Tools to scrape social media, process GPS Data and host web apps/sockets

1. Instagram scraper

- 1. Download or clone repository through link provided here: <u>https://github.com/andresvillatorres/</u> instagram-scraper-iad-zhdk
- 2. Follow the instructions on the GitHub repository or follow the next instructions (recommended for Mojave 10.14.6)
- 3. Install this version of python: 3.7.4 (https://www.python.org/downloads/)
- 4. Get pip3 : https://pip.pypa.io/en/latest/installing/
- 5. With python3.7 installed, install latest pip version: \$ python3.7 -m pip install --upgrade pip
- 6. With pip3.7 install following dependencies:
 - 1. \$ sudo pip3.7 install requests==2.21.0
 - 2. \$ sudo pip3.7 install python-slugify==3.0.2
- You should be ready to run your test python script to scrape content containing certain tags. Go to the tests folder over the terminal and run: \$ python3.7 test_scraping_tags.py
 - Remember replacing your login credentials [line 6] --> instagram.with_credentials('username', 'password')
 - You can edit the file "test_scraping_tags.py" and modify the targeted tag and the amount of results you want to get per query [line 12] — -> thisMedias=instagram.get medias by tag("icecream",25,")

2. Google location history KML format

- 1. Download and install processing and all the necessary libraries
- 2. Download your location history from google takeout in KML format
- 3. Download the following sketch from processing through the link provided: <u>https://github.com/</u> <u>andresvillatorres/google_location_history</u>
- 4. Place your location_history.kml file inside the "data" folder in your processing sketch and edit the name of the file inside the sketch [line 48]
- 5. Run the sketch on processing

3. socket.io + localhost.run

- 1. Download and install node js : <u>https://nodejs.org/en/download/current/</u>
- 2. With "npm" install socket.io over the Terminal : \$ npm install socket.io
- 3. Download or clone socket.io through the following link: https://github.com/socketio/socket.io
- 4. Go on the terminal to the path <u>socket.io</u>-master/examples/whiteboard ("<u>socket.io</u>-master/ examples/whiteboard"), we will call this your web application repository.
- 5. Now start the npm server there providing the following command: \$ npm start
- 6. You should see the following message "listening on port 8080" and you should be able to access the broadcasted web app with the open web sockets on any browser from any device connected to the local network over : http://"your ip address":8080/ —> <u>http://192.168.1.44:8080/</u>. Try it out with your laptop or desktop and your smart phone. **** Be aware that you might need to change the port inside your index.js file inside your whiteboard folder <[line 6] : const port = process.env.PORT || 8080 > to be 8080 instead of 3000.

7. Now, a nice trick is to broadcast this locally hosted web application by doing a port forwarding with your own router. This will allow you to host your own web application which can be a powerful tool for prototyping, developing or hosting a temporary website or application without the need of any external host provider and the need of a domain. This can be done by accessing the configuration settings from your router providing the IP address 192.168.1.1 on a browser in a computer connected to the router's network. Depending on your router this can be set differently. As an example you can access this settings on a *fiber salt router* by logging into settings after providing this address "192.168.1.1" in the navigation bar of your browser. In expert modus go to Network >NAT > Port Mapping and click on [ADD+]. Here some images so you can see what are the settings that worked for me.

Salt.	Power saving LAN- Dynamic DNS NAT- Firewall- LTE Config LTE Status IPv6+	Network > Po	Network > Port Mapping - #1 (Edit)						
Overview	Network > Port Mapping	Enable this rule	Enable this rule						
Network	You can configure the score as a white anews on the termine same accessing particles such as the VMa or YTFP at your Used SNA via public IP addresses can be ave redenced to local servers configured with private IP addresses. In coller words, depending on the requested service ITCPFUDP part numberit, the notar redences service request to the appropriate some docated as a worden termine IP address.	e external Private IP	192 . 168 . 1 . 44						
WLAN	Port Mapping List (Max Limit : 32)	Protocol	TCD8.						
Telephony	No. Status Private IP Protocol Private Port Public Port Configure	FIOLOCOI	ICFA						
USB	1 ✔ 192.168.1.44 TCP&UDP 8080 8080 2 10 10 10 10 10 10 10 10 10 10 10 10 10	Private Port	8080 - 8080						
Parental control	Cancel	Public Port	8080 - 8080						
Administration			Save settings Cancel						

- 8. Now your local IP is being forwarded and listening on the ports 8080. You can also change the private and public ports to be different from one another. Now you need to find out what is your external IP address. This can be done following this link <u>https://www.yougetsignal.com/tools/open-ports/</u> or over the terminal with the following command \$ curl <u>https://ipinfo.io/ip</u>
- 9. Once you have this, you should be able to access the broadcast web socket through 51.154.217.59:8080
- 10. Another alternative is to use an external, free service which forwards your local ip to the world and generates a unique link. This service is called <u>localhost.run</u>. Here the only step to follow after having opened the npm server inside your application folder, is to open a new terminal instance, copy, paste and commit the following command: \$ ssh -R 80:localhost:8080 ssh.localhost.run>. This will generate a temporary link forwarding your locally hosted web app.
- 11. Now you should be able to access your local hosted web app through the link generated, i.e. <u>https://yourmachineor</u>username-fmhu.localhost.run. And the cool thing about this is that both solutions work independently from one another.
- 12. ***** Don't forget killing your broadcasting processes once you are done (with "cmnd" + "." at the Terminal) and disabling the port forwarding when not necessary. Keep in mind that this leave some vulnerabilities open.
- 4. Realtime location using processing, custom map and GYROSC
 - 1. Download or clone the repository through link provided here: <u>https://github.com/andresvillatorres/</u> google_location_history

- 2. Download and install processing and all libraries necessary (osc p5)
- 3. Download and install GYROSC
- 4. Connect laptop and phone to the same network (not ZHdK Network) or create a hotspot with your phone
- 5. Provide the correct IP Adress and selected listening port (12000) from your laptop into your GYROSC App
- 6. Be sure your GYROSC is broadcasting your GPS signal and also running on the background.
- 7. Be sure to close the app once you stop using it, otherwise it will drain your battery power.
- 8. Run the processing sketch.
- 9. *** cool extra Feature : guess what? the cool thing of forwarding your IP (*look at the previous part of the tutorial with the salt router as example for enabling that*) is that you can send your sensor data from the GYROSC App to your remote locations, addressing your external IP address to which your local IP and open ports are being forwarded. This can be helpful in order to keep track of your measured data remotely :) ***** remember to change the listening port on your processing application from 12000 to 8080 or whatever to choose as the public port. Here a parsing example on processing is provided, which can be found inside the repository. In the image bellow you can see some settings configuration on the gyrosc app addressing an external IP.



5. Tinder Scraper

- 1. Currently developing this as part of a PhD Project, at the moment this is a very basic, sneaky but powerful scraper.
- 2. Download or clone repository through the link provided here: <u>https://github.com/andresvillatorres/</u> <u>tinder light_scraper</u>
- 3. In order to use the Tinder Scraper you need to login to the network. And for that you need an account and get your access token. Here are two methods:

 With inspector tool: open Tinder on Firefox. Open inspector with right clicking on mouse or trackpad. Select the "Network" tab. Look for the XHR GET or POST request and unfold one, scroll down until you find the X-Auth-Token which should look something like this: 24762xxxxxxx-xxxx-xxxxdf213316d.

${\bf F}$	ſ	D Inspecto	or 👂	Console	D De	bugger	↑↓ Netwo	rk {}	Style Edi	or 🕜 Per	formance	C Memor	y 🗄 Storage	🕇 Acce	essibility		j
Ŵ	7	Filter URL	S										Persist Logs	Disable c	ache	No throttling	\$ HAR \$
All	H	HTML CS	s JS	XHR Fo	onts Ima	ges Mec	ia WS C)ther									
Statu	s	- Me	Domain	File	Cause	Ty; Tran	s Size	ms	¦1 🖻	Headers	Cookies	Params	Response	Cache	Timings	Stack Tra	ce S-
200		G	ima	172x21	. img	jp serv	ic 5.6		Rei /6/	uest URL: ht	ttps://ima 12b-5fdf-	ages-ssl.go -4f76-8750-	tinder.com/5c	b82c6bbda3	261600ad	ae3c	
200		G	🔒 ima	172x21	. fetch	jp 6.08	kB 5.6	38 ms	Re	uest method	: GET	4170 0750	c919902002001	163			
200		G	🔒 ima	640x80.	img	jp 39.9	0 39	28 ms	Rei	note address	: 13.224.9	06.49:443					
Ō	4	80 requests	3 7.4	4 MB / 3.5	9 MB tran	sferred	Finish: 23.1	3 min	DOM	tus code: 30	4 Not Mod	dified (?)					
Ŵ	5	Filter outp	out						UA.	Errors	Warnings	Logs Info	Debug CSS		quests	🗌 Persist L	ogs 🗙
•	XHI	R GEI htt	ps://ap	pi.gotin	der.com/	pass/5da	a1a511433	501006	/1d†0?loc	a Le=en-GB&s	_number=8	3196/418			[HTTP/2	.0 200 OK	1//msj
	[Service Worker] Received message SUSPEND_SW_NOTIFICATIONS												:1:4492				
•	XH	R OPTIONS	https	://etl.t	inderspa	rks.com/	v2/batch/	event							[HTTP/1	.1 200 OK	[136 ms]
	[Se	ervice Wo	rker] F	Received	message	SUSPEND	_SW_NOTIF	CATIONS	5						service	-worker.js	:1:4492
►	XH	R PUT http	ps://e	tl.tinde	rsparks.	com/v2/b	atch/even [.]	:							[HTTP/1	.1 200 Ok	[125ms]
_	[Se	ervice Wo	rker] P	Received	message	SUSPEND	_SW_NOTIF	CATIONS	S					25	service	-worker.js	:1:4492
•	XH	R POST ht	tps://a	api.goti	nder.com	/updates	?locale=e	n-GB							[HTTP/2	.0 200 OK	(140m s]
	ŀ	leaders	Cook	ies Pa	arams	Respons	e Timir	ngs S	stack Tra	ce Secur	ity						
		2 Conten	t-Leng	en os	ation/is	0.0											
	() Content-Type, application/jsoll																
	Operation by the second s																
	persistent-device-id: b374da00-e898-4115-94d0-4594730b8427																
		platfo	rm: wel	b													
	(? Refere	r: htt	ps://tin	der.com/												
		tinder	-versi	on: 2.14	.0												
	⑦ User-Agent: Mozilla/5.0 (Macintosh; Intel) Gecko/20100101 Firefox/70.0 user-session-id: f35f5050-f6f0-4419-a6d8-b97e8a9b7dfa																
		user-s	ession	-time-el	apsed: 6	2257											
		X-Auth	-Token	: 247628	ed-39a5-	-	-8d4df21	3316d									
		x Supp	orted	image fo	rmatsi j	beg											

- 2. With sms verification. Run the python script sms_auth.py inside utils folder. Provide your phone number. Provide the authorisation code you'll get through sms. It should give you back the same X-Auth-Token.
- 4. Once you got your auth token paste it inside the basic_example.py script on [line 17] ---> token = "24762xxx-xxxx-xxxx-xxxdf213316d".
- 5. Save it and run the script on Terminal at the right folder path with: \$ python3.7 basic_example.py
- 6. The script should print out name, age, gender, distance to you and the bio if available. Be aware that you might need to update or install modules that may be missing. Errors in Terminal are self explaining and will guide you through.

Alright, that's it. Don't hesitate contacting me if any questions: andres.villa_torres@zhdk.ch