

In this lecture, we will discuss...

- ✧ Pagination

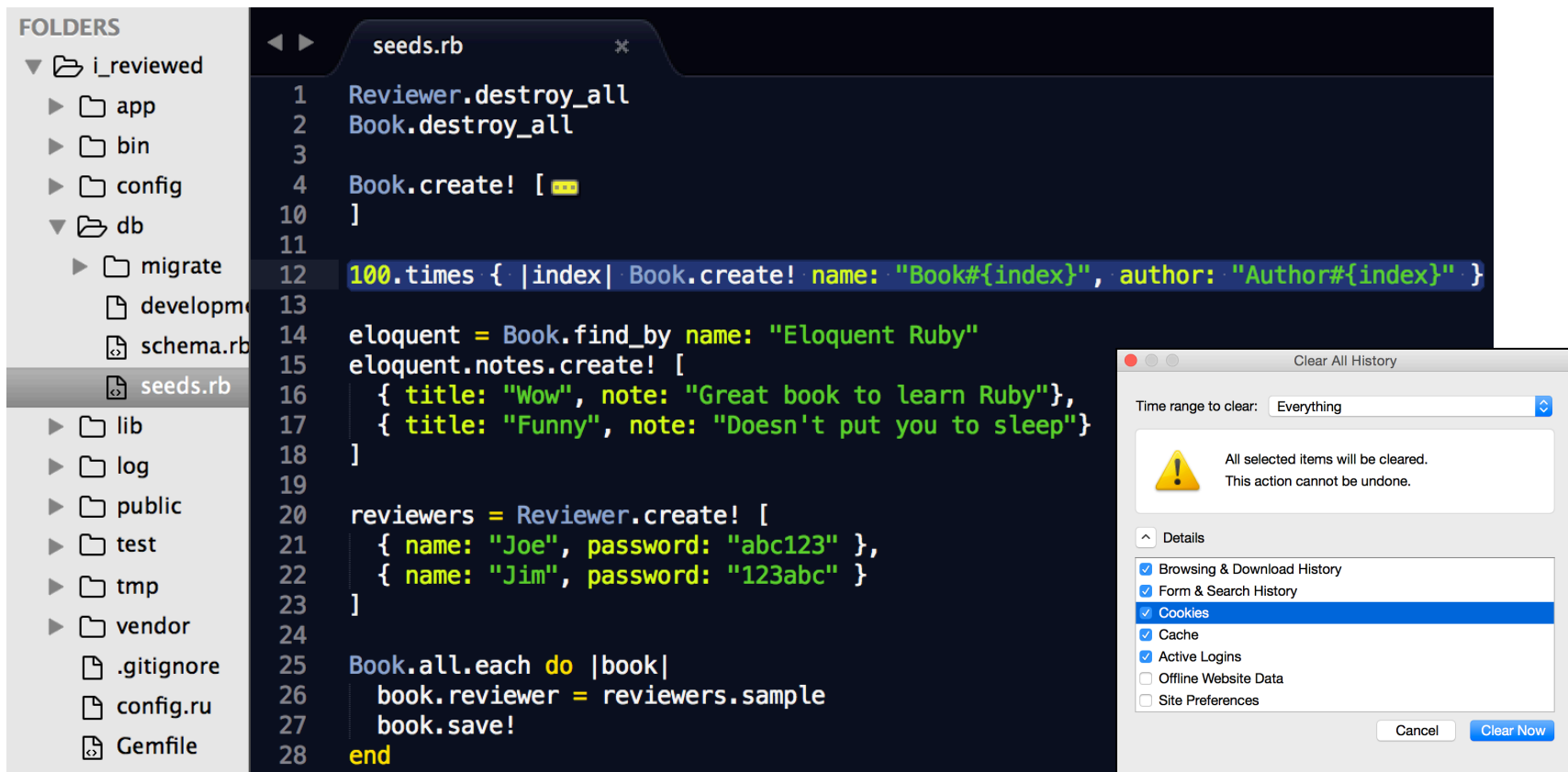


Pagination Summary

1. Shut down your server
2. Include `will_paginate` gem
3. Run `$bundle`
4. One line of code in the controller
5. One line of code in your view
6. Restart your server



Seed (Many More) Books

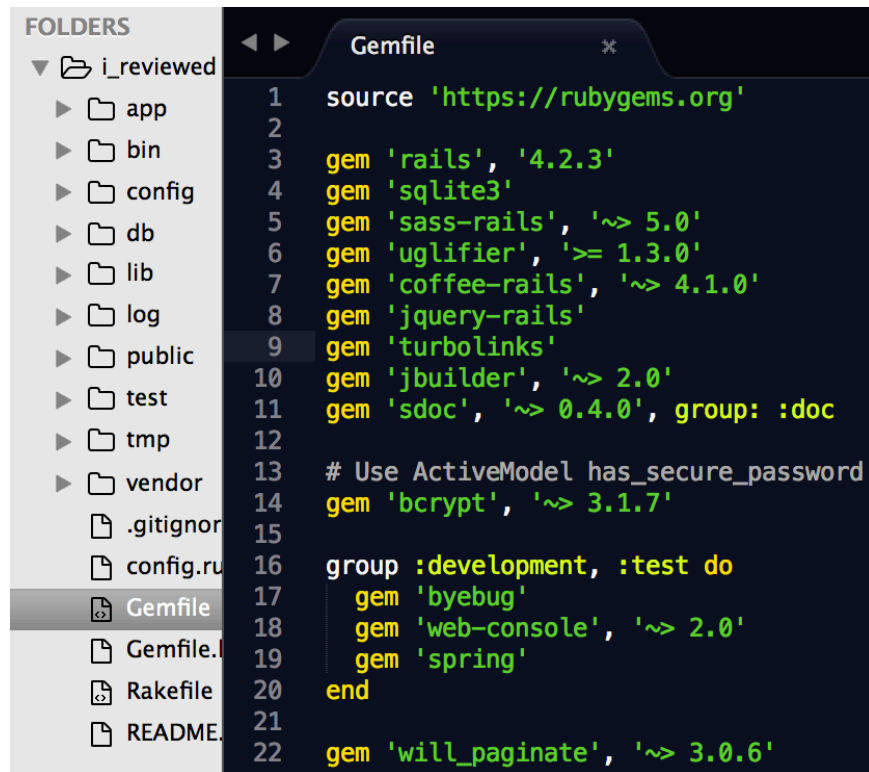


The image shows a code editor with a file explorer on the left and a Ruby script in the main area. The file explorer shows a project structure with folders like `i_reviewed`, `app`, `bin`, `config`, `db`, `lib`, `log`, `public`, `test`, `tmp`, and `vendor`. The `seeds.rb` file is selected. The Ruby script in `seeds.rb` is as follows:

```
1 Reviewer.destroy_all
2 Book.destroy_all
3
4 Book.create! [
10 ]
11
12 100.times { |index| Book.create! name: "Book#{index}", author: "Author#{index}" }
13
14 eloquent = Book.find_by name: "Eloquent Ruby"
15 eloquent.notes.create! [
16   { title: "Wow", note: "Great book to learn Ruby" },
17   { title: "Funny", note: "Doesn't put you to sleep" }
18 ]
19
20 reviewers = Reviewer.create! [
21   { name: "Joe", password: "abc123" },
22   { name: "Jim", password: "123abc" }
23 ]
24
25 Book.all.each do |book|
26   book.reviewer = reviewers.sample
27   book.save!
28 end
```

Overlaid on the bottom right is a 'Clear All History' dialog box. It has a title bar with standard window controls. Below the title bar is a 'Time range to clear:' dropdown menu set to 'Everything'. A warning message with a yellow triangle icon states: 'All selected items will be cleared. This action cannot be undone.' Below this is a 'Details' section with a list of items to be cleared, each with a checkbox: ☒ Browsing & Download History, ☒ Form & Search History, ☒ Cookies (highlighted), ☒ Cache, ☒ Active Logins, ☐ Offline Website Data, and ☐ Site Preferences. At the bottom right of the dialog are 'Cancel' and 'Clear Now' buttons.

will_paginate in the Gemfile



The image shows a code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with folders like app, bin, config, db, lib, log, public, test, tmp, vendor and files like .gitignore, config.ru, Gemfile, Gemfile.lock, Rakefile, and README. The Gemfile is selected and its contents are displayed in the code editor. The code in the Gemfile is as follows:

```
1 source 'https://rubygems.org'
2
3 gem 'rails', '4.2.3'
4 gem 'sqlite3'
5 gem 'sass-rails', '~> 5.0'
6 gem 'uglifier', '>= 1.3.0'
7 gem 'coffee-rails', '~> 4.1.0'
8 gem 'jquery-rails'
9 gem 'turbolinks'
10 gem 'jbuilder', '~> 2.0'
11 gem 'sdoc', '~> 0.4.0', group: :doc
12
13 # Use ActiveRecord has_secure_password
14 gem 'bcrypt', '~> 3.1.7'
15
16 group :development, :test do
17   gem 'byebug'
18   gem 'web-console', '~> 2.0'
19   gem 'spring'
20 end
21
22 gem 'will_paginate', '~> 3.0.6'
```

BooksController

books_controller.rb

```
class BooksController < ApplicationController
  before_action :set_book, only: [:show, :edit, :update, :destroy]

  def index
    @books = current_user.books.paginate(page: params[:page], per_page: 10)
  end
end
```

~/sandbox/i_reviewed\$ rails c

Loading development environment (Rails 4.2.3)

irb(main):001:0> Reviewer.first.books.paginate(page: 3, per_page: 10)

Reviewer Load (0.1ms) SELECT "reviewers".* FROM "reviewers" ORDER BY "reviewers"."id" ASC LIMIT 1

Book Load (0.2ms) SELECT "books".* FROM "books" WHERE "books"."reviewer_id" = ? LIMIT 10 OFFSET 20 [["reviewer_id", 6]]

(0.1ms) SELECT COUNT(*) FROM "books" WHERE "books"."reviewer_id" = ? [["reviewer_id", 6]]



books/index.html.erb

FOLDERS

- ▼ i_reviewed
 - ▼ app
 - assets
 - controllers
 - helpers
 - mailers
 - models
 - ▼ views
 - ▼ books
 - _form.html.erb
 - edit.html.erb
 - index.html.erb**
 - index.json.jbuild
 - new.html.erb
 - show.html.erb
 - show.json.jbuild
 - layouts
 - notes
 - sessions
 - bin

index.html.erb

```
1 <h1>Listing Books</h1>
2
3 <table>
4   <thead> ...
10  </thead>
11
12  <tbody>
13    <%= @books.each do |book| %>
14      <tr>
15        <td><%= book.name %></td>
16        <td><%= book.author %></td>
17        <td><%= link_to 'Show', book %></td>
18        <td><%= link_to 'Edit', edit_book_path(book) %></td>
19        <td><%= link_to 'Destroy', book, method: :delete, data: { confirm: 'Are you sure?' } %></td>
20      </tr>
21    <%= end %>
22  </tbody>
23 </table>
24 <p/>
25
26 <%= will_paginate @books %>
27
28 <br>
29
30 <%= link_to 'New Book', new_book_path %>
```



Pagination Results

Listing Books

Logged in as Joe | [Logout](#)

Name	Author	
Eloquent Ruby	Russ Olsen	Show Edit Destroy
Book1	Author1	Show Edit Destroy
Book6	Author6	Show Edit Destroy
Book8	Author8	Show Edit Destroy
Book12	Author12	Show Edit Destroy
Book21	Author21	Show Edit Destroy
Book22	Author22	Show Edit Destroy
Book23	Author23	Show Edit Destroy
Book26	Author26	Show Edit Destroy
Book27	Author27	Show Edit Destroy

← Previous / [2](#) [3](#) [4](#) [5](#) Next →








[New Book](#)

localhost:3000/?page=3

REMEMBER:
Restart the server after
installing new gem



Pagination Results

  localhost:3000/?page=3    Search  

Listing Books

Logged in as Joe | [Logout](#)

Name	Author	
Book49	Author49	Show Edit Destroy
Book50	Author50	Show Edit Destroy
Book51	Author51	Show Edit Destroy
Book58	Author58	Show Edit Destroy
Book62	Author62	Show Edit Destroy
Book63	Author63	Show Edit Destroy
Book69	Author69	Show Edit Destroy
Book70	Author70	Show Edit Destroy
Book71	Author71	Show Edit Destroy
Book73	Author73	Show Edit Destroy

[← Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next →](#)

[New Book](#)



Summary

- ✧ `will_paginate` gem makes it easy to paginate your results
- ✧ Scope your Active Record call from controller and pass in a “which page?” parameter from your view after installing `will_paginate` is all you need

What's Next?

- ✧ Deploying the app to Heroku

