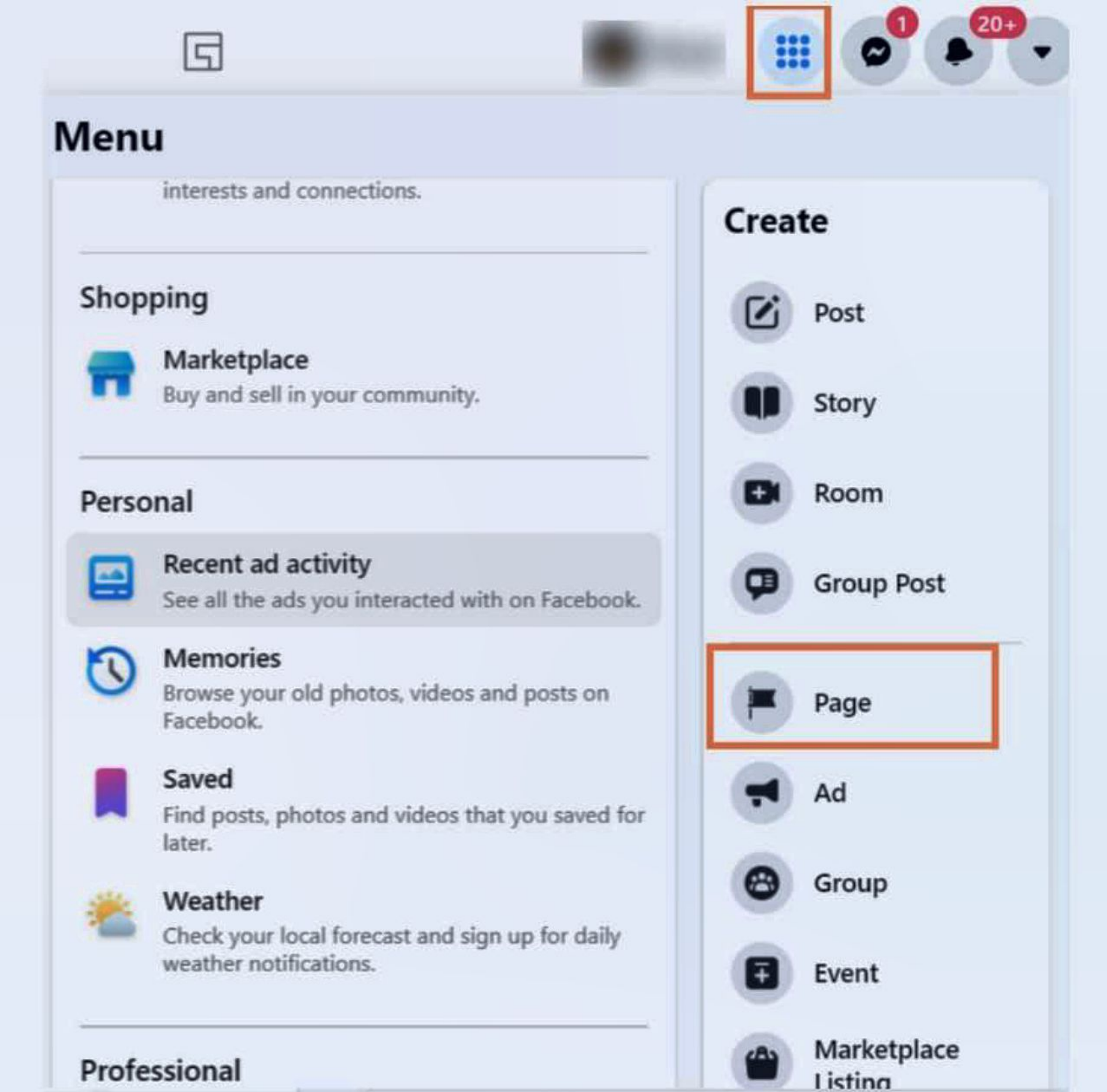


interests and connections.

Shopping

Marketplace
Buy and sell in your community.

Personal

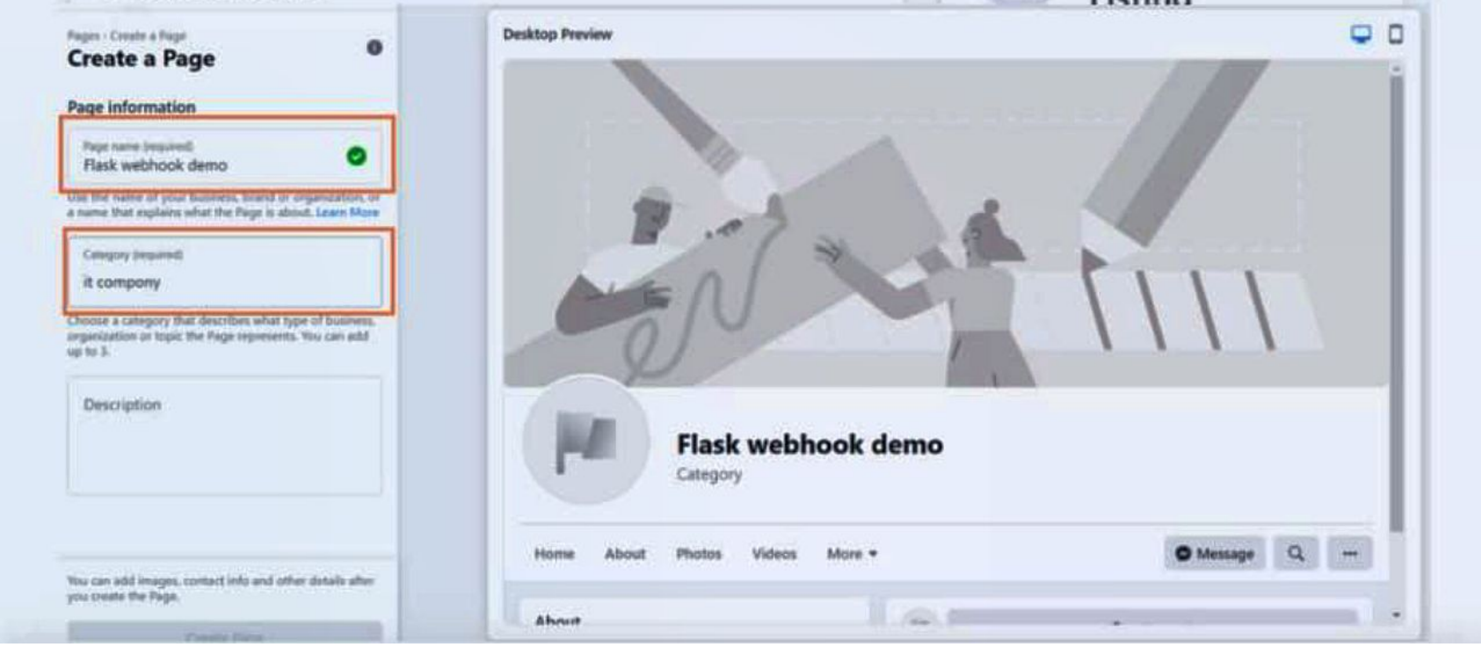
Recent ad activity
See all the ads you interacted with on Facebook.

Memories
Browse your old photos, videos and posts on Facebook.

Saved
Find posts, photos and videos that you saved for later.

Weather
Check your local forecast and sign up for daily weather notifications.

Professional



Pages › Create a Page

Create a Page

Page information

Page name (required)
Flask webhook demo

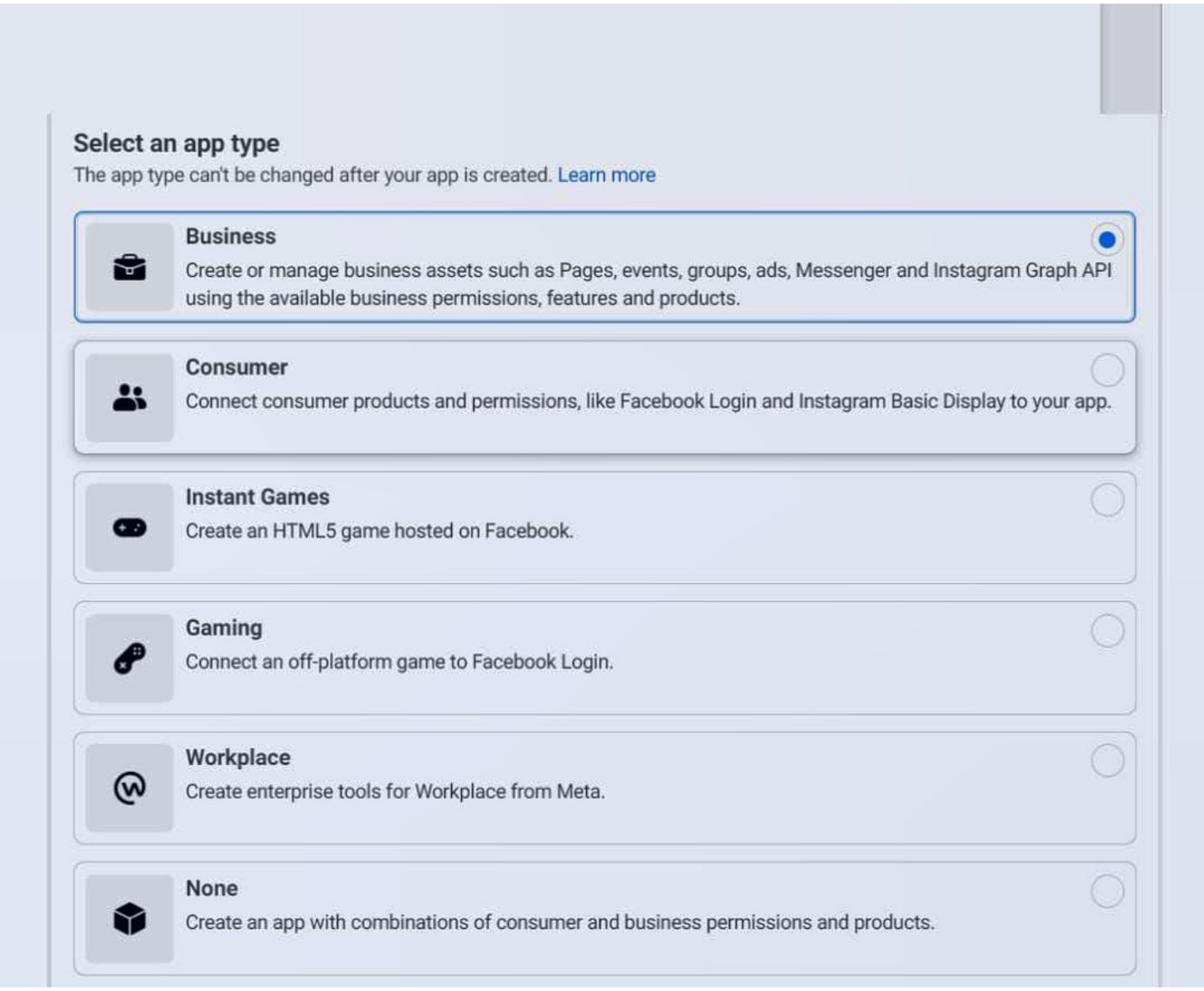
Use the name of your business, brand or organization, or a name that explains what the Page is about. Learn More

Category (required)
it compony

Choose a category that describes what type of business, organization or topic the Page represents. You can add up to 3.

Description

You can add images, contact info and other details after you create the Page.



Select an app type

The app type can't be changed after your app is created. Learn more

**Business** ●
Create or manage business assets such as Pages, events, groups, ads, Messenger and Instagram Graph API using the available business permissions, features and products.

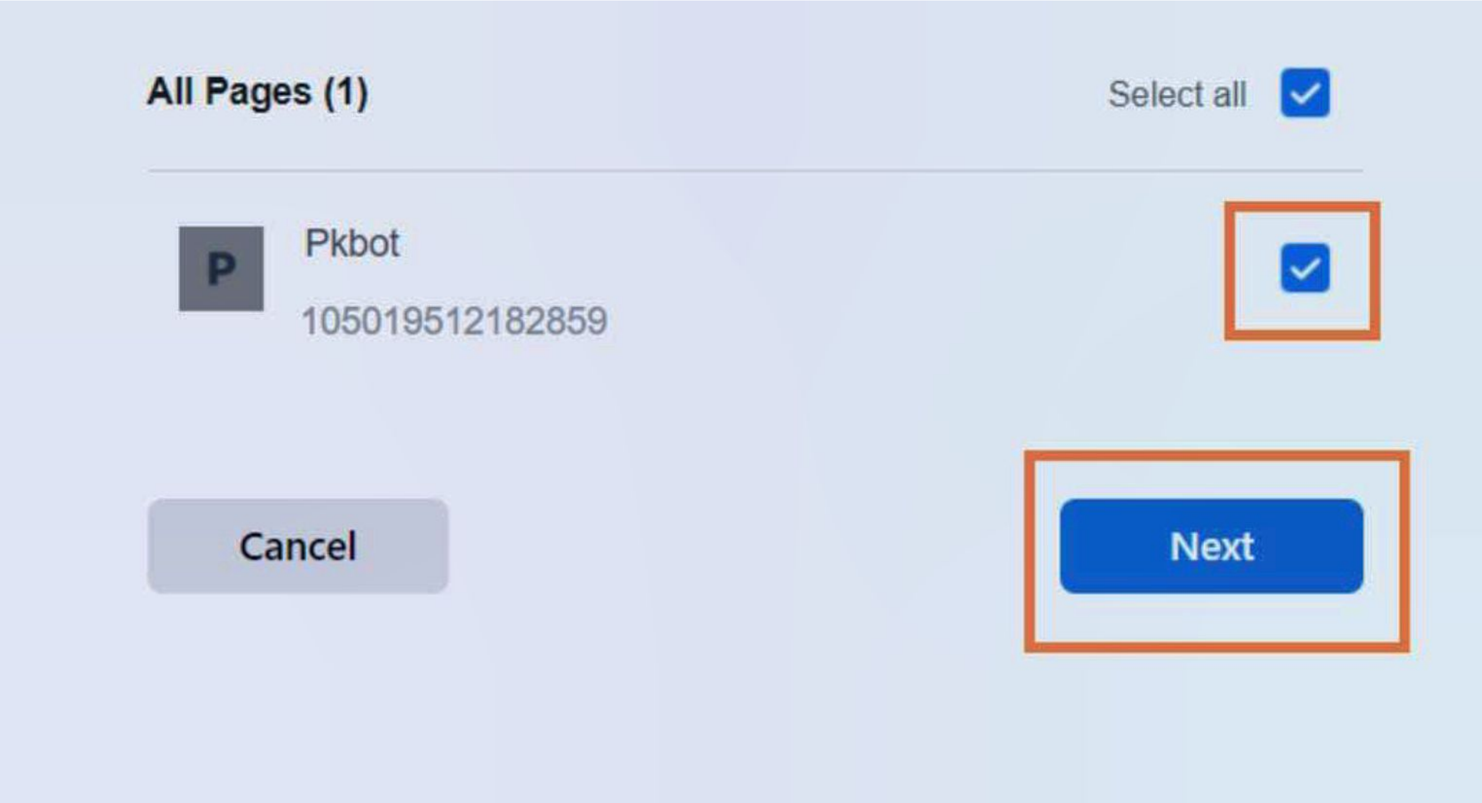
**Consumer** ○
Connect consumer products and permissions, like Facebook Login and Instagram Basic Display to your app.

**Instant Games** ○
Create an HTML5 game hosted on Facebook.

**Gaming** ○
Connect an off-platform game to Facebook Login.

**Workplace** ○
Create enterprise tools for Workplace from Meta.

**None** ○
Create an app with combinations of consumer and business permissions and products.

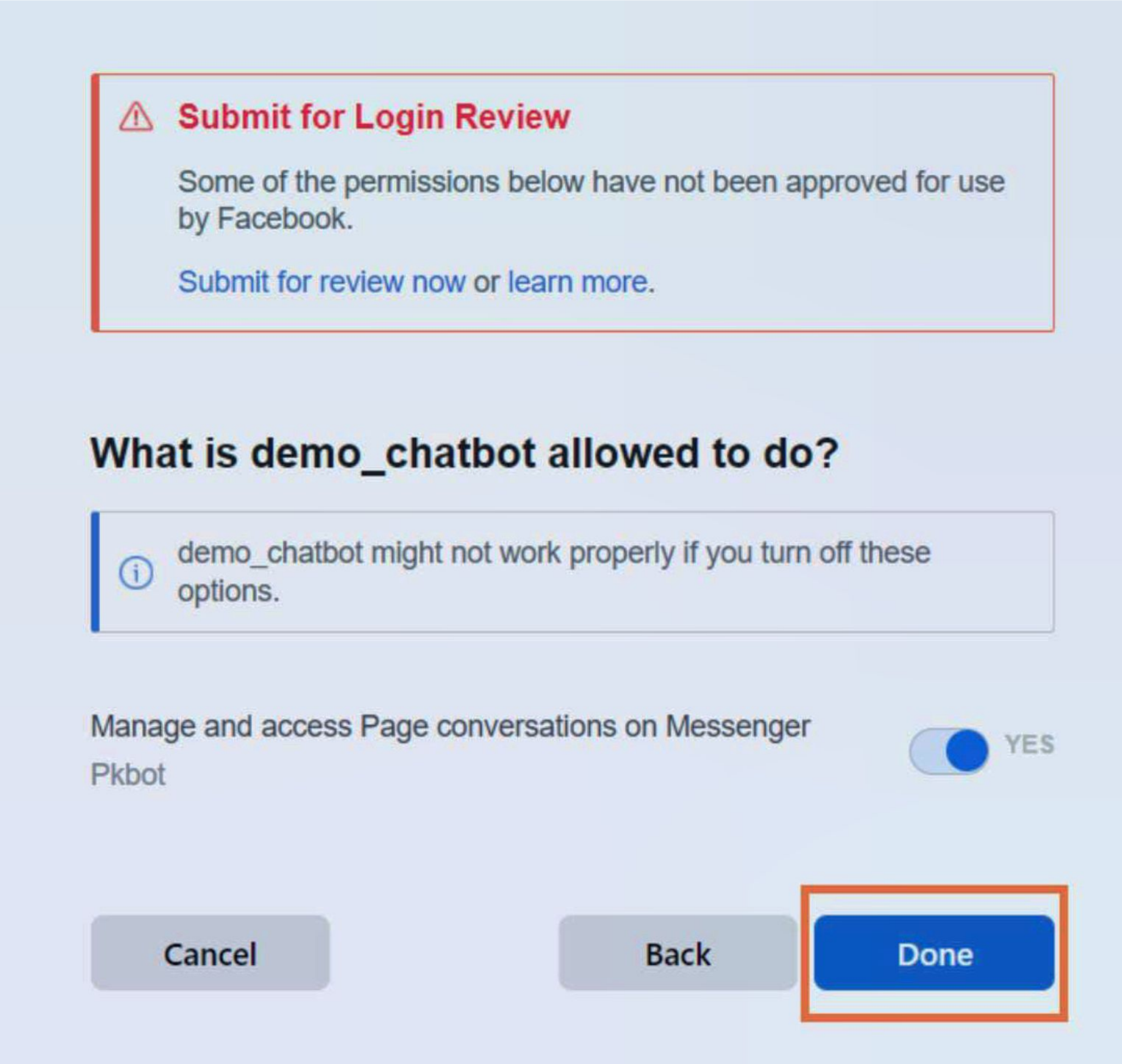


All Pages (1)

Select all ☑

P Pkbot

105019512182859 ☑



⚠ Submit for Login Review

Some of the permissions below have not been approved for use by Facebook.

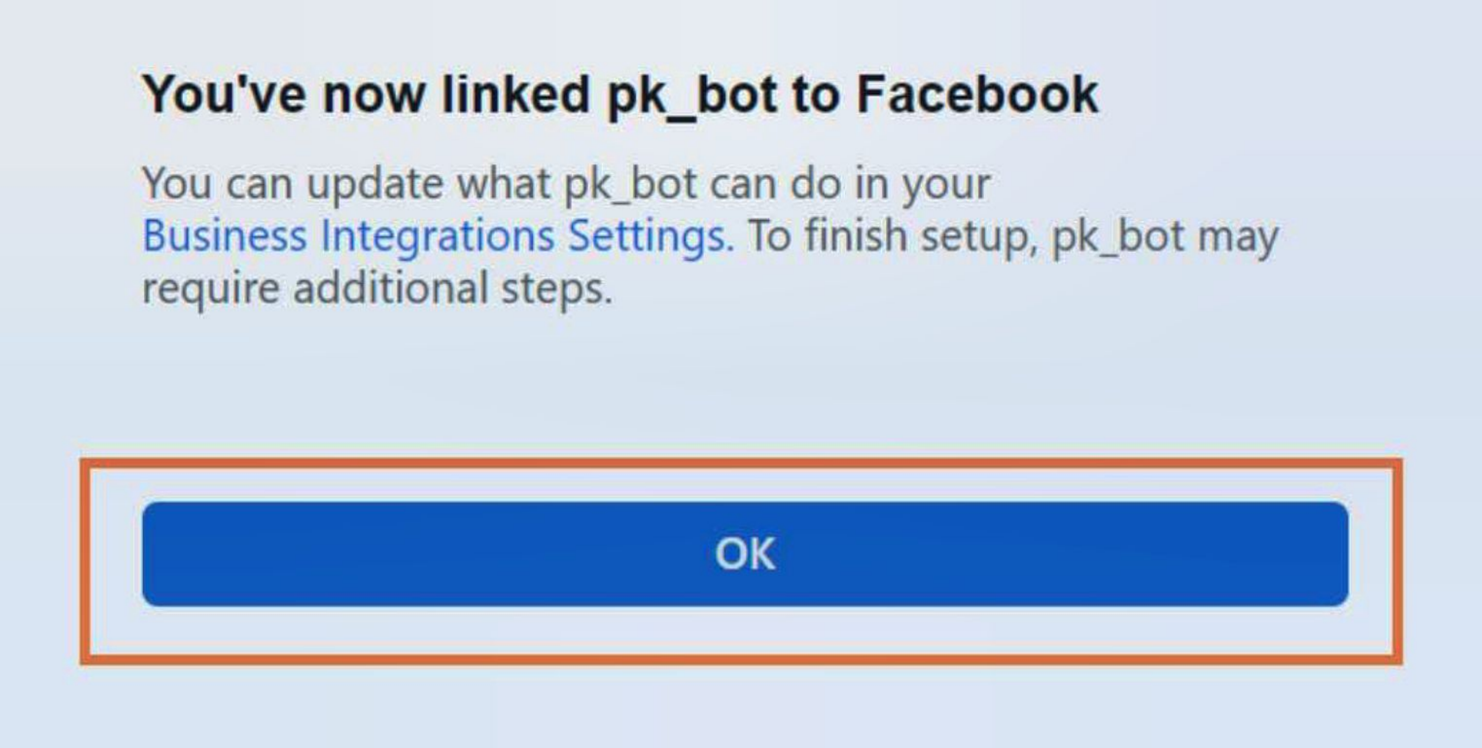
Submit for review now or learn more.

What is demo_chatbot allowed to do?

ⓘ demo_chatbot might not work properly if you turn off these options.

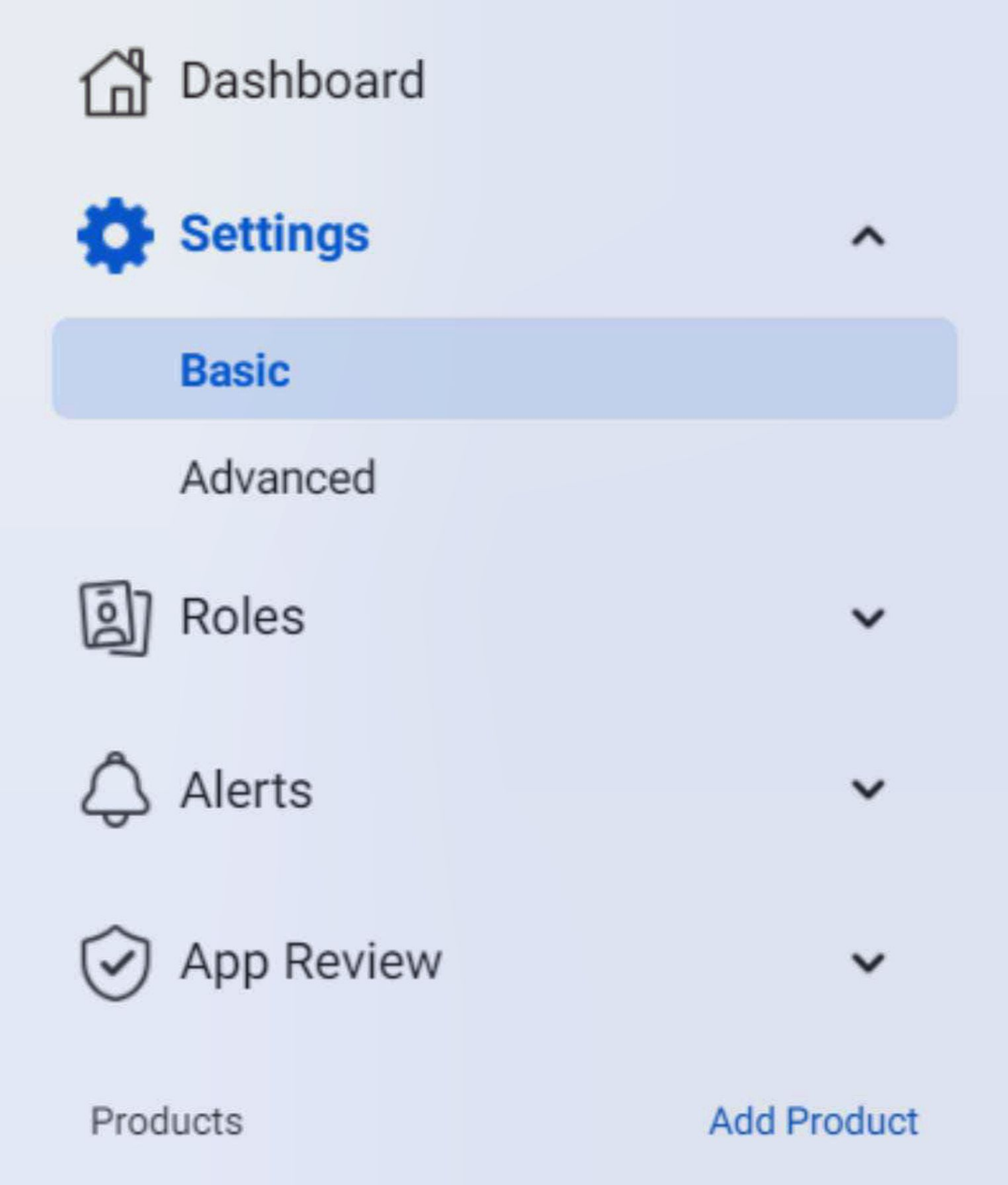
Manage and access Page conversations on Messenger ● YES

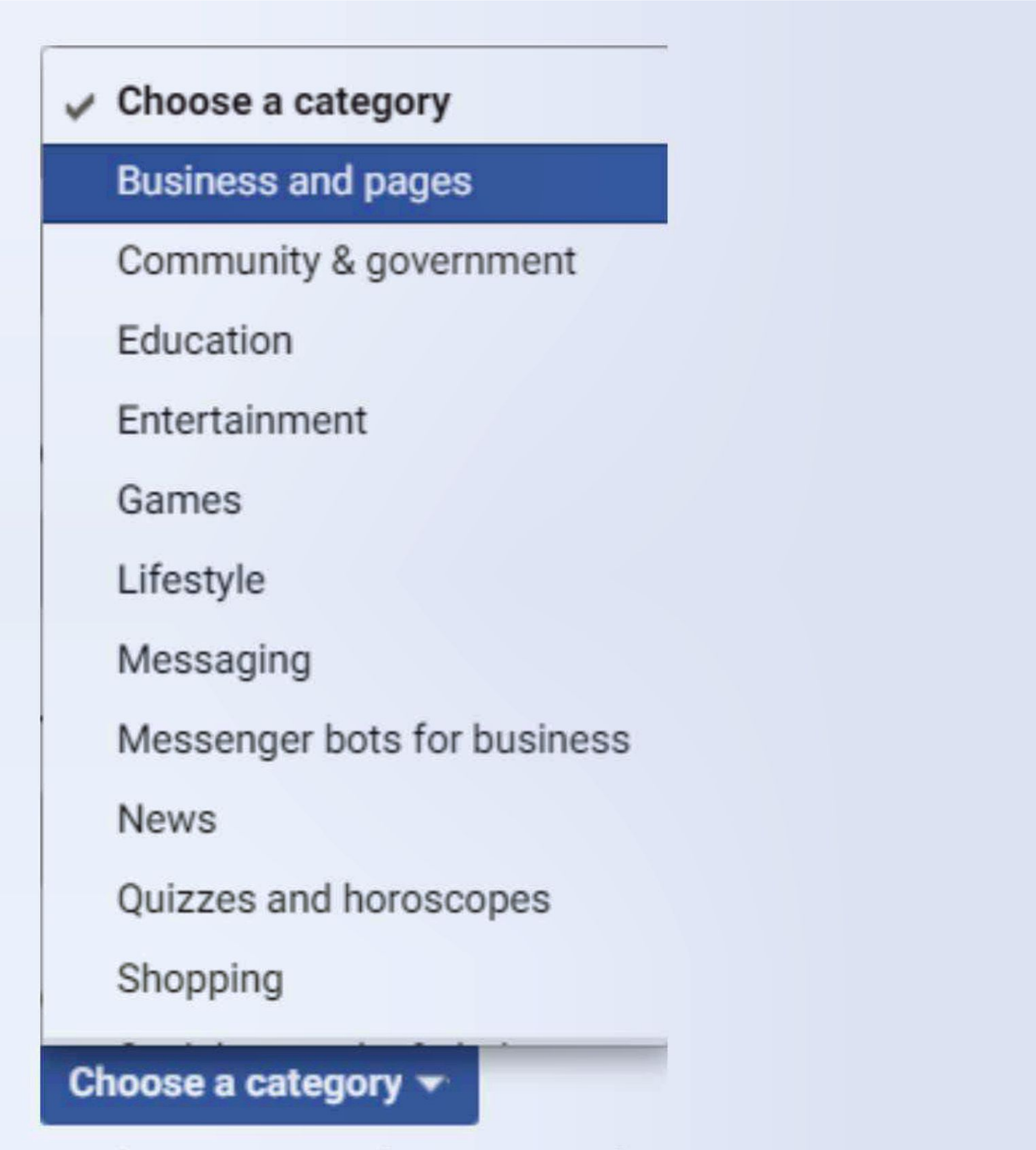
Pkbot

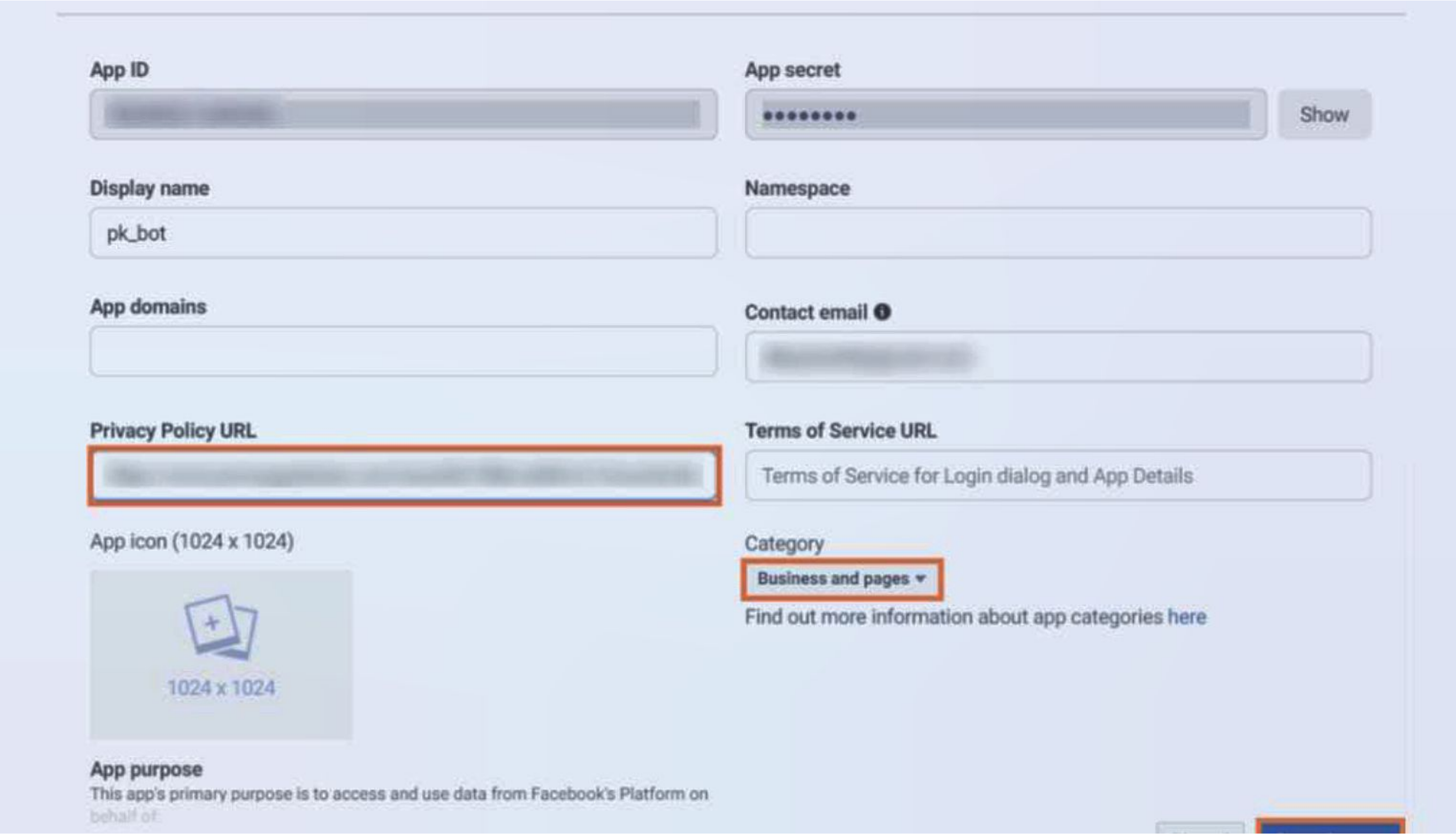


You've now linked pk_bot to Facebook

You can update what pk_bot can do in your Business Integrations Settings. To finish setup, pk_bot may require additional steps.


Dashboard
Settings
Basic
Advanced
Roles
Alerts
App Review
Products
Add Product


✓ Choose a category
Business and pages
Community & government
Education
Entertainment
Games
Lifestyle
Messaging
Messenger bots for business
News
Quizzes and horoscopes
Shopping
Choose a category ▾



App ID

[redacted]

App secret

•••••••••

Display name

pk_bot

Namespace

App domains

Contact email

[redacted]

Privacy Policy URL

[redacted]

Terms of Service URL

Terms of Service for Login dialog and App Details

App icon (1024 x 1024)

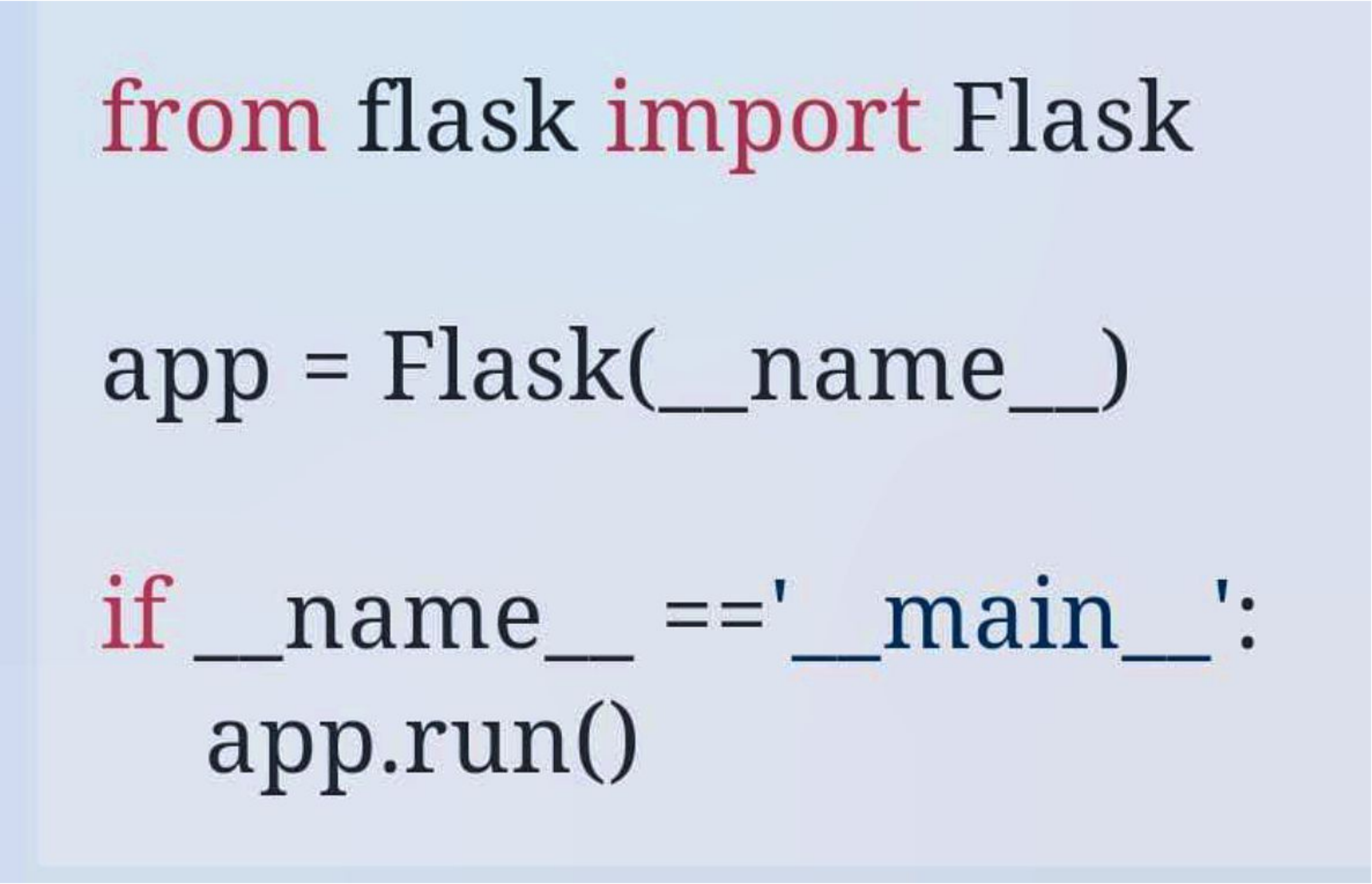
Category

Business and pages ▾

Find out more information about app categories here

App purpose

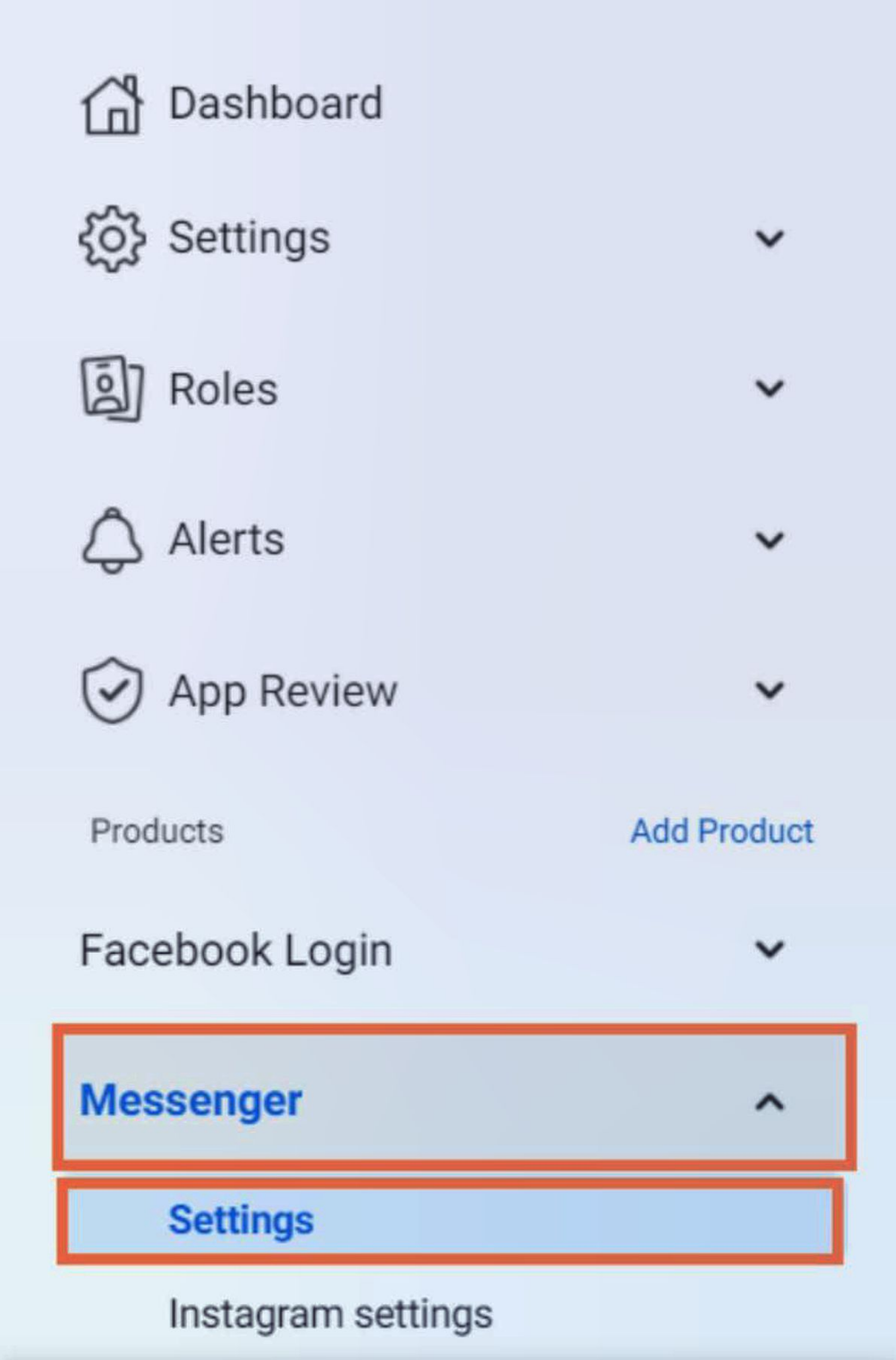
This app's primary purpose is to access and use data from Facebook's Platform on behalf of:

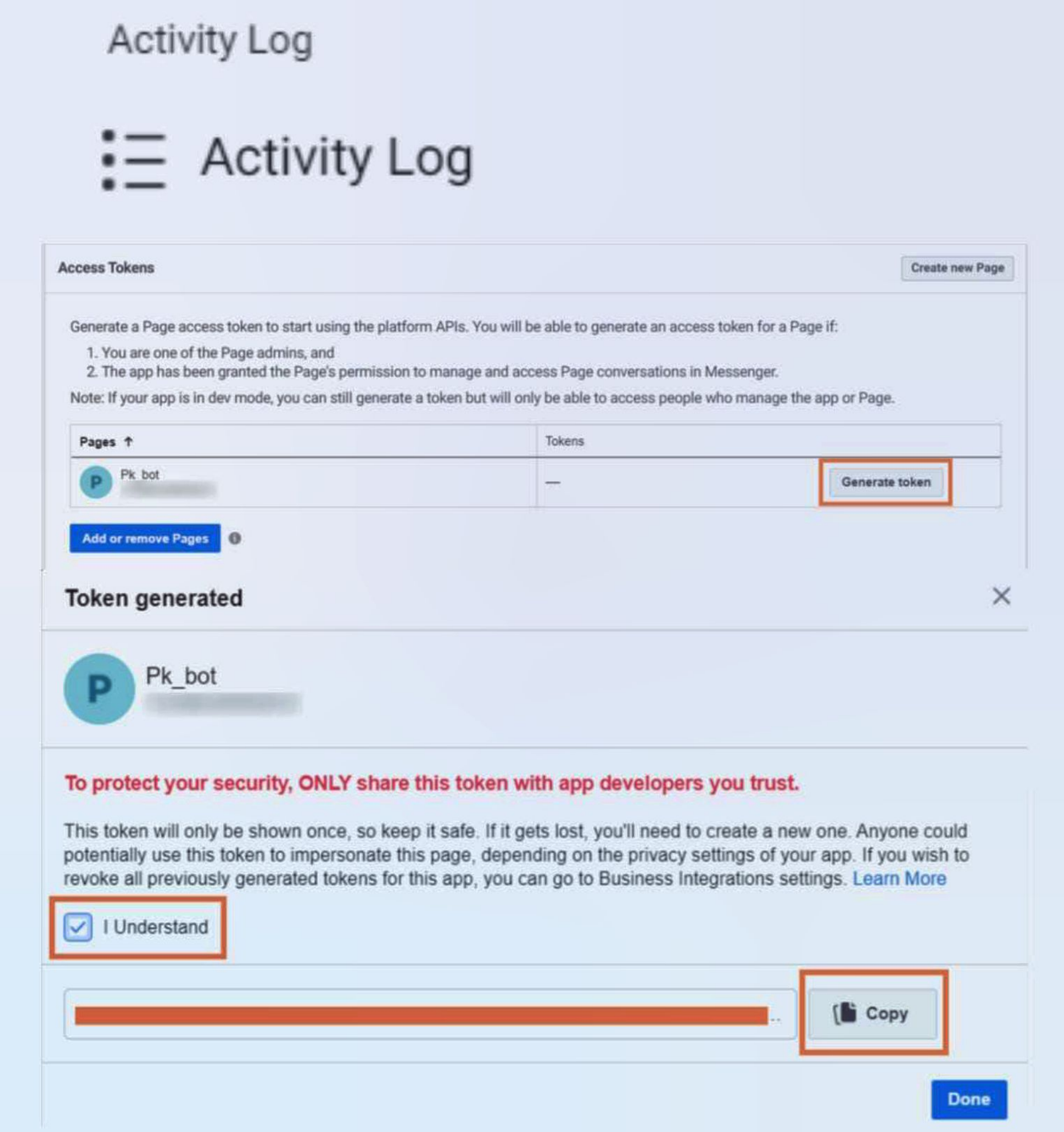


```
from flask import Flask

app = Flask(__name__)

if __name__ =='__main__':
    app.run()
```


Dashboard
Settings
Roles
Alerts
App Review
Products
Add Product
Facebook Login
Messenger
Settings
Instagram settings



Activity Log

Activity Log

Access Tokens

Create new Page

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

| Pages ↑ | Tokens | |
|---|---|---|
| P Pk_bot<br>[redacted] | — | Generate token |

Add or remove Pages

Token generated ✕

P Pk_bot
[redacted]

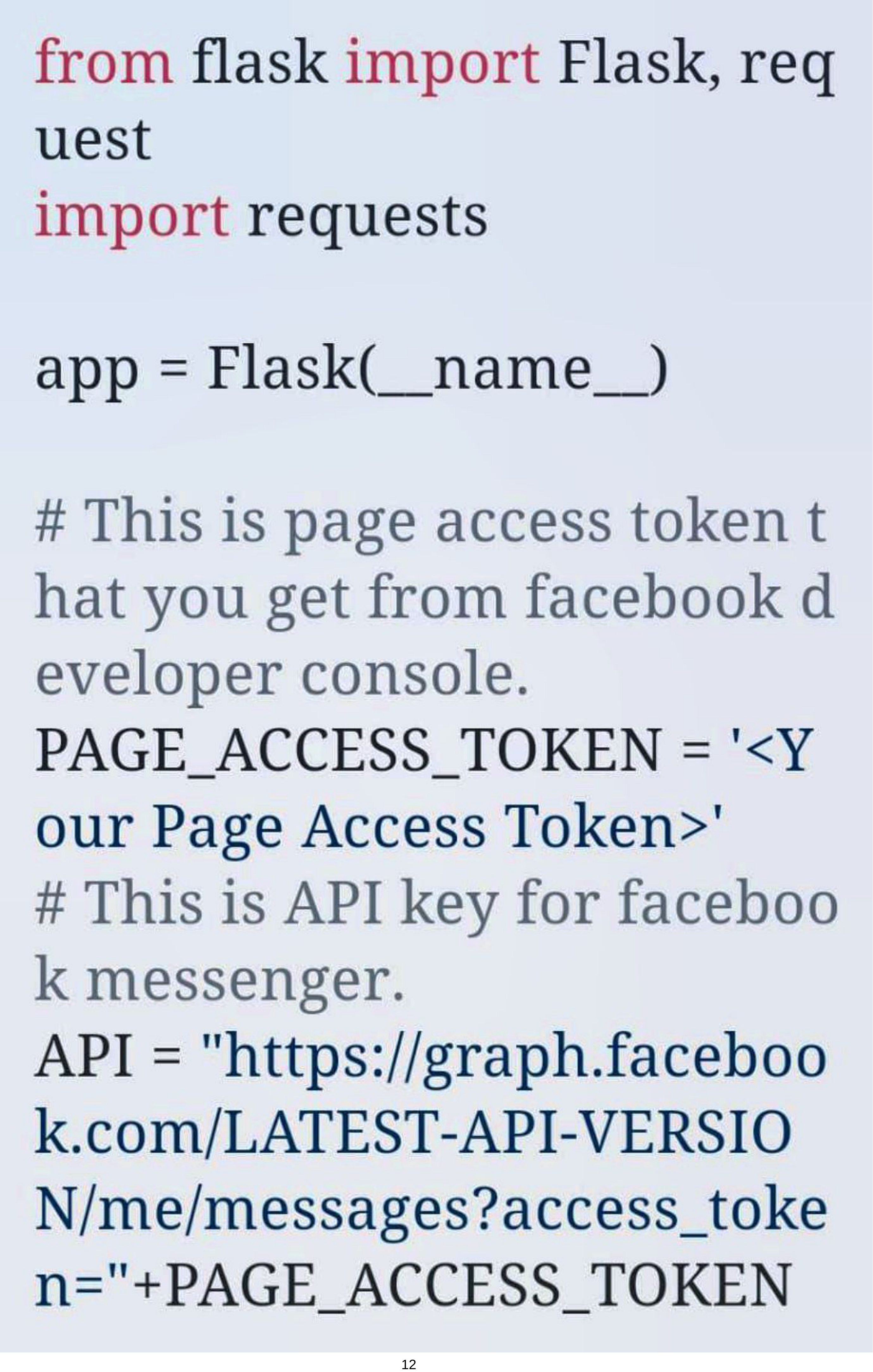
To protect your security, ONLY share this token with app developers you trust.

This token will only be shown once, so keep it safe. If it gets lost, you'll need to create a new one. Anyone could potentially use this token to impersonate this page, depending on the privacy settings of your app. If you wish to revoke all previously generated tokens for this app, you can go to Business Integrations settings. Learn More

☑ I Understand

[redacted]... Copy

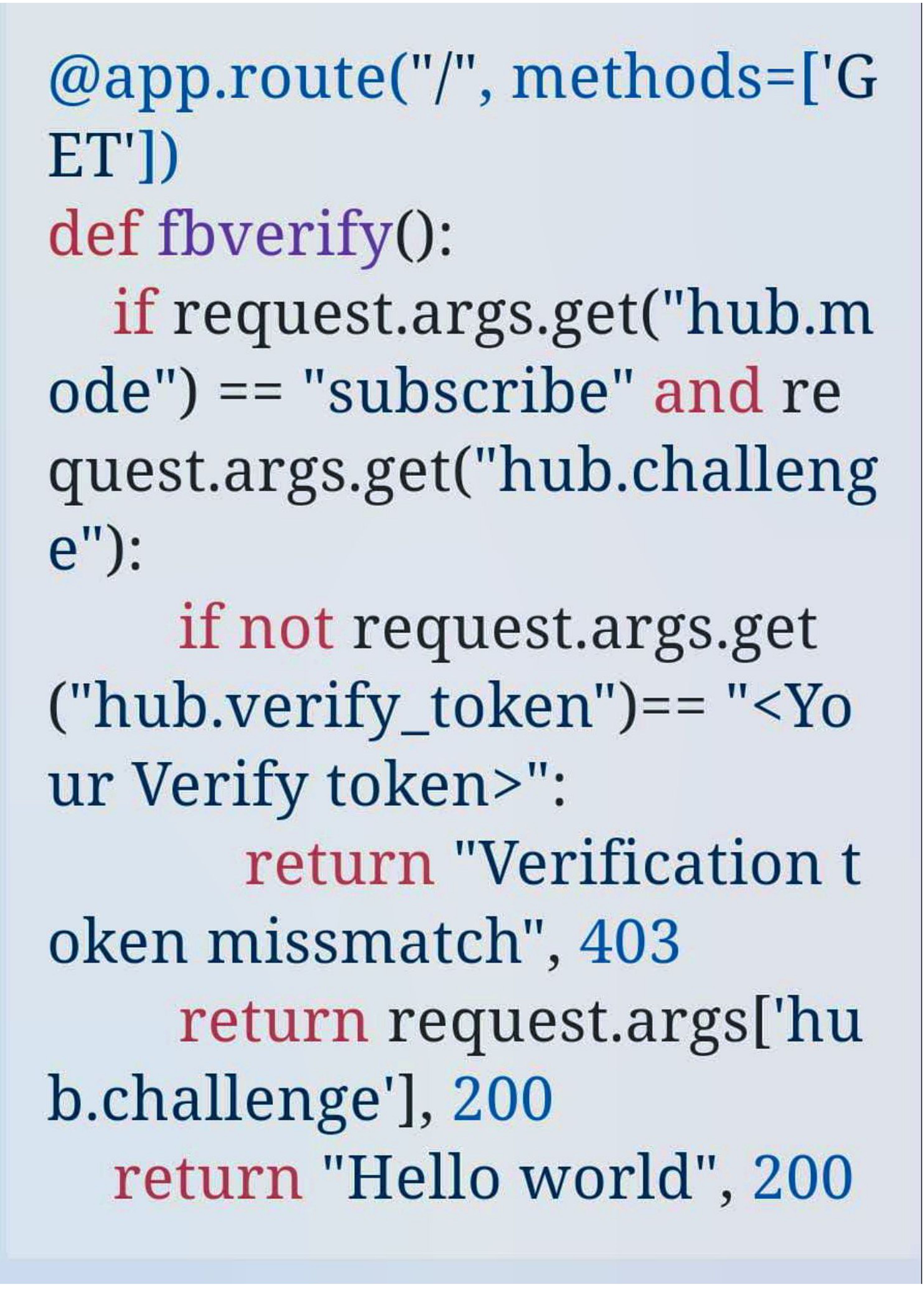
Done



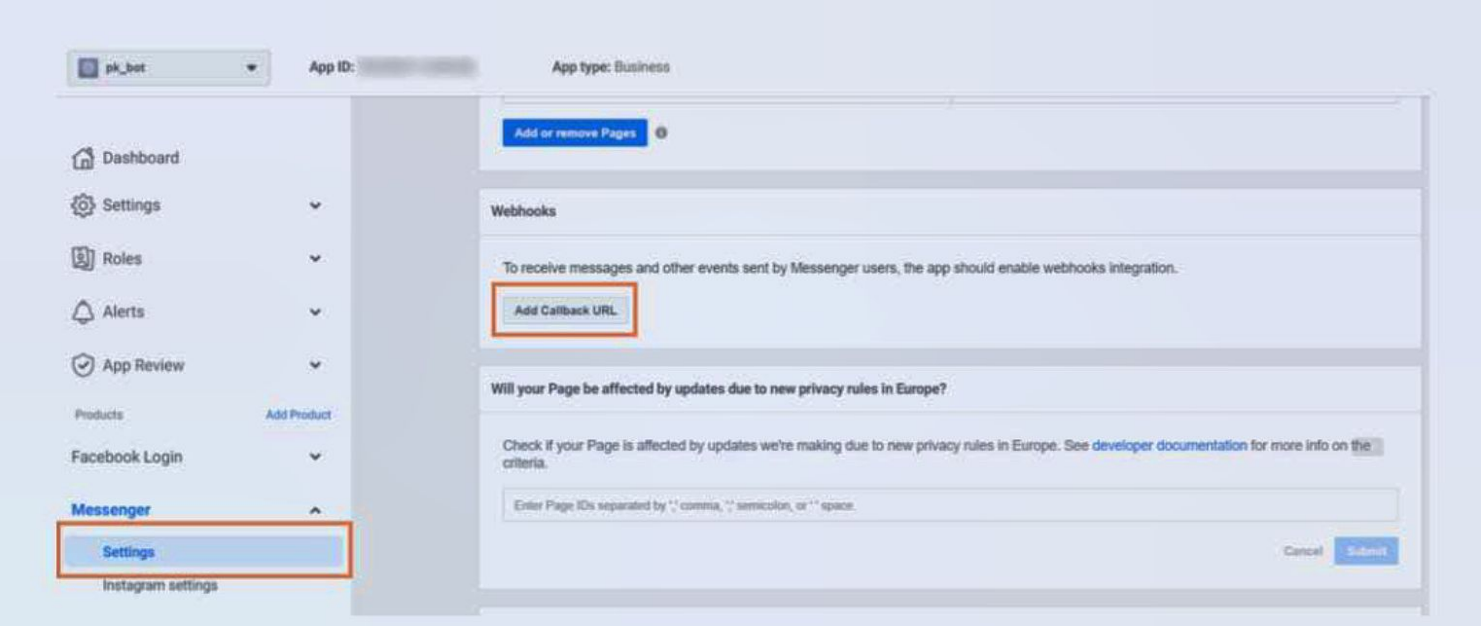
```
from flask import Flask, request
import requests

app = Flask(__name__)

# This is page access token that you get from facebook developer console.
PAGE_ACCESS_TOKEN = '<Your Page Access Token>'
# This is API key for facebook messenger.
API = "https://graph.facebook.com/LATEST-API-VERSION/me/messages?access_token="+PAGE_ACCESS_TOKEN
```



```
@app.route("/", methods=['GET'])
def fbverify():
    if request.args.get("hub.mode") == "subscribe" and request.args.get("hub.challenge"):
        if not request.args.get("hub.verify_token")== "<Your Verify token>":
            return "Verification token missmatch", 403
        return request.args['hub.challenge'], 200
    return "Hello world", 200
```

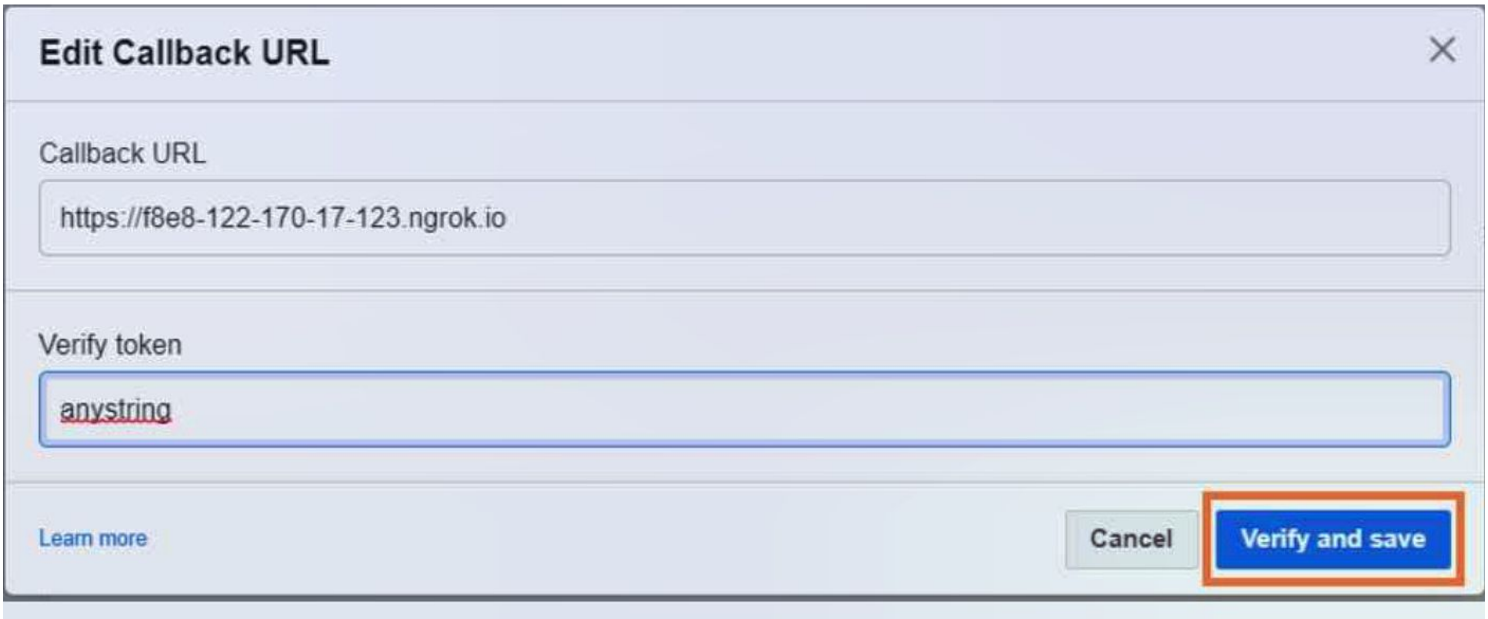


To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Will your Page be affected by updates due to new privacy rules in Europe?

Check if your Page is affected by updates we're making due to new privacy rules in Europe. See developer documentation for more info on the criteria.

Enter Page IDs separated by ',' comma, ';' semicolon, or ' ' space.



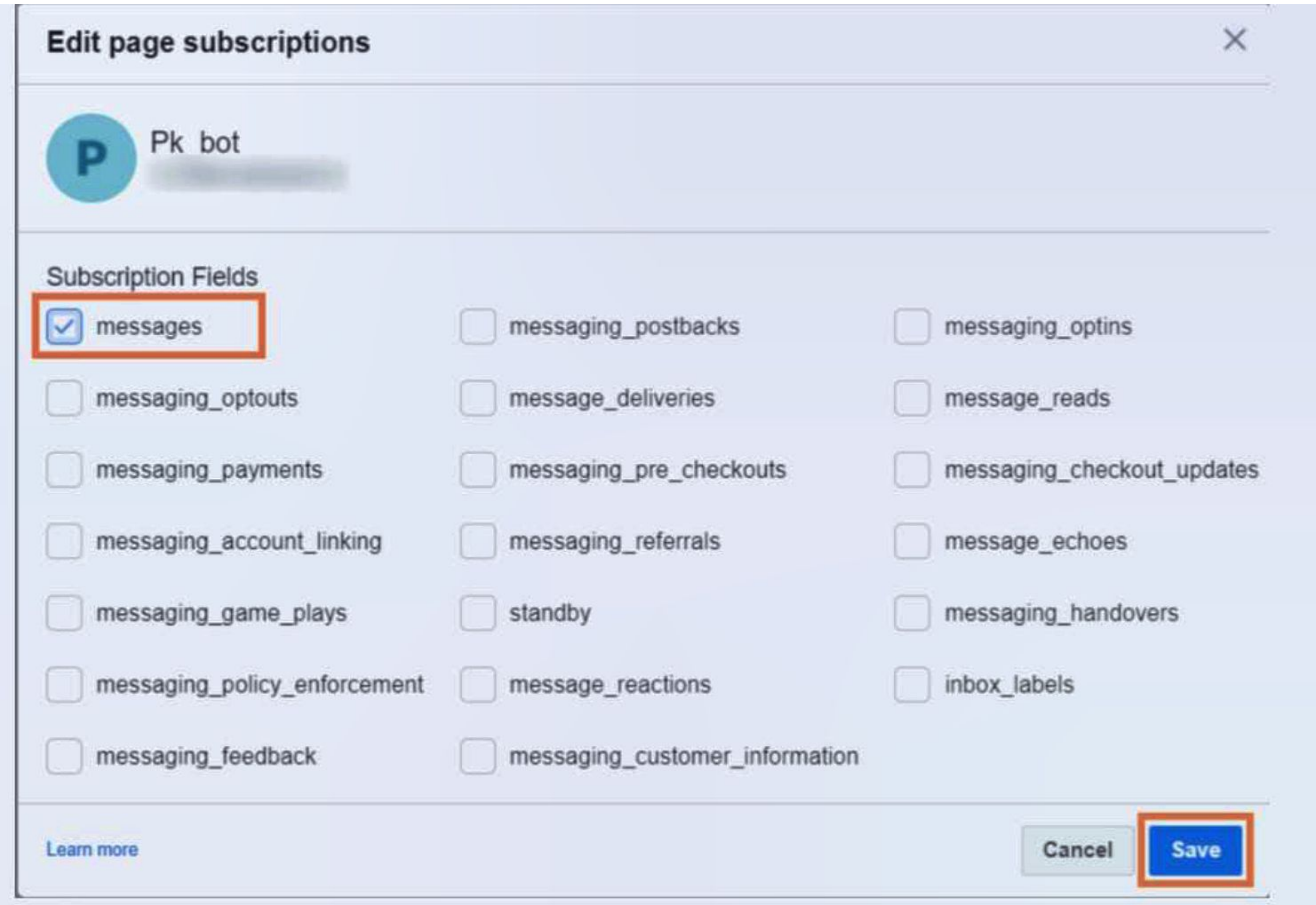
Edit Callback URL

Callback URL

https://f8e8-122-170-17-123.ngrok.io

Verify token

anystring



Edit page subscriptions

Pk bot

[redacted]

Subscription Fields

☑ messages

□ messaging_optouts

□ messaging_payments

□ messaging_account_linking

□ messaging_game_plays

□ messaging_policy_enforcement

□ messaging_feedback

□ messaging_postbacks

□ message_deliveries

□ messaging_pre_checkouts

□ messaging_referrals

□ standby

□ message_reactions

□ messaging_customer_information

□ messaging_optins

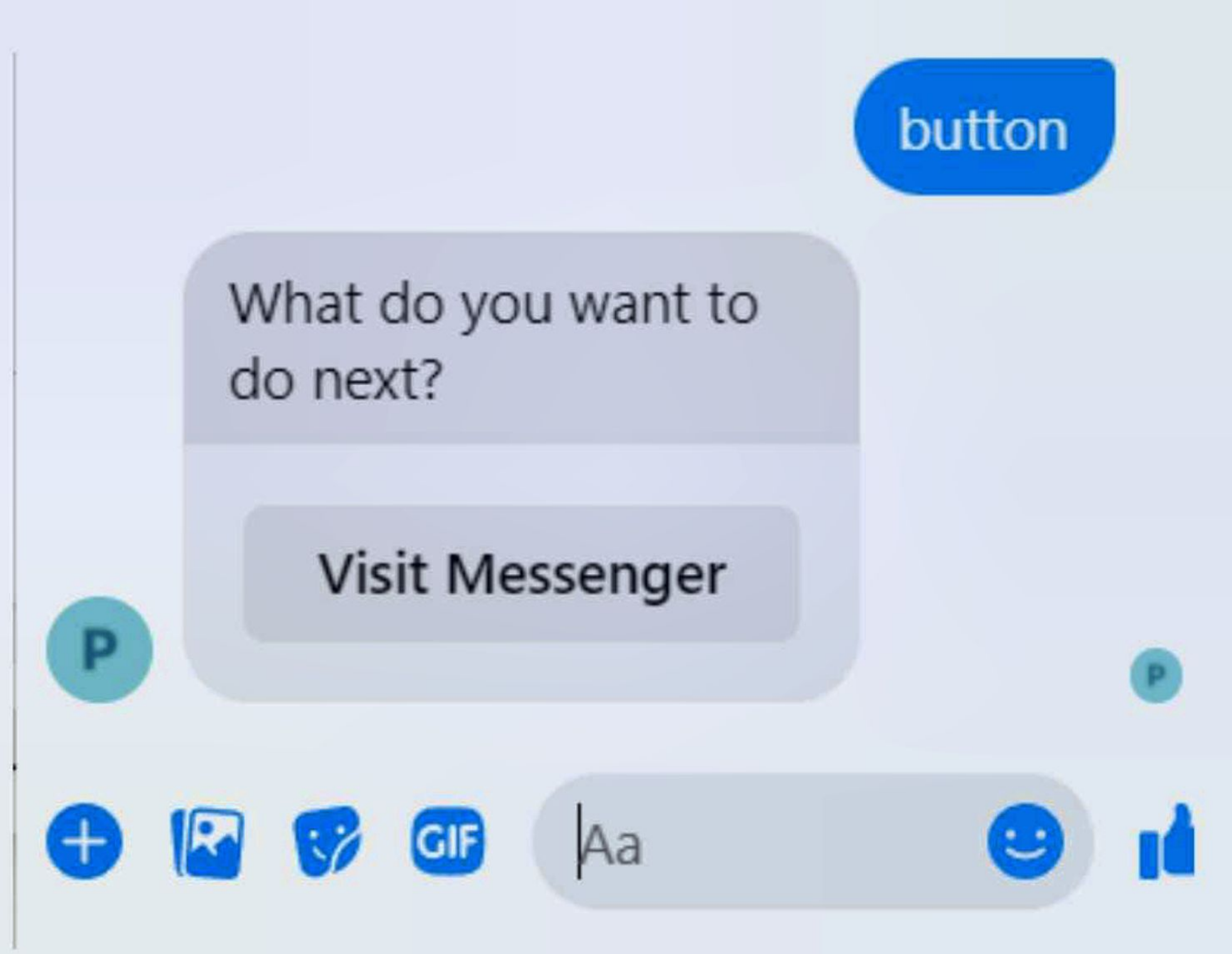
□ message_reads

□ messaging_checkout_updates

□ message_echoes

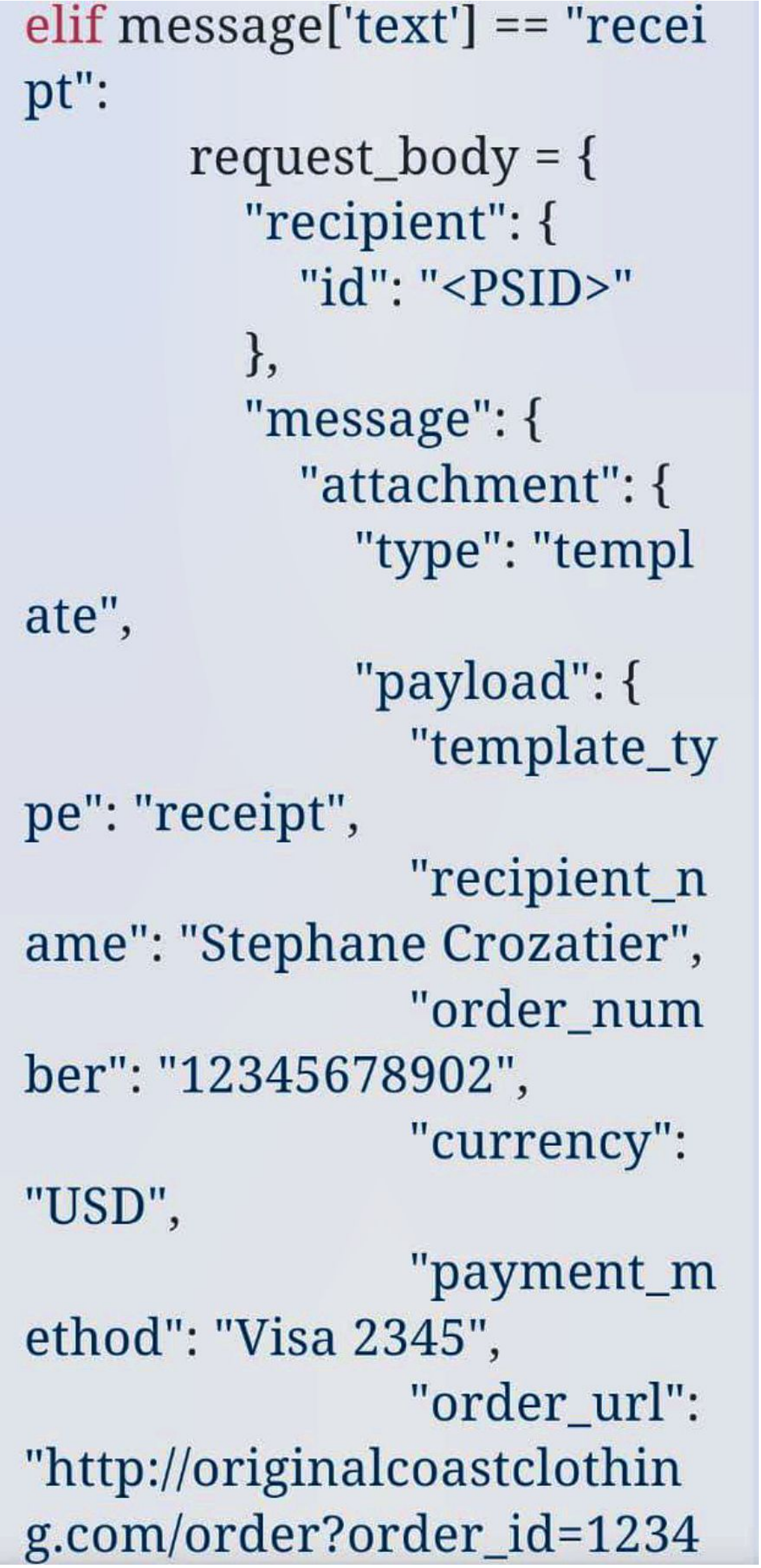
□ messaging_handovers

□ inbox_labels

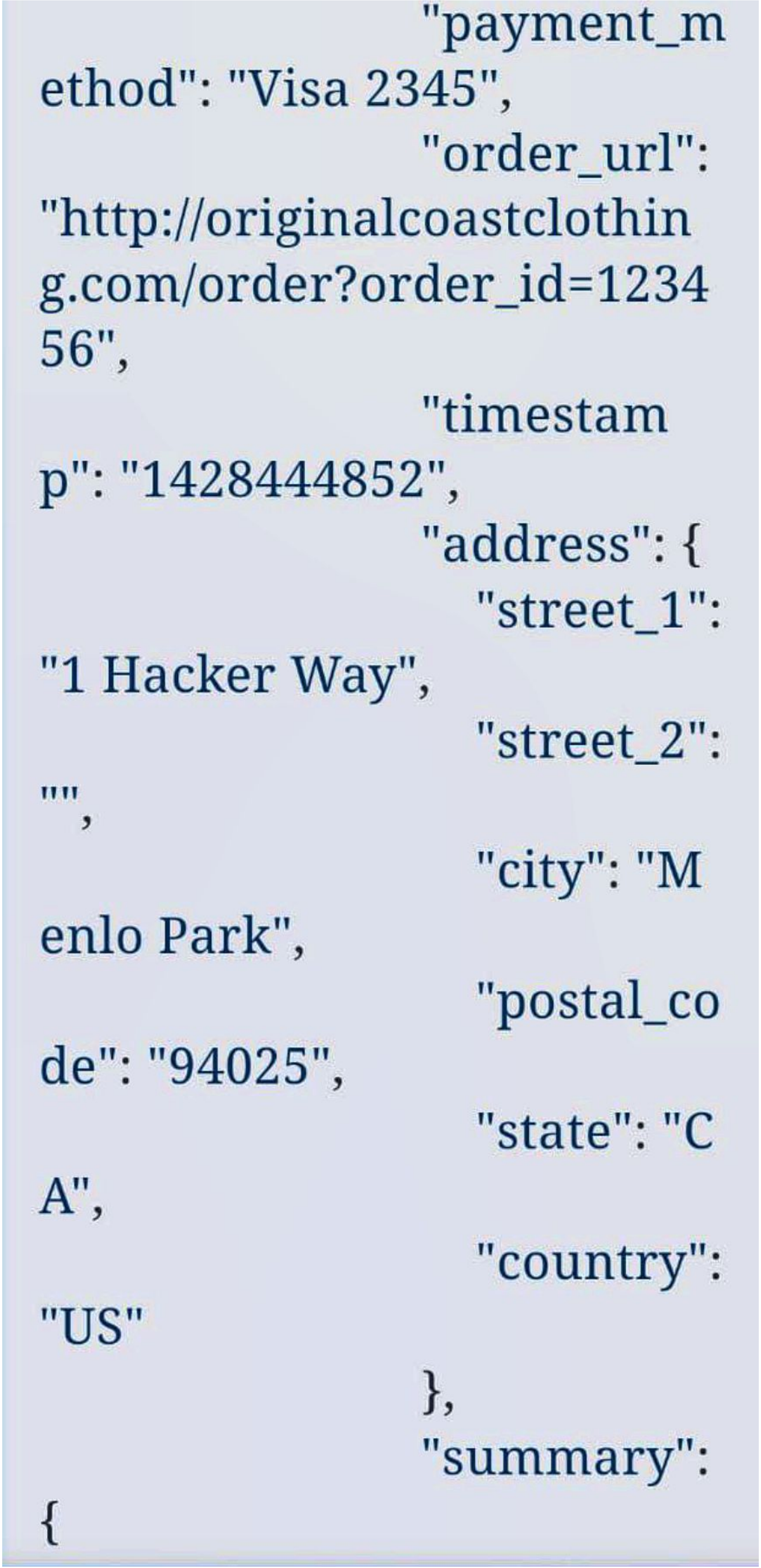


button

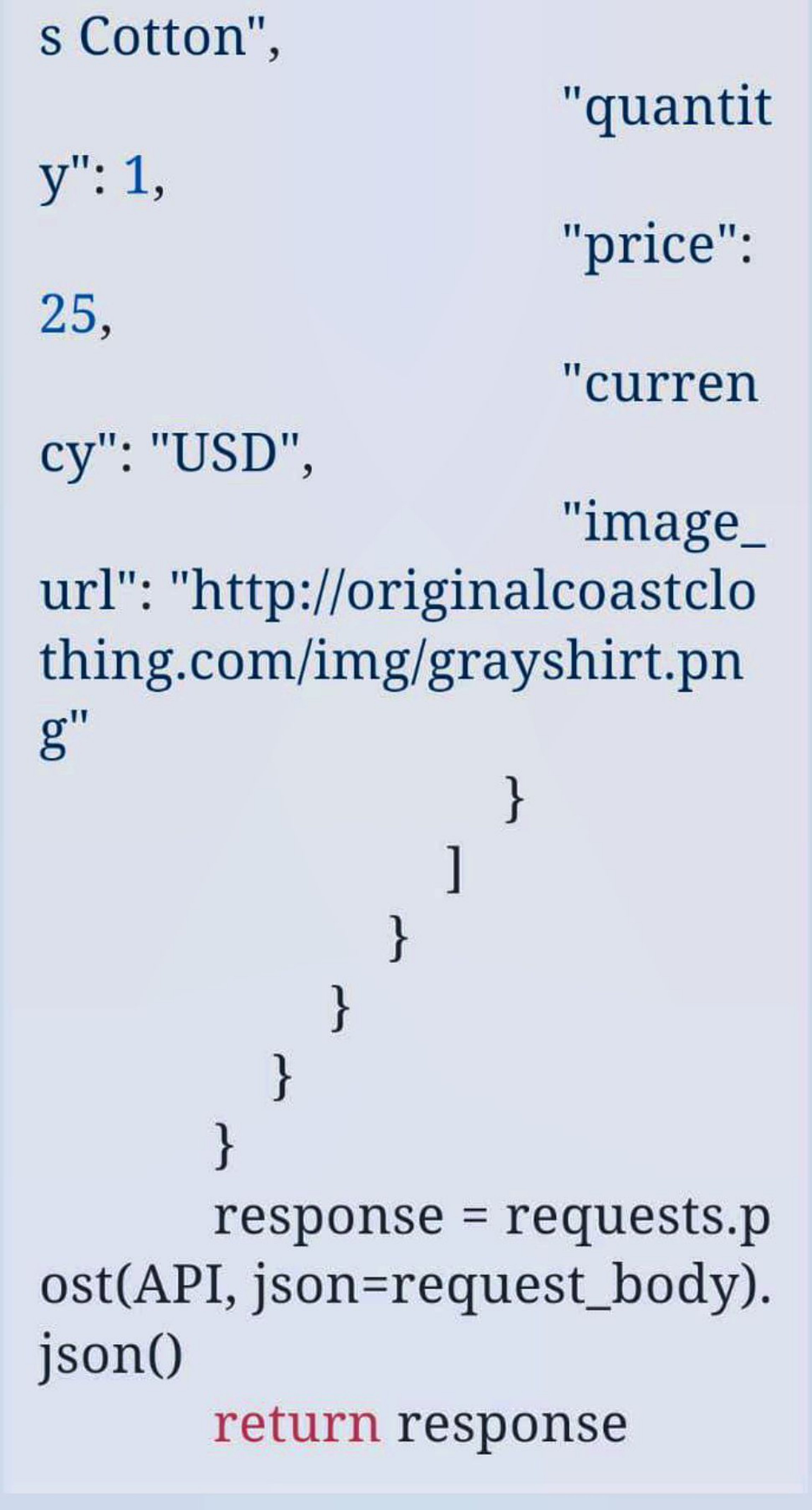
What do you want to do next?



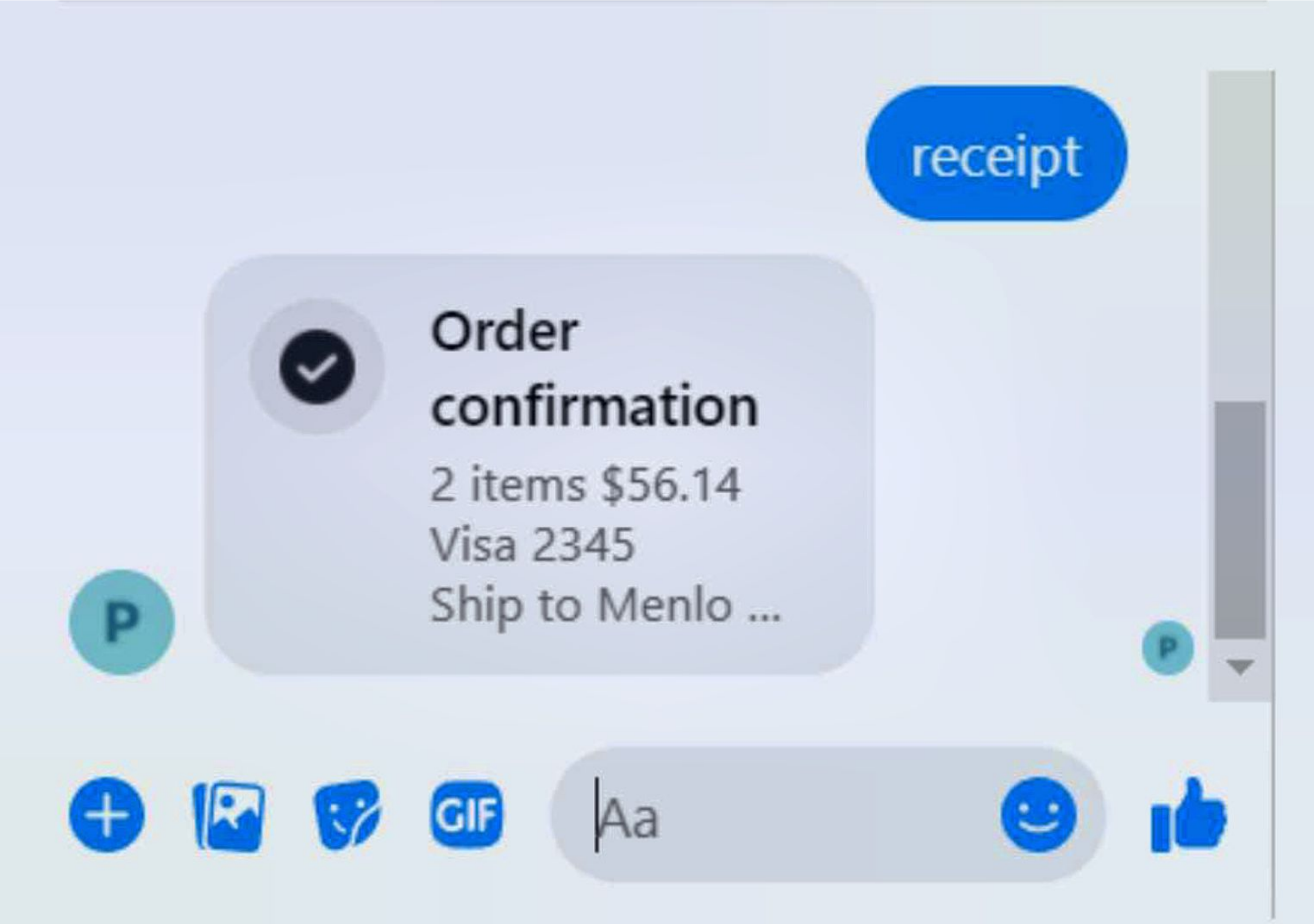
```
elif message['text'] == "recei
pt":
        request_body = {
            "recipient": {
                "id": "<PSID>"
            },
            "message": {
                "attachment": {
                    "type": "templ
ate",
                    "payload": {
                        "template_ty
pe": "receipt",
                        "recipient_n
ame": "Stephane Crozatier",
                        "order_num
ber": "12345678902",
                        "currency":
"USD",
                        "payment_m
ethod": "Visa 2345",
                        "order_url":
"http://originalcoastclothin
g.com/order?order_id=1234
```



```
                    "payment_m
ethod": "Visa 2345",
                    "order_url":
"http://originalcoastclothin
g.com/order?order_id=1234
56",
                    "timestam
p": "1428444852",
                    "address": {
                       "street_1":
"1 Hacker Way",
                       "street_2":
"",
                       "city": "M
enlo Park",
                       "postal_co
de": "94025",
                       "state": "C
A",
                       "country":
"US"
                    },
                    "summary":
{
```



```
s Cotton",
                                "quantit
y": 1,
                                "price":
25,
                                "curren
cy": "USD",
                                "image_
url": "http://originalcoastclo
thing.com/img/grayshirt.pn
g"
                            }
                        ]
                    }
                }
            }
        }
        response = requests.p
ost(API, json=request_body).
json()
        return response
```



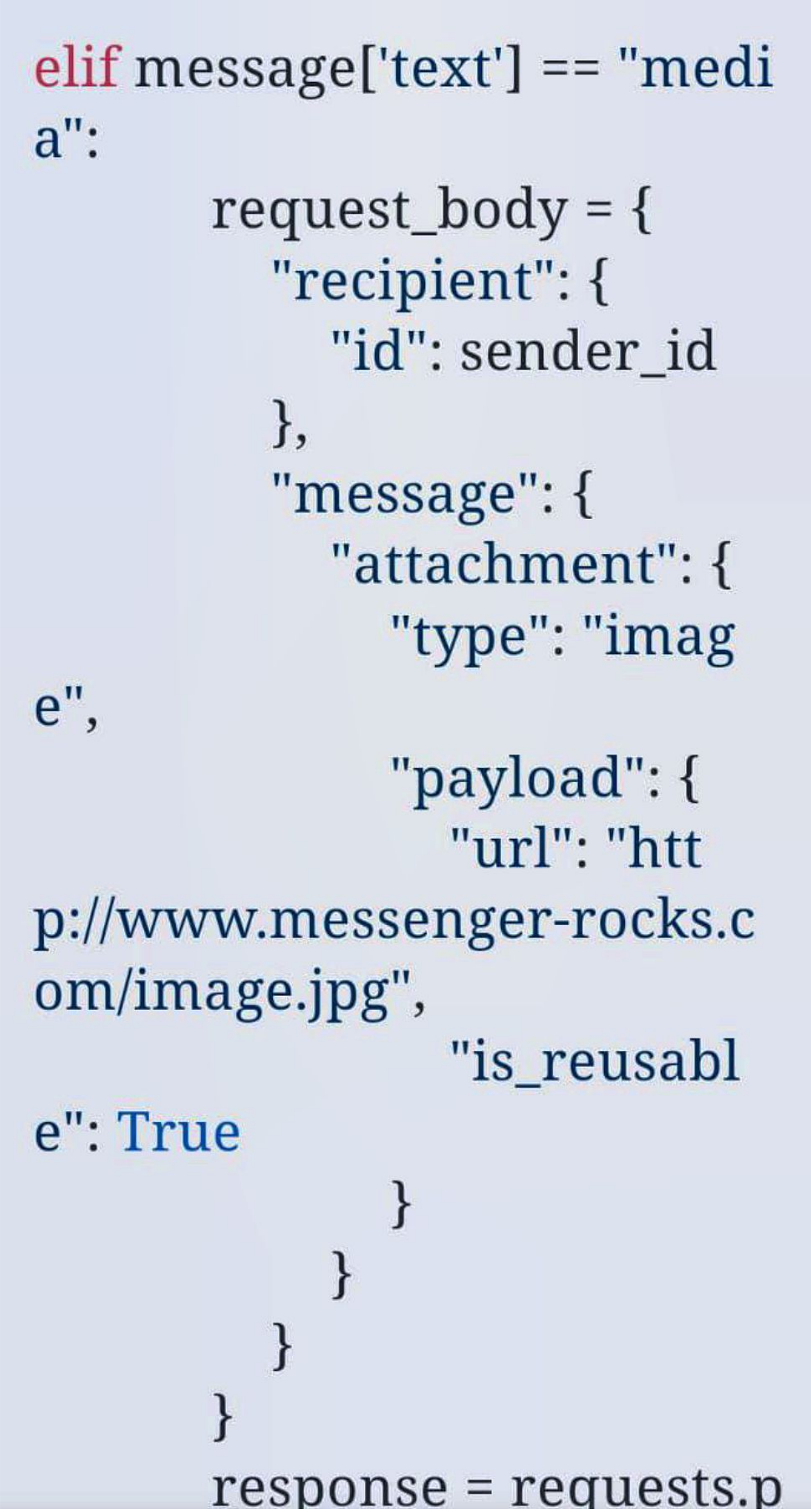
receipt

**Order confirmation**

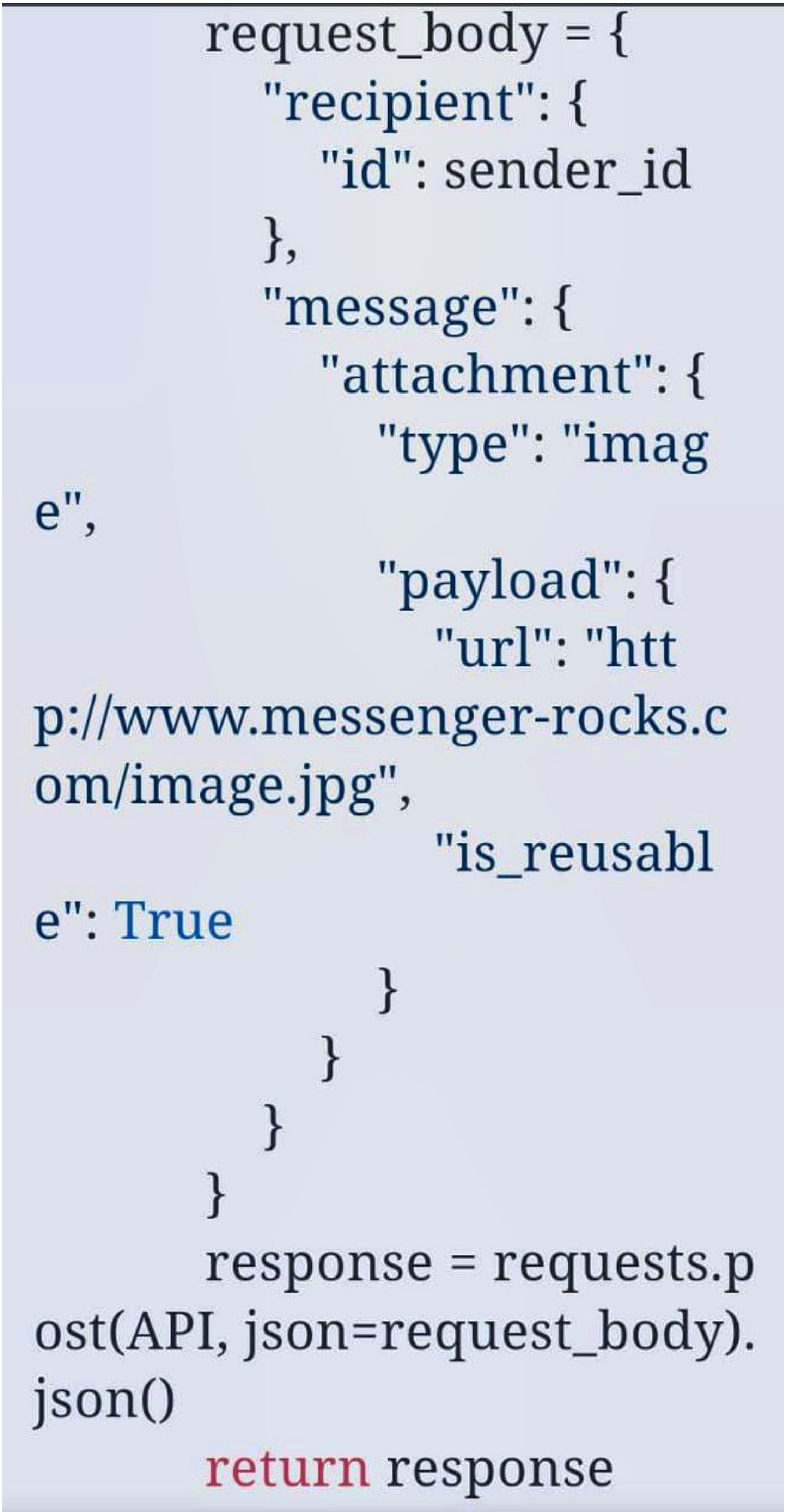
2 items $56.14

Visa 2345

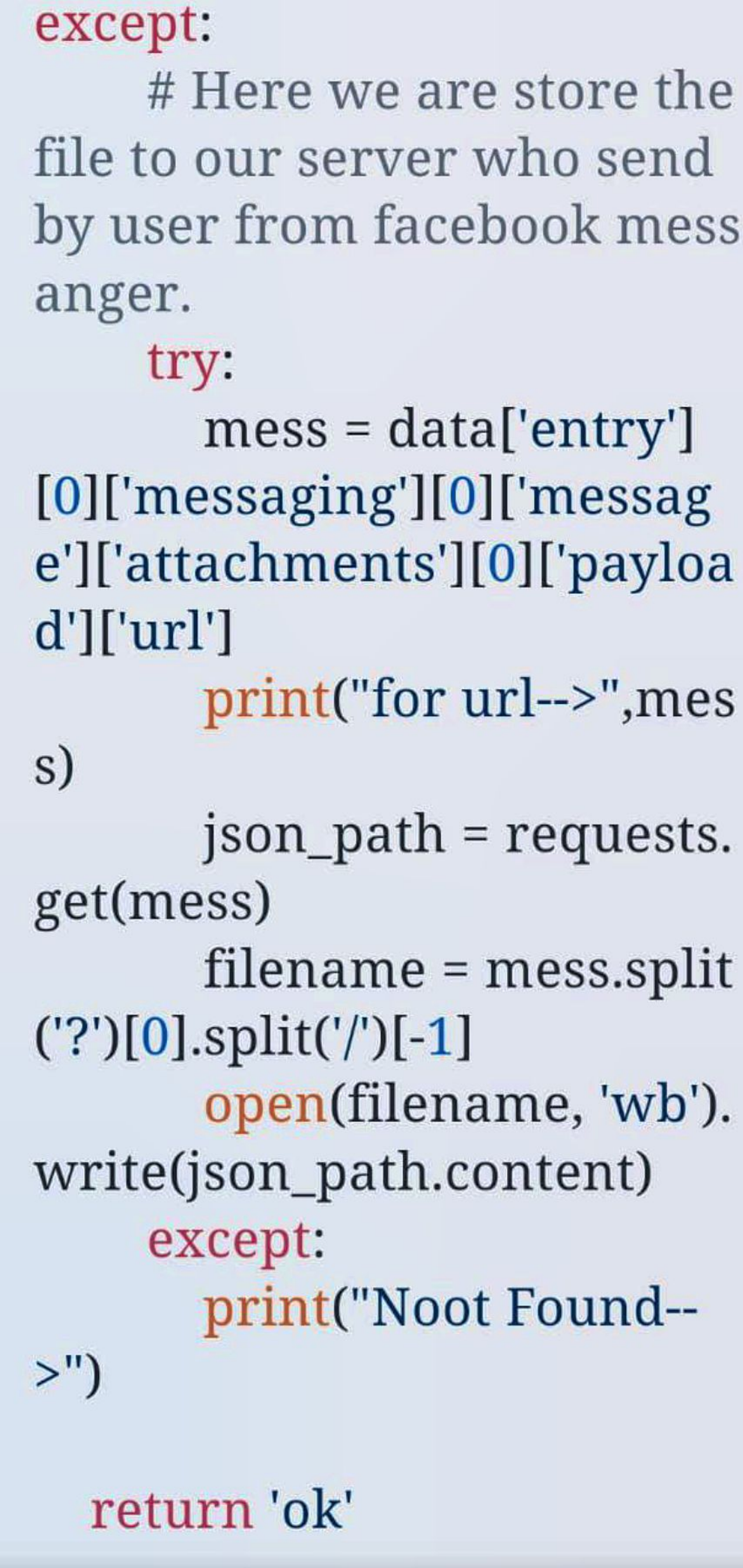
Ship to Menlo ...



```
elif message['text'] == "medi
a":
        request_body = {
           "recipient": {
              "id": sender_id
           },
           "message": {
              "attachment": {
                 "type": "imag
e",
                 "payload": {
                    "url": "htt
p://www.messenger-rocks.c
om/image.jpg",
                    "is_reusabl
e": True
                 }
              }
           }
        }
        response = requests.p
```



```
        request_body = {
           "recipient": {
              "id": sender_id
           },
           "message": {
              "attachment": {
                 "type": "imag
e",
                 "payload": {
                    "url": "htt
p://www.messenger-rocks.c
om/image.jpg",
                    "is_reusabl
e": True
                 }
              }
           }
        }
        response = requests.p
ost(API, json=request_body).
json()
        return response
```



```
except:
        # Here we are store the
file to our server who send
by user from facebook mess
anger.
        try:
            mess = data['entry']
[0]['messaging'][0]['messag
e']['attachments'][0]['payloa
d']['url']
            print("for url-->",mes
s)
            json_path = requests.
get(mess)
            filename = mess.split
('?')[0].split('/')[-1]
            open(filename, 'wb').
write(json_path.content)
        except:
            print("Noot Found--
>")

    return 'ok'
```