

## PRIVATE PLUGINS

# Webhooks

[Copy](#)

Send a payload of merge variables to create a custom screen.

## Before you begin

Learn how to build Private Plugins [here](#). The guide below only explains how to use the "Webhook" data retrieval strategy.

[Rate Limits](#)[Authorization](#)[Set new content](#)[Get merge variable content](#)[Update existing content](#)[Troubleshooting](#)

## Rate Limits

### Request volume

You may send data to TRMNL's server up to 12x per hour. [TRMNL+ subscribers](#) may send up to 30x payloads per hour. Webhooks sent at a faster pace will receive a `429` rate limit response. To temporarily increase your rate limit during development, enable "Debug Logs" on your plugin settings page.

### Request size

You may send up to 2kb of data. [TRMNL+ subscribers](#) may send up to 5kb of data. To stay within these boundaries while also creating a data rich experience, consider using the `deep_merge` and `stream` strategies documented below.

## Authorization

TRMNL has Device API Keys, User API Keys, and Plugin Setting UUIDs. For private plugin screen generation, we'll use your Plugin Settings UUID.

This is accessible from your plugin instance's configuration form > Webhook URL field.

Webhook URL  
(Webhook strategy only) Paste this URL into your webhook integration

`https://usetrmnl.com/api/custom_plugins/9e4491c2-0f0d-4f00-8c41-0d7f5a4a` [Copy](#)

Private Plugin Webhook URL w/ UUID

[Note: you must "save" \(create\) a private plugin instance to generate a UUID and Webhook URL.](#)

## Set new content

Send a `POST` request to your Webhook URL. Put data inside a `merge_variables` node like so:

```
curl "https://usetrmnl.com/api/custom_plugins/asdfqwert1234" \
-H "Content-Type: application/json" \
-d '{"merge_variables": {"text": "You can do it!", "author": "Rob Schneider"}}' \
-X POST
```

You will see this payload inside the Your Variables dropdown of the Markup Editor.



Your variables - available inside the Markup Editor

## Get merge variable content

To fetch existing `merge_variables` from a private plugin, `GET` from the same endpoint:

```
curl "https://usetrmnl.com/api/custom_plugins/asdfqwert1234"
```

## Update existing content

If your private plugin needs to maintain state over time, for example an ever-growing todo list or a data visualization, you may prefer to send only "new" data points to your TRMNL plugin.

There are two strategies to accomplish this: `deep_merge`, and `stream`.

### Deep merge strategy

The `deep_merge` strategy combines existing key/value pairs with the new values incoming on the webhook. It's a good way to update nested data with only a few values here and there.

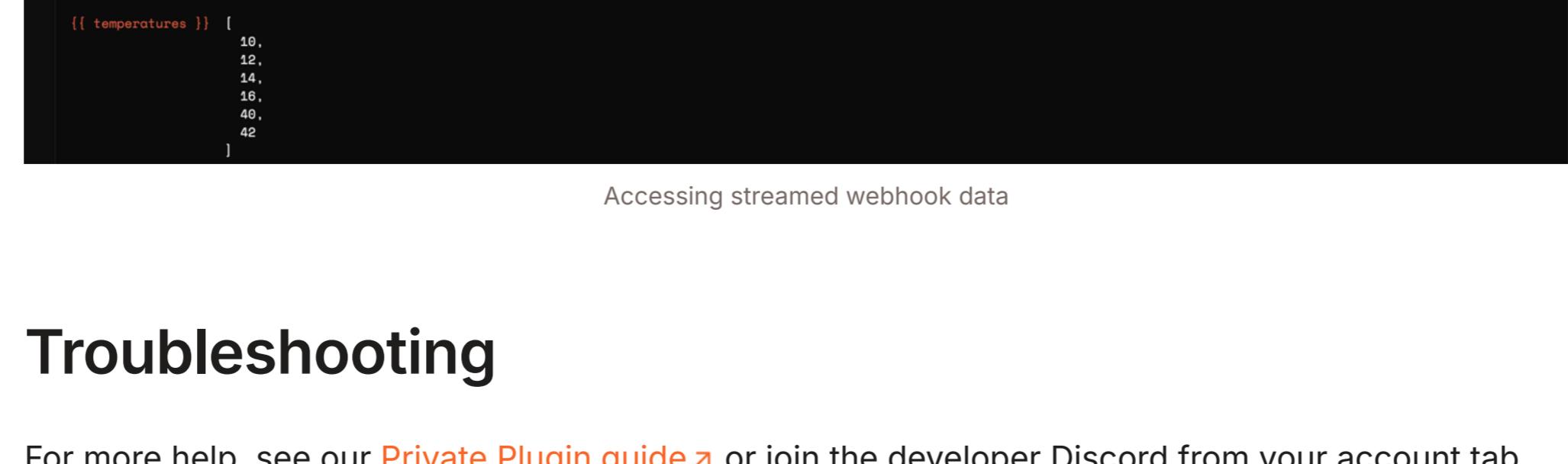
```
curl "https://usetrmnl.com/api/custom_plugins/asdfqwert1234" \
-H "Content-Type: application/json" \
-d '{"merge_variables": {"sensor": {"temperature": 42}}, "merge_strategy": "deep_merge"}' \
-X POST
```

### Stream strategy

The `stream` strategy is useful for accumulating values in arrays. Any top-level arrays are appended with the incoming values, and the `stream_limit` parameter ensures that old values drop off the arrays so they don't grow forever.

```
curl "https://usetrmnl.com/api/custom_plugins/asdfqwert1234" \
-H "Content-Type: application/json" \
-d '{"merge_variables": {"temperatures": [40, 42]}, "merge_strategy": "stream", "stream_limit": 2}' \
-X POST
```

Now you may iterate through the combined data inside your markup, for example:



Accessing streamed webhook data

## Troubleshooting

For more help, see our [Private Plugin guide](#) or join the developer Discord from your account tab.

Screen Templating (Graphics) [Previous](#)

Next Reusing Markup

Last updated 2 months ago

This site uses cookies to deliver its service and to analyze traffic. By browsing this site, you accept the [privacy policy](#).

[Accept](#)

[Reject](#)