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

Cordial (Actions) provides the following benefits over the classic Cordial destination:

- **Transparent data mapping**. The Classic Cordial destination receives data from Segment as is. The Cordial backend then converts those Segment events to Cordial's format and clients have limited control over this conversion. The Cordial (Actions) destination allows clients to fully define their own mappings of Segment events, making sure they receive data structured specifically for their needs.
- **Only sends the data you need**. With Cordial (Actions) you can define only those destination actions and mappings that make sense for your use cases, while Cordial Classic always sends four predefined API calls: identify, track, group, and page.
- **Sends Engage components to Cordial**. With Cordial (Actions) it's possible to define action mappings that will send audiences and user computed traits defined in the Segment Engage platform to Cordial.

Getting started

To enable destination actions to connect to Cordial:

1. Make sure you have your Cordial API Key. To create a new API key, navigate to the account settings menu

and select **API Keys**. Learn more.

2. From the Segment web app, click **Connections > Catalog**.
3. Search for the **Cordial (Actions)** destination item in the left navigation, and click it.
4. Click **Configure Cordial**.
5. Select the Source you want to connect to Cordial (Actions).

Destination Settings

SETTING	DESCRIPTION
API Key	*Required.* Your Cordial API Key
Endpoint	*Required.* Cordial API endpoint. Leave default, unless you've been provided with another one. See more details
User ID attribute key	Cordial string unique attribute key to store Segment User ID in (e.g. `segment_id`)

Available Actions

Build your own Mappings. Combine supported triggers with the following Cordial-supported actions:

> **Mapping limits per destination**
>
> Individual destination instances have support a maximum of 50 mappings.

- Add Contact to List
- Remove Contact from List
- Create Contactactivity
- Upsert Contact
- Upsert Order
- Remove Product from Cart
- Add Product to Cart
- Merge Contacts

Add Contact to List

Add contact to a list. If the list does not exist in Cordial it will be created.

Add Contact to List is a **Cloud** action. The default Trigger is: `type = "group"`

Click to show / hide fields

FIELD	DESCRIPTION
User Identities	Type: `OBJECT` An ordered list of contact identifiers in Cordial. Each item in the list represents an identifier. For example, `channels.email.address -> userId` and/or `customerId -> traits.customerId`. If a contact is found using the identifiers it is updated, otherwise a new contact is created.
Group ID *	Type: `STRING` Segment group id. Required.

FIELD	DESCRIPTION
List name	Type: `STRING` Cordial list name. Optional. If list name is empty, the name of the list will be set to segment_[groupId].

Remove Contact from List

Remove Contact from Cordial List

Remove Contact from List is a **Cloud** action.

Click to show / hide fields

FIELD	DESCRIPTION
User Identities	Type: `OBJECT` An ordered list of contact identifiers in Cordial. Each item in the list represents an identifier. For example, `channels.email.address -> userId` and/or `customerId -> traits.customerId`. If a contact is found using the identifiers it is updated, otherwise a new contact is created.
Group ID *	Type: `STRING` Segment group id. Required.

Create Contactactivity

Create a new contact activity.

Create Contactactivity is a **Cloud** action. The default Trigger is: `type = "track" or type = "page"`

Click to show / hide fields

FIELD	DESCRIPTION
User Identities	Type: `OBJECT` An ordered list of contact identifiers in Cordial. Each item in the list represents an identifier. For example, `channels.email.address -> userId` and/or `customerId -> traits.customerId`. If a contact is found using the identifiers it is updated, otherwise a new contact is created.
Event name *	Type: `STRING` Event name. Required.
Event timestamp	Type: `DATETIME` Event timestamp. Optional. Date format is ISO 8601 standard. If empty, the request upload time will be used.
Event properties	Type: `OBJECT` An object of additional event attributes. Optional.
Event context	Type: `OBJECT` Event context as it appears in Segment. Optional. We use context to capture event metadata like sender ip and device info.

Upsert Contact

Create or update a contact in Cordial.

Upsert Contact is a **Cloud** action. The default Trigger is: `type = "identify"`

Click to show / hide fields

FIELD	DESCRIPTION

FIELD	DESCRIPTION
User Identities	Type: `OBJECT` An ordered list of contact identifiers in Cordial. Each item in the list represents an identifier. For example, `channels.email.address -> userId` and/or `customerId -> traits.customerId`. If a contact is found using the identifiers it is updated, otherwise a new contact is created.
Contact Attributes	Type: `OBJECT` Contact attributes to update. Optional. Contact attributes must exist in Cordial prior to updating. Attributes that do not exist in Cordial will be ignored. Complex attribute types to be mapped via dot notation, for example, `cordialPerson.first_name -> traits.segmentPerson.firstName`, `cordialPerson.last_name -> traits.segmentPerson.lastName`. Segment trait address can be mapped directly to geo Cordial attribute: `geo_cordial_attribute -> traits.address`.

Upsert Order

Upserts order to Cordial

Upsert Order is a **Cloud** action. The default Trigger is: `event = "Order Completed" or event = "Order Updated" or event = "Order Refunded" or event = "Order Cancelled"`

Click to show / hide fields

FIELD	DESCRIPTION
User Identities	Type: `OBJECT` An ordered list of contact identifiers in Cordial. Each item in the list represents an identifier. For example, `channels.email.address -> userId` and/or `customerId -> traits.customerId`. If a contact is found using the identifiers it is updated, otherwise a new contact is created.
Order ID *	Type: `STRING` Internal identifier of an order
Order purchase date *	Type: `DATETIME` Order purchase date
Order status *	Type: `STRING` Order status (e.g. completed/cancelled/returned)
Order total *	Type: `NUMBER` Order total amount
Order properties	Type: `OBJECT` Additional order properties (e.g. affiliation/tax/revenue)
Order items *	Type: `OBJECT` Order items

Remove Product from Cart

Removes product from Cordial contact cart

Remove Product from Cart is a **Cloud** action. The default Trigger is: `type = "track" and event = "Product Removed"`

Click to show / hide fields

FIELD	DESCRIPTION
User Identities	Type: `OBJECT` An ordered list of contact identifiers in Cordial. Each item in the list represents an identifier. For example, `channels.email.address -> userId` and/or `customerId -> traits.customerId`. If a contact is found using the identifiers it is updated, otherwise a new contact is created.
Product ID *	Type: `STRING` Internal identifier of a product

FIELD	DESCRIPTION
Quantity *	Type: `INTEGER` Quantity of a product

Add Product to Cart

Add product to Cordial contact cart

Add Product to Cart is a **Cloud** action. The default Trigger is: `type = "track" and event = "Product Added"`

Click to show / hide fields

FIELD	DESCRIPTION
User Identities	Type: `OBJECT` An ordered list of contact identifiers in Cordial. Each item in the list represents an identifier. For example, `channels.email.address -> userId` and/or `customerId -> traits.customerId`. If a contact is found using the identifiers it is updated, otherwise a new contact is created.
Product ID *	Type: `STRING` Internal identifier of a product
SKU *	Type: `STRING` SKU of a product
Quantity *	Type: `INTEGER` Quantity of a product
Category *	Type: `STRING` Category of a product
Name *	Type: `STRING` Name of a product
Description	Type: `STRING` Description of a product
Price	Type: `NUMBER` Price of a product
URL	Type: `STRING` URL of a product
Image URL	Type: `STRING` Image of a product
Properties	Type: `OBJECT` Properties of a product (e.g brand, color, size)

Merge Contacts

Merge contacts in Cordial.

Merge Contacts is a **Cloud** action. The default Trigger is: `type = "alias"`

Click to show / hide fields

FIELD	DESCRIPTION

FIELD	DESCRIPTION
New Segment User ID	Type: `STRING` New Segment User ID value
New Segment Anonymous ID	Type: `STRING` New Segment Anonymous ID value
Previous Segment ID *	Type: `STRING` Previous Segment User or Anonymous ID value

Migration from the classic Cordial destination

User Identities Mappings

Every Cordial destination action needs to be invoked with data identifying a Cordial contact. To identify a contact every destination action has a `User Identities` field, which maps Segment event fields to Cordial contact identifiers. Each entry in the list represents a contact identifier and how it maps from a Segment event. For example, `channels.email.address <- userId` or `customerId <- traits.customerId`. At least one identifier should be valid, otherwise the contact won't be identified and the request will be ignored.

Typically, the `User Identities` field maps the Segment events `userId` field to the Cordial secondary identifier field. For example, if Segment's `userId` field is known to be an email, the mapping will be `channels.email.address <- userId` , meaning the value of `userId` will be sent as `channels.email.address` to Cordial.

Updating contacts

If you plan to create and update contacts in Cordial, define the `upsertContact` destination action. In addition to the `User Identities` field, the action defines the `Contact Attributes` field. This field defines an exclusive set of attributes that will be updated in a contact. Typically, you map them from Segment traits. For example, `customerId <- traits.customerId`. For the Cordial Classic destination, these mappings are stored in Cordial's database. In the Cordial (Actions) destination, they become explicit in the `upsertContact` destination action mappings.

Sending events

To send events, define the `createContactactivity` destination action. In addition to the `User Identities` field, additional fields such as `Event name`, `Event timestamp`, and `Event properties` may be defined. Refer to documentation for each field when defining the destination action.

Adding users to and removing from lists

If you plan to segment users in Cordial, make sure you define the `addContactToList` and `removeContactFromList` destination actions. Both actions require the Segment group ID. `addContactToList` optionally accepts a list name.

Although optional, you should consider the list name as a required option, because it simplifies segmenting contacts in Cordial based on lists. Lists without a name are called following the `segment_[groupId]` pattern.

This page was last modified: 30 Aug 2023

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.

Your work e-mail

Request Demo

or

Create free account