



FlowBoost Documentation

Last Updated: February 24, 2017

Overview

FlowBoost is a product meant to enhance the functionality of Marketo flow steps, opening up an enormous range of new possibilities for data manipulation, communicating with outside platforms, or doing anything that an advanced marketer might need to do within Marketo.

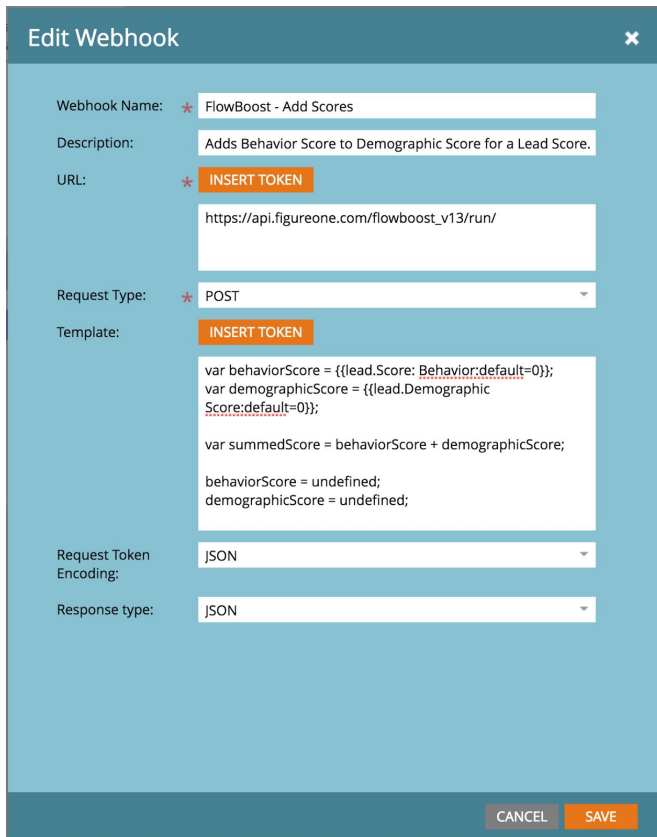
FlowBoost consists of special webhooks with combinations of code meant to run on inputs, and inputs (program tokens as well as lead tokens) that can be manipulated. Results are returned as JSON, which can be mapped to lead fields. This is meant to provide significant building blocks for advanced users, opening up possibilities that couldn't be considered before.

Webhook Setup

Create the webhook.

Edit Webhook:

- **Webhook Name:** Whatever you'd like! We preface names with "FlowBoost" so that they're easy to tell from glancing.
- **Description:** Whatever you'd like again. We use human-readable descriptions so we know what the functionality is supposed to do.
- **URL:** https://api.figureone.com/flowboost_v13/run/
- **Request Type:** POST
- **Template:** This is where you put in the JavaScript that you want to run. For Community and Standard versions, variables you define here end up being the JSON results for the webhook. There are examples of what kinds of things you can put in this section below.
- **Request Token Encoding:** JSON
- **Response type:** JSON



Edit Webhook

Webhook Name: * FlowBoost - Add Scores

Description: Adds Behavior Score to Demographic Score for a Lead Score.

URL: * INSERT TOKEN
https://api.figureone.com/flowboost_v13/run/

Request Type: * POST

Template: INSERT TOKEN

```
var behaviorScore = {{lead.Score: Behavior:default=0}};  
var demographicScore = {{lead.Demographic  
Score:default=0}};  
  
var summedScore = behaviorScore + demographicScore;  
  
behaviorScore = undefined;  
demographicScore = undefined;
```

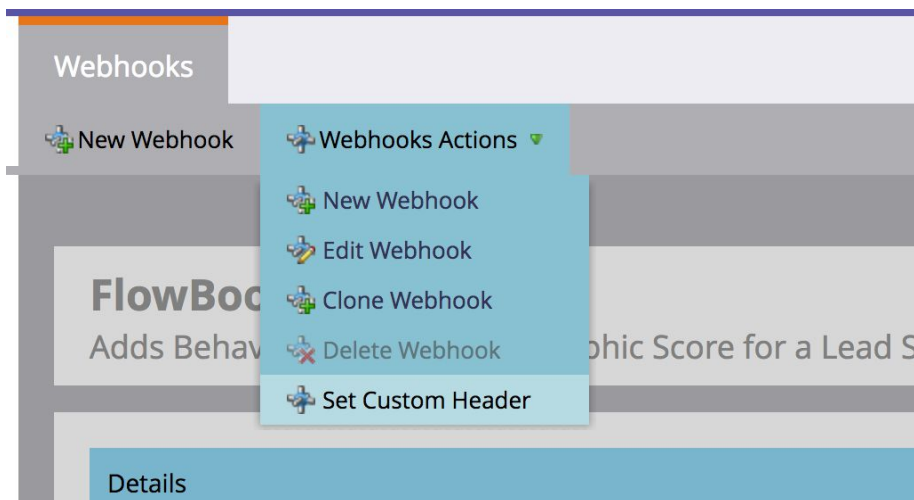
Request Token Encoding: JSON

Response type: JSON

CANCEL SAVE

Set up custom headers on the webhook:

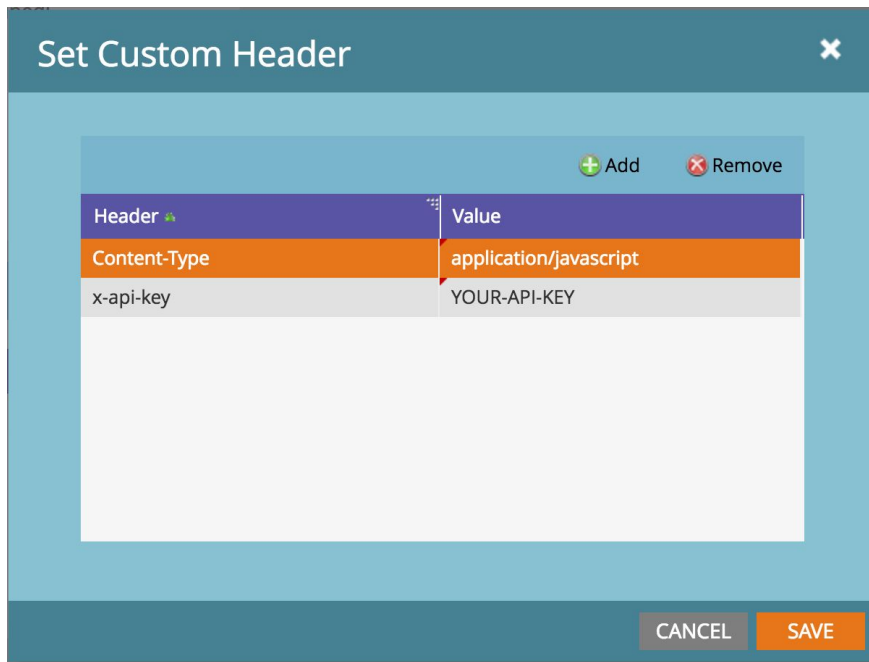
On your new FlowBoost webhook, click into “Webhooks Actions” and then “Set Custom Header.”



After that, you'll need to add in two separate values:

- Content-Type: application/javascript
- x-api-key: YOUR-API-KEY

Below is a screenshot of what this should look like:



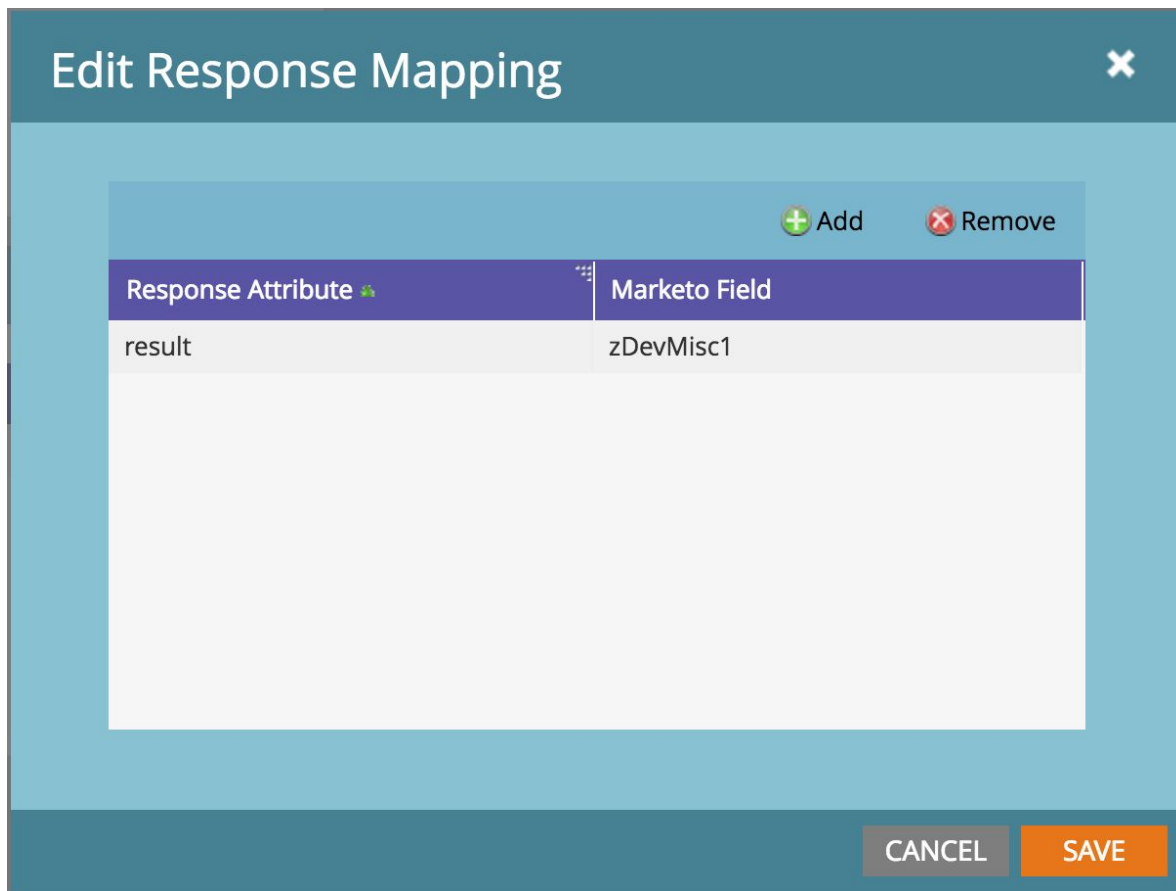
That's what you need to do to set up the webhook! You'll want to set up new Webhooks with your specific code based on what you want to do. The way that these all happen separately is through separate Payload values in the webhooks.

Response Mapping:

The responses sent back from FlowBoost will be the variables that you defined within the Payload of the webhook. For the most part, you'll want to map the response variables from FlowBoost to individual lead/contact values. You can have lots of separate responses from the FlowBoost webhook, but you don't have to map any of them to lead/contact fields.

If you do want to map a response from FlowBoost to a lead/contact field, you'll have to map the response to the lead/contact field.

For example, if you have a FlowBoost webhook that creates a variable called "result" which has a value of "Hello World," you would take that variable name ("result") and add it as the Response Attribute. If we wanted to map this response to a custom lead/contact field in Marketo called "zDevMisc1," we would have the below response mapping:



Use Cases and Examples:

The contents of the commands happen through the webhook's Payload. The variables defined within the Payload section are then returned as the response.

Return "Hello World!"

As an example, if we were interested in having "Hello World!" as the response from a webhook, the contents of the payload would be:

```
-----  
var result = "Hello World!";  
-----
```

That would return with the JSON response:

```
{"result": "Hello World!"}
```

Get a lead's email domain from its email address

In the case of wanting to get a lead's email domain from the email address, the following JavaScript as the webhook's Payload:

```
-----  
var domainMatch = {{lead.Email Address}}.match(/@(.+)/).pop();  
-----
```

If the lead's email address were eunthank@etumos.com, this webhook would return the following JSON response:

```
{"domainMatch": "etumos.com"}
```