ITERABLE

Data Integration Guide

This technical guide provides an overview of the data integration options in Iterable. It details a review of the data model, the different methods of sending data to Iterable, exporting data from Iterable, and key personalization and integration considerations. If you're a technology leader within your company or will be involved in implementing the Iterable platform, this guide is for you.

Table of Contents

The Iterable Data Model 3
- User Data 3
- Unique Identifier and Available Project Types 4
- Event Data 8
- Segmentation 8
- Supported Data Types 9
- Reserved and Special Fields 11

Data Import Options 12
- REST APIs 12
- CSV Uploads 13

Data Export Options 14
- REST APIs (JSON and CSV) 15
- System Webhook 15
- GUI Export 16

Integrations 16
- Partner Page 17
- Journey Webhooks 18

Template Personalization 19
- User & Event Data 19
- Data Feeds 20
- Catalog 22
- Localization 25
- SMS 26

Mobile 27
- Push 28
- In-App 28
- Mobile Inbox 29
- In-Browser Messaging 30

Appendix 1: Sample Integration Timeline 31

Appendix 2: Limits in Iterable 31

Data in Iterable

Iterable enables customers to send highly targeted and personalized messages. Iterable does this by ingesting contextual and behavioral data about your users from your different systems (e.g., shopping carts, website/app behavioral data, internal event tracking system) and makes this data available for building segmentations, creating filters, setting triggers, and message personalization in real-time. Having the right data in Iterable is key to a speedy implementation and a successful marketing team. Our friendly Solutions Consulting team here at Iterable is happy to assist you in structuring and transferring your data to Iterable.

The Iterable Data Model

Iterable is built on a true NoSQL data model. Unlike a traditional ESP, Iterable models user data and events in JSON, giving it an edge by simplifying complex relationships. Marketers can focus on asking the right questions about their users and executing on their personalization strategy, not on wasting time fighting the tools or figuring out the data data schema.

> **"Everything is a JSON object, so marketers can focus on pulling relevant data and not on primary/foreign key fiascos."**

The main place where data lives in Iterable is the user profile. There is a unique user profile for every user tracked by the Iterable platform. Within a user profile there are two main types of data: user data and event data.

User Data

User data fields are usually things like personal characteristics, geographic location and the like. Marketers can use all these values to create targeted user segmentations, as well as dynamic and personalized templates. Marketers also have the added option of using control structures (including loops) to further personalize template structures.

> **"There is a soft limit of 1,000 fields that can be stored on the user profile. If you think your integration needs require more than 1,000 unique fields, please reach out to your Solutions Consultant."**

User Data Preview

The purple box is an example of an array on a profile and the blue box is the first object in the array (note how it has its own values).

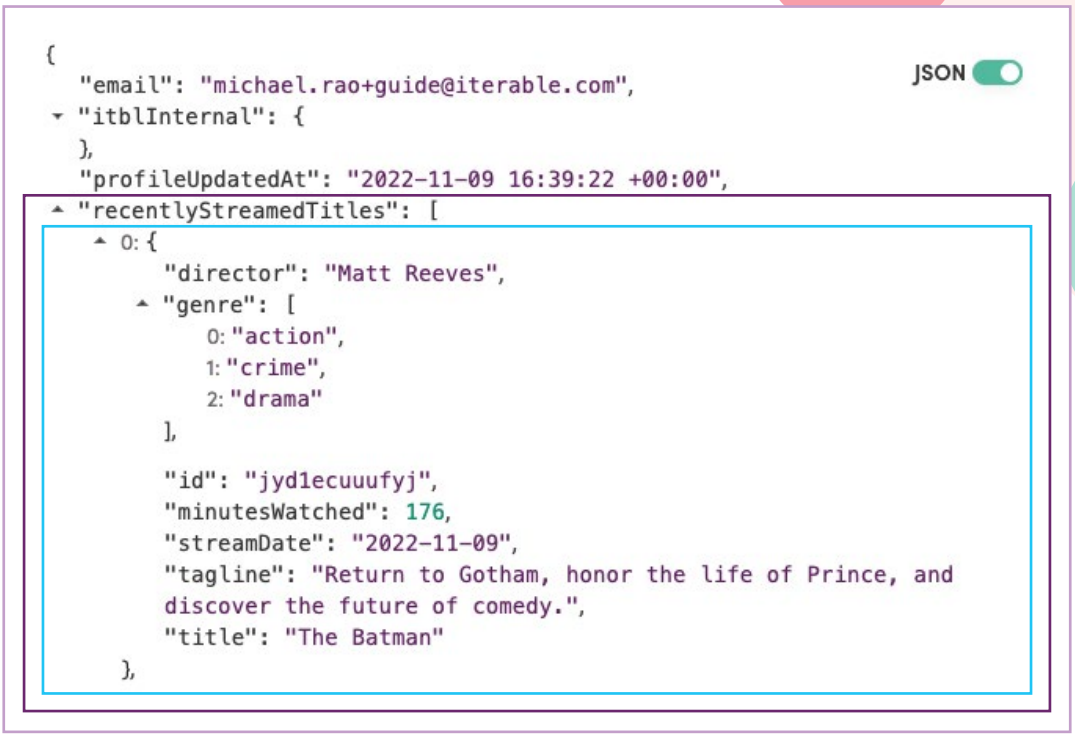


Figure 1: User Data

Accessing the first director in the 'recentlyStreamedTitles' array in a template, is a simple syntax of *{{recentlyStreamedTitles.[0].director}}*.

The advantage of storing data as an array of objects is that it allows concise and accurate access to specific properties within a collection of related data points. Once data is passed to Iterable in this format, our Segmentation tool can query the database to answer questions like: "who recently streamed a movie from the 'Oscar Nominees' category?", " who recently streamed a movie directed by Greta Gerwig?", or "who has streamed movie 'x' and movie 'y' in the last 'n' number of days?

Unique Identifier & Available Project Types

Iterable accepts either `email` or `userId` as the primary key for user data. When you create a project in the Iterable platform you can choose to use one of these fields as the primary identifier, or you can choose to use both (where at least one field is required for each new user and both fields are enforced as unique).

There are three different schemas available to identify users in a project:

- **Email-based project -** Uses `email` as the unique identifier.
- **UserID-based project -** Uses `userId` as a unique identifier.
- **Hybrid project** - Uses `email` or `userId` as a unique identifier.

The unique identifier for your project impacts how Iterable behaves in many scenarios. Here is a comparison of how each schema impacts your data:

	Email-based project	UserID-based project	Hybrid project
Unique identifier	`email`	`userId`	`email` and `userId` (at least one of these fields is required)
Email uniqueness	Unique	Non-unique	Unique
userId uniqueness	Non-unique	Unique	Unique
Identify User in API requests	`email` or `userId` (more info)	`userId`	`email` or `userId`
Multiple users with same email	No	Yes	No
Placeholder emails for anonymous users	Yes (***NOTE:*** *Iterable recommends using a Hybrid project if email address is the primary identifier for known users and userId is the primary identifier for anonymous users. Please see more detail under "Hybrid Projects - UserID or Email as the Unique Identifier" on Page 7.)*	No	No
GDPR forget	`email`	`userId`	`email` or `userId`

Email-based Projects - Email as the Unique Identifier

When email is the primary key you can optionally provide a userId to be used as a secondary key giving you the ability to update user and event data using either key. While Iterable enforces uniqueness on the primary key, we do not enforce uniqueness on the secondary key, so it is important to ensure your integration will provide a unique secondary key if you choose to include it. A non-unique userId will cause data discrepancies with future updates and event attribution. This gives you the following options:

- Create a user using `email` as the primary key and `userId` as the secondary key, where future updates and event attribution can be done with either `email` or `userId`.
- Create a user with just the `email`, future updates and event attribution can be done with the `email`.

If you only have userId, and email is set as the primary key, you'll need to create a user using the `userId` and a placeholder email, which should have the format <**userId**>@placeholder.email. Any attempted email sent to a "placeholder.email" domain will be automatically suppressed. You can update the user's email via API call when it becomes available in the Iterable platform. Future updates and event attribution can be done with either the `userId` or the placeholder `email`.

If you need to change a given email address for a user in an Email-based project, there is a dedicated POST users/updateEmail API endpoint for that purpose. A sample request is provided below:

```
{
 "currentEmail": "oldEmail@example.com",
 "newEmail": "newEmail@example.com"
}
```

You can update an existing user's userId by calling the POST /api/users/update API. A sample request is provided below:

```
{
 "email": "docs@example.com",
 "dataFields": {
    "userId": "new-user-ID"
 },
 "mergeNestedObjects": true
}
```

UserID-based Projects - UserID as the Unique Identifier

Iterable maintains a unique constraint on `userId` when it is set at the primary key. `Email` cannot be set as the secondary key. This means that the `userId` is required when creating a new user. Future updates and event attribution must also be done with the `userId`.

If you need to change a given userId or email for a user in a Hybrid project, call the POST /api/users/update API. A sample request is provided below:

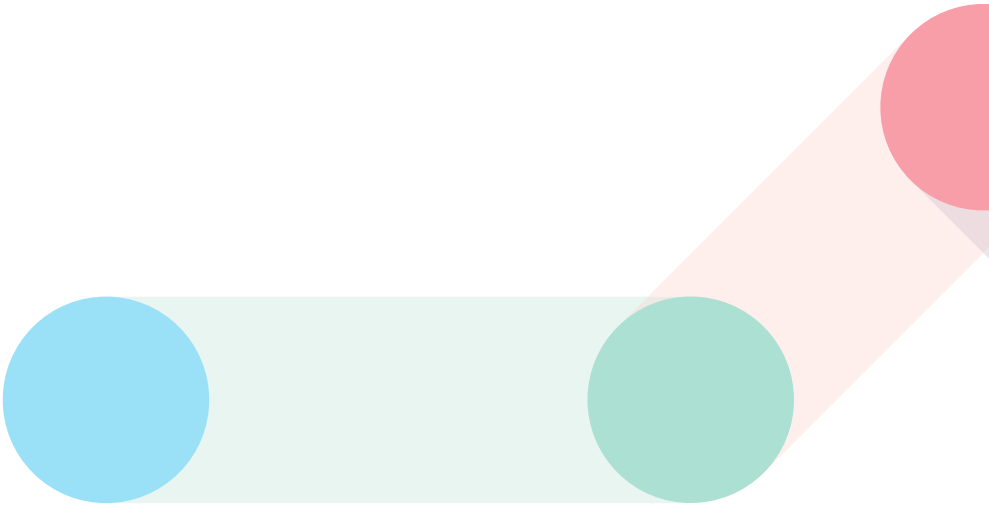
```
{
  "userId": "original-user-ID",
  "dataFields": {
    "userId": "new-user-ID",
    "email": "new-email@example.com"
  },
  "mergeNestedObjects": true
}
```

Hybrid Projects - UserID or Email as the Unique Identifier

Iterable maintains a unique constraint on the primary key, whether it is `userId` or `Email`. Future updates and event attribution must also be done with the primary key.

If you need to change a given userId or email for a user in a UserID-based project, call the POST /api/users/update API. A sample request is provided below:

```
{
  "userId": "original-user-ID",
  "dataFields": {
    "userId": "new-user-ID",
    "email": "new-email@example.com"
  },
  "mergeNestedObjects": true
}
```



Event Data

Custom Events

Custom Events are actions a user takes on your properties (e.g., website or app). These are specific to your business and can be any unique activity you want to track. Custom events enrich user profiles, and you can use them to personalize the messages that you send.

Here are some examples:

Sign up ● View an item ● Purchase ● Complete a level ● Video stream started

Each custom event contains a `dataFields` object that can include any relevant metadata about the event. Data included in this manner is available for user segmentation and template personalization.

If you are streaming events in real-time, you do not need to provide a `createdAt` data field. Iterable will timestamp the incoming event automatically. If you are backfilling events, provide the `createdAt` data field as a UNIX Epoch (seconds).

Custom events and their associated metadata can be stored and used indefinitely, meaning that you are not limited to a certain lookback period when segmenting users or triggering Journeys. You also have the option to set retention periods for your discrete custom events that dictate how long Iterable stored event data before it is deleted.

System Events

Iterable will generate System Events for all messaging and channel touch points. These system events include data points around each channels' sends, opens, clicks, bounces, and unsubscribes in an easy to ingest format. A full list of system events can be found here.

Segmentation

Because Iterable's NoSQL approach doesn't require users to memorize tables (and the various associated foreign/primary keys), marketers can instead focus on deriving value from their data.

You can run user segmentations using all user and event data without coding and receive real-time results on exactly how many people fall into a specific segment, all from the Iterable UI.

Segments are generated in seconds, not hours or days. There is no limit in how complex you can make the rules and no limit to the total number of segments you can create.

Here are some example segmentations that marketers can run.

1. Users who have items worth more than $100 in their shopping cart and haven't purchased an item in the "computers" category.

2. Residents of San Francisco who have opened more than three emails in the last month, clicked on two emails or more, and went on a run between two miles and five miles in the last week.

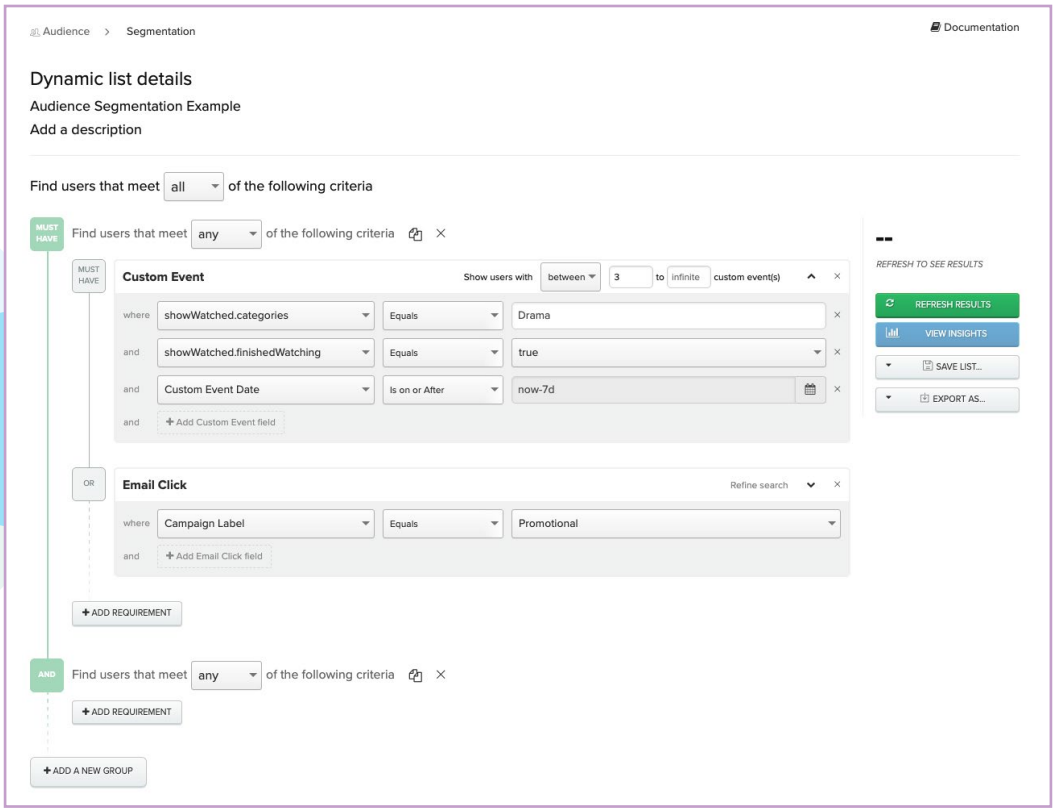


Figure 2: Segmentation screenshot

Supported Data Types

String

Strings can contain both characters and numbers. If the field already exists in the environment, then the CSV import will match the datatype of the user field. In an API upload, strings have double quotes around them.

The maximum size for an individual string data field is 32KB.

Examples: `"Justin", "San Francisco", "0qksnskjanxjk"`

DateTime

A date field is a formatted string type in Iterable. Fields that are dates have many uses for segmentation in Iterable, like relative dates and date ranges.

Here is the list of acceptable date formats:

- `yyyy-MM-dd HH:mm:ss ZZ`
- `yyyy-MM-dd HH:mm:ss`
- `yyyy-MM-dd'T'HH:mm:ss.SSSZZ`
- `yyyy-MM-dd`

Examples: `"2018-10-04 00:54:00 +00:00"`, `"2000-01-01 00:00:00"`, `"2000-01-01T00:00:00.000-0400"`

Integer/Long

Integers and Longs are numbers without decimals.

Examples: `-24, 1922, 11`

Boolean

A field that only can read true or false. Do not include quotation marks.

Examples: `true, false`

Double or Float

Doubles and floats are numbers that have decimals.

Examples: `1.0, 1982.12, -82619.1314`

Objects

An object contains multiple components. For instance, if a user had a cat object with multiple fields that describe the cat, the data about the cat could be contained in an object.

Objects can be added as a data field via the Iterable API, but not through the CSV list import. More on this in the data import section.

Examples: `"cat": { "name": "Oscar", "age": 4, "favoriteToy": "fluffy stick"}`

Array

An array is a list of objects. Arrays can be added as a data field via the Iterable API, but not through the CSV list import. More on this in the data import section.

Example: `"favoriteFoods": ["pizza", "bagel", "ramen", "burger"]`

Reserved and Special Fields

Reserved Fields

In Iterable, some fields on the user profile are automatically created and maintained by Iterable. In most cases, these fields cannot be edited directly, and should not be included when updating a user's profile.

For example, the following fields are reserved fields and maintained by the Iterable platform:

profileUpdatedAt

`profileUpdatedAt` is the date the user profile was last updated, either via CSV upload or API call. The data type is DateTime and the format looks like "`2016-08-02 18:53:45 +00:00`".

signupDate

`signupDate` is the date when the user profile was created in Iterable. The data type is DateTime and the format looks like "`2016-08-02 18:53:45 +00:00`".

signupSource

`signupSource` is how the user entered the Iterable database. Includes options like "`API`" or "`IMPORT`". The data type is String.

For a full list of Reserved Fields please refer to this document: Iterable-Specific Fields.

Special Fields

There are case-sensitive and space-sensitive data fields that Iterable expects. When updating the value of these data fields, be sure that the key is properly formatted.

For example, the following fields are special fields:

- `campaignId` *integer*
- `createdAt` *long (UNIX Epoch in seconds)*
- `email` *string*
- `userId` *string*
- `phoneNumber` *string*
- `ip` *string*
- `locale` *string*
- `devices` *array*

For a full list of Iterable maintained fields please refer to this document: Iterable Specific Fields.

Data Import Options

There are several ways of importing data into Iterable.

This document covers the following methods:

- CSV Uploads
- REST APIs
- Partner Integration

It's important to note that once a data field has been set in Iterable's database, the key name and data type cannot be changed. For this reason, we recommend testing your imports in a sandbox before moving to your production project. If you accidentally add an incorrect data field, you can hide it from the Iterable UI in the Project > Settings page.

REST API

We have a robust set of REST API endpoints to get user and event data into Iterable. **All endpoints can be found at:** https://api.iterable.com/api/docs

Rate Limits
Any applicable rate limits will be listed on the API Explorer page.

Sample Payloads
Sample payloads for all endpoints can be found in our API Overview.

User Data

There are two endpoints for updating user data fields. The first operates on a single user, and the other operates on a batch of users. Typically, the batch operation is used to import historical data, and the individual operation is used to stream updates in real-time.

POST **/api/users/update**

Updates a user profile with new data fields. Missing fields are not deleted and new data is merged.

POST **/api/users/bulkUpdate**

Updates the user profiles for one or more users. The maximum number of users that can be included per request depends on the object size per user. The size limit of the request is 4MB. We generally find that using batches of 1,000 user profiles is a good rule of thumb. Depending on the size of each user object, you may be able to include more or less than 1,000 users per request.

Note: This endpoint is not recommended for time-sensitive user updates, as there may be a delay before the updates are completely processed due to the potentially high volume of these requests.

CSV Uploads

The simplest user data import option is CSV upload. This is a standard approach that requires matching headers in the CSV. In case a column header is not found, Iterable automatically creates a new key on the fly.

Keep in mind that the headers are case sensitive. This means that an upload with the headers "age" and "Age" will result in two different keys on the user profiles. Leaving headers blank will not overwrite any individual keys.

> **"The keys in the key-value setup are case and space-sensitive."**

Note: CSV uploads can only be used for user profile data. In order to input event data (or arrays /objects) you'll have to use the REST API.

Note: All new, unfamiliar uploaded headers are automatically casted as String. In order to upload integers, booleans, and other data types, you must first set the data types of the field via the API. However, array and object fields are not supported via CSV. Our Customer Success team can aid with this process.

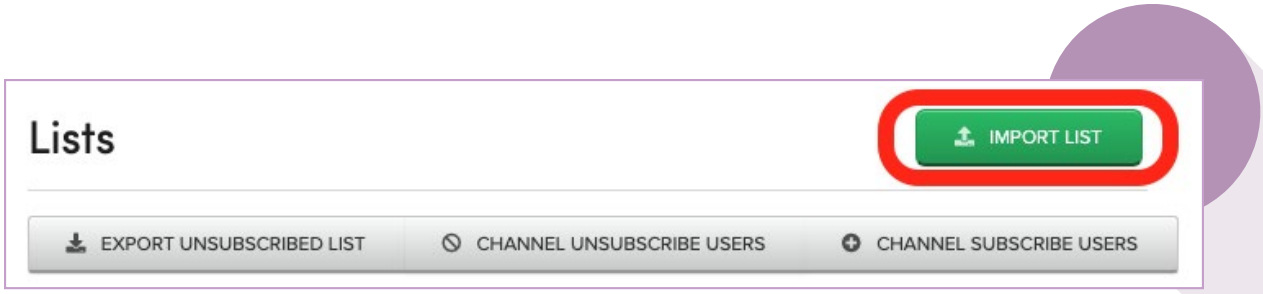


Figure 3: CSV upload screenshot

Data Export Options

Iterable is an open platform. Our system is built with two-way integrations in mind. Many of our customers leverage our data export options to analyze and optimize their user engagement strategy.

This document covers the following options:

- API Data Export
- System Webhook
- Via GUI
- Partner Integrations

Each option offers different capabilities. Whatever your ideal setup might be, from CSV export all the way to real time user engagement feedback to your internal machine learning models, we're here to consult with you on your best marketing set up.

REST APIs

We have a robust set of REST API endpoints to get engagement data out of Iterable. This approach is recommended for batch exports.

All endpoints can be found at: https://api.iterable.com/api/docs

GET **/api/export/data.json**

Exports specified data type into JSON.

GET **/api/export/data.csv**

Exports specified data type into CSV.

GET **/api/campaigns/metrics**

Exports aggregate engagement metrics for campaign(s). .

Rate Limits

Any applicable rate limits will be listed on the API Explorer page.

Sample Payloads

Sample payloads for all endpoints can be found in our API Overview.

System Webhooks

Iterable can webhook out all engagement events as they occur in real-time. These simple webhooks will send a POST request with up to two headers to your specified endpoint. The headers are: **Content-Type, value = application/json**, and an Authorization header (unless you choose NoAuth when creating the webhook).

The body of the request will depend on the engagement event being exported. See System Webhooks for a full list of available webhooks and payload details.

Note: If the initial system webhook call fails, either with an error or after a 10 second timeout, 1 retry will occur after 3 minutes.

GUI Export

At the end of the day, marketers can always pull campaign data via our GUI for a quick analysis.

1. Go to a campaign; click on Export Data
2. Select export criteria; Click EXPORT CAMPAIGNS TO CSV

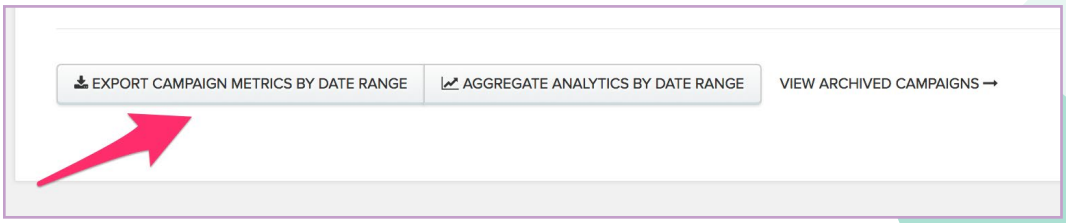


Figure 4: GUI export

Partner Integrations

Iterable has native integrations for a variety of use cases with many partner solutions. These pre-built solutions can help you save time and resources when it comes to exchanging data between Iterable and the rest of your technology ecosystem. Iterable's partner network is constantly growing, so please go to the Partnerships page on the Iterable website to see the most up-to-date information.

If you have any questions regarding these integrations, please reach out to your Solutions Consultant.

EMAIL SERVICES

aws
AtData
Dyspatch
Email on Acid
EVEREST
Hi-iQ
Iterate
kickbox
mailfloss
mailgun
Playable
PUSH PLANET
QuickEmailVerification
Return Path
TWILIO SendGrid
Sendtric
SPARKPOST
stensul
TAXI FOR EMAIL
wove

PERSONALIZATION

crowdin
eduMe
Faraday
freshrelevance
inkit
Kickdynamic
LiftIgniter
Lob
lokalise
Movable Ink
NiftyImages
northpass
one spot
PFL
phrasee
poplar
Reelevant.
Reignite
WRENCH AI
zembula

LEAD CAPTURE

Digioh
Instapage
revv
VIRALSWEEP

DATA MANAGEMENT

ACTIONIQ
Census
CROWDSKOUT
DATAGRAIL
Fivetran
Freshpaint
hightouch
Kustomer
LeadsBridge
Lytics
Microsoft Dynamics 365
mparticle
n8n.io
PieEye
POLYTOMIC
rudderstack
segment
snowflake
SNOWPLOW
Stitch
TEALIUM
Transcend
tray.io
workato
zapier
zendesk support

MOBILE

adjust
AppsFlyer
branch
factual
FOURSQUARE
HEROW
KOCHAVA
PassKit
Radar
REGALVOICE
RevenueCat
TELNYX
twilio

E-COMMERCE

friendbuy
recharge
sageflo
shopify
Talon.One
zinrelo

ANALYTICS

Amplitude
AURYC
Google Analytics
Heap
mixpanel

Journey Webhooks

Occasionally, you will want to use Journey Webhooks to send data from Iterable to external servers (your own or third-party) as part of a Journey.

A Journey webhook makes a standard HTTP request to an external server, including relevant data in the request body (JSON or form URL-encoded). To authenticate with the external server, a webhook request can include authentication information.

Journey webhooks have many uses. For example:

- Setting up a direct mail integration with Inkit or Lob.
- Sending data to a BI tool such as Amplitude or Mixpanel.
- Saving data to a data infrastructure product such as Segment.
- Initiating NPS surveys with Iterate.
- Triggering an action on Zapier.
- Using Twilio to automate phone calls.
- Capturing data to your own servers.

You can also create custom webhooks to send data to an endpoint of your choosing. Within these custom webhooks you may choose: how to format the data (JSON vs form), what data to send, where to send it to and an authorization header for security purposes.

For more information around our Journey Webhooks solution, please review this documentation or contact your Iterable Solutions Consultant.

Template Personalization

The heart of modern marketing is personalization. Our segmentation tool will allow marketers to send messages to targeted cohorts, but what if you want to go further with content personalization? Where does the data live? How do you generate millions of templates with different recommendations, while still empowering marketers with creative freedom?

In addition to merging user and event data into templates, Iterable supports template personalization from data tables and external resources.

This document covers the following options:

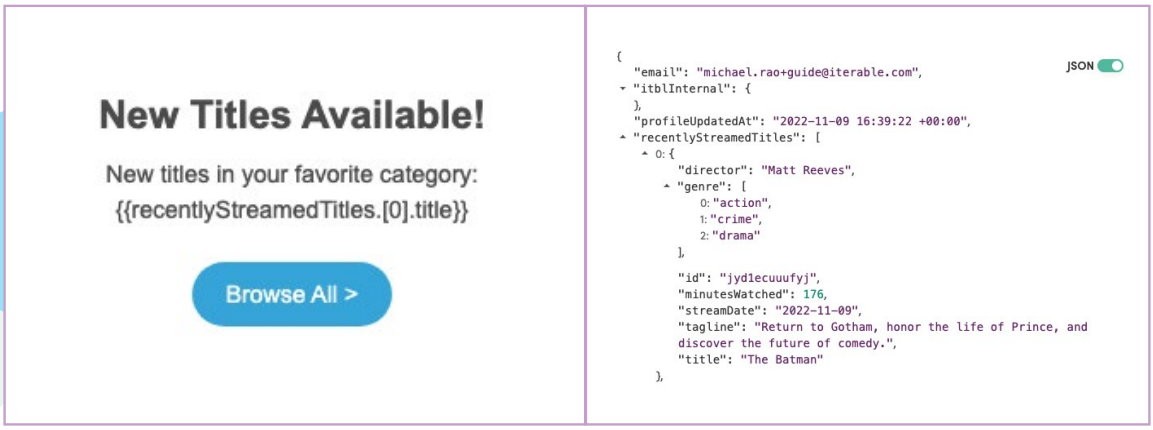
- User & Event Data
- Data Feeds
- Catalog (Iterable's Metadata Store)

User & Event Data

The core of Iterable's template personalization is built on rendering User Profile and Event data with the help of handlebars.java merge parameters. All User & triggered Event data fields can be used in any Iterable message template.

While many other NoSQL based messaging platforms will flatten JSON objects, Iterable fully supports nested data fields. In practice, this means that marketers can easily personalize templates with complex data models. For example:

User Profile Data: Example Template Rendering:



In addition to merging User and Event data, Iterable templates support handlebars.java helpers. For a full list of available helpers, see our Support documentation on Personalizing Templates with Handlebars.

Data Feeds

Iterable's data feed feature gives marketers the ability to request template personalization data from an external resource at send time. Because the data is requested at send time, data feeds are a great option for time-sensitive data. For example, SeatGeek is using data feeds to insert ticket recommendations into their emails on a 1:1 user basis.

> **"Data feeds allow you to host an endpoint that returns data (in JSON, XML, Atom, RSS) that can be merged in a template."**

At the time of sending the message, we'll make a GET request to an external endpoint for each send. We give the option to cache the endpoint response for up to 1 hour, and we allow embedding user attributes in the endpoint URL, which is useful for returning user specific personalization data.

Example Feed URL:

```
https://www.yourcompany.com/api/ml.json{{#urlEncode}}?email={{email}}{{/urlEncode}}
```

In this setup, we'll include the user's email as a parameter for the request. The feed should return a JSON object.

Example Feed Response:

```
{
        "recommendation": [{
                "name": "house2",
                "homeZip": 94107,
                "postingUrl": "https: //www.zillow.com/home/1906157",
                "homeImage1": "https://ssl.zillow.com/photo/
                genMid.457926_0.jpg",
                "price": 1950000,
                "br": 3,
                "ba": 1.5,
                "sqft": 1291
        }, {
                "name ": "house3",
                "homeZip": 94107,
                "postingUrl": "https: //www.zillow.com/home/1906157",
                "homeImage1": "https://ssl.cdn-zillow.com/
                genMid.457926_0.jpg",
                "price": 1950000,
                "br": 3,
                "ba": 1.5,
                "sqft": 1291
        }]
}
```

Data Feed Architecture Diagram:

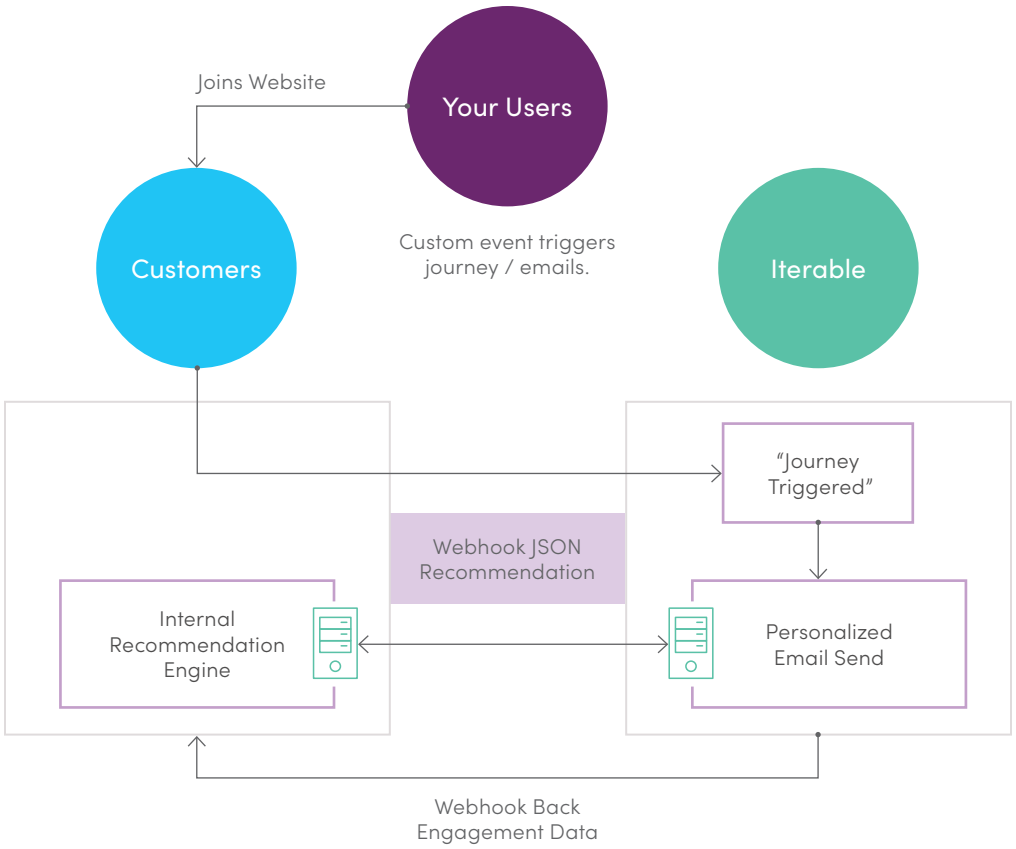


Figure 5: Data Feed Architecture

See our Support Documentation for a full explanation of this feature.

Catalog

Please contact your account executive to discuss enabling this feature within your Iterable account.

When sending messages to your users—using any medium—it's almost always necessary to include information about your organization's products, places, events, or services. With Catalog, you can store this information directly in Iterable and use it to personalize the messages you send.

Catalog is a set of related items, each of which has various descriptive fields. Each item stored in a catalog is a catalog item.

> For example, a **Restaurants** catalog might contain items such as `JoesRestaurant`, `KellysCafe`, and `SalsKitchen`. Each of these items could have descriptive fields such as `cuisine`, `averagePrice`, and `rating`.

Common uses for Catalog include product catalogs, job listings, and real estate properties. For example, a food delivery service could create a metadata store with restaurant information in every available zip code. When sending a campaign, a collection can examine the items in a catalog, searching for data specifically relevant to each of the campaign's recipients. This data can then be used to personalize the message before it is sent.

For information about APIs that can be used to work with catalogs and catalog items, take a look at the following documents:

- API Overview and Sample Payloads
- API documentation

Creating and modifying catalog items

Iterable's web interface provides three ways to create and edit catalog items:

JSON editor

Useful for editing single catalog items, the JSON editor provides a point-and-click interface for creating and editing catalog items.

Text editor

The text editor can be used to input raw JSON that represents a catalog item.

CSV files

Uploading a CSV file makes it possible to create up to 1000 catalog items at once. However, CSV files may only include fields of type string, date, boolean, long, and double.

To view the items in a catalog:

1. Navigate to **Content > Catalogs** and click the name of the catalog you'd like to view.
2. Click the **JSON / Tabular** toggle to switch between the two ways of viewing a catalog's items. **JSON** is optimized for viewing the fields in a single catalog item

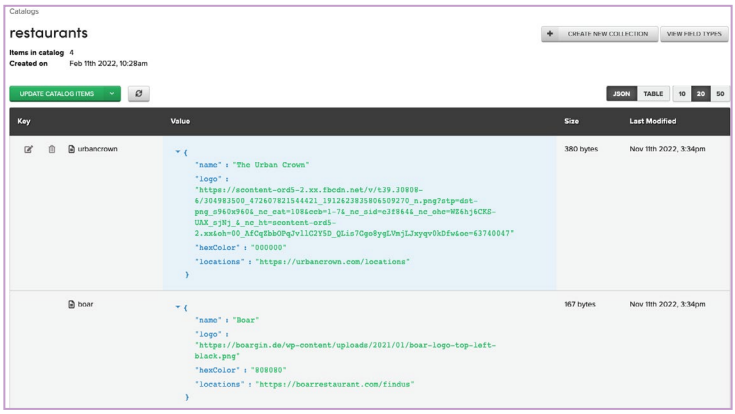


Figure 6: Example Catalog

To display data from a catalog in a message, use Handlebars to either:

- Embed a collection
- Directly reference a catalog item by name (the name can be known ahead of time, or it can be a value stored on a user's Iterable profile).

Catalog Collections

Embedding a collection in a message

One way to display catalog data in a message is to embed a collection. To do this, follow these steps:

When building a collection, click the Use In Template button to retrieve a Handlebars snippet that can be used to reference its data.

For example, the following Handlebars snippet embeds the **DeliveryBelowDinnerBudget** collection in a template:

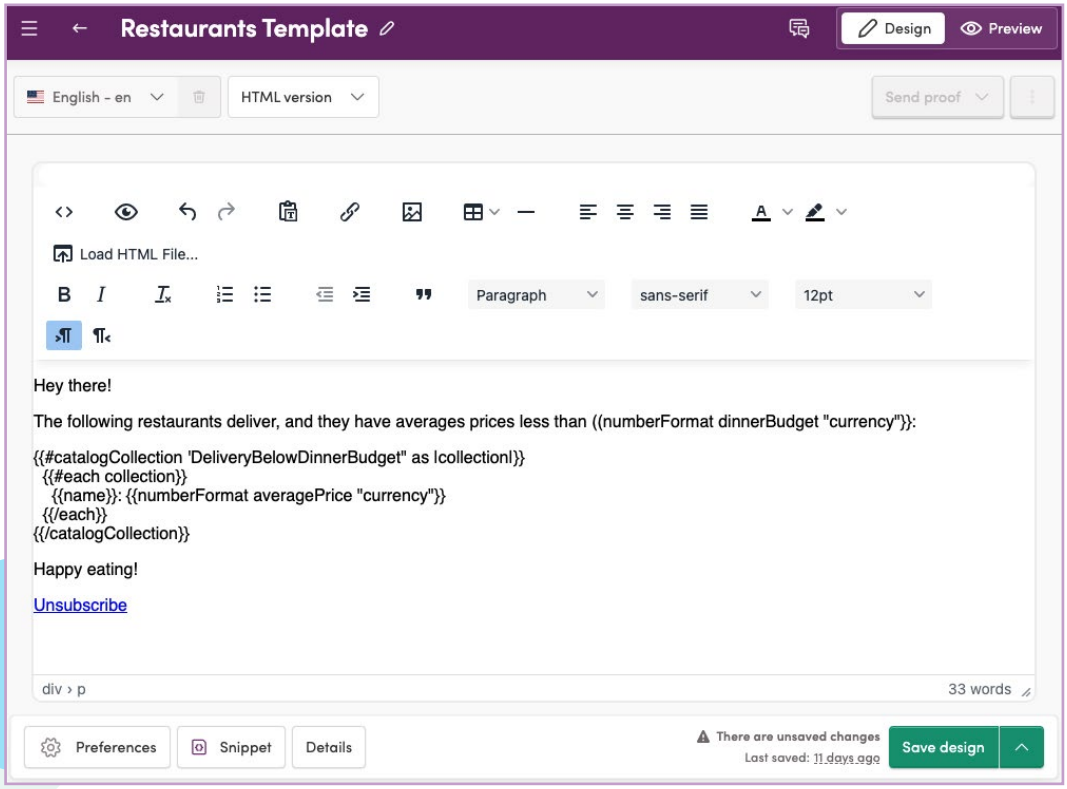


Figure 7:

```
{{#catalogCollection "DeliveryBelowDinnerBudget" as |collection|}}
<!-- Your HTML goes here -->
{{/catalogCollection}}
```

Localization

Iterable supports message localization, meaning that a single campaign can be sent to users around the world with a native language version of the template being selected for each user. Message localization is available for all native messaging channels.

It should be noted that this is not a language translator. You'll need to input the template content for each language.

To get started with this feature, navigate to the **Settings > Project** page within Iterable. There is an option to "+ADD LOCALE"

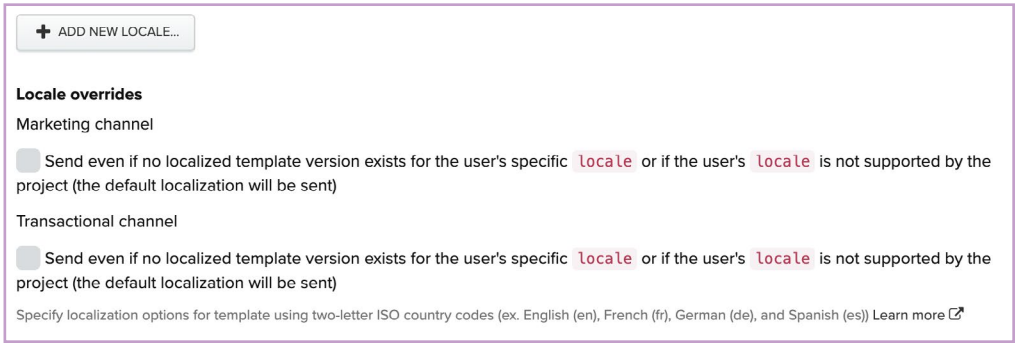


Figure 8: Project settings page with "+ADD LOCALE"

You can add as many locales as needed. Locales can be called whatever you like as long as the names correspond exactly to the value you provide in the locale field on the Iterable User Profile. We recommend using the two-letter ISO country codes.

After updating the Project Settings, you will now see a locale dropdown at the top right section of the template editor. Each of your locales will be displayed in the drop down. If you want to add localized variations of your template, simply select the locale from the dropdown and enter your content.

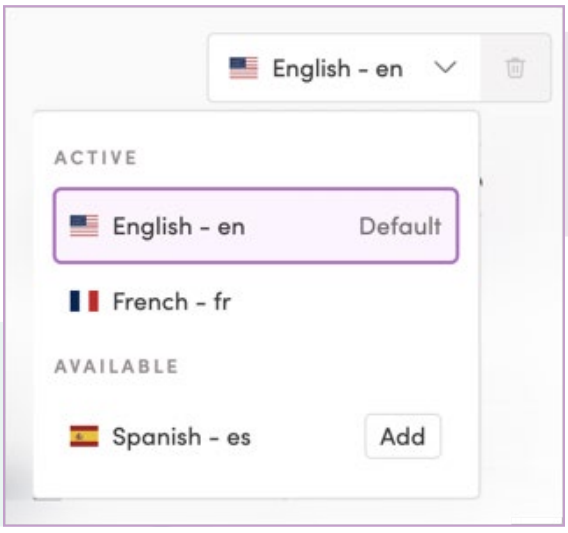


Figure 9: Template dropdown screenshot

At message send time, the correct version of the template will be sent to each user based on the locale field specified on the User Profile.

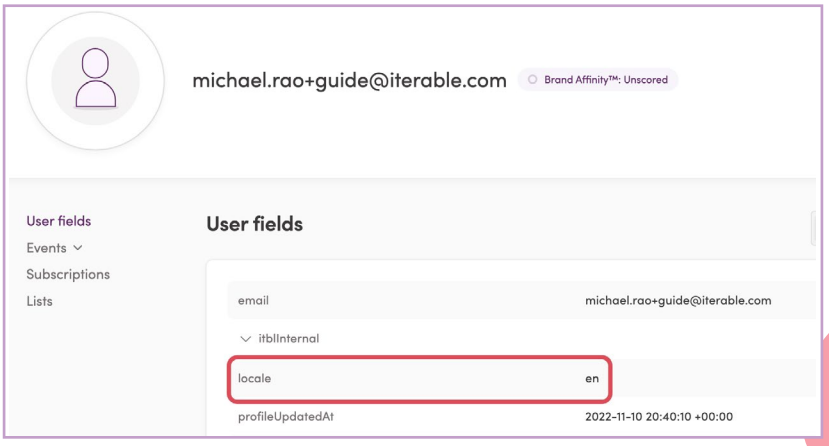


Figure 10: User profile screenshot

SMS

You can use Iterable to send marketing and transactional SMS and MMS messages. This channel generally has high open rates, and you can use it to interact almost immediately with your audience.

Iterable SMS is a proprietary, first-party SMS solution with a full suite of compliance tools to ensure safe and secure messaging.

In order to send text messages, your users must have a user field with the key of `phoneNumber`.

Phone numbers must also be in the proper format and must include a country code. If the phone number is (650) 392-6753, then the `phoneNumber` field must have a value of `16503926753`.

International phone numbers must include a plus sign (+) in front. For example, for a German number, whose country code is 49, it must be added in the format of `+496503926753`.

A sample JSON payload might be:

```
{
   "email": "jeremy@iterable.com",
   "dataFields": {
      "phoneNumber": "16503926753"
   }
}
```

For an international number, a sample JSON payload might be:

```
{
   "email": "jeremy@iterable.com",
   "dataFields": {
      "phoneNumber": "+4916503926753"
   }
}
```

Third-Party Integration

If you have an existing relationship with Twilio or Telnyx SMS that you would prefer to keep in place, Iterable can integrate directly. Both integrations have full support for SMS and MMS based feedback events, Journey triggers, and Campaigns within Iterable.

Mobile Apps

Mobile channels are an important part of Iterable's omni-channel messaging capabilities. The two mediums currently supported are mobile push messages and in-app messages. This mobile messaging functionality is available to your app in 2 ways: via integration with our native open-source SDKs for iOS and Android, or directly via API.

> **"The mobile SDKs are not required, but will simplify the process of implementing common tasks like token registration, notification tracking, and rendering of in-app messages."**

If you prefer not to include outside packages in your app, Iterable's mobile messaging features can also be accessed directly from your app via network calls to our REST API. See our guide to integrating your mobile app with Iterable without the SDK.

Integration Start

See our iOS SDK Documentation, Android SDK Documentation, and React Native SDK Documentation for more information on implementation.

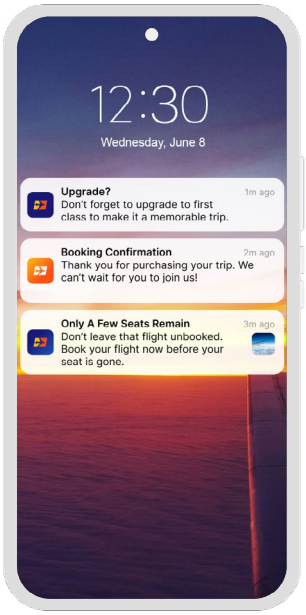
Our Guide for Mobile Engineers is also a great place to find more documentation on setting up and using mobile push and in-app messages.

Mobile Push

Iterable can create and send push notification campaigns to iOS and Android devices, all from a single message template.

Push notifications can be customized with sounds, images, videos, and action buttons (iOS and Android support different options)—all the elements needed to create engaging and valuable content for your users. Iterable's push notification editor provides live previews as you edit, validates errors, and can be used to integrate data stored on a user's Iterable profile, events, datafeed or catalog into a message.

To integrate Iterable into your push campaigns, you will want to follow this guide if you're using Native iOS or Native Android and this guide if you are using a server to server integration or cross-platform mobile framework.



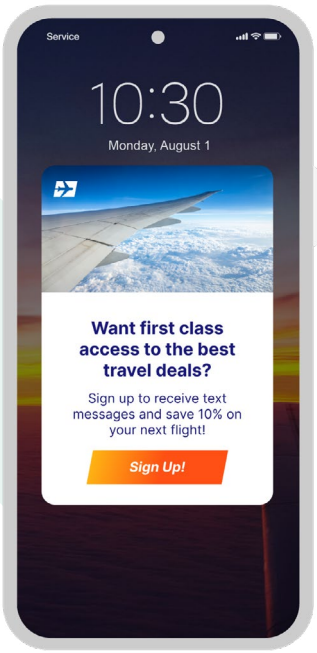
In-App Messaging

In-app messages are custom messages that appear while a mobile app is in the foreground. Unlike push notifications, they do not appear when the app is in the background or closed.

In-app messages can:

- Be created with raw HTML or a drag-and-drop editor.
- Contain buttons for closing the message or navigating to specific content.
- Be sent as part of Blast Campaign, Triggered Campaign, or Journey.
- Be displayed in any order, or prevented from displaying at all.
- Contain metadata that the associated mobile app can inspect.
- Point to new content or functionality in your app, promotions, and other important information.

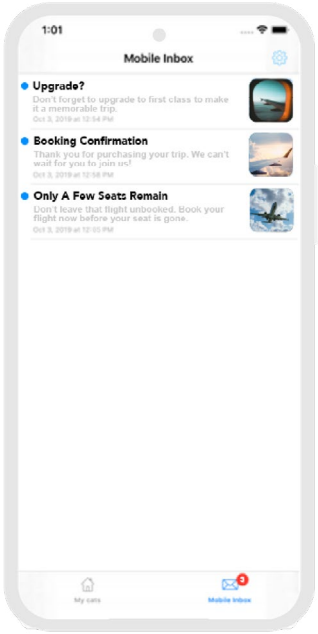
To integrate Iterable into your in-app campaigns, you will want to follow this guide if you're using Native iOS or Native Android and this guide if you are using a server to server integration or cross-platform mobile framework.



Mobile Inbox

Users may sometimes want to save in-app messages to read when they have more time or when they're more relevant. In other cases, you may want to send promotions or announcements in a non-intrusive way, making them available for users to read as they see fit.

To accomplish these things, use Mobile Inbox: a place in your mobile apps for users to receive, store, browse, read and delete in-app messages. You can read more about Iterable's Mobile Inbox here.



In-Browser Messaging

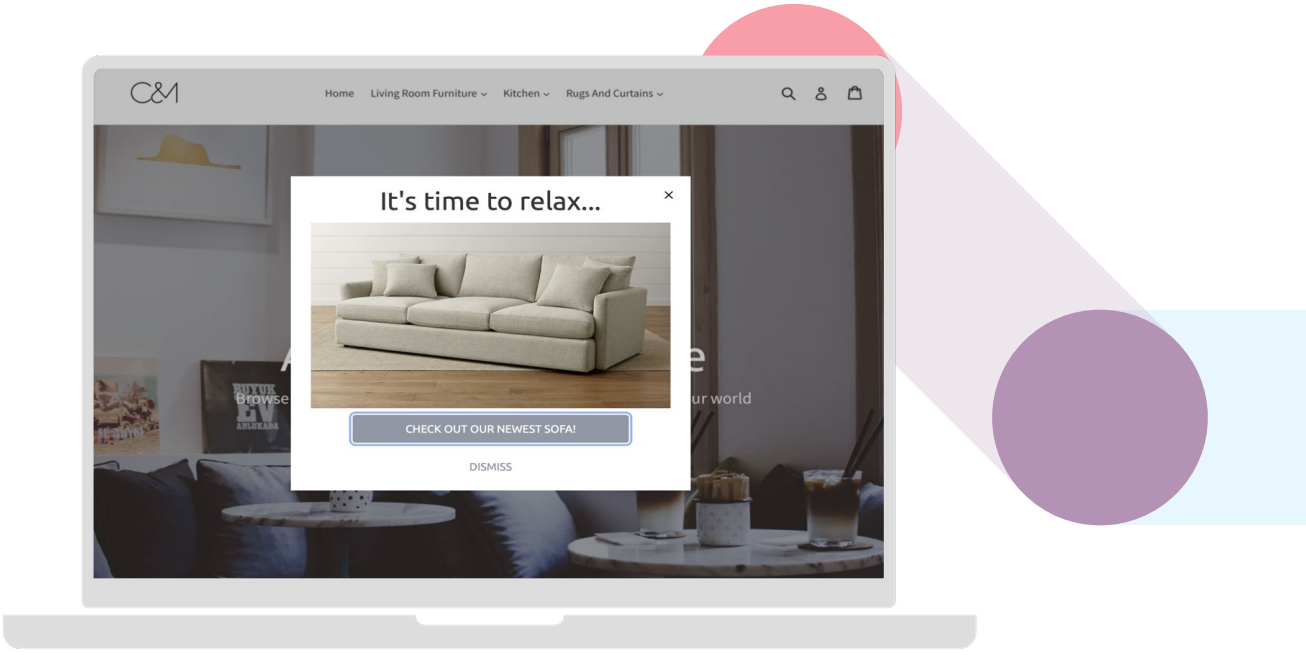
With In-Browser Messaging, you can deliver personalized modal messaging over your desktop and mobile websites, adding a new layer of immersion to the web viewing experience with the ability to trigger unique messaging experiences targeting individual users as they browse your website.

Through easy integration with a robust web-specific SDK, secured with JWT authentication, In-Browser Messaging provides message presentation, event & purchase tracking, and leverages Iterable's powerful data and journey capabilities.

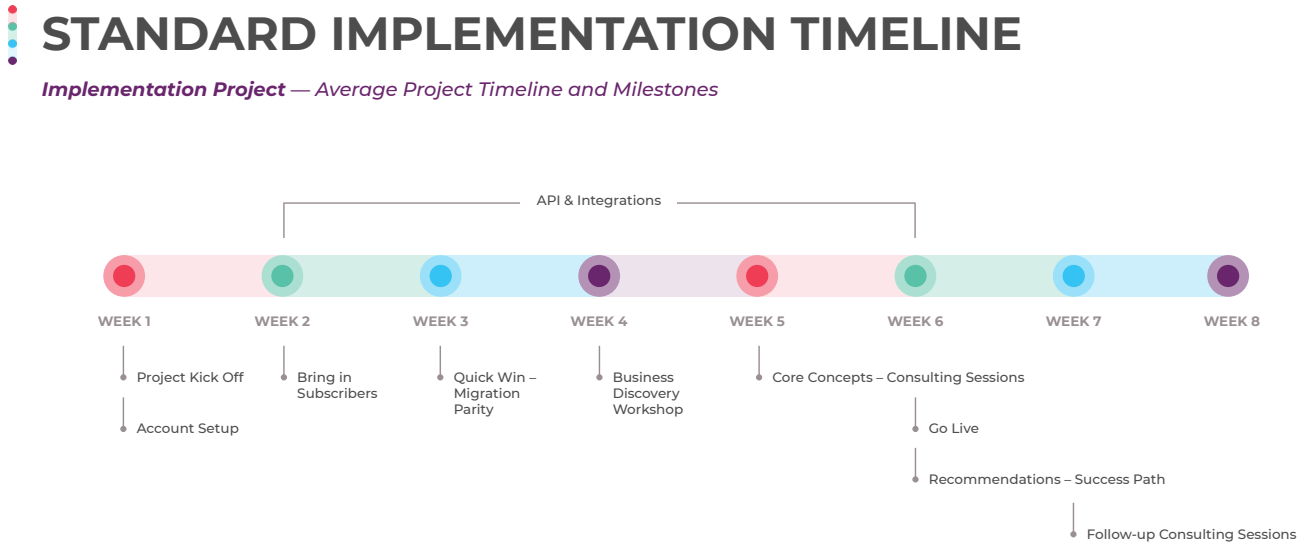
With In-Browser Messaging, you can:

- Create messages with raw HTML and a drag-and-drop editor.
- Set message configurations, like priority, animations, expiration, background overlay, and more.
- Set custom margins and display intervals, auto-generate close buttons, and more.
- Send messages as part of a Blast Campaign, Triggered Campaign, or Journey.
- Include metadata that the associated website can inspect.
- Override an In-App or In-Browser Message from being seen after a certain action is taken on any channel.

To integrate Iterable into your in-browser campaigns, you will want to reference this guide, which describes the capabilities and installation of our Javascript SDK.



Appendix 1: Sample Integration Timeline



Note: :This is an estimated timeline for a basic implementation (est. 60 days). Some can take longer or shorter based on resources, constraints, and goals. For more complex environments, we can scope that out with your team to give you a more accurate timeline specific to your needs.

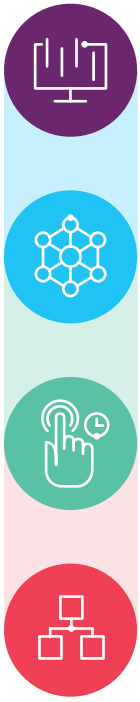
Appendix 2: Limits in Iterable

Below is a consolidated list of limits for key data import and export resources. The comprehensive list of limits for every API endpoint is available on our API Docs page, and it should be considered the definitive source for resource limits.

About Iterable

Iterable is the growth marketing platform that enables brands to create, execute and optimize cross channel campaigns with unparalleled data flexibility.

Leading brands, like DoorDash, SeatGeek, and Box, choose Iterable to power world-class customer engagement throughout the entire lifecycle.



Data Flexibility at Your Fingertips

Access real-time user, behavioral and event data to trigger personalized messaging at virtually unlimited scale. Support an audience of millions while appealing to each subscriber's unique preferences.

Unified Brand Experience

Orchestrate seamless customer engagement across email, mobile push, SMS, in-app, web push, direct mail and more throughout every lifecycle stage, from activation to re-engagement.

Agile Iteration & Optimization

Launch, measure and fine-tune campaigns with ease to deliver more relevant messaging faster than the competition. Experiment and iterate on-demand to determine the right content, channel and cadence for each user.

Customer Journey Mapping

Visualize the entire customer journey and build sophisticated, cross-channel segments and campaigns with Iterable's intuitive, drag-and-drop Workflow Studio.

If you want to learn more about Iterable, please request a demo:

Request Demo

TRUSTED BY

DOORDASH box fabfitfun SEAT GEEK Fender STRAVA