

Join the Stack Overflow Community

Stack Overflow is a community of 6.9 million programmers, just like you, helping each other. Join them; it only takes a minute:

Sign up

Ruby/Rails - How to Create a Class and Access it from the Controller

I've been trying to use a few different gems for displaying googlemaps within rails 3 and have had several problems.

Luckily I found this <https://github.com/YouthTree/bhm-google-maps> which is a helper and it seems to work for others.

I've installed it properly but in the readme <https://github.com/YouthTree/bhm-google-maps/blob/master/README.md> it mentions creating a class for the object to display in the view.

The example they gave was

```
class Location
  attr_accessor :address, :lat, :lng
  def initialize(address, lat, lng)
    @address = address
    @lat = lat
    @lng = lng
  end
  def to_s; address.to_s; end
end
```

And then running

```
<%= draw_map_of Location.new("My House", 12.345, 56.789) %>
```

in the view.

It seems simple enough but I haven't experienced the need of creating a class before in rails so I have some questions.

Should I create a location.rb file and place the above code in it, but where should I place the file? (model folder, application folder????)

Is there a way for me to create this class within my controller?

Ideally I would like to manipulate the lat/lng values as variables and display a dynamic map.

[ruby-on-rails](#) [ruby](#) [class](#)

asked Mar 2 '11 at 22:33



[ChrisWesAllen](#)

1,892 13 46 77

1 Answer

You should put `location.rb` wherever you feel it makes the most sense. Having it at `app/models/location.rb` will ensure that it's automatically required when your app starts, but some people expect that classes in `app/models` are backed by ActiveRecord.

You could also put it under `lib/` if you prefer.

To make it available to the app, you can include require statement in project initializers inside your config folder:

```
require "#{RAILS.root}/lib/location.rb
```

As for creating it inside your Controller - definitely! It's just another instance of a class:


```
def show
  @location = Location.new("My House", 12.345, 56.789)
end
```

And then in your view:

```
<%= draw_map_of @location %>
```

Don't forget – beneath Rails is all the power and flexibility of pure Ruby, ready to be used. You're not only limited to what Rails gives you.


edited Oct 2 '15 at 7:49



Dan D.

38.6k66485

answered Mar 2 '11 at 22:42



Dan Cheail

8,29012639