**\*\*세션 사용 로그인&로그아웃&helper사용방법**

[**https://www.flearning.net/lectures/10/clips/51**](https://www.flearning.net/lectures/10/clips/51)

rails generate controller Sessions new

**세션(Session)**

HTTP 는 상태 비저장 프로토콜로, 단어 그대로 상태가 존재하지 않아 이전에 사용했던 정보들을 전혀 사용할 수 없습니다. 말하자면 HTTP 는 요청이 끝나면 모든 것을 초기화하고 다시 시작하는 프로토콜입니다. 이러한 본질적인 특성 때문에 브라우저의 페이지에서 다른 페이지로 이동할 때 로그인한 사용자의 ID 를 유지할 수단이 HTTP 에는 존재하지 않습니다. 이와 같은 사용자 로그인을 세션(Session)이라는 반 영구적인 연결을 컴퓨터와 해당 웹 사이트의 서버에 별도로 설정할 수 있으며, HTTP 프로토콜과는 별도로 연결을 확보할 수 있습니다.

이러한 세션을 레일즈 내에서 구현하는 가장 쉬운 방법은 쿠키(Cookie)를 이용하는 방법입니다. 쿠키는 서버가 아닌, 사용자의 브라우저에 저장되는 텍스트 정보로 다른 페이지로 이동했을 때도 사라지지 않기 때문에, 여기에 로그인한 사용자와 관련된 정보를 저장하여 이를 통해 현재 로그인한 사용자를 데이터베이스에서 조회, 가져올 수 있습니다. 세션은 브라우저를 종료하면 자동으로 사라집니다.

이 세션을 RESTful 한 리소스로 구성을 할 것인데, 그렇게 하면 통일적으로 이해할 수 있어 편합니다. 로그인 페이지에서 new 로 새로운 세션에 관한 데이터를 작성하고, create 에서는 세션을 실제로 만들고 로그아웃 시에 destroy 액션에서 세션을 파기하는 식입니다. Users 자원과는 다른 점은 Users 자원은 User 모델을 통해서 데이터베이스에 액세스하는 방면에 세션은 이러한 자원 대신에 쿠키 저장소로서 이용합니다. 앞서 말했듯이 로그인 방식은 쿠키를 이용한 매커니즘으로 대부분 구축되어 있습니다.

**세션 컨트롤러**

로그인, 로그아웃 요소를 Sessions 컨트롤러의 특정 액션과 매칭시킬 수 있습니다. 로그인 양식은 new 액션으로 처리하고, create 액션에 POST 요청을 통해 실제로 로그인을 합니다. destroy 액션에 DELETE 요청을 하는 것으로 로그아웃을 할 수 있습니다.

$ rails generate controller Sessions new

위와 같은 명령어를 입력하여 세션 컨트롤러를 만들면, new 액션과 new 뷰도 함께 생성됩니다. create, destroy 의 경우에는 대응되는 뷰가 불필요하므로 이렇게 new 만 명령어로 생성해주도록 합니다.

config/routes.rb 수정

get 'login' => 'sessions#new' post 'login' => 'sessions#create' delete 'logout' => 'sessions#destroy'

**라우팅은 아래의 표를 코드로 추가한 것입니다.**

| **HTTP 요청** | **URL** | **명명된 경로** | **액션** | **용도** |
| --- | --- | --- | --- | --- |
| GET | /login | login\_path | new | 새로운 세션의 페이지 (로그인) |
| POST | /login | login\_path | create | 새로운 세션 생성 (로그인) |
| DELETE | /logout | logout\_path | destroy | 세션 삭제 (로그 아웃) |

**작성할 코드**

app/views/sessions/new.html.erb 작성

<%= form\_tag '/login' do %> 이메일 <input type="email" name="email"><br> 패스워드 <input type="password" name="password"><br> <input type="submit" value="로그인"> <% end %>

app/controllers/application\_controller.rb 수정

class ApplicationController < ActionController::Base protect\_from\_forgery with: :exception include SessionsHelper end

app/helpers/sessions\_helper.rb 수정

module SessionsHelper *# 전달된 사용자로 로그인* def log\_in(user) session[:user\_id] = user.id end *# 현재 로그인 한 사용자를 반환(로그인 한 경우에만)* def current\_user @current\_user ||= User.find\_by(id: session[:user\_id]) end *# 로그인 되어있다면 true 를 그렇지 않다면 false 를 반환* def user\_signed\_in? !current\_user.nil? end end

app/controllers/sessions\_controller.rb 수정

class SessionsController < ApplicationController def new end def create user = User.find\_by(email: params[:session][:email].downcase) if user && user.authenticate(params[:session][:password]) log\_in user redirect\_to user else flash.now[:danger] = 'Invalid email/password combination' render 'new' end end def destroy log\_out redirect\_to root\_url end end