

# In this lecture, we will discuss...

- ✧ Nested URI
- ✧ Collection Resource
- ✧ `rake routes` – nested URI
- ✧ Movie example

# Nested URI

✧ `$ rails g scaffold_controller MovieRole  
character actor_id`

```
Rails.application.routes.draw do
  resources :movies do
    resources :movie_roles, as: :role, path: "roles"
  end
  resources :actors
```



# rake routes

## ✧ Before `:as` property

```
$ rake routes
```

	Prefix	Verb	URI Pattern	Controller#Action
movie_movie_roles		GET	/movies/:movie_id/movie_roles(.:format)	movie_roles#index
movie_movie_role		GET	/movies/:movie_id/movie_roles/:id(.:format)	movie_roles#show

## ✧ After `:as` property

```
$ rake routes
```

	Prefix	Verb	URI Pattern	Controller#Action
movie_roles		GET	/movies/:movie_id/roles(.:format)	movie_roles#index
movie_role		GET	/movies/:movie_id/roles/:id(.:format)	movie_roles#show



# Controller

✧ `app/controllers/movie_roles_controller.rb`

```
before_action :set_movie_role, only: [:show, :edit, :update, :destroy]

# GET /movie_roles/1
# GET /movie_roles/1.json
def show
end

...

def set_movie_role
  @movie_role = Movie.find(params[:movie_id]).roles.find_by(:id=>params[:id])
end
```



# JSON Marshaller

- ✧ Default JSON marshaller definition expects timestamp
- ✧ `app/views/movie_roles/show.json.jbuilder`

```
json.extract! @movie_role, :id, :character, :actor_id, :created_at, :updated_at
```

- ✧ remove field for display

```
json.extract! @movie_role, :id, :character, :actor_id
```

# Access URI – MovieRole

✧ MovieRole as a nested resource (**Single**) below Movie

```
> HTTParty.get("http://localhost:3000/movies/12345/roles/0.json").parsed_response  
=> {"id"=>"0", "character"=>"rocky", "actor_id"=>"100"}
```

✧ Nested Resource (**Collection**)

```
> HTTParty.get("http://localhost:3000/movies/12345/roles.json").parsed_response  
[]
```



# Nested Resource - Collection

```
> movie.roles.create(:id=>"1",:character=>"challenger")  
> Movie.find("12345").roles  
=> [#<MovieRole _id: 0, character: "rocky", actor_id: "100">,  
     #<MovieRole _id: 1, character: "challenger", actor_id: nil>]
```

- ✧ Define `before_action` and update `set_movie_role`
- ✧ Update JSON marshaller



# Controller and index changes

```
class MovieRolesController < ApplicationController
  before_action :set_movie
  before_action :set_movie_role, only: [:show, :edit, :update, :destroy]
  ...

  def set_movie_role
    @movie_role = @movie.roles.find_by(:id=>params[:id])
  end
  def set_movie
    @movie = Movie.find(params[:movie_id])
  end
end
```

```
# GET /movie/:movie_id/roles
# GET /movie/:movie_id/roles.json
def index
  @movie_roles=@movie.roles
end
```





# JSON Marshaller

- ✧ Add the `@movie` as a parameter to the `movie_role_url` helper method.

```
json.array!(@movie_roles) do |movie_role|  
  json.extract! movie_role, :id, :character, :actor_id  
  json.url movie_role_url(movie_role, format: :json)  
end
```

```
json.array!(@movie_roles) do |movie_role|  
  json.extract! movie_role, :id, :character, :actor_id  
  json.url movie_role_url(@movie, movie_role, format: :json)  
end
```



# Nested Resource - Collection

```
> pp HTTPParty.get("http://localhost:3000/movies/12345/roles.json").parsed_response
[{"id"=>"0",
  "character"=>"rocky",
  "actor_id"=>"100",
  "url"=>"http://localhost:3000/movies/12345/roles/0.json"},
 {"id"=>"1",
  "character"=>"challenger",
  "actor_id"=>nil,
  "url"=>"http://localhost:3000/movies/12345/roles/1.json"}]
```



# Summary

- ✧ Collection resource URI
- ✧ Nested data

## What's Next?

- ✧ Query Parameters and Payload