Beginning Ruby on Rails

Session Seven

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

```
<% 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

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

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

Update View

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

Abilities

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

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

Update Views

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

```
<% if can? :create, Question %>
  <%= link_to 'New Question', new_question_path %>
<% end %>
```

Update Views

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

```
<%= link_to 'Show', question %>
<% if can? :manage, question %>
  | <%= link_to 'Edit', edit_question_path(question) %>
  | <%= link_to 'Destroy', question,
        :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/questions_controller.rb

```
class QuestionsController < ApplicationController
  authorize_resource
```

Pagination

What if we have lots of questions?

Will_Paginate

- Edit Gemfile and add:
  - gem ‘will_paginate’, ‘~> 3.0.pre2’
- bundle install

Update Controller

- Edit app/controllers/question_controller.rb

```
def index
  @questions = Question.paginate
    :page => params[:page], :per_page => 5

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

Update View

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

```
<%= will_paginate @questions %>
```

Search

Why click through all the pages?

MetaSearch

- Edit Gemfile and add:
  - gem ‘meta_search’
- bundle install

Update Controller

- Edit app/controllers/question_controller.rb

```
def index
  @search = Question.search params[:search]
  @questions = @search.paginate
    :page => params[:page], :per_page => 5

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

Update View

- Edit app/views/questions/index.html.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

```
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:

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

Update Views

- app/views/devise/registration/edit.html.erb
- Add an image tag to display it:

```
<%= 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