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

This destination is maintained by Userlist. For any issues with the destination, contact the Userlist Support team.

Getting Started

1. From the Segment web app, click **Catalog**.
2. Search for “Userlist” in the Catalog, select it, and choose which of your sources to connect the destination to.
3. In the Userlist Segment destination settings, enter your Userlist “Push API Key”. You can find this key in your Userlist Push API settings.

> ℹ
>
> **NOTE:** The Userlist Destination does not support tracking anonymous users, and returns a 400 error if you send `track` or `group` call for unidentified users. To prevent this, make sure you make an Identify call before you make Track or Group calls. You can also disregard this error if you sent calls you do not intend Userlist to process.

Identify

If you're not familiar with the Segment Specs, take a look to understand what the Identify method does. An example call would look like:

```
analytics.identify('userId123', {
  email: 'john.doe@example.com',
  name: 'John Doe',
  role: 'Owner',
  createdAt: '2019-03-21T12:12:54.735+01:00'
});
```

Identify calls will be sent to Userlist as user records. If the `userId` is already known, it'll update the user record, otherwise it'll create a new one.

Here's how Segment fields map to Userlist users:

SEGMENT FIELD	USERLIST FIELD	DESCRIPTION
`userId`	`identifier`	The unique identifier for this user.
`traits`	`properties`	Additional properties describing the user.
`traits.email`	`email`	The user's email address.
`traits.createdAt`	`signed_up_at`	The time when the user was created.

Track

If you're not familiar with the Segment Specs, take a look to understand what the Track method does. An example call would look like:

```
analytics.track('Project created', {
  projectName: 'Party Planning'
});
```

Track calls will be sent to Userlist as a new event. You may send additional properties to describe the event in more detail. Both the event name and additional properties will be stored with the event and normalized to snake case (`project_created` and `project_name`) automatically within Userlist.

To associate an event with both a user and a company, please include the company's identifier as `context.groupId` on the Track call:

```
analytics.track(
  'Project created',
  {
    projectName: 'Party Planning'
  },
  {
    context: {
      groupId: 'companyId123'
    }
  }
);
```

Group

If you're not familiar with the Segment Specs, take a look to understand what the Group method does. An example call would look like:

```
analytics.group('companyId123', {
  name: 'Segment'
});
```

SEGMENT FIELD	USERLIST FIELD	DESCRIPTION
`groupId`	`identifier`	The unique identifier for this company.
`traits`	`properties`	Additional properties describing the user.
`traits.name`	`name`	The company's name.
`traits.createdAt`	`signed_up_at`	The time when the user was created.

Group calls will be sent to Userlist as company records. If the `groupId` is already known, it'll update the company record, otherwise it'll create a new one.

Userlist supports custom properties to describe the relationship between users and groups (such as the user's role in a company). You can pass relationship properties in a Group call by using Userlist specific extensions.

```
analytics.group(
  'companyId123',
  {
    name: 'Segment'
  },
  {
    integrations: {
      Userlist: {
        extensions: {
          relationship: {
            properties: {
              role: 'owner'
            }
          }
        }
      }
    }
  }
);
```

Engage

You can send computed traits and audiences generated using Engage to this destination as a **user property**. To learn more about Engage, schedule a demo.

For user-property destinations, an identify call is sent to the destination for each user being added and removed. The property name is the snake_cased version of the audience name, with a true/false value to indicate membership. For example, when a user first completes an order in the last 30 days, Engage sends an Identify call with the property `order_completed_last_30days: true`. When the user no longer satisfies this condition (for example, it's been more than 30 days since their last order), Engage sets that value to `false`.

When you first create an audience, Engage sends an Identify call for every user in that audience. Later audience syncs only send updates for users whose membership has changed since the last sync.

Real-time to batch destination sync frequency

Real-time audience syncs to Userlist may take six or more hours for the initial sync to complete. Upon completion, a sync frequency of two to three hours is expected.

Settings

Segment lets you change these destination settings from the Segment app without having to touch any code.

SETTING	DESCRIPTION

SETTING	DESCRIPTION
API Key *(required)*	`string`. Please enter the token available in your Userlist Push API settings.

This page was last modified: 11 Sep 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.

Your work e-mail

Request Demo

or

Create free account