

Create a webhook

Webhooks are a powerful tool for integrating your activities with external systems, enabling automated workflows and real-time data synchronization. Here's a straightforward guide to get you started:

1. Enter the Workspace

Navigate to your workspace. Once inside, look for the sidebar menu and click on "Webhooks".

2. Initiate Webhook Creation

Within the Webhooks section, find and click the "Create Webhook" button. This action will take you to the form where you can define the details of your new webhook.

3. Fill Out the Webhook Form

Now, you'll need to provide information about your webhook. Here's what you'll need to fill in:

- **Name:** Give your webhook a unique and descriptive name that helps you identify its purpose.
- **Callback URL:** Enter the URL where you want the webhook events to be sent. This should be a URL on your server ready to handle incoming HTTP requests.
- **Method:** Choose the HTTP method (typically POST) that the webhook should use to send data to your callback URL.
- **Content-Type:** Specify the content type of the data being sent. Commonly, this is `application/json`, but you might have other requirements based on your server setup.
- **Max attempts:** Decide how many times the webhook should attempt to send the event data if the first try fails.
- **Status:** Check this box to activate/deactivate the webhook.
- **Secret (Optional):** Enter a secret key for securing your webhook. This will be used to generate a signature for each request, helping you verify that requests to your callback URL are legitimate.
- **Events:** Choose the specific events you want this webhook to listen to. Your selection here defines what actions will trigger the webhook to send data to your callback URL.

5. Create the Webhook:

- After filling out the form and reviewing your settings, click the "Create" button to finalize the webhook creation process.

Secure your webhook

By entering a secret key into the **Secret** input, every delivery will automatically include a hash signature. The hash signature will appear in each delivery as the value of the `X-Signature` header.

The hash signature is generated using your webhook's secret token and the payload contents. You can use your programming language of choice to implement HMAC verification in your code. Following are some examples showing how an implementation might look in various programming languages.

PHP Example

For example, you can define the following *"verifySignature"* function:

```
function verifySignature(string $payloadContents, string $receivedSignature) {
    $secretKey = getenv('SECRET_KEY');

    $signature = hash_hmac('sha256', $payloadContents, $secretKey);

    if (!hash_equals($signature, $receivedSignature)) {
        http_response_code(500);
        echo "Could not verify the request signature.";
        exit;
    }
}
```

Then you can call it when you receive a webhook payload:

```
$payloadContents = file_get_contents('php://input');

// And we're getting the signature from an HTTP header named 'X-Signature'
$receivedSignature = isset($_SERVER['HTTP_X_SIGNATURE']) ? $_SERVER['HTTP_X_SIGNATURE'] : '';


// Now, call the function with the payload and the received signature
verifySignature($payloadContents, $receivedSignature);

// If the function doesn't exit, the signature is verified
echo "Signature verified successfully.";
```

Conclusion

Your webhook is now set up and will start listening to the events you selected, sending data to your specified callback URL as those events occur.

Ensure your callback URL is properly configured to handle incoming webhook data, and always secure your webhooks using the secret key to maintain data integrity and security.

 Mixpost

CREATE POST

Dashboard

Content

Posts

Calendar

Media Library

Templates

Configuration

Posting Schedule

Accounts

Webhooks

D

Dima Botezatu

My Team

Create webhook

Details

Name *

Callback URL *

Method

Content-Type

Max attempts

Status

Name

https://example.com/webhook

POST

application/json

1

☒ Active

Security

Secret (Optional)

Secret

Events

☐ Account Added☐ Account Updated☐ Account Deleted☐ Post Deleted

☐ Post Scheduled☐ Post Published☐ Publishing Post Failed

CREATE