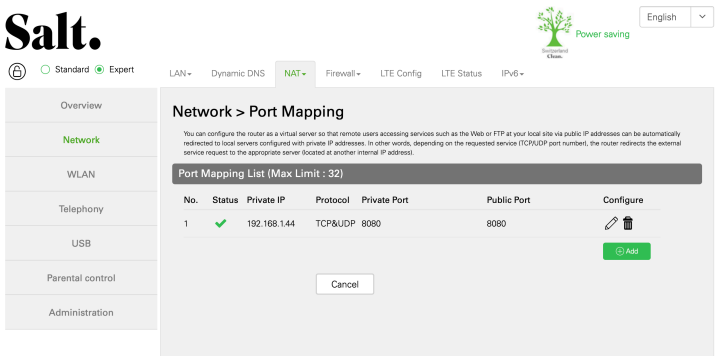
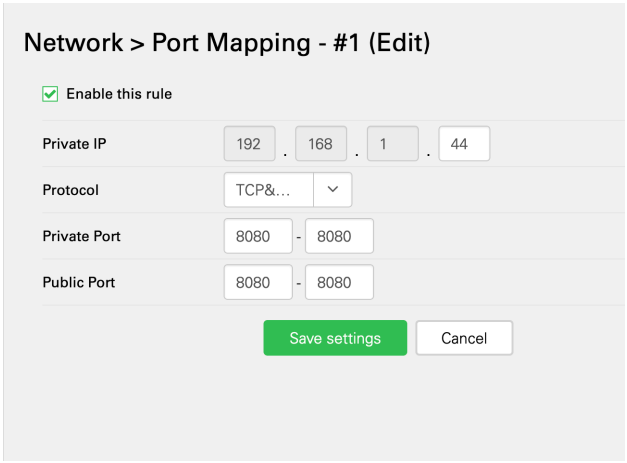
Tools to scrape social media, process GPS Data and host web apps/sockets

1. **Instagram scraper**
    1. Download or clone repository through link provided here: https://github.com/andresvillatorres/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
    7. 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***
        1. Remember replacing your login credentials [line 6] ——> instagram.with_credentials(‘username', ‘password')
        2. 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: https://github.com/andresvillatorres/google_location_history
    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 : https://nodejs.org/en/download/current/
    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 socket.io-master/examples/whiteboard ("socket.io-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/ ——> http://192.168.1.44:8080/ . 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.



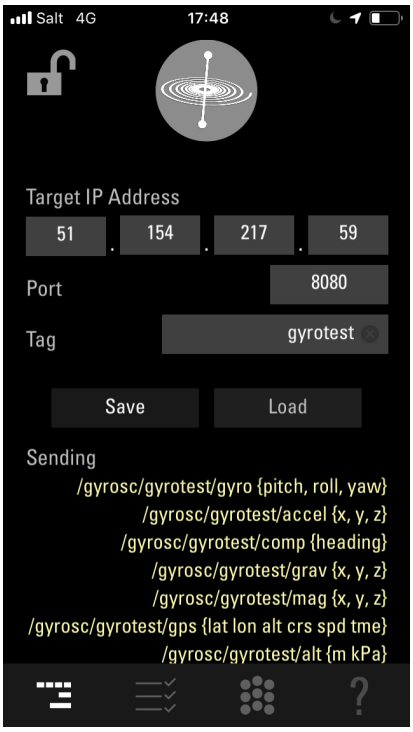


8. Now your local IP is being forwarded and listening on the ports 8080. You can also change the private and public ports to be different from one another. Now you need to find out what is your external IP address. This can be done following this link https://www.yougetsignal.com/tools/open-ports/ or over the terminal with the following command $ **curl https://ipinfo.io/ip**
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 localhost.run. 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. https://yourmachineorusername-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: https://github.com/andresvillatorres/google_location_history

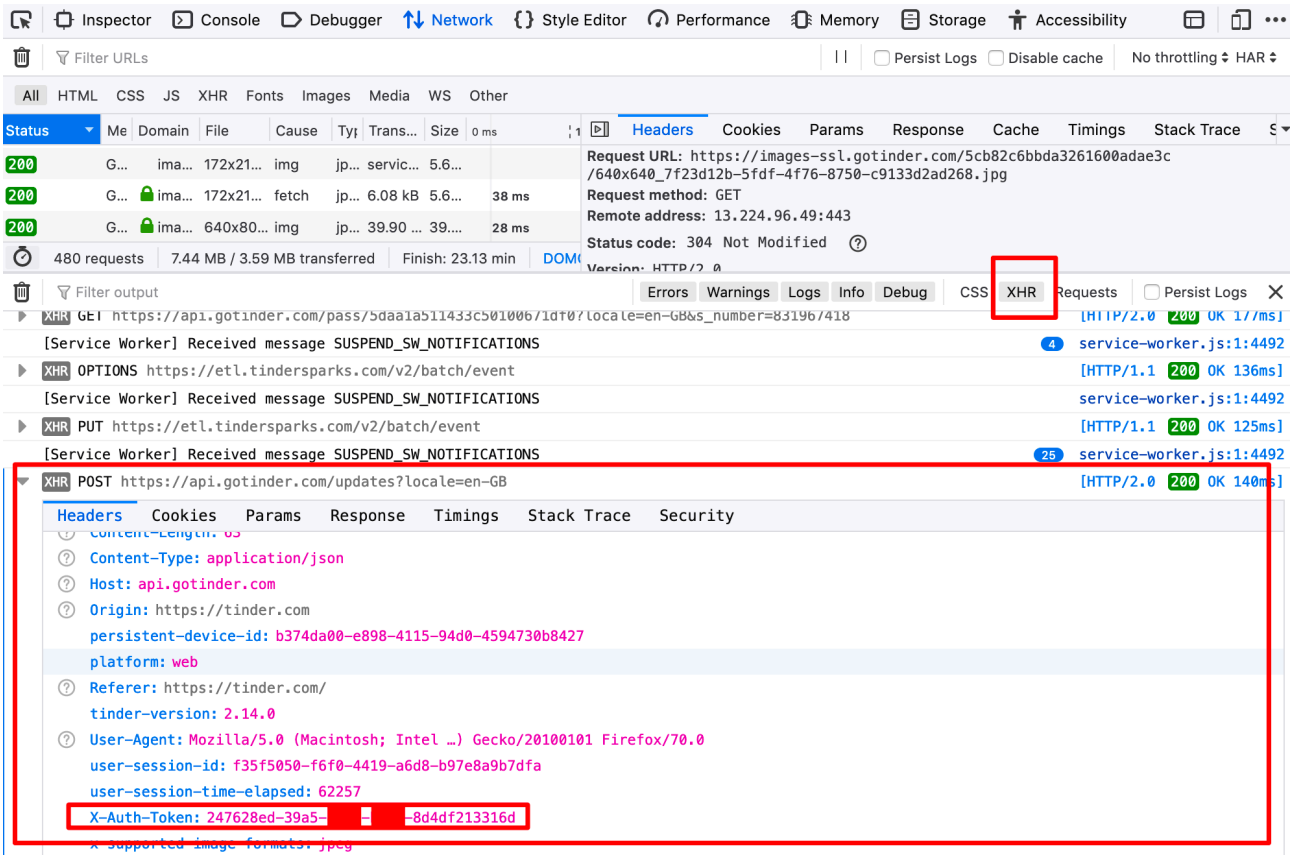
2. Download and install processing and all libraries necessary (osc p5)
3. Download and install GYROSC
4. Connect laptop and phone to the same network (not ZHdK Network) or create a hotspot with your phone
5. Provide the correct IP Adress and selected listening port (12000) from your laptop into your 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: https://github.com/andresvillatorres/tinder_light_scraper
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:

1. 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: 24762xxx-xxxx-xxxx-xxxx-xxxdf213316d.



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-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**