

0403

3주차 : CRUD#1

LIKE LION

EWHALION

5th 엄서영

3주차 : CRUD#1

CRUD

Create / Read / Update / Delete

간단한 게시판 기능



3주차 : CRUD#1

CRUD

Create / Read / Update / Delete

간단한 게시판 기능



CRUD#1[CR]

지난주...

다음 세션 예고

VC는 만들었고...
그럼 이제 M만 남았네!

EWHA LIKELION

EWHA LIKELION

CRUD#1[CR]

지난주...

다음 세션 예고

VC는 만들었고...
그럼 이제 M만 남았네!



MVC...?

EWHA LIKELION

EWHA LIKELION

CRUD#1[CR]

MVC & Route

[간단한 실습]

나와 내 옆자리 사람을 입력받아
출력해주기!

오늘의 내 짝궁!

나는 :

오늘 내 옆에는 :

Home#show

나는 : 아기사자

내 짝궁: 라이온킹

EWHA LIKELION

View파일

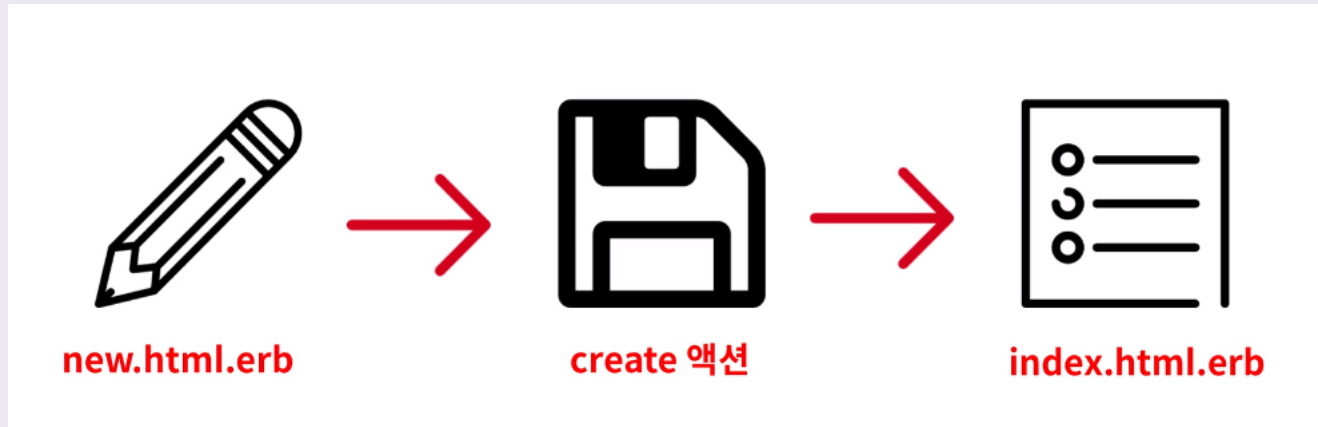
라우팅

액션 정의

VC!!!

EWHA LIKELION

CR! Create & Read



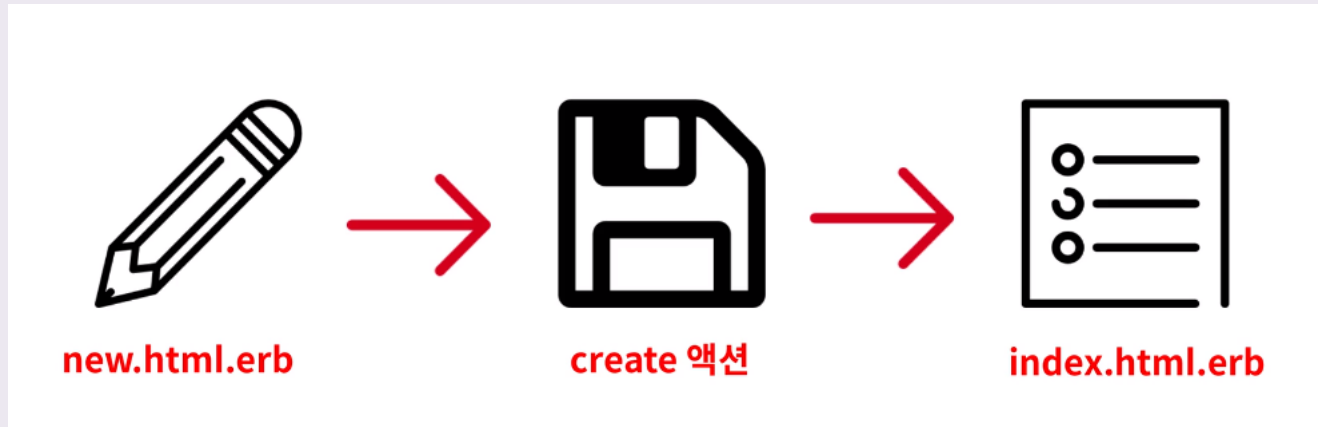
글 작성 폼 만들기

입력 받은 정보 저장

게시물 목록으로 보여주기

CRUD#1[CR]

CR! Create & Read



글 작성 폼 만들기

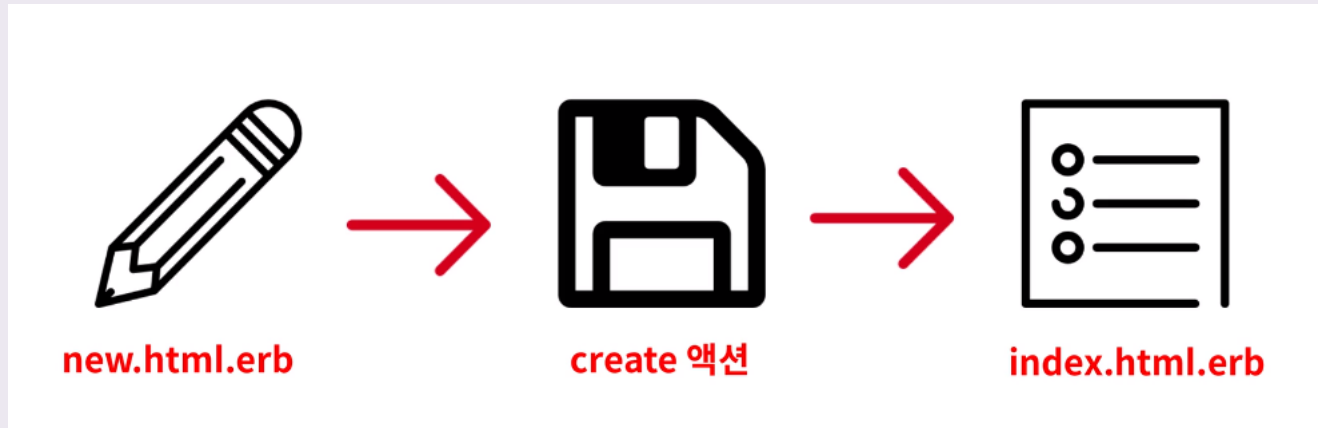
입력 받은 정보 **저장**

MVC!!!

게시물 목록으로 보여주기

CRUD#1[CR]

CR! Create & Read

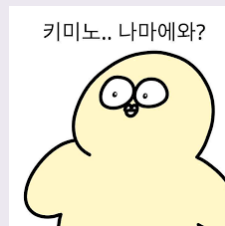


글 작성 폼 만들기

입력 받은 정보 **저장**

Model!!!

게시물 목록으로 보여주기





변수 : 일시적 정보저장

<p>오늘의 내작공!</p> <p>나는 : <input type="text" value="아기사자"/></p> <p>오늘 내 옆에는 : <input type="text" value="라이온킹"/></p> <p><input type="button" value="확인"/></p>	<p>Home#show</p> <p>나는 : 아기사자</p> <p>내작공: 라이온킹</p>
--	--



DB : 영구적 정보저장

여기는 게시물 목록

[새로운 글 쓰기](#)

제목 : 안녕
내용 : 와우웅와
[\[삭제\]](#)

제목 : 끼요우우웅
내용 : 후헤헤헤헹
[\[삭제\]](#)

CRUD#1[CR]

Workspace name

crud2018

Description


Make a short description of your workspace


Hosted workspace

Clone workspace

Remote SSH workspace

Salesforce


☐  **Private**
This is a workspace for your eyes only


☐  **Public**
This will create a workspace for everybody to see


Clone from Git or Mercurial URL (optional)


e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git


Choose a template



HTML5



Node.js



PHP, Apache & ...



Python



Django



Ruby


C++


Wordpress


Rails Tutorial


Blank


Harvard's CS50

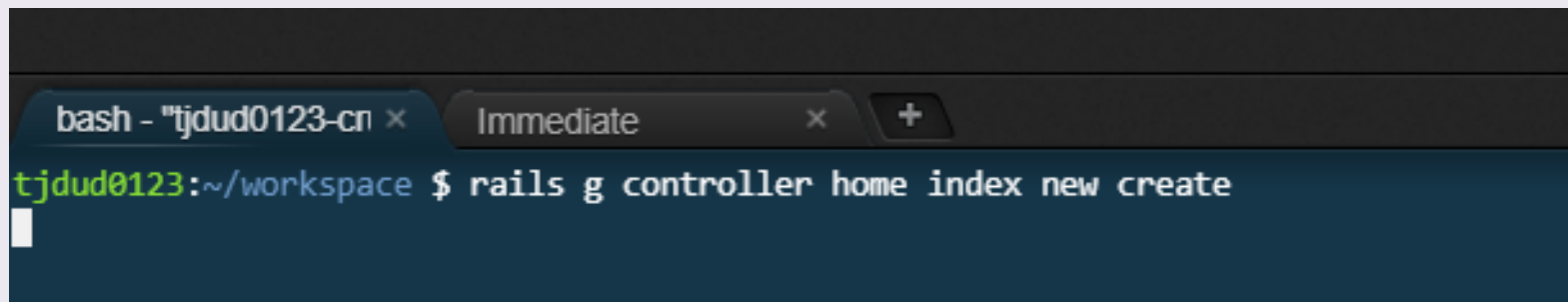
Create workspace

EWHA LIKELION

2018.4.3

rails g controller home index new create

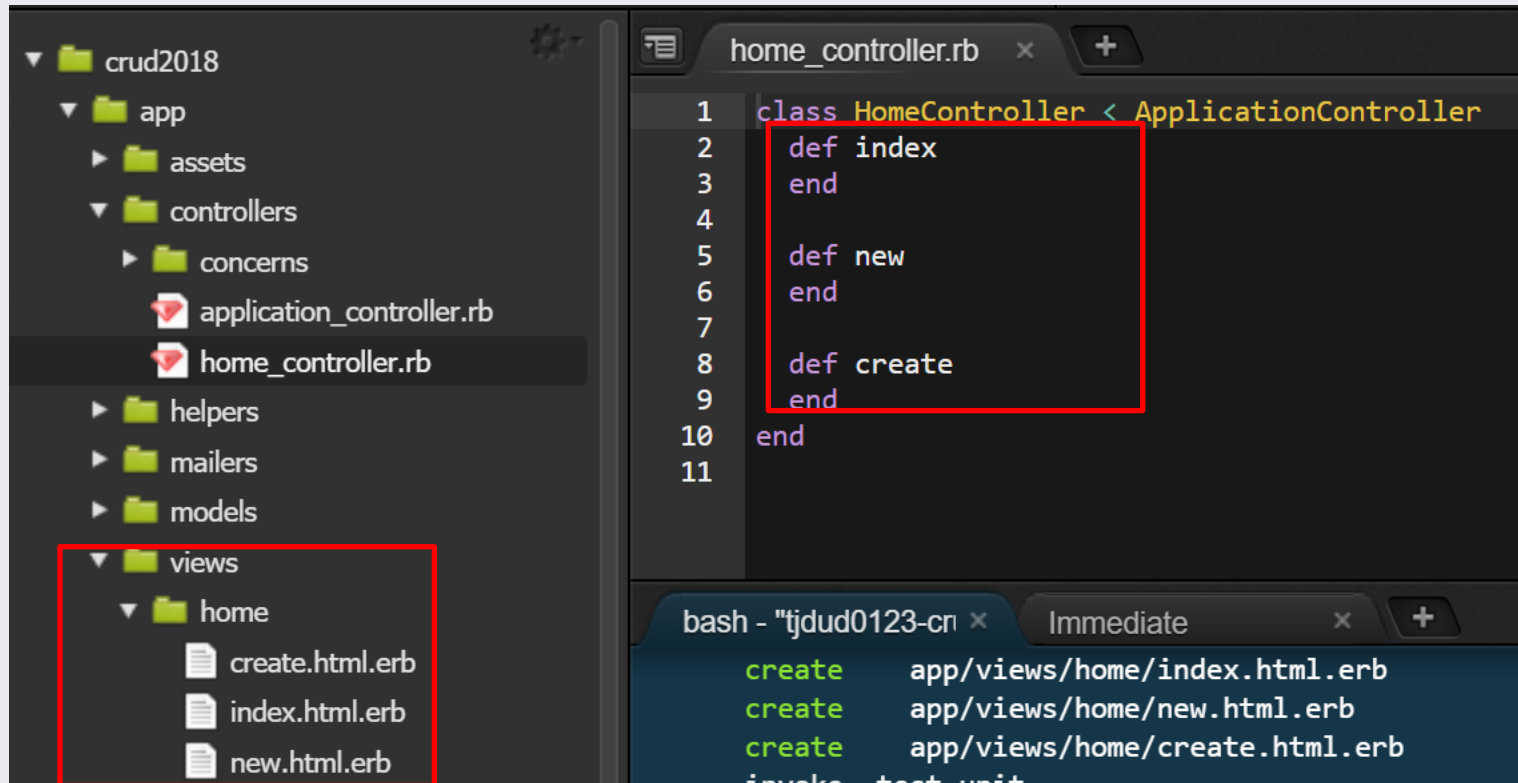
Index, new, create 액션과 뷰를 가진 home이라는 컨트롤러를 생성해라

A screenshot of a terminal window with a dark background. The title bar shows 'bash - "tjdud0123-cr' and 'Immediate'. The prompt is 'tjdud0123:~/workspace \$' and the command entered is 'rails g controller home index new create'.

```
bash - "tjdud0123-cr × Immediate × +  
tjdud0123:~/workspace $ rails g controller home index new create
```

CRUD#1[CR]

뷰/액션 자동생성

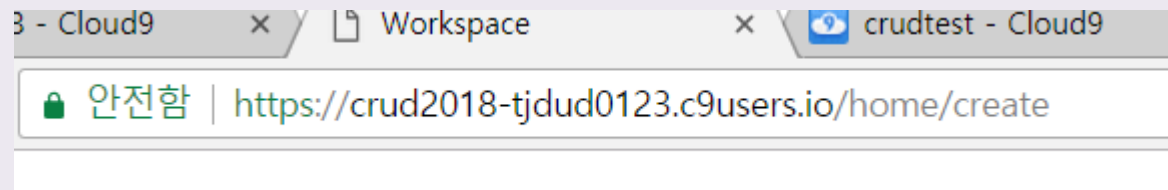


CRUD#1[CR]

config/<routes.rb>

```
Rails.application.routes.draw do
  root 'home#index'

  get 'home/index'
  #get 'home/index' => 'home#index'
  get 'home/new'
  #get 'home/new' => 'home#new'
  get 'home/create'
  #get 'home/create' => 'home#create'
end
```



같은 말!!

```
get 'home/index'  
#get 'home/new' => 'home#new'
```

```
post 'home/index'  
#post 'home/new' => 'home#new'
```

1단계 : index (게시물 목록 페이지) view 파일 작성

Views/home/<Index.html.erb>

The screenshot shows a code editor with three tabs: `home_controller.rb`, `routes.rb`, and `index.html.erb`. The `index.html.erb` file contains the following code:

```
1 <h1>여기는 게시물 목록이 나올 페이지 입니다!</h1>
2 <a href='/home/new'>새로운 글 쓰기</a>
3
```

Below the code editor, a web browser window is shown. The address bar displays `https://crud2018-tjdud0123.c9users.io`. The browser content shows the rendered HTML:

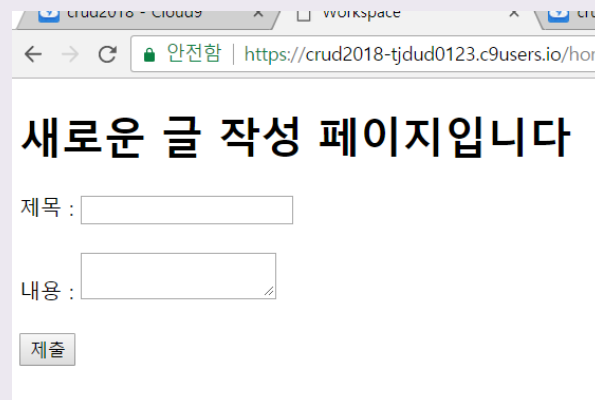
여기는 게시물 목록이 나올 페이지 입니다!

[새로운 글 쓰기](/home/new)

2단계 : new (글 작성 페이지) view 파일 작성

<new.html.erb>

```
1 <h1>새로운 글 작성 페이지입니다</h1>
2 <form action='/home/create' method='post'>
3
4   제목 : <input type='text' name='post_title'><br><br>
5   내용 : <textarea name='post_content'></textarea><br><br>
6   <input type='submit' value='제출'>
7 </form>
```



crud2018 - Cloud9 workspace

← → ↻ 🔒 안전함 | <https://crud2018-tjdud0123.c9users.io/home>

새로운 글 작성 페이지입니다

제목 :

내용 :

Error?



CSRF

웹 페이지에 악성코드나 링크를 포함하는 공격방법

```
1 <h1>새로운 글 작성 페이지입니다</h1>
2 <form action='/home/create' method='post'>
3
4   제목 : <input type='text' name='post_title'><br><br>
5   내용 : <textarea name='post_content'></textarea><br><br>
6   <input type='submit' value='제출'>
7 </form>
```

<https://stackoverflow.com/questions/8503447/rails-how-to-add-csrf-protection-to-forms-created-in-javascript>

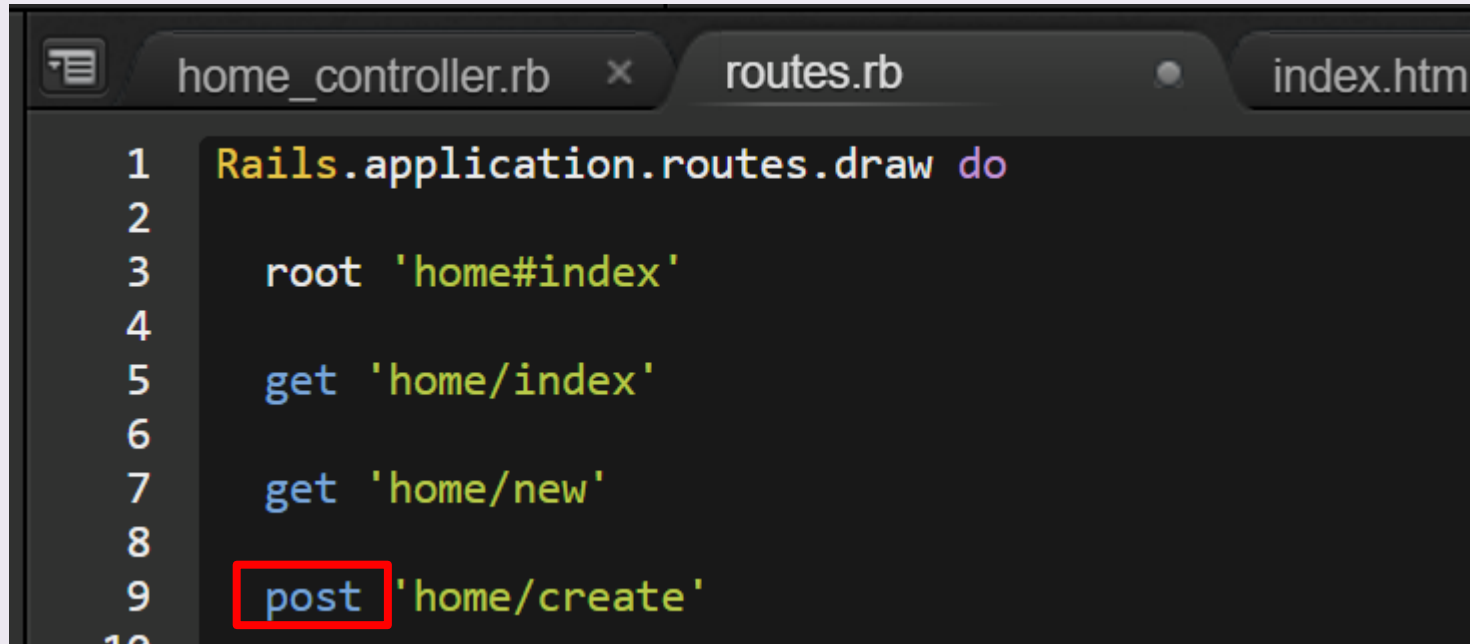
```
<%= hidden_field_tag :authenticity_token, form_authenticity_token %>
```

2단계 : new (글 작성 페이지) view 파일 작성

<new.html.erb>

```
1 <h1>새로운 글 작성 페이지입니다</h1>
2 <form action='/home/create' method='post'>
3   <%= hidden_field_tag :authenticity_token, form_authenticity_token %>
4   제목 : <input type='text' name='post_title'><br><br>
5   내용 : <textarea name='post_content'></textarea><br><br>
6   <input type='submit' value='제출'>
7 </form>
```

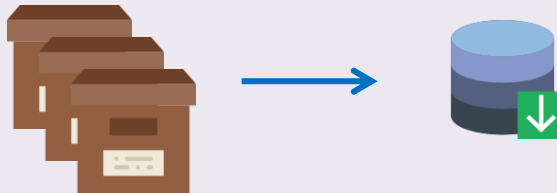
3단계 : route 작성(수정)



```
1 Rails.application.routes.draw do
2
3   root 'home#index'
4
5   get 'home/index'
6
7   get 'home/new'
8
9   post 'home/create'
```

4단계 : DB만들기

DB : 영구적 정보저장



제목, 내용을 저장할 테이블(표)

+

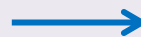
입력한 정보를 테이블에 넣어주는 기능



Model

+

Create action



4단계 : DB만들기

rails g(enerate) model **모델이름**

모델을 생성해보자

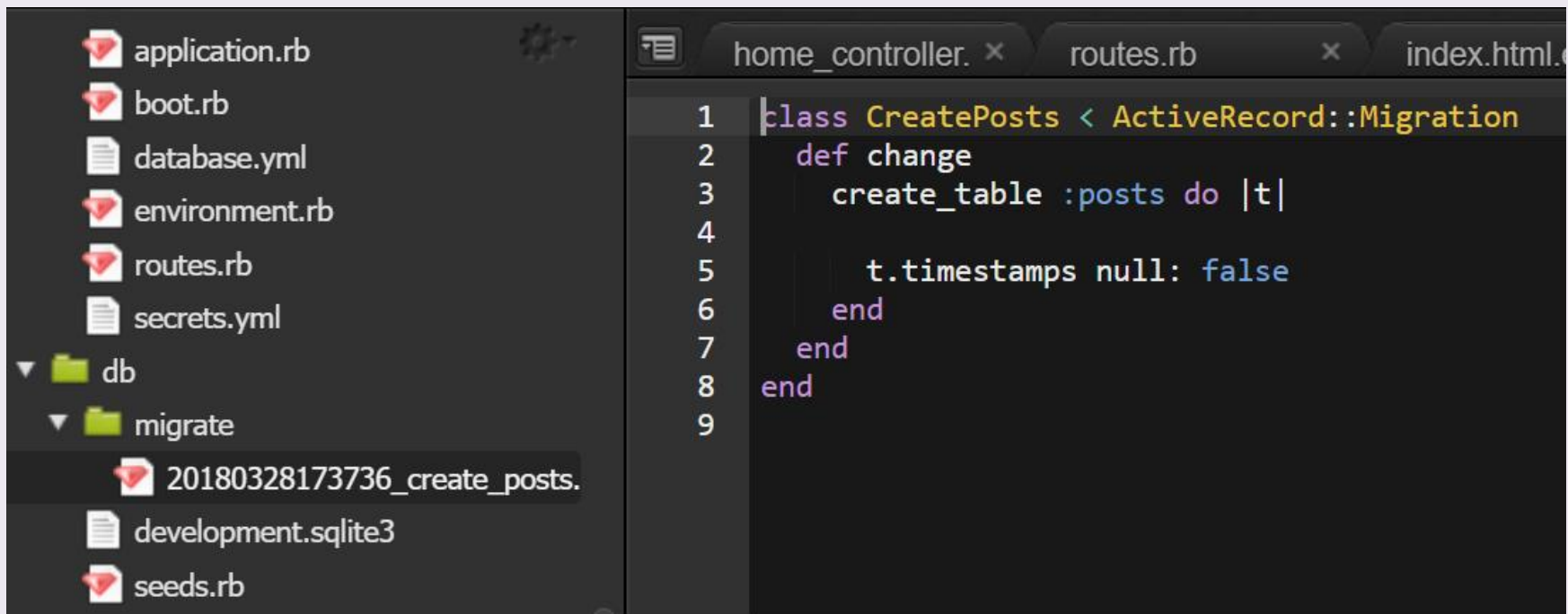
```
tjdud0123:~/workspace $ rails g model post
```

```
ec2-user:~/environment/crud $ rails g model post
Running via Spring preloader in process 6598
  invoke  active_record
  create  db/migrate/20180128172020_create_posts.rb
  create  app/models/post.rb
  invoke  test_unit
  create  test/models/post_test.rb
  create  test/fixtures/posts.yml
```

4단계 : DB만들기

<db/migrate/2018.....>

테이블 파일



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with files like application.rb, boot.rb, database.yml, environment.rb, routes.rb, secrets.yml, db, migrate, 20180328173736_create_posts., development.sqlite3, and seeds.rb. The code editor shows the content of the 20180328173736_create_posts. file, which is a Rails migration class named CreatePosts. The code defines a change method that creates a table named :posts with a timestamps column that is not null.

```
1 class CreatePosts < ActiveRecord::Migration
2   def change
3     create_table :posts do |t|
4
5       t.timestamps null: false
6     end
7   end
8 end
9
```

4단계 : DB만들기

테이블의 모양 결정

```
class CreatePosts < ActiveRecord::Migration
  def change
    create_table :posts do |t|

      t.string :title
      t.text :content

      t.timestamps null: false
    end
  end
end
```

string		text
id	title	content

※id는 자동적 생성

string => 문자열

text => string보다 더 긴 문자열

<\$bash 창>

rake db:migrate

\$테이블 확정 명령어

rake db:migrate

\$테이블 삭제 명령어

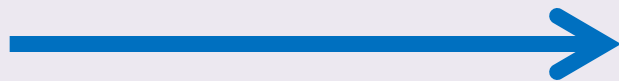
4단계 : DB만들기

id	title	contetnt

rake db:migrate

\$테이블 확정 명령어

id	title	contetnt



```
tjdud0123:~/workspace $ rake db:migrate
```

4단계 : DB만들기

<db/migrate/2018.....>

```

class CreatePosts < ActiveRecord::Migration
  def change
    create_table :posts do |t|
      t.string :title
      t.text :content

      t.timestamps null: false
    end
  end
end

```

column 생성!

id	title	content

string타입

Text타입

```
t.string :title  
t.text :content
```

t.string :title



string타입의 데이터를 저장하는
title이라는 제목을 가진 column을 만들겠다.



Quiz!

t.**text** : **content**



Quiz!

t.**text** : **content**

text타입의 데이터를 저장하는
content이라는 제목을 가진 column을 만들겠다.

4단계 : DB만들기

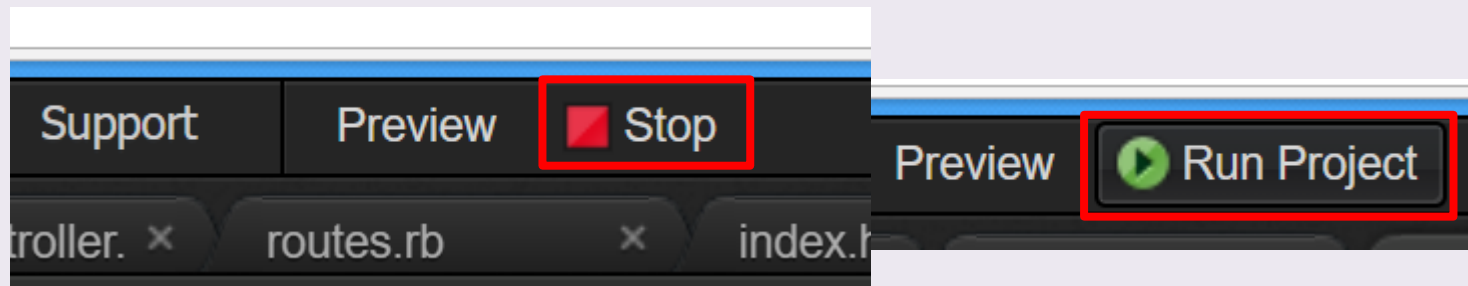
테이블 파일 완성 후 명령어 입력!

rake db:migrate

\$테이블 확정 명령어

```
tjdud0123:~/workspace $ rake db:migrate
```

※서버를 껐다 켜야 함!



5단계 : create 액션작성하기

<new.html.erb>

```
1 <h1>새로운 글 작성 페이지입니다</h1>
2 <form action='/home/create' method='post'>
3   <%= hidden_field_tag :authenticity_token, form
4   제목 : <input type='text' name='post_title'><b
5   내용 : <textarea name='post_content'></textare
6   <input type='submit' value='제출'>
7 </form>
```

post

'home/create'

// 'home create' => home#create

home controller의 create 액션을 수행하라

5단계 : create 액션작성하기

=>테이블에 데이터 채우기!

<home_controller.rb>

```
def create
  @post = Post.new //테이블에의 한 행을 추가
  @post.title = params[:post_title] //post_title이라는 이름의 내용을 title행 아래 저장
  @post.content = params[:post_content] //post_content라는 이름의 내용을 content행 아래 저장
  @post.save //테이블에 써준 내용을 모두 저장

  redirect_to '/home/index' //redirect_to '요청 URL주소' : 다시 '...'로 연결합니다
end
```

<new.html.erb>

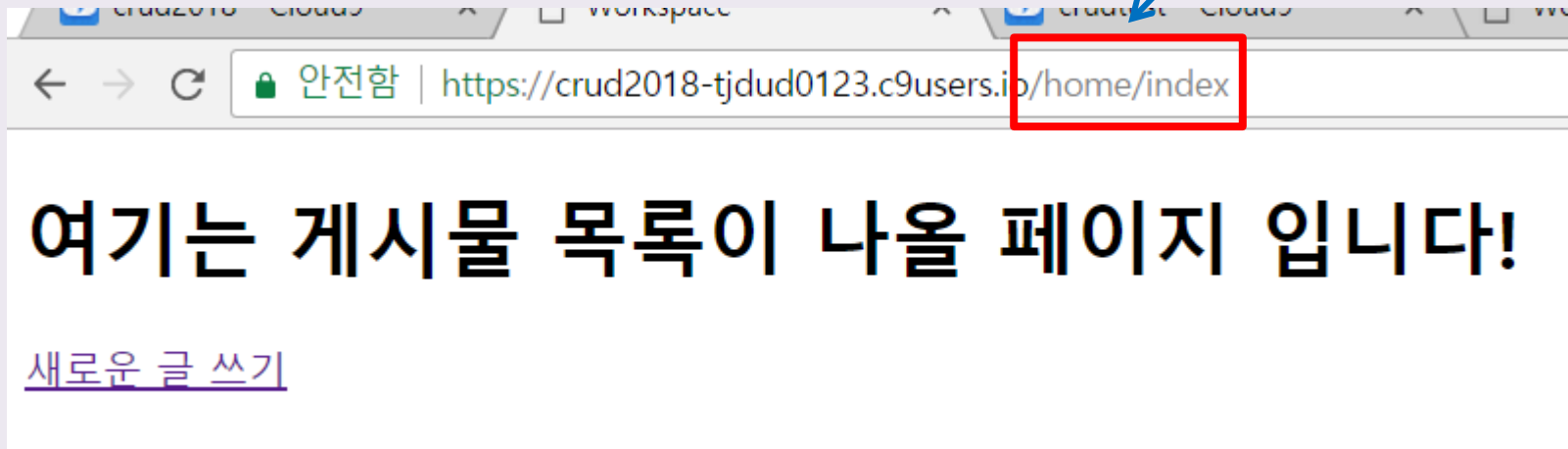
```
1 <h1>새로운 글 작성 페이지입니다</h1>
2 <form action='/home/create' method='post'>
3   <%= hidden_field_tag :authenticity_token, form_authenticity_token %>
4   제목 : <input type='text' name='post_title'></input>
5   내용 : <textarea name='post_content'></textarea>
6   <input type='submit' value='제출'>
7 </form>
```

CRUD#1[CR]

```
redirect_to '/home/index'
```

```
get 'home/index'
```

Index.html.erb파일



5단계 : create 액션작성하기

```
@post = Post.new
```



```
tjdud0123:~/workspace $ rails g model post
```

※모델이름 대문자로!

5단계 : create 액션작성하기

<home_controller.rb>

```
class HomeController < ApplicationController
  def index
  end

  def new
  end

  def create
    @post = Post.new
    @post.title = params[:post_title]
    @post.content = params[:post_content]
    @post.save

    redirect_to '/home/index'
  end
end
```

@post=Post.new (대문자 주의)

id	title	content

@post.title=params[:post_title]

id	title	content
	"멋쟁이 사자처럼"	

@post.content=params[:post_content]

id	title	content
	"멋쟁이 사자처럼"	"crud 강의"

@post.save

id	title	content
	"멋쟁이 사자처럼"	"crud 강의"

확인해 볼까나



```
tjdud0123:~/workspace $ rails c
```

//콘솔 창 켜기

```
2.3.4 :001 > Post.all
```

//post model에 저장된 데이터 보기

```
title: nil, content: nil, created_at: "2018-03-28 18:15:02", updated_at: "2018-03-28 18:15:02">, #<Post id: 3, title: "asdasd", content: "asdasd", created_at: "2018-03-28 18:23:15", updated_at: "2018-03-28 18:23:15">]>
```

```
2.3.4 :002 > exit
```

//콘솔 창 나가기

6단계 : index 액션작성하기

=>테이블 전체를 불러오기!

<home_controller.rb>

```
class HomeController < ApplicationController
  def index
    @posts = Post.all
  end
end
```

여기는 게시물 목록

새로운 글 쓰기

제목 : 안녕

내용 : 와우웅와

[삭제]

제목 : 끼요우우웅

내용 : 후헤헤헤헿

[삭제]

7단계 : index (게시물 목록 페이지) view파일 수정

Views/home/<Index.html.erb>

```
<h1>여기는 게시물 목록이 나올 페이지 입니다!</h1>
<a href='/home/new'>새로운 글 쓰기</a>

<br><br>
<%@posts.each do |x|%>
제목 : <%=x.title%> <br>
내용 : <%=x.content%> <br>
-----<br>
<%end%>
```

※참고

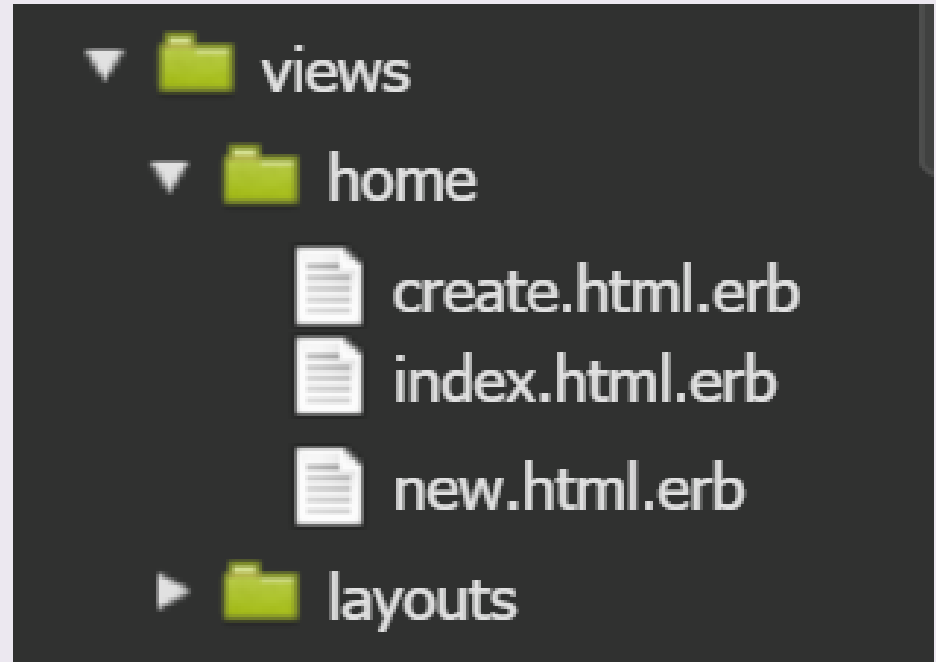
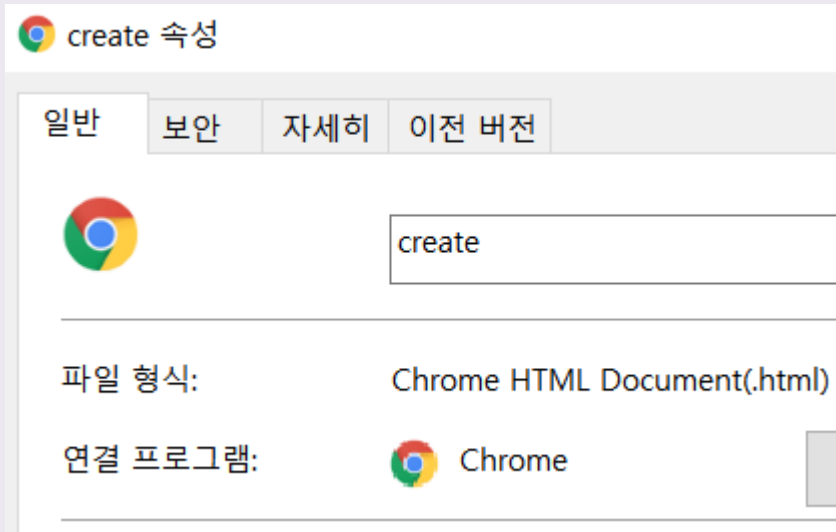
<%...%> 그 외

<%=...%> 출력

<home_controller.rb>

```
class HomeController < Appl
  def index
    @posts = Post.all
  end
end
```

CRUD#1[CR]



.html ↔ .html.erb

차이점...?



.html.erb

.html 파일에 루비문법을 사용할 수 있다!

```
<%=posts.each do |x|%>
제목 : <%=x.title%> <br>
내용 : <%=x.content%> <br>
-----<br>
<%end%>
```

<%...%> : 그 외

<%=...%> : 출력 시 사용

7단계 : index (게시물 목록 페이지) view파일 수정

Views/home/<Index.html.erb>

```
<%@posts.each do |x|%> //한 게시물 씩 작업 => .each  
제목 : <%=x.title%> <br> //그 행의 title에 들어있는 내용  
내용 : <%=x.content%> <br> //그 행의 content에 들어있는 내용  
-----<br>  
<%end%> //한 게시물씩(테이블 한 줄씩) do~end 사이의 작업 반복
```

CR! Create & Read 실습시작!!

Workspace name

crud2018

Description

Make a short description of your workspace

[Hosted workspace](#)
[Clone workspace](#)
[Remote SSH workspace](#)
[Salesforce](#)

☐
Private
 This is a workspace for your eyes only

☐
Public
 This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template

HTML5

Node.js

PHP, Apache & ...

Python

Django

Ruby

C++

Wordpress

Rails Tutorial

Blank

Harvard's CS50

Create workspace

2018.4.3

EWHA LIKELION

CR! Create & Read 7단계

1단계 : index (게시물 목록 페이지) view 파일 작성

2단계 : new (글 작성 페이지) view 파일 작성

3단계 : route 작성(수정)

4단계 : DB만들기

5단계 : create 액션작성하기

6단계 : index 액션작성하기

7단계 : index (게시물 목록 페이지) view파일 수정

CRUD#2[UD]

넘모 넘모
수고하셨습니다~!

