



Recipies

Infinite Scroll

```
def index

  @blocked_users = current_user.blocked_users
  @pagy, @blocked_users = pagy(@blocked_users, limit: 20)

  respond_to do |format|
    format.turbo_stream
    format.html do
      render("users/blocked_users/index", layout: "settings")
    end
  end
end
```

index.html.erb

```
<div class="bg-white p-4 flex flex-col gap-2">
  <h3 class="font-medium text-sm">List of blocked users</h3>
  <div class="flex flex-col gap-2 relative">
    <%= turbo_frame_tag "blocked_users" do %>
      <% @blocked_users.each do |blocked_user| %>
        <%= render "users/blocked_users/user", blocked_user: blocked_user %>
      <% end %>

      <% if @pagy.next.present? %>
        <%= turbo_frame_tag :pagination,
          loading: :lazy,
          src: settings_blocked_users_path(page: @pagy.next, format: :turbo_stream),
          class: "" do %>
          <%= render "shared/skeleton/notification", count: 6 %>
        <% end %>
      <% end %>
    <% end %>

    <% if current_user.blocked_users.empty? %>
      <p class="text-sm">You don't have any blocked users</p>
    <% end %>
  </div>
</div>
```

index.turbo_stream.erb

```
<%= turbo_stream.remove "pagination" %>
<%= turbo_stream.append "blocked_users" do %>
  <% @blocked_users.each do |blocked_user| %>
    <%= render "users/blocked_users/user", blocked_user: blocked_user %>
  <% end %>
<% end %>

<% if @pagy.next.present? %>
  <%= turbo_stream.append "blocked_users" do %>
    <%= turbo_frame_tag "pagination",
      loading: :lazy,
      src: settings_blocked_users_path(page: @pagy.next, format: :turbo_stream),
      class: "" do %>
      <%= render "shared/skeleton/notification", count: 6 %>
    <% end %>
  <% end %>
<% end %>
```

```
<% end %>
<% end %>
```

Testing

Request Specs

How to open automatically request specs responses in the browser:

1. gem "launchy"
2. test.rb - config.action_controller.asset_host = "<http://localhost:3000>"
3. inside spec, add `save_and_open_page`

If it breaks feature specs:

```
# rails_helper.rb

Rails.root.glob("spec/support/**/*.rb").sort_by(&:to_s).each { |f| require f }
```



```
# spec/support/feature_ext.rb

RSpec.configure do |config|
  # Apply this around-hook only to feature specs
  config.around(:each, type: :feature) do |example|
    old_asset_host = Rails.application.config.action_controller.asset_host
    Rails.application.config.action_controller.asset_host = nil
    ActionController::Base.asset_host = nil
    example.run
  ensure
    Rails.application.config.action_controller.asset_host = old_asset_host
    ActionController::Base.asset_host = old_asset_host
  end
end
```

View Components

Tips & Tricks

Use `render?` to block rendering of the component.

```
module Chat
  class ContentViewerComponent < ApplicationViewComponent
    option :message

    def render?
      message.content.size > 1
    end
  end
end
```