

멋쟁이사자처럼 심화 스터디 가이드라인

17년 1월 3주차

LIKE LION

01

3주차 목표

레.알.못

우리는 사실 Rails를 잘 모른다.

LIKE LION

01

3주차 목표



Rails CRUD - Reboot

- 정말 아무 것도 안 보고 혼자 다 만들 수 있음? 진짜?



\$ rails g scaffold

- g 의 의지 no.3

LIKE LION

02

3주차 개인 과제



Create, Read, Update, Delete

CRUD 라는 단어만 기억에 남은 당신을 위한 Back to the code.

LIKE LION

03

Ruby on Rails 과제 Direction

우리가 만들고자 하는 것.

디자인 전혀 없는
가장 간단한 게시판

* scaffolding 사용 금지! 밑바닥 부터 코딩!

Ruby on Rails 과제 Direction

C9 환경

- **ruby** version = C9 default
- **rails** version = C9 default
- 프로젝트 이름 = crud_reboot

Local 환경

- **ruby** version = 2.3.3
- **rails** version = 5.0.1
- 프로젝트 이름 = crud_reboot

03

Ruby on Rails 과제 Direction

\$ rails g 명령어를 통해 만들 것들

model 이름 = Post

controller 이름 = Posts

03

Ruby on Rails 과제 Direction

Post **Model**의 구성 (columns)

- **title** = 게시글의 제목 (**string** 타입)
- **content** = 게시글의 내용 (**text** 타입)

마이그레이션 파일에서 다른건 건드리지 않고
title과 **content** 만 추가하도록 하겠습니다.

03

Ruby on Rails 과제 Direction

`posts_controller.rb`의 액션들

`index` : `index.html.erb` 뷰 파일을 위한 액션

`show` : `show.html.erb` 뷰 파일을 위한 액션.

`new` : `new.html.erb` 뷰 파일을 위한 액션.

`create` : 새로운 `post`에 입력받은 값을 DB에 저장하는 액션.

`edit` : `edit.html.erb` 뷰 파일을 위한 액션.

`update` : 특정 `post`를 수정하여 DB에 저장하는 액션.

`delete` : 특정 `post`를 DB에서 삭제하는 액션.

03

Ruby on Rails

과제 Direction

우리가 사용할 **View** 들과 포함해야 하는 **UI**.

- **index.html.erb** : 모든 **post**들을 모아서 제목만 보여줌.
 - ‘new post’를 클릭하면 새로운 **post**를 작성할 수 있는 **new.html.erb**로 이동.
 - 특정 **post**의 제목을 클릭하면 **show.html.erb**로 이동.
- **new.html.erb** : 새로운 **post**를 작성하는 form.
- **show.html.erb** : 특정 **post**의 **title**, **content**를 보여줌.
 - ‘edit post’를 클릭하면 해당 **post**를 수정할 수 있는 **edit.html.erb**로 이동.
 - ‘delete post’를 누르면 해당 **post**를 삭제.
- **edit.html.erb** : 특정 **post**의 **title**, **content**를 수정하는 form.

03

Ruby on Rails

과제 Direction

제출 방법

`routes.rb` 파일의 코드

`post_controller.rb` 파일의 코드

각 `View File` 들의 코드

들을

제공하는 양식에 맞춰 1개의 txt 파일에 복사-붙여넣기 하여 제출.
(`render_partial` 을 사용했을 경우 해당 `_form` 파일도 제출.)

다시 말씀드리지만 디자인은 참아주세요ㅠ

LIKE LION

03

개인 과제

Q&A

Q : 뷰 파일들 이름이 왜 .html.erb죠??

강의에서는 .erb라고 했던것 같은데..

A : .erb와 .html.erb는 완전히 같은 파일입니다.

둘중에 어떤걸 사용하셔도 상관 없습니다!

03

개인 과제

Q&A

Q : **액션**이랑 **뷰 파일**들이 4기 강의랑은 다르게 왜 이렇게 많죠..?

굳이 이렇게 많이 만들어야 해요?? 번거롭게;

A : 모두 알고 계시겠지만, 이두희 대장님의 강의는
‘야매정신’

을 기본으로 하고있습니다... 즉 정석이 아니라는 얘기죠.

과제 Direction의 액션/뷰 구성이
Rails CRUD 기본 구성의 정석입니다!

LIKE LION

04

Rails - 스터디 권장



\$ rails g scaffold 로 생성된 코드를 파헤쳐서
가장 Rails스러운 코드란 무엇인지 알아보자.

Rails - 스터디 권장

스터디를 위한 사전준비.

1. 새로운 rails 프로젝트 (프로젝트 이름 = learn_scaffold)
2. 터미널에
\$ rails g scaffold post title:string content:text 를 입력.
3. \$ rake db:migrate 를 통해 migrate 수행.
4. 서버를 실행하고, url 끝에 /posts 를 입력, 이동.
5. 이것 저것 눌러보며 감탄하기.

04

Rails - 스터디 권장

자 그러면 이제 이 **scaffold** 라는 녀석이
어떻게 동작하는지 살펴볼까?

Rails - 스터디 권장

Scaffold 뜯어보기 가이드 - 무엇을 봐야할까?

- app/controllers/posts_controller.rb 파일
- 터미널에 \$ rake routes 입력해보기.
#rails version > 5.0 은 \$ rails routes 도 가능
#혹은 브라우저 새 탭에 project url 뒤에 /rails/info 입력, 이동
- config/routes.rb 파일
- 모든 view file (.html.erb) 들

04

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

app/controllers/**posts_controller.rb**

컨트롤러는 5주차에 뜯어볼 예정!

지금은 2주차에 배운 Ruby를 바탕으로
최대한 코드를 해석해 보는 정도면 충분해요!

LIKE LION

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

`$ rake routes`

Rails version > 5.0은 `$ rails routes` 도 가능!

좀 더 큰 화면에서 보려면 `project_url/rails/info`

위의 명령어는 `routes.rb` 를 분석하여,
현재 사용되고 있는 라우트를 모아서
목록으로 보여주는 명령어 or 화면!

결과 화면은 다음과 같은 구성!

04

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

Prefix	Verb	URI pattern	Controller# Action
posts	GET	/posts(:format)	posts#index
	POST	/posts(:format)	posts#create
new_post	GET	/posts/new(:format)	posts#new
edit_post	GET	/posts/:id/edit(:format)	post#edit
book	GET	/posts/:id(:format)	posts#show
	PATCH	/posts/:id(:format)	posts#update
	PUT	/posts/:id(:format)	posts#update
	DELETE	/posts/:id(:format)	posts#destroy

04

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

Verb 에서 GET 이랑 POST는 뭔지 아는데..

routes.rb 랑 <form> 맨날 적던 그거 아니야?

GET 'post/index'

POST 'post/write'

<form action='/home/write' method='POST'>

그런데.. PATCH? PUT? DELETE?

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

강의에서는 GET 과 POST 만 알아도
웹사이트 만들 수 있다고 했잖아! 꼭 알아야 돼?

네. 이제는 알아야 합니다. 우리는 선생님이니깐요

만들 수 있는 것과, **잘** 만드는 것은 전혀 다른 얘기!

그런데 뭘 어떻게 잘 만드는 건데?

04

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

3주차 1번 목표

이제부터 우리는 **RESTful** 하게 잘 만든다!

Rails - 스터디 권장

RESTful 학습 가이드

1. HTTP 란 무엇인가?
2. HTTP method 에는 어떤것이 있으며 왜 있는가?
3. RESTful 이란 무엇인가?
4. 그래서 `$ rake routes` 가 말해주는게 뭐지?

04

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

app/views/**모든 View** 파일들

index.html.erb

show.html.erb

new.html.erb

edit.html.erb

_form.html.erb

(.json.jbuilder 확장자 파일들은 pass)

LIKE LION

04

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

3주차 2번 목표

View Helper 란 무엇인가?

04

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

지금까지 봐온 HTML 코드들은 잊어라!

Rails 유저라면 View는 View-Helper!

참고자료

퍼펙트 루비 온 레일즈 4장 View Helper

guides.rubyonrails.org

구.글.링

LIKE LION

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

Scaffolding 에서 사용된 View Helper 들은
최대한 모두 알고 가야 해요!

Scaffolding을 통해 View Helper 와 어느정도 친숙해 졌다면?
다른 View Helper들을 뜯어보자!

Rails - 스터디 권장

Scaffold 뜯어보기 가이드

gem 'devise'

\$ rails g devise:install

\$ rails g devise User

\$ rails g devise:views

devise 가 사용하는 View 들 역시 View Helper로 구성되어 있다!

Devise View Code들을 분석해보자!

2주차 스터디 정리

1. scaffold 뜯어보기 (routes.rb 와 View File 들 위주로)
 - scaffold 를 더 자세히 알아보고 싶다면?
 - **퍼펙트 루비온 레일즈 3장**을 강력하게 추천!
2. scaffold 와 devise 에 사용된 View Helper들 공부하기!
 - View File 과 실제 브라우저의 화면을 개발자 도구로 비교 분석 해보면 이해가 더욱 쏙쏙!

06

다음주 스포일러



LIKE LION

06

다음주 스포일러

Rails Model 의 모든것.

- 부제 : Do you know Active Record?

LIKE LION

07

정리

개인별로

1/27 (금) 까지

CRUD reboot

모여서

scaffold 뜯어보기

- controller.rb
- 라우팅
- View 파일들

추가로 뜯어보려면

- Devise View

설날 전에

최소 1회 스터디 하기
&
즐거운 연휴 보내기