

Chapter 12 - Password Reset

- 12.1
 - General sequence for password reset:
 - When a user requests a password reset, find the user by the submitted email address.
 - If the email address exists in the database, generate a reset token and corresponding reset digest.
 - Save the reset digest to the database, and then send an email to the user with a link containing the reset token and user's email address.
 - When the user clicks the link, find the user by email address, and then authenticate the token by comparing it to the reset digest.
 - If authenticated, present the user with the form for changing the password.
 - Generate a controller for the Password Reset Resource - make both a *new* and *edit* action
 - `$ rails generate controller PasswordResets new edit --no-test-framework`
 - Note the flag for skipping the test generation
 - Add the resources for new, create, edit, and update to the config/routes.rb:
 - `resources :password_resets, only: [:new, :create, :edit, :update]`
 - Now add a link on the new.html.erb view for the forgot password action
 - `<%= link_to "(forgot password)", new_password_reset_path %>`
 - Now we need to add two attributes to the Users model:
 - `Reset_digest` : for storing a password reset digest
 - `Reset_sent_at` : for expiring the reset request
 - Do this with the following migrate command:
 - `$ rails generate migration add_reset_to_users reset_digest:string \ reset_sent_at:datetime`
 - Then regular rails `db:migrate`
 - Now we build the Forgot password page, `app/views/password_resets/new.html.erb`
 - See tutorial for html that's added at this step (12.1.2)
 - When a user submits this new page we must find the user by email, and update the password reset token and sent-at timestamp. Then we redirect to the root URL.
 - Some of this work will be handled in the `app/controllers/password_resets_controller.rb`
 - Code will be added to the create method. See tutorial for code (fig 12.5)
 - Additional code must be added to the `app/model/user.rb`
 - `attr_accessor :remember_token, :activation_token, :reset_token`


```
    reset_sent_at < 2.hours.ago
  end
```

- We can now test valid and invalid submissions. Start by generating integration tests for the password reset:
 - `$ rails generate integration_test password_resets`
 - Next ass ALL the code he supplied for the `test/integration/password_resets.test.rb`. See Listing 12.18 (it is a lot of code)
- **12.4**
 - Like chapter 11 we'll use SendGrid on Heroku for this email service. If you already did this in chapter 11 then there's nothing left to do but the normal wrap up procedure