# Call script SendToSlack.sh (on Linux) with 3 parameters

# Jython Automationscript for TPAe 7.1.1.7 and above to execute a command on the Application Server

import sys

from java.io import \*

from java.lang import Runtime

#---------------------------------------------------------------------------------------------------------------------------------------

#

# script CASE\_SEND\_2\_SLACK.py

#

#

# Inputs:

#

# in\_CASE\_SLACK\_CHANNEL

# - Slack Channel for this incident

#

# in\_TICKETID

# - Ticket ID of the incident

#

# in\_STATUS

# - Status the incident record

#

# in\_STATUSDATE

# - Closing date of the incident record

#

# Outputs:

#

# N/A

#

# Description

#

# When an incident is CLOSED, call a Linux script which add comments to the Slack Channel of the incident

#

#---------------------------------------------------------------------------------------------------------------------------------------

from java.lang import Exception as javaException

from java.lang import System as javaSystem

from java.lang import String

from psdi.util import MXApplicationException;

from psdi.util import MXException;

from psdi.util.logging import MXLogger

from psdi.util.logging import MXLoggerFactory

from psdi.app.ci import CIRemote;

from psdi.mbo import Mbo, MboRemote,MboSet, MboSetRemote, MboConstants, SqlFormat;

from psdi.server import MXServer

from java.io import IOException as javaIOException

from com.ibm.json.java import JSONArray, JSONObject;

from java.util import Hashtable

from java.util import Date

from java.text import SimpleDateFormat

# only execute the script when the incident record is closed

if in\_STATUS=='CLOSED':

## for convenience, set a log-prefix, such that all log messages from this script start with the same prefix

LOGPREFIX = "AutoScript EXEC\_CMD | "

# change the command object to the command you would like to execute

#command = "/tmp/leescript.sh"

command = "/root/Documents/SendToSlack.sh"

slack = ""

slack = String(in\_CASE\_SLACK\_CHANNEL)

pos = (str(slack)).find("<!-- RICH TEXT -->")

if int(pos) > 0:

slack = (str(slack))[0:pos]

#slack = (str(slack))[0:3]

# this script was originally designed to run on Linux. If there is not /bin/sh on your system available, simply change to the desired shell/program

# or just leave the command as the only array member. The reason for this array is to be able to pipe commands

commands = [command, str(slack), str(in\_TICKETID), str(in\_STATUSDATE)]

print "#############################"

print commands

print "#############################"

r = Runtime.getRuntime()

p = r.exec(commands)

stdin = BufferedReader(InputStreamReader(p.getInputStream()))

stderr = BufferedReader(InputStreamReader(p.getErrorStream()))

# print all output to stdout

z = 1

while z>0:

s = stdin.readLine()

if (s is not None):

print >> sys.stdout, LOGPREFIX, s

else:

z = 0

print >> sys.stdout, LOGPREFIX, "--- done iterating stdout ---"

# end of while - while z>0:

# print all output to stderr

y = 1

while y>0:

s = stderr.readLine()

if (s is not None):

print >> sys.stdout, LOGPREFIX, s

else:

y = 0

print >> sys.stdout, LOGPREFIX, "--- done iterating stderr ---"

# end of while - while y>0:

# end of If - if in\_STATUS=='CLOSED':