcha on the bot (see section 4) we gotta be able to move the bot code from the snapshot to the repository.

Repositories

1. 2Captcha
   1.1. https://2captcha.com/2captcha-api
      1.1.1. This is the service we used before for getting past captchas.
   1.2. https://github.com/2captcha/2captcha-python
      1.2.1. Same thing as (1) but it is the library, this has the best support because it supports many types of google captcha types.
   1.3. https://github.com/Zaczero/2CaptchaBest for our bot since our bot is node js based

2. Imagetyperz
    2.1. It can be used, it has support for many languages, from working with anticaptcha before I've found that the most important aspect is time of completion, for the person doing the challenge, so we would have to see which one is faster, 2Captcha (1) is the best choice here
    2.2. https://www.imagetyperz.com/Forms/api/api.html
    2.3. https://github.com/imagetyperz-api/imagetyperz-api-python3
    2.4. https://github.com/theY4Kman/imagetyperz-async
3. Anti-Captcha
    3.1. https://anti-captcha.com/apidoc
4. Slider Captcha
    4.1. https://github.com/jchen293/bypass_slider_captcha
5. Recaptcha
    5.1. https://github.com/jchen293/bypass_slider_captcha
6.