



Documentation

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



Mobile

Server



Mobile

Server

Getting Started

Before you enable Responsys in your destinations page, there are a few things in your Segment destination settings you must set up. Once the set up is complete, you'll be able to use `.identify()` calls to add records to **Profile Lists**, **Profile Extension Tables** and `.track()` calls to add records to **Supplemental Tables**.

Pre-requisite steps

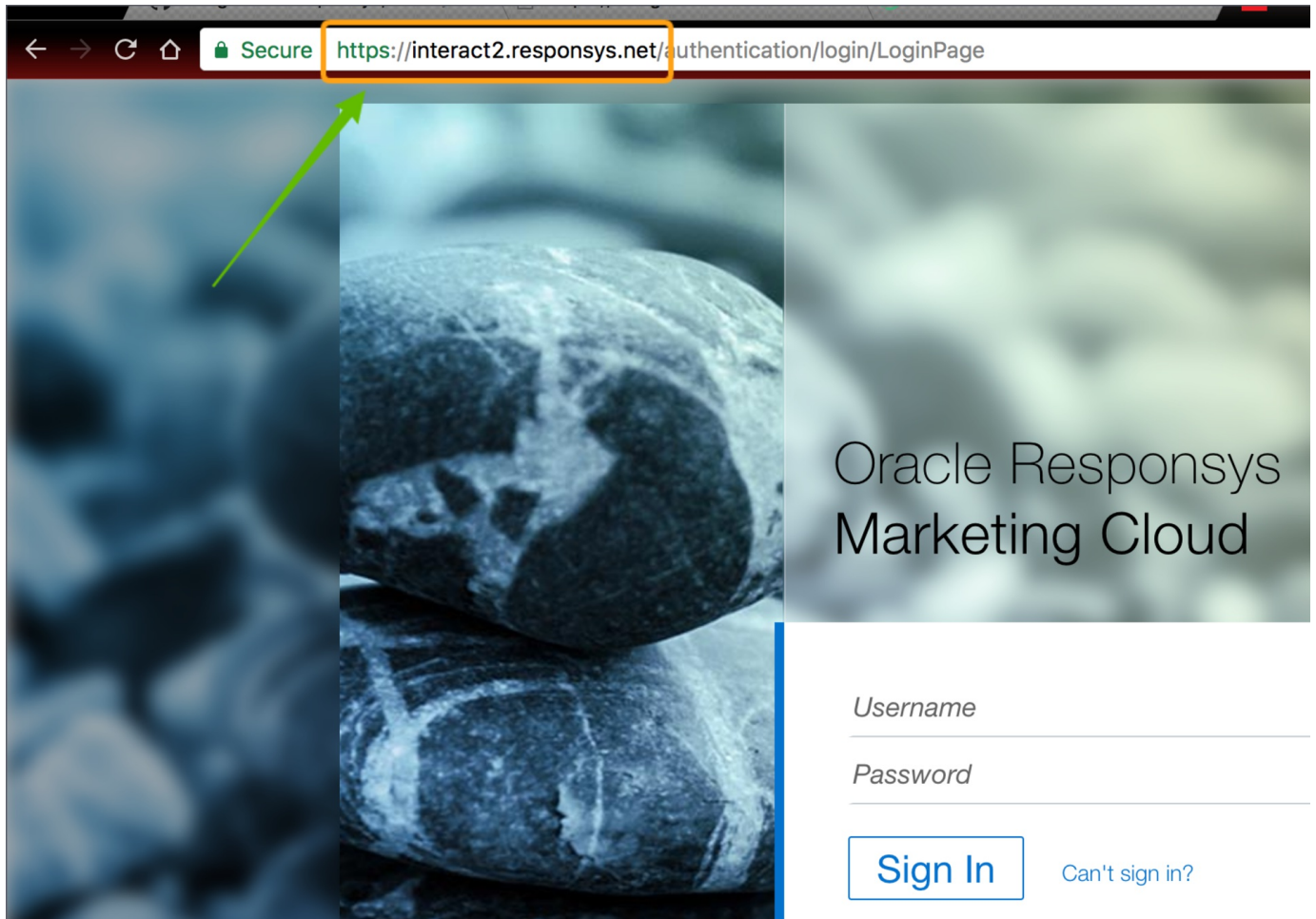
- Choose the **pod** that hosts your Responsys account.

Pod

Login Endpoint

interact2	interact2.responsys.net
interact5	interact5.responsys.net
rsys8	interact.rsys8.net
rsys9	interact.rsys9.net

NOTE: You can find your account's endpoint by simply going to your Oracle Responsys login screen and looking at your url.



1 Enter your **username** and **password** for your Responsys account. We require these credentials in order to retrieve an auth token from Oracle in order to send data to your account on your behalf.

2 Enter the name of the default **Folder** you'd like to send your Segment data to. Don't worry, you can override the default folder name on a per-call basis using destination specific options for `.identify()`.

3 Enter the name of your default **Profile List** where you would like to store your `.identify()` calls. Again, this can be overridden using destination specific options on a per-call basis as seen in the [Overriding default folder and list names](#) section.

4 Choose whether you'd like to set your **Default Permission Status** to be `OPTIN` or `OPTOUT`. Unless configured otherwise, the default will be `OPTOUT`.

Once you have entered these required settings, you are ready to integrate your Oracle Responsys account through the Segment platform!

Identify

There are two things you can do with Segment's `.identify()` calls in regards to Responsys:

- 1 Insert records to a **Profile List**.

Specify records to a Profile List.

Extend that record by upserting a corresponding record in a **Profile Extension Table**.

Merging Records to a Profile List

Take a sample `.identify()` call:

```
// analytics.js

analytics.identify('rick', {
  email: 'wubba-lubba-dub-dub@morty.com',
  seasonTwo: true,
  phone: '4012221738',
  address: {
    street: '19 Forest Lane',
    city: 'East Greenwich',
    state: 'RI',
    postalCode: '02818',
    country: 'USA'
  }
});
```

IMPORTANT: In order to merge records properly, our destination requires that all `.identify()` contain at least a `userId` or a `traits.email`!

The above call will try to first find an existing record in the provided Profile List with a matching `userId` of 'rick' and or `email` of 'wubba-lubba-dub-dub@morty.com'. If a record is found, we will update the rest of the columns so long as you pass the information using our [identify spec](#). The below is how we map our spec'd Segment traits to Responsys' semantic Profile List columns:

Segment Trait Name	Responsys Profile List Column Names
userId	CUSTOMER_ID__
email	EMAIL_ADDRESS__
phone	MOBILE_NUMBER__
address.street	POSTAL_ADDRESS_1__
address.city	CITY__
address.state	STATE__
address.postalCode	POSTAL_CODE__
address.country	COUNTRY__

Email and Mobile Permission Statuses

If you would like to keep track of users who are opting in or out of marketing communications in your apps and websites indicate a key of the custom trait to map to Responsys `EMAIL_PERMISSION_STATUS__` or `MOBILE_PERMISSION_STATUS__` fields in your Segment settings. **NOTE:** The value of this custom trait key must be a boolean. When the value is true that indicates the user wants to opt in and false indicates the user wants to opt out. Segment will transform that boolean into the appropriate Responsys accepted format (1 or 0).

Responsys does not have an API to create Profile Lists so you must create one manually inside their UI.

Merging Records to a Profile Extension Table

If you would like to send records to a **Profile Extension Table (PET)**, this destination can either create the PET for you or you can simply enter the name of any of your existing PETs. The match column name will be the `userId`

and `email` (must send at least one). So be sure to include the `userId` or `traits.email` in your `.identify()` calls!

Creating a Profile Extension Table through Segment:

All you have to do is enter the desired name of your PET in your Segment destination settings.

Say that the following is your first `.identify()` call after you've entered the PET name that does not exist yet in your Responsys account:

```
// analytics.js

analytics.identify('rick', {
  email: 'wubba-lubba-dub-dub@morty.com',
  name: 'rick',
  age: 60,
  genius: true
});
```

This would create a PET where its columns would be `NAME`, `AGE` and `GENIUS`. Since `email` is mapped already in your Profile List, we will not create a duplicate column in your PET. We will also automatically set the column type according to the value of the trait you've sent. We will run a quick validation such as checking the length of any string values to make sure we set the appropriate `STR###` as well as checking any `number` values should be an `INTEGER` or `NUMBER` in Responsys.

The following is how we map the column types as well as what the default type will be in the event more than one valid column type can be set (i.e. A string value that had 24 chars will still default to `STR500`).

Segment Trait Value Type	Possible Responsys Column Types	Default
<code>timestamp</code>	<code>TIMESTAMP</code>	<code>TIMESTAMP</code>
<code>number</code>	<code>INTEGER</code> , <code>NUMBER</code>	<code>NUMBER</code>
<code>string</code>	<code>CHAR</code> , <code>STR25</code> , <code>STR100</code> , <code>STR255</code> , <code>STR500</code> , <code>STR4000</code>	<code>STR500</code>
<code>boolean*</code>	<code>STR25</code> , <code>STR100</code> , <code>STR255</code> , <code>STR500</code> , <code>STR4000</code>	<code>STR25</code>

NOTE: *Responsys does not have a boolean column type, so we will just stringify to 'true' or 'false'.*

IMPORTANT: Since Responsys will reject any messages that try to merge columns that do not exist in a given Profile Extension Table, we will make sure to *only* send accepted column names. So it is very important to keep your tracking code, the destination settings, and your table schema in Responsys in sync!

Merging Records to Existing Profile Extension Table

If you already have a Profile Extension Table you'd like to use, simply enter the name of the list in your settings. Note that we will *only* send traits with matching column names in your schema, meaning that we will drop any traits that are not pre-defined in your PET before sending the request.

Overriding Default Folder and List Names

If you need more flexibility or need to add different users to various Folders or Profile Lists/Extension Tables, you can use our **destination specific options**.

```
// analytics.js

analytics.identify('rick', {
  vip: true
}, {
  Responsys: {
    listName: 'segment_vip_emails', // Profile List Name
    folderName: 'segment_vips', // Folder Name
    petName: 'segment_vip_users' // Profile Extension Table Name
  }
});
```

The above call will override all your default names in your Segment destination settings. It is a great way to implement programmatic bucketing of your users into specific folders and lists!

Track

You can send Segment's `.track()` events to add (**NOT** merge) records to any **Supplemental Table**. However, it is currently not possible to pull all the valid columns for a given Supplemental Table as you would with Profile Lists, meaning that you **must** explicitly map your the names of your track events and properties and provide their corresponding Responsys table names and column names and types.

Creating a Supplemental Table through Segment:

Similar to how our `.identify()` call works, you can opt to let Segment create the Supplemental tables by providing all the necessary mappings in your settings. We will try to upsert records to that table name first and if it doesn't exist, we will create the table for you using the exact column mappings you have provided in the Segment destination settings. You do not need to map every column to Segment properties if you do not wish to do so, as Responsys allows you to update records partially, with the exception of **Primary Keys**.

IMPORTANT: If you make any updates to your Segment `.track()` or Supplemental Table schema without updating the changes in your destination settings, you can risk records not being added correctly to simply having the messages rejected entirely. Especially if the changes relate to **Primary Keys configuration**.

Merging Records to Existing Supplemental Tables

As long as you provide the exact mappings that reflects the existing schema of your Supplemental Table, we can add records. You do not need to map every column to Segment properties if you do not wish to do so, as Responsys allows you to update records partially, with the exception of **Primary Keys**.

Primary Keys

All Supplemental Tables must have at least one **Primary Key**. Responsys supports up to four per table. It's very important that whichever properties you choose to set as one of the Primary Keys always have unique values. Otherwise, you will risk overwriting historical records in your Supplemental Tables.

Note that if you are using more than one primary key for a given table, only one of those need to send a unique value in order to successfully add a new record.

Using Segment's messageId as a Primary Key

The easiest way to ensure you choose a unique Primary Key is to enable **Use Segment Message ID**. This feature will send `SEGMENT_MESSAGE_ID` as a primary key column with the value being Segment's `messageId` field. Since this property is uniquely generated by Segment, you can ensure that you do not overwrite any historical records in your tables.

Note that this does count against your limit of four Primary Keys per table.

If you'd like to use this feature for an existing Supplemental Table, you can modify your schema and add the column `SEGMENT_MESSAGE_ID`. If you do not do this but enable this option, all your track events will be rejected by Responsys.

Furthermore, if you do not provide Segment with **all** of the primary key column names, the message will be

rejected by Responsys API.

Map UserId

In order to attribute the event record in your Supplemental Tables with the matching records in your Profile Lists or Profile Extension Tables, we recommend enabling the **Map Segment UserId** option. If enabled, we will send the `userId` attached to the `.track()` payload as a Responsys field.

IMPORTANT: Make sure if you are mapping an existing Supplemental Table, you have created a field in your table called `CUSTOMER_ID_`! Otherwise if you have not created the table and is relying on Segment to do so, we will create the field `CUSTOMER_ID_` and set the field type as `STR500`.

Map Email

Since we still accept `.identify()` calls without `userIds` so long as it has `traits.email`, you can also enable this for your `.track()` events if you want to attribute event records using `email`. We will lookup `context.traits.email` (which is automatically cached if you are using a mobile library), `properties.email`, and fallback on the `userId` field if it is a valid email address.

IMPORTANT: Make sure if you mapping an existing Supplemental Table, you have created a field in your table called `EMAIL_ADDRESS_`! Otherwise, if you have not created the table and is relying on Segment to do so, we will create the field `EMAIL_ADDRESS_` and set the field type as `STR500`.

FAQ

How do you handle nested objects?

We will flatten any traits or properties that are nested objects of arrays and objects with the delimiter of an underscore.

Take the following `.track()` call with some nested properties:

```
// analytics.js

analytics.track('Watched Rick and Morty', {
  show: {
    title: 'Rick and Morty',
    episode: {
      season: 1,
      categories: ['comedy']
    }
  }
});
```

We would send column names as:

- `show_title: 'Rick and Morty'`
- `show_episode_season: '1'` (*note: Responsys accepts strings for column types NUMBER and INTEGER so we always stringify numbers*)
- `show_episode_categories_0: 'comedy'`

IMPORTANT: When mapping your Segment Properties for those that you know you will be sending as nested objects, be sure to type the flattened version. So for example, if you are sending:

```
// properties
{
  address: {
    street: '100 California Street'
  }
}
```

You should enter `address_street` as your Segment Property Name.

I'm seeing a 'Merge cannot be done on one field' in the Segment debugger!

This error occurs when you try to upsert a record to a Profile Extension Table but you did not provide any additional traits other than what we used to match the columns. For PETs, we use your `userId` or `traits.email` from your `.identify()` calls to upsert the record. You need to send at least one other trait that would map to a field in your PET.

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 Oracle Responsys 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
Events	<code>mixed</code> , defaults to . Use these fields to map your Segment event names to Responsys Supplemental Tables. We'll only send events and properties that you specify.
Folder Name (required)	<code>string</code> . Please enter the default Folder Name that holds the provided Profile List and Profile Extension Table.
Global Routing Subdomain	<code>string</code> . Global Routing Service Endpoint. Users can find this URL by logging into their Responsys account, click Account > Global settings > Account configuration.
Profile List Name (required)	<code>string</code> . Please enter the default Profile List. Make sure that the Profile List is inside the provided Folder Name.
Set Default Permission Setting to OPTIN	<code>boolean</code> , defaults to <code>FALSE</code> . Choose whether records added to Profile Lists should be <code>'OPTIN'</code> rather than the default <code>'OPTOUT'</code>
Email Permission Status Trait Mapping	<code>string</code> . Please enter the trait you would like to be mapped to <code>EMAIL_PEERMISSION_STATUS_</code> . This will allow you to track users who opt in and out of marketing communications in your apps and websites. The value of this trait MUST be a boolean.
Mobile Permission Status Trait Mapping	<code>string</code> . Please enter the trait you would like to be mapped to <code>MOBILE_PERMISSION_STATUS_</code> . This will allow you to track users who opt in and out of marketing communications in your apps and websites. The value of this trait MUST be a boolean.
Password (required)	<code>password</code> . Please enter your account's password.

SETTING	DESCRIPTION
Profile Extension Table	<code>string</code> . Please enter the name of your default Profile Extension Table.
Pod (required)	<code>select</code> , defaults to 2. The pod that hosts your Responsys account. Simply check your url during your login screen. This is required to any requests to your account. The default is <code>interact2</code> .
Username (required)	<code>string</code> . Please enter your account's username.

This page was last modified: 09 Aug 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](#)

© 2025 Segment.io, Inc.

[Privacy](#)

[Terms](#)

[Website Data Collection Preferences](#)



