3/8/2022: 下面是CP传过来的问题详细：

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated

下面是CP上传用的Python script: package name 因涉及隐私，用的不是CP的

import argparse

import requests

import os

import sys

#base URL for all API calls

baseApi = '<https://connect-api.cloud.huawei.com/api>'

def logError(resp):

print("Error: " + resp.status\_code + " " + resp.reason)

def logSuccess(s):

print("Success: " + s)

def getAccessToken():

apiCredentials = {

'grant\_type': 'client\_credentials',

'client\_id': clientId,

'client\_secret': clientSecret

}

resp = requests.post(baseApi + "/oauth2/v1/token", json=apiCredentials)

if resp.status\_code == 200:

logSuccess("Got access token")

creds = resp.json()

if 'access\_token' not in creds:

print("Error: access\_token not found, clientId: " + clientId)

print("Error: access\_token not found, resp: " + str(creds))

token = creds['access\_token']

getUploadUrl(token)

else:

logError(resp)

def getUploadUrl(token):

params = {

'appId': appId,

'suffix': "apk"

}

header = {

'client\_id': clientId,

'Authorization': 'Bearer ' + token

}

resp = requests.get(baseApi + "/publish/v2/upload-url", params=params, headers=header)

if resp.status\_code == 200:

logSuccess("Got upload url")

creds = resp.json()

uploadUrl = creds['uploadUrl']

authCode = creds['authCode']

uploadApp(authCode, uploadUrl, token)

else:

logError(resp)

def uploadApp(authCode, uploadUrl, accessToken):

header = {

'client\_id': clientId,

'Authorization': 'Bearer ' + accessToken,

}

req\_body = {

'authCode': authCode,

'fileCount': 1

}

params = {

'appId': appId,

}

with open(aabPath, 'rb') as f:

prep = requests.post(uploadUrl,

files={aabName: f},

data=req\_body,

headers=header)

logSuccess("Upload prep")

if prep.status\_code == 200:

j = prep.json()

print(j)

body = {

'fileType': 5,

'files': [{

'fileName': aabName,

'fileDestUrl': j['result']['UploadFileRsp']['fileInfoList'][0]['fileDestUlr'],

'size': str(j['result']['UploadFileRsp']['fileInfoList'][0]['size'])

}]

}

upload = requests.put(baseApi + "/publish/v2/app-file-info",

headers=header,

json=body,

params=params)

if upload.status\_code == 200:

logSuccess("Upload complete!")

return upload

else:

logError(upload)

else:

logError(prep)

def launchUpload():

getAccessToken()

if \_\_name\_\_ == "\_\_main\_\_":

parser = argparse.ArgumentParser()

parser.add\_argument("package\_name", help="Package name of the app (e.g. `com.aaa.aaa.whatever.whatever.huawei`)")

parser.add\_argument("sem\_ver", help="Version (e.g. `0.6.5`)")

parser.add\_argument("aab\_path", help="Path to the aab file to upload")

args = parser.parse\_args()

if args.package\_name == "com.aaa.aaa.android.huawei":

appId = "105091411"

elif args.package\_name == "com.aaa.aaa.android.test.aaa.huawei":

appId = "105124145"

elif args.package\_name == "com.aaa.aaa.android.dev.aaa.huawei":

appId = "105123359"

else:

assert False, f"Package name `{args.package\_name}` is currently not supported for uploading"

assert "HUAWEI\_CLIENT\_ID" in os.environ

clientId = os.environ["HUAWEI\_CLIENT\_ID"]

assert "HUAWEI\_CLIENT\_SECRET" in os.environ

clientSecret = os.environ["HUAWEI\_CLIENT\_SECRET"]

#absolute path and filename of the .aab

aabPath = args.aab\_path

aabName = os.path.basename(aabPath)

print("Uploading " + aabPath)

print("Basename = " + aabName)

launchUpload()