Integrate with LoyaltyLion

09/18/2025 2:08 pm EDT

Customer requirements

- Intro to LoyaltyLion contacts.
- LoyaltyLion to provide Bluecore with a Client ID and Secret to use during installation.

Bluecore provides

- Set up requirements for OAuth flow
- Set up custom cloud functions to ingest data
- Add a mapping to extensionrouter.go to enable routing from api.bluecore.com to the custom cloud function

Workflows

Partner workflow

- LoyaltyLion will provide Bluecore with a ClientID and Secret to use for setting up any webhook subscriptions
- Brand team may need to provide points to dollars conversion rate and any creative assets for banners (if they're going to be used).

Bluecore workflow

- Complete OAuth flow and installation
- Create a custom cloud function to fire customer patch events upon receipt of LoyaltyLion payloads
- Add a mapping to extensionrouter.go to enable routing from api.bluecore.com to the custom cloud function

Use cases

- **Use Case 1**: Include a banner in emails that displays points in dollar value on a user's account (e.g. You have $XX in your account).
- **Use Case 2**: Trigger an email reminder when points are available.

Integrations setup

Most steps are covered in the LoyaltyLion's OAuth flow page here.

1. FDE sends LoyaltyLion an email with the requested info. The most important piece is to include the install and register URLs in the OAuth Flow:
    a. Installation URL: https://api.bluecore.com/extensions/EKV3NZMLDKQ1D6YO/loyaltylion_install?token=namespace

    b. Redirect URL: https://api.bluecore.com/extensions/3FG7G9NLGY9XMTLF/loyaltylion_register

2. LoyaltyLion will then send over the Client ID and Secret. These will be exchanged in the OAuth flow for an Authorization token.

3. Use the token to set up or manage webhook subscriptions
    a. Before the webhook can be set up, the FDE may need to already have set up an HTTP address for the cloud function that will process the loyalty data.

4. Deploy a custom cloud function to parse LoyaltyLion's payloads and fire a customer_patch event with loyalty data.
    a. More information: https://developers.loyaltylion.com/api/resources/webhooks and the cloud-functions repository.

Campaign setup

Creating loyalty global widgets

1. Navigate to Global Widgets.

2. Click Create New Global Widget.

3. Write custom Jinja to specific requirements. For example, Logic to dynamically to display a version of the banner for customers who have above a certain points threshold, points to dollar conversions, etc.

4. Publish global widget and add to any templates.

Proof

Run a proof of the content. Note that the proof only shows what's in the HTML content created in earlier steps.

FAQ

Q: How does data from LoyaltyLion make it into a Bluecore email?

- A: Once set-up is complete, Bluecore will have a webhook subscription to LoyaltyLion. Whenever a customer's information gets updated, LoyaltyLion will ping the cloud function URL that was given to them during the installation process.

  From within the cloud function, Bluecore parses the payload for a customer's data and then sends a customer_patch event to the Bluecore API to update that customer's information in our customer databases.
  The customer data is then leveraged in email sends as normal.

Q: What does a payload from LoyaltyLion look like?

- A: The resource page has an example payload, although the actual webhook payload may include additional fields depending on the set up for that specific brand.

Q: How do we handle errors?

- A: Errors are logged and documented on the Google Cloud Platform. The FDE on-call receives notifications for failures and will conduct initial triage. There are no automatic retries on failed runs. Whenever a customer's next update comes in (e.g. a customer making their next purchase or when their points go from pending to current), we will receive that next update and "catch up" missed data.

  System failures on the webhook send an email notification tofdeserviceaccount@bluecore.com.