



WORDCAMP
MALAYSIA 2024

SPEAKER

Using AI to Help You Write an API for a WordPress Website

Victor K. Nip

22 - 23 November 2024 (Fri & Sat)
The Space, Menara Ken TTDI, Kuala Lumpur

#wcmv24





使用 AI 幫助您為 WordPress 網站編寫 API (中文加長版) Using AI to Help You Write an API for a WordPress Website (Chinese Extended Version)

Victor K. Nip

Hong Kong WordPress Meetup #82

2024-12-09

Victor K. Nip x WordPress

- Have used WordPress for development since 2016
- An organizer of WordCamp Hong Kong and Hong Kong WordPress Meetup
- First WordCamp: WordCamp Kuala Lumpur 2017
- Attended 10 in-person and 3 online WordCamps
- 4th time at a WordCamp in Malaysia





WORDCAMP
MALAYSIA 2024

Introducing **Our First Batch Of Speakers**



Julian Song

**Implementing Robust Security
Measures for WordPress Websites**



Low Chee Cheong

**3 New SEO Lessons That
Changed How I Think**



Victor K. Nip

**Using AI to Help You Write an
API for a WordPress Website**



Ganga Kafle

**Benefits of Contributing to
WordPress Open Source
(Five for the Future)**



Dan Gabriel A. Tabifranca

**Cooking Up WordPress:
A Chef's WordPress Journey**



Tarun Sikder

**Artificial Intelligence In Content
Creation: How AI Can Enhance,
Not Replace, Human Creativity!**

22 - 23 November 2024 (Fri & Sat)
The Space, Menara Ken TTDI, Kuala Lumpur

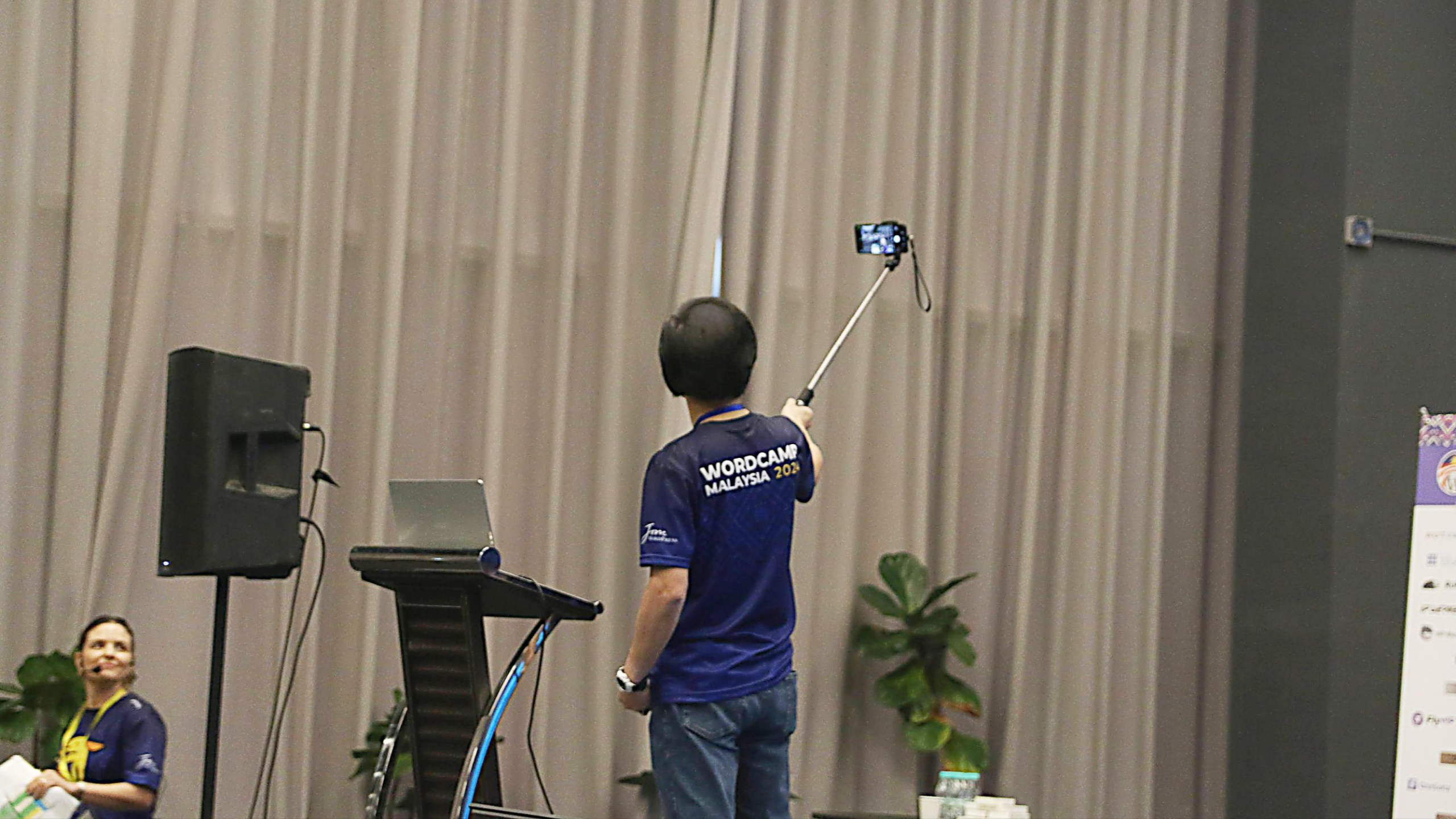
#wcmy24













Using AI to Help You Write an API for a WordPress Website

- We'll explore how AI can help a **novice** WordPress developer in writing an API for a WordPress website, using my website development journey as a case study.
- This will cover both calling API from the frontend using JavaScript and the coding of the API using PHP.

The Project

- Create a SaaS (software as a service) to do calculations for a new accounting requirement in Hong Kong.
- Timeframe: January 2024
- Relevant facts:
 - Users need to purchase tokens to use the service.
 - 1 token is spent for the calculation for 1 person.

Note: What I present here is a “re-enactment” of what I did in January.

What is an API?

- An application programming interface (API) is a connection between computers or between computer programs.
- It is not intended to be used directly by a person (the end user) other than a computer programmer who is incorporating it into software.
- The term API is often used to refer to web APIs, which allow communication between computers that are joined by the internet. This is the focus of the session.

What is an API?

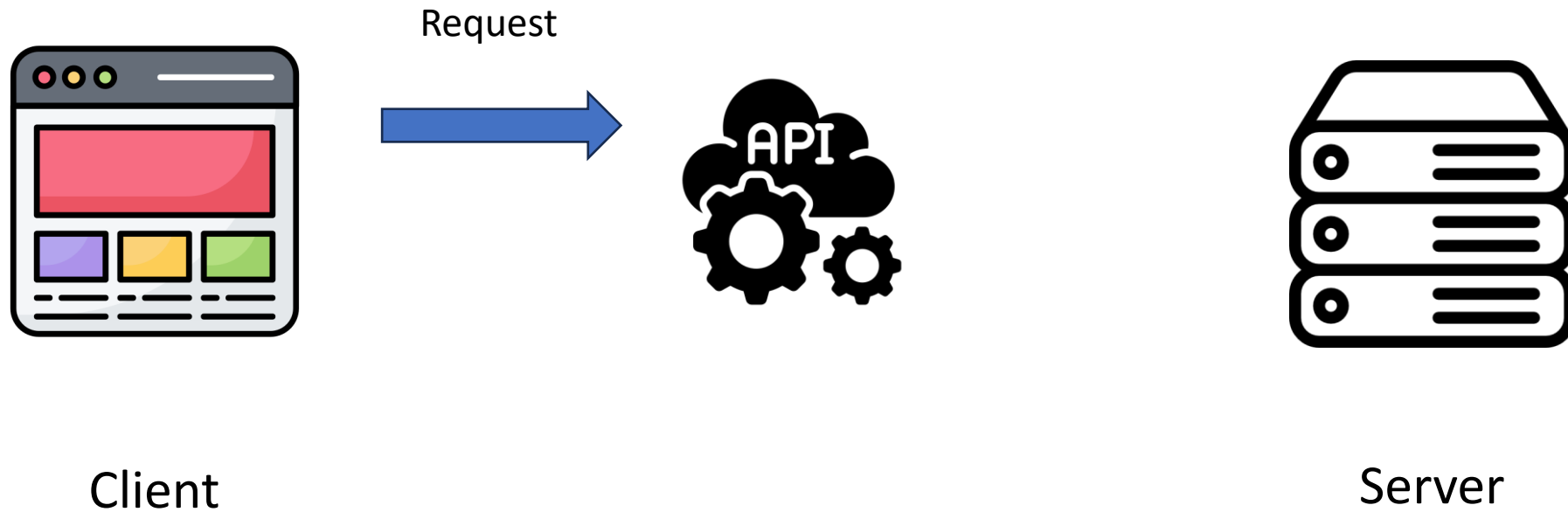


Client

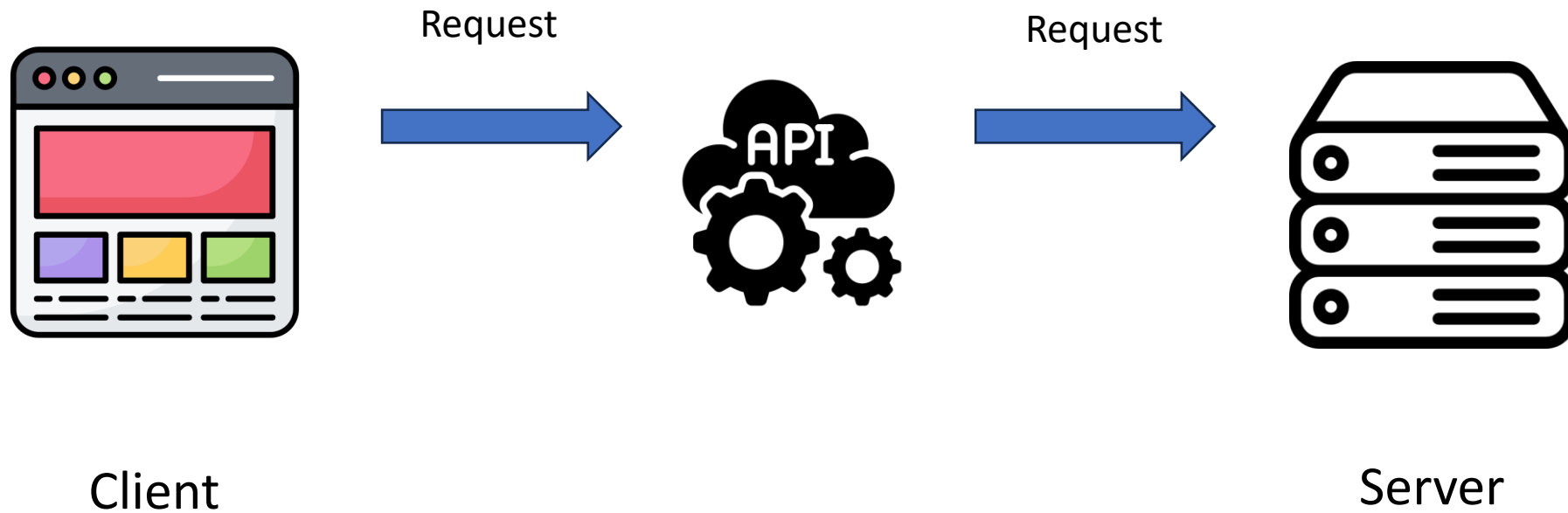


Server

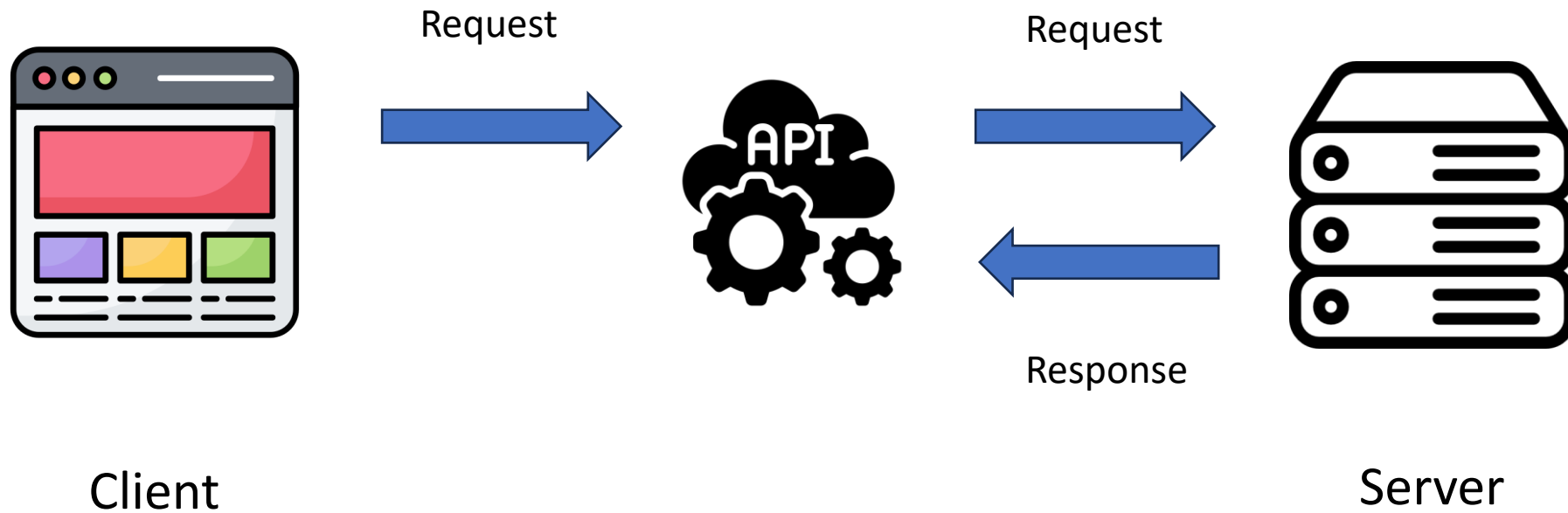
What is an API?



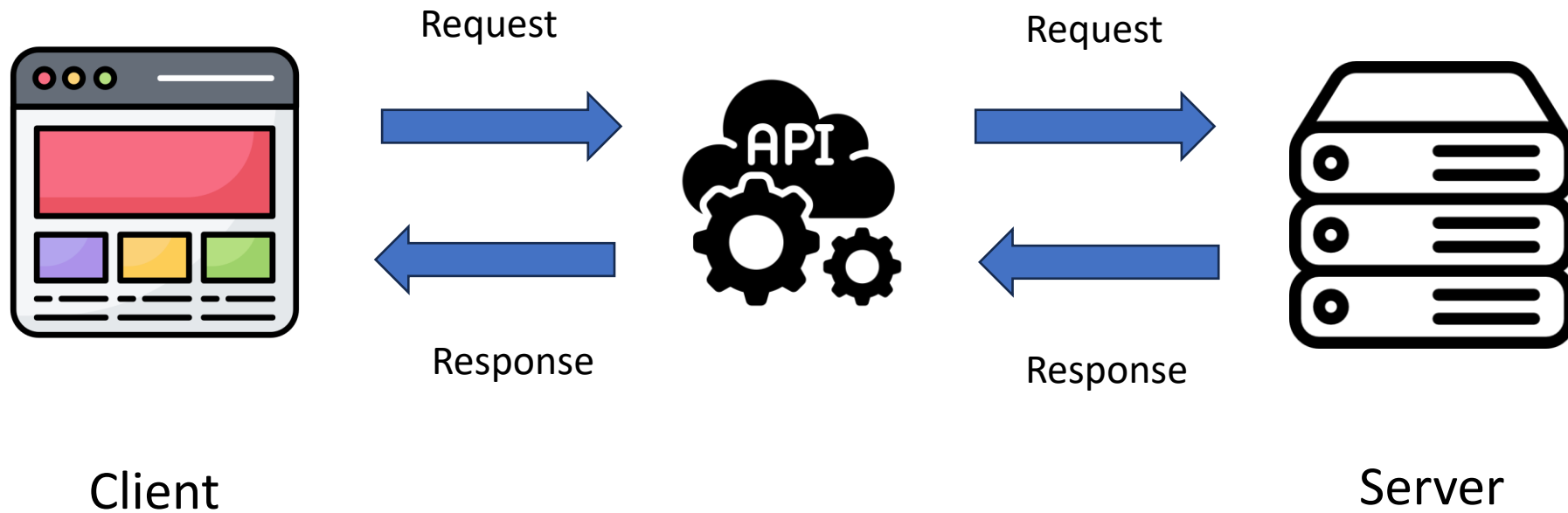
What is an API?



What is an API?



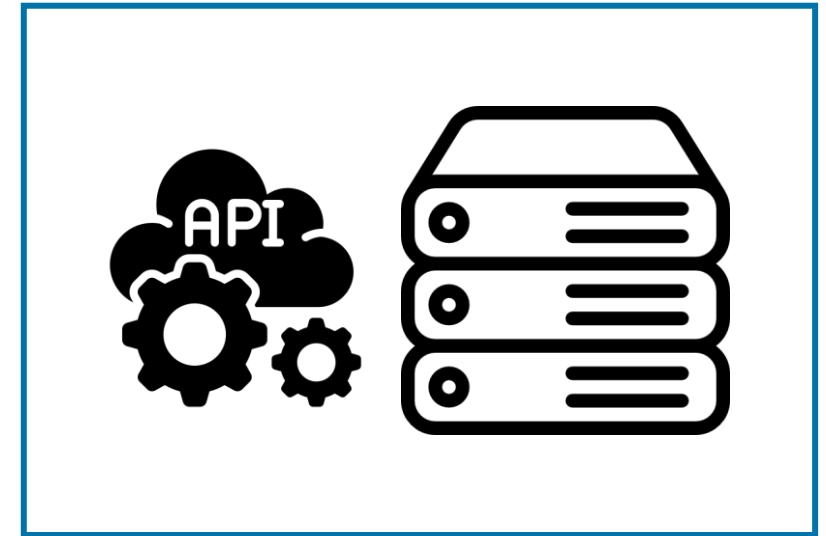
What is an API?



API in WordPress

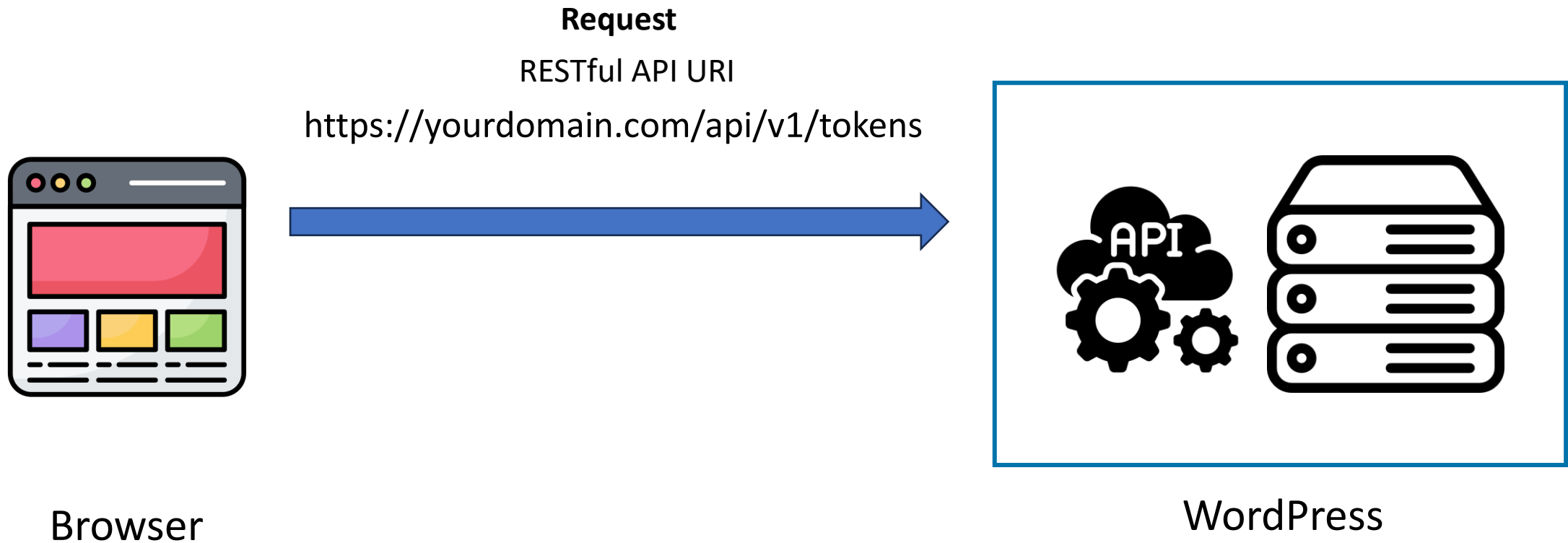


Browser

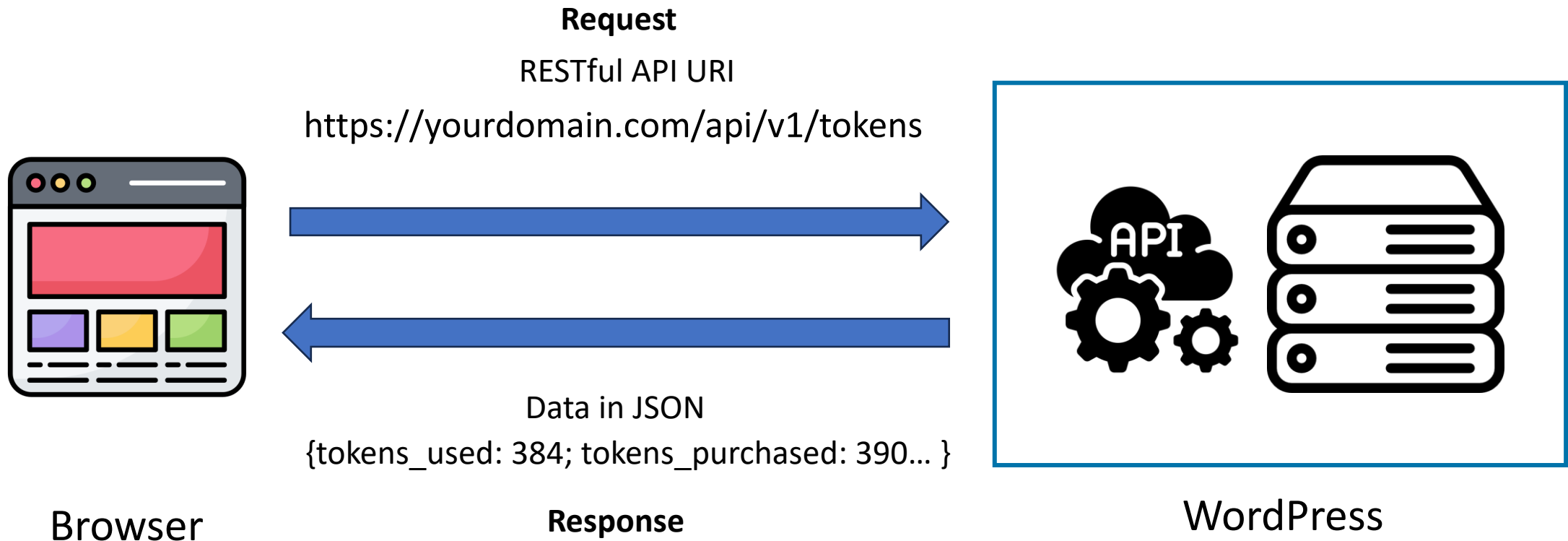


WordPress

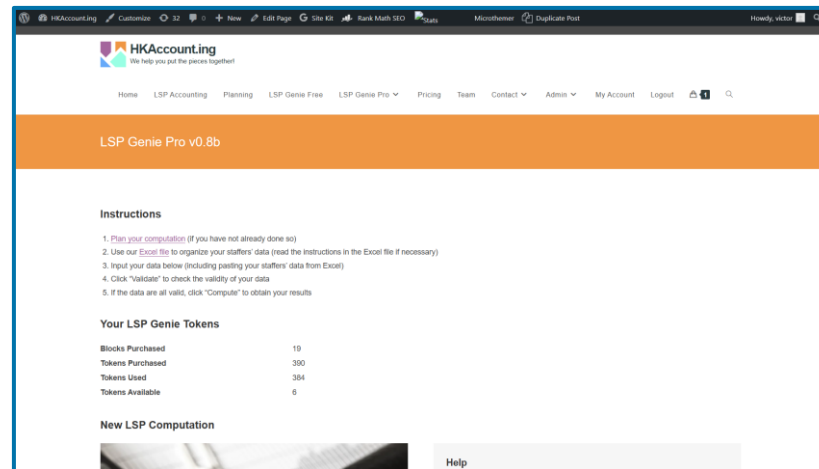
API in WordPress



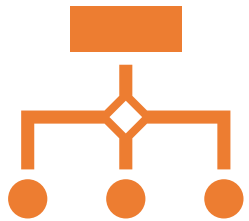
API in WordPress



The Website

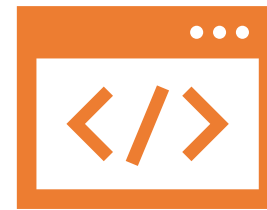


Why Use API for that?



Require up-to-second data

Data can change outside of the page and the data from the server are needed for checking



User experience

The data need to be updated without loading a new page (i.e., AJAX: Asynchronous JavaScript And XML)



How to Write an API?

1) Break down the program logic into manageable pieces

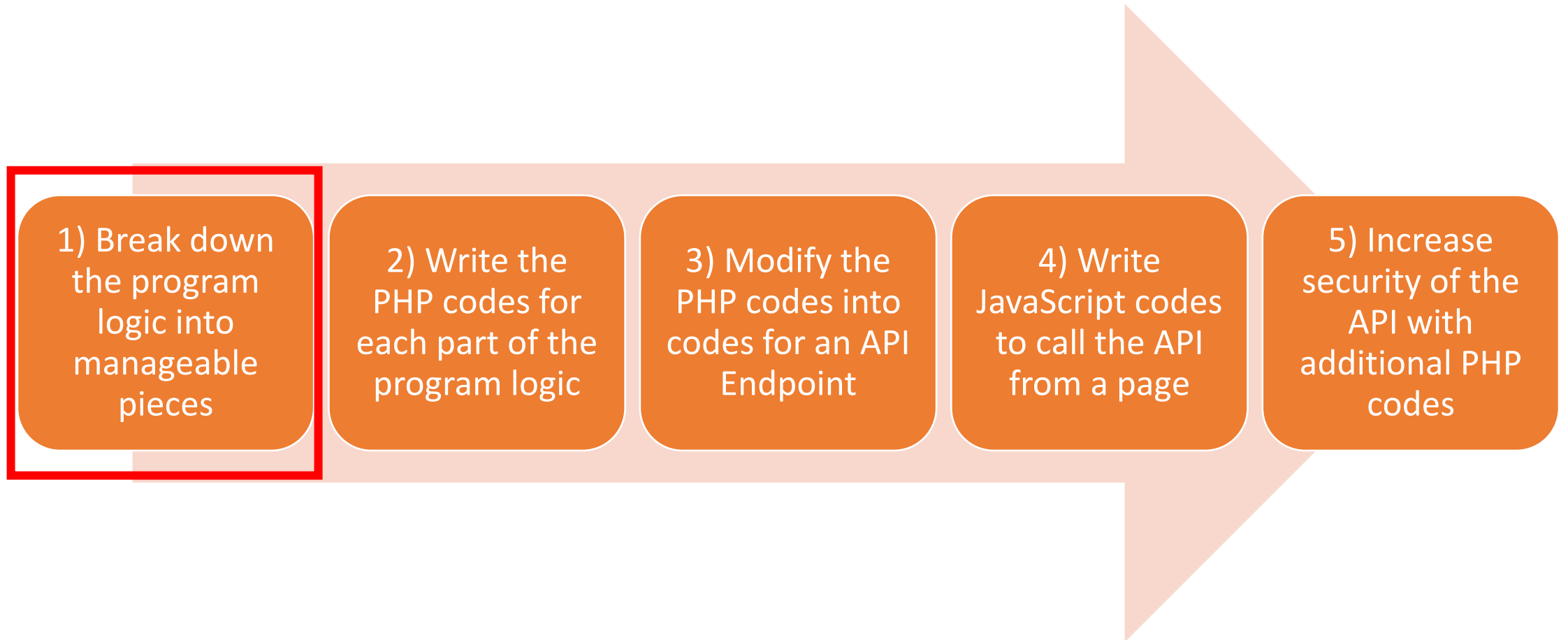
2) Write the PHP codes for each part of the program logic

3) Modify the PHP codes into codes for an API Endpoint

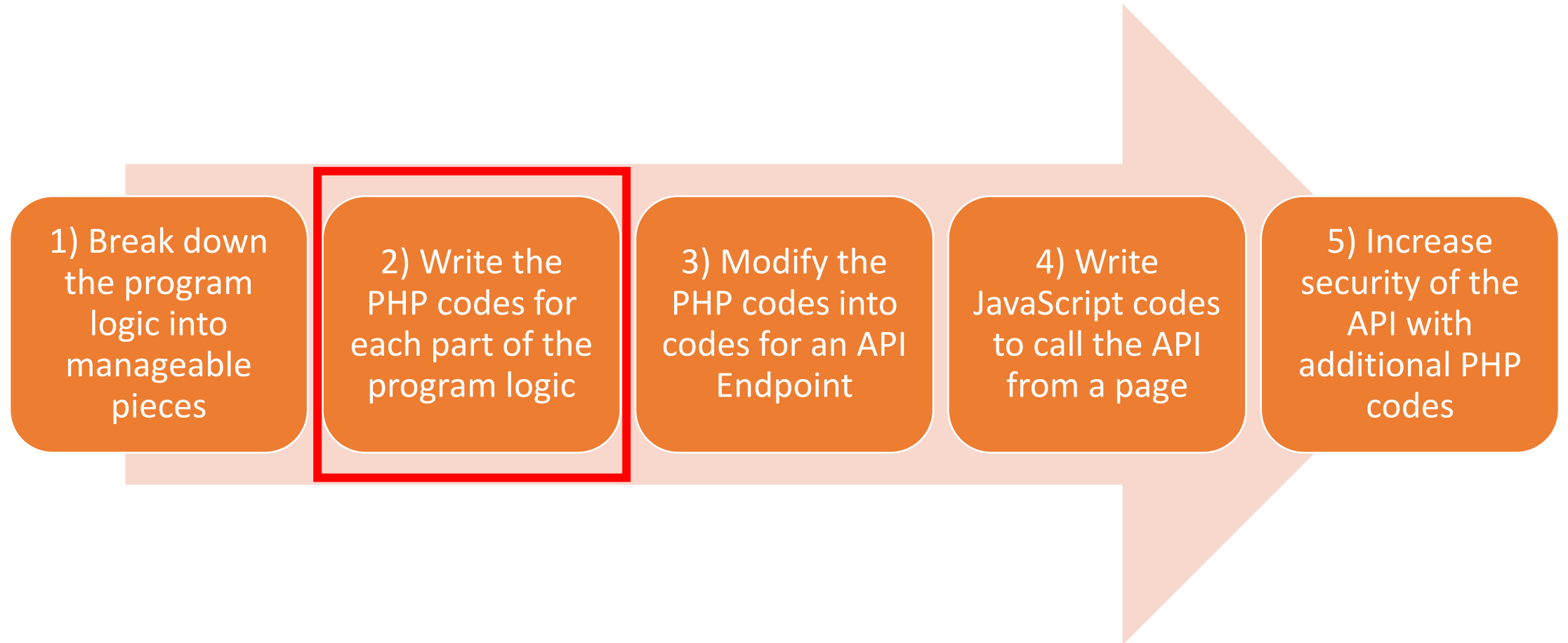
4) Write JavaScript codes to call the API from a page

5) Increase security of the API with additional PHP codes

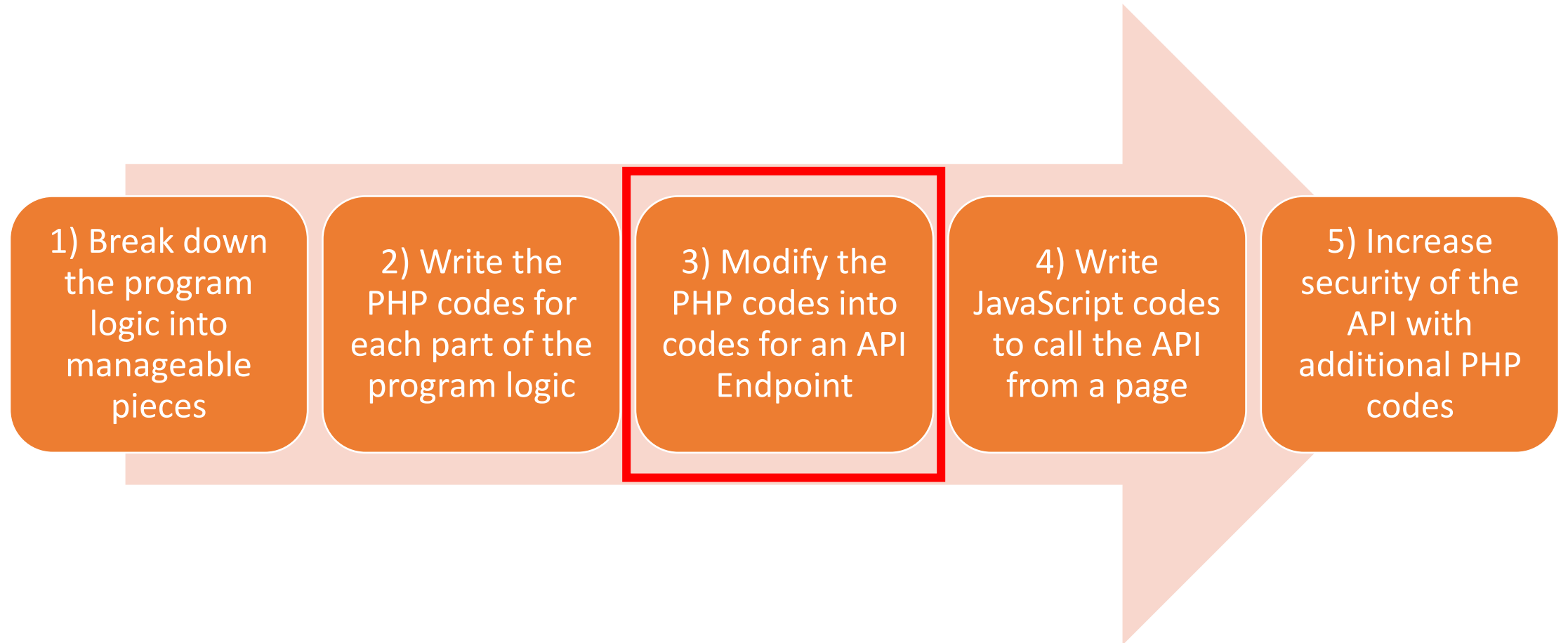
How to Write an API?



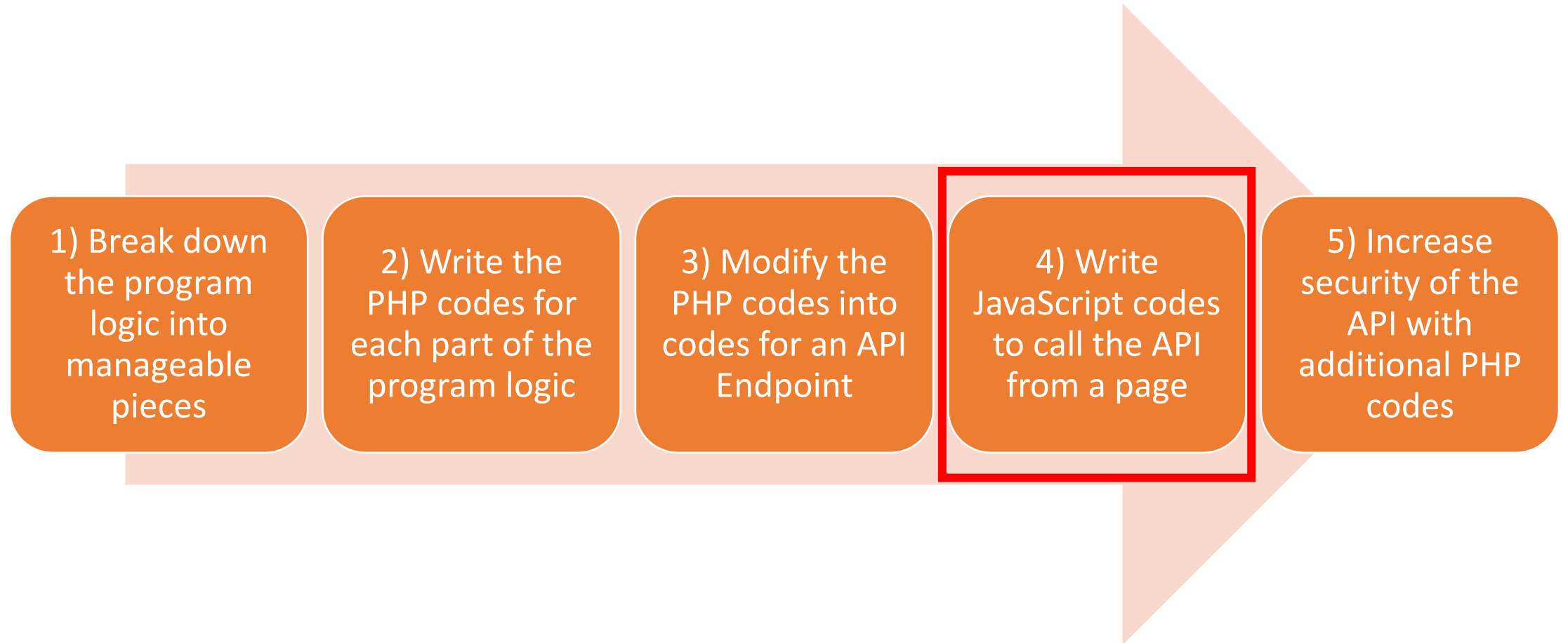
How to Write an API?



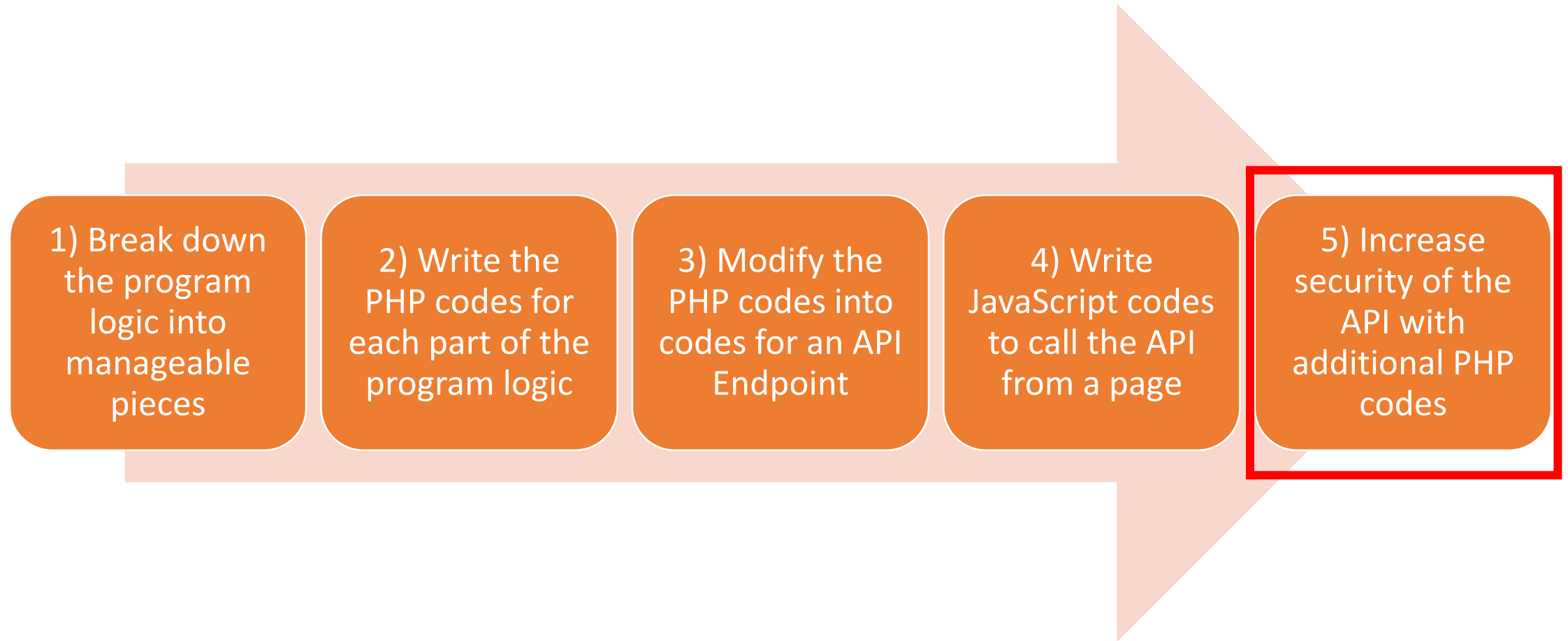
How to Write an API?



How to Write an API?



How to Write an API?





Let's relive my journey
together!

Your LSP Genie Tokens

Blocks Purchased	19
Tokens Purchased	390
Tokens Used	384
Tokens Available	6

Step 1: Break down the program logic into manageable pieces

The Function of the API Required

**I want to
know**

how many blocks of tokens have been
purchased by the user,

so that I know how many tokens have been
purchased by the user,

and then how many tokens have been used
by the user so far,

and with that I will know how many tokens
are left for the user to use.

The Function of the API Required

I want to
know

Step 2.1

how many blocks of tokens have been
purchased by the user,

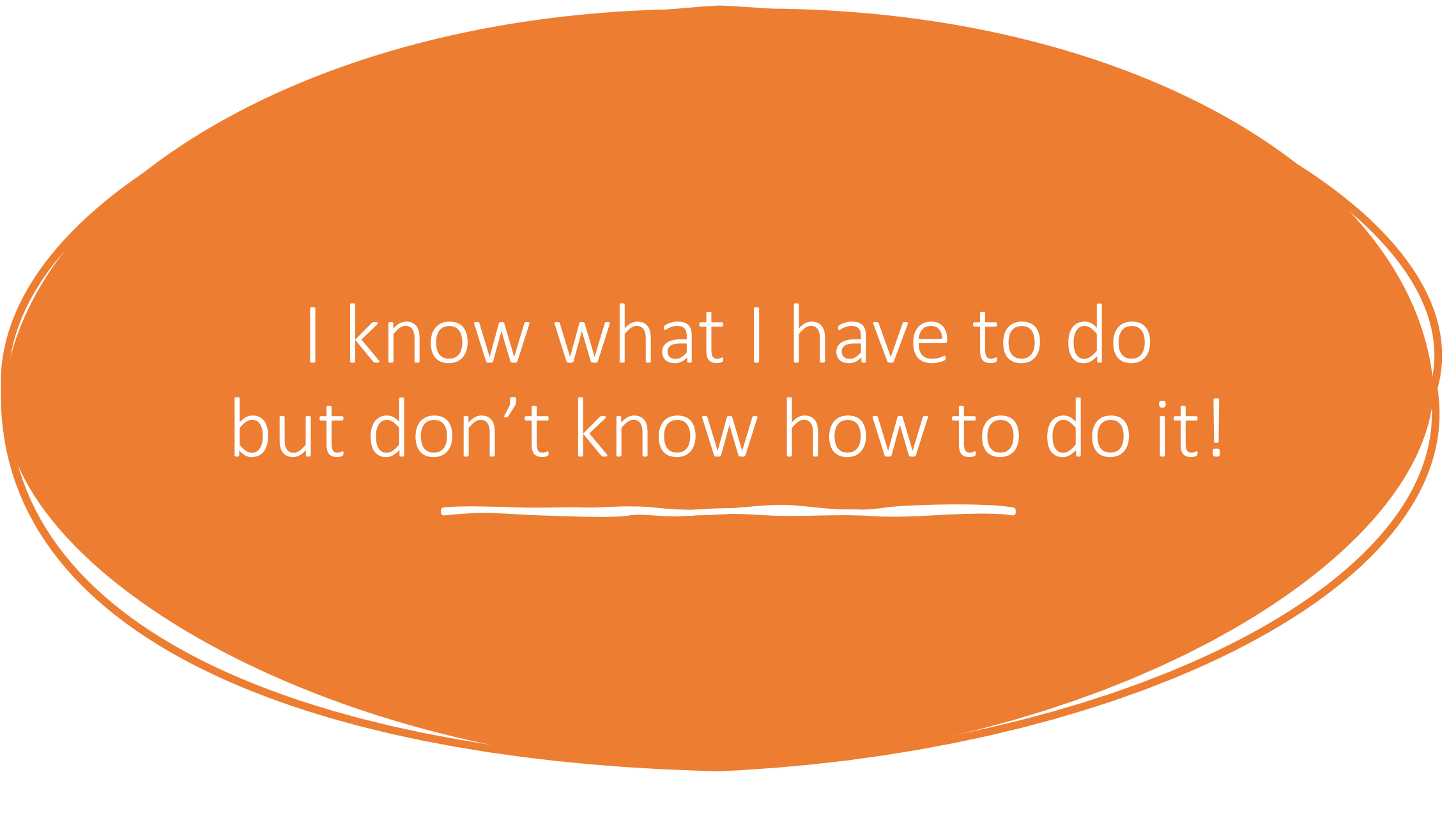
~~so that I know how many tokens have been
purchased by the user,~~

Step 2.2

and then how many tokens have been used
by the user so far,

~~and with that I will know how many tokens
are left for the user to use.~~

Step 2: Write the PHP codes for each part of the program logic

A large, solid orange oval with a thin white border, centered on a white background. Inside the oval, the text "I know what I have to do but don't know how to do it!" is written in white, sans-serif font. Below the text is a short, horizontal white line.

I know what I have to do
but don't know how to do it!

What I Would Have Done Before 2022

- Do Google searches (e.g., starting with “how to create an api in wp”)
- Issues:
 - Your problem may not be well-defined, and you may need many searches, using many different search terms, before getting the relevant answers
 - The articles you read may have the right answer but is written for a skill level above yours
 - Examples may only be remotely similar to what you want to do and you need to code a lot yourself
 - You may have to read a lot of irrelevant contents before getting the right answer
 - Sometimes helpless when the codes do not work

What I Do in 2024

- Use AI (Large Language Models: LLM), of course!
- Service: poe.com
 - Free tier
 - Available in Hong Kong
- Benefits of using AI
 - LLM's can understand contexts.
 - LLM's are often very good at writing JavaScript, PHP, HTML, CSS and can help you debug codes in these languages.
 - Therefore, LLM's can give you codes that are almost (if not totally) ready

Forget About User Bots

Explore - Poe

poe.com/search?q=wordpress

Incognito Relaunch to update

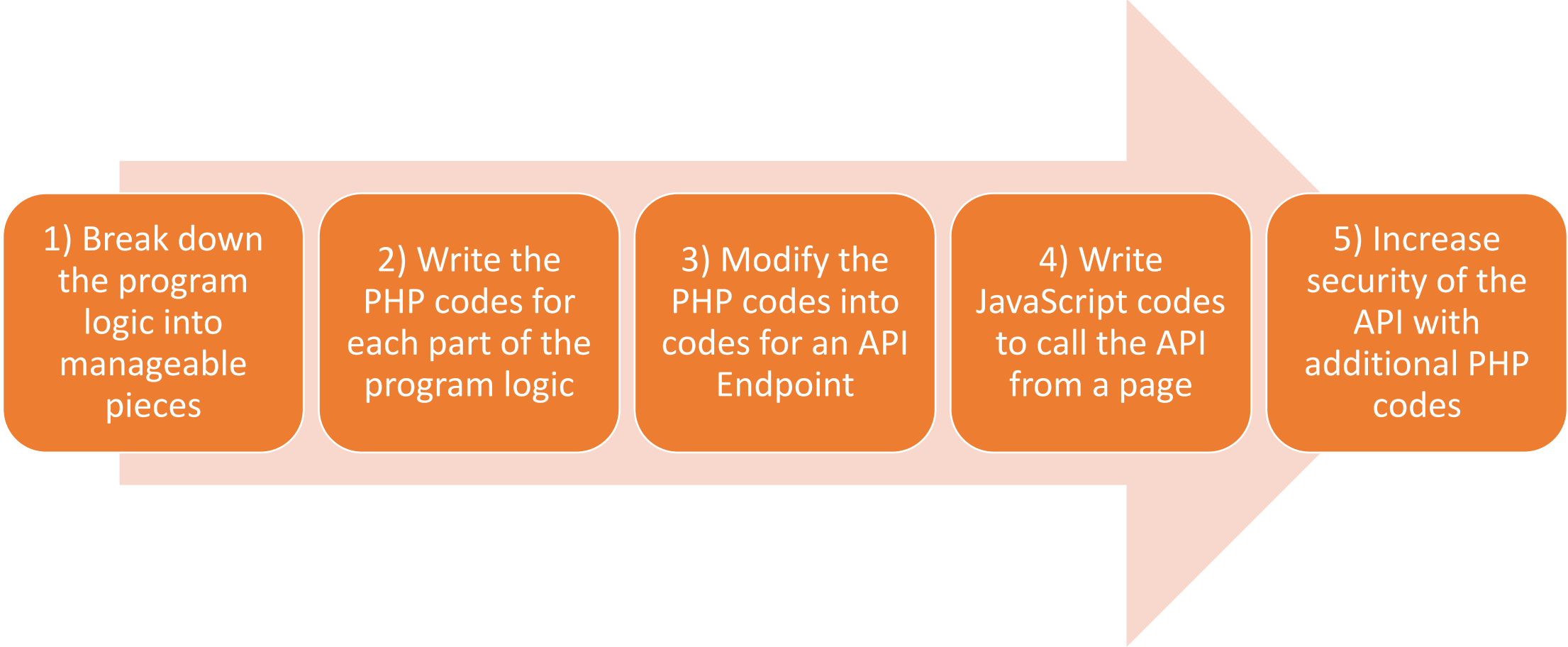
Explore

Search: wordpress Close

Bots People

-  **Wordpress**
15 monthly users
-  **WordPressDude**
Unleash your creativity with your new WordPress buddy! I'm a smart chatbot that suggests themes, plugins, and custom PHP code. Let's turn your vision into a stunning website, effortlessly!
291 monthly users
-  **Full-Stack WordPress**
Full-Stack_Wordpress: Your WordPress Development Assistant Welcome to Full-Stack_Wordpress, your dedicated partner in WordPress development. Assume the role of a skilled full-stack developer, ready to...
-  **wordpress-blog**
You can use this tool to write posts from your wordpress blog.
12 monthly users

How to Write an API?



1) Break down the program logic into manageable pieces

2) Write the PHP codes for each part of the program logic

3) Modify the PHP codes into codes for an API Endpoint

4) Write JavaScript codes to call the API from a page

5) Increase security of the API with additional PHP codes

How to Write an API with the Help of LLM?

Can take advantage of LLM's

1) Break down the program logic into manageable pieces

2) Write the PHP codes for each part of the program logic

3) Modify the PHP codes into codes for an API Endpoint

4) Write JavaScript codes to call the API from a page

5) Increase security of the API with additional PHP codes

Prompt Structure

According to a Google's talk, a good prompt should include

- Persona
- Context
- Task
- Format

Example:

- I am an Google Cloud account executive. Draft a cold outreach email to a CISO based on challenges in the recent months in the cybersecurity industry. Use bullet points.

Why prompt structure not important for this?

With the correct context in the prompt,

- Persona -> Implied to be a developer
- Format -> Implied to be program codes
- Task -> Often implied to be writing program codes

(Almost)
only
context
matters

- Easy!
- Mention WordPress (or simply “WP”) in your first prompt.
- If the logic is related to a plugin, mention the plugin, e.g., WooCommerce.

Step 2.1 - Coding Task 1 (Token blocks purchase)

Prompt:

I am building a website with WordPress using WooCommerce as the e-commerce plugin. I want to know how many of a product have been purchased by the current user.

Step 2.1 - Coding Task 1 (Token blocks purchase)

Prompt:

I am building a website with WordPress using WooCommerce as the e-commerce plugin. I want to know how many of a product have been purchased by the current user.

All contexts!



Sample Code Snippet





Sample Code Snippet

You can add this code to your theme's `functions.php` file or a custom plugin:

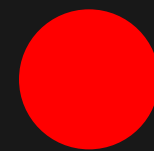
php

Copy

```
function get_user_product_purchase_count($product_id) {  
    // Check if user is logged in  
    if (!is_user_logged_in()) {  
        return 0; // Return 0 if not logged in  
    }  
  
    $user_id = get_current_user_id();  
    $purchase_count = 0;  
  
    // Get all orders for the current user  
    $customer_orders = wc_get_orders(array(  
        'customer_id' => $user_id,  
        'status' => 'completed', // You can adjust this for other statuses  
        'limit' => -1, // Get all orders  
    ));  
  
    // Loop through each order  
    foreach ($customer_orders as $order) {  
        // Check if the product is in the order  
        foreach ($order->get_items() as $item) {  
            if ($item->get_product_id() == $product_id) {  
                $purchase_count += $item->get_quantity(); // Add the quantity purchased  
            }  
        }  
    }  
  
    return $purchase_count; // Return total count  
}
```



Talk to Assistant or @ a bot





```
// Example usage
$product_id = 123; // Replace with your product ID
$count = get_user_product_purchase_count($product_id);
echo "You have purchased this product $count times.";
```

Explanation

- **is_user_logged_in():** Checks if the user is logged in.
- **wc_get_orders():** Retrieves the orders for the user, filtered by status.
- **Looping through orders:** Checks each order for the specified product and sums the quantities.

Usage

Replace 123 in the example usage with the actual product ID you want to check. You can place this code wherever you need to display the purchase count (e.g., in a template file or a shortcode).

This will give you a clear count of how many times the user has purchased a specific product.

Share



Compare



@Web-Search



Compare



@Llama-3-70b-Groq



Compare



@Mixtral-8x7B-Chat



How can I display this purchase count on my product page?



Can I modify the code to include other order statuses?



Talk to Assistant or @ a bot



Since I did not tell the LLM which product, it tells me I need to replace the placeholder product ID with the right ID



Edit PHP Snippet

Put the PHP codes into a shortcode

Tracking Name *

:

wcmy-tokens-purchased

PHP code *

:

```
<?php
function get_user_product_purchase_count($product_id) {
    // Check if user is logged in
    if (!is_user_logged_in()) {
        return 0; // Return 0 if not logged in
    }
}
```

```
$user_id = get_current_user_id();
$purchase_count = 0;
```

```
// Get all orders for the user
$customer_orders = wc_get_orders([
    'customer_id' => $user_id,
    'status' => 'completed',
    'limit' => -1, // Get all orders
```



Insert PHP Code Snippet

[Update Now](#)

Add PHP code to your pages and posts easily using shortcodes.

[More Details](#)

By [xyzscripts.com](#)

★★★★★ (672)

100,000+ Active Installations

Last Updated: 2 months ago

✓ Compatible with your version of WordPress

[back](#)

[Update](#)

Edit Page

Add New Page

Put the shortcode into a new page

Page updated. [View page](#)

WCMY-Demo

Permalink: <https://hkaccount.ing/wcmy-demo/> Edit

Add Media

Pods Shortcode

Add Form

Add Contact Form



Classic Editor

Update Now

More Details

Enables the previous "classic" editor and the old-style Edit Post screen with TinyMCE, Meta Boxes, etc. Supports all plugins that extend this screen.

By WordPress Contributors

★★★★★ (1,191)

10+ Million Active Installations

Last Updated: 3 weeks ago

✓ Compatible with your version of WordPress

b i link b-quote del ins img ul ol li code

more close tags contact form

[xyz-ips snippet="wcmy-tokens-purchased"]

WPOL LOCK USER

☐ Lock Post Permanently

Lock user from editing this post from the frontend dashboard

Publish

Edit Page

Add New Page

Put the shortcode into a new page

Page updated. [View page](#)

WCMY-Demo

Permalink: <https://hkaccount.ing/wcmy-demo/> Edit

Add Media

Pods Shortcode

Add Form

Add Contact Form



Classic Editor

Update Now

Enables the previous "classic" editor and the old-style Edit Post screen with TinyMCE, Meta Boxes, etc. Supports all plugins that extend this screen.

More Details

Most Popular
WordPress
Plugin Ever!

★★★★★ (1,191)
10+ Million Active Installations

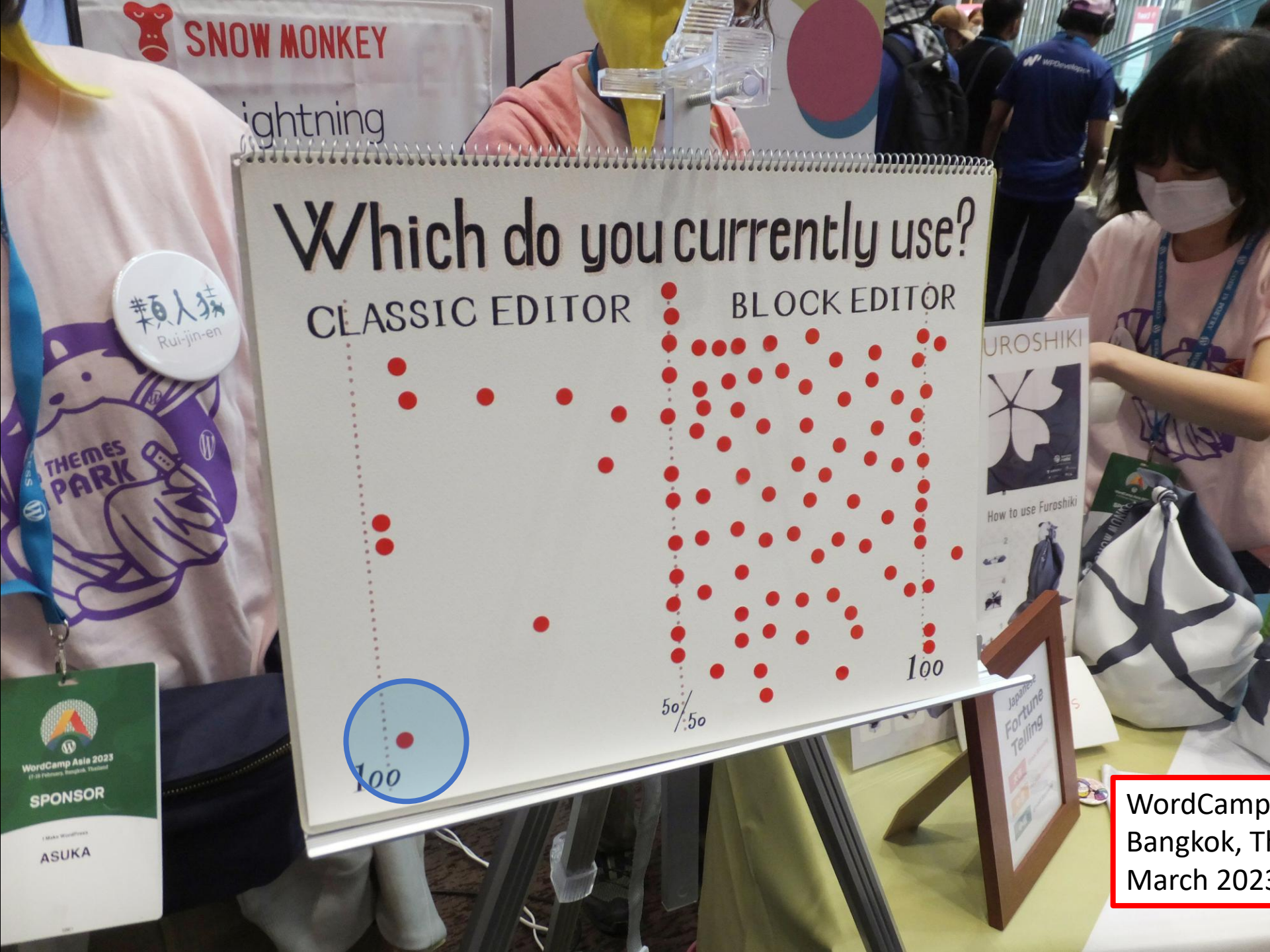
Last Updated: 3 weeks ago

✓ Compatible with your version of WordPress

b i link b-quote del ins img ul ol li code
more close tags contact form

[xyz-ips snippet="wcmy-tokens-purchased"]

Publish



WordCamp Asia 2023
Bangkok, Thailand
March 2023

Load the page

WCMY-Demo

You have purchased this product 18 times.

WCMY-Demo

You have purchased this product 18 times.

Correct!

Your LSP Genie Tokens	
Blocks Purchased	19
Tokens Purchased	390
Tokens Used	384
Tokens Available	6

Expecting 19 - 1 = 18

Step 2.3 – Coding Task 3 (Tokens used)

When a user do a calculation, an “lsp_calculation” post is created and the user designated as the author of the post. The number of tokens used will be stored in the custom field “lsp_tokens_used” in the post.

Prompt:

I want to know how many tokens have been used by the current user. It is the sum of values in the field "lsp_tokens_used" in all posts of type "lsp_calculation" when the field "post_author" is the user ID of the current user.



I want to know how many tokens have been used by the current user. It is the sum of values in the field "lsp_tokens_used" in all posts of type "lsp_calculation" when the field "post_author" is the user ID of the current user.



Assistant Poe

To calculate the total number of tokens used by the current user based on the `lsp_tokens_used` field from all posts of type `lsp_calculation`, you can follow these steps:

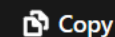
Steps to Retrieve Tokens Used by Current User

1. **Get Current User ID:** Retrieve the ID of the currently logged-in user.
2. **Query Posts:** Use a custom query to get all posts of type `lsp_calculation` authored by the current user.
3. **Sum the `lsp_tokens_used` Values:** Loop through the retrieved posts and sum the values in the `lsp_tokens_used` field.

Sample Code Snippet

You can add this code to your theme's `functions.php` file or a custom plugin:

php

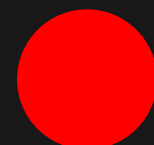


Copy

```
function get_user_tokens_used() {  
    // Check if user is logged in  
    if (!is_user_logged_in()) {  
        return 0; // Return 0 if not logged in  
    }  
  
    $user_id = get_current_user_id();
```



Talk to Assistant or @ a bot





```
$total_tokens_used = 0;

// Query posts of type 'lsp_calculation' authored by the current user
$args = array(
    'post_type' => 'lsp_calculation',
    'author' => $user_id,
    'posts_per_page' => -1, // Get all posts
    'fields' => 'ids' // Only retrieve post IDs for efficiency
);

$posts = get_posts($args);

// Loop through each post and sum the lsp_tokens_used
foreach ($posts as $post_id) {
    $tokens_used = get_post_meta($post_id, 'lsp_tokens_used', true);
    $tokens_used = intval($tokens_used); // Ensure it's treated as an integer
    $total_tokens_used += $tokens_used; // Add to total
}

return $total_tokens_used; // Return total tokens used
}

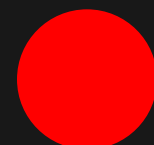
// Example usage
$count = get_user_tokens_used();
echo "You have used a total of $count tokens.";
```

Explanation

- **is_user_logged_in():** Checks if the user is logged in.
- **get_current_user_id():** Retrieves the ID of the current user.
- **get_posts():** Custom query to fetch all posts of type `lsp_calculation` authored by the user.
- **get_post_meta():** Fetches the value of `lsp_tokens_used` for each post.



Talk to Assistant or @ a bot





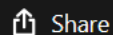
```
$count = get_user_tokens_used();  
echo "You have used a total of $count tokens.";
```

Explanation

- `is_user_logged_in()`: Checks if the user is logged in.
- `get_current_user_id()`: Retrieves the ID of the current user.
- `get_posts()`: Custom query to fetch all posts of type `lsp_calculation` authored by the user.
- `get_post_meta()`: Fetches the value of `lsp_tokens_used` for each post.

Usage

You can call `get_user_tokens_used()` anywhere in your theme or plugin where you want to display the total tokens used by the user. This function will return the sum based on the specified criteria.



Share



Compare



@Web-Search



Compare



@Llama-3-70b-Groq



Compare



@Mixtral-8x7B-Chat



How can I display the total tokens used on my profile page?



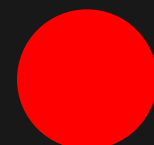
Can you help me create a shortcode for this function?



What if I want to filter posts by date range?



Talk to Assistant or @ a bot





Edit PHP Snippet

Put the new PHP codes into another
shortcode

Tracking Name * :

wcmy-tokens-used

PHP code * :

```
<?php
function get_user_tokens_used() {
    // Check if user is logged in
    if (!is_user_logged_in()) {
        return 0; // Return 0 if not logged in
    }

    $user_id = get_current_user_id();
    $total_tokens_used = 0;

    // Query posts of type 'lsp_calculation' authored by the current user
    $args = array(
        'post_type' => 'lsp_calculation',
        'author' => $user_id,
        'posts_per_page' => -1, // Get all posts
    );
```

back

Update

Edit Page

[Add New Page](#)

Page updated. [View page](#)

WCMY-Demo

Permalink: <https://hkaccount.ing/wcmy-demo/> [Edit](#)

[Add Media](#)

[Pods Shortcode](#)

[Add Form](#)

[Add Contact Form](#)

Visual

Text

[b](#) [i](#) [link](#) [b-quote](#) [del](#) [ins](#) [img](#) [ul](#) [ol](#) [li](#) [code](#)

[more](#) [close tags](#) [contact form](#)

```
[xyz-ips snippet="wcmy-tokens-purchased"]
```

```
[xyz-ips snippet="wcmy-tokens-used"]
```

Put the new shortcode
into the page

WPUF Form

--

[Learn more](#)

WPUF Lock User

☐ Lock Post Permanently

Lock user from editing this post from the
frontend dashboard

[Publish](#)

WCMY-Demo

You have purchased this product 18 times.

You have used a total of 384 tokens.

Load the page

WCMY-Demo

You have purchased this product 18 times.

You have used a total of 384 tokens.

Correct!

Your LSP Genie Tokens

Blocks Purchased	19
Tokens Purchased	384
Tokens Used	384
Tokens Available	0

Step 3: Modify the PHP codes into codes for an API Endpoint

Prompt:

I want to make results of the above functions available in an API.



I want to make results of the above functions available in an API.

Assistant Poe

To expose the results of the functions for user purchase counts and token usage via a REST API in WordPress, you can create custom endpoints. Here's how to do it step by step.

Step 1: Register Custom API Endpoints

You can add the following code to your theme's `functions.php` file or a custom plugin to create the API endpoints.

```
php
// Function to get user product purchase count
function get_user_product_purchase_count($product_id) {
    if (!is_user_logged_in()) {
        return 0; // Return 0 if not logged in
    }

    $user_id = get_current_user_id();
    $purchase_count = 0;

    $customer_orders = wc_get_orders(array(
        'customer_id' => $user_id,
        'status' => 'completed',
        'limit' => -1,
    ));

    foreach ($customer_orders as $order) {
        foreach ($order->get_items() as $item) {
            if ($item->get_product_id() == $product_id) {
                $purchase_count += $item->get_quantity();
            }
        }
    }
}
```

Copy

Note that the codes are to be put in a different location



Talk to Assistant or @ a bot

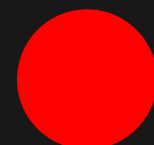




```
    }  
  }  
  
  return $purchase_count;  
}  
  
// Function to get user tokens used  
function get_user_tokens_used() {  
  if (!is_user_logged_in()) {  
    return 0; // Return 0 if not logged in  
  }  
  
  $user_id = get_current_user_id();  
  $total_tokens_used = 0;  
  
  $args = array(  
    'post_type' => 'lsp_calculation',  
    'author' => $user_id,  
    'posts_per_page' => -1,  
    'fields' => 'ids'  
  );  
  
  $posts = get_posts($args);  
  
  foreach ($posts as $post_id) {  
    $tokens_used = get_post_meta($post_id, 'lsp_tokens_used', true);  
    $tokens_used = intval($tokens_used);  
    $total_tokens_used += $tokens_used;  
  }  
  
  return $total_tokens_used;  
}  
  
// Register REST API routes  
add_action('rest_api_init', function () {  
  register_rest_route('custom/v1', '/purchase-count/(?P<product_id>\d+)', array(  
    'methods' => 'GET',  
  ));  
});
```



Talk to Assistant or @ a bot





```
'callback' => 'get_user_purchase_count_api',
'permission_callback' => '__return_true', // Adjust permissions as needed
));

register_rest_route('custom/v1', '/tokens-used', array(
    'methods' => 'GET',
    'callback' => 'get_user_tokens_used_api',
    'permission_callback' => '__return_true', // Adjust permissions as needed
));
});

// Callback for purchase count API
function get_user_purchase_count_api($data) {
    $product_id = intval($data['product_id']);
    $count = get_user_product_purchase_count($product_id);
    return rest_ensure_response(array('purchase_count' => $count));
}

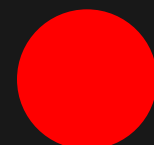
// Callback for tokens used API
function get_user_tokens_used_api() {
    $count = get_user_tokens_used();
    return rest_ensure_response(array('tokens_used' => $count));
}
```

Explanation of the Code

- **REST API Initialization:** The `rest_api_init` action is used to register custom routes.
- **register_rest_route:** Creates two endpoints:
 - GET `/wp-json/custom/v1/purchase-count/{product_id}`: Returns the number of times a specific product has been purchased by the current user.
 - GET `/wp-json/custom/v1/tokens-used`: Returns the total number of tokens used by the current user.
- **Callbacks:** Each route has a corresponding callback function that processes the request



Talk to Assistant or @ a bot





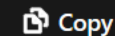
and returns the data as a JSON response.

Step 2: Accessing the API

You can access the API endpoints as follows:

- **Purchase Count:**

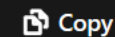
```
GET /wp-json/custom/v1/purchase-count/{product_id}
```



Replace {product_id} with the actual ID of the product.

- **Tokens Used:**

```
GET /wp-json/custom/v1/tokens-used
```



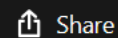
The API URI's to call

Permissions

- The `permission_callback` is set to `__return_true` for demonstration purposes. You may want to implement stricter permission checks to ensure that only authenticated users can access these endpoints.

Testing the API

You can test the API using tools like Postman or your browser (if you're authenticated) to ensure it returns the expected results.



Share



Talk to Assistant or @ a bot





Edit Snippet

Add New

Put the PHP codes into functions.php
using a plugin

wcmy-api

Code

PHP

Save Changes

Activate

<?php

```
1 // Function to get user product purchase count
2 function get_user_product_purchase_count($product_id) {
3     if (!is_user_logged_in()) {
4         return 0; // Return 0 if not logged in
5     }
6
7     $user_id = get_current_user_id();
8     $purchase_count = 0;
9
10    $customer_orders = wc_get_orders(array(
11        'customer_id' => $user_id,
12        'status' => 'completed',
13        'limit' => -1,
14    ));
15
```



Code Snippets

Update Now

An easy, clean and simple way to enhance your site with code snippets.

[More Details](#)

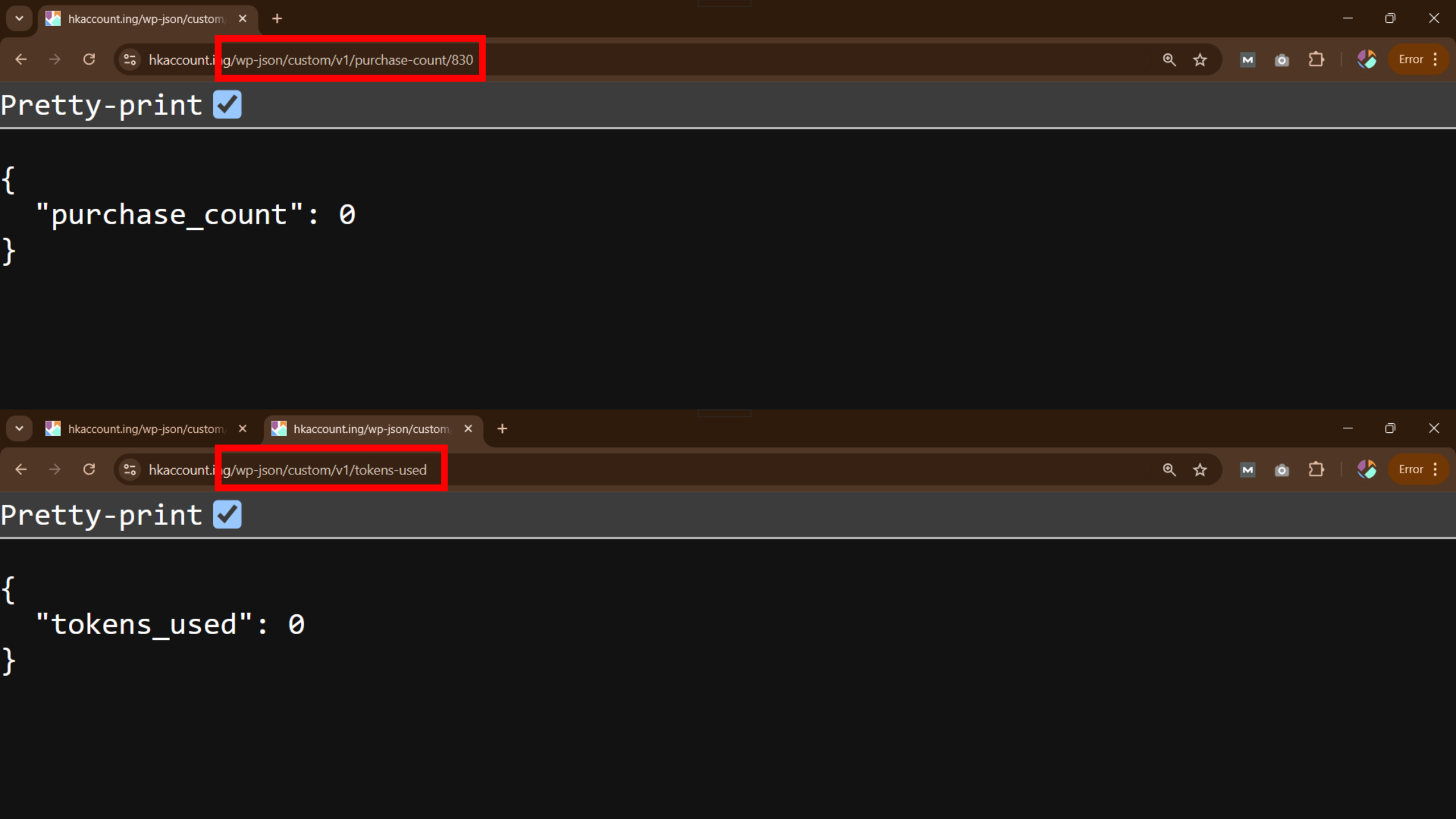
By Code Snippets Pro

★★★★☆ (451)

1+ Million Active Installations

Last Updated: 1 month ago

✓ Compatible with your version of WordPress





hkaccount.ing/wp-json/custom/v1/purchase-count/830

Pretty-print ✓

```
{  
  "purchase_count": 0  
}
```

Wrong!

hkaccount.ing/wp-json/custom/v1/tokens-used

Pretty-print ✓

```
{  
  "tokens_used": 0  
}
```

Step 3 Debugging

- Question: How does the server know whose data are needed without any parameters passed in the API?
- The LLM just copied the PHP functions into the API, and the API does not know who the user is with `get_current_user_id()` in the codes.

Revised prompt:

I want to make results of the above functions available in an API. But instead of using `get_current_user_id` in the API, I want to pass a designated userid to the API.



I want to make results of the above functions available in an API. But instead of using `get_current_user_id` in the API, I want to pass a designated userid to the API.

Assistant Poe

To modify the API endpoints to accept a designated user ID instead of using `get_current_user_id()`, you can adjust the callback functions to take a user ID parameter. Here's how to do it:

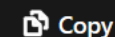
Step 1: Modify the Functions to Accept User ID

You will need to update both the functions and the API endpoints to accept a user ID as a parameter.

Updated Code Snippet

Add the following code to your theme's `functions.php` file or a custom plugin:

php



Copy

```
// Function to get user product purchase count
function get_user_product_purchase_count($user_id, $product_id) {
    if (empty($user_id)) {
        return 0; // Return 0 if user ID is not provided
    }

    $purchase_count = 0;

    $customer_orders = wc_get_orders(array(
        'customer_id' => $user_id,
        'status' => 'completed',
        'limit' => -1,
    ));
```



Talk to Assistant or @ a bot





```
foreach ($customer_orders as $order) {
    foreach ($order->get_items() as $item) {
        if ($item->get_product_id() == $product_id) {
            $purchase_count += $item->get_quantity();
        }
    }
}

return $purchase_count;
}

// Function to get user tokens used
function get_user_tokens_used($user_id) {
    if (empty($user_id)) {
        return 0; // Return 0 if user ID is not provided
    }

    $total_tokens_used = 0;

    $args = array(
        'post_type' => 'lsp_calculation',
        'author' => $user_id,
        'posts_per_page' => -1,
        'fields' => 'ids'
    );

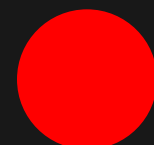
    $posts = get_posts($args);

    foreach ($posts as $post_id) {
        $tokens_used = get_post_meta($post_id, 'lsp_tokens_used', true);
        $tokens_used = intval($tokens_used);
        $total_tokens_used += $tokens_used;
    }

    return $total_tokens_used;
}
```



Talk to Assistant or @ a bot





```
// Register REST API routes
add_action('rest_api_init', function () {
    register_rest_route('custom/v1', '/purchase-count/(?P<user_id>\d+)/(?P<product_id>\d+)',
    array(
        'methods' => 'GET',
        'callback' => 'get_user_purchase_count_api',
        'permission_callback' => '__return_true', // Adjust permissions as needed
    ));

    register_rest_route('custom/v1', '/tokens-used/(?P<user_id>\d+)', array(
        'methods' => 'GET',
        'callback' => 'get_user_tokens_used_api',
        'permission_callback' => '__return_true', // Adjust permissions as needed
    ));
});

// Callback for purchase count API
function get_user_purchase_count_api($data) {
    $user_id = intval($data['user_id']);
    $product_id = intval($data['product_id']);
    $count = get_user_product_purchase_count($user_id, $product_id);
    return rest_ensure_response(array('purchase_count' => $count));
}

// Callback for tokens used API
function get_user_tokens_used_api($data) {
    $user_id = intval($data['user_id']);
    $count = get_user_tokens_used($user_id);
    return rest_ensure_response(array('tokens_used' => $count));
}
```

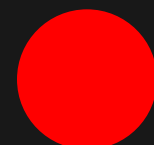
Explanation of Changes

1. Modified Functions:

- Both `get_user_product_purchase_count` and `get_user_tokens_used` now



Talk to Assistant or @ a bot





accept a `$user_id` parameter.

2. Updated API Endpoints:

- The GET `/wp-json/custom/v1/purchase-count/{user_id}/{product_id}` endpoint now accepts a `user_id` parameter.
- The GET `/wp-json/custom/v1/tokens-used/{user_id}` endpoint now also accepts a `user_id` parameter.

3. Updated Callbacks:

- The API callback functions now read the user ID from the request data and pass it to the respective functions.

Step 2: Accessing the Updated API

You can access the updated API endpoints as follows:

• Purchase Count:

```
GET /wp-json/custom/v1/purchase-count/{user_id}/{product_id}
```

[Copy](#)

Replace `{user_id}` with the actual user ID and `{product_id}` with the actual product ID.

• Tokens Used:

```
GET /wp-json/custom/v1/tokens-used/{user_id}
```

[Copy](#)

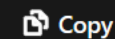
Replace `{user_id}` with the actual user ID.

The URI's now take product ID and user ID as parameters



Talk to Assistant or @ a bot





Copy

```
GET /wp-json/custom/v1/tokens-used/{user_id}
```

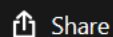
Replace {user_id} with the actual user ID.

Permissions

- The `permission_callback` is still set to `__return_true` for demonstration purposes. You may want to implement stricter permission checks to ensure that only authorized users can access these endpoints.

Testing the API

You can test the updated API using tools like Postman or a browser, ensuring that you receive the expected results based on the provided user ID.



Share



Compare



@Web-Search



Compare



@Command-R



Compare



@Gemini-1.5-Flash



How can I implement authentication for these API endpoints?



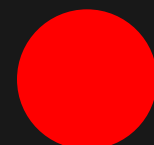
What are the best practices for securing my API?



Can you help me with error handling for the API responses?



Talk to Assistant or @ a bot





Edit Snippet

[Add New](#)

Snippet updated.

Put the revised PHP codes into
functions.php using plugin

wcmy-api

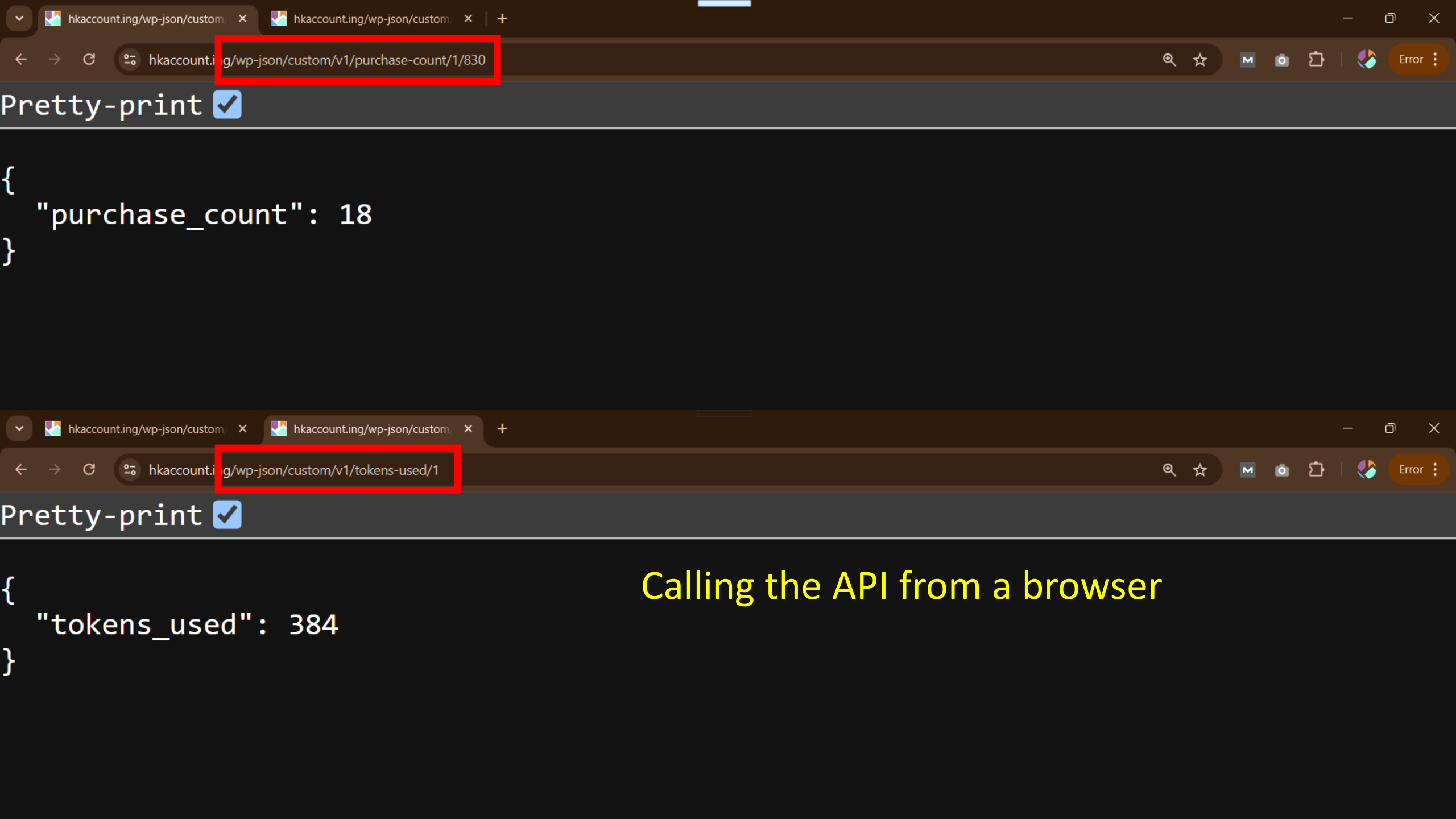
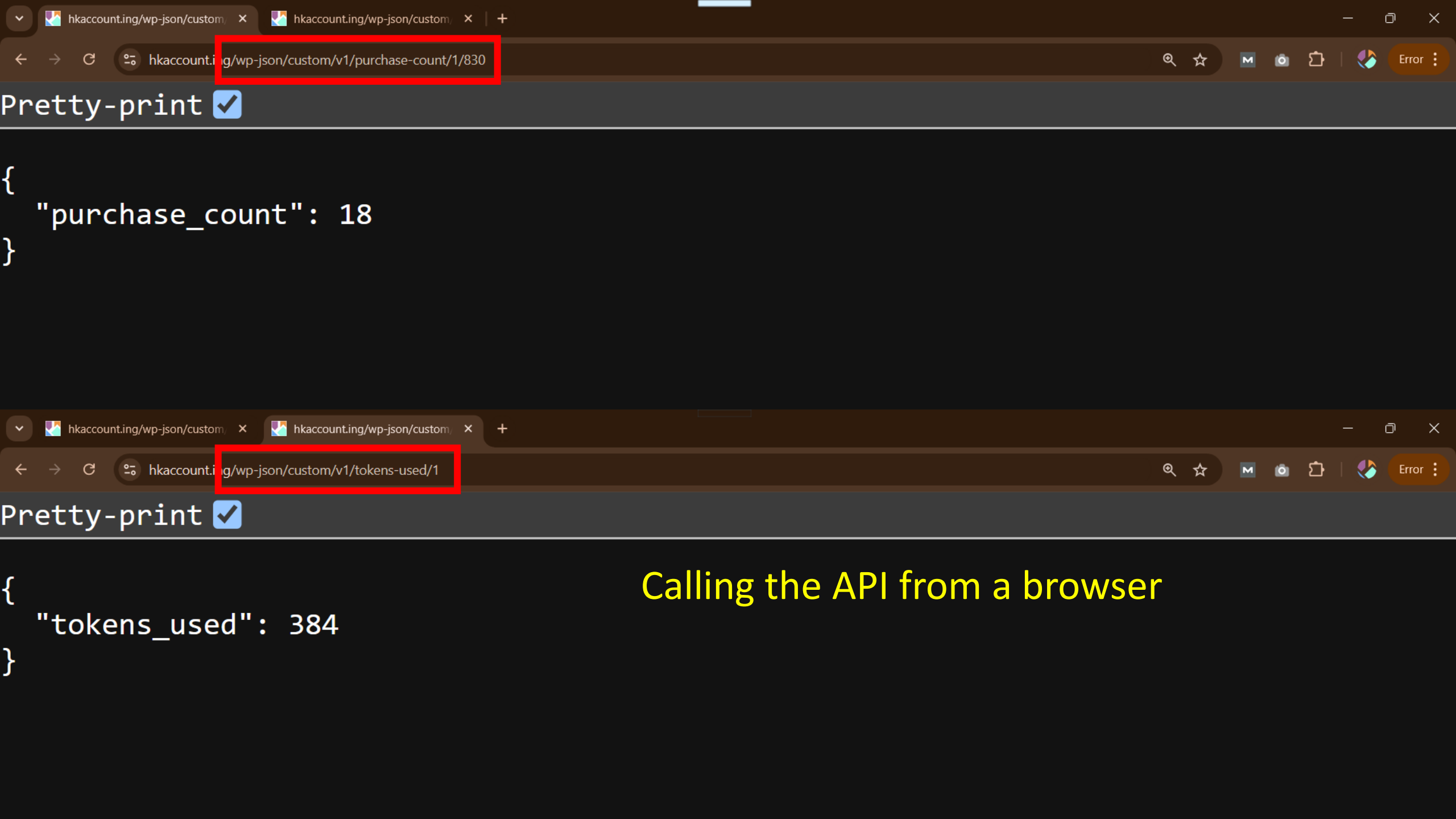
Code

PHP

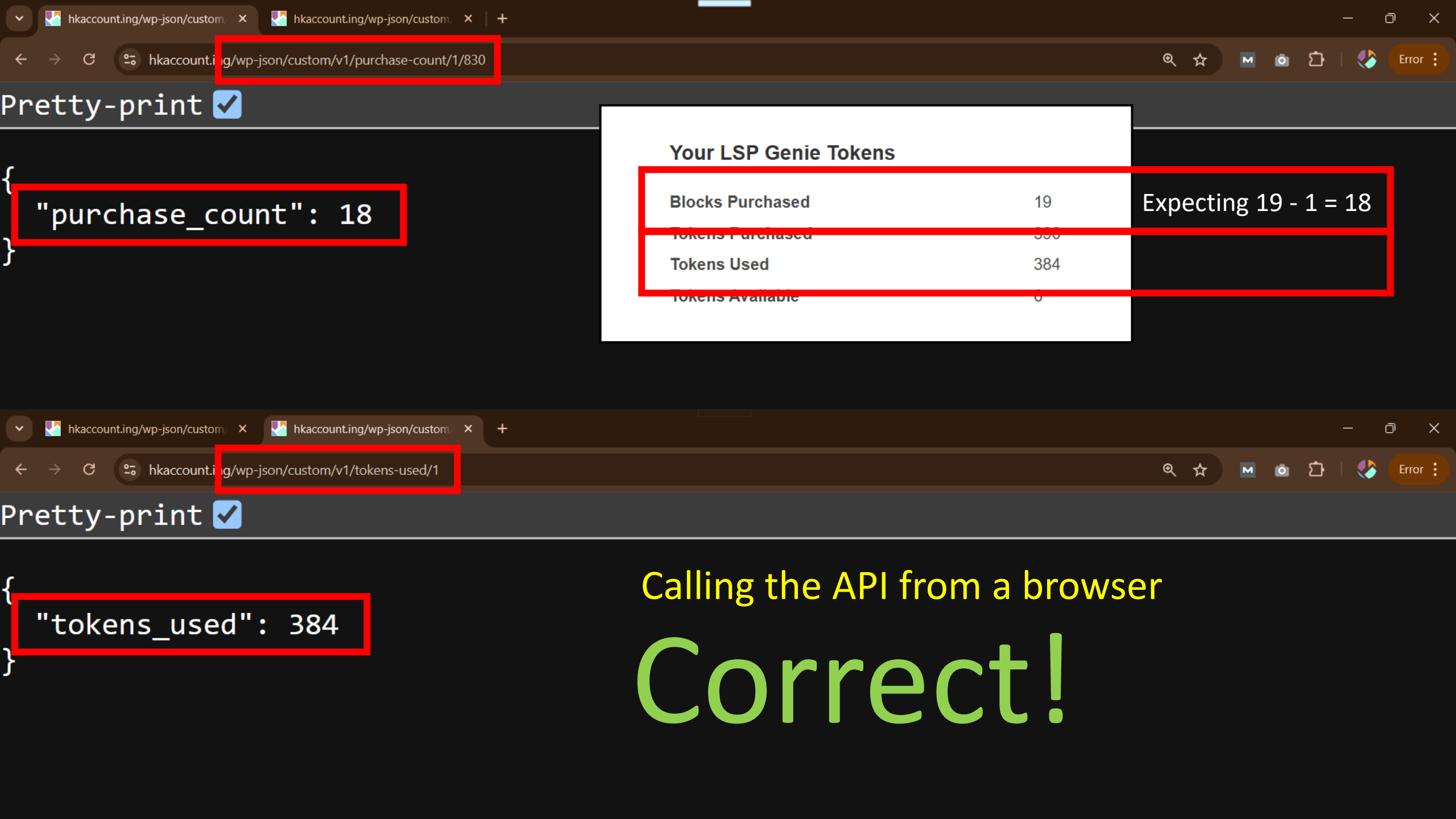
[Save Changes](#)[Deactivate](#)

<?php

```
1 // Function to get user product purchase count
2 function get_user_product_purchase_count($user_id, $product_id) {
3     if (empty($user_id)) {
4         return 0; // Return 0 if user ID is not provided
5     }
6
7     $purchase_count = 0;
8
9     $customer_orders = wc_get_orders(array(
10         'customer_id' => $user_id,
11         'status' => 'completed',
12         'limit' => -1,
13     ));
```



Calling the API from a browser



Calling the API from a browser

Correct!

Step 4: Write JavaScript codes to call the API from a page

- This step was skipped for time in WCMY...



Step 4: Write JavaScript codes to call the API from a page

Prompt:

How can I use JavaScript to call the API and display the result on a page?



How can I use JavaScript to call the API and display the result on a page?



Assistant

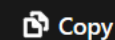
Poe

To call the API using JavaScript and display the results on a webpage, you can use the Fetch API. Below is a step-by-step guide on how to do this.

Step 1: Create an HTML Structure

Add an HTML structure with placeholders for displaying the results. For example:

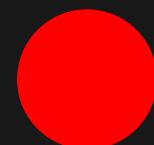
html



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>API Example</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    .result {
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <h1>User Token and Purchase Count</h1>
  <div>
    <label for="userId">User ID:</label>
    <input type="number" id="userId" required>
    <label for="productId">Product ID:</label>
    <input type="number" id="productId" required>
```



Talk to Assistant or @ a bot





```

    <input type="text" id="productId" required />
    <button id="fetchData">Fetch Data</button>
  </div>
  <div class="result">
    <h2>Results:</h2>
    <p id="purchaseCount"></p>
    <p id="tokensUsed"></p>
  </div>

  <script>
    document.getElementById('fetchData').addEventListener('click', function() {
      const userId = document.getElementById('userId').value;
      const productId = document.getElementById('productId').value;

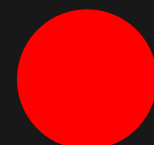
      // Fetch purchase count
      fetch(`https://yourdomain.com/wp-json/custom/v1/purchase-count/${userId}/${productId}`)
        .then(response => {
          if (!response.ok) {
            throw new Error('Network response was not ok');
          }
          return response.json();
        })
        .then(data => {
          document.getElementById('purchaseCount').innerText = `Purchase Count: ${data.purchase_count}`;
        })
        .catch(error => {
          console.error('Error fetching purchase count:', error);
          document.getElementById('purchaseCount').innerText = 'Error fetching purchase count.';
        });

      // Fetch tokens used
      fetch(`https://yourdomain.com/wp-json/custom/v1/tokens-used/${userId}`)
        .then(response => {
          if (!response.ok) {
            throw new Error('Network response was not ok');
          }
        });
    });
  </script>

```



Talk to Assistant or @ a bot





```
if (!response.ok) {  
    throw new Error('Network response was not ok');  
}  
return response.json();  
})  
.then(data => {  
    document.getElementById('tokensUsed').innerText = `Tokens Used: ${data.tokens_used}`;  
})  
.catch(error => {  
    console.error('Error fetching tokens used:', error);  
    document.getElementById('tokensUsed').innerText = 'Error fetching tokens used.';  
});  
});  
</script>  
</body>  
</html>
```

I did not tell the LLM where the values of the parameters come from, so it suggests to use a form

Refresh New tab Share

User Token and Purchase Count

User ID: Product ID: Fetch Data

Results:

Step 2: Explanation of the Code

1. HTML Structure:

- Two input fields for user ID and product ID.
- A button to trigger the fetch operation.



Talk to Assistant or @ a bot





...ation to trigger the fetch operation.

- Two paragraphs to display the results.

2. JavaScript Code:

- An event listener on the button (fetchData) calls the API when clicked.
- The Fetch API is used to make GET requests to your custom endpoints.
- The results are displayed in the respective paragraphs (purchaseCount and tokensUsed).
- Error handling is included to display messages if the fetch fails.

I did not tell the LLM my domain name

Step 3: Replace yourdomain.com

Make sure to replace `https://yourdomain.com` with your actual domain where your WordPress site is hosted.

Step 4: Testing

1. Save the HTML structure in a file (e.g., `index.html`).
2. Open the file in your browser.
3. Enter a user ID and product ID, then click "Fetch Data" to see the results.

Note on CORS

If you are calling the API from a different domain (e.g., a separate frontend application), ensure that your WordPress site has CORS (Cross-Origin Resource Sharing) configured to allow requests from that domain. You can do this by adding the following code to your `functions.php`:

php

Copy



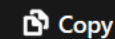
Talk to Assistant or @ a bot





If you are calling the API from a different domain (e.g., a separate frontend application), ensure that your WordPress site has CORS (Cross-Origin Resource Sharing) configured to allow requests from that domain. You can do this by adding the following code to your `functions.php`:

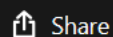
php



```
add_action('rest_api_init', function () {  
    header("Access-Control-Allow-Origin: *"); // Adjust as needed  
});
```

Conclusion

This setup will allow you to call your custom API endpoints and display the results dynamically on a webpage using JavaScript. Adjust the styling and functionality as needed to fit your requirements!



Share



Compare



@Web-Search



Compare



@Mixtral-8x7B-Chat



Compare



@Claude-3-Haiku



How can I handle errors more effectively in the fetch requests?



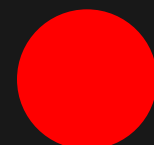
What are some best practices for securing my API endpoints?



Can you explain how to implement pagination for API results?



Talk to Assistant or @ a bot



Step 4 Follow Up

Prompt:

Instead of getting the product and user using a form, I want the page to call the APIs itself and display the results on the page. The userId should be the userId of the current user, while the productId is always 830.



Instead of getting the product and user using a form, I want the page to call the APIs itself and display the results on the page. The userId should be the userId of the current user, while the productId is always 830.



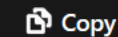
Assistant Poe

To modify the setup so that the page automatically calls the API using the current user's ID and a fixed product ID (830), you can use JavaScript to retrieve the current user ID from a WordPress global variable or a localized script. Below is how to do that.

Step 1: Localize the Script to Get the Current User ID

First, you need to ensure that the current user's ID is available in your JavaScript code. You can do this by using `wp_localize_script` in your theme's `functions.php` file:

php



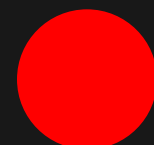
```
function enqueue_custom_script() {  
    // Enqueue your custom JavaScript file (adjust the path as necessary)  
    wp_enqueue_script('custom-api-script', get_template_directory_uri() . '/js/custom-api.js', array('jquery'), null, true);  
  
    // Localize script to pass the current user ID  
    wp_localize_script('custom-api-script', 'currentUser', array(  
        'id' => get_current_user_id()  
    ));  
}  
add_action('wp_enqueue_scripts', 'enqueue_custom_script');
```

Step 2: Create the JavaScript File

Create a JavaScript file named `custom-api.js` in your theme's `js` directory (adjust the path as necessary). In this file, you will make the API calls:

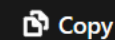


Talk to Assistant or @ a bot





javascript



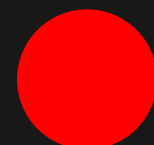
```
document.addEventListener('DOMContentLoaded', function() {
  const userId = currentUser.id; // Get the current user's ID
  const productId = 830; // Fixed product ID

  // Fetch purchase count
  fetch(`https://yourdomain.com/wp-json/custom/v1/purchase-count/${userId}/${productId}`)
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.json();
    })
    .then(data => {
      document.getElementById('purchaseCount').innerText = `Purchase Count: ${data.purchase_count}`;
    })
    .catch(error => {
      console.error('Error fetching purchase count:', error);
      document.getElementById('purchaseCount').innerText = 'Error fetching purchase count.';
    });

  // Fetch tokens used
  fetch(`https://yourdomain.com/wp-json/custom/v1/tokens-used/${userId}`)
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.json();
    })
    .then(data => {
      document.getElementById('tokensUsed').innerText = `Tokens Used: ${data.tokens_used}`;
    })
  });
```



Talk to Assistant or @ a bot



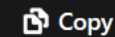


```
.catch(error => {  
    console.error('Error fetching tokens used:', error);  
    document.getElementById('tokensUsed').innerText = 'Error fetching tokens used.';  
});  
});
```

Step 3: HTML Structure

Ensure your HTML structure is ready to display the results. It should include the necessary elements:

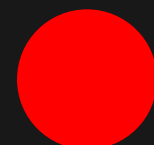
html



```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>User Token and Purchase Count</title>  
    <style>  
        body {  
            font-family: Arial, sans-serif;  
        }  
        .result {  
            margin-top: 20px;  
        }  
    </style>  
</head>  
<body>  
    <h1>User Token and Purchase Count</h1>  
    <div class="result">  
        <h2>Results:</h2>  
        <p id="purchaseCount"></p>  
        <p id="tokensUsed"></p>  
    </div>
```



Talk to Assistant or @ a bot





```
<!-- Enqueued script will be added here automatically -->
</body>
</html>
```

Refresh

New tab



Share

User Token and Purchase Count

Results:

Step 4: Replace yourdomain.com

Make sure to replace <https://yourdomain.com> with your actual domain where your WordPress site is hosted.

Step 5: Testing

1. Save all changes and ensure your JavaScript and PHP files are uploaded correctly.
2. Load the page in your browser. It should automatically call the APIs and display the results based on the current user's ID and the fixed product ID (830).

Conclusion

With this setup, the page will automatically retrieve the current user's ID and fetch the respective data from your API without any user input. Adjust styling and functionality as needed to fit your requirements!

Share



Talk to Assistant or @ a bot



Again, need to take care
of the domain name



Edit Snippet

[Add New](#)

Snippet updated and activated.

Replace the old PHP codes with the new PHP codes in functions.php using plugin

wcmy-api-2

Code [PHP](#)

[Save Changes](#)[Deactivate](#)

<?php

```
1 function enqueue_custom_script() {
2     // Enqueue your custom JavaScript file (adjust the path as necessary)
3     wp_enqueue_script('custom-api-script', get_template_directory_uri() . '/js/custom-api.js',
4     array('jquery'), null, true);
5
6     // Localize script to pass the current user ID
7     wp_localize_script('custom-api-script', 'currentUser', array(
8         'id' => get_current_user_id()
9     ));
10 add_action('wp_enqueue_scripts', 'enqueue_custom_script');
```

Edit Page

[Add New Page](#)

Page updated. [View page](#)

Put the HTML and JavaScript codes into a new page

WCMY-Demo-2

Permalink: <https://hkaccount.ing/wcm-demo-2/>

[Edit](#)

[Add Media](#)

[Pods Shortcode](#)

[Add Form](#)

[Add Contact Form](#)

Visual

Text

[b](#) [i](#) [link](#) [b-quote](#) [del](#) [ins](#) [img](#) [ul](#) [ol](#) [li](#) [code](#)
[more](#) [close tags](#) [contact form](#)

```
<div class="result">
  <h2>Results:</h2>
  <p id="purchaseCount"></p>
  <p id="tokensUsed"></p>
```

WPUF Form

--

[Learn more](#)

WPUF Lock User

☐ Lock Post Permanently

Lock user from editing this post from the frontend dashboard

Publish

 Add Media  Pods Shortcode  Add Form

 Add Contact Form

Visual Text

b *i* link **b-quote** ~~del~~ **ins** **img** **ul** **ol** **li** **code**

more close tags contact form

```
<script>
<!--
document.addEventListener('DOMContentLoaded', function() {
  const userId = currentUser.id; // Get the current user's ID
  const productId = 830; // Fixed product ID

  // Fetch purchase count
  fetch(`https://yourdomain.com/wp-json/custom/v1/purchase-
count/${userId}/${productId}`)
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.json();
    })
    .then(data => {
      // Handle the response data
    })
    .catch(error => {
      console.error('Error:', error);
    })
  });
</script>
```

Edit the domain
name of the URI

☐ Lock Post Permanently

Howdy, victor

Lock user from editing this post from the
frontend dashboard

Publish

Preview Changes

 Status: **Published** [Edit](#)

 Visibility: **Public** [Edit](#)

 Revisions: **10** [Browse](#)

 Published on: **Oct 19, 2024 at 17:46** [Edit](#)

 **SEO: 8 / 100**

[Copy to a new draft](#)

[Rewrite & Republish](#)

[Move to Trash](#)

Update

WCMY-Demo-2

Results:

Purchase Count: 18

Tokens Used: 384

WCMY-Demo-2

Your LSP Genie Tokens

| | |
|------------------|-----|
| Blocks Purchased | 19 |
| Tokens Purchased | 998 |
| Tokens Used | 384 |
| Tokens Available | 0 |

Expecting $19 - 1 = 18$

Results:

Purchase Count: 18

Tokens Used: 384

Correct!

Step 5: Increase security of the API with additional PHP codes

Prompt:

I want to increase the security of the API. It should only be called from a page from the server “yourdomain.com”.



I want to increase the security of the API. It should only be called from a page from the server "yourdomain.com".



Assistant Poe

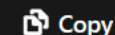
To enhance the security of your API and ensure that it can only be called from your server (e.g., yourdomain.com), you can implement a few measures:

1. **CORS (Cross-Origin Resource Sharing):** Allow only requests from your domain.
2. **Nonce Verification:** Use WordPress nonces to verify requests.
3. **Authentication:** Ensure the user is logged in (if applicable).

Step 1: Set Up CORS

You can restrict access to your API by adding a CORS header in your functions.php. This will allow requests only from your specified domain.

php



```
add_action('rest_api_init', function () {  
    if (isset($_SERVER['HTTP_ORIGIN'])) {  
        // Here you can specify your domain  
        $allowed_origin = 'https://yourdomain.com';  
  
        if (strpos($_SERVER['HTTP_ORIGIN'], $allowed_origin) !== false) {  
            header("Access-Control-Allow-Origin: $allowed_origin");  
        }  
    }  
});
```

Tells what additionally need to be done to tighten security

Step 2: Use Nonces for Security



Talk to Assistant or @ a bot

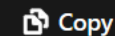




To prevent CSRF (Cross-Site Request Forgery) attacks, you can implement WordPress nonces. Here's how to add nonce verification:

1. **Generate a Nonce:** In your `functions.php` where you enqueue the script, generate a nonce.

php

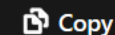


```
function enqueue_custom_script() {
    wp_enqueue_script('custom-api-script', get_template_directory_uri() . '/js/custom-api.js', array('jquery'), null, true);

    wp_localize_script('custom-api-script', 'apiData', array(
        'nonce' => wp_create_nonce('wp_rest'),
        'user_id' => get_current_user_id(),
    ));
}
add_action('wp_enqueue_scripts', 'enqueue_custom_script');
```

2. **Verify the Nonce in Your API Call:** Update your API callbacks to check the nonce.

php

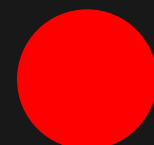


```
function get_user_purchase_count_api($data) {
    // Verify nonce
    if (!isset($_SERVER['HTTP_X_WP_NONCE']) || !wp_verify_nonce($_SERVER['HTTP_X_WP_NONCE'], 'wp_rest')) {
        return new WP_Error('rest_forbidden', __('You do not have permission to access this API.'), array('status' => 401));
    }

    $user_id = intval($data['user_id']);
    $product_id = intval($data['product_id']);
```



Talk to Assistant or @ a bot





```
product_id = intval($data['product_id']);
$count = get_user_product_purchase_count($user_id, $product_id);
return rest_ensure_response(array('purchase_count' => $count));
}

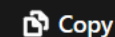
function get_user_tokens_used_api($data) {
    // Verify nonce
    if (!isset($_SERVER['HTTP_X_WP_NONCE']) || !wp_verify_nonce($_SERVER['HTTP_X_WP_NONCE'],
'wp_rest')) {
        return new WP_Error('rest_forbidden', __('You do not have permission to access this A
PI.'), array('status' => 401));
    }

    $user_id = intval($data['user_id']);
    $count = get_user_tokens_used($user_id);
    return rest_ensure_response(array('tokens_used' => $count));
}
```

Step 3: Update the JavaScript to Include the Nonce

Modify your JavaScript to send the nonce with the API requests:

javascript

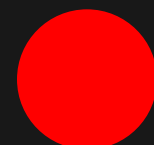


```
document.addEventListener('DOMContentLoaded', function() {
    const userId = apiData.user_id; // Get the current user's ID
    const productId = 830; // Fixed product ID
    const nonce = apiData.nonce; // Get the nonce

    // Fetch purchase count
    fetch(`https://yourdomain.com/wp-json/custom/v1/purchase-count/${userId}/${productId}`, {
        method: 'GET',
        headers: {
            'X-WP-Nonce': nonce, // Include nonce in the request
        },
    })
    .then(response => {
```



Talk to Assistant or @ a bot



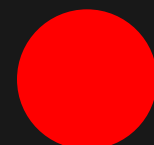


```
.then(response => {
  if (!response.ok) {
    throw new Error('Network response was not ok');
  }
  return response.json();
})
.then(data => {
  document.getElementById('purchaseCount').innerText = `Purchase Count: ${data.purchase_count}`;
})
.catch(error => {
  console.error('Error fetching purchase count:', error);
  document.getElementById('purchaseCount').innerText = 'Error fetching purchase count.';
});

// Fetch tokens used
fetch(`https://yourdomain.com/wp-json/custom/v1/tokens-used/${userId}`, {
  method: 'GET',
  headers: {
    'X-WP-Nonce': nonce, // Include nonce in the request
  },
})
.then(response => {
  if (!response.ok) {
    throw new Error('Network response was not ok');
  }
  return response.json();
})
.then(data => {
  document.getElementById('tokensUsed').innerText = `Tokens Used: ${data.tokens_used}`;
})
.catch(error => {
  console.error('Error fetching tokens used:', error);
  document.getElementById('tokensUsed').innerText = 'Error fetching tokens used.';
});
});
```



Talk to Assistant or @ a bot







Edit Snippet

[Add New](#)

Add the PHP codes in
functions.php using plugin

Snippet updated.

wcmy-api-2

Code

PHP

[Save Changes](#)

[Deactivate](#)

<?php

```
1 function enqueue_custom_script() {  
2     wp_enqueue_script('custom-api-script', get_template_directory_uri() . '/js/custom-api.js',  
   array('jquery'), null, true);  
3  
4     wp_localize_script('custom-api-script', 'apiData', array(  
5         'nonce' => wp_create_nonce('wp_rest'),  
6         'user_id' => get_current_user_id(),  
7     ));  
8 }  
9 add_action('wp_enqueue_scripts', 'enqueue_custom_script');
```



```
// Register REST API routes
```

```
53 add_action('rest_api_init', function () {
54     register_rest_route('custom/v1', '/purchase-count/(?P<user_id>\d+)/(?P<product_id>\d+)', array(
55         'methods' => 'GET',
56         'callback' => 'get_user_purchase_count_api',
57         'permission_callback' => '__return_true', // Adjust permissions as needed
58     ));
59
60     register_rest_route('custom/v1', '/tokens-used/(?P<user_id>\d+)', array(
61         'methods' => 'GET',
62         'callback' => 'get_user_tokens_used_api',
63         'permission_callback' => '__return_true', // Adjust permissions as needed
64     ));
65 });
66
67 add_action('rest_api_init', function () {
68     if (isset($_SERVER['HTTP_ORIGIN'])) {
69         // Here you can specify your domain
70         $allowed_origin = 'https://yourdomain.com';
71
72         if (strpos($_SERVER['HTTP_ORIGIN'], $allowed_origin) !== false) {
73             header("Access-Control-Allow-Origin: $allowed_origin");
74         }
75     }
76 });
77
```

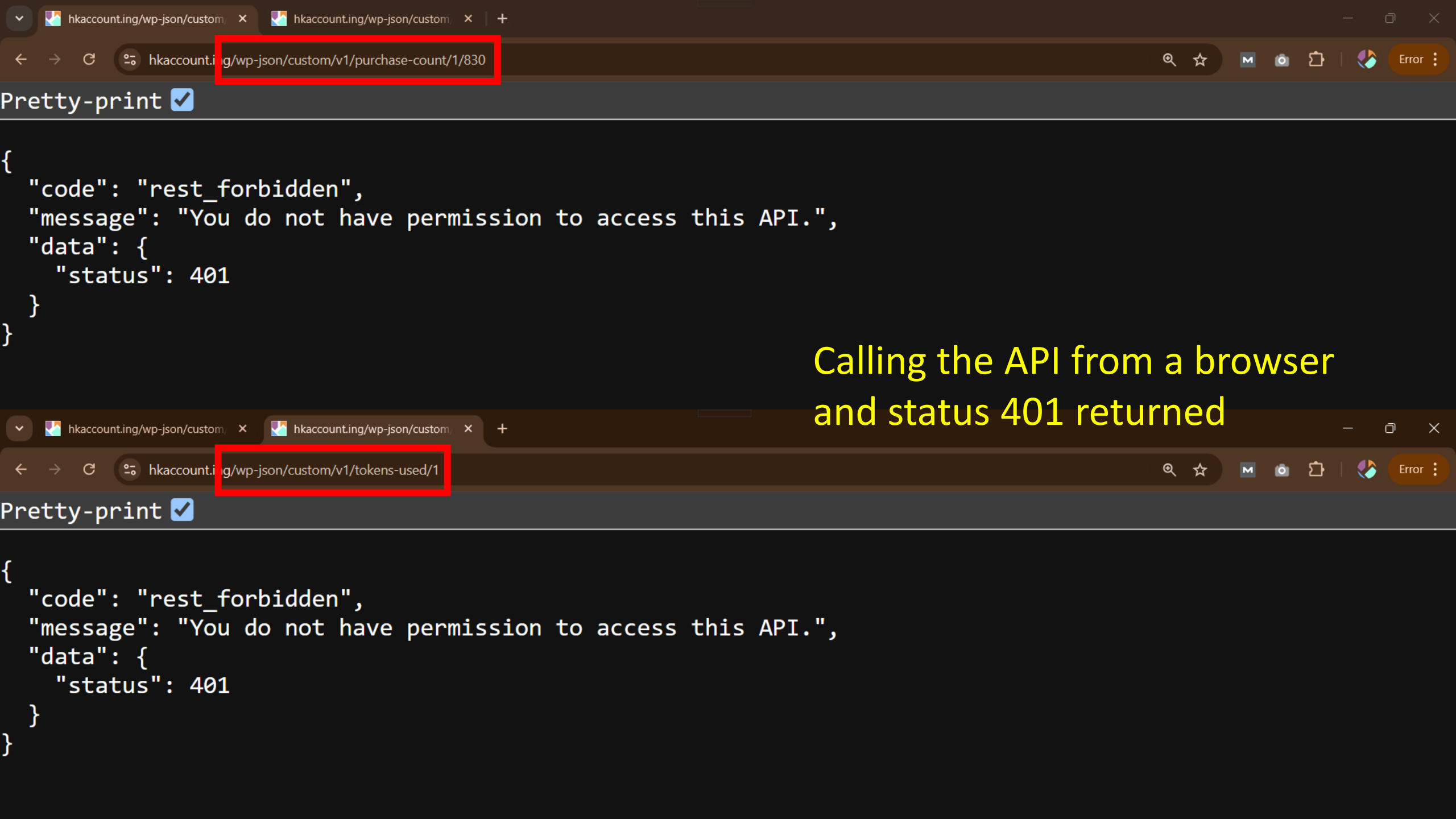
Update the other PHP codes
in functions.php with plugin

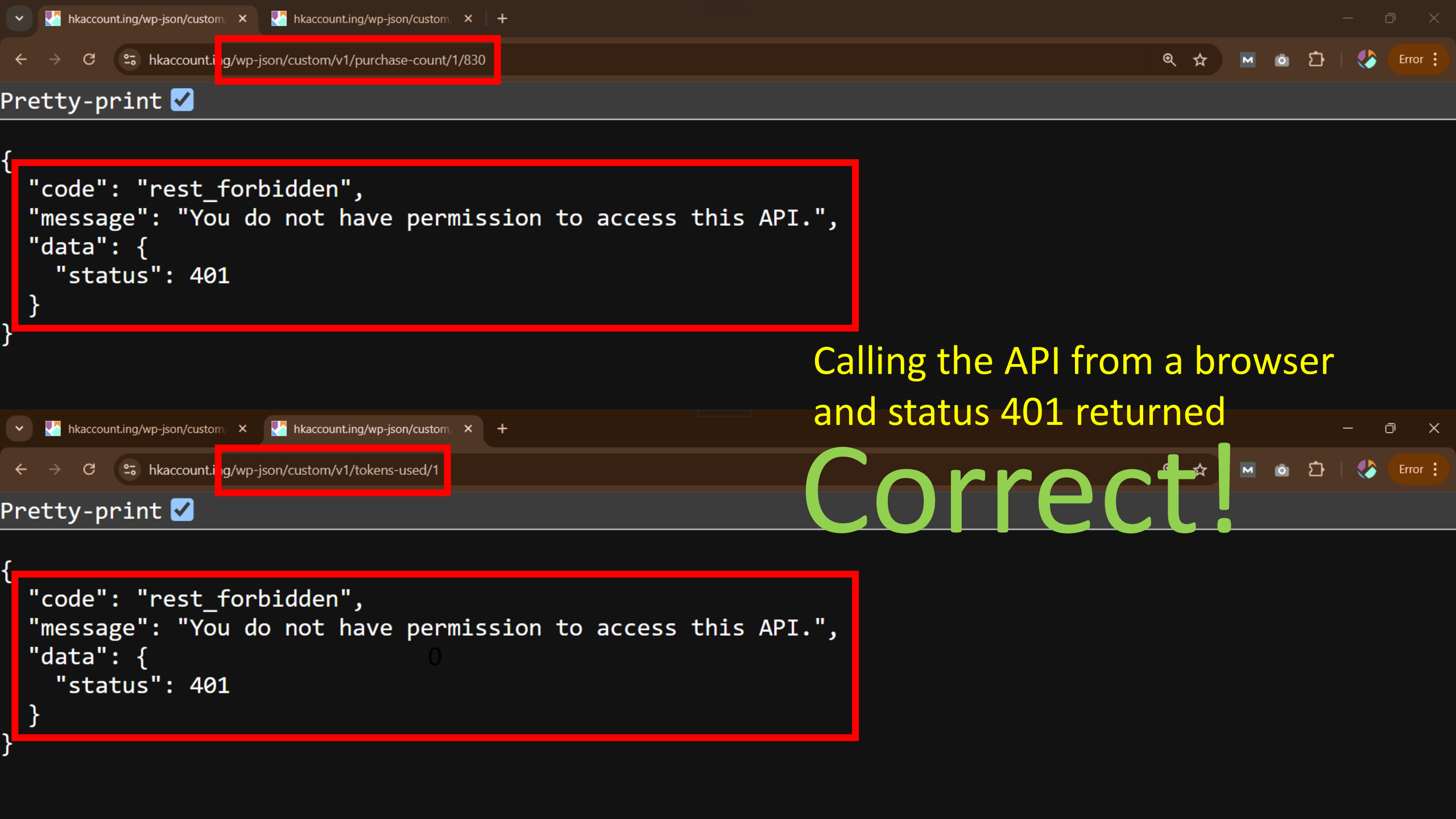
Edit the domain name of the URI
in the API codes



```
66
67 function get_user_purchase_count_api($data) {
68     // Verify nonce
69     if (!isset($_SERVER['HTTP_X_WP_NONCE']) || !wp_verify_nonce($_SERVER['HTTP_X_WP_NONCE'],
    'wp_rest')) {
70         return new WP_Error('rest_forbidden', __('You do not have permission to access this API.'),
    array('status' => 401));
71     }
72
73     $user_id = intval($data['user_id']);
74     $product_id = intval($data['product_id']);
75     $count = get_user_product_purchase_count($user_id, $product_id);
76     return rest_ensure_response(array('purchase_count' => $count));
77 }
78
79 function get_user_tokens_used_api($data) {
80     // Verify nonce
81     if (!isset($_SERVER['HTTP_X_WP_NONCE']) || !wp_verify_nonce($_SERVER['HTTP_X_WP_NONCE'],
    'wp_rest')) {
82         return new WP_Error('rest_forbidden', __('You do not have permission to access this API.'),
    array('status' => 401));
83     }
84
85     $user_id = intval($data['user_id']);
86     $count = get_user_tokens_used($user_id);
87     return rest_ensure_response(array('tokens used' => $count));
```







```
{
  "code": "rest_forbidden",
  "message": "You do not have permission to access this API.",
  "data": {
    "status": 401
  }
}
```

Calling the API from a browser
and status 401 returned

Correct!

```
{
  "code": "rest_forbidden",
  "message": "You do not have permission to access this API.",
  "data": {
    "status": 401
  }
}
```

Add Media Pods Shortcode Add Form

Add Contact Form

Update the JavaScript codes
in the page

Visual

Text

b i link b-quote del ins img ul ol li code

more close tags contact form

```
document.addEventListener('DOMContentLoaded', function() {  
  const userId = apiData.user_id; // Get the current user's ID  
  const productId = 830; // Fixed product ID  
  const nonce = apiData.nonce; // Get the nonce
```

```
  // Fetch purchase count  
  fetch(`https://yourdomain.com/wp-json/custom/v1/purchase-  
count/${userId}/${productId}`, {  
    method: 'GET',  
    headers: {  
      'X-WP-Nonce': nonce, // Include nonce in the request  
    },  
  })  
  .then(response => {
```

Edit the domain name of
the URI in the page

Publish

Preview Changes

Status: **Published** [Edit](#)

Visibility: **Public** [Edit](#)

Revisions: 12 [Browse](#)

Published on: Oct 19, 2024 at 17:46 [Edit](#)

SEO: 8 / 100

[Copy to a new draft](#)

[Rewrite & Republish](#)

[Move to Trash](#)

Update

Content AI

0 ? ^ v ▲



**HKAccount.ing**
We help you put the pieces together!

Load the page



WCMY-Demo-2

User Data

Tokens Used: 0

Purchase Count: 0

But there is an error displaying
the values on the page

Elements Console >> 5 18 15 | Settings | X

top | Filter Default levels

16 Issues: 15 1 | 3 hidden

JQMIGRATE: Migrate is installed, version 3.4.1

GET <https://hkaccount.ing/wp-content/themes/oceanwp/js/custom...> net::ERR_ABORTED 404 (Not Found)

17 Chrome is moving towards a new experience that lets people make an informed choice with respect to third-party cookies.

Error fetching tokens used: TypeError: [wcmly-demo-2/:683](#) Cannot set properties of null (setting 'innerText') at [wcmly-demo-2/:680:57](#)

Uncaught (in promise) TypeError: [wcmly-demo-2/:684](#) Cannot set properties of null (setting 'innerText') at [wcmly-demo-2/:684:57](#)

Error fetching purchase count: TypeError: Cannot set properties of null (setting 'innerText') at [wcmly-demo-2/:659:60](#)

Uncaught (in promise) TypeError: [wcmly-demo-2/:663](#) Cannot set properties of null (setting 'innerText') at [wcmly-demo-2/:663:60](#)

WCMY-Demo-2

Permalink: <https://hkaccount.ing/wcmy-demo-2/> [Edit](#)

 Add Media

 Pods Shortcode

 Add Form

 Add Contact Form

Visual

Text

b i link b-quote del ins img ul ol li code

more close tags contact form

```
<div id="user-data">
  <h2>User Data</h2>
  <p id="tokens-used">Tokens Used: <span id="tokens-
value">0</span></p>
  <p id="purchase-count">Purchase Count: <span id="purchase-
value">0</span></p>
</div>
```

```
<script>
```

```
<!--
```

The LLM forgot how it named the HTML elements

WPUF Form

Howdy, victor 

-- 

[Learn more](#)

WPUF Lock User

^ v ^

☐ Lock Post Permanently

Lock user from editing this post from the frontend dashboard

Publish

^ v ^

[Preview Changes](#)

 Status: **Published** [Edit](#)

 Visibility: **Public** [Edit](#)

 Add Media  Pods Shortcode  Add Form

 Add Contact Form

Visual

Text

b *i* [link](#) **b-quote** ~~del~~ **ins**  **ul** **ol** **li** **code**

more **close tags** **contact form**

```
}
return response.json();
})
.then(data => {
  document.getElementById('purchaseCount').innerText =
  `Purchase Count: ${data.purchase_count}`;
})
.catch(error => {
  console.error('Error fetching purchase count:', error);
  document.getElementById('purchaseCount').innerText =
  'Error fetching purchase count.';
});

// Fetch tokens used
```

Camel case is used in the
JavaScript codes

[Copy to a new draft](#)

Howdy, victor 

[Rewrite & Republish](#)

[Move to Trash](#)

Update

Content AI

0 ? ^ v ▲

 Research



Supercharge Your Content With AI

To access this Content AI
feature, you need to have an
active subscription plan.

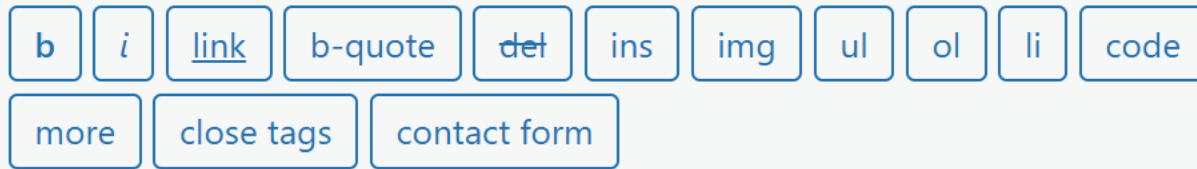
✓ 1-Click SEO Content

✓ 1-Click SEO Meta



Visual

Text



```
}
return response.json();
})
.then(data => {
  document.getElementById('tokensUsed').innerText = `Tokens
Used: ${data.tokens_used}`;
})
.catch(error => {
  console.error('Error fetching tokens used:', error);
  document.getElementById('tokensUsed').innerText = 'Error
fetching tokens used.';
});
});
//-->
</script>
```

Word count: 6

Last edited by victor on October 21, 2024 at 3:02 pm

Rank Math SEO



Howdy, victor



Learn More

USE KEYWORD IN

Content

Click on any keyword to copy it.

There are no recommended
Keywords for this researched
keyword.

Page Attributes



Parent

(no parent)



Template

WCMY-Demo-2

Permalink: <https://hkaccount.ing/wcmy-demo-2/> [Edit](#)

 Add Media

 Pods Shortcode

 Add Form

 Add Contact Form

Visual

Text

b*i*[link](#)**b-quote**~~del~~**ins****ul****ol****li****code**

[more](#)[close tags](#)[contact form](#)

```
<div id="user-data">
  <h2>User Data</h2>
  <p id="tokensUsed">Tokens Used: <span id="tokens-
value">0</span></p>
  <p id="purchaseCount">Purchase Count: <span id="purchase-
value">0</span></p>
</div>
```

Manually correct the ID's of
the HTML elements

WPUF Form

--

[Learn more](#)

WPUF Lock User

☐ Lock Post Permanently

Lock user from editing this post from the
frontend dashboard

Publish

[Preview Changes](#)

 Status: **Published** [Edit](#)

 Visibility: **Public** [Edit](#)

Load the page again

User Data

Tokens Used: 384

Purchase Count: 18

Load the page again

User Data

Tokens Used: 384

Purchase Count: 18

| Your LSP Genie Tokens | |
|-----------------------|-----|
| Blocks Purchased | 19 |
| Tokens Purchased | 366 |
| Tokens Used | 384 |
| Tokens Available | 0 |

Expecting 19 - 1 = 18

Correct!

I know what I have to do
and now I have done it!

I know what I have to do
and now I have done it!

**Mission
Accomplished!**

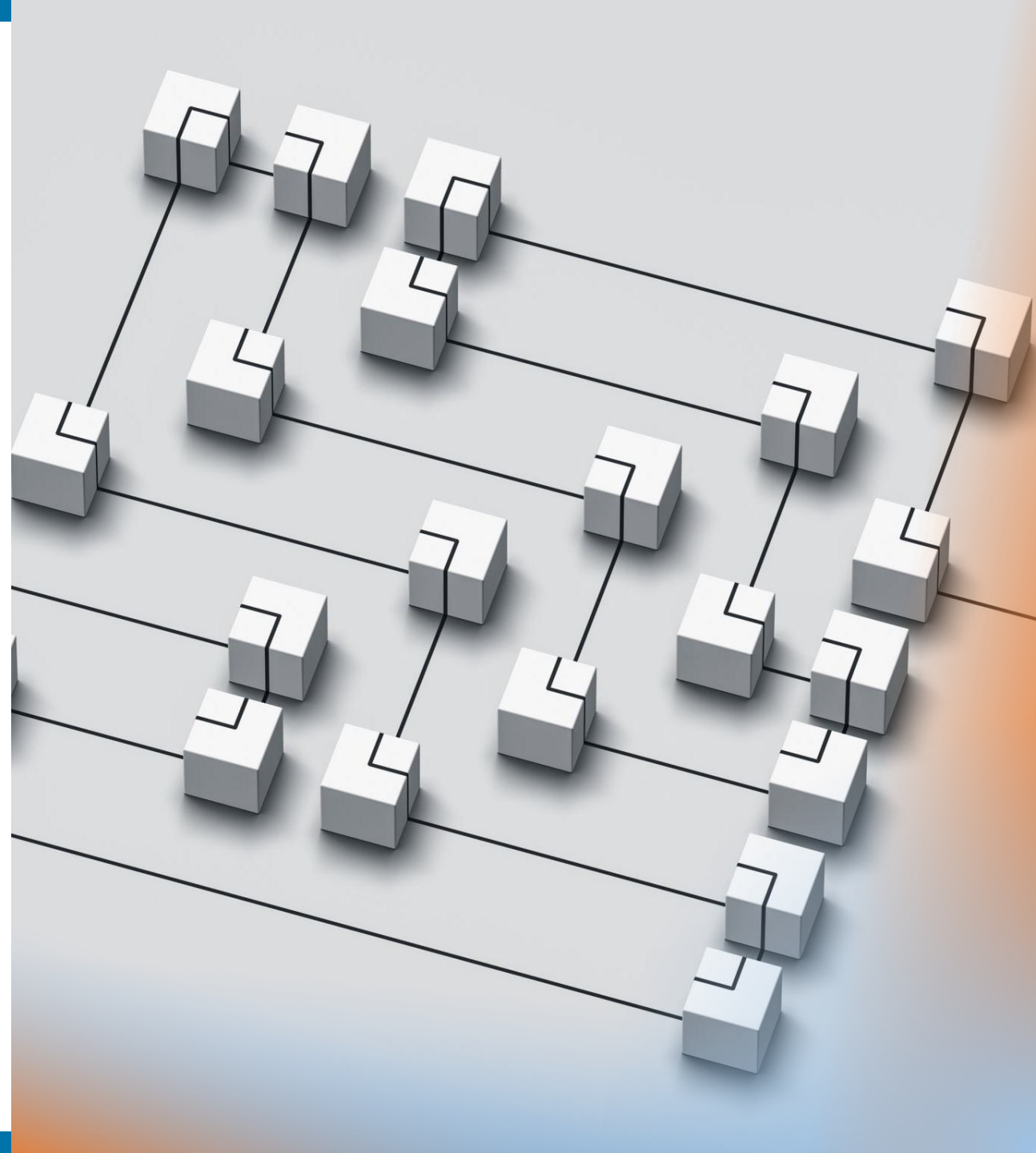
Next Step

- Modify the code to integrate with other parts of the website and beautify the page
- Obviously out-of-scope for today



Conclusion

- Now you know how RESTful API works in WordPress and you can use LLM to help you write API's in WordPress.
- LLM may help you do more than 90% of the coding, but you need to have the knowledge to fill in the gap.



How to Write an API with the Help of LLM?

Can take advantage of LLM's

1) Break down the program logic into manageable pieces

2) Write the PHP codes for each part of the program logic

3) Modify the PHP codes into codes for an API Endpoint

4) Write JavaScript codes to call the API from a page

5) Increase security of the API with additional PHP codes

Services/Plugins Used

- Poe.com (LLM)
- Insert PHP Code Snippet (PHP code insertion as shortcode)
- Classic Editor
- Code Snippets (PHP code insertion into functions.php)

Takeaway Question

How can you enhance your skills to deliver more sophisticated WordPress websites with the help of AI?



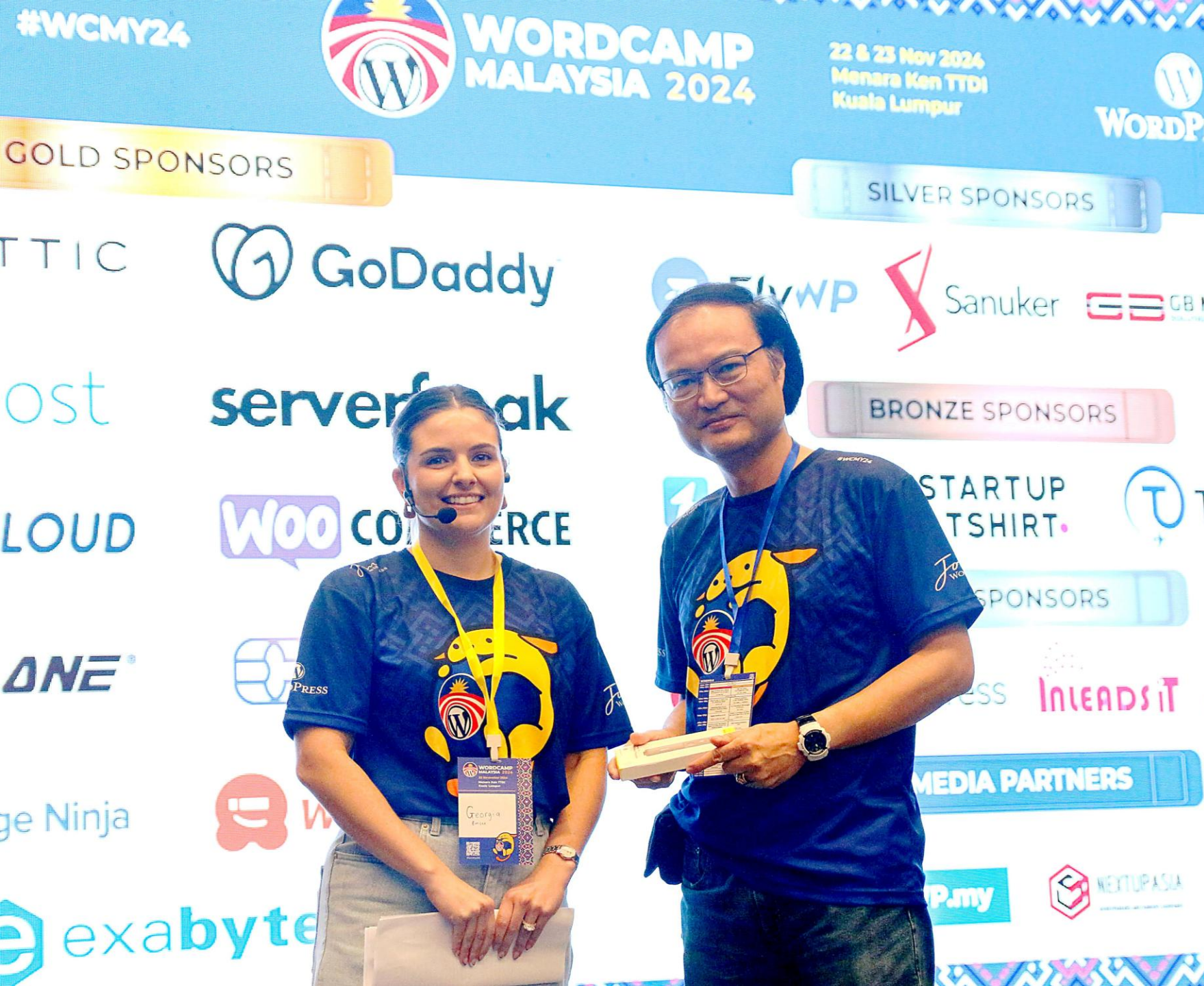
Takeaway Question

Hooks?

Filters?

How can you enhance your skills to deliver more sophisticated WordPress websites with the help of AI?

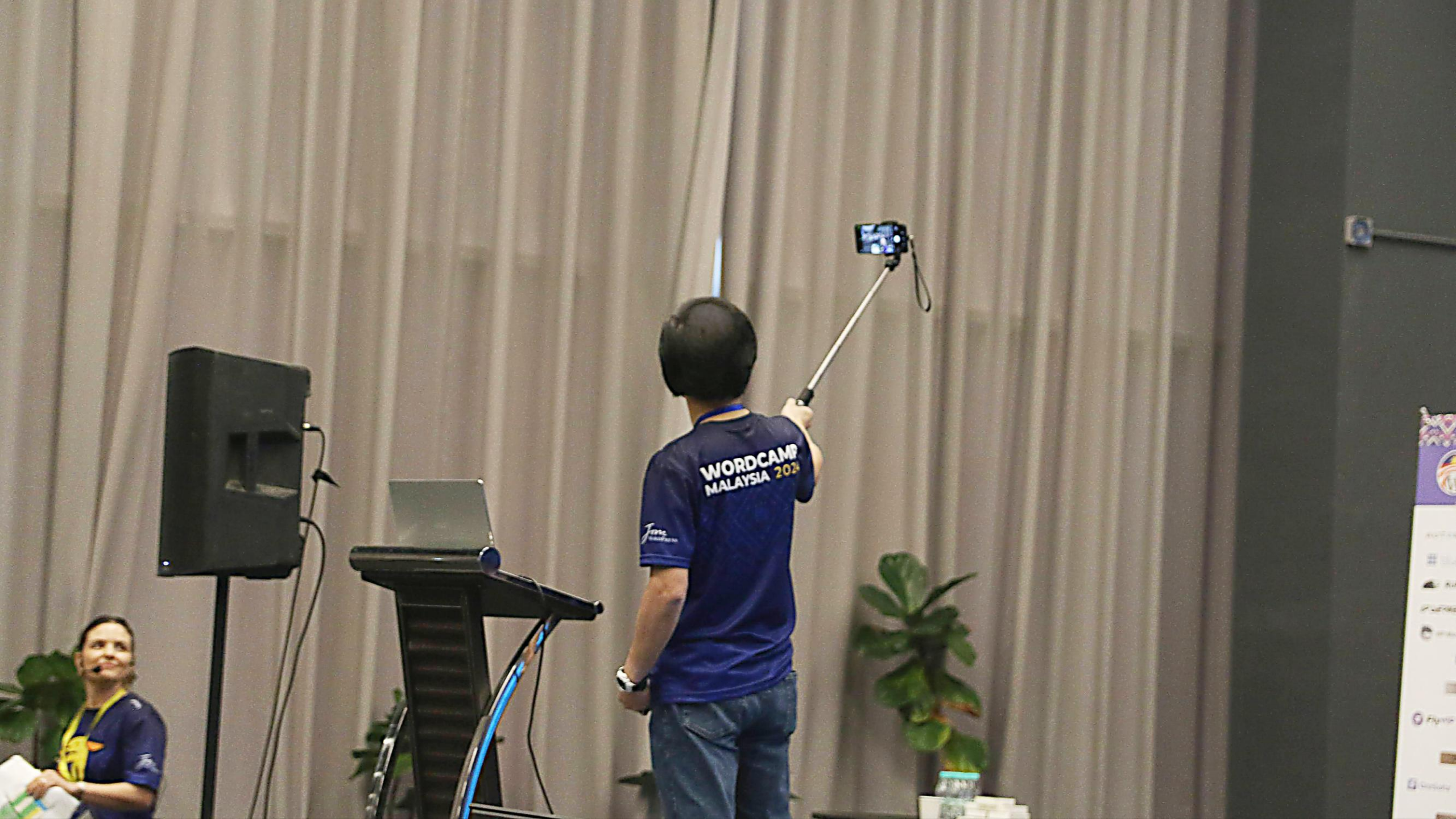




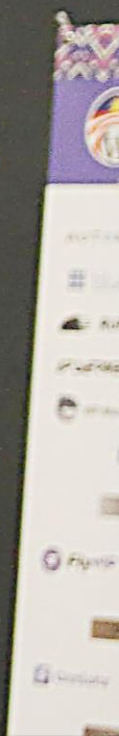
Thank you Speakers

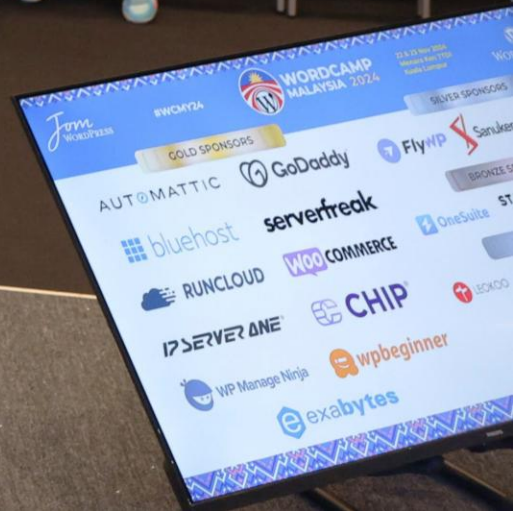
- Ahmed Kabir Chaion
- Aisha Preece
- A. Kumar
- C. L. ...
- P. G. ...
- Nathan Onn
- Nebu John
- Noor Khan
- Steven ...
- Tariq Sika





WORDCAMP
MALAYSIA 2024







WORDCAMP MALAYSIA 2024
22 & 23 Nov 2024
Merdeka City TTD
Kuala Lumpur

GOLD SPONSORS

- GoDaddy
- serverfreak
- RUNCLOUD
- COMMERCE
- CHIP
- wpbeginner
- bytes

SILVER SPONSORS

- FlyWP
- Sanuker
- LINEWORK

BRONZE SPONSORS

- STARTUP TSHIRT
- Tourfic

IRON SPONSORS

- INLEADS IT

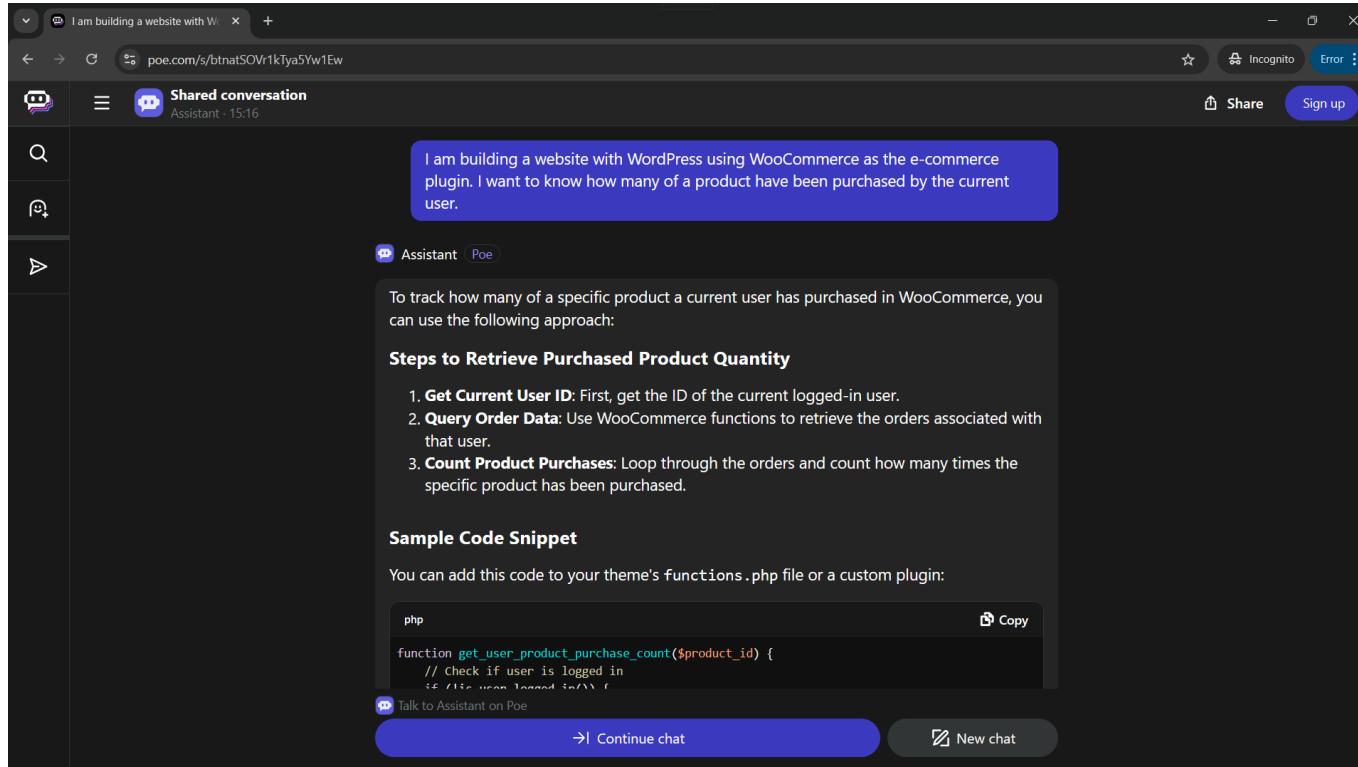
MEDIA PARTNERS

- WordPress



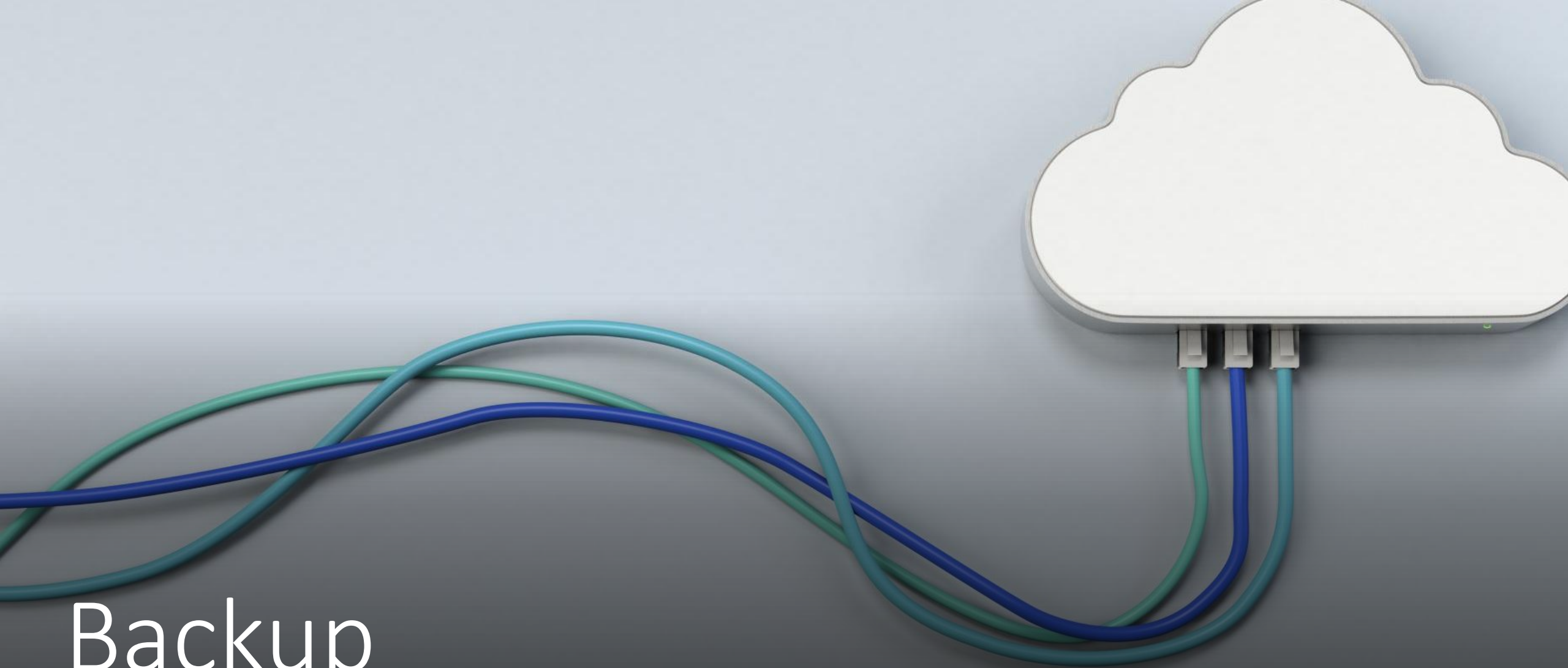
Shared Conversation on poe.com

<https://poe.com/s/btnatSOVr1kTya5Yw1Ew>





Q&A



Backup



Comparison of “Original Journey” vs “Re-enactments”

| | Original Journey | “Re-enactments” |
|---------------------|------------------------------|--|
| Date | January 2024 | 3 times in October 2024
(Draft 1, draft 2, final) |
| Platform | Poe.com | Poe.com |
| LLM | Claude (poe.com default) | Assistant (poe.com default) |
| Location of API | Standalone PHP file | functions.php |
| API’s Created | 1 | 1, 1, 2 |
| Query | Use SQL in PHP/\$wpdb | Use PHP/get_posts() |
| Calling API | XMLHttpRequest and Fetch API | Fetch API only |
| Warn about Security | No | Yes |
| Code Accuracy | Lower | Higher |
| Achieve Goal? | Yes | Yes |