

CONNECT WEBHOOK

| Version | Date | Author | Description |
|------------|-------------|--------|---|
| 1.0 | 31-Oct-2019 | | |
| 1.1 | 26-Jun-2020 | | - Update Authenticate webhook - Add Step 7 : unsubscribe webhook |

Table of contents

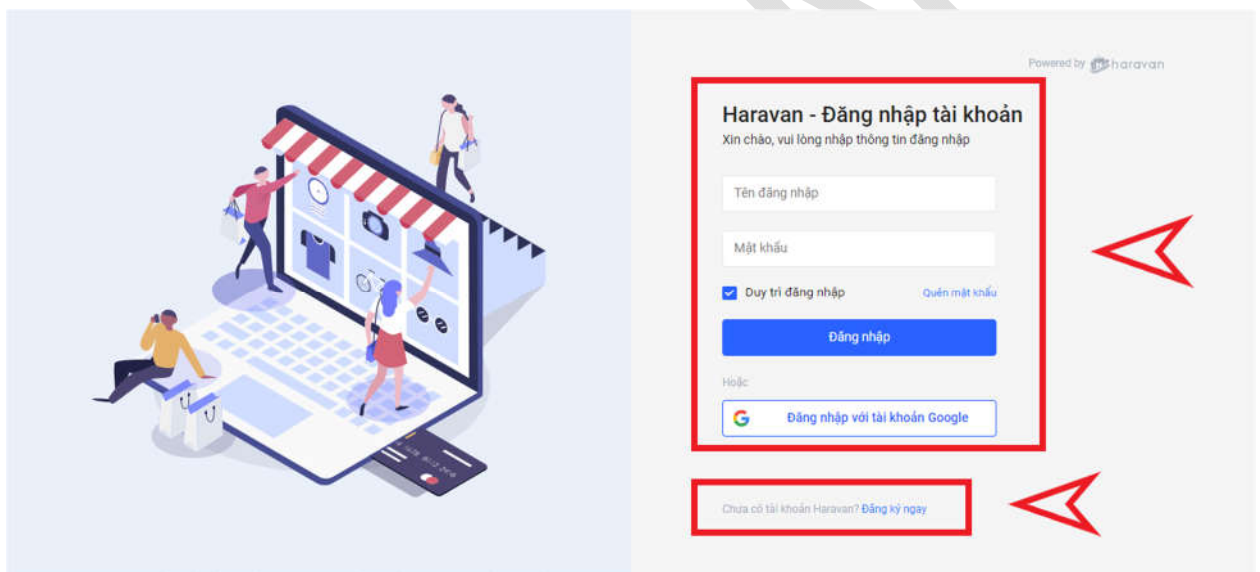
| | |
|---|----|
| Connect webhook..... | 3 |
| 1. Step 1: Create app | 3 |
| 2. Step 2: Use ngrok to register webhook | 5 |
| 3. Step 3: Register webhook | 6 |
| 4. Step 4: Call api subscribe webhook | 9 |
| 4.1. Request..... | 10 |
| 4.2. Response | 10 |
| 4.3. Use Postman..... | 11 |
| 5. Step 5: Receive notifications from webhook | 11 |
| 6. Step 6: Authenticate webhook | 13 |
| 7. Note:..... | 16 |

Connect webhook

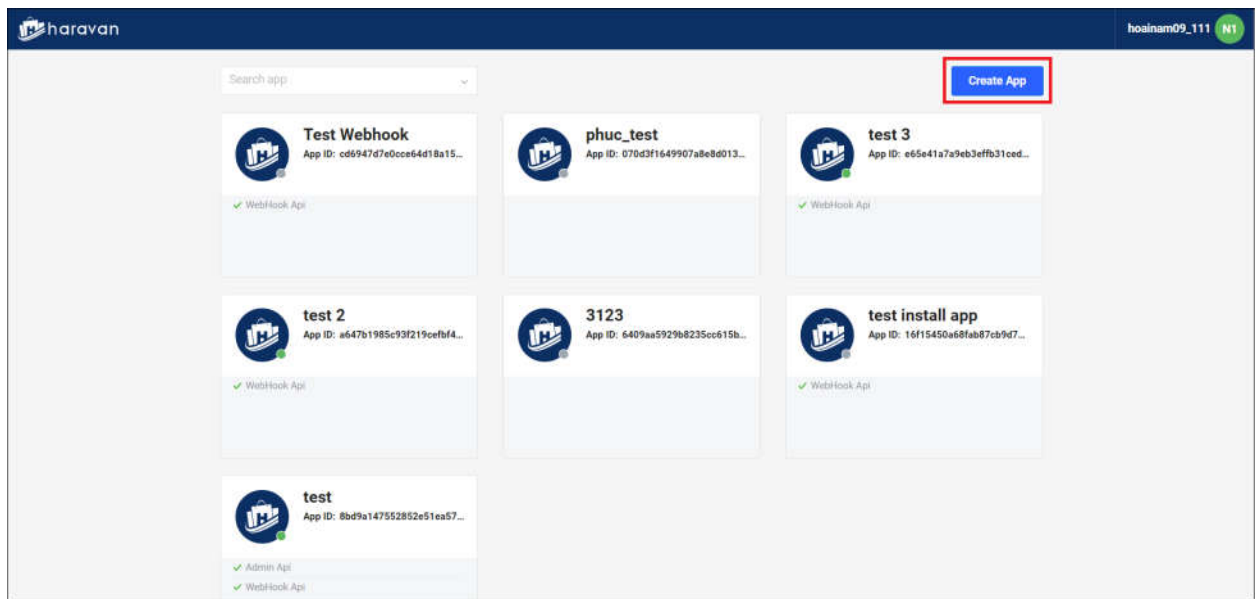
- You can register webhook to receive notifications about particular events in a shop.
- **Note :**
 - + Notification contains a JSON payload.
 - + Only support HTTPS.

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.

haravan

hoalnam09_111

Settings

Webhooks

Haraweb

Commerce

Private Token

Test Webhook

App ID: cd6947d7e0cce64d18a1531eef1c2db5

App Secret: 02dd4ab509173985fb339f6375b53ae0ae62956ff4a80f2c33e

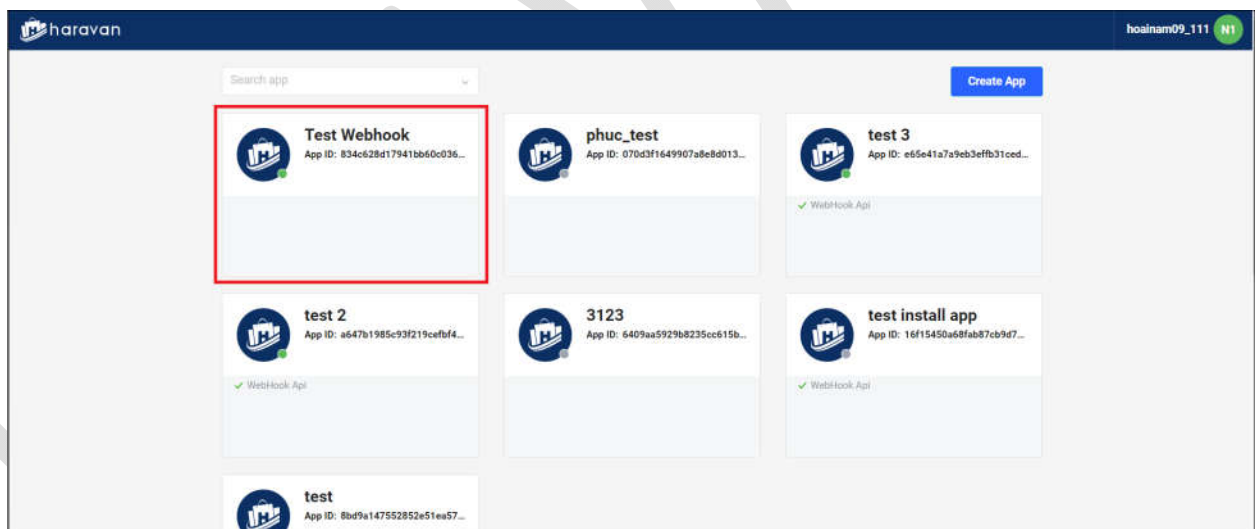
Name: Test Webhook

Description:

Redirect Url: http://localhost:3000/promotionbar/install/grandservice, http://localhost:3000/promotionbar/install/login

Icon App(160 x 160), Banner Login(600 x 600), Icon Login(200 x 60), Banner App(465 x 150)

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



2. Step 2: Use ngrok to test 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:**
 - + There is no need for ngrok if you have an online domain.
 - + 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)

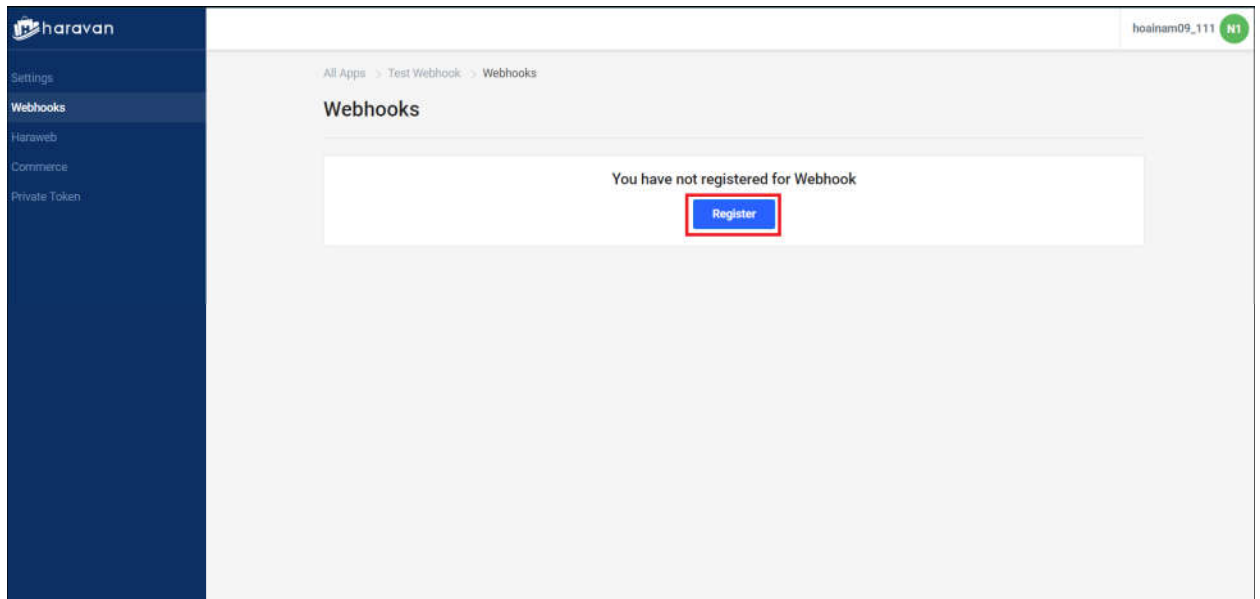
Session Status      online
Session Expires     6 hours, 20 minutes
Update              update available (version 2.3.35, Ctrl-U to update)
Version              2.3.27
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://e01249e5.ngrok.io -> http://localhost:3000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

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

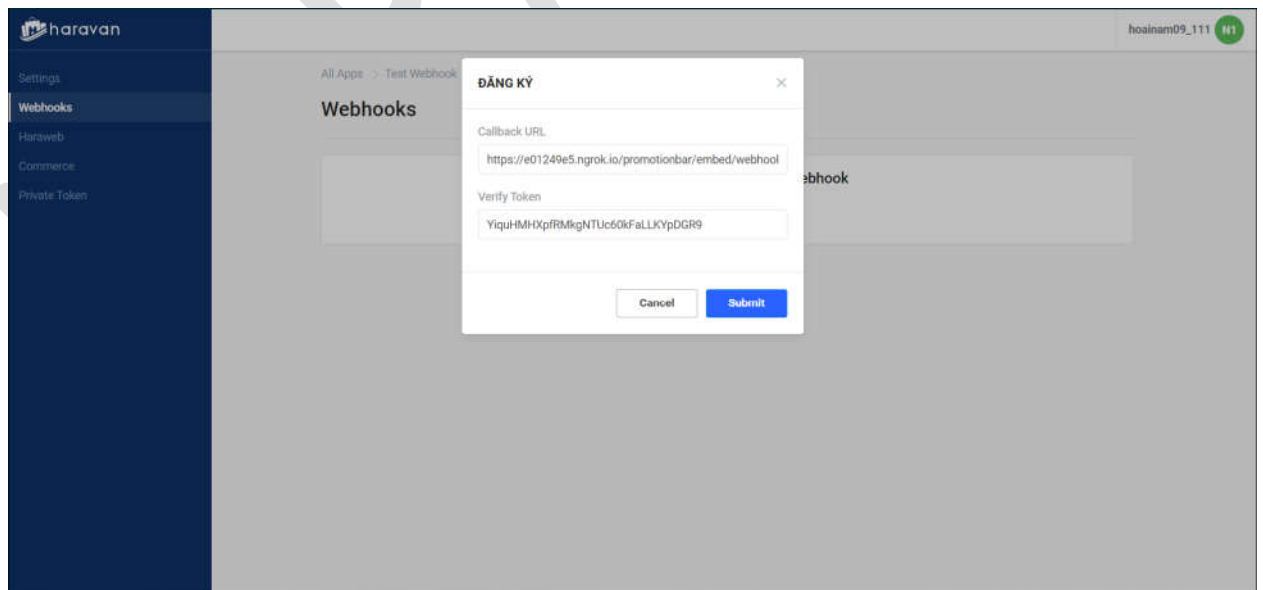
HTTP Requests
-----
GET /promotionbar/embed/webhooks 200 OK
GET /promotionbar/embed/webhooks 200 OK
```

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"  
}
```

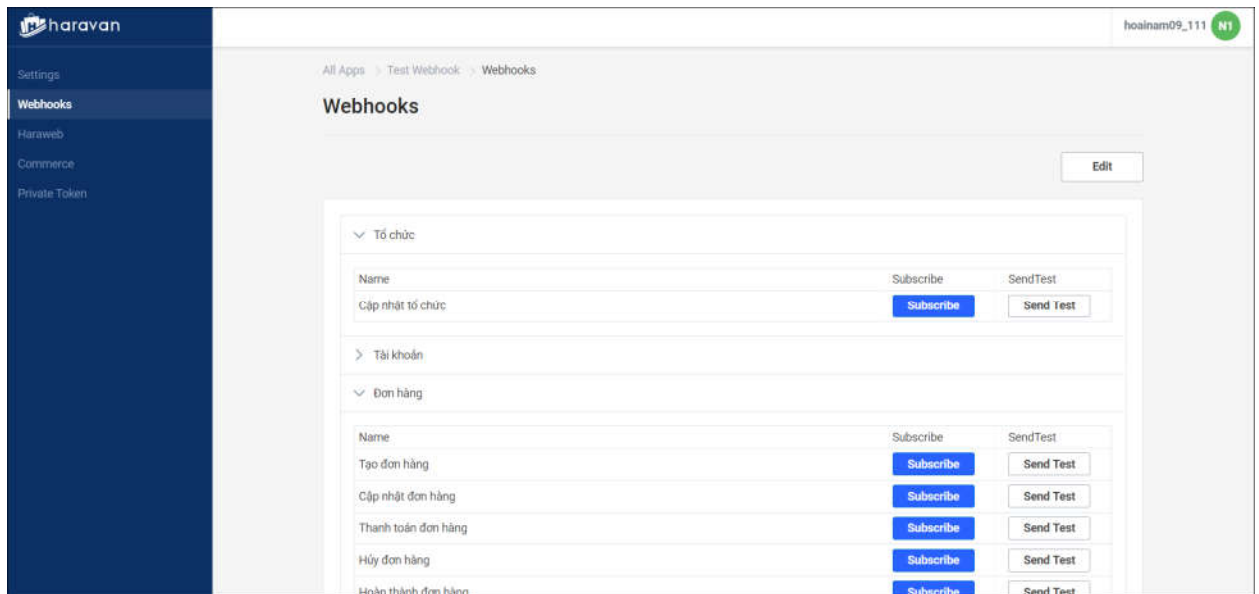
- You need to compare hub.verify_token (in object) with verify_token (in your app).
 - + If not match, response status 401.
 - + If match, response status 200 and value of hub.challenge.

The screenshot displays a web browser interface showing a successful GET request to the endpoint `/phoneconvert/webhooks`. The status is `200 OK` and the duration is `2.9ms`. The response body is a JSON object containing the following fields:

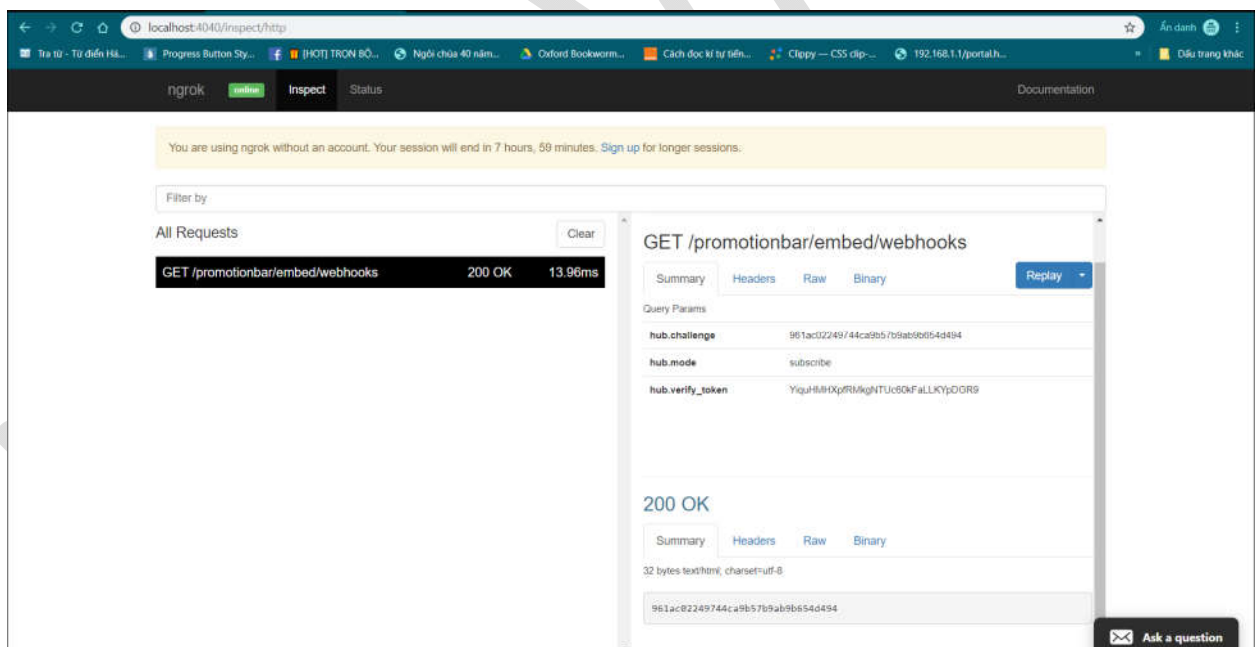
- `hub.challenge`: `dd279ae60d9645e19c06a34ad24db650`
- `hub.mode`: `subscribe`
- `hub.verify_token`: `YiquHMHXpfRMkgNTUc60kFaLLKYpDGR9`

Annotations in the image highlight the `200 OK` status and the `hub.challenge` value in the response body.

- 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/tutorial-use-scope>
 - Access_token:
 - <https://docs.haravan.com/blogs/omni/create-app-and-connect-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:

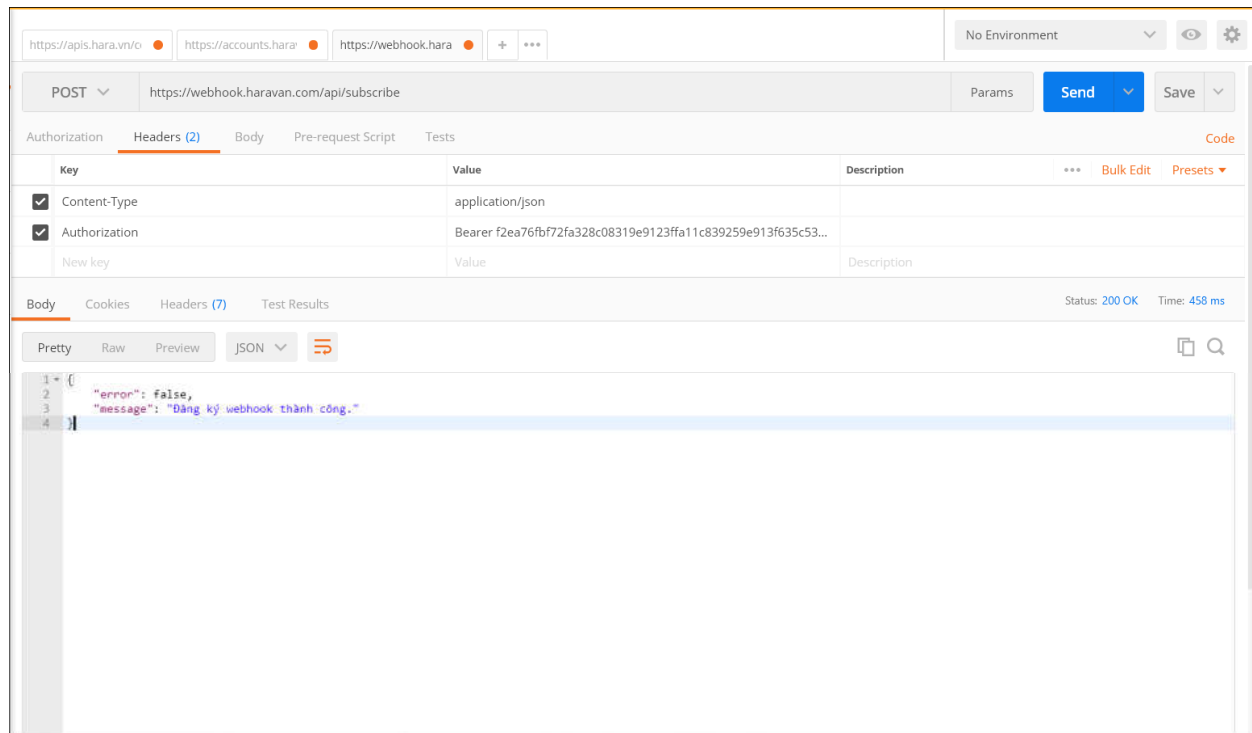
Content-Type: application/json

Authorization: Bearer + access_token

4.2.Response

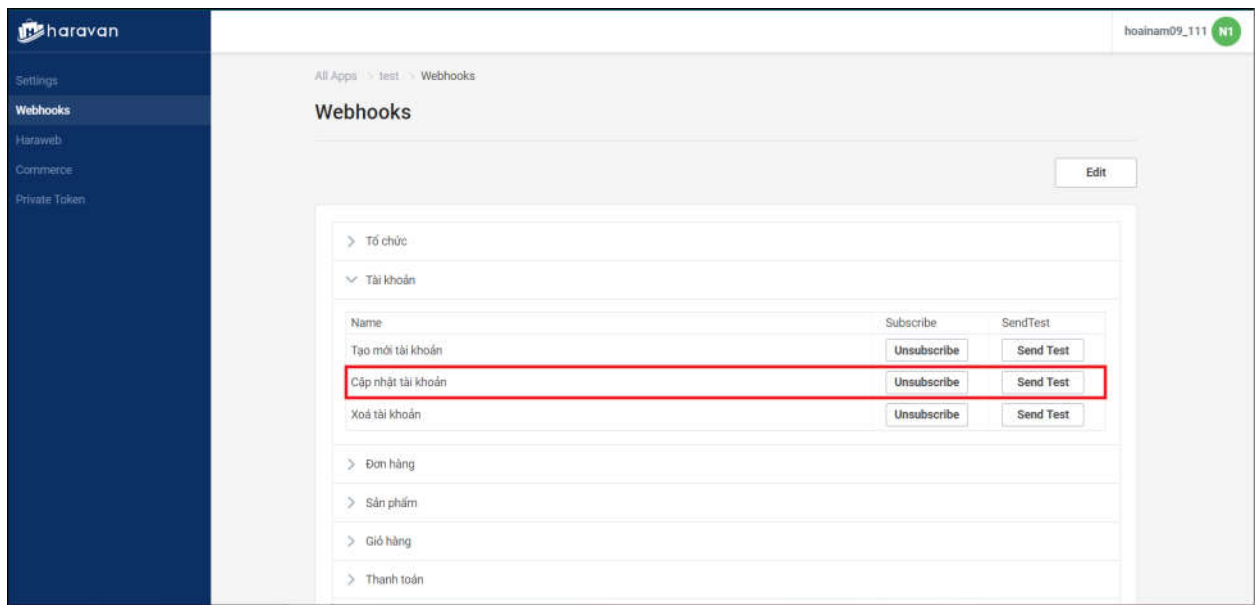
| Status | Response |
|--------|---|
| 200 | { "error": false, "message": "Đăng ký webhook thành công." } |
| 422 | {"error": "Unprocessable Entity"} |
| 401 | Unauthorized |
| 429 | Too many requests |
| 500 | Something went wrong. Please try again later. |

4.3. Use Postman

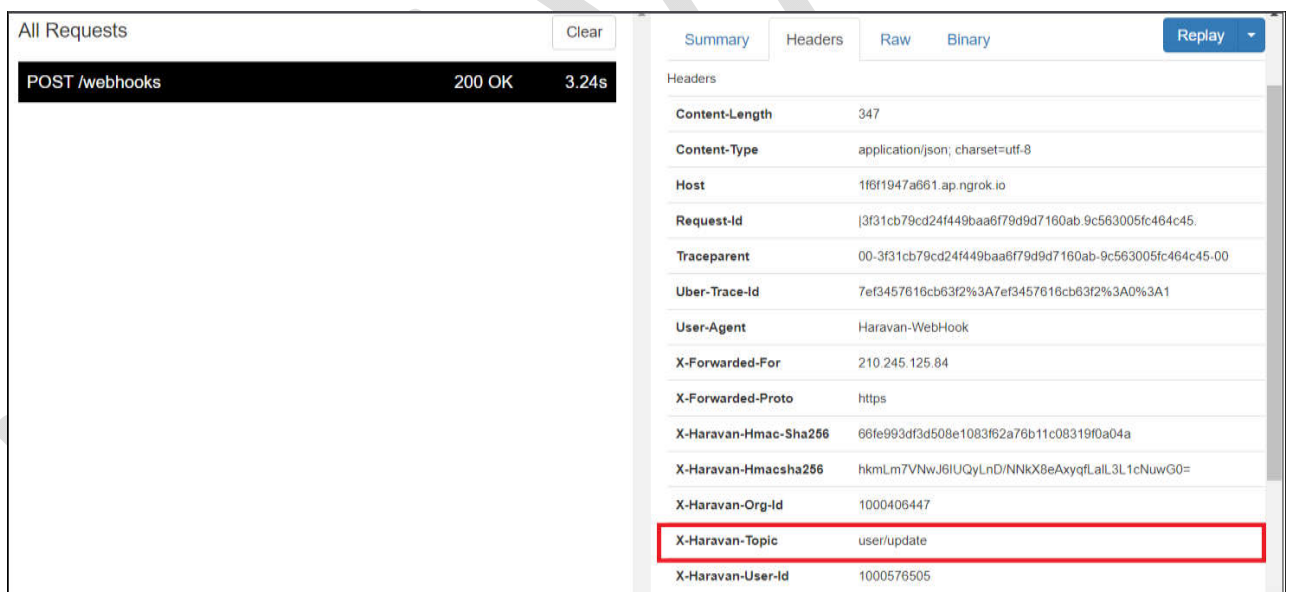


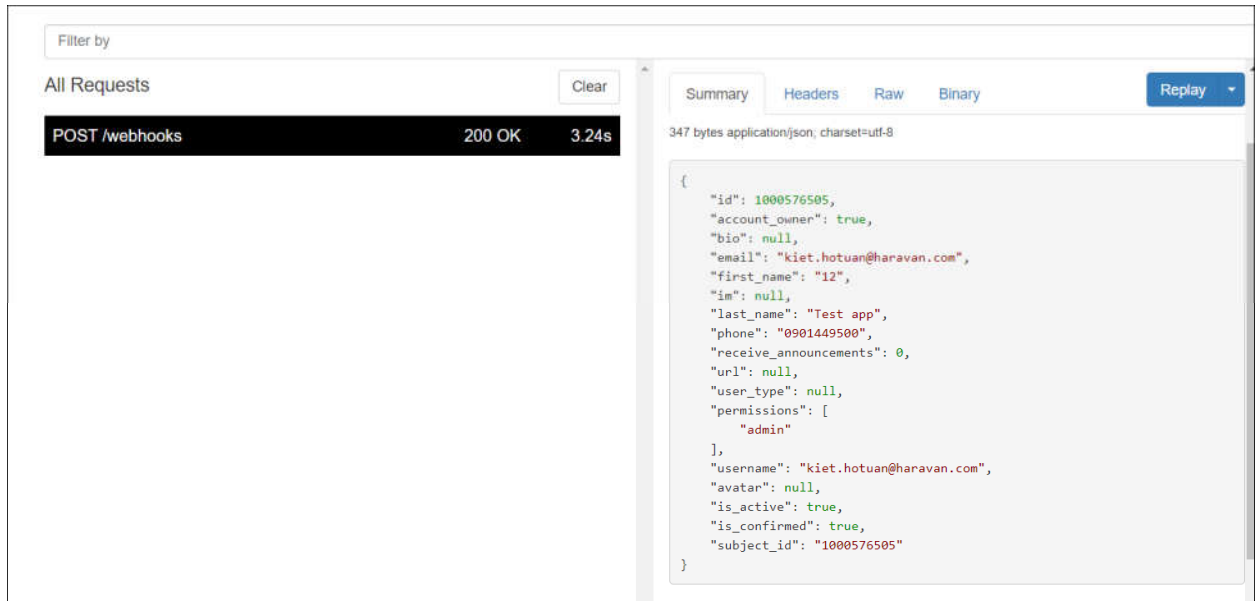
5. Step 5: Receive notifications from webhook

- 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



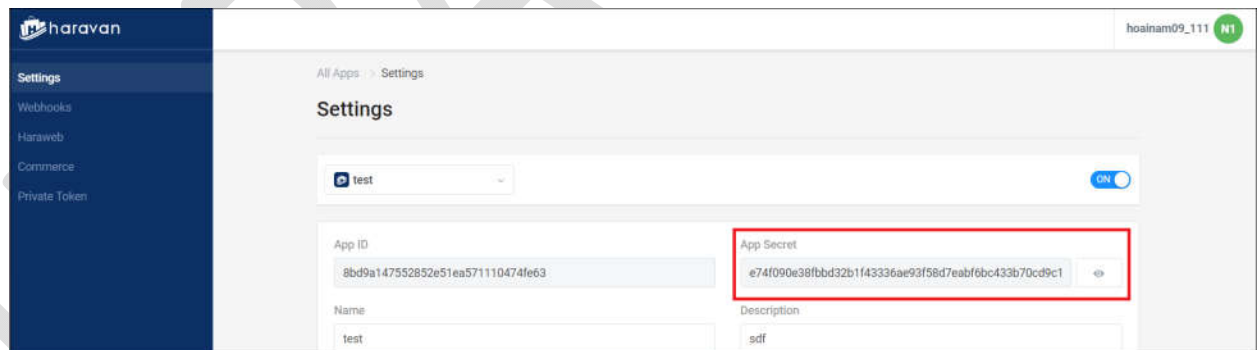
- When you update your account at <https://accounts.haravan.com/>, webhook will notify your application with the topic is "user / update" and data of your account via CallBack URL.





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-Hmacsha256**

Filter by

All Requests Clear

POST /webhooks 200 OK 3.03ms

Summary Headers Raw Binary Replay

Headers

| | |
|-----------------------|---|
| Content-Length | 346 |
| Content-Type | application/json; charset=utf-8 |
| Host | 66e2f562888c.ap.ngrok.io |
| Request-Id | j4da8406b49971548b4d9c58422b8b0a0_c19a84d97df5a14d |
| Traceparent | 00-4da8406b49971548b4d9c58422b8b0a0-c19a84d97df5a14d-00 |
| Uber-Trace-Id | 1f253ebf9ac97464%3A1f253ebf9ac97464%3A0%3A1 |
| User-Agent | Haravan-WebHook |
| X-Forwarded-For | 210.245.125.84 |
| X-Forwarded-Proto | https |
| X-Haravan-Hmac-Sha256 | 279bce8a0d3bb4186094435267458d148e92f295 |
| X-Haravan-Hmacsha256 | za7FoTVP8o3FWsPd1/DglM5YfwKgM0eWJZRT15NIZIU= |
| X-Haravan-Org-Id | 1000406447 |
| X-Haravan-Topic | user/update |
| X-Haravan-User-Id | 1000576505 |

+ To authenticate the webhook :

- First, we use `hash_mac` function with “**sha256**” algorithms to encode a string (data in webhook request + App Secret) to get the hash code.
- Second, use **base64** function to encode this hash.
- Finally, compare it with **X-Haravan-Hmacsha256** (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 = base64_encode(hash_hmac('sha256', $data, Haravan_APP_SECRET, true));
    return ($hmac_header == $calculated_hmac);
}

$hmac_header = $_SERVER['X-Haravan-Hmacsha256'];
$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. Step 7: Call api unsubscribe 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/tutorial-use-scope>
 - Access_token:
 - <https://docs.haravan.com/blogs/omni/create-app-and-connect-api>
 - https://www.youtube.com/watch?v=gzJ0_qbrvYM&feature=youtu.be
- Do this step after you have access_token.
- The purpose is to stop receiving webhooks notifications.

7.1.Request

| Method | URL |
|---------------|---|
| DELETE | https://webhook.haravan.com/api/subscribe |

Header:

Content-Type: application/json

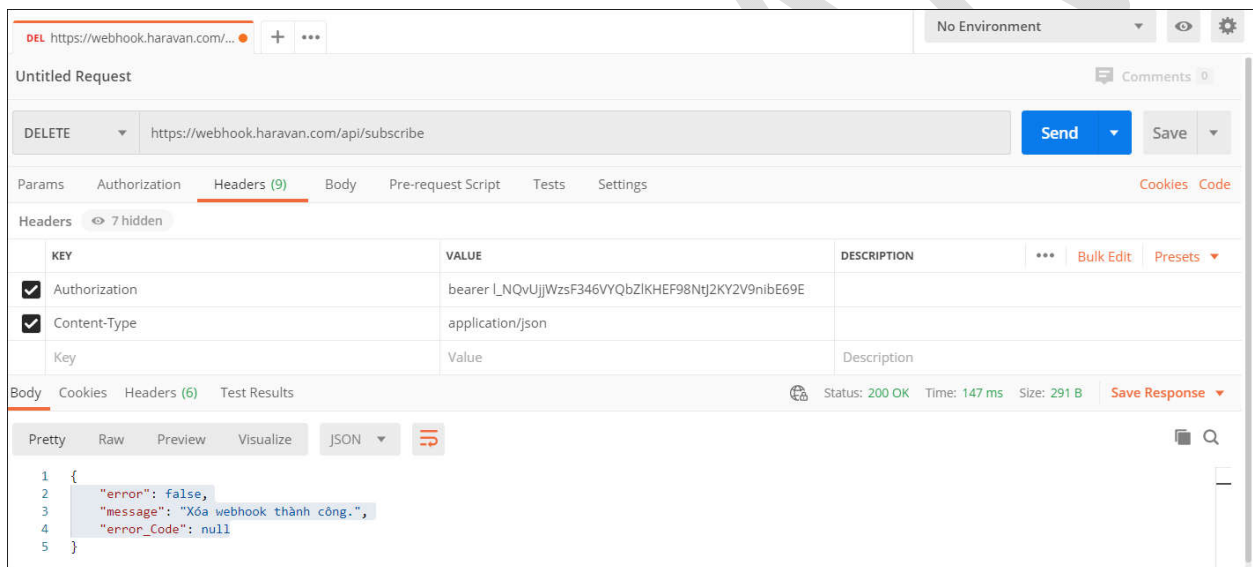
Authorization: Bearer + access_token

7.2.Response

| Status | Response |
|--------|--|
| 200 | { "error": false, "message": "Xóa webhook thành công.", "error_Code": null } |

| | |
|-----|---|
| | } |
| 422 | {"error": "Unprocessable Entity"} |
| 401 | Unauthorized |
| 429 | Too many requests |
| 500 | Something went wrong. Please try again later. |

7.3. Use Postman



8. 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.