Beginning Ruby on Rails

Session Seven

Catching Up

git clone git://github.com/anthonylewis/issues.git

cd issues

bundle install

rake db:migrate

More Rails Gems

- Add features with little code

- Customizing Devise

- Authorization

- File Uploads

- Pagination

- Search

Customizing Devise

Making users work the way we want.

Adding Usernames

- rails generate migration add_username_to_users username:string

- rake db:migrate

- Add attr_accessible :username in model

Changing Views

- rails generate devise:views

- app/views/devise/registration/edit.html.erb

```
<p><%= f.label :username %><br />
<%= f.email_field :username %></p>
```

Update Layout

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

```erb
<% if user_signed_in? %>
  <%= link_to "Profile", edit_user_registration_path %> |
  <%= 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 %>
```

Authorization

Different users can do different things.

CanCan

- Edit Gemfile and add:

  - gem 'cancan'

- bundle install

- rails generate cancan:ability

Adding Roles

- rails generate migration add_role_to_users role:string

- rake db:migrate

Defining Roles

- Edit app/models/user.rb

```ruby
attr_accessible :email, :password,
  :password_confirmation, :remember_me,
  :role

ROLES = ['admin', 'moderator', 'user']
```

Update View

- app/views/devise/registration/edit.html.erb

```erb
<p><%= f.collection_select :role, User::ROLES,
          :to_s, :humanize %></p>
```

Abilities

- Edit app/models/ability.rb

- Abilities

  - :create, :read, :update, :destroy, :manage

- Conditions

  - Hash of attributes

```ruby
def initialize(user)
  user ||= User.new # guest user (not logged in)

  if user.role == 'admin'
    can :manage, :all
  elsif user.role == 'moderator'
    can [:create, :read], Issue
    can :manage, Note
  elsif user.role == 'user'
    can [:create, :read], :all
    can :manage, Issue, :user_id => user.id
  else
    can :read, [Issue, Note]
  end
end
```

Update Views

- Edit app/views/issues/index.html.erb

```erb
<% if can? :create, Issue %>
  <p><%= link_to 'New Issue', new_issue_path %></p>
<% end %>
```

Update Views

- Edit app/views/issues/index.html.erb

```erb
<%= link_to 'Show', issue %>
<% if can? :manage, issue %>
  | <%= link_to 'Edit', edit_issue_path(issue) %>
  | <%= link_to 'Destroy', issue,
      :confirm => 'Are you sure?', :method => :delete %>
<% end %>
```

Update Controllers

- app/controllers/application_controller.rb

```ruby
rescue_from CanCan::AccessDenied do |exception|
  flash[:alert] = "Access denied."
  redirect_to root_url
end
```

Update Controllers

- app/controllers/issues_controller.rb

```
class IssuesController < ApplicationController
  authorize_resource
```

Pagination

What if we have lots of issues?

Will_Paginate

- Edit Gemfile and add:

  - gem 'will_paginate'

- bundle install

Update Controller

- Edit app/controllers/issues_controller.rb

```ruby
def index
  @issues = Issue.paginate
    :page => params[:page], :per_page => 5

  respond_to do |format|
    format.html # index.html.erb
    format.xml { render :xml => @issues }
  end
end
```

Update View

- Edit app/views/issues/index.html.erb

```erb
<%= will_paginate @issues %>
```

Search

Why click through all the pages?

MetaSearch

- Edit Gemfile and add:

  - gem 'meta_search'

- bundle install

Update Controller

- Edit app/controllers/issues_controller.rb

```ruby
def index
  @search = Issue.search params[:search]
  @issues = @search.paginate
    :page => params[:page], :per_page => 5

  respond_to do |format|
    format.html # index.html.erb
    format.xml { render :xml => @issues }
  end
end
```

Update View

- Edit app/views/issues/index.html.erb

```erb
<%= form_for @search do |f| %>
  <p>
    <%= f.text_field :body_contains %>
    <%= f.submit 'Search' %>
  </p>
<% end %>
```

File Uploads

Paperclip

- Edit Gemfile and add:

  - gem "paperclip", "~> 2.3"

- bundle install

Paperclip

- rails generate paperclip user photo

- rake db:migrate

Update Model

- Edit app/models/user.rb

```ruby
attr_accessible :email, :password,
    :password_confirmation, :remember_me,
    :username, :role, :photo

ROLES = ['admin', 'moderator', 'user']

has_attached_file :photo
```

Update Views

- app/views/devise/registration/edit.html.erb

- Form must be 'multipart' to upload files:

```
:html => { :method => :put, :multipart => true }
```

Update Views

- app/views/devise/registration/edit.html.erb

- Add a file field to select the file:

```erb
<p><%= f.file_field :photo %></p>
```

Update Views

- app/views/devise/registration/edit.html.erb

- Add an image tag to display it:

```erb
<%= image_tag current_user.photo.url,
      :size => "120x120" %>
```

Homework

There's always more to do.

Learn More

- Keep up-to-date with RailsCasts

  - http://railscasts.com

- Screencasts covering current topics

- Most are only 10-15 minutes long