Events API

Bluecore's events API can be used to capture data from your mobile app or other sources, in the same way that our JavaScript integration captures data from your website. You may POST various events to identify/update customers as well as record commerce events/behaviors, including custom events.

Endpoint

```
$ POST https://api.bluecore.app/api/track/mobile/v1
```

> ⓘ As of March 2024, the endpoint for API calls changed from `api.bluecore.com` to `api.bluecore.app`. Pre-existing integrations do NOT require any action as their URLs will be automatically re-routed to new endpoint. New integrations must be configured with the `.app` endpoint address.

Parameters

Field	Description
distinct_id *required*	A unique identifier for a given customer. This can be an email address when it is available, a customer ID in your database, a device id, or UUID that you generate and consistently use for the browsing customer. • **IMPORTANT**: For each new non-email `distinct_id` that you use for a customer, you must fire an "identify" event to link it to the customer email address. See *Linking identifiers to email*.
customer *(required for Purchase, Identify, Unsubscribe, and Opt-in events)*	An object that at minimum contains email, and optionally contains additional customer attributes and values to associate with this customer. • **Caution**: Any customer attributes included in these calls (e.g. `first_name`) will be permanently created within Bluecore if they do not exist already. Therefore to avoid excess or redundant attributes/columns, only send attributes critical to your campaigns, and ensure you confirm the exact syntax of attribute names. • **Note:** All customer attributes must be lowercase. • **Note:** Customer attributes can't include a space or dash (`-`). Instead, they should use an underscore (`_`). • **Note:** If using boolean attributes, they must use boolean values. For example: `"boolean_attribute": true`
products *(required for all commerce events except Search)*	An array of objects representing the product(s) being interacted with, where each object contains the product ID. Note that IDs must be consistent with those captured via the website and any other incoming product data. • ***NOTE:*** To update your product catalog, please reach out to your CSM.

Field	Description
search_term *(required for Search events)*	The keyword/phrase searched by the customer, or the category being browsed by the customer.
order_id *(required for Purchase events)*	Your unique ID for a given order, e.g. Order Confirmation number, Order ID number, Purchase Confirmation code, etc. If not provided, a random `order_id` will be generated.
total *(required for Purchase events)*	The sum of the purchases made in this order. Inclusion of tax and shipping charges is optional; the approach should be consistent with how other Purchase data is captured (e.g. via Purchase Pixel).
token (required)	Also known as your Bluecore "namespace."
client *(optional)*	You may use this field to populate the version of your mobile app, or any other data that you'd like to differentiate on when building Audiences
device*(optional)*	You may use this field to populate the customer's device e.g. "iPadOS 15.0 iPad11,6."

Types of calls

Capturing commerce events and behaviors

You may make calls to represent these standard events/behaviors:
- `viewed_product`
- `search` (used for keyword searches as well as category browse)
- `add_to_cart`
- `remove_from_cart`
- `wishlist`
- `purchase`

> ⓘ To update your product catalog, please reach out to your CSM.

Example - a product view event

```
{
  "event": "viewed_product",
  "properties": {
    "products": [
        {"id": "1234"}
    ],
    "distinct_id": "xyz@example.com",
    "token": "bluestore",
    "client": "4.0",
    "device": "mobile/iPad"
  }
}
```

Example - a search event

```
{
   "event": "search",
   "properties": {
      "search_term": "logo design",
      "distinct_id": "xyz@example.com",
      "token": "bluestore",
      "client": "4.0",
      "device": "mobile/iPad"
   }
}
```

Example - an add to cart event

```
{
   "event": "add_to_cart",
   "properties": {
      "products": [
         {"id": "1234"}
      ],
      "distinct_id": "1424532133a16a-07943fba1-1960426e-2c600-1424532133c98",
      "token": "bluestore",
      "client": "4.0",
      "device": "mobile/iPad"
   }
}
```

Example - a remove from cart event

```
{
   "event": "remove_from_cart",
   "properties": {
      "products": [
         {"id": "1234"},
         {"id": "5678"}
      ],
      "distinct_id": "1424532133a16a-07943fba1-1960426e-2c600-1424532133c98",
      "token": "bluestore",
      "client": "4.0",
      "device": "mobile/iPad"
   }
}
```

Example - a purchase event

```
{
  "event": "purchase",
  "properties": {
    "products": [
        {"id": "1234"},
        {"id": "5678"}
    ],
    "total": 400.20,
    "order_id": "ABCDE12345",
    "distinct_id": "xyz@example.com",
    "token": "bluestore",
    "client": "4.0",
    "device": "mobile/iPad",
    "customer": {
        "email": "xyz@example.com"
    }
  }
}
```

Example - an add to wishlist event

```
{
  "event": "wishlist",
  "properties": {
    "products": [
        {"id": "1234"}
    ],
    "distinct_id": "1424532133a16a-07943fba1-1960426e-2c600-1424532133c98",
    "token": "bluestore",
    "client": "4.0",
    "device": "mobile/iPad"
  }
}
```

Creating and updating customers

You may make calls to create a new customer with optional attributes, or update attribute values for an existing customer.

Example - creating / updating a customer with attributes

```
{
  "event": "customer_patch",
  "properties": {
    "customer": {
        "first_name": "X",
        "last_name": "Yz"
    },
    "distinct_id": "1424532133a16a-07943fba1-1960426e-2c600-1424532133c98",
    "token": "bluestore"
  }
}
```

ⓘ All customer attributes must be lowercase.

ⓘ Customer attributes can't include a space or dash (-). Instead, they should use an underscore

(_).

> ⓘ If using boolean attributes, they must use boolean values. For example `"boolean_attribute": true`

Linking identifiers to emails

Across call types, the `distinct_id` parameter represents a unique customer identifier. This can be an email address when it is available, a customer ID in your database, a device id, or UUID that you generate and consistently use for the browsing customer.

IMPORTANT: For each new non-email distinct_id that you use for a customer, you must fire an "identify" event to link this identifier to the customer email address. This enables you to tie together all of a customer's behaviors for a holistic view. Note this means you can capture behaviors for a customer prior to knowing their email, and have those behaviors intact and linked once you do obtain the email address (e.g. upon login or registration).

Example - linking a device ID with an email address

```
{
   "event": "identify",
   "properties": {
      "distinct_id": "1424532133a16a-07943fba1-1960426e-2c600-1424532133c98",
      "token": "bluestore",
      "client": "4.0",
      "device": "mobile/iPad",
      "customer": {
         "email": "xyz@example.com"
      }
   }
}
```

Opting in and unsubscribing customers

You may make calls to opt-in or unsubscribe an email address.

Example - opting-in a customer email

```
{
   "event": "optin",
   "properties": {
      "distinct_id": "xyz@example.com",
      "email": "xyz@example.com",
      "token": "bluestore",
      "client": "4.0",
      "device": "mobile/iPad"
   }
}
```

Example - unsubscribing a customer email

```
{
    "event": "unsubscribe",
    "properties": {
        "distinct_id": "xyz@example.com",
        "email": "xyz@example.com",
        "token": "bluestore",
        "client": "4.0",
        "device": "mobile/iPad"
    }
}
```

Server response

At this time, all calls will result in a 202 response (we are updating this!); please see Testing and QA for further guidance.

Testing and QA

The best way to test your calls to the Events API is to create example calls with example customers, and leverage our Audience Builder to verify successful capture. (Bluecore does not have traditional QA or dev environments). You may also send your test calls that you have made to Product Support (support@bluecore.com) and ask them to verify successful capture. Please ask your Customer Success Manager for more detail as needed.

Differentiating mobile events

Within Audience Builder, events that are captured via these API calls are differentiable from events captured via the JS integration or file feeds, using the `is_mobile` attribute.

Custom events

Generally, Bluecore can create custom events (e.g. `sweepstakes_signup`) that can be captured via any of your incoming data sources (website, mobile app, feeds, etc.) Please speak with your CSM about any desired custom events.