

## Chapter 9 演習問題解答

### 1.

staff/sessions#create アクションを次のように書き換えます:

LIST app/controllers/staff/sessions\_controller.rb

```
class Staff::SessionsController < Staff::Base
```

(2-9行省略)

```
  def create
    @form = Staff::LoginForm.new(params[:staff_login_form])
    if @form.email.present?
      staff_member = StaffMember.find_by(email_for_index: @form.email.downcase)
    end
    if Staff::Authenticator.new(staff_member).authenticate(@form.password)
      if staff_member.suspended?
        flash.now.alert = 'アカウントが停止されています。'
        render action: 'new'
      else
        session[:staff_member_id] = staff_member.id
        flash.notice = 'ログインしました。'
        redirect_to :staff_root
      end
    else
      flash.now.alert = 'メールアドレスまたはパスワードが正しくありません。'
      render action: 'new'
    end
  end
end
```

(以下省略)

Staff::Authenticator のソースコードを次のように書き換えます(10行目を削除):

LIST app/services/staff/authenticator.rb

```
class Staff::Authenticator
  def initialize(staff_member)
    @staff_member = staff_member
  end

  def authenticate(raw_password)
    @staff_member &&
    !@staff_member.suspended? &&
    @staff_member.hash_password &&
    @staff_member.start_date <= Date.today &&
    (@staff_member.end_date.nil? || @staff_member.end_date > Date.today) &&
  end
end
```

```
BCrypt::Password.new(@staff_member.hashed_password) == raw_password
end
end
```

Staff::Authenticator の spec ファイルを次のように書き換えます:

```
LIST spec/services/staff/authenticator_spec.rb
```

```
require 'spec_helper'
```

```
describe Staff::Authenticator do
  describe '#authenticate' do
```

(5-18 行省略)

```
    example '停止フラグが立っていても true を返す' do
      m = build(:staff_member, suspended: true)
      expect(Staff::Authenticator.new(m).authenticate('pw')).to be_truthy
    end
```

(以下省略)

## 2.

app ディレクトリの下にサブディレクトリ forms/admin を作成します:

```
$ mkdir -p app/forms/admin
```

そこに新規ファイル login\_form.rb を次のような内容で作成します:

```
LIST app/forms/admin/login_form.rb
```

```
class Admin::LoginForm
  include ActiveRecord::Model

  attr_accessor :email, :password
end
```

admin/sessions コントローラのスケルトンを生成します:

```
$ bin/rails g controller admin/sessions
```

admin/sessions#new アクションを追加します:

```
LIST app/controllers/admin/sessions_controller.rb
```

```
class Admin::SessionsController < Admin::Base
  def new
    if current administrator
      redirect_to :admin_root
    else
      @form = Admin::LoginForm.new
      render action: 'new'
    end
  end
end
```

staff/sessions#new アクションの ERB テンプレートを app/views/admin/sessions ディレクトリにコ

ピーします:

```
$ cp app/views/staff/sessions/new.html.erb app/views/admin/sessions
```

ERB テンプレートを次のように修正します:

```
LIST app/views/admin/sessions/new.html.erb
```

```
<% @title = 'ログイン' %>
```

```
<div id="login-form">
  <h1><%= @title %></h1>
```

```
  <%= form_for @form, url: :admin_session do |f| %>
(以下省略)
```

スタイルシートを sessions.css.scss を staff ディレクトリから admin ディレクトリにコピーします:

```
$ cp app/assets/stylesheets/staff/sessions.css.scss app/assets/stylesheets/admin
```

admin ディレクトリのスタイルシートを次のように修正します:

```
LIST app/assets/stylesheets/admin/sessions.css.scss
```

```
@import 'colors';
```

```
@import 'dimensions';
```

```
div#wrapper {
  div#container {
    div#login-form {
      width: 400px;
      margin: $wide * 2 auto;
      padding: $wide * 2;
      border-radius: $wide;
      border: solid 4px $dark_magenta;
      background-color: $very_light_gray;

      h1 {
        background-color: transparent;
        color: $very_dark_magenta;
        font-size: $huge;
        font-weight: bold;
      }
    }
  }
}
```

(以下省略)

### 3.

admin/sessions#create アクションを追加します:

```
LIST app/controllers/admin/sessions_controller.rb
```

```
class Admin::SessionsController < Admin::Base
```

(1-9 行省略)

```
  def create
```

```

@form = Admin::LoginForm.new(params[:admin_login_form])
if @form.email.present?
  administrator = Administrator.find_by(email_for_index: @form.email.downcase)
end
if administrator
  if administrator.suspended?
    render action: 'new'
  else
    session[:administrator_id] = administrator.id
    redirect_to :admin_root
  end
else
  render action: 'new'
end
end
end
end

```

#### 4.

admin/sessions#destroy アクションを追加します:

```
LIST app/controllers/admin/sessions_controller.rb
```

```
class Admin::SessionsController < Admin::Base
```

(1-22 行省略)

```

  def destroy
    session.delete(:administrator_id)
    redirect_to :admin_root
  end
end
end

```

#### 5.

app ディレクトリの下にサブディレクトリ services/admin を作成します:

```
$ mkdir -p app/services/admin
```

そこに新規ファイル authenticator.rb を次のような内容で作成します:

```
LIST app/services/admin/authenticator.rb
```

```

class Admin::Authenticator
  def initialize(administrator)
    @administrator = administrator
  end

  def authenticate(raw_password)
    @administrator &&
      @administrator.hashed_password &&
      BCrypt::Password.new(@administrator.hashed_password) == raw_password
  end
end

```

```
end
```

## 6.

spec/factories ディレクトリに新規ファイル administrators.rb を次のような内容で作成します:

```
LIST spec/factories/administrators.rb
FactoryGirl.define do
  factory :administrator do
    sequence(:email) { |n| "admin#{n}@example.com" }
    password 'pw'
    suspended false
  end
end
```

## 7.

spec/services ディレクトリの下にサブディレクトリ admin を作成します:

```
$ mkdir -p spec/services/admin
```

そこに新規ファイル authenticator\_spec.rb を次のような内容で作成します:

```
LIST spec/services/authenticator_spec.rb
require 'spec_helper'

describe Admin::Authenticator do
  describe '#authenticate' do
    example '正しいパスワードなら true を返す' do
      m = build(:administrator)
      expect(Admin::Authenticator.new(m).authenticate('pw')).to be_truthy
    end

    example '誤ったパスワードなら false を返す' do
      m = build(:administrator)
      expect(Admin::Authenticator.new(m).authenticate('xy')).to be_falsey
    end

    example 'パスワード未設定なら false を返す' do
      m = build(:administrator, password: nil)
      expect(Admin::Authenticator.new(m).authenticate(nil)).to be_falsey
    end

    example '停止フラグが立っていても true を返す' do
      m = build(:administrator, suspended: true)
      expect(Admin::Authenticator.new(m).authenticate('pw')).to be_truthy
    end
  end
end
```

spec ファイルを実行してすべてのエグザンプルが成功することを確認してください:

```
$ bin/rspec spec/services/admin/authenticator_spec.rb
....

Finished in 0.65109 seconds
4 examples, 0 failures
```

## 8.

admin/sessions コントローラの create アクションと destroy アクションを次のように変更します:

```
LIST app/controllers/admin/sessions_controller.rb
```

```
class Admin::SessionsController < Admin::Base
  (1-9 行省略)

  def create
    @form = Admin::LoginForm.new(params[:admin_login_form])
    if @form.email.present?
      administrator = Administrator.find_by(email_for_index: @form.email.downcase)
    end
    if Admin::Authenticator.new(administrator).authenticate(@form.password)
      if administrator.suspended?
        flash.now.alert = 'アカウントが停止されています。'
        render action: 'new'
      else
        session[:administrator_id] = administrator.id
        flash.notice = 'ログインしました。'
        redirect_to :admin_root
      end
    else
      flash.now.alert = 'メールアドレスまたはパスワードが正しくありません。'
      render action: 'new'
    end
  end

  def destroy
    session.delete(:administrator_id)
    flash.notice = 'ログアウトしました。'
    redirect_to :admin_root
  end
end
```

app/views/admin/shared/\_header.html.erb を次のように変更します:

```
LIST app/views/admin/shared/_header.html.erb
```

```
<header>
  <span class="logo-mark">BAUKIS</span>
  <%= content_tag(:span, flash.notice, class: 'notice') if flash.notice %>
```

```
<%= content_tag(:span, flash.alert, class: 'alert') if flash.alert %>
<%=
  if current_administrator
    link_to 'ログアウト', :admin_session, method: :delete
  else
    link_to 'ログイン', :admin_login
  end
%>
</header>
```

スタイルシート flash.css.scss を staff ディレクトリから admin ディレクトリにコピーします:

```
$ cp app/assets/stylesheets/staff/flash.css.scss app/assets/stylesheets/admin
```

admin ディレクトリの色変数定義ファイルを変更します:

```
LIST app/assets/stylesheets/admin/_colors.css.scss
```

(1-11 行省略)

```
/* 赤系 */
$red: #cc0000;
```

```
/* 緑系 */
$green: #00cc00;
```