

CONNECT WEBHOOK

Version	Date	Author	Description
1.0	31-Oct-2019		

Table of contents

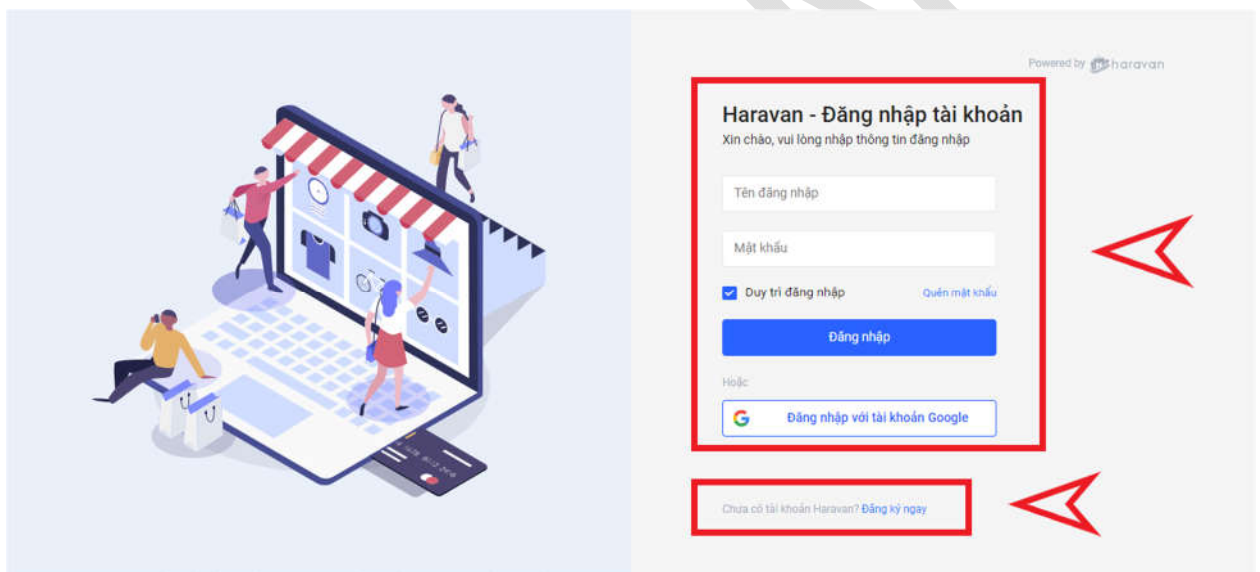
Connect webhook.....	3
1. Step 1: Create app	3
2. Step 2: Use ngrok to register webhook	5
3. Step 3: Register webhook	6
4. Step 4: Call api subscribe webhook	9
4.1. Request.....	10
4.2. Response	10
4.3. Use Postman.....	11
5. Step 5: Receive notifications from webhook	11
6. Step 6: Authenticate webhook	13
7. Note:.....	15

Connect webhook

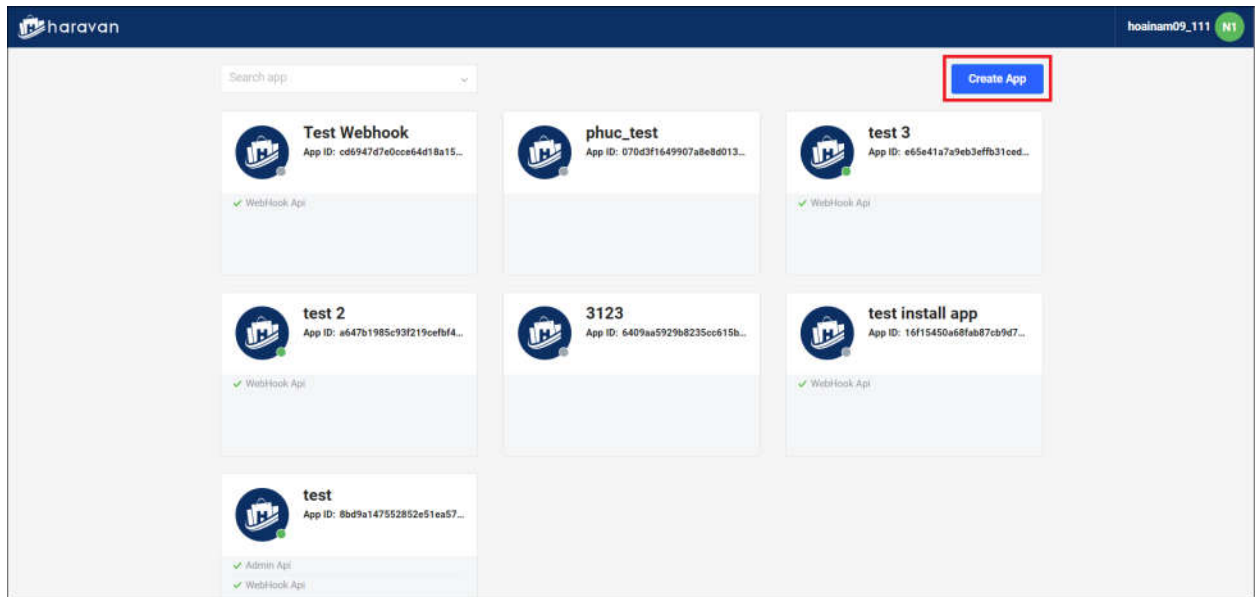
- You can register webhook to receive notifications about particular events in a shop.
- **Note :**
 - + Notification contains a JSON payload.
 - + Only support HTTPS.

1. Step 1: Create app

- Follow the link <https://developers.haravan.com/apps> (If you haven't logged in, the system will redirect to sign in in <https://accounts.haravan.com>, and at this page you can sign in or sign up).

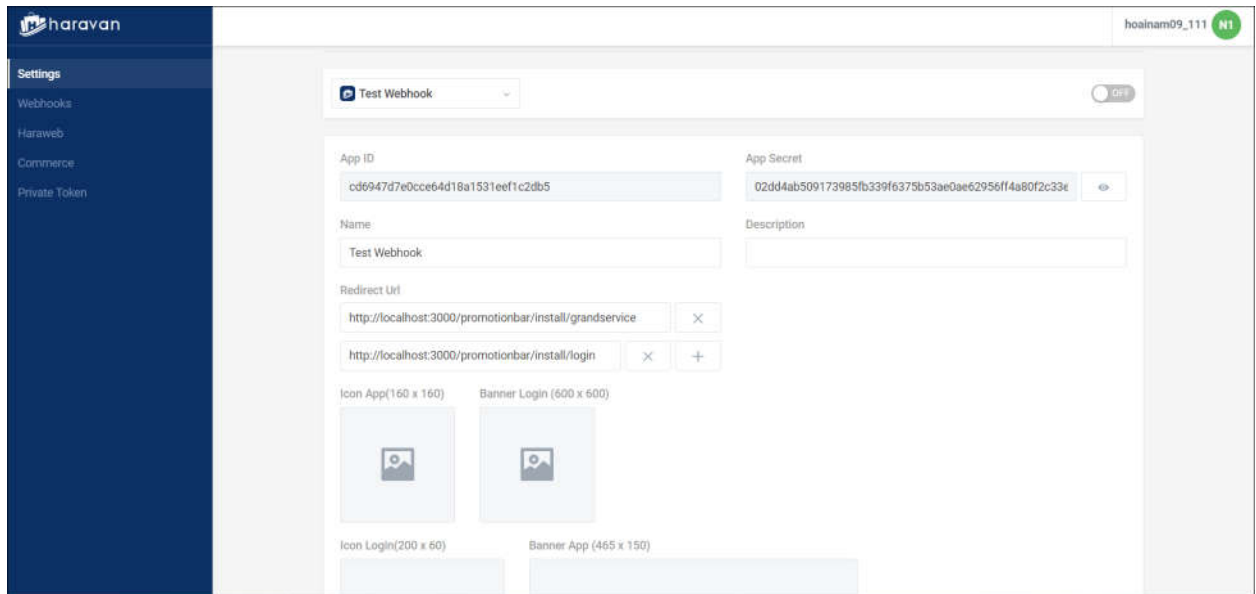


- After sign in/sign up successful, the system will show a list of your applications, click "Create App" button to create a new application.

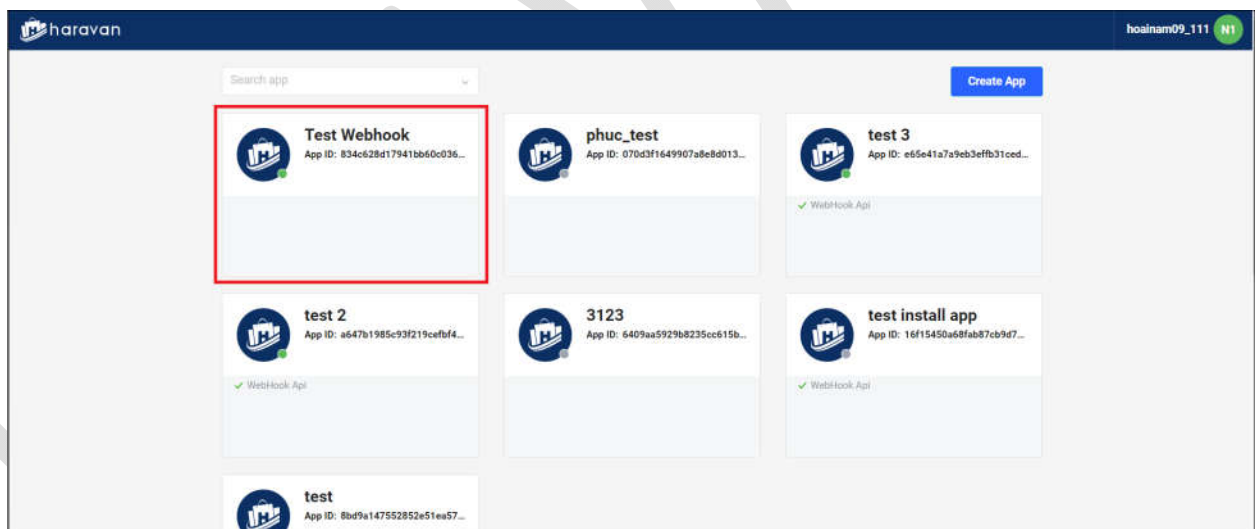


- Screen for create application.

- + Name: application name.
 - + Description: application description.
 - + Redirect Url: application domain, the system will redirect to this url when starting application.
- Create success.
 - + Some field need to save in config file to use: App Id, App Secret, Redirect Url.



- After created, your application will be appeared on list <https://developers.haravan.com/apps>.



2. Step 2: Use ngrok to register webhook

- Ngrok (<https://ngrok.com/>) creates a secure tunnel on your local machine along with a public URL you can use for browsing your local site.
- If you're running your local web server on port 3000, in terminal, you'd type in: ngrok http 3000.
- **Note:**
 - + A url can be available for 8 hours
 - + Only use **https://**

- + Use ngrok's web inspection interface to understand the HTTP request and response traffic over your tunnel : <http://localhost:4040>

```
F:\My_soft\work_soft\ngrokexe - ngrok http 3000
ngrok by @inconshreveable (Ctrl+C to quit)

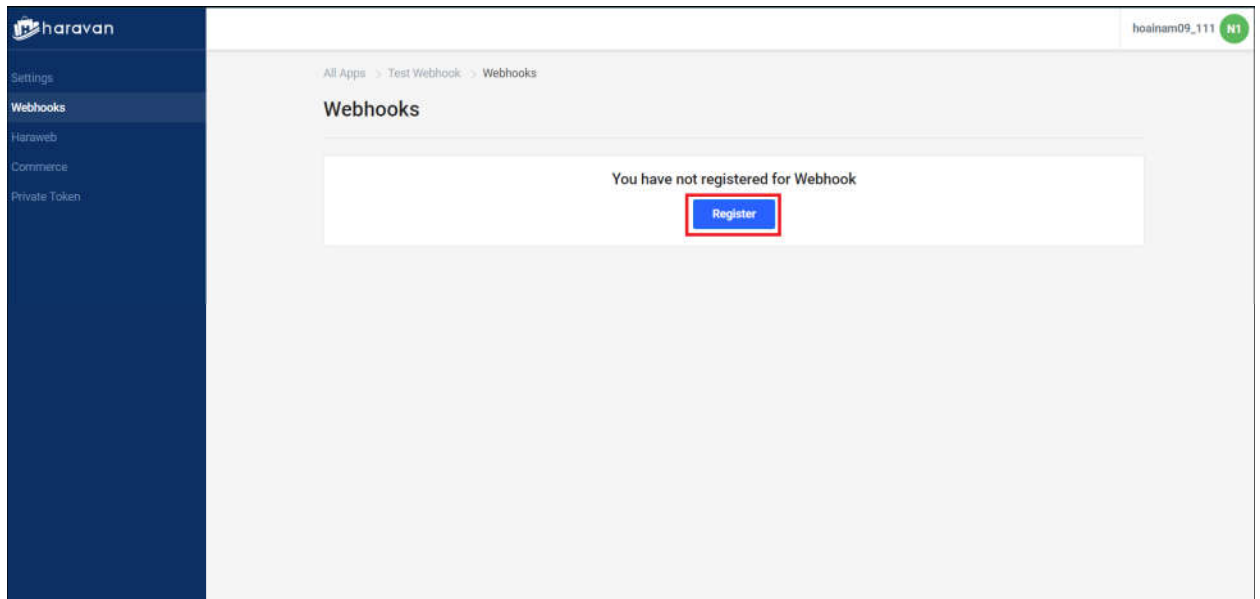
Session Status      online
Session Expires     6 hours, 20 minutes
Update              update available (version 2.3.35, Ctrl-U to update)
Version             2.3.27
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://e01249e5.ngrok.io -> http://localhost:3000
                    https://e01249e5.ngrok.io -> http://localhost:3000

Connections
  ttl    opn    rt1    rt5    p50    p90
    2     0     0.00  0.00  5.32  5.32

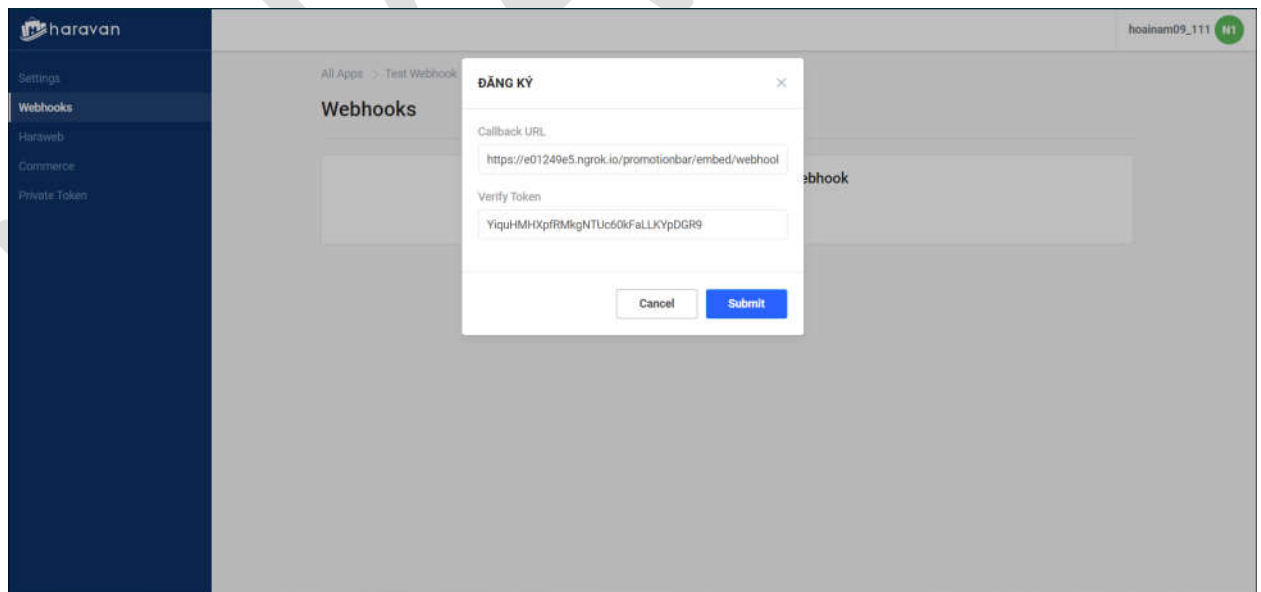
HTTP Requests
-----
GET /promotionbar/embed/webhooks 200 OK
GET /promotionbar/embed/webhooks 200 OK
```

3. Step 3: Register webhook

- Choose “Webhooks” on the sidebar, click “Register” button.



- You need to fill in some information :
 - + Callback URL: Webhooks will POST a message with data to this URL when certain things happen. Make sure the url is configured on your application to receive webhooks.
 - + Verify Token: is random string and has at least 1 character, because it's used for authenticating with an app, make sure this token matches the string configured on your app.



- When you submit, the system will call to your app via the Callback URL.
- When callback to your app, the system will send an object to verify.

```
{  
  "hub.mode": "subscribe",  
  "hub.verify_token": "YiquHMHXpfRMkgNTUc60kFaLLKYpDGR9",  
  "hub.challenge": "23236cb70fe64c1ebcf047574a41eb14"  
}
```

- You need to compare hub.verify_token (in object) with verify_token (in your app).
 - + If not match, response status 401.
 - + If match, response status 200 and value of hub.challenge.

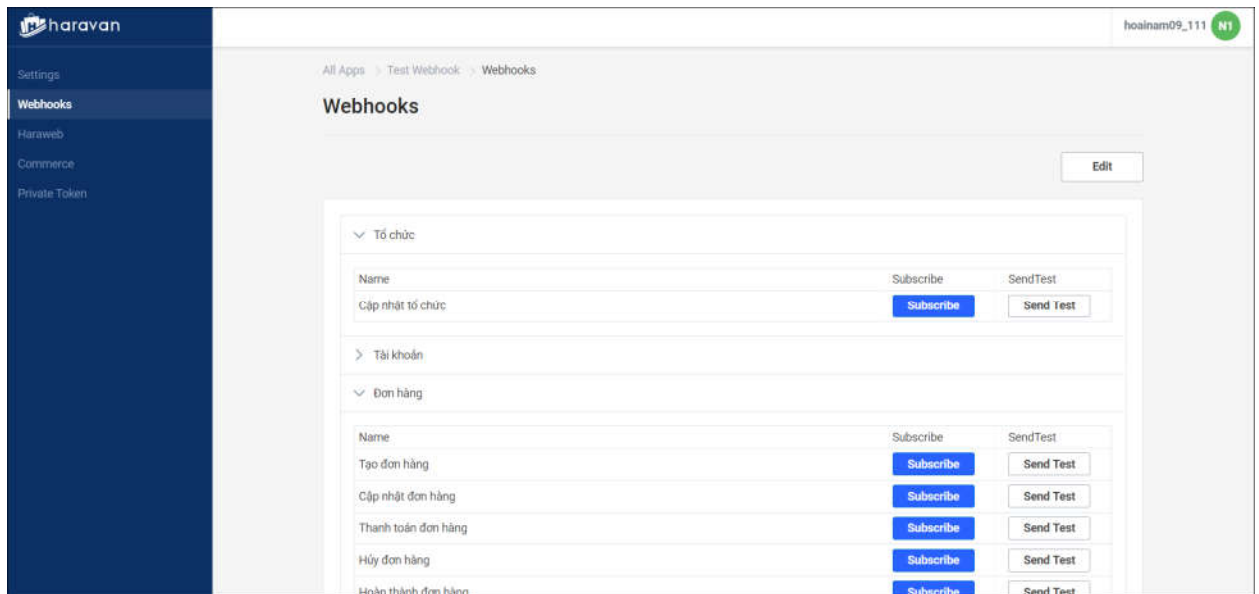
The screenshot shows a web browser interface with a 'All Requests' panel on the left and a detailed view of a GET request on the right. The 'All Requests' panel shows a single request: GET /phoneconvert/webhooks with a status of 200 OK and a duration of 2.9ms. The detailed view on the right shows the request parameters: hub.challenge (dd279ae60d9645e19c06a34ad24db650), hub.mode (subscribe), and hub.verify_token (YiquHMHXpfRMkgNTUc60kFaLLKYpDGR9). The response status is 200 OK, and the response body is 32 bytes of text/html. A red callout points to the '200 OK' status, and another red callout points to the 'hub.challenge' value in the response body.

response status 200

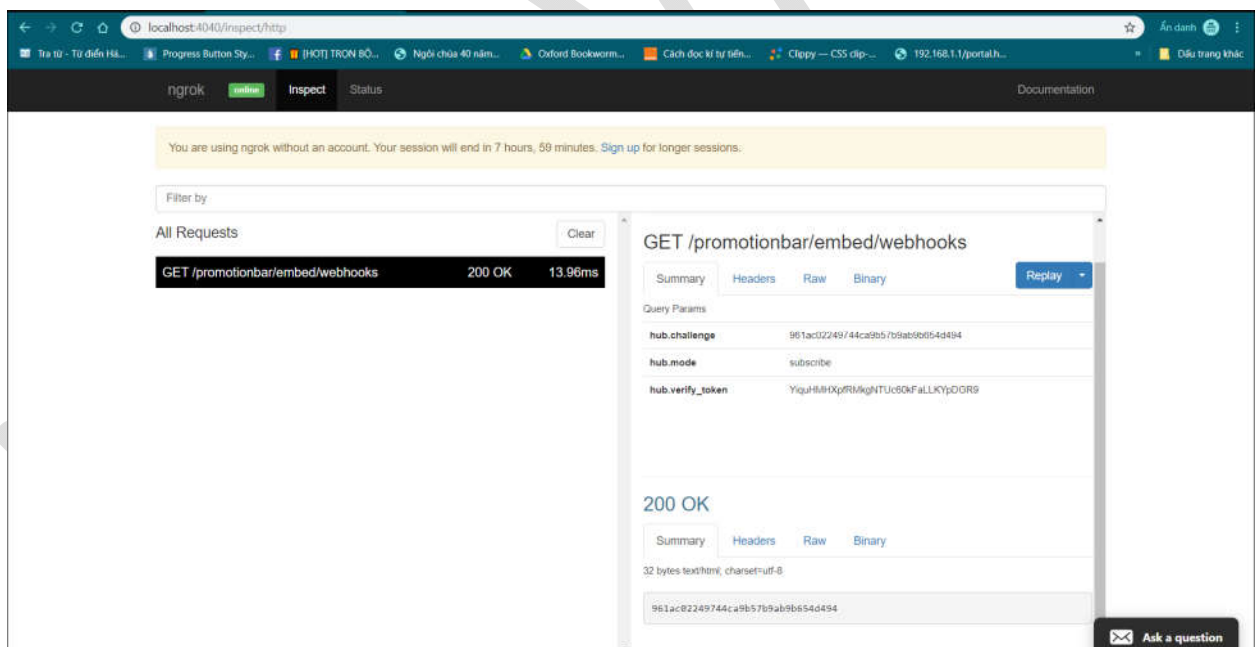
200 OK

hub.challenge

- After registered, you need to reload this page to see the list of events that can subscribe to get notifications.



- Use ngrok's web inspection interface to understand the HTTP request and response traffic over your tunnel : <http://localhost:4040>



- The purpose is to subscribe which events your app want receive notifications.

4. Step 4: Call api subscribe webhook

- Note :

- + First, you need Haravan's access_token to use Haravan's Api.
- + You also need right scope to get Haravan's access_token.
- + The scope to use webhook is: **wh_api**. (You need more scope for install and use other api).
 - Scope:
 - <https://docs.haravan.com/blogs/omni/apis-va-scope>
 - Access_token:
 - <https://docs.haravan.com/blogs/omni/huong-dan-cao-app-ket-noi-api>
 - https://www.youtube.com/watch?v=gzJ0_qbrvYM&feature=youtu.be
- Do this step after you have access_token.
- The purpose is to declare that your application can receive webhook notifications.

4.1.Request

Method	URL
POST	https://webhook.haravan.com/api/subscribe

Header:

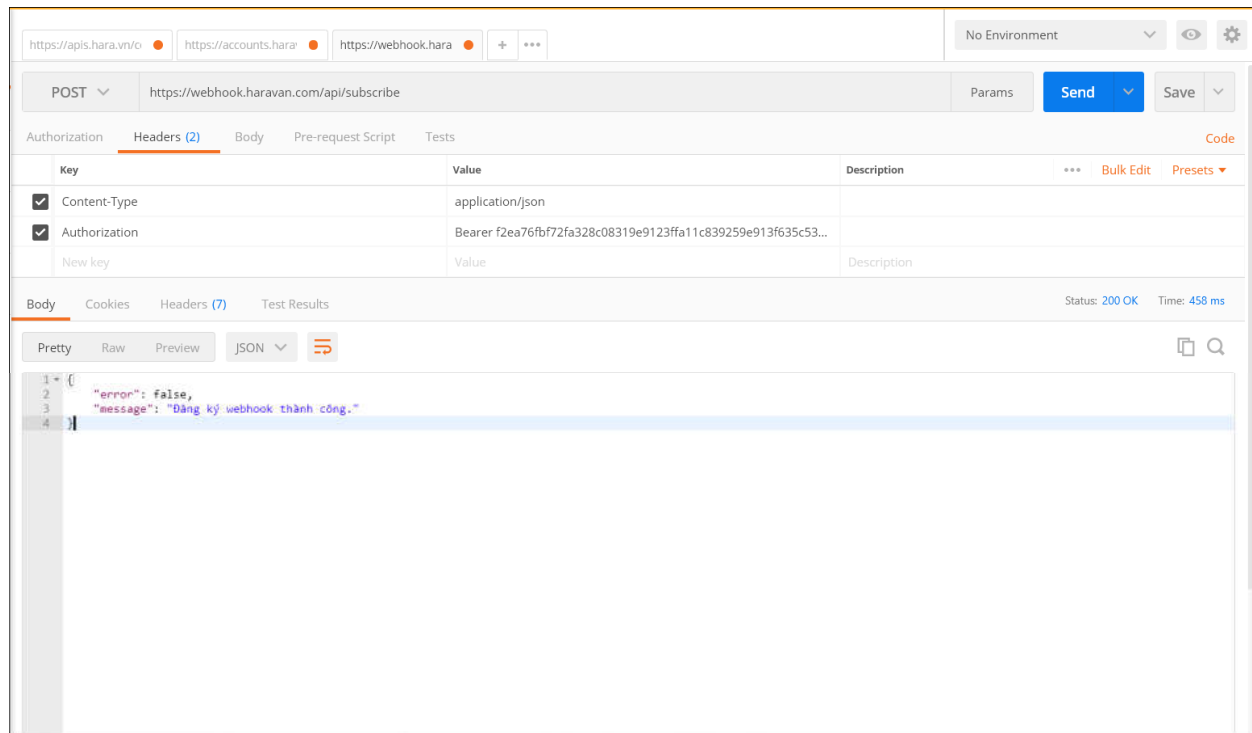
Content-Type: application/json

Authorization: Bearer + access_token

4.2.Response

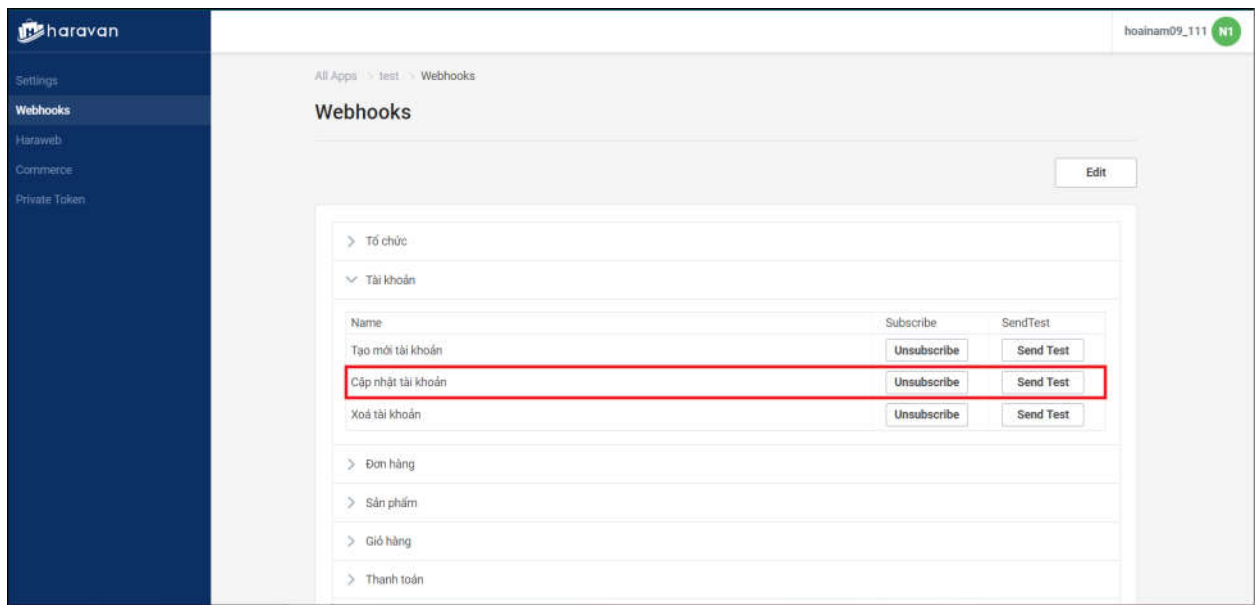
Status	Response
200	<pre>{ "error": false, "message": "Đăng ký webhook thành công." }</pre>
422	{"error": "Unprocessable Entity"}
401	Unauthorized
429	Too many requests
500	Something went wrong. Please try again later.

4.3. Use Postman

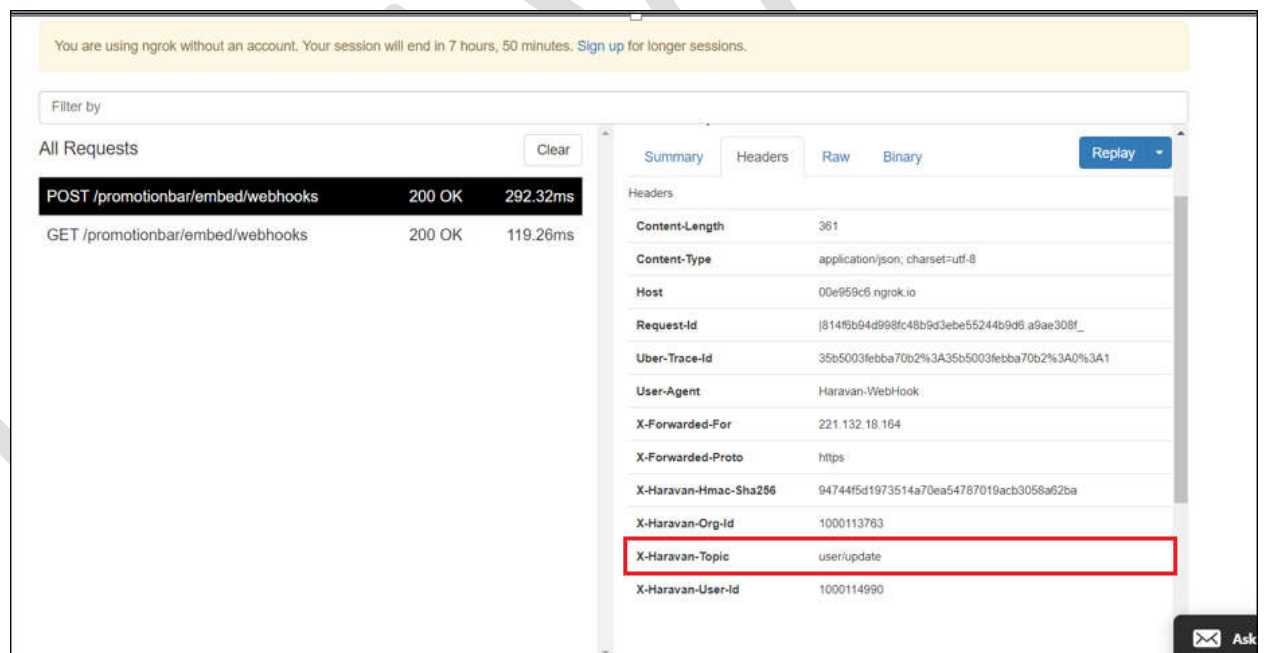


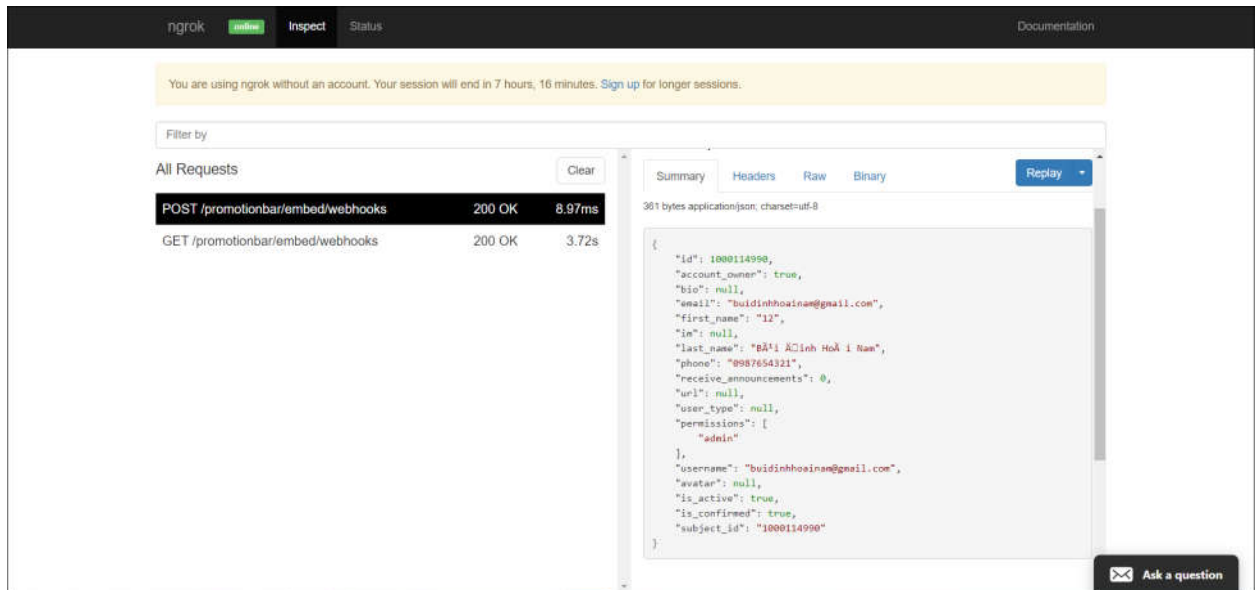
5. Step 5: Receive notifications from webhook

- After completing the above steps, webhooks will send request to notify your application with method POST.
- When sending notify, the webhook will include data in the request, but first you need to pay attention to:
 - + `org_id` : to know which shop
 - + `topic`: to know which notifications for which events.
- For example, we register an account update event



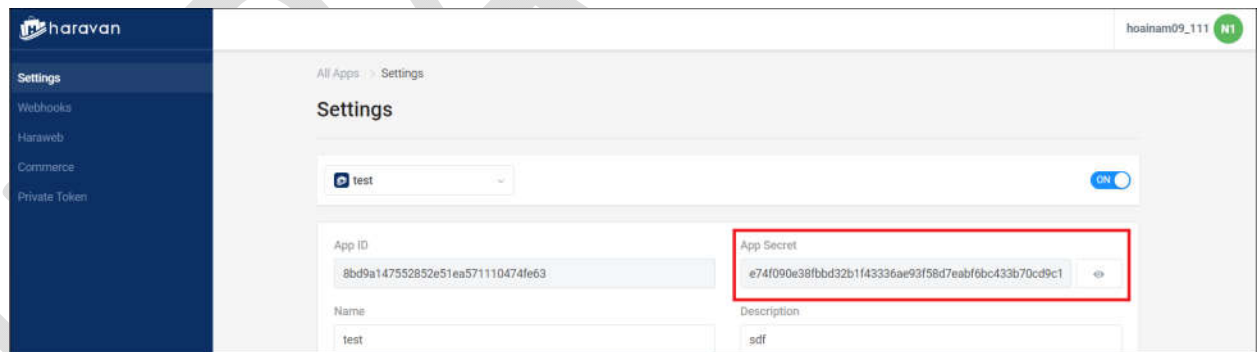
- When you update your account at <https://accounts.haravan.com/>, webhook will notify your application with the topic is "user / update" and data of your account via CallBack URL.



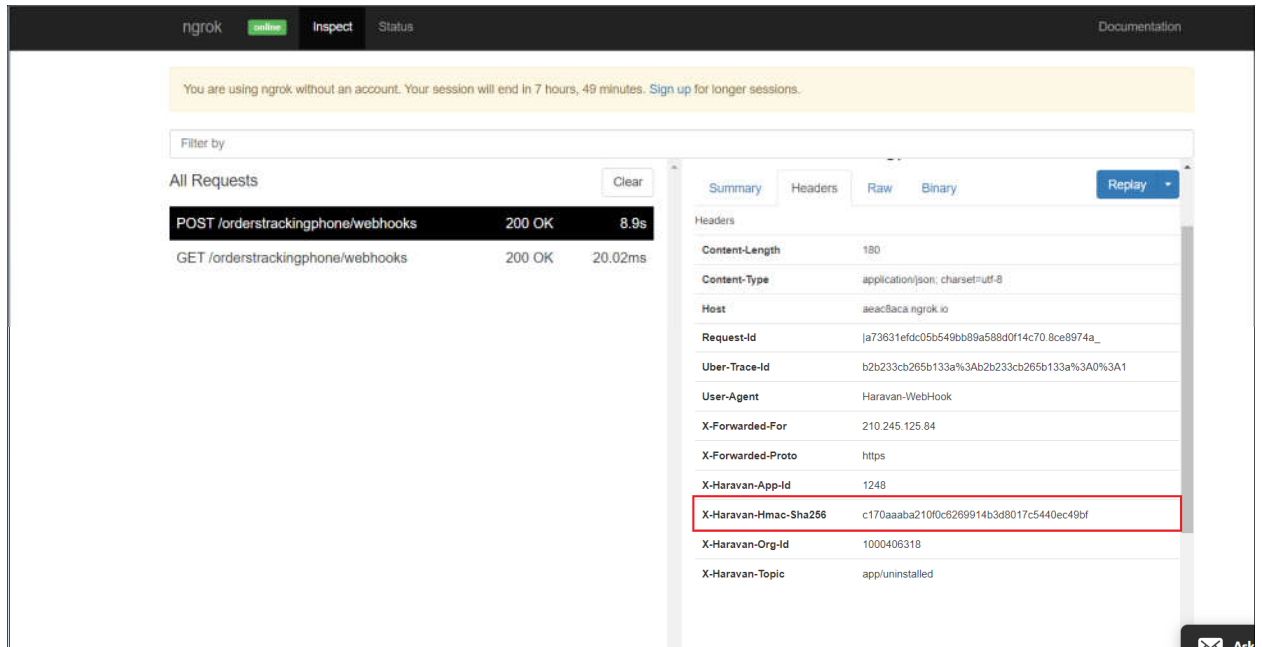


6. Step 6: Authenticate webhook

- When you receive data from webhook, you need to verify that it's data sent from the system.
 - + Choose "Settings" on the sidebar, copy App secret.



- + Next, get **X-Haravan-Hmac-SHA256**



+ To authenticate the webhook :

- First, we use `hash_mac` function with “sha1” algorithms to encode a string (data in webhook request + App Secret) to get the hash code.
- Second, use `hex` function to encode this hash.
- Finally, compare it with **X-Haravan-Hmac-SHA256** (header of webhook request).
 - If not matches, response status 401.
 - If matches, response status 200.

+ Example in PHP:

```
<?php
define('Haravan_APP_SECRET', 'my_shared_secret');

function verify_webhook($data, $hmac_header)
{
    $calculated_hmac = hex_encode(hash_hmac('sha1', $data, Haravan_APP_SECRET, true));
    return ($hmac_header == $calculated_hmac);
}

$hmac_header = $_SERVER['X-Haravan-Hmac-Sha256'];
$data = file_get_contents('php://input');
$verified = verify_webhook($data, $hmac_header);
error_log("Webhook verified: ".var_export($verified, true)); //check error.log to see the result
```

?>

7. Note:

- If you do not response status 200 when you received a notification from the webhook, the system will automatically send again.
- Remove ngrok if not used anymore to avoid system spam.
- When uninstall your application, it will lose subscribers, you need to call api subscriber like step 4 again to subscriber the webhook for your application every time you reinstall the app.