









Code:

import sys

import dropbox

from dropbox.files import WriteMode

from dropbox.exceptions import ApiError, AuthError

import datetime

dt = datetime.datetime.today()

TOKEN = 'sl.BH5LkK03aRLRLVw1PFEZjRNSbgKEjmH2D\_iJlAiDHDYFWQT0wYgFy1zjGBianKplXs1hWbBszVgkUoMgv4rLQ8qrfc4Ifg-TsSHvDjRLaIi3179z76vCBW1q8MA\_3ydp9f2UWnM'

LOCALFILE = 'C:\\Users\\rajna\\Downloads\\GOOGLECLOUDPROGRAM\\kitten.png'

# Don't forget to add the file extension at the end of BACKUPPATH.

BACKUPPATH ='/kitten.png'

*def* backup(*localFile*, *backupPath*):

    with open(*localFile*, 'rb') as f:

        # We use WriteMode=overwrite to make sure that the settings in the file

        # are changed on upload

        print("Uploading " + *localFile* + " to Dropbox as " + *backupPath* + "...")

        try:

            dbx.files\_upload(f.read(), *backupPath*, *mode*=WriteMode('overwrite'))

        except ApiError as err:

            # This checks for the specific error where a user doesn't have

            # enough Dropbox space quota to upload this file

            if (err.error.is\_path() and

                    err.error.get\_path().reason.is\_insufficient\_space()):

                sys.exit("ERROR: Cannot back up; insufficient space.")

            elif err.user\_message\_text:

                print(err.user\_message\_text)

                sys.exit()

            else:

                print(err)

                sys.exit()

if \_\_name\_\_ == '\_\_main\_\_':

    # Create an instance of a Dropbox class, which can make requests to the API.

    print("Creating a Dropbox object...")

    dbx = dropbox.Dropbox(TOKEN)

    # Check that the access token is valid

    try:

        dbx.users\_get\_current\_account()

    except AuthError:

        sys.exit("ERROR: Invalid access token; try re-generating an "

                 "access token from the app console on the web.")

    # Create a backup of the current settings file

    backup(LOCALFILE, BACKUPPATH)

    print("Done!")



