

# Ruby on Rails

3주차

수요일

멋쟁이사자처럼  건국대학교

# Controller 생성

- Controller만 생성한 경우!

```
$ rails g(generate) controller [C]
```

[C] : 컨트롤러 이름

[An] : 액션 이름

```
create app/controllers/home_controller.rb
```

```
create app/views/home
```

# Controller 생성

- Controller와 Action을 동시에 생성하는 경우!

```
$ rails g controller [C] [A1] [A2] ..
```

[C] : 컨트롤러 이름  
[An] : 액션 이름

```
create    app/controllers/home_controller.rb
route     get 'home/index'
create    app/views/home
create    app/views/home/index.html.erb
```

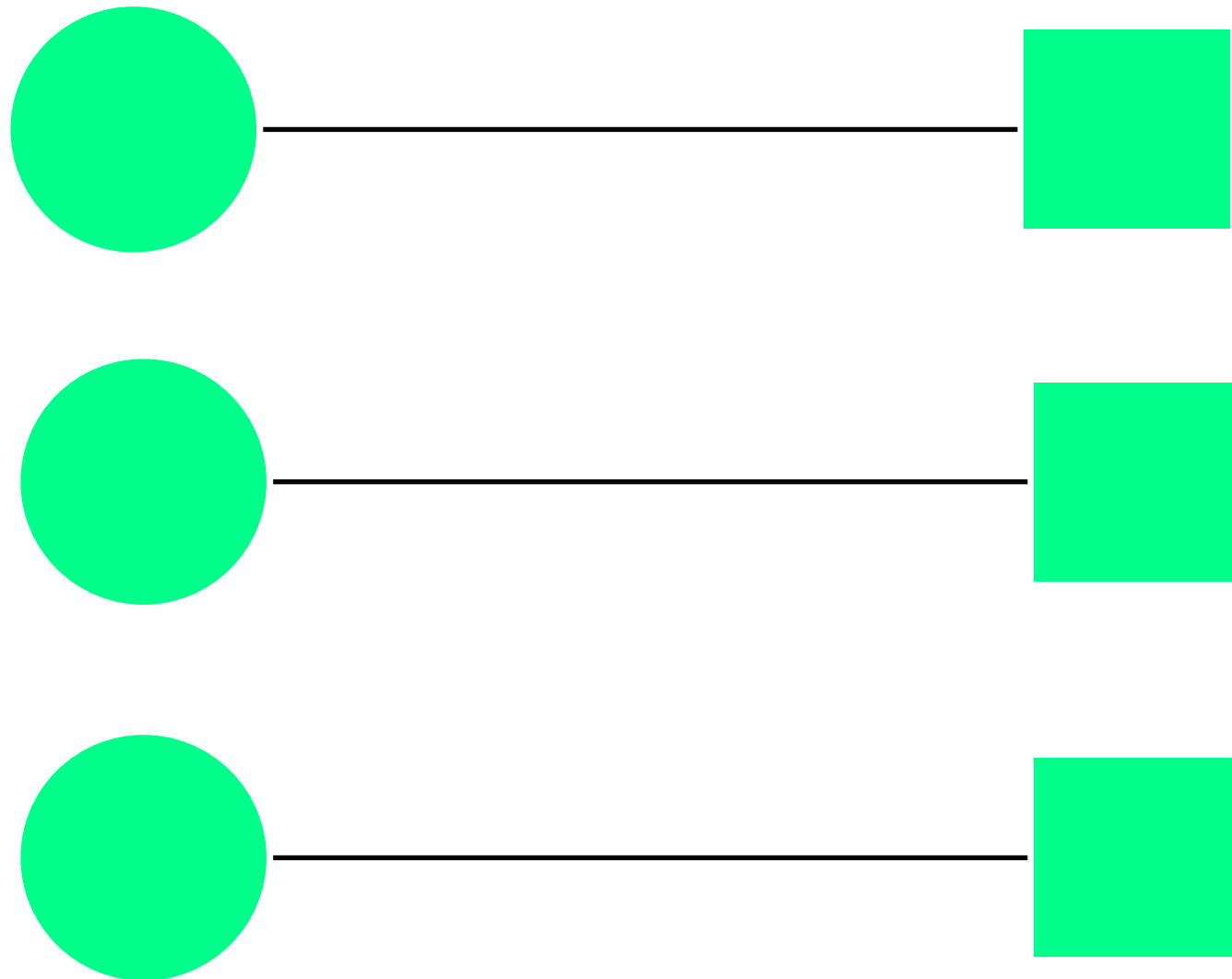
# Controller 삭제

```
$ rails d(delete) controller [C]
```

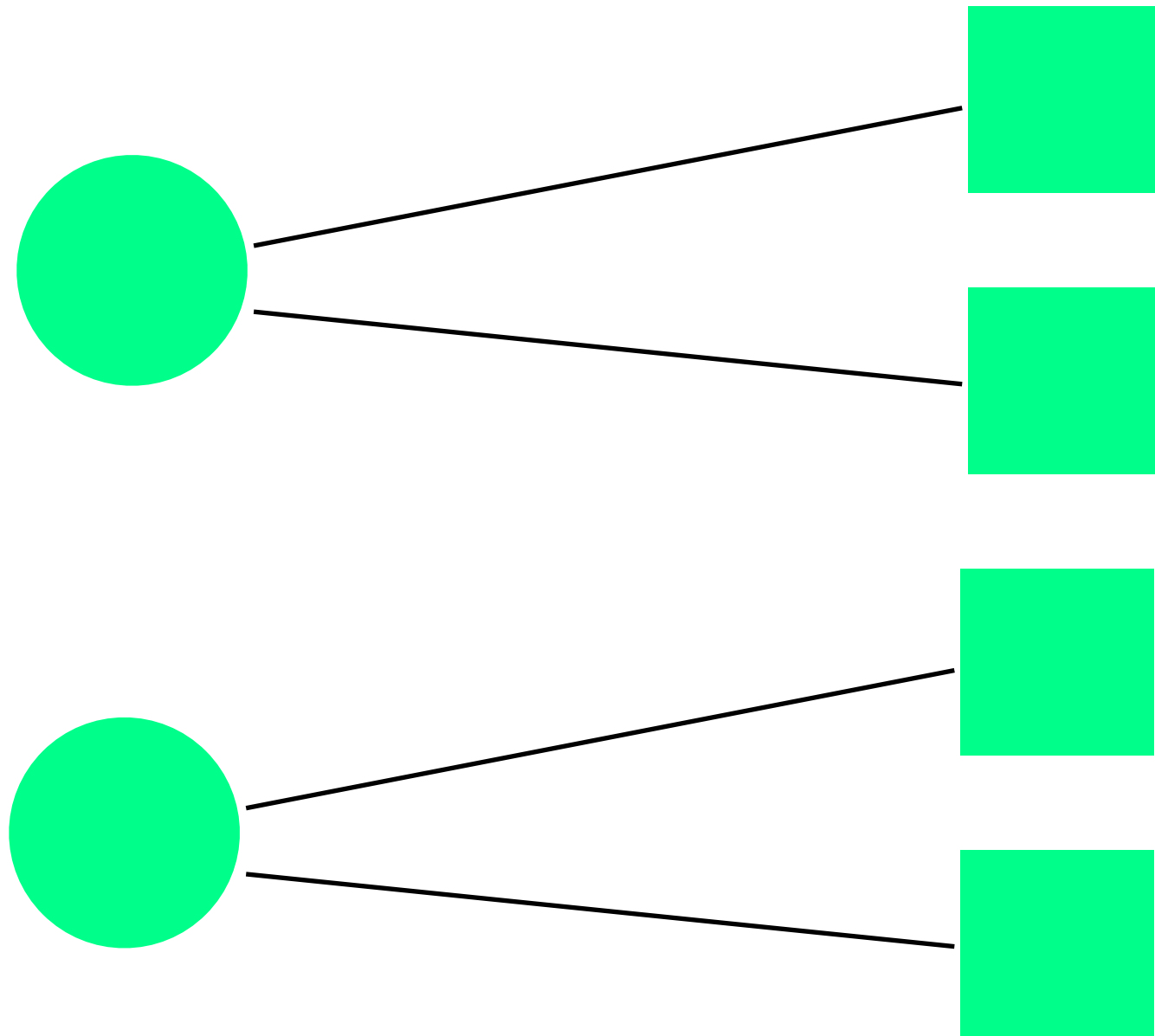
```
$ rails d controller [C] [A1] [A2] ..
```

[C] : 컨트롤러 이름  
[An] : 액션 이름

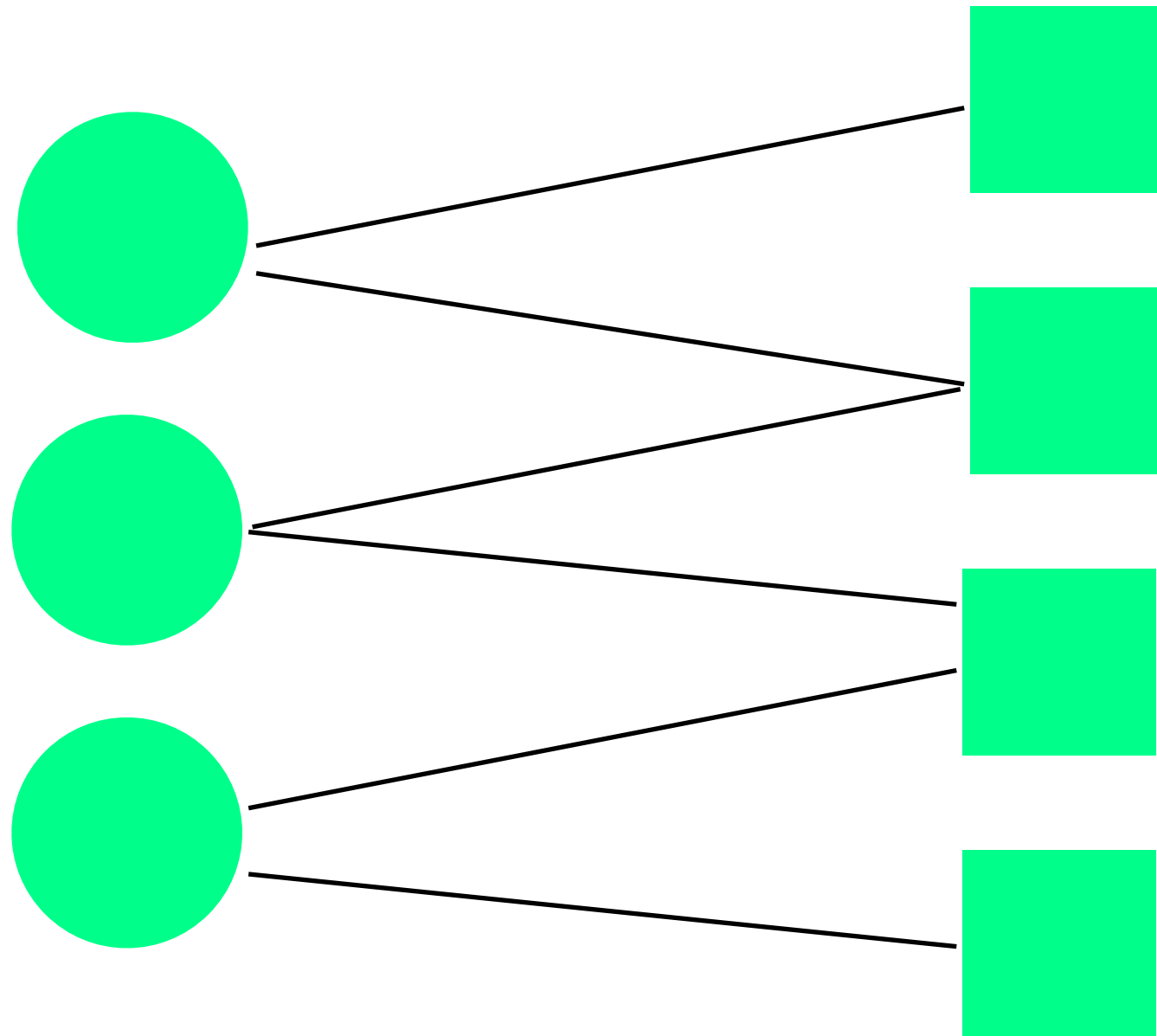
# MODEL 관계



# MODEL 관계



# MODEL 관계



# RESTful

장비간 통신을 위해 복잡한 방법을 사용하는 대신,  
간단하게 **HTTP**를 이용하는 것이 목적

HTTP URI를 통해 Resource를 명시  
HTTP Method(**Post, Get, Put, Delete**)를 통해  
해당 Resource에 대한 **CRUD** Operation을 적용





# CRUD

Create(생성), Read(읽기), Update(갱신), Delete(삭제)  
사용자 인터페이스가 갖추어야 할 기능(정보의 참조/검색/갱신)



CREATE

C



READ

R



UPDATE

U



DELETE

D

# ROUTES

- 요청받은 URL을 인식
- 적절한 컨트롤러의 액션이나 애플리케이션에 매칭
- 경로나 URL도 제공

```
#routes.rb
```

```
Rails.application.routes.draw do  
  resources :posts  
end
```

resources  
표준적인 routes(RESTful)

# Resources

\$ rake routes (rails routes)

← 정의된 route 확인

Prefix	Verb	URI Pattern	Controller#Action	역할
posts	GET	/posts(.:format)	posts#index	목록 화면 표시
	POST	/posts(.:format)	posts#create	새로운 데이터 등록화면의 입력을 받아 데이터 등록 처리
new_post	GET	/posts/new(.:format)	posts#new	새로운 데이터 등록 화면 표시
edit_post	GET	/posts/:id/edit(.:format)	posts#edit	수정 화면 표시
post	GET	/posts/:id(.:format)	posts#show	개별 상세 화면 표시
	PATCH	/posts/:id(.:format)	posts#update	수정화면의 입력을 받아 수정 처리
	PUT	/posts/:id(.:format)	posts#update	
	DELETE	/posts/:id(.:format)	posts#destroy	목록화면에서 지정된 데이터 제거처리

MVC를 이용해 CRUD 만들기 

# <form> 태그

- 사용자에게 입력을 받기 위한 HTML 태그
  - <input>
  - <select>
  - <option>

# View helper

- 레일즈에서 form 마크업을 생성하기 위한 뷰 헬퍼를 제공

```
<%= form_for(post) do |f| %>

  <div class="field">
    <%= f.label :content %>
    <%= f.text_area :content %>
  </div>

  <div class="actions">
    <%= f.submit %>
  </div>

<% end %>
```

# View helper

- 레일즈에서 form 마크업을 생성하기 위한 뷰 헬퍼를 제공

```
<form action="/users/sign_in" accept-charset="UTF-8"
method="post">

  <div class="field">
    <label for="post_content">Content</label>
    <textarea name="post[content]" id="post_content"></
textarea>
  </div>

  <div class="actions">
    <input type="submit" name="commit" value="Create Post"
data-disable-with="Create Post">
  </div>

</form>
```

CRUD 개념을 제대로 모르겠어요ㅠㅠ







멋쟁이사자처럼  
건국대학교