

Forum:[Hack Forums](#)

Thread: [Efficient and Solid way to Upload or Send Large Files??](#)

SIM-4791248 - February 9, 2021, 12:48 am

i've been trying to auto upload without selenium large files random sizes up to 400 MB

but all not working with persistence (telegram - discord - dropbox ...) ? so the only solution is ftp transfer or there is ways around ? in windows not linux ?

using python +3 is there any solutions for fast upload large sized files ...

Thanks

Pernat1y - February 9, 2021, 3:45 am

Google drive, dropbox, mega. All have SDKs or example clients in Python.

FogFox - February 12, 2021, 6:53 am

Why dont host the files yourself?

Qismon - February 16, 2021, 9:36 am

You can upload direct using SSH to sFTP.

```
Code import os
import sys
import time
import paramiko
```

```
class ControlSSH:
```

```
    """
```

```
    SSH Class.
```

```
    """
```

```
    def __init__(self):
```

```
        self.hostPort = 22
```

```
        self.sshTimeout = 5
```

```
    def ssh_connect(self, sshData):
```

```
        """
```

```
        Connect to SSH.
```

```
        """
```

```
        hostDomain = sshData[0]
```

```
        sshUsername = sshData[1]
```

```
        sshPassword = sshData[2]
```

```
        sshClient = paramiko.SSHClient()
```

```
        sshClient.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
try:
    sshClient.connect(hostDomain, port=self.hostPort, username=sshUsername,
password=sshPassword, timeout=self.sshTimeout)
except:
    sshClient = None

return sshClient
```

```
def ssh_close(self, sshClient):
    """
    Close SSH session.
    """
    return sshClient.close()
```

```
def ssh_execute(self, sshClient, sshCommand):
    """
    Execute SSH remote command.
    """
    sshInput, sshOutput, sshError = sshClient.exec_command(sshCommand, environment=None)
    return sshOutput
```

```
def ssh_sendfile(self, sshClient, localFile, remoteFile):
    """
    Send file over SFTP.
    """
    try:
        secureFTP = sshClient.open_sftp()
        secureFTP.put(localFile, remoteFile)
        secureFTP.close()
        sendStatus = True
    except IOError as sftpError:
        sendStatus = False

    return sendStatus
```