
Ahoy Email

First-party email analytics for Rails

:fire: For web and native app analytics, check out Ahoy

:bullettrain_side: To manage email subscriptions, check out Mailkick



Installation

Add this line to your application's Gemfile:

```
1 gem "ahoy_email"
```

Getting Started

There are three main features, which can be used independently:

- Message history
- UTM tagging
- Click analytics

Message History

To encrypt email addresses with Lockbox, install Lockbox and Blind Index and run:

```
1 rails generate ahoy:messages --encryption=lockbox
2 rails db:migrate
```

To use Active Record encryption (Rails 7+, experimental), run:

```
1 rails generate ahoy:messages --encryption=activerecord
2 rails db:migrate
```

If you prefer not to encrypt data, run:

```
1 rails generate ahoy:messages --encryption=none
2 rails db:migrate
```

Then, add to mailers:

```
1 class CouponMailer < ApplicationMailer
2   has_history
3 end
```

Use the `Ahoy::Message` model to query messages:

```
1 Ahoy::Message.last
```

Use `only` and `except` to limit actions

```
1 class CouponMailer < ApplicationMailer
2   has_history only: [:welcome]
3 end
```

To store history for all mailers, create `config/initializers/ahoy_email.rb` with:

```
1 AhoyEmail.default_options[:message] = true
```

Users

By default, Ahoy Email tries `@user` then `params[:user]` then `User.find_by(email: message.to)` to find the user. You can pass a specific user with:

```
1 class CouponMailer < ApplicationMailer
2   has_history user: -> { params[:some_user] }
3 end
```

The user association is polymorphic, so use it with any model.

To get all messages sent to a user, add an association:

```
1 class User < ApplicationRecord
2   has_many :messages, class_name: "Ahoy::Message", as: :user
3 end
```

And run:

```
1 user.messages
```

Extra Data

Add extra data to messages. Create a migration like:

```
1 class AddCouponIdToAhoyMessages < ActiveRecord::Migration[7.1]
2   def change
3     add_column :ahoy_messages, :coupon_id, :integer
```

```
4   end
5 end
```

And use:

```
1 class CouponMailer < ApplicationMailer
2   has_history extra: {coupon_id: 1}
3 end
```

You can use a proc as well.

```
1 class CouponMailer < ApplicationMailer
2   has_history extra: -> { {coupon_id: params[:coupon].id} }
3 end
```

Options

Set global options

```
1 AhoyEmail.default_options[:user] = -> { params[:admin] }
```

Use a different model

```
1 AhoyEmail.message_model = -> { UserMessage }
```

Or fully customize how messages are tracked

```
1 AhoyEmail.track_method = lambda do |data|
2   # your code
3 end
```

Data Retention

Delete older data with:

```
1 Ahoy::Message.where("created_at < ?", 1.year.ago).in_batches.delete_all
```

Delete data for a specific user with:

```
1 Ahoy::Message.where(user_id: 1, user_type: "User").in_batches.
  delete_all
```

UTM Tagging

Use UTM tagging to attribute visits or conversions to an email campaign. Add UTM parameters to links with:

```
1 class CouponMailer < ApplicationMailer
2   utm_params
3 end
```

The defaults are:

- `utm_medium` - email
- `utm_source` - the mailer name like `coupon_mailer`
- `utm_campaign` - the mailer action like `offer`

You can customize them with:

```
1 class CouponMailer < ApplicationMailer
2   utm_params utm_campaign: -> { "coupon#{params[:coupon].id}" }
3 end
```

Use `only` and `except` to limit actions

```
1 class CouponMailer < ApplicationMailer
2   utm_params only: [:welcome]
3 end
```

Skip specific links with:

```
1 <%= link_to "Go", some_url, data: {skip_utm_params: true} %>
```

Click Analytics

You can track click-through rate to see how well campaigns are performing. Stats can be stored in your database, Redis, or any other data store.

Database Run:

```
1 rails generate ahoy:clicks
2 rails db:migrate
```

And create `config/initializers/ahoy_email.rb` with:

```
1 AhoyEmail.subscribers << AhoyEmail::DatabaseSubscriber
2 AhoyEmail.api = true
```

Redis Add this line to your application's Gemfile:

```
1 gem "redis"
```

And create `config/initializers/ahoy_email.rb` with:

```
1 # pass your Redis client if you already have one
2 AhoyEmail.subscribers << AhoyEmail::RedisSubscriber.new(redis: Redis.
  new)
3 AhoyEmail.api = true
```

Other Create `config/initializers/ahoy_email.rb` with:

```
1 class EmailSubscriber
2   def track_send(data)
3     # your code
4   end
5
6   def track_click(data)
7     # your code
8   end
9
10  def stats(campaign)
11    # optional, for AhoyEmail.stats
12  end
13 end
14
15 AhoyEmail.subscribers << EmailSubscriber
16 AhoyEmail.api = true
```

Usage

Add to mailers you want to track

```
1 class CouponMailer < ApplicationMailer
2   track_clicks campaign: "my-campaign"
3 end
```

If storing stats in the database, the mailer should also use `has_history`

Use `only` and `except` to limit actions

```
1 class CouponMailer < ApplicationMailer
2   track_clicks campaign: "my-campaign", only: [:welcome]
3 end
```

Or make it conditional

```
1 class CouponMailer < ApplicationMailer
2   track_clicks campaign: "my-campaign", if: -> { params[:user].opted_in
3     ? }
4 end
```

You can also use a proc

```
1 class CouponMailer < ApplicationMailer
2   track_clicks campaign: -> { "coupon-#{action_name}" }
3 end
```

Skip specific links with:

```
1 <%= link_to "Go", some_url, data: {skip_click: true} %>
```

By default, unsubscribe links are excluded. To change this, use:

```
1 AhoyEmail.default_options[:unsubscribe_links] = true
```

You can specify the domain to use with:

```
1 AhoyEmail.default_options[:url_options] = {host: "mydomain.com"}
```

Stats

Get stats for a campaign

```
1 AhoyEmail.stats("my-campaign")
```

Upgrading

2.0

Ahoy Email 2.0 brings a number of changes. Here are a few to be aware of:

- The `to` field is encrypted by default for new installations. If you'd like to encrypt an existing installation, install Lockbox and Blind Index and follow the Lockbox instructions for migrating existing data.

For the model, create `app/models/ahoy/message.rb` with:

```
1 class Ahoy::Message < ActiveRecord::Base
2   self.table_name = "ahoy_messages"
3
4   belongs_to :user, polymorphic: true, optional: true
```

```
5
6   encrypts :to, migrating: true
7   blind_index :to, migrating: true
8 end
```

- The `track` method has been broken into:
 - `has_history` for message history
 - `utm_params` for UTM tagging
 - `track_clicks` for click analytics
- Message history is no longer enabled by default. Add `has_history` to individual mailers, or create an initializer with:

```
1 AhoyEmail.default_options[:message] = true
```

- For privacy, open tracking has been removed.
- For clicks, we encourage you to try aggregate analytics to measure the performance of campaigns. You can use a library like Rollup to aggregate existing data, then drop the `token` and `clicked_at` columns.

To keep individual analytics, use `has_history` and `track_clicks campaign: false` and create an initializer with:

```
1 AhoyEmail.save_token = true
2 AhoyEmail.subscribers << AhoyEmail::MessageSubscriber
```

If you use a custom subscriber, `:message` is no longer included in click events. You can use `:token` to query the message if needed.

- Users are shown a link expired page when signature verification fails instead of being redirected to the homepage when `AhoyEmail.invalid_redirect_url` is not set

History

[View the changelog](#)

Contributing

Everyone is encouraged to help improve this project. Here are a few ways you can help:

- Report bugs
- Fix bugs and submit pull requests

-
- Write, clarify, or fix documentation
 - Suggest or add new features

To get started with development:

```
1 git clone https://github.com/ankane/ahoy_email.git
2 cd ahoy_email
3 bundle install
4 bundle exec rake test
```