



A request with parameters

Sergio Revilla
@elrepicante

based on Martin Tschischauskas (@martynasc) slides

Imagine you want to load a
single location...

How do you identify it in a
URL?

/locations/384

id attribute

/locations/party

name attribute

/locations

Get all Locations

/locations/482

Get location with id=482

/locations

Get all the locations → #index

/locations/482

Get locations with id=482 → #show

/locations/482

Get locations with `id=482` → **#show**

/locations/34

/locations/81

...

`/locations/<id>`

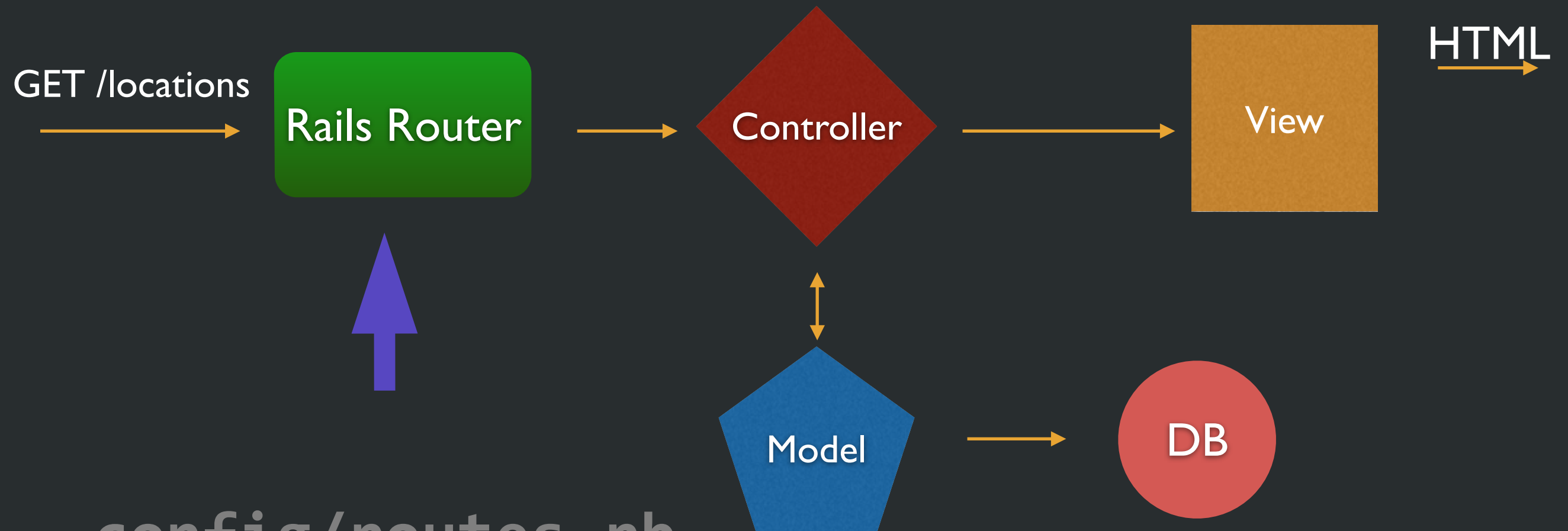


Parameter!

CRUD verbs and actions

| HTTP Verb | Path | Controller#Action | Used for |
|-----------|------------------|-------------------|--|
| GET | /photos | photos#index | display a list of all photos |
| GET | /photos/new | photos#new | return an HTML form for creating a new photo |
| POST | /photos | photos#create | create a new photo |
| GET | /photos/:id | photos#show | display a specific photo |
| GET | /photos/:id/edit | photos#edit | return an HTML form for editing a photo |
| PATCH/PUT | /photos/:id | photos#update | update a specific photo |
| DELETE | /photos/:id | photos#destroy | delete a specific photo |

<http://guides.rubyonrails.org/routing.html#crud-verbs-and-actions>



config/routes.rb

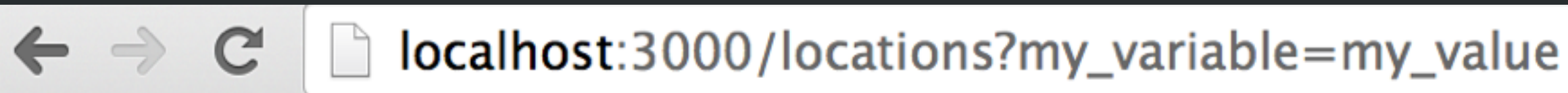
```
Rails.application.routes.draw do
  root 'home#welcome'
  get 'locations' => 'locations#index'
  get 'locations/:id' => 'locations#show'
end
```


The params object

The params object

- Every request with dynamic parameters stores them in the params object
- It's a Hash, where the key is the name of the parameter and the value is the value of the parameter

In Browser try



← → ↻  localhost:3000/locations?my_variable=my_value

Check the logs

```
Started GET "/locations?my_variable=my_value" for 10.0.2.2 at 2014-06-29 12:21:47 +0000
Processing by LocationsController#index as HTML
  Parameters: {"my_variable"=>"my_value"}
```

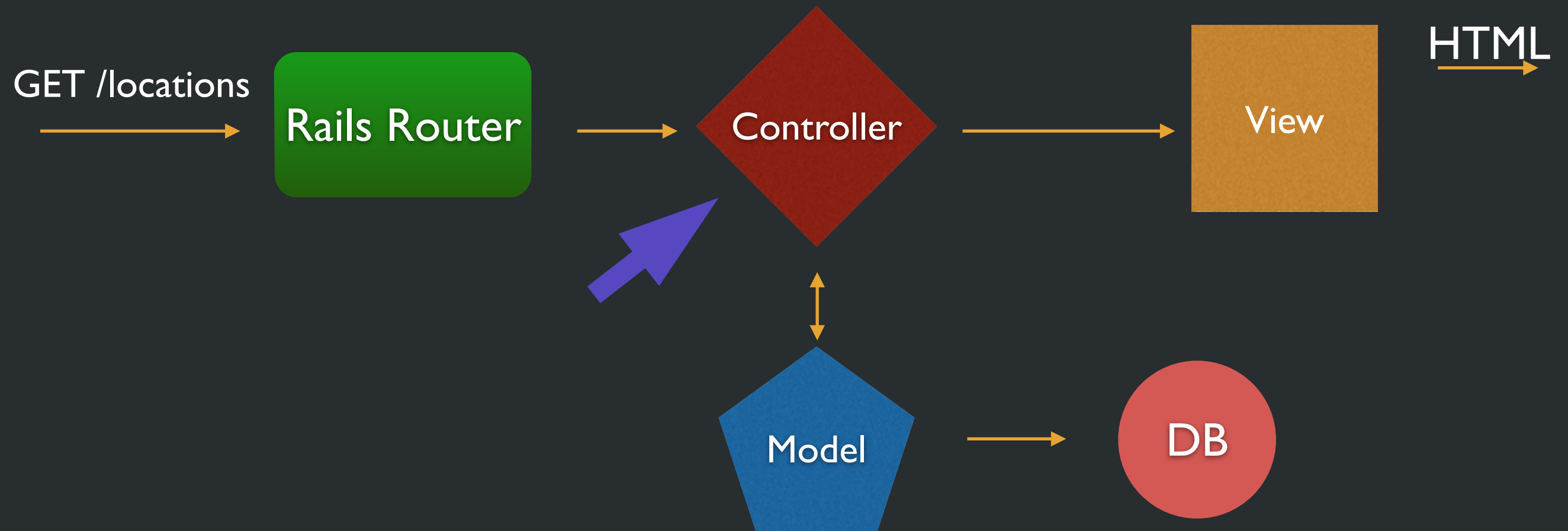
`/locations/<id>`



Parameter!

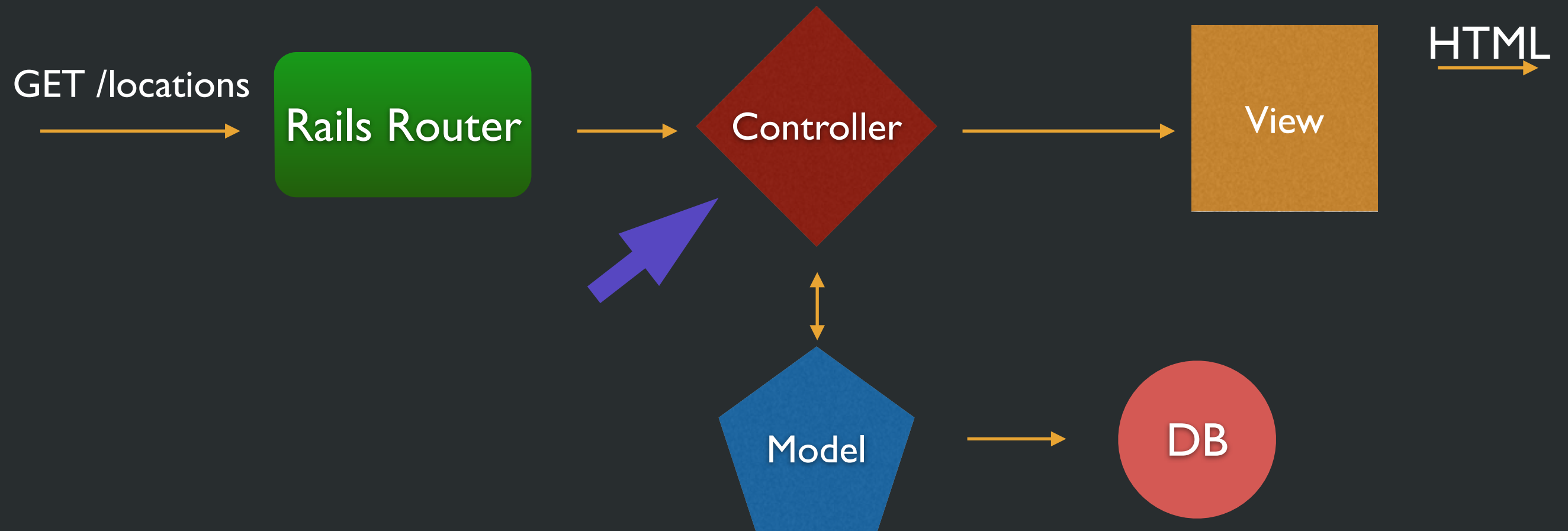
`/locations/482 → params[:id] == 482`

`/locations/34 → params[:id] == 34`



`/controllers/locations_controller.rb`

```
class LocationsController < ApplicationController
  def index
    @locations = Location.last_created(10)
  end
end
```



`/controllers/locations_controller.rb`

```
class LocationsController < ApplicationController
  def index
    @locations = Location.last_created(10)
  end

  def show
    @location = Location.find(params[:id])
  end
end
```

Exercise #17

Create a template for the `#show` action that displays the name and additional details of the location

your tasks:

- how do I test a controller? -> rspec rails docu
- how do I test which template is rendered?
- how do I test if a location is assigned?
- how do I test if the response is a success?

Solution #17

What is the name and the path of the file?

`app/views/locations/show.html.erb`

Solution #17

What is the ERB template that renders this HTML?

```
<h1>Home</h1>  
<p>City</p>
```

Solution #17

app/views/locations/show.html.erb

```
<h1><%= @location.name %></h1>
```

```
<p><%= @location.city %></p>
```

...

Run the server

```
$ rails s
```

And visit

<http://localhost:3000/locations/1>

Exercise #18

What happens when the id doesn't belong to an existing location?

ActiveRecord::RecordNotFound in LocationsController#show

Couldn't find Location with 'id'=666

Extracted source (around line #7):

```
5
6   def show
7     @location = Location.find(params[:id])
8   end
9 end
10
```

Rails.root: /vagrant/meet_me

[Application Trace](#) | [Framework Trace](#) | [Full Trace](#)

app/controllers/locations_controller.rb:7:in `show'

Request

Parameters:

```
{"id"=>"666"}
```

ActiveRecord::RecordNotFound

- It's a Ruby exception
- We need to rescue it to avoid a noise error 500 in production logs

How to rescue: #1

- Use rescue

```
def show
  @location = Location.find(params[:id])
rescue ActiveRecord::RecordNotFound
  render status: 404,
    file: "#{Rails.root}/public/404"
end
```

How to rescue: #2

- rescue for all controllers

```
class LocationsController < ApplicationController
```

```
class ApplicationController < ActionController::Base
```

```
  # Prevent CSRF attacks by raising an exception.
```

```
  # For APIs, you may want to use :null_session instead.
```

```
  protect_from_forgery with: :exception
```

```
  rescue_from ActiveRecord::RecordNotFound, with: :record_not_found
```

```
  private
```

```
    def record_not_found
```

```
      render status: 404, file: "#{Rails.root}/public/404"
```

```
    end
```

```
end
```

How to rescue: #3

- Avoid the exception

```
def show
  @location = Location.find_by(id: params[:id])
  render status: 404,
    file: "#{Rails.root}/public/404" unless @location
end
```



http status Code for not Found

What do you prefer?

COMMIT



ALL THE THINGS

memegenerator.net