import sys

import time

import os

BEE\_PATH = sys.argv[1]

TEMP\_BEE\_PATH = BEE\_PATH + "/temp"

APIUsername = "alexpimania"

APIPassword = "wsxcdeWSXCDE99!"

repositoryIsPrivate = True

releasesUrl = "https://api.github.com/repos/alexpimania/beeDev/releases"

def getLatestReleaseURL():

import requests

import json

from requests.auth import HTTPBasicAuth

if repositoryIsPrivate:

response = requests.get(releasesUrl, auth=HTTPBasicAuth(APIUsername, APIPassword))

else:

response = requests.get(releasesUrl)

releasesJson = response.content.decode("utf-8")

releases = json.loads(releasesJson)

if response.status\_code == 200 and len(releases) >= 1:

downloadUrl = releases[0]["zipball\_url"]

else:

downloadUrl = "error"

return downloadUrl

def makeDirectories(directoryList):

import os

for directory in directoryList:

if not os.path.exists(directory):

os.makedirs(directory)

def deleteDirectories(directoryList):

import os

import shutil

for directory in directoryList:

if os.path.exists(directory):

shutil.rmtree(directory)

def getImprovementsList(latestReleaseURL):

#This function gets all release notes for all releases and the latest release name and latest release source code location

import requests

from requests.auth import HTTPBasicAuth

import os

import shutil

import zipfile

extractionPath = TEMP\_BEE\_PATH + "/extractedRelease"

zippedReleasePath = TEMP\_BEE\_PATH + "/zippedRelease"

if latestReleaseURL == "error":

return "<improvements></improvements>"

deleteDirectories([TEMP\_BEE\_PATH])

makeDirectories([extractionPath, zippedReleasePath])

if repositoryIsPrivate:

response = requests.get(latestReleaseURL, stream=True, auth=HTTPBasicAuth(APIUsername, APIPassword))

else:

response = requests.get(latestReleaseURL, stream=True)

releaseZipFile = open(zippedReleasePath + "/latestRelease.zip", "wb+")

releaseZipFile.write(response.content)

releaseZipFile.close()

releaseZipFileObject = zipfile.ZipFile(zippedReleasePath + "/latestRelease.zip", 'r')

releaseZipFileObject.extractall(extractionPath)

releaseZipFileObject.close()

extractedFilesPath = extractionPath + "/" + os.listdir(extractionPath)[0]

improvementsFilePath = extractedFilesPath + "/backend/improvementsList.ini"

if os.path.exists(improvementsFilePath):

rawImprovementsList = open(improvementsFilePath).read().replace("\n", "")

else:

return "<improvements></improvements>"

return rawImprovementsList

def updateImprovementsFile(rawImprovementsList):

improvementsFile = open(BEE\_PATH + "/backend/improvementsList.ini", "w+")

improvementsFile.write(rawImprovementsList)

improvementsFile.close()

def checkForImprovements():

latestReleaseURL = getLatestReleaseURL()

rawImprovementsList = getImprovementsList(latestReleaseURL)

updateImprovementsFile(rawImprovementsList)

while not os.path.exists(BEE\_PATH + "/beeUpdatesStop"):

checkForImprovements() # check to log stats for each profile

for i in range(1, 40): # 6 times ten second sleeps = one minute

time.sleep(5) # sleep for ten seconds

if os.path.exists(BEE\_PATH + "/beeUpdatesStop"):

break

os.remove(BEE\_PATH + "/beeUpdatesStop")