Using Iterable handlebars to support multiple Push Planet preference centers

Last Modified on 07/01/2026 1:25 pm EDT

If you manage multiple brands or business units inside a single Iterable project, you can serve a different Push Planet preference center for each one using Iterable's handlebars conditional logic. Iterable evaluates the logic at send time and inserts the matching URL into each subscriber's email automatically, so a single Hosted Unsubscribe URL setting can power any number of preference centers.

For background on Iterable's handlebars syntax, see Iterable's Handlebars Reference: Conditional Logic Helpers.

Before you start

You'll need:

- One Push Planet preference center page already built for each brand.
- A property in Iterable you can use to identify which brand a given send belongs to. Most teams use the Message Channel ID or Message Type ID, but any handlebars-accessible property works.
- Access to your Iterable project's Project Settings.

A quick note on hosting and pricing: adding additional preference center pages under a single domain (for example, multiple paths under preferences.yourcompany.com) is included with your Push Planet plan at no extra cost, so most multi-brand setups can be supported without changing your contract. If you need each preference center on its own dedicated domain, that requires a separate domain configuration. Reach out to your account representative before you build to talk through the options and any associated cost.

Step 1: Collect your preference center URLs

Before you set up the conditional logic, gather the URL for every Push Planet preference center page you want to route to. When using a single domain, the only thing that differs from one URL to the next is the path after the domain. Note each path so you have it ready when you build the conditional in Step 3.

For example, your URLs might look like this:

- https://preferences.yourcompany.com/manage-preferences (Brand A)
- https://preferences.yourcompany.com/digioh (Brand B)
- https://preferences.yourcompany.com/pushplanet (Brand C)

If you need to confirm or change a URL path for any page, see What is my preference center URL?.

Step 2: Set up distinct Message Channels or Message Types in Iterable

In Iterable, configure a Message Channel and Message Type for each brand if you don't already have them. We recommend using a separate channel per brand so the handlebars logic in Step 3 can use the channel ID to decide which preference center URL to insert. Note the channelId (or messageTypeId) for each brand, you'll need them in the next step.

Step 3: Add the conditional logic to your Hosted Unsubscribe URL

In Iterable, go to **Project Settings → Email** and find the **Hosted Unsubscribe URL** field. Replace the static URL with a handlebars conditional that maps each channel or message type ID to the matching preference center URL. Keep ?email={{email}} (or uid={{userId}} if you use the userId as your identifier) on the end of every URL so each page can pre-fill the subscriber.

Here's an example using channelId:

https://preferences.yourcompany.com{{#ifEq channelId "12345"}}/manage-preferences{{else ifEq channelId "12346"}}/digioh{{else}}/pushplanet{{/ifEq}}?email={{email}}

In this example:

- Subscribers in channel 12345 get the Brand A preference center at /manage-preferences.
- Subscribers in channel 12346 get the Brand B preference center at /digioh.
- All other subscribers fall through to the Brand C preference center at /pushplanet.

You can swap channelId for messageTypeId, or for any other property your project uses to differentiate brands. Chain as many else ifEq branches as you need to cover every brand.

When you save the page, make sure **Use the hosted unsubscribe URL** is toggled on, otherwise Iterable will fall back to its default unsubscribe page.

Step 4: Test before you go live

Digioh recommends sending a test email from each Message Channel before beginning your full sends. Click the unsubscribe or manage preferences link in each one and confirm you land on the correct brand's preference center.

If a recipient lands on the wrong page, the most common causes are a typo in the channel or type ID, a missing ?email={{email}} parameter, or a fallback URL being matched before the specific case. Walk through the conditional in order and check each branch.

Additionally, if you encounter errors when testing, make sure that the unsubscribe URLs being generated in the emails contain a proper email or uid parameter that resolves back to a valid profile in your Iterable project.

Once your conditional is live, every send out of that Iterable project routes subscribers to the correct preference center automatically.