TUTORIAL FOR USE SCOPE

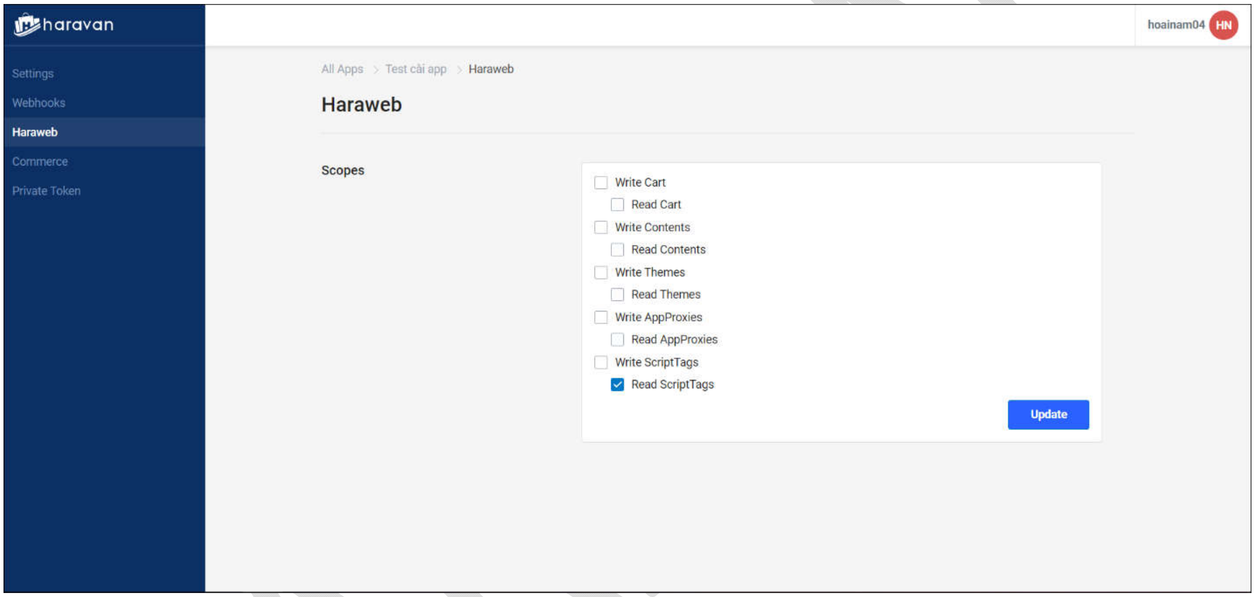
Version	Date	Author	Description
1.0	30-Oct-2019		
1.1	11-Feb-2020		**Edit SCOPE COMMERCE** **+ com.write_inventorie => com.write_inventories** **+ com.read_inventorie => com.read_inventories**

Table of contents

1 SCOPE HARAWEB

1.1 Scope

- These are scopes in “Haraweb” of https://developers.haravan.com
- Once selected here, you must pass the corresponding scopes when you install the application.
- For the API use at the sales page.
- The way to declare haraweb scope: **web.{scope_name}.**
  Ex: web.read_script_tags.



Haraweb

Scopes

☐ Write Cart
☐ Read Cart
☐ Write Contents
☐ Read Contents
☐ Write Themes
☐ Read Themes
☐ Write AppProxies
☐ Read AppProxies
☐ Write ScriptTags
☑ Read ScriptTags

1.2 API List for haraweb scope

- Corresponding to the scope will be the access to the corresponding api.
- The write scope will include read permissions and use the methods: GET, POST, PUT, DELETE.
- The Read scope only uses the GET method.
- API prefix: https://apis.haravan.com/web
- Call the api with the syntax: **https://apis.haravan.com/web/{api}.**
  EX: Call API get scriptTags : https://apis. haravan.com /web/script_tags.json (GET).

	Name	Scope	API
1	Write Contents	web.write_contents	**blog** /blogs/{blogId}/articles.json /blogs/{blogId}/articles/{article_id}.json /blogs.json /blogs/{blog_id}.json /blogs/{blog_id}/metafields.json /blogs/{blog_id}/metafields/{metafield_id}.json /blogs/{blogId}/articles/{article_id}/metafields.json
			comments /comments.json /comments/{comment_id}.json /comments/{comment_id}/spam.json /comments/{comment_id}/not_spam.json /comments/{comment_id}/approve.json /comments/{comment_id}/restore.json /comments/{comment_id}/remove.json
			pages /pages/{page_id}/metafields.json /pages/{page_id}/metafields/{metafield_id}.json /pages.json /pages/{page_id}.json
			redirects /redirects/{urlredirect_id}.json /redirects.json
2	Read Contents	web.read_contents	**blogs** /blogs.json /blogs/count.json /blogs/{blog_id}.json /blogs/{blog_id}/articles.json /blogs/{blog_id}/articles/{article_id}.json /blogs/{blog_id}/articles/count.json /blogs/{blog_id}/articles/tags.json /blogs/{blog_id}/metafields.json /blogs/{blog_id}/metafields/count.json /blogs/{blog_id}/metafields/{metafield_id}.json /blogs/{blog_id}/articles/{article_id}/metafields.json /blogs/{blog_id}/articles/{article_id}/metafields/count.json

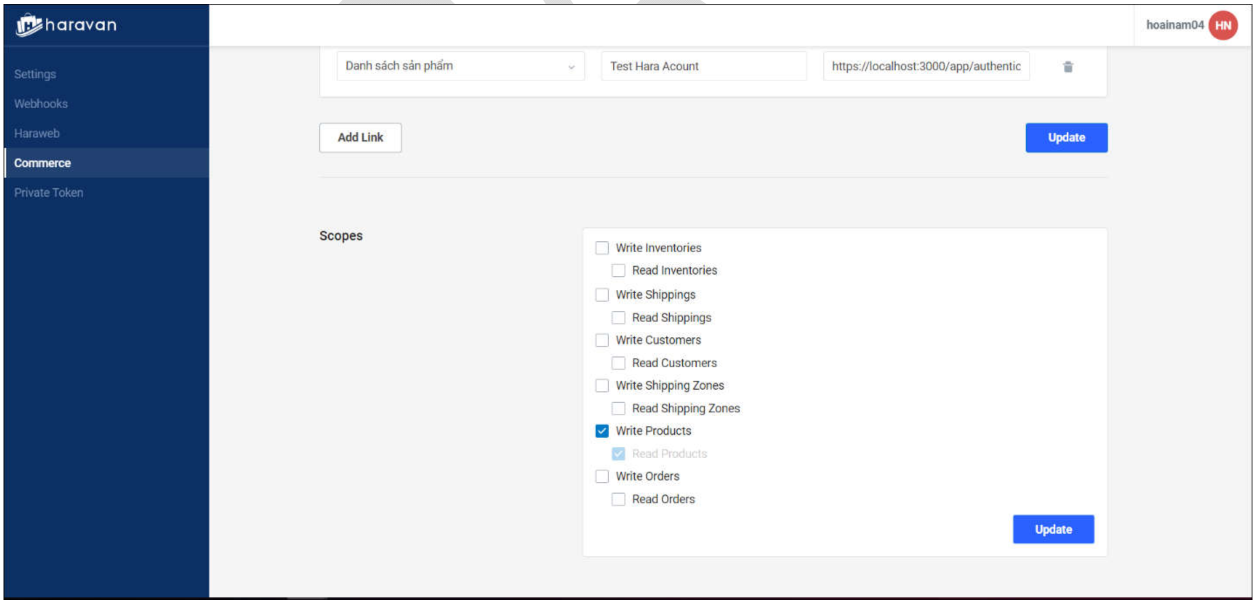
			/blogs/{blog_id}/articles/{article_id}/metafields/{metafieldId}.json
			comment /comments.json /comments/count.json /comments/{comment_id}.json
			pages /pages.json /pages/count.json /pages/{page_id}.json /pages/{page_id}/metafields.json /pages/{page_id}/metafields/count.json /pages/{page_id}/metafields/{metafieldId}.json
			redirect /redirects.json /redirects/count.json /redirects/{urlredirect_id}.json
			articles /articles/authors.json /articles/tags.json
3	Write Themes	web.write_themes	/themes.json /themes/{theme_id}.json /themes/{theme_id}/assets.json
4	Read Themes	web.read_themes	
5	Write AppProxies	web.write_app_proxies	/apps/{appId}/app_proxy.json

6	Read AppProxies	web.read_app_proxies	
7	Write ScriptTags	web.write_script_tags	/script_tags.json /script_tags/{id}.json /script_tags/count.json
8	Read ScriptTags	web.read_script_tags	/script_tags/{id}.json /script_tags.json

2 SCOPE COMMERCE

2.1 Scope

- These are scopes in “Commerce” of https://developers.haravan.com
- Once selected here, you must pass the corresponding scopes when you install the application.
- For the API use at the admin page.
- The way to declare commerce scope: **com.{scope_name}.**
  Ex: com.write_products.



Danh sách sản phẩm | Test Hara Acount | https://localhost:3000/app/authentic

Scopes

☐ Write Inventories
☐ Read Inventories
☐ Write Shippings
☐ Read Shippings
☐ Write Customers
☐ Read Customers
☐ Write Shipping Zones
☐ Read Shipping Zones
☑ Write Products
☑ Read Products
☐ Write Orders
☐ Read Orders

2.2 API List for commerce scope

- Corresponding to the scope will be the access to the corresponding api.
- The write scope will include read permissions and use the methods: GET, POST, PUT, DELETE.

- The Read scope only uses the GET method
- API prefix: https://apis.haravan.com/com
- Call the api with the syntax: **https://apis.haravan.com/com/{api}.**
  EX: Call API products : https://apis. haravan.com /com/products.json (GET).

	Name	**Scope**	**API**
1	Write Inventories	com.write_inventories	/inventories/transfer.json /inventories/transfer/{transferId}/receive.json /inventories/adjustorset.json
2	Read Inventories	com.read_inventories	/inventories.json /inventories/adjustments/{adjustmentId}.json /inventories/adjustments.json /inventories/adjustments/count.json /inventories/transfers.json /inventories/transfers/count.json /inventories/purchase_orders/{purchaseId}.json /inventories/purchase_orders.json /inventorytransaction/count.json /inventorytransaction/detail/{id}.json /inventorylocationbalance/count.json /inventorylocationbalance/listids.json /inventorylocationbalance/detail/{id}.json /inventorytransfer/count.json /inventorytransfer/listids.json /inventorytransfer/detail/{id}.json /inventorytransaction/listids.json
5	Write Shippings	com.write _shippings	/carrier_services.json /carrier_services/{carrierId}.json
6	Read Shippings	com.read_shippings	
7	Write Customers	com.write_customers	/customers.json /customers/{customerId}/addresses/{addressId}.json /customers/{customerId}/addresses.json /customers/{customerId}.json /customers/{customer_id}/timeline_comments.json /customers/{customerId}/addresses/set.json /customers/{customerId}/addresses/{addressId}/default.json

			/customers/{customer_id}/metafields.json /customers/{customer_id}/metafields/{metafield_id}.json
8	Read Customers	com.read_customers	/customers.json /customers/groups.json /customers/search.json /customers/{customerId}/addresses/{addressId}.json /customers/{customerId}/addresses.json /customers/{customerId}.json /customers/{customer_id}/timeline_comments.json /customers/{customer_id}/timeline_comments/count.json /customers/{customer_id}/metafields.json /customers/{customer_id}/metafields/count.json /customers/{customer_id}/metafields/{metafield_id}.json
9	Write Shippings Zones	com.write_shipping_zones	
10	Read Shippings Zones	com.read_shipping_zones	
11	Write Products	com.write_products	**products** /products.json /products/{product_id}.json /products/{productId}/variants.json /products/{productId}/variants/{variantId}.json /products/{productId}/images.json /products/{productId}/images/{imageId}.json /products/{ productId }/metafields.json /products/{productId}/metafields/{metafield_id}.json
			smart_collections /smart_collections/{collection_id}.json /smart_collections/{collection_id}/metafields/{metafield_id}.json
			collects /collects.json /collects/{collect_id}.json

			custom_collections /custom_collections.json /custom_collections/{collection_id}/metafields/{metafield_id}.json
			collections /collections/{collectionId}/metafields/{metafieldId}.json
			variants /variants/{variantId}.json /variants/{variant_id}/metafields.json /variants/{variant_id}/metafields/{metafield_id}.json
			images /images/{image_id}/metafields.json /images/{image_id}/{metafield_id}.json /images/{image_id}/metafields/{metafield_id}.json

s	Read Products	com.read_products	**products** /products.json /products/count.json /products/{product_id}.json /products/types.json /products/vendors.json /products/images/feature.json /products/images/feature/count.json /products/{productId}/images/count.json /products/{productId}/images/{imageId}.json /products/{productId}/variants.json /products/{productId}/variants/count.json /products/{productId}/images.json /products/{ productId }/metafields.json /products/{ productId }/metafields/count.json /products/{ productId }/metafields/{metafield_id}.json
			smart_collections /smart_collections/{collection_id}.json /smart_collections/count.json /smart_collections.json /smart_collections/{collection_id}/metafields.json /smart_collections/{collection_id}/metafields/count.json /smart_collections/{collection_id}/metafields/{metafieldId}.json
			collects /collects.json /collects/count.json /collects/{collect_id}.json

			custom_collections /custom_collections.json /custom_collections/count.json /custom_collections/{collection_id}.json /custom_collections/{collection_id}/metafields.json /custom_collections/{collection_id}/metafields/count.json /custom_collections/{collection_id}/metafields/{metafield_id}.json
			variants /variants/{variantId}.json /variants/{variant_id}/metafields.json /variants/{variant_id}/metafields/count.json /variants/{variant_id}/metafields/{metafield_id}.json
			images /images/{image_id}/metafields.json /images/{image_id}/metafields/count.json /images/{image_id}/metafields/{metafield_id}.json
			collections /collections/{collection_id}/product_ids.json /collections/{collection_id}/metafields.json /collections/{collection_id}/metafields/count.json /collections/{collection_id}/metafields/{metafield_id}.json
13	Write Orders	com.write_orders	/orders.json /orders/{orderId}.json /orders/{orderId}/transactions.json /orders/validatecoupon.json /orders/{orderId}/confirm.json /orders/{orderId}/close.json /orders/{orderId}/open.json /orders/{orderId}/cancel.json /orders/{orderId}/fulfillments.json /orders/{order_id}/metafields.json /orders/{order_id}/metafields/{metafield_id}.json /carts/promotions/calculate.json /carts/calculate.json /couponcode/validate.json /promotions/calculate.json

14	Read Orders	com.read_orders	/orders.json /orders/{orderId}.json /orders/count.json /orders/{orderId}/transactions.json /orders/{orderId}/transactions/count.json /orders/{orderId}/transactions/{id}.json /orders/sum.json /orders/affiliate.json /orders/internal/count.json /orders/listids.json /orders/{orderId}/fulfillments/{fulfillmentId}/events.json /orders/{orderId}/fulfillments/{fulfillmentId}/events/count.json /orders/{orderId}/fulfillments.json /orders/{orderId}/fulfillments/count.json /orders/{orderId}/fulfillments/{fulfillmentId}.json /orders/{order_id}/metafields.json /orders/{order_id}/metafields/count.json /orders/{order_id}/metafields/{metafield_id}.json /setting_payments.json

3 SCOPE WEBHOOK

- This is a scope to using webhooks for the application. Only shop owner (role contains 'admin') can use it.
- When using webhooks, this scope is required.
- You need to register webhook on https://developers.haravan.com before using this scope.

	Scope	Description
1	wh_api	Scope use webhook

4 SCOPE LOGIN

- These are the required scopes to log in and get user information
- Also you can add more scope from haraweb and commerce

	Scope	Description

1	openid	
2	profile	
3	email	Get email of login user
4	org	Get org information (org_id , org_name)
5	userinfo	Get information of login user

5 SCOPE INSTALL

- These are the scopes used to install the application.
- These are the scopes include:
    + Required scopes.

	Scope	Description
1	openid	
2	profile	
3	email	Get email of login user
4	org	Get org information (org_id , org_name)
5	userinfo	Get information of login user
6	grant_service	This is the scope that only the shop owner (role contains 'admin') can use. function: + Get long-lived access_token + Install the application on the Seller application list

- + Scope use webhook (optional)
- + Scopes are selected at haraweb and commerce

6 HOW TO USE SCOPE WHEN INSTALLING THE APP

- When installing, you need to focus on the scope login and scope install.
- As you can see, the scope login and install are mostly the same and both are used to pass to the authorize url to get the code and id_token. So, depending on how to use the scope, you can install the app in two options.

- **Note :**
  - **Here only describes how the scope works**
  - **Refer to the link below for more information :**
    - https://docs.haravan.com/blogs/omni/huong-dan-tao-app-ket-noi-api

6.1 Option 1: Use scope login to install the app

6.1.1 How it works

- As mentioned, login and install are both call url authorize but different scope is passed (scope login or install).
- So, we can pass the selected scope at haraweb and commerce with the scope login right from the first call to the authorize url.
- You still have the code corresponding to the scope passed, using the oauth 2 library to render access_token.

6.1.2 Features

- Just call the authorization url once.
- This access_token is called access_token user, and it's short-lived access_token
- The application can only be used by users who install it.
- Does not appear on the seller app list.
- Unable to use webhook.

6.2 Option 2: Use scope login and scope install to install the app (Recommended)

6.2.1 How it works

- First, call the authorization url with scope login to get id_token.
- Use **jwt** to decode this id_token to get an object including user information, role users, shop information.
- You need to verify the role of the logged in user :
  - If the user is the shop owner (role contains 'admin') then call url authorize with scope install (because **webhook** scope and **grant_service** scope are only used by the shop owner)
  - If the user isn't the shop owner (role doesn't contain 'admin') then show the error message.
- You have the code corresponding to the scope passed, using the oauth 2 library to render access_token.

6.2.2 Features

- Can verify user and shop information twice, increased security and ability to manage users.
- Access_token is a long-lived access_token.

- Install the application on the Seller application list.

7 HOW TO USE SCOPE LOGIN WHEN USING THE APP

- When the application was installed, we need to verify that the logged in user has access to the application
- There are 2 types of user authorization:
  - + User authorization on seller
  - + User authorization on Application (configured on the application)
- Before the user starts the application, call the authorization url with scope login to get id_token
- Use **jwt** to decode this id_token to get an object including user information, role users, shop information.
- You need to verify the role of the logged in user:
  - + If the user is the shop owner (role contains 'admin') then starts the application.
  - + If the user isn't the shop owner (role doesn't contain 'admin'), We have 3 cases:
    - ▪ **Case 1:** authorization on seller of the user's account do not accept to use the scope of the application, show messages "you are not authorized to use the application".

    - ▪ **Case 2:** That user has permission to use the application's scopes but the user is not authorized to use the app (if the application has its own authorization system), show messages "you are not authorized to use the application".

    - ▪ **Case 3:** That user does not have permission to use the application's scopes, but the user is allowed to use the application (if the application has its own authorization system), starts the application.