**שימו לב לשנות את הביטויים ואת הקישור לWebhook! (מסומן בירוק)**

import json

import urllib3

def lambda\_handler(event, context):

http = urllib3.PoolManager()

phrases = [

"ראיתי את זה בפייסבוק",

"הגעתי מאתר האינטרנט",

"הגעתי דרך האתר",

"אשמח לקבל פרטים נוספים",

"אפשר לקבל מידע נוסף",

"פרטים על הסטודיו",

"אשמח לשמוע עוד פרטים",

"Can I get more info",

"אפשר פרטים בבקשה",

"אני רוצה עוד מידע. יש מישהו שפנוי לצ'אט?",

"like to know more. Is anyone free to",

"היי, אשמח לשמוע עוד פרטים בבקשה",

"היי, אשמח לשמוע"

]

try:

if "body" not in event:

print("Missing 'body' in event")

return {

'statusCode': 400,

'body': json.dumps('Request does not contain body')

}

body = json.loads(event["body"])

print("Received message:", body)

if body.get("typeWebhook") == "quotaExceeded":

print("Quota exceeded notification received. Ignoring.")

return {

'statusCode': 200,

'body': json.dumps('Quota exceeded notification ignored')

}

messages = body if isinstance(body, list) else [body]

for message in messages:

message\_text = None

if (

"messageData" in message and

"textMessageData" in message["messageData"] and

"textMessage" in message["messageData"]["textMessageData"]

):

message\_text = message["messageData"]["textMessageData"]["textMessage"]

elif (

"messageData" in message and

"extendedTextMessageData" in message["messageData"] and

"text" in message["messageData"]["extendedTextMessageData"]

):

message\_text = message["messageData"]["extendedTextMessageData"]["text"]

if message\_text:

print("Message content:", message\_text)

if any(phrase in message\_text for phrase in phrases):

print("The message contains a specified phrase. Sending data to external webhook...")

response = http.request(

"POST",

"https://hook.eu2.make.com/w2c9m3v4ib99iht11vpy6ttvcubs3iw6",

headers={'Content-Type': 'application/json'},

body=json.dumps(message)

)

print("Webhook response status:", response.status)

print("Webhook response data:", response.data.decode('utf-8'))

else:

print("The message does not contain any specified phrase. Ignoring.")

return {

'statusCode': 200,

'body': json.dumps('Message processed')

}

except json.JSONDecodeError:

print("Failed to decode JSON")

return {

'statusCode': 400,

'body': json.dumps('Invalid JSON format')

}

except Exception as e:

print("Error processing request:", e)

return {

'statusCode': 500,

'body': json.dumps(f'Internal Server Error: {str(e)}')

}