

# SumoLogic App for DropBox Business Installation Instructions

1. Choose a server that is running a SumoLogic collector. This server will need to access the dropbox API over the internet.
2. Install Python 2.7
3. Download and install the Python SDK for DropBox from:  
<https://www.dropbox.com/developers/core>
4. Create a DropBox for Business App. Go to  
<https://www.dropbox.com/developers/apps/create/business>
  - a. Choose Team Auditing for permissions
  - b. Provide an App Name (it needs to be unique)
  - c. Note the App Key and App Secret. These details will be put into a config file.
5. Create the script, calling it dropbox\_collector\_for\_sumologic.py (see Appendix A)
6. Create a config file, calling it dropbox\_collector\_for\_sumologic.ini (see Appendix B)
  - a. Put the App Key and App Secret inside the ini file in the [app\_key] section
7. `chmod +x dropbox_collector_for_sumologic.py`
8. Execute the logger to create an access token:  
`./dropbox_collector_for_sumologic.py login`
9. Test the logger to display up to 10 log lines. If this fails, retry the steps.  
`./dropbox_collector_for_sumologic.py test`
10. Add source to collector as a script
  - a. Frequency can be chosen as required (Every 5 minutes should be OK)
  - b. Command: `/usr/bin/python`
  - c. Enter full path to collector script, e.g.:  
`/usr/local/dropbox_collector_for_sumologic.py`
  - d. Working directory, e.g.: `/usr/local`

## Screenshots

Config file has just app\_key and app\_secret in it (we are at stage 8)

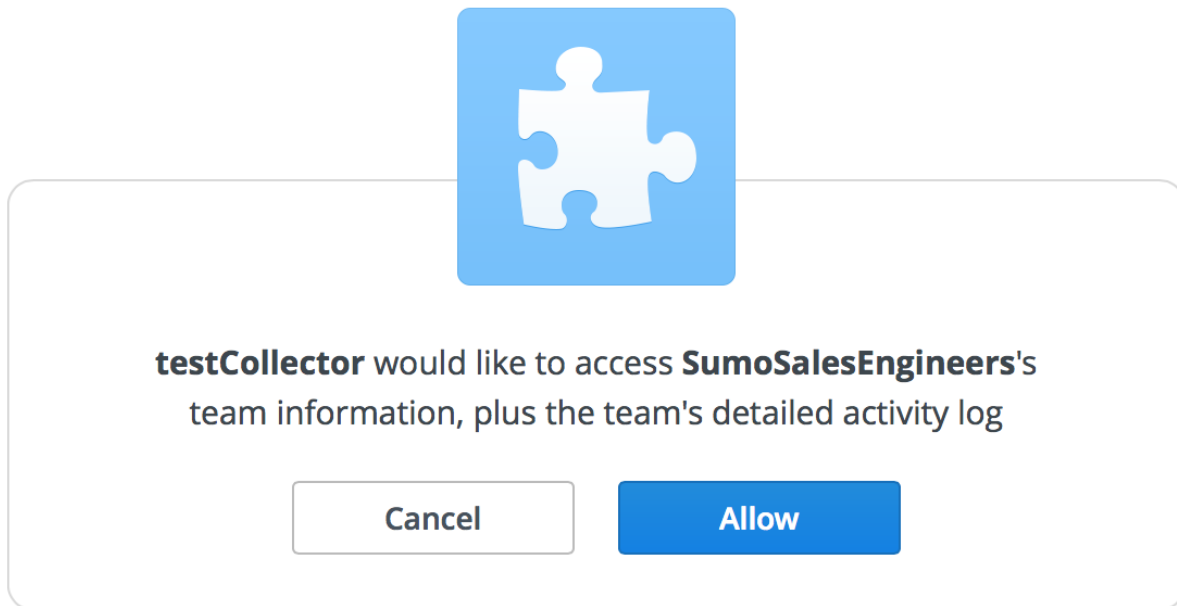
```
example — Python — 98x19
mjeffery-mac:example mjeffery$ ./dropbox_collector_for_sumologic.py login
1. Go to: https://www.dropbox.com/1/oauth2/authorize?response_type=code&client_id=cpfxpmpf5hyh9t9

2. Click "Allow" (you might have to log in first).

3. Copy the authorization code.

Enter the authorization code here: █
```

Follow the instructions: Allow the app



Copy the key given into the terminal window



Enter this code into **testCollector** to finish the process.

**qf-tegjVBgMAAAAAAAAAAKlOtVuHBno8NNNo2mXHIG4PE**

Paste the code into the terminal window and press return

```
example — Python — 98x19
mjeffery-mac:example mjeffery$ ./dropbox_collector_for_sumologic.py login
1. Go to: https://www.dropbox.com/1/oauth2/authorize?response_type=code&client_id=cpfxpmpf5hyh9t9

2. Click "Allow" (you might have to log in first).

3. Copy the authorization code.

Enter the authorization code here: qf-tegjVBgMAAAAAAAAAAKlOtVuHBno8NNNo2mXHIG4PE
```

The script adds to the config file, which now looks like this:

```
3. Copy the authorization code.

Enter the authorization code here: qf-tegjVBgMAAAAAAAAAAKlOtVuHBno8NNNo2mXHIG4PE
mjeffery-mac:example mjeffery$ cat dropbox_collector_for_sumologic.ini
[app_key]
app_key = cpfxpmpf5hyh9t9
app_secret = 0g7l5lpiuez2uz3

[token]
token = oauth2:qf-tegjVBgMAAAAAAAAAAKwAu8T8sLsArWBZyEdX5pisWgl08BbrAuq8n0LsSXyxI

[cursor]
cursor = AACKLdirfiJXUosZnGB0INPvTPa644-ufBUrsl0810rNwlnzekvEf2wk91f_X0VKI601scS97nT1UJziYyt59gE6
nKEwSIFf0dv_JQtKwkAeVVkmIqews--2aZHfk8z0m3xDbV4gM0AJ-G_CdHIbo4MfcUaTMupdEE70zQaEP1bL1A8ZKtBx00EB4i
ZmHmav62QpA9zesqqnG5DdKzb5s2imTfK8oFJ9IR80VJWp2RX0SkGVruue_QrpsB6WZ5jNQGhiV4IocStuJwj4mya69IANwp_X
-LRmQQcE6mQalSWtXTxByvnEmhqicSnWvh4C_uVQDHDdVQ0SxIhQAH9bBX9fzeT_qr-ypqtGYQwpYtb3rS5F3rN7PIeNFPKNFD
c_nCqZkXkDtDKmIFCj6XFYh0Vp-i_05KZc4JG2WwxY51Tpw

mjeffery-mac:example mjeffery$
```

Next, run a test by executing, which should show up to 10 events.

```
mjeffery-mac:example mjeffery$ ./dropbox_collector_for_sumologic.py test
{
  "user_id": 370099599,
  "name": "Mark Jeffery",
  "country": "GB",
  "time": "2014-12-22T16:46:23+00:00",
  "event_type_description": "Joined the team",
  "event_category": "members",
  "info_dict": {
    "initial_devices": "[]",
    "initial_apps": "[]"
  },
  "member_id": "dbmid:AACsY3xyeIR9C_fhF_qQMJubaH0ofU5y7Zw",
  "ip_address": "86.24.189.158",
  "email": "mjeffery@sumologic.com",
  "event_type": "member_join"
}
{
  "user_id": 370099599,
```

## Dashboard Design

Dashboards can be organised around the event categories and event types in Dropbox for Business. A list of event categories and types can be found:

<https://www.dropbox.com/developers/business/docs#log-get-events>

## Appendix A - dropbox\_collector\_for\_sumologic.py

```
#!/usr/bin/env python

import cmd
import locale
import os
import pprint
import shlex
import sys
import pdb
import json
import ConfigParser
from dropbox.rest import RESTClientObject, RESTResponse, SDK_VERSION, ErrorResponse

PY3 = sys.version_info[0] == 3

if PY3:
    from io import StringIO
else:
    from StringIO import StringIO

from dropbox import client, rest, session

class DropBoxAccess():
    def __init__(self):
        self.api_client = None
        self.config = ConfigParser.RawConfigParser()
        self.config.read('dropbox_collector_for_sumologic.ini')

        try:
            self.app_key = self.config.get('app_key', 'APP_KEY')
            self.app_secret = self.config.get('app_key', 'APP_SECRET')
        except (ConfigParser.NoOptionError, ConfigParser.NoSectionError):
            print("Config file does not have entries for App Key and App Secret")
            print("Check the installation instructions to generate them")
            return

        try:
            serialized_token = self.config.get('token', 'TOKEN')
            if serialized_token.startswith('oauth1:'):
                access_key, access_secret = serialized_token[len('oauth1:'):].split(':', 1)
                sess = session.DropboxSession(self.app_key, self.app_secret)
                sess.set_token(access_key, access_secret)
                self.api_client = client.DropboxClient(sess)
            elif serialized_token.startswith('oauth2:'):
                access_token = serialized_token[len('oauth2:'):]
                self.api_client = client.DropboxClient(access_token)
            else:
                print "Malformed access token in %r." % (self.TOKEN_FILE,)
        except ConfigParser.NoSectionError:
            pass

    def do_log(self, action):
        """show auditlog"""
```

```

if self.api_client is None:
    print("Please 'login' to execute this command\n")
    return
if action == "test":
    body={'limit':10}
else:
    body={'limit':1000}
if action == "showlog":
    body['start_ts'] = 1418651258000 # Mon, 15 Dec 2014 13:47:38 GMT
    if self.config.has_option("cursor","cursor"):
        body['cursor'] = self.config.get("cursor","cursor")
url, params, headers = self.api_client.request("/team/log/get_events", method="POST")
headers['Content-Type'] = "application/json"
bodytext = json.dumps(body)
post_params = params
try:
    rc = RESTClientObject()
    ret = rc.request(method = "POST", url = url, body=bodytext, headers = headers)
except rest.ErrorResponse, e:
    print('Error: %s\n' % str(e))
    return

if action == "showlog" and "cursor" in ret:
    if not self.config.has_section("cursor"):
        self.config.add_section("cursor")
    self.config.set("cursor","cursor",ret["cursor"])
for k in ret["events"]:
    timestring = k["time"].replace("T"," ",1)
    timestring = timestring.replace("+00:00"," -0000",1)
    print timestring + ", " + json.dumps(k)
with open('dropbox_collector_for_sumologic.ini', 'wb') as configfile:
    self.config.write(configfile)

def do_login(self):
    """log in to a Dropbox account"""
    flow = client.DropboxOAuth2FlowNoRedirect(self.app_key, self.app_secret)
    authorize_url = flow.start()
    print("1. Go to: " + authorize_url + "\n")
    print("2. Click \"Allow\" (you might have to log in first).\n")
    print("3. Copy the authorization code.\n")
    code = raw_input("Enter the authorization code here: ").strip()

    try:
        access_token, user_id = flow.finish(code)
    except rest.ErrorResponse, e:
        print('Error: %s\n' % str(e))
        return

    if not self.config.has_section("token"):
        self.config.add_section("token")
    self.config.set("token","TOKEN",'oauth2:' + access_token)
    with open('dropbox_collector_for_sumologic.ini', 'wb') as configfile:
        self.config.write(configfile)
    self.api_client = client.DropboxClient(access_token)

def main():
    # TODO: Write up procedure for installing

```

```

dba = DropBoxAccess()

prog_name = sys.argv[0]
args = sys.argv[1:]

if len(args) == 0:
    dba.do_log("showlog")
    sys.exit(0)

command = args[0]
if command == 'login':
    # Implement do_login based on existing, but add APP_KEY & APP_SECRET
    dba.do_login()
elif command == 'test':
    # Implement test, requesting 10 lines, but not storing position
    dba.do_log("test")
elif command == 'showlog':
    # Show 1000 lines of log, storing position
    dba.do_log("showlog")
else:
    sys.stderr.write("ERROR: Unknown command: %r\n" % command)
    sys.stderr.write("Run with no arguments for help.\n")
    sys.exit(1)

if __name__ == '__main__':
    main()

```



## Appendix B - dropbox\_logger\_for\_sumologic.ini

```
[app_key]  
APP_KEY:  
APP_SECRET:
```