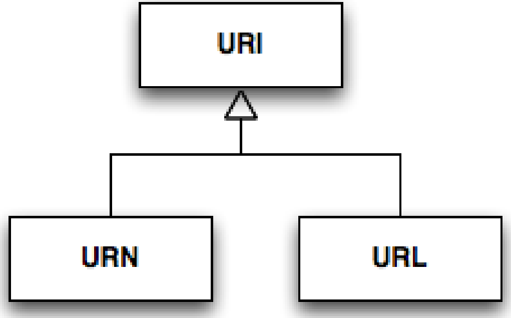
# In this lecture, we will discuss…

- URI – Uniform Resource Indicator
- Exposing resource as URI
- `rake routes`
- httparty

# URI vs. URL vs. URN

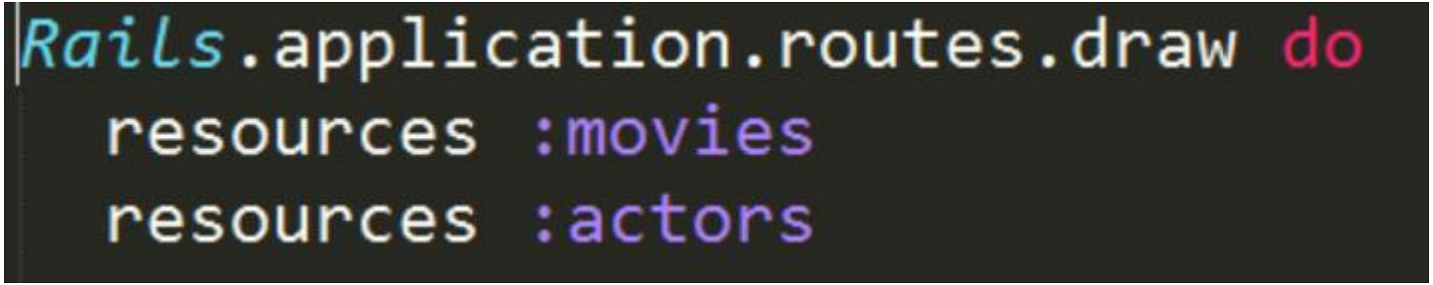
- A **Uniform Resource Identifier** (URI)
  - string of characters which identifies an Internet resource
  - www.coursera.org
- A **Uniform Resource Locator**(URL)
  - Most common URI out there
  - http://www.coursera.org
- A **Uniform Resource Name** (URN)
  - Another form of URI
  - urn:isbn:0-619-0125356-5



# URI

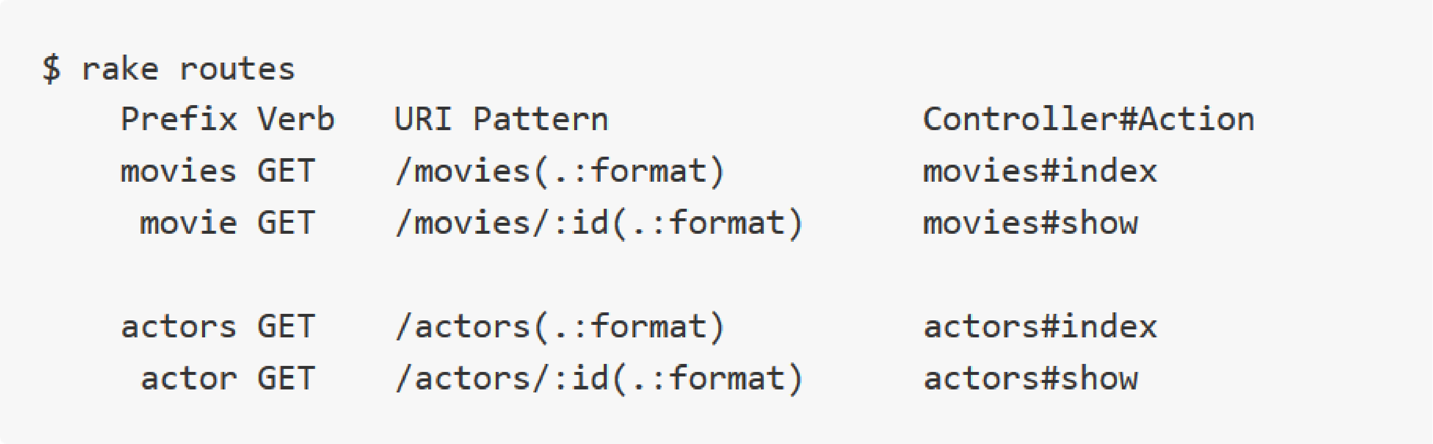
- Expose the resources using standard URIs
- Rails will automatically create URIs
  - Will register the resource in config/routes.rb
- `$ rails g scaffold_controller Movie title`
- `$ rails g scaffold_controller Actor name`

# config/routes.rb



```ruby
Rails.application.routes.draw do
  resources :movies
  resources :actors
```

# rake routes



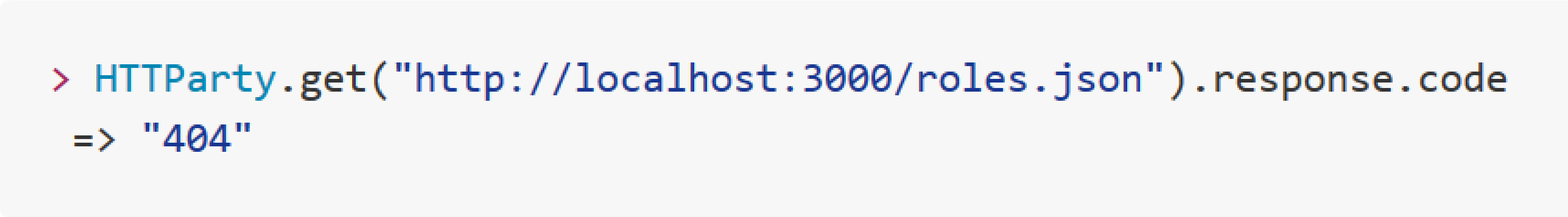
```
$ rake routes
    Prefix Verb   URI Pattern                Controller#Action
    movies GET    /movies(.:format)          movies#index
     movie GET    /movies/:id(.:format)      movies#show

    actors GET    /actors(.:format)          actors#index
     actor GET    /actors/:id(.:format)      actors#show
```

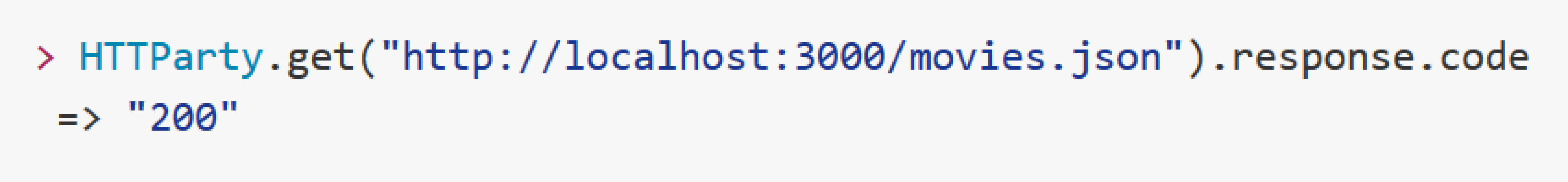
***Scaled down to show unique URI – methods not shown***

# Access URI

- gem ‘httparty’

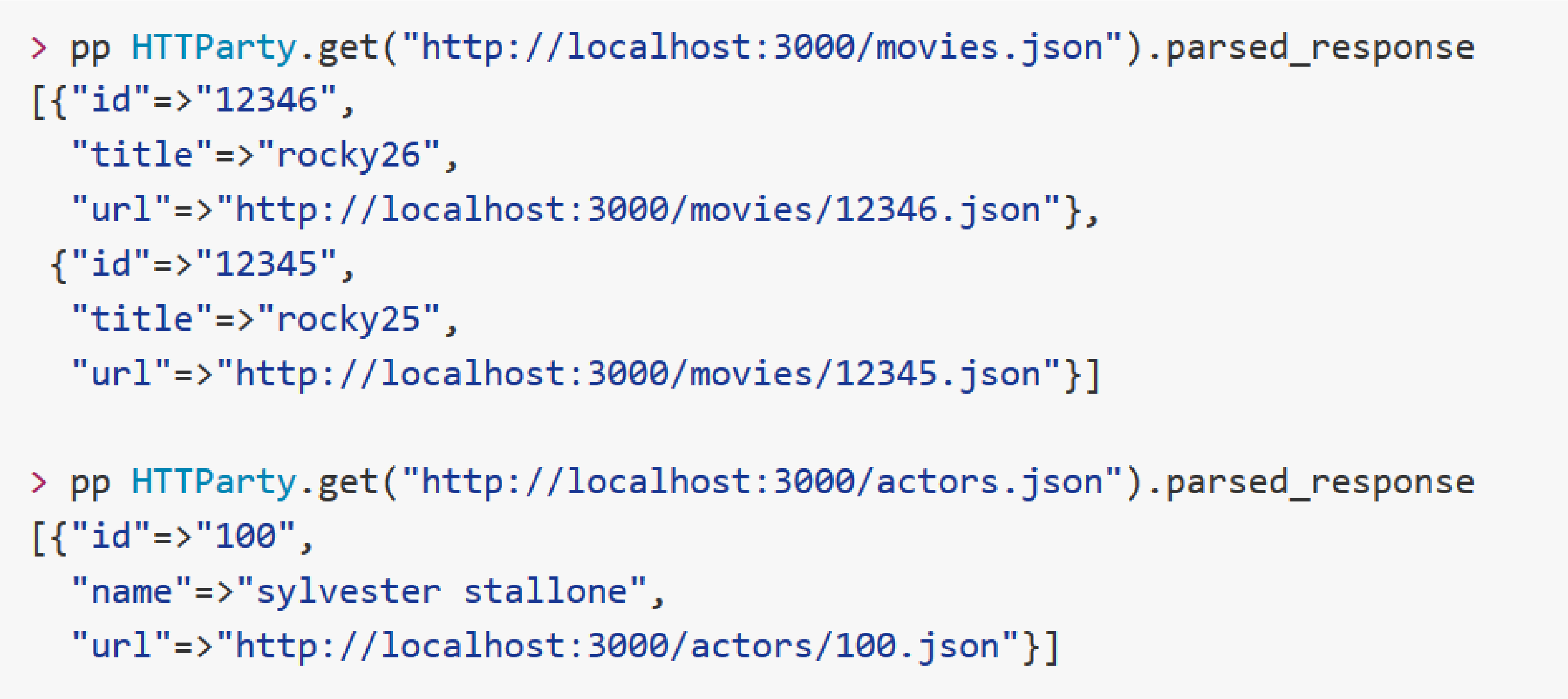


```
> HTTParty.get("http://localhost:3000/roles.json").response.code
 => "404"
```



```
> HTTParty.get("http://localhost:3000/movies.json").response.code
 => "200"
```

# Access URI – Actors and Movies

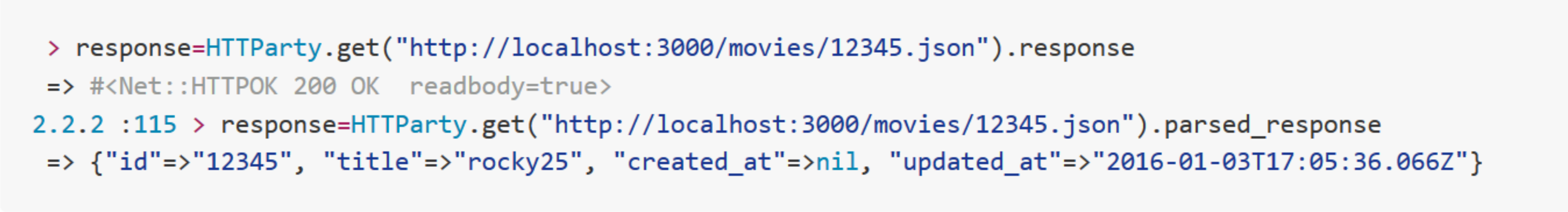


```
> pp HTTParty.get("http://localhost:3000/movies.json").parsed_response
[{"id"=>"12346",
  "title"=>"rocky26",
  "url"=>"http://localhost:3000/movies/12346.json"},
 {"id"=>"12345",
  "title"=>"rocky25",
  "url"=>"http://localhost:3000/movies/12345.json"}]

> pp HTTParty.get("http://localhost:3000/actors.json").parsed_response
[{"id"=>"100",
  "name"=>"sylvester stallone",
  "url"=>"http://localhost:3000/actors/100.json"}]
```

# Access URI – Movie

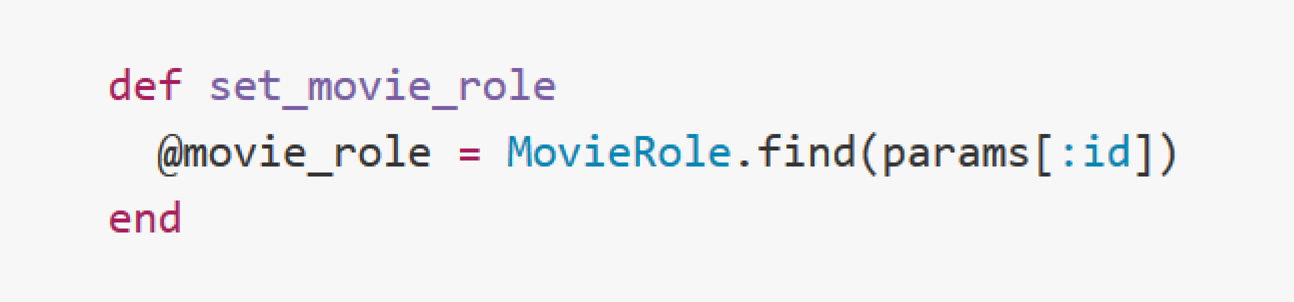
- ✧ Specific resource
  - `/movies/:id` and `/actors/:id`



```
 > response=HTTParty.get("http://localhost:3000/movies/12345.json").response
 => #<Net::HTTPOK 200 OK  readbody=true>
2.2.2 :115 > response=HTTParty.get("http://localhost:3000/movies/12345.json").parsed_response
 => {"id"=>"12345", "title"=>"rocky25", "created_at"=>nil, "updated_at"=>"2016-01-03T17:05:36.066Z"}
```

# Controller

- Update MovieRoles controller



```
def set_movie_role
  @movie_role = MovieRole.find(params[:id])
end
```

# Summary

- ✧ HTTP provides an excellent interface to implement RESTful services with features like a URI and existing HTTP states

**What’s Next?**

- ✧ Nested Resource URI