

Web System Development with Ruby on Rails



Day 12(13/Dec/2012)
Session Control

What is Session?

- Session is a dialogue.
- When user signs in, the dialogue starts. Session is a series of the user request, to perform a set of server access.
 - Ex. User login, start purchase things, complete payment form, and logout.

How to maintain the session.

The frequently used methods are;

- (1) Cookies,
- (2) Hidden form parameter,
- (3) URL relocation

Controllers' Actions are invoked every time independently, which means, there is no consistency to keep the information from the previous access, without session management.

The Session Program Logic

In Ruby on Rails, when a server starts a session, a unique ID is generated and stored in the server.

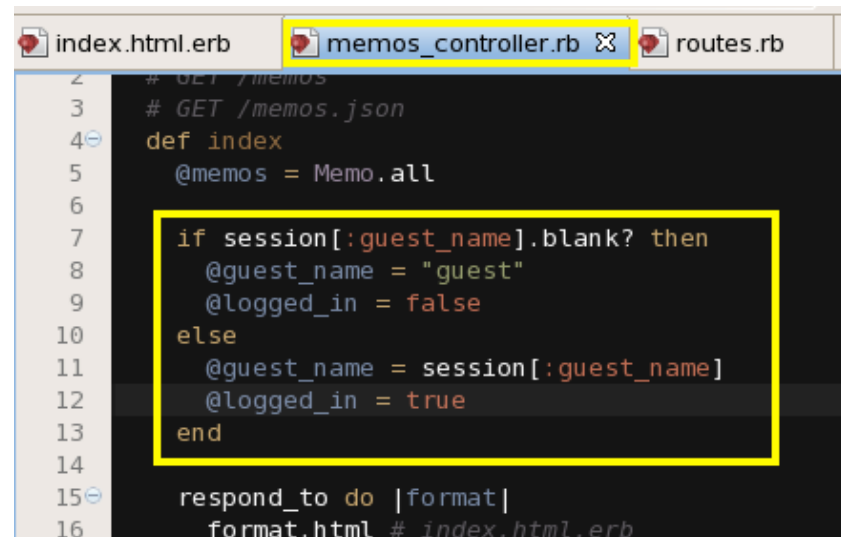
The same ID will be returned to the client, and the ID will be transmitted always to the server in all the requests from that client.

Server can identify the client (user) with the ID to maintain the session.

First session trial (controller)

Add the following lines to the index method of memos_controllers.rb

```
if session[:guest_name].blank? then
  @guest_name = "guest"
  @logged_in = false
else
  @guest_name = session[:guest_name]
  @logged_in = true
end
```



```
index.html.erb memos_controller.rb routes.rb
2 # GET /memos
3 # GET /memos.json
4 def index
5   @memos = Memo.all
6
7   if session[:guest_name].blank? then
8     @guest_name = "guest"
9     @logged_in = false
10  else
11    @guest_name = session[:guest_name]
12    @logged_in = true
13  end
14
15  respond_to do |format|
16    format.html # index.html.erb
```

First session trial (views)

Add the following lines to app/views/memos/index.html.erb

```
<div align="right">
Hello, <%= @guest_name %><br />
<% if @logged_in then %>
  <%= form_tag({:action=>"logout"}) do %>
    Logout:
    <%= submit_tag "Logoug" %>
  <% end%>
<% else %>
  <%= form_tag({:action=>"login"}) do %>
    Login:
    <%= text_field_tag("guest_name") %>
    <%= submit_tag "Login" %>
  <% end%>
<% end %>
</div>
```

Index.html.erb

```
index.html.erb x memos_controller.rb figure.rb routes.rb
1 <%= @page_title = 'app_title' %>
2 <h1><%= t :list_memo %></h1>
3
4 <div align="right">
5 Hello, <%= @guest_name %><br />
6 <%= if @logged_in then %>
7   <%= form_tag({:action=>"logout"}) do %>
8     Logout:
9     <%= submit_tag "Logoug" %>
10    <%= end%>
11  <%= else %>
12    <%= form_tag({:action=>"login"}) do %>
13      Login:
14      <%= text_field_tag("guest_name") %>
15      <%= submit_tag "Login" %>
16    <%= end%>
17  <%= end %>
18 </div>
19 <table border="1">
20   <tr>
21     <th><%= t :content %></th>
22     <th>Category</th>
```

First session trial (login actions)

Add the following lines to the
memos_controller.rb

```
def login
```

```
  session[:guest_name] = params[:guest_name]
```

```
  redirect_to :action=>"index"
```

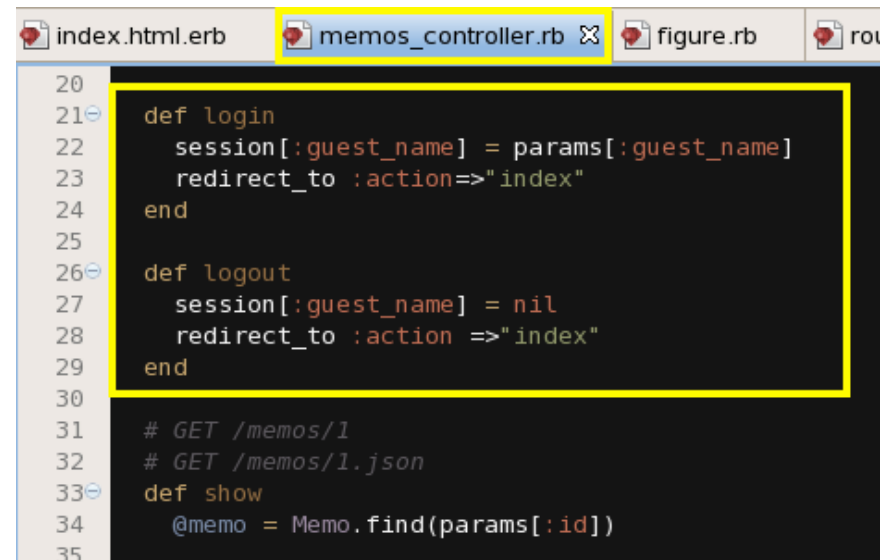
```
end
```

```
def logout
```

```
  session[:guest_name] = nil
```

```
  redirect_to :action =>"index"
```

```
end
```



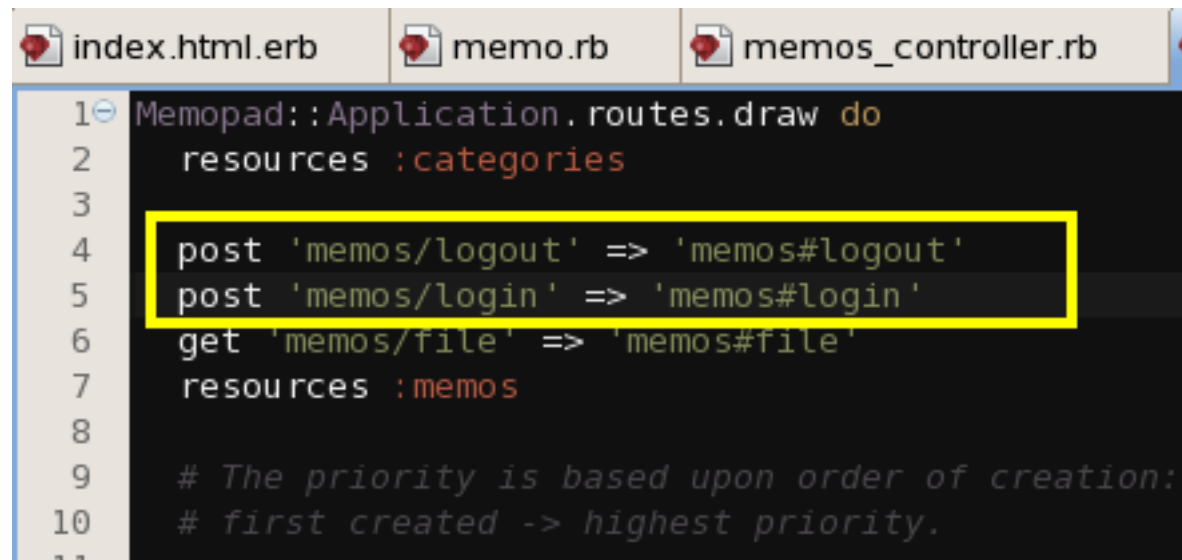
The screenshot shows a code editor with several tabs: 'index.html.erb', 'memos_controller.rb', 'figure.rb', and 'rou'. The 'memos_controller.rb' tab is active and highlighted with a yellow border. The code in this tab is as follows:

```
20
21 def login
22   session[:guest_name] = params[:guest_name]
23   redirect_to :action=>"index"
24 end
25
26 def logout
27   session[:guest_name] = nil
28   redirect_to :action =>"index"
29 end
30
31 # GET /memos/1
32 # GET /memos/1.json
33 def show
34   @memo = Memo.find(params[:id])
35
```


First session trial (routes)

Add the following two lines to routes.rb

```
post 'memos/logout' => 'memos#logout'  
post 'memos/login' => 'memos#login'
```



```
index.html.erb | memo.rb | memos_controller.rb  
1 Memopad::Application.routes.draw do  
2   resources :categories  
3  
4   post 'memos/logout' => 'memos#logout'  
5   post 'memos/login' => 'memos#login'  
6   get 'memos/file' => 'memos#file'  
7   resources :memos  
8  
9   # The priority is based upon order of creation:  
10  # first created -> highest priority.  
11
```

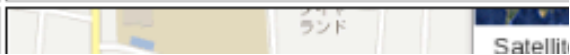
First Session Trial (Screen shots)

The screenshot shows a Mozilla Firefox browser window titled "My Memopad - Mozilla Firefox". The address bar displays "http://127.0.0.1:3000/memos". The browser's menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The address bar contains navigation icons (back, forward, refresh, stop, home) and a search engine dropdown set to "Google". Below the address bar, there are bookmarks for "よく見るページ", "CentOS", and "Support".

The main content area features a header with a cat image on the left and a cyan background on the right. Below the header, there are navigation links: "Listing memos", "New Memo", "Ruby Official Site", and "My Theme Page(Preparing)".

The main heading is "Listing memos". To the right of the heading, there is a login section with the text "Hello, guest", a "Login:" label, an input field, and a "Login" button. This login section is highlighted with a yellow box.

Below the heading, there is a table with the following structure:

Content	Category	Figures
Yesterday, it was rainy because of the typhoon. Strong wind.	Idea	
		

At the bottom right of the page, there is another login section with the text "Hello, hoge", a "Logout:" label, a "Logoug" button, and a table structure below it.

Application of session parameter

When you write the application of shopping site, such parameters as 'purchase list' can be stored in the session parameter.

The other way is to store the 'purchase list' information in the database, and use the session parameter as the access key for the database.

Login user management

We can design the user table to look up the registered user information, and the password in the database table too.

However, parameters stored in the database can be visible to the administrators and the users in general.

How to keep the password invisible?

Devise

One of the simple answers is to use 'devise' gem.

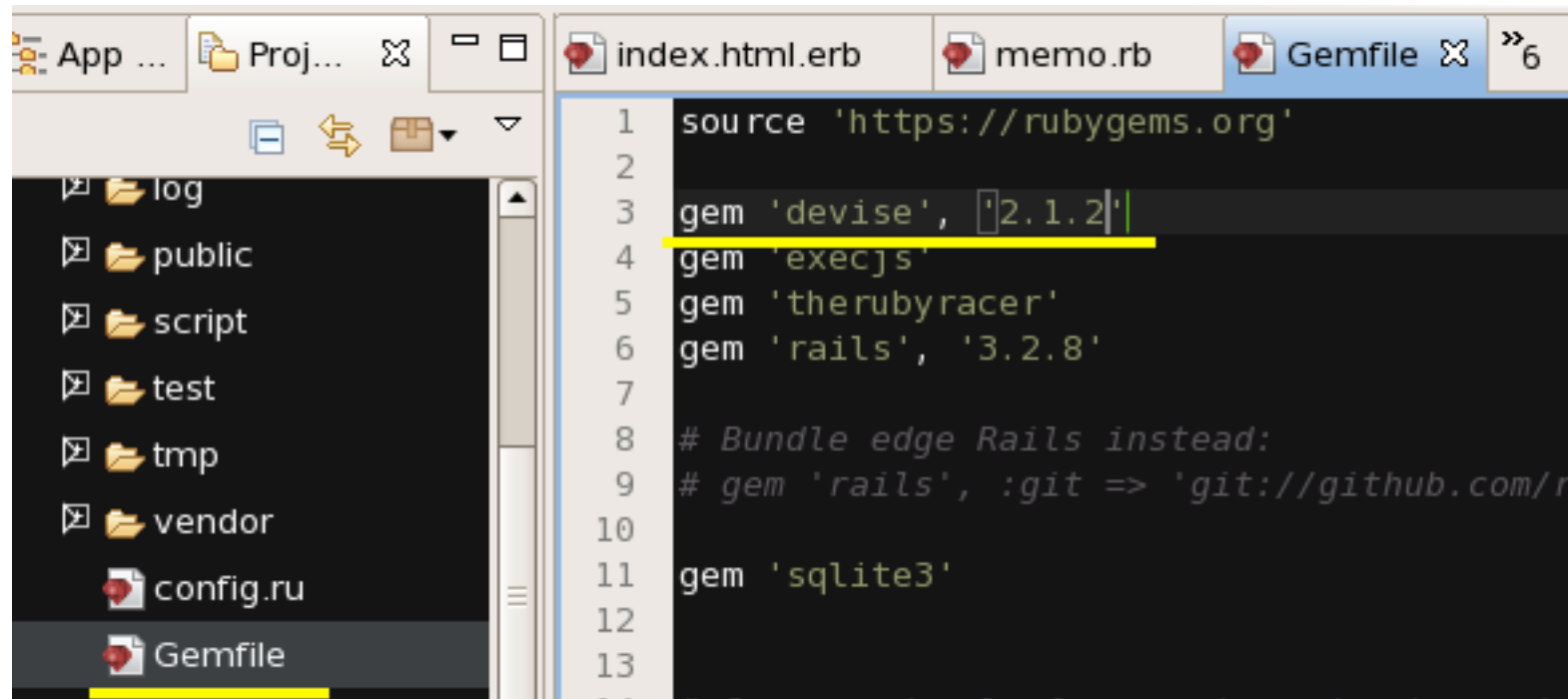
We use '**Devise**' to manage the login control.

Now we start using devise gem.

Fix Gemfile

- ▣ Add one line to the file 'project/Gemfile'

gem 'devise', '2.1.2'



The screenshot shows a code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with folders like 'log', 'public', 'script', 'test', 'tmp', and 'vendor', and files like 'config.ru' and 'Gemfile'. The 'Gemfile' is selected. The code editor shows the following content:

```
1 source 'https://rubygems.org'
2
3 gem 'devise', '2.1.2'
4 gem 'execjs'
5 gem 'therubyracer'
6 gem 'rails', '3.2.8'
7
8 # Bundle edge Rails instead:
9 # gem 'rails', :git => 'git://github.com/rails/rails.git'
10
11 gem 'sqlite3'
12
13
```

The line `gem 'devise', '2.1.2'` is highlighted in yellow.

Bundle install

Type

`bundle install`

In the project folder

```
Using mail (2.4.4)
Using actionmailer (3.2.8)
Using arel (3.0.2)
Using tzinfo (0.3.33)
Using activerecord (3.2.8)
Using activeresource (3.2.8)
Using bcrypt-ruby (3.0.1)
Using bundler (1.2.0)
Using coffee-script-source (1.3.3)
Using execjs (1.4.0)
Using coffee-script (2.2.0)
Using rack-ssl (1.3.2)
Using json (1.7.5)
Using rdoc (3.12)
Using thor (0.16.0)
Using railties (3.2.8)
Using coffee-rails (3.2.2)
Using orm_adapter (0.4.0)
Using warden (1.2.1)
Using devise (2.1.2)
Using jquery-rails (2.1.3)
```

```
Using libv8 (3.3.10.4)
Using rails (3.2.8)
Using sass (3.2.1)
Using sass-rails (3.2.5)
Using sqlite3 (1.3.6)
Using therubyracer (0.10.2)
Using uglifier (1.3.0)
```

```
Your bundle is complete! Use `bundle show [gemname]` to see where a bundled gem is installed.
```

```
[root@cisnote memopad]#
```

```
[root@cisnote memopad]# bundle install
Using rake (0.9.2.2)
Using i18n (0.6.1)
Using multi_json (1.3.6)
Using activesupport (3.2.8)
Using builder (3.0.3)
Using activemodel (3.2.8)
Using erubis (2.7.0)
Using journey (1.0.4)
Using rack (1.4.1)
Using rack-cache (1.2)
Using rack-test (0.6.2)
Using hike (1.2.1)
Using tilt (1.3.3)
Using sprockets (2.1.3)
Using actionpack (3.2.8)
Using mime-types (1.19)
Using polyglot (0.3.3)
Using treetop (1.4.10)
```

Confirm installation of the Gem

□ Type

`gem list devise`

Make sure that
devise (2.1.2)
replied.

```
[root@cisnote memopad]# gem list devise  
  
*** LOCAL GEMS ***  
  
devise (2.1.2)  
[root@cisnote memopad]#
```


Install Devise to the application

□ Type

`rails generate devise:install`

```
[root@cisnote memopad]# rails generate devise:install
  create  config/initializers/devise.rb
  create  config/locales/devise.en.yml
```

Some setup you must do manually if you haven't yet:

1. Ensure you have defined default url options in your environments files. Here is an example of default_url_options appropriate for a development environment in config/environments/development.rb:

```
config.action_mailer.default_url_options = { :host => 'localhost:3000' }
```

In production, :host should be set to the actual host of your application.

Read the message from the system carefully

Some setup you must do manually if you haven't yet:

1. Ensure you have defined default url options in your environments files. Here is an example of default_url_options appropriate for a development environment in config/environments/development.rb:

```
config.action_mailer.default_url_options = { :host => 'localhost:3000' }
```

In production, :host should be set to the actual host of your application.

2. Ensure you have defined root_url to *something* in your config/routes.rb. For example:

```
root :to => "home#index"
```

3. Ensure you have flash messages in app/views/layouts/application.html.erb. For example:

```
<p class="notice"><%= notice %></p>
<p class="alert"><%= alert %></p>
```

4. If you are deploying Rails 3.1 on Heroku, you may want to set:

```
config.assets.initialize_on_precompile = false
```

On config/application.rb forcing your application to not access the DB or load models when precompiling your assets.

3 steps to use devise

1. Ensure you have defined default url options in your your environments files.
2. Ensure you have defined `root_url` to `*something*` in your `config/routes.rb`.
3. Ensure you have flash messages in `app/views/layouts/application.html.erb`.
4. If you are deploying Rails 3.1 on Heroku, you may want to set: ... (it does not meet this condition.)

Set default_url_options (Step 1)

- ❑ Add the following statement to /config/environments/development.rb
- ❑ read the set-up message carefully

```
config.action_mailer.default_url_options = { :host => 'localhost:3000' }
```

```
7   config.cache_classes = false
8
9   # Log error messages when you accidentally call methods on nil.
10  config.whiny_nils = true
11
12  # Show full error reports and disable caching
13  config.consider_all_requests_local       = true
14  config.action_controller.perform_caching = false
15
16  # Don't care if the mailer can't send
17  config.action_mailer.raise_delivery_errors = false
18
19  config.action_mailer.default_url_options = { :host => 'localhost:3000' }
20
21  # Print deprecation notices to the Rails logger
22  config.active_support.deprecation = :log
23
24  # Only use best-standards-support built into browsers
25  config.action_dispatch.best_standards_support = :builtin
```

Set the Login Default Screen(Step 2)

- We set the Login Default Screen to the welcome screen, which we had created in the last page. First, we modify

(project name)/config/routes.rb

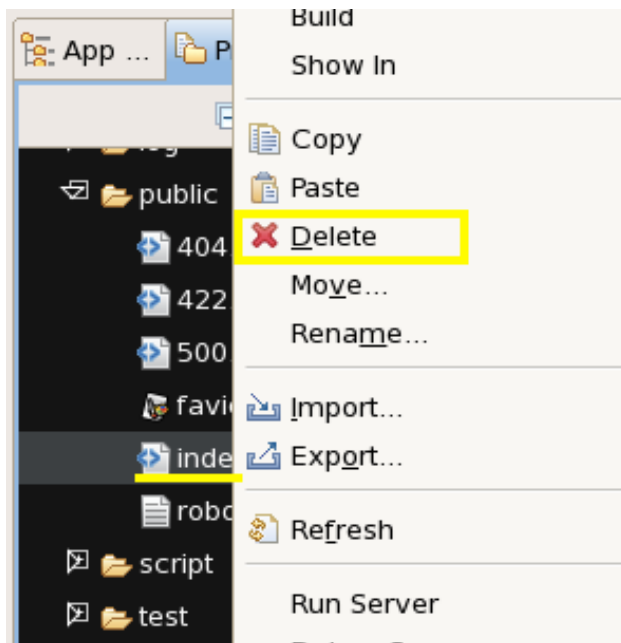
- Uncomment the line around line #58, to activate, and modify to lead to 'memos#index'

`root :to => 'memos#index'`

```
52 # # (app/controllers/admin/products_controller.rb)
53 #   resources :products
54 # end
55
56 # You can have the root of your site routed with "root"
57 # just remember to delete public/index.html.
58 root :to => 'memos#index'
59
60 # See how all your routes lay out with "rake routes"
61
62 # This is a legacy wild controller route that's not recom
```

Remove public/index.html

Read the comment in routes.rb carefully.
We have to make sure to remove the file
public/index.html



Add two lines for login result message display (Step 3)

- Modify

(project)/app/views/layouts/application.html.erb

- Add the following two lines before `<%= yield %>`

`<p class="notice"><%= notice %></p>`

`<p class="alert"><%= alert %></p>`

```
9 <body>
10 <p style="color: green"><%= flash[:notice] %></p>
11 <div id="container">
12 <div id="header">
13 <%= image_tag('cat.jpeg', :size=>"133x94") %>
14 <%= render :partial => 'shared/menu_bar' %>
15 </div>
16 <div id="left">
17 <p class="notice"><%= notice %></p>
18 <p class="alert"><%= alert %></p>
19 <%= yield %>
20 </div>
21 <div id="right">
22 <%= render :partial => 'shared/right_bar' %>
23 </div>
```

View for devise

- ▣ Here we generate views for devise. Type `rails generate devise:views`

```
[root@cisnote memopad]# rails generate devise:views
  invoke  Devise::Generators::SharedViewsGenerator
  create  app/views/devise/shared
  create  app/views/devise/shared/_links.erb
  form_for
  create  app/views/devise/confirmations
  create  app/views/devise/confirmations/new.html.erb
  create  app/views/devise/passwords
  create  app/views/devise/passwords/edit.html.erb
  create  app/views/devise/passwords/new.html.erb
  create  app/views/devise/registrations
  create  app/views/devise/registrations/edit.html.erb
  create  app/views/devise/registrations/new.html.erb
  create  app/views/devise/sessions
  create  app/views/devise/sessions/new.html.erb
  create  app/views/devise/unlocks
  create  app/views/devise/unlocks/new.html.erb
  erb
  create  app/views/devise/mailer
  create  app/views/devise/mailer/confirmation_instructions.html.erb
  create  app/views/devise/mailer/reset_password_instructions.html.erb
  create  app/views/devise/mailer/unlock_instructions.html.erb
[root@cisnote memopad]#
```


User Model for Authentication

- Generate Class User for devise. Type, rails generate devise user

```
[root@cisnote memopad]# rails generate devise user
      create  active_record
      create  db/migrate/20121211075114_devise_create_users.rb
      create  app/models/user.rb
      create  test_unit
      create  test/unit/user_test.rb
      create  test/fixtures/users.yml
      insert  app/models/user.rb
      route  devise_for :users
[root@cisnote memopad]# █
```

migration

- Now we migrate, using all 'automatic' result as default. Type `rake db:migrate`

```
[root@cisnote memopad]# rake db:migrate
== DeviseCreateUsers: migrating =====
-- create_table(:users)
   -> 0.0972s
-- add_index(:users, :email, {:unique=>true})
   -> 0.0012s
-- add_index(:users, :reset_password_token, {:unique=>true})
   -> 0.0006s
== DeviseCreateUsers: migrated (0.0996s) =====

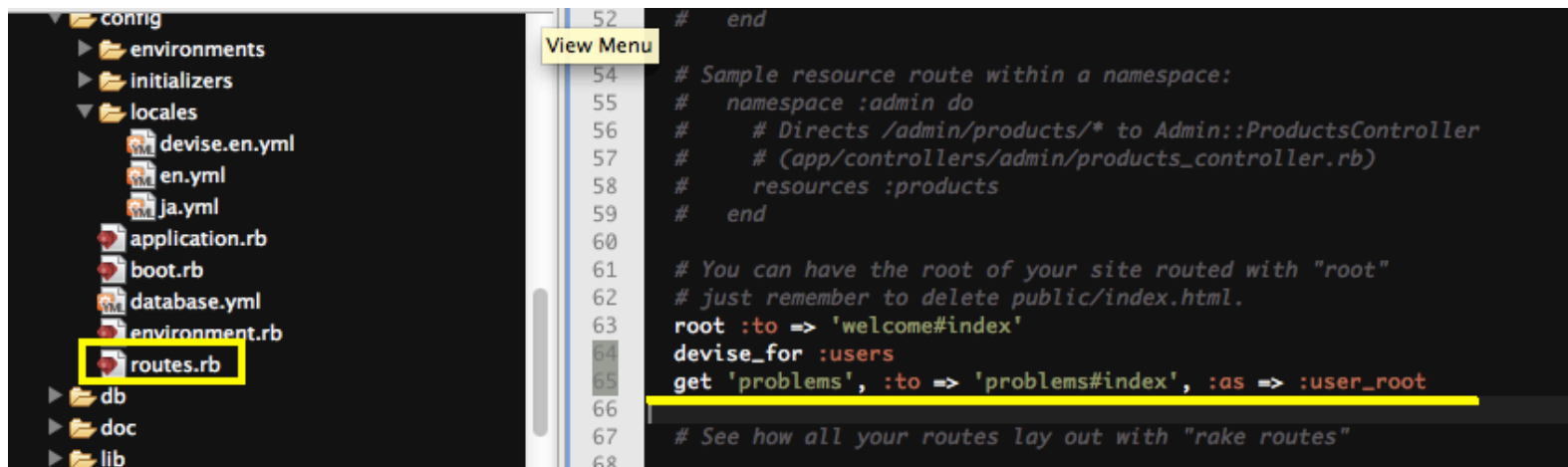
[root@cisnote memopad]#
```

Authentication Path/Redirection

Once again, modify `config/routes.rb`, add one line under the following line.

```
devise_for :users  
get 'memos', :to => 'memos#index', :as => :user_root
```

Reference: <http://railscasts.com/episodes/209-introducing-devise>



```
52 # end  
54 # Sample resource route within a namespace:  
55 # namespace :admin do  
56 # # Directs /admin/products/* to Admin::ProductsController  
57 # # (app/controllers/admin/products_controller.rb)  
58 # resources :products  
59 # end  
60  
61 # You can have the root of your site routed with "root"  
62 # just remember to delete public/index.html.  
63 root :to => 'welcome#index'  
64 devise_for :users  
65 get 'problems', :to => 'problems#index', :as => :user_root  
66  
67 # See how all your routes lay out with "rake routes"  
68
```

config/routes.rb

`devise_for :users`

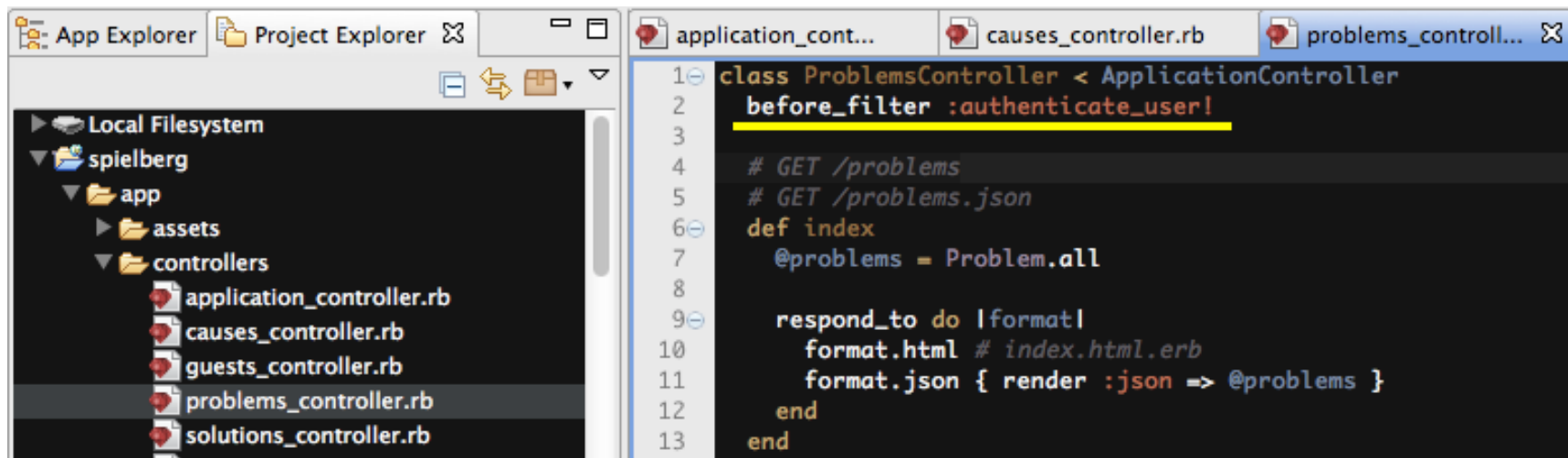
- This line is automatically added, and this modification is to register paths to the login form and user registration form.

`get 'memos', :to => 'memos#index', :as => :user_root`

- This line is the redirection path after the authentication.

For other controllers

- ▣ Add authentication request as a `before_filter`, in `memos_controller.rb`
`before_filter :authenticate_user!`



```
1 class ProblemsController < ApplicationController
2   before_filter :authenticate_user!
3
4   # GET /problems
5   # GET /problems.json
6   def index
7     @problems = Problem.all
8
9     respond_to do |format|
10      format.html # index.html.erb
11      format.json { render :json => @problems }
12    end
13  end
```

Now the screen is



[Listing memos](#) | [New Memo](#) | [Ruby Official Site](#) | [My Theme Page\(Pre](#)

You need to sign in or sign up before continuing.

Sign in

Email

Password

Remember me

[Sign up](#)

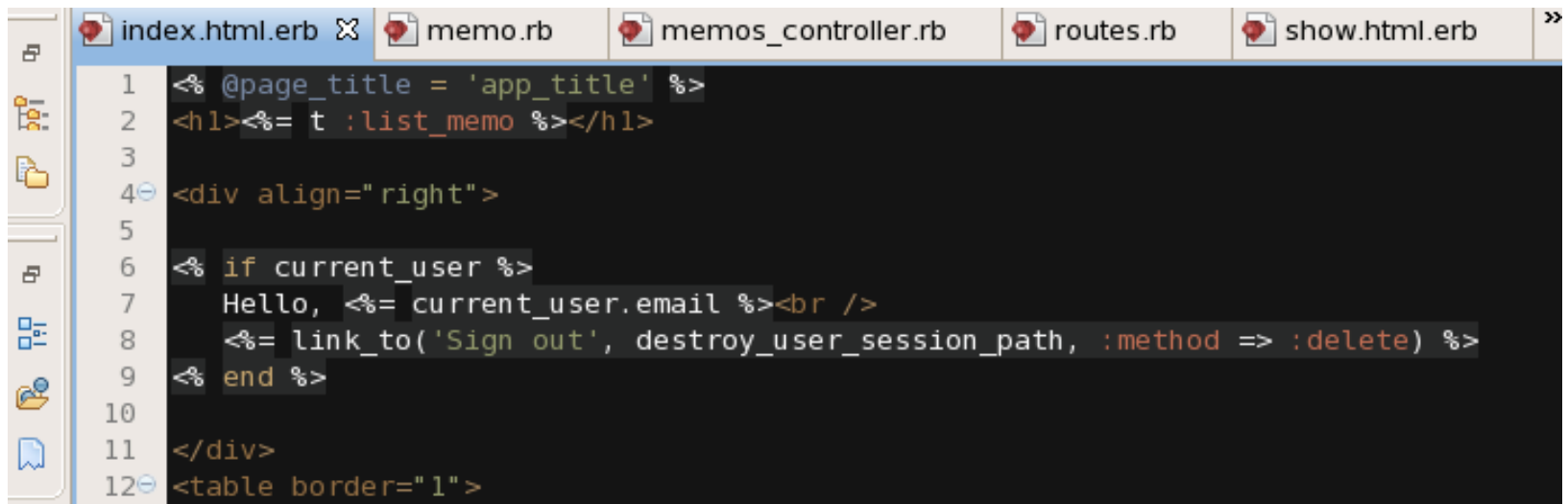
Cancel our first session trial

To replace our user session management with devise gem, cancel the modification we have done, described in the page 5 to 9 in the slides.

Sign out procedure

To destroy the signed-in session, add the following lines in index.html.erb

```
<% if current_user %>  
  Hello, <%= current_user.email %><br />  
  <%= link_to('Sign out', destroy_user_session_path, :method  
    => :delete) %>  
<% end %>
```



The screenshot shows a code editor with several tabs open: index.html.erb, memo.rb, memos_controller.rb, routes.rb, and show.html.erb. The active tab is index.html.erb, which contains the following code:

```
1 <%= @page_title = 'app_title' %>  
2 <h1><%= t :list_memo %></h1>  
3  
4 <div align="right">  
5  
6 <%= if current_user %>  
7   Hello, <%= current_user.email %><br />  
8   <%= link_to('Sign out', destroy_user_session_path, :method => :delete) %>  
9 <%= end %>  
10  
11 </div>  
12 <table border="1">
```


Now the first screen is

Signed in successfully.



[Listing memos](#) | [New Memo](#) | [Ruby Official Site](#) | [My Theme Page\(Preparing\)](#)

Signed in successfully.

Listing memos

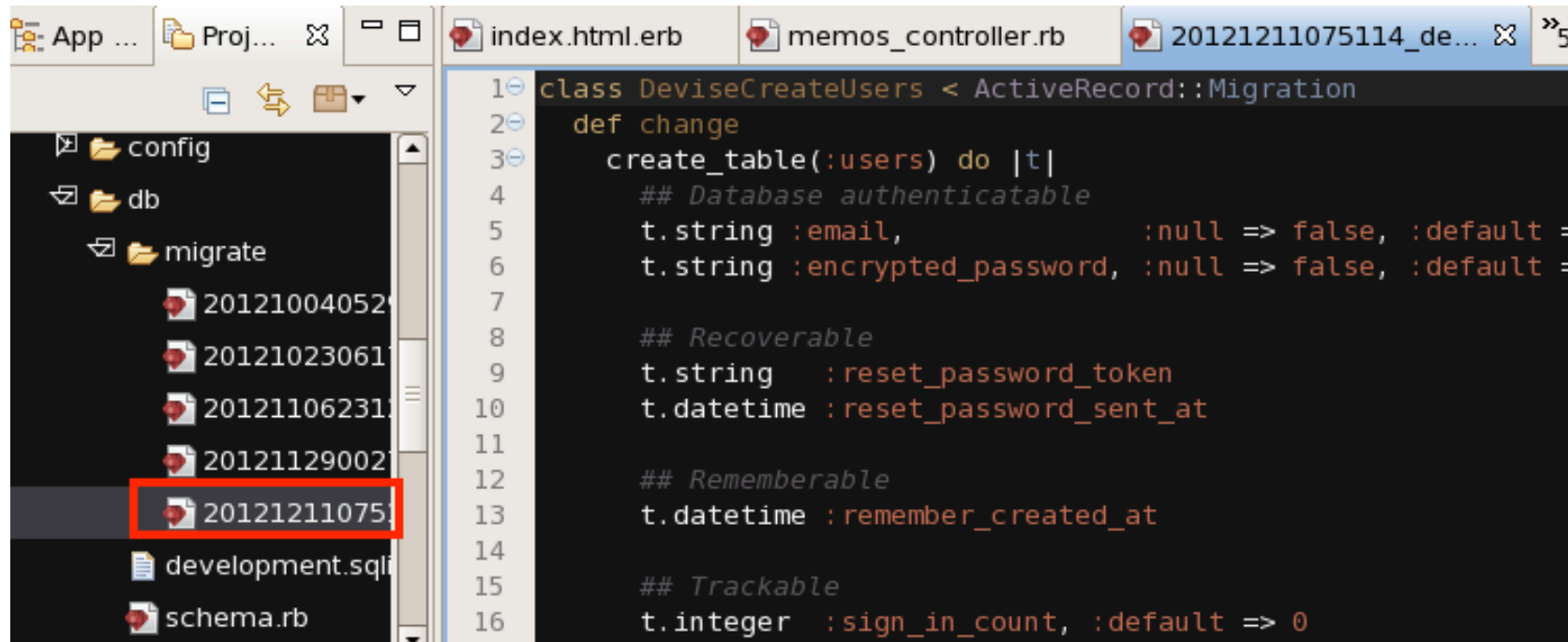
Hello, hoge@hosei.ac.jp
[Sign out](#)

Content	Category	Figures			
Yesterday, it was rainy because of the typhoon.	Idea		Show	Edit	Destroy

Where is 'User' information?

The user information can be looked up with `current_user` variable.

We can see the default user information in the migration file.



```
1 class DeviseCreateUsers < ActiveRecord::Migration
2   def change
3     create_table(:users) do |t|
4       ## Database authenticatable
5       t.string :email, :null => false, :default => ''
6       t.string :encrypted_password, :null => false, :default => ''
7
8       ## Recoverable
9       t.string :reset_password_token
10      t.datetime :reset_password_sent_at
11
12      ## Rememberable
13      t.datetime :remember_created_at
14
15      ## Trackable
16      t.integer :sign_in_count, :default => 0
```

How to keep session information?

When we use devise, this `current_user` can be the 'key' of the session.

Once the user signs out, the session is destroyed.

Routes for users

Type `rake routes` to see the routing table.
Users' paths are added.

```
[root@cisnote memopad]# rake routes
      new_user_session GET    /users/sign_in(.:format)    devise/sessions#new
      user_session POST   /users/sign_in(.:format)    devise/sessions#create
destroy_user_session DELETE /users/sign_out(.:format)   devise/sessions#destroy
      user_password POST   /users/password(.:format)   devise/passwords#create
      new_user_password GET    /users/password/new(.:format) devise/passwords#new
      edit_user_password GET    /users/password/edit(.:format) devise/passwords#edit
                                PUT    /users/password(.:format)   devise/passwords#update
cancel_user_registration GET    /users/cancel(.:format)     devise/registrations#cancel
cancel
      user_registration POST   /users(.:format)            devise/registrations#create
create
      new_user_registration GET    /users/sign_up(.:format)    devise/registrations#new
      edit_user_registration GET    /users/edit(.:format)       devise/registrations#edit
edit
                                PUT    /users(.:format)            devise/registrations#update
update
                                DELETE /users(.:format)            devise/registrations#destroy
destroy
      user_root GET    /memos(.:format)            memos#index
      categories GET    /categories(.:format)       categories#index
                                POST   /categories(.:format)       categories#create
```

Today's Theme

Read the routing table, and add the user maintenance applications.

Hint: When a user is not signed in, the following lines are useful.

```
<p><%= link_to 'Login', [ :new, :user_session ] %></p>
```

```
<p><%= link_to 'User Registration', [ :new, :user_registration ] %></p>
```

```
<p><%= link_to 'Reissue Password', [ :new, :user_password ] %></p>
```

Also, try the link to [edit_user_registration_path](#) and such, listed in the routing table.

Do not forget to add `_path` at the end of the path name.

The last report theme.

This is the last report theme;

- (1) Add a owner column to the memos table, and then allow the memos destruction only to the owner.
- (2) Add a users face photo(or icon) table linked to the users table, then, show the memos' owners face (or icon) to the memos listing screen.
- (3) Add any other function you think you need for the system.

The deadline of this report is the first lecture day in 2013, after the Christmas Holidays.

The presentation

We will have the presentation of your memos WEB page, on 10/January/2013.

Add your original functions to this memos WEB page, and show the system in the class.



Absence report for today

Report the screen shots of the login screen of devise.