Documentation

Here's the payload of a typical Identify call with most common fields removed:

```
{
  "type": "identify",
  "traits": {
    "name": "Peter Gibbons",
    "email": "peter@example.com",
    "plan": "premium",
    "logins": 5
  },
  "userId": "97980cfea0067"
}
```

And here's the corresponding JavaScript event that would generate the above payload:

```
analytics.identify("97980cfea0067", {
  name: "Peter Gibbons",
  email: "peter@example.com",
  plan: "premium",
  logins: 5
});
```

Beyond the common fields, an Identify call has the following fields:

FIELD	TYPE	DESCRIPTION
traits *optional*	Object	Free-form dictionary of traits of the user, like `email` or `name`. See the Traits field docs for a list of reserved trait names.
userId *required; optional if* `anonymousID` *is set instead*	String	Unique identifier for the user in your database. A `userId` or an `anonymousId` is required. See the Identities docs for more details.

Example

Here's a complete example of an Identify call:

```
{
  "anonymousId": "507f191e810c19729de860ea",
  "channel": "browser",
  "context": {
    "ip": "8.8.8.8",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.115 Safari/537.36"
  },
  "integrations": {
    "All": false,
    "Mixpanel": true,
    "Salesforce": true
  },
  "messageId": "022bb90c-bbac-11e4-8dfc-aa07a5b093db",
  "receivedAt": "2015-02-23T22:28:55.387Z",
  "sentAt": "2015-02-23T22:28:55.111Z",
  "timestamp": "2015-02-23T22:28:55.111Z",
  "traits": {
    "name": "Peter Gibbons",
    "email": "peter@example.com",
    "plan": "premium",
    "logins": 5,
    "address": {
      "street": "6th St",
      "city": "San Francisco",
      "state": "CA",
      "postalCode": "94103",
      "country": "USA"
    }
  },
  "type": "identify",
  "userId": "97980cfea0067",
  "version": "1.1"
}
```

Create your own Identify call

Use the following interactive code pen to see what your Identify calls would look like with user-provided information:

Sample Identify

Name: `John Smith`   Email: `john-smith@gmail.com`   Plan: `Premium`   Logins:

`2`   Street: `101 Spear Street`   City: `San Francisco`   State:

`CA`   Zip Code: `94105`   Country: `US`

See a sample identify call

Sample Identify Call

```
Sample output goes here!
```

Identities

The Identify call specifies a customer identity that you can reference across the customer's whole lifetime. **Every Identify call must have a User ID or an Anonymous ID**, depending on how much you know about the user in question.

Anonymous ID

There are certain cases where you don't actually know who the user is according to your database, but you still want to be able to tie them to traits, events, or page views. For example, you may not know who a user is when tracking newsletter signups or anonymous page views.

In these cases, you should use an Anonymous ID.

The Anonymous ID can be any pseudo-unique identifier. For example, on your servers you can use a session id. If you don't have any readily available identifier, you can always generate a new random one — Segment recommends UUIDv4 format.

> ⓘ
> Segment's browser and mobile libraries automatically use Anonymous IDs to keep track of users as they navigate around your website or app, so you don't need to worry about them when using those libraries.

Here's an example of a JavaScript event for an anonymous user:

```
analytics.identify({
  subscriptionStatus: 'inactive'
});
```

User ID

User IDs are a more permanent and robust identifier, like a database ID. Since these IDs are consistent across a customer's lifetime, Identify calls should include a User ID as often as possible.

A User ID is usually the unique identifier that you recognize a user by in your own database. For example, if you're using MongoDB, User IDs might look something like this: `507f191e810c19729de860ea`.

Segment recommends using database IDs, in `uuidv4` format, instead of email addresses or usernames because database IDs *never* change. That guarantees that even if the user changes their email address, you can still recognize them as the same person in all of your analytics tools, and you'll be able to correlate analytics data with your own internal database.

> ✅
> Instead of using an email address or a username as a User ID, send them along as custom traits.

Custom traits

Custom traits are pieces of information you know about a user that are included in an Identify call. These could be demographics like `age` or `gender`, account-specific like `plan`, or even things like whether a user has seen a particular A/B test variation.

Segment has reserved some custom traits that have semantic meanings for users, and will handle them in special ways. For example, Segment always expects `email` to be a string of the user's email address. Segment sends this on to destinations like *Mailchimp* that require an email address for their tracking.

> ⚠️
> Only use reserved traits for their intended meaning.

Reserved custom traits Segment has standardized:

TRAIT	TYPE	DESCRIPTION
`address`	Object	Street address of a user optionally containing: `city`, `country`, `postalCode`, `state`, or `street`
`age`	Number	Age of a user
`avatar`	String	URL to an avatar image for the user
`birthday`	Date	User's birthday
`company`	Object	Company the user represents, optionally containing: `name` (String), `id` (String or Number), `industry` (String), `employee_count` (Number) or `plan` (String)
`createdAt`	Date	Date the user's account was first created. Segment recommends using ISO-8601 date strings.
`description`	String	Description of the user
`email`	String	Email address of a user
`firstName`	String	First name of a user
`gender`	String	Gender of a user
`id`	String	Unique ID in your database for a user
`lastName`	String	Last name of a user
`name`	String	Full name of a user. If you only pass a first and last name Segment automatically fills in the full name for you.
`phone`	String	Phone number of a user
`title`	String	Title of a user, usually related to their position at a specific company. Example: "VP of Engineering"
`username`	String	User's username. This should be unique to each user, like the usernames of Twitter or GitHub.
`website`	String	Website of a user

> ℹ️
> You might be used to some destinations recognizing special traits by slightly different names. For example, Mixpanel

recognizes a `$created` trait when the user's account was first created, while Intercom recognizes the same trait as `created_at` instead. Segment attempts to handle all the destination-specific conversions for you automatically. If you need help understanding if a specific field will be converted to a destination, take a look at Segment's open source integration code, view the destination's documentation, or contact Segment support.

You can pass these reserved traits using camelCase or snake_case, so in JavaScript you can match the rest of your camelCase code by sending `firstName`, while in Ruby you can match your snake-case code by sending `first_name`. That way the API never seems alien to your code base. Keep in mind that not all destinations support these reserved traits, so sending these traits in camelCase and snake_case can result in two sets of traits in other destinations.

This page was last modified: 23 Apr 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