Beginning Ruby on Rails

Session Six

Advanced Topics

- A new app from scratch
- User Sessions
- More Associations
- Select Boxes

The New App

A simple “issue tracker”

Features

- Anonymous users will be able to register for an account and log in.
- Logged in users will be able to post issues.
- Logged in users will also be able to add notes and assign issues to other users.

Models

- We will need to create these models:
  - Issue
  - Note
  - User

Creating the App

- rails new issues
- cd issues
- bundle install
- bundle exec rake db:create

Adding Models

- rails generate scaffold issue title:string description:text status:string
- rails generate model note body:text issue:references
- edit app/models/issue.rb
- bundle exec rake db:migrate

Setup Routes

- Edit config/routes.rb
- Notes are nested inside Issues
  - Just like Comments and Posts
- root :to => “issues#index”
- Delete public/index.html

Update Views

- You can use views similar to the blog
- Obviously, change post to issue and comment to note
- https://github.com/anthonylewis/issues

Sessions

Web servers have terrible memory

Sessions

- The web is “stateless”
- We generate a session id for each user so the server can remember who they are.
- The session id is stored in a cookie that is passed to the server with each request.
- This sounds like a lot of work...

Devise

- https://github.com/plataformatec/devise
- A flexible Rails authentication system
- Takes care of user registration, log in, log off, password resets, etc.
- Devise is a Rails Engine - it has models, views and controllers

Installing Devise

- Edit Gemfile
  - gem ‘devise’
- bundle install
- rails generate devise:install
- rails generate devise user

Configuring Devise

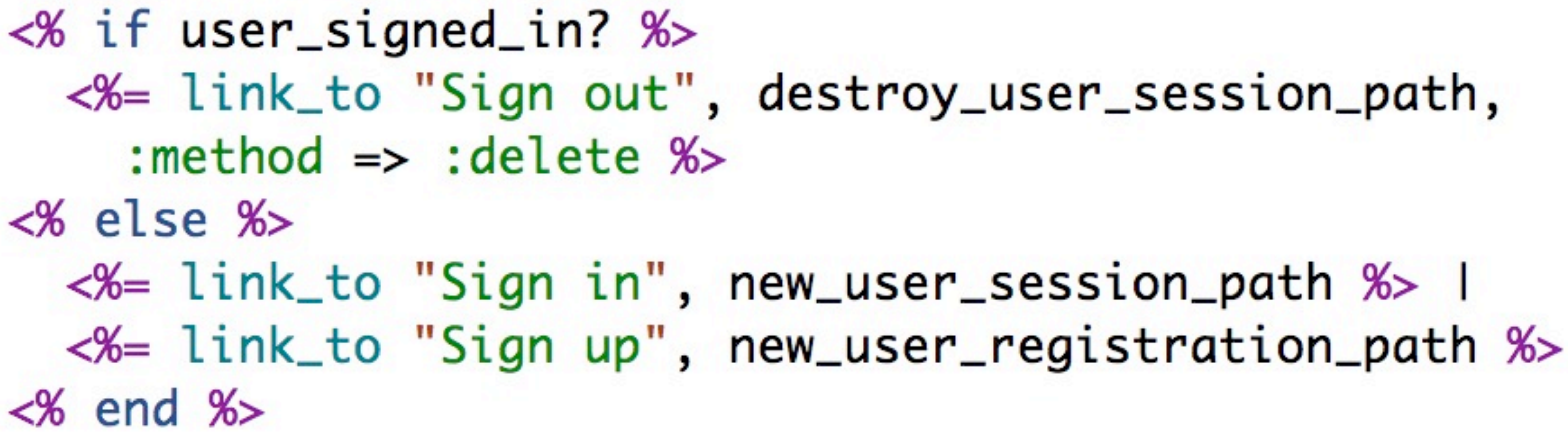
- Make sure you have a root route
- Make sure your views include flash messages (notice and alert)
- Customize the User model and migration for your application
- bundle exec rake db:migrate

Devise Helpers

- authenticate_user!
- user_signed_in?
- current_user
- user_session

Update View

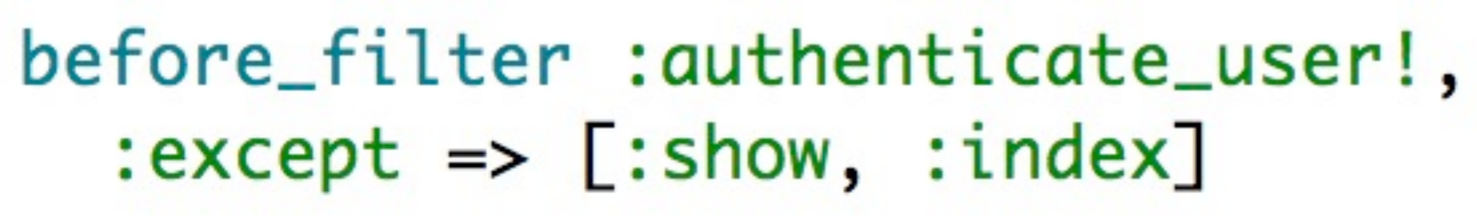
- Edit app/views/layouts/application.html.erb



```
<% if user_signed_in? %>
  <%= link_to "Sign out", destroy_user_session_path,
    :method => :delete %>
<% else %>
  <%= link_to "Sign in", new_user_session_path %> |
  <%= link_to "Sign up", new_user_registration_path %>
<% end %>
```

Update Controller

- Edit app/controllers/issues_controller.rb



```
before_filter :authenticate_user!,
  :except => [:show, :index]
```

More Associations

Everything is connected

User Associations

- Edit model files to reflect these:
  - Issues belong to a User
  - Notes belong to a User
  - A User has many Issues and Notes

Adding Columns

- rails generate migration add_associations
- Edit the migration file:

```
add_column :issues, :user_id, :integer

add_column :notes, :user_id, :integer
```

- bundle exec rake db:migrate

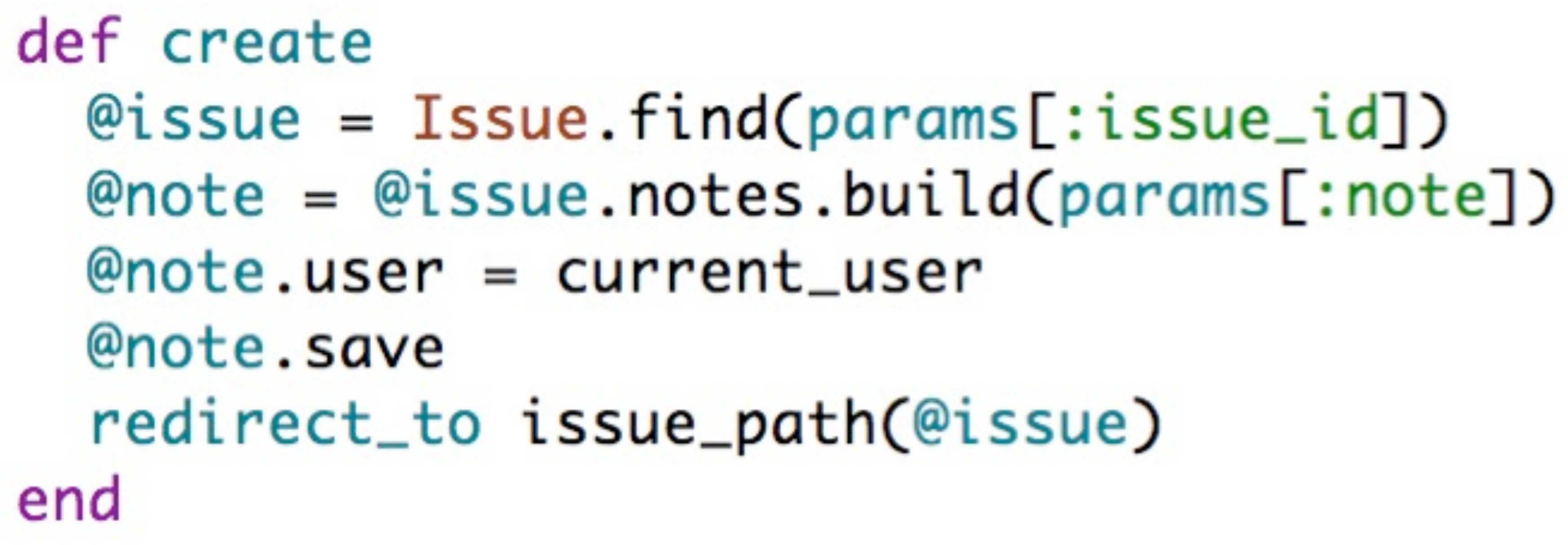
Issues Controller

- Edit app/controllers/issues_controller.rb
- Inside the create action add:

```
@issue.user_id = current_user
```

Notes Controller

- rails generate controller notes
- Edit app/controllers/notes_controller.rb



```
def create
  @issue = Issue.find(params[:issue_id])
  @note = @issue.notes.build(params[:note])
  @note.user = current_user
  @note.save
  redirect_to issue_path(@issue)
end
```

More Associations

- The app should let us assign issues to users
  - A user has many ‘assigned issues’
  - An issue belongs to an ‘assignee’
- rails g migration add_assignee_id_to_issues assignee_id:integer
- bundle exec rake db:migrate

Issue Model

- Edit app/models/issue.rb

```
belongs_to :assignee, :class_name => 'User',
    :foreign_key => 'assignee_id'
```

User Model

- Edit app/models/user.rb

```
has_many :assigned_issues,
    :class_name => 'Issue',
    :foreign_key => 'assignee_id'
```

Select Boxes

You know, drop-downs or whatever

Basics

- Edit app/views/issues/_form.html.erb
- Change line 24 from this:

```
f.text_field :status
```

- To this:

```
f.select :status, [ ‘open’, ‘closed’ ]
```

Refactoring

- Let’s move the options for the select box into the Issue model.
- Edit app/models/issue.rb

```
STATUS_OPTIONS = [‘open’, ‘closed’]
```

Collection Select

- Create a select box with data from a model
- Add a new field to the issue form:

```
f.collection_select :assignee_id,
User.all, :id, :email
```

Homework

Thought exercises

Work On The App

- Clone the app from GitHub
  - https://github.com/anthonylewis/issues
- Make the views a little prettier
- Think about other features to add