



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

[SlicingDice](#) is an all-in-one data warehouse. It's a fully managed cloud data warehouse with optional built-in tools for data integration, exploration, visualization and machine learning.

This destination is maintained by SlicingDice. For any issues with the destination, [contact the SlicingDice Support team](#).

Getting Started

- 1 Login into your SlicingDice's Control Panel account to create a connection between SlicingDice and Segment.
- 2 Follow the [Connecting to Segment guide](#) available in SlicingDice documentation to create and allow this connection.

Page

If you're not familiar with the Segment Specs, take a look to understand what the [Page method](#) does. An example call would look like:

```
analytics.page()
```

Page calls will save your data in SlicingDice using the following columns:

- `{source_name}-pages-page-event`: contains the `name` attribute of the call
- `{source_name}-pages-{attribute_name}`: contains the `properties` attributes of the call, saving each one as an individual column
- `{source_name}-pages-context-{attribute_name}`: contains the `context` attributes of the call, saving each one as an individual column

Screen

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

```
[[SEGAnalytics sharedAnalytics] screen:@"Home"];
```

Screen calls will be sent to SlicingDice using the following columns:

- `{source_name}-screens-screen-event`: contains the `name` attribute of a Screen call
- `{source_name}-screens-{attribute_name}`: contains the `properties` attributes of a Screen call, saving each one as an individual column
- `{source_name}-screens-context-{attribute_name}`: contains the `context` attributes of a Screen call, saving each one as an individual column

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'  
});
```

Identify calls will be sent to SlicingDice using the following columns:

- `{source_name}-identifies-user-id`: contains the `"userId"` attribute of an Identify call
- `{source_name}-identifies-{attribute_name}`: contains the `"traits"` attributes of an Identify call, saving each one as an individual column
- `{source_name}-identifies-context-{attribute_name}`: contains the `"context"` attributes of an Identify call, saving each one as an individual column

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('Clicked Login Button')
```

Track calls will be sent to SlicingDice using the following columns:

- `{source_name}-track-event`: contains the `"event"` attribute of a Track call
- `{source_name}-track-{attribute_name}`: contains the `"properties"` attributes of a Track call, saving each one as an individual column
- `{source_name}-track-context-{attribute_name}`: contains the `"context"` attributes of a Track call, saving each one as an individual column

Observation: if your context attribute has nested attributes, your column names will follow this pattern:

`{source_name}-track-context-{attribute_name}{nested_attribute_name}`

Groups

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("0e8c78ea9d97a7b8185e8632", {
  name: "Initech",
  industry: "Technology",
  employees: 329,
  plan: "enterprise",
  "total billed": 830
});
```

Group calls will be sent to SlicingDice using the following columns:

- `{source_name}-groups-group-id`: contains the `groupId` attribute of a Group call
- `{source_name}-groups-{attribute_name}`: contains the traits attributes of a Group call, saving each one as an individual column
- `{source_name}-groups-context-{attribute_name}`: contains the "context" attributes of a Group call, saving each one as an individual column

Observation: Group calls will also be sent to SlicingDice using the following [Account](#) columns that holds the latest state of a group.

- `{source_name}-accounts-group-id`: contains the `groupId` attribute of a Group call
- `{source_name}-accounts-{attribute_name}`: contains the traits attributes of a Track call, saving each one as an individual column
- `{source_name}-accounts-context-{attribute_name}`: contains the context attributes of a Track call, saving each one as an individual column

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 SlicingDice 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
API Key (required)	<code>string</code> . Access the list of databases on SlicingDice’s control panel and click on “Show Key” button for the corresponding database.

This page was last modified: 08 Jan 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](#)

© 2025 Segment.io, Inc.

[Privacy](#)

[Terms](#)

[Website Data Collection Preferences](#)

