Devise + CanCan實作

複習一下

新增專案 (I)

- 建立MyWeb專案，請在cmd中執行：
  - rails new MyWeb
- 在Gemfile中加入devise, cancan：

```
gem 'devise'
gem 'cancan'
```

- 執行bundle install

新增專案 (II)

- 建立welcome#index
  - rails generate controller Welcome index
  - 在app/views/welcome/index.html.erb中加上你想要的首頁文字。
- 設定為首頁，請在config/routes.rb中加入
  - root :to => 'welcome#index'
- 將public/index.html刪除

設定devise

- 在cmd中執行：
  - rails generate devise:install
- 在config/development.rb中加入:
  - config.action_mailer.default_url_options =
    { :host => 'localhost:3000' }

新增User model

- 在cmd中執行：
  - rails generate devise User
  - rake db:migrate
- 請各位建立幾個使用者
  - /users/sign_up # 註冊
  - /users/sign_in #登入
  - /users/sign_out #登出

使用者列表

- 我們要建立使用者列表的頁面。
- 請在cmd中執行：
  - rails g controller Users index

Users#index (controller)

```
def index
  @users = User.all
end
```

Users#index (view)

```
<h1>使用者列表</h1>

<table cellspacing="0">
    <tr>
        <th>Email</th>
        <th></th>
    </tr>
    <% @users.each do |user| -%>
        <tr>
            <td><%= user.email %></td>
            <td></td>
        </tr>
    <% end -%>
</table>
```

Scaffold

- 為了測試cancan所以新增一個company，來進行操作。
- 請在cmd中執行：
  - rails generate scaffold Company name:string tel:string address:string
  - rake db:migrate

CanCan

CanCan

- CanCan是一個處理授權的gem。
- CanCan可以限制某些使用者對某些resources進行操作。
- 所有的權限都存放在Ability class。

Define Ability

- 請在cmd中執行：
  - rails generate cancan:ability

新增Role

- 請在cmd中執行：
  - rails generate scaffold Role name:string
  - rake db:migrate

User與Role (I)

- User與Role是多對多的關係，需建立UserRole model
- 請在cmd中執行：
  - rails generate model UserRole user:references role:references
  - rake db:migrate

User與Role (II)

- 在app/models/user.rb中加入：

```
has_many :user_roles
has_many :roles, :through => :user_roles
```

- 在app/models/role.rb中加入：

```
has_many :user_roles
has_many :roles, :through => :user_roles
```

User與Role (III)

- 我們需要一個方法來檢查某位user是否屬於某個role。
- 請在app/models/user.rb中加入：

```
def role?(role)
  return !!self.roles.find_by_name(role.to_s)
end
```

User與Role (IV)

- 在console中測試看看:
  - user = User.first
  - user.role?('admin')
  - role = Role.create(:name => 'admin')
  - user.roles << role
  - user.role?('admin')

需求

- admin角色可以對全站做任何事。
- engineer角色可以對company做任何事。
- op角色只可以瀏覽company。
- 請在roles中加入這三種角色

編輯Role (I)

- 在config/routes.rb中加入：

```
resources :users do
    member do
      get :edit_role
      post :update_role
    end
  end
```

編輯Role (I)

- 請在app/controllers/users_controller.rb中加入：

```
def edit_role
  @user = User.find(params[:id])
end
```

編輯Role (I)

- 請在app/controllers/users_controller.rb中加入：

```
def update_role
  @user = User.find(params[:id])
  if @user.update_attributes(params[:user])
    flash[:notice] = "Edit role successful"
    redirect_to "/users"
  else
    flash[:error] = "Edit role fail"
    render :action => "edit_role"
  end
end
```

編輯Role (I)

- 請新增檔案app/views/users/edit_role.html.erb。

```
<h1>修改<%= @user.email %>的角色</h1>
<%= form_tag update_role_user_path(@user) do %>
    <table cellspacing="0">
        <tr>
            <th></th>
            <th>角色</th>
        </tr>
```

編輯Role (I)

- 接上頁

```
<% Role.all.each do |role| %>
<tr>
    <td>
        <%= check_box_tag "user[role_ids][]",
                          role.id,
                          @user.roles.include?(role) %>
        </td>
        <td><%= role.name %></td>
    </tr>
    <% end -%>
    </table>
    <%= submit_tag "確定" %>
<% end -%>
```

設定ability (I)

- ability中利用can方法來判斷對某一個resources有什麼權限
  - can :manage, :all #可以讀寫所有resources
  - can :read, :all #可以讀取所有resources
  - can :manage, Company #可以讀寫Company
  - can :read, Company #只可以讀取Company

設定ability (II)

- 請在app/models/ability.rb中加入：

```
def initialize(user)
  user ||= User.new
  case
  when user.role?("admin")
    can :manage, :all
  when user.role?("engineer")
    can :manage, Company
  when user.role?("op")
    can :read, Company
  end
end
```

設定Company

- 請在app/controllers/ companies_controller.rb中的第二行加入：
- load_and_authorize_resource