



## Getting Started

- What is Segment?
- [How Segment Works](#)
- Getting Started Guide
- A Basic Segment Installation
- Planning a Full Installation
- A Full Segment Installation
- Sending Data to Destinations
- Testing and Debugging
- What's Next
- Use Cases

## Guides

### Connections

### Unify

### Engage

### Privacy

### Protocols

### Segment App

### API

## Partners

## Glossary

## Config API

## Help

published event-triggered journeys.



### Public Beta

Event-Triggered Journeys is in public beta, and Segment is actively working on this feature. Some functionality may change before it becomes generally available. Event-Triggered Journeys is not currently HIPAA eligible.

## Overview

Event-triggered journeys help you create a responsive approach for time-sensitive use cases, like cart abandonment campaigns and transactional messages.

Where [audience-based journeys](#) activate based on aggregated conditions, event-triggered journeys respond instantly to individual events, delivering personalized experiences based on the full context of each event.

Opt for an event-triggered journey in situations like these:

- When campaigns require real-time action in response to user behavior.

- For transactional messages (like receipts and confirmations) that require specific event-based triggers.
- In abandonment campaigns where a follow-up is needed if a corresponding completion event doesn't occur.

## Build an event-triggered journey



### Before you begin

Before you start building an event-triggered journey, make sure that you've enabled all [destinations](#) you plan to send data to and that the events you want to use as triggers are already available in your Segment workspace.

To set up an event-triggered journey:

1. In your Segment workspace, navigate to **Engage > Journeys**, then click **+ Create journey**.
2. On the **Create journey** page, select **User performs an event**, then click **Next**.
3. Give your new journey a name and, optionally, a description.
4. Select entry event:
  - Choose the event that will trigger user entry into the journey.
  - (Optional) Use an audience filter to restrict entry to users who are already part of a specific audience when they perform the triggering event.
  - (Optional) Apply filters based on event property values to refine entry conditions. For example, enter only if {property} = value A, value B, or value C.
5. Configure entry rules:
  - **Re-enter every time event occurs** (default): Users enter the journey each time they trigger the specified event.
  - **Enter one time**: Users enter the journey once only, regardless of repeated event triggers.
6. If you chose **Re-enter every time event occurs** in Step 5, select a [unique identifier](#).
7. Build your journey using logical operators.
8. Configure event delivery to destinations by selecting a destination or setting up a custom destination function.
9. Review the contextual payload that Segment will send to your destination(s).
10. After you've finished setting up your journey, click **Publish**, then click **Publish** again in the popup.

## Send data to downstream destinations

When a journey instance reaches a **Send to Destination** step, you can configure how data is sent to your desired destination. This step allows you to define where the data goes, what actions are performed, and how information is mapped, giving you control over the integration. Event-Triggered Journeys currently supports all [Actions Destinations](#).

For other destinations or more complex logic, you can use [Destination Functions](#).

### Configure the Destination Send Step

#### Select a Destination

Choose the destination where you want to send data. Currently, only [Actions Destinations](#) and [Destination Functions](#) are supported.

#### Choose an Action

Specify the action to take within the selected destination. For example, you might update a user profile, trigger an email, or log an event.

### Define the Event Name

Add a descriptive event name to send to your destination.

### Define the Payload Attributes

- The **journey context** provides a set of attributes from the entry event or events used in the Hold Until operator that can be included in the payload.
- You may also add a user's profile traits to the destination payload.
- Review the available attributes and decide which ones to include in your data send.

### Map Attributes to Destination Keys

- Use the mapping interface to link payload attributes to the appropriate keys required by the destination.
- For example, map `user_email` from the journey context to the `email` field expected by the destination.

### Test the Integration

- Send a **test event** to validate the configuration.
- Ensure that the data is received correctly by the destination and mapped as expected.

When a journey reaches this step, the Segment prepares and sends the payload based on your configuration. The integration ensures compatibility with the selected destination's API, allowing seamless data transfer and execution of the specified action.

## Journey setup configuration options

Event-Triggered Journeys includes advanced options to help you tailor journey behavior and customize data delivery to downstream destinations.

### Unique identifiers

Unique identifiers in event-triggered journeys help you manage multiple journey instances when a user triggers the same event more than once.

When you select **Re-enter every time event occurs** when you create an event-triggered journey, you can choose an event property as a unique identifier. Selecting this option does two things:

- It creates a separate journey instance for each unique identifier value, allowing multiple instances to run in parallel for the same user.
- It ensures that any follow-up events link back to the right journey instance, preserving context for tracking and personalization.

For example, in an abandonment journey, suppose a user starts two applications (like `application_started`), each with a different `application_id`. By setting `application_id` as the unique identifier, Segment can match follow-up events (like `application_completed`) to the correct application journey. As a result, each journey instance only receives the completion event for its specific application.

### Notes and limitations

- **Supported destinations:** Only Actions Destinations in the Segment catalog are supported.
- **Data mapping:** Ensure all required keys for the destination are properly mapped to avoid errors.

## Best practices

Follow the best practices in this table to optimize your event-triggered journeys:

RECOMMENDATION	DETAILS
----------------	---------

RECOMMENDATION	DETAILS
Use specific event filters	When you configure entry events, apply precise filters based on event property values to refine which users enter the journey. This helps target specific user actions and improves the journey's relevance.
Use unique identifiers	If a journey allows users to enter multiple times, set a unique identifier to track each instance accurately. Using an identifier like <code>application_id</code> ensures that follow-up events stay associated with the right journey instance.
Preview payloads before publishing	Review the journey payload to verify that it includes all necessary context from the triggering event. This helps confirm that the data reaching destinations matches your campaign needs.
Test journey after publishing	Consider setting up a live test right after publishing to confirm that the journey behaves as expected and that data flows correctly to destinations.

## Working with Event-Triggered Journeys

Segment built Event-Triggered Journeys to respond instantly to events, offering real-time capabilities with a few considerations in mind.

**Entry event requirements:** The entry event you use must already exist in your Segment workspace for it to appear as a selection in journey setup. Make sure that you've already created the event before setting up your journey.

**Event property filters:** You can filter event properties using the `equals` or `equals any of` operators. When you apply multiple conditions, filters operate with **AND** logic, meaning all conditions must be true for the event to trigger entry into the journey.

**Audience filtering:** You can only use active, pre-existing audience records as filters. For more complex filtering, like specific profile traits or multiple audiences, first [create the audience](#) in **Engage > Audiences**, then apply it as a filter once it's live.

**Destination options:** While Event-Triggered Journeys support all [actions-based destinations](#) and Destination Functions, you can only add one destination per Send to Destination step. If you need to send to multiple destinations, you can use multiple Send to Destination steps.

**Event payload structure:** Each payload sent to a destination includes a unique key to identify the specific send step within the journey, rather than the journey instance itself. You can also set a custom event name to make it easier to identify the specific event instance you want to track in your destination.

**Editing and versioning:** After you publish an event-triggered journey, you won't be able to edit it. To modify a journey, create a new journey.

**Real-time delivery:** Event-Triggered Journeys aim for an expected delivery time of under 5 minutes from the moment an event is performed to when the payload reaches the destination, assuming there is no delay step in the journey. However, external factors outside of Segment's control may occasionally introduce latency.

## Use Cases

Event-Triggered Journeys can power a variety of real-time, personalized experiences. This section details some common scenarios to help you see how they might work in practice.

### Real-time event forwarding

Suppose you want to instantly send a personalized message whenever a user completes a specific action on your site, like filling out a form or subscribing to a service. With Event-Triggered Journeys, you can configure the journey to trigger each time this entry event occurs. Segment will forward the event data, including all relevant details, to your connected destination in real-time.

### Real-time abandonment Campaigns

Imagine you’re running an e-commerce site and want to follow up with users who start the checkout process but don’t complete it within a certain timeframe. You can create an event-triggered Journey to watch for abandonment cases like these.

Start by setting the `checkout_started` event as the trigger and specify a unique identifier like `session_id` to track each user’s journey instance. Then, configure the journey to check for the `purchase_completed` event within a defined window (for example, 1 hour). If the user doesn’t complete the purchase, the journey can automatically send a nudge to encourage them to finish their order.

### Personalized follow-up Messages

Say you want to follow up with users after they engage with specific content, like downloading an e-book or watching a demo video. Event-Triggered Journeys can help you send timely, personalized messages based on these interactions.

To do this, set the entry event to `content_downloaded` or `video_watched` and configure the journey to send a follow-up email. You could even personalize the email with details from the triggering event, like the content title or timestamp, by configuring your destination payload to enrich the message with event-specific context.

This page was last modified: 19 Dec 2024

### Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

[Visit our Support page](#)

### Help improve these docs!

[Edit this page](#)

[Request docs change](#)

### Was this page helpful?

☐ Yes

☐ No

### Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

[Request Demo](#)

or

[Create free account](#)

[Privacy](#)

[Terms](#)

[Website Data Collection Preferences](#)

