Can Rails Routing Helpers (i.e. mymodel_path(model)) be Used in Models?

Asked 16 years ago Modified 3 months ago Viewed 191k times

403

Say I have a Rails Model called Thing. Thing has a url attribute that can **optionally** be set to a URL somewhere on the Internet. In view code, I need logic that does the following:

```
<% if thing.url.blank? %>
<%= link_to('Text', thing_path(thing)) %>
<% else %>
<%= link_to('Text', thing.url) %>
<% end %>
```

This conditional logic in the view is ugly. Of course, I could build a helper function, which would change the view to this:

```
<%= thing_link('Text', thing) %>
```

That solves the verbosity problem, but I would really prefer having the functionality in the model itself. In which case, the view code would be:

```
<%= link_to('Text', thing.link) %>
```

This, obviously, would require a link method on the model. Here's what it would need to contain:

```
def link
  (self.url.blank?) ? thing_path(self) : self.url
end
```

To the point of the question, thing_path() is an undefined method inside Model code. I'm assuming it's possible to "pull in" some helper methods into the model, but how? And is there a real reason that routing only operates at the controller and view layers of the app? I can think of lots of cases where model code may need to deal with URLs (integrating with external systems, etc).

ruby-on-rails rails-routing helpermethods

Share

Improve this question

Follow

edited Jul 24, 2009 at 20:36

Simone Carletti

176k ● 50 ● 367 ● 368

asked Dec 4, 2008 at 16:07

Aaron Longwell

8,213 ● 5 ● 23 ● 10

A use case would be: to generate shortened url from goo.gl in an aftersave, – lulalala Jun 14, 2012 at 2:55

2 You should probably wrap you model in a presenter if you want to add view logic, this will keep the MVC layers separated. See Draper(github.com/jcasimir/draper). – Kris Jun 26, 2012 at 11:49

2 See also the "URL generation for named routes" section in the documentation at api.rubyonrails.org/classes/ActionDispatch/Routing/UrlFor.html – Backo Sep 26, 2013 at 10:43

7 Answers

Sorted by: Highest score (default)

747

In Rails 3 and higher:

```
Rails.application.routes.url_helpers
```

e.g.

```
Rails.application.routes.url_helpers.posts_path
Rails.application.routes.url_helpers.posts_url(host: 'example.com')
```

Related documentation: ActionDispatch::Routing::UrlFor

Share

Improve this answer

Follow

edited Sep 18 at 14:48

Nikita Fedyashev

18.9k ● 13 ● 56 ● 103

answered Mar 28, 2011 at 7:56

Paul Horsfall

8,182 ● 1 ● 20 ● 7

15 Ypu can include this into any module and after it you can access url-helpers there. Thx – Viacheslav Molokov Apr 7, 2011 at 7:39

44 Save yourself from copying your `:host` option everywhere and set it once in your environment config files: `Rails.application.routes.default_url_options[:host] = 'localhost:3000'` – Andrew Feb 18, 2014 at 0:12

6 `include Rails.application.routes.url_helpers` works for me in Rails 4.1 – Jan Aug 19, 2014 at 13:55

1 If you just need the path you can use :only_path => true as an option without passing the host parameter. – trueinViso Oct 29, 2014 at 17:07

1 this seems to be good option: hawkins.io/2012/03/... – Renars Sirotins Mar 10, 2015 at 11:53

190

I've found the answer regarding how to do this myself. Inside the model code, just put:

For Rails <= 2:

```
include ActionController::UrlWriter
```

For Rails 3:

```
include Rails.application.routes.url_helpers
```

This magically makes `thing_path(self)` return the URL for the current thing, or `other_model_path(self.association_to_other_model)` return some other URL.

Share

Improve this answer

Follow

edited Nov 15, 2011 at 16:35

iwasrobbed

46.7k ● 21 ● 151 ● 195

answered Dec 4, 2008 at 16:42

Aaron Longwell

8,213 ● 5 ● 23 ● 10

29 Just an update, since the above is deprecated. This is the current include: `include Rails.application.routes.url_helpers` – yalestar Jun 2, 2011 at 17:46

5 I get NoMethodError (undefined method `optimize_routes_generation?' for #<ActionView::Base:0x007fe8c0eecbd0>) when I try this – moger777 Jan 23, 2015 at 15:31

134

You may also find the following approach cleaner than including every method:

```
class Thing
  delegate :url_helpers, to: 'Rails.application.routes'

  def url
    url_helpers.thing_path(self)
  end
end
```

Share Improve this answer Follow

answered Aug 24, 2011 at 0:24

matthuhiggins

1,971 ● 1 ● 12 ● 13

9 Documentation on this seemed hard to find, so here is a decent link regarding the use of delegate in ActiveSupport. simonecarletti.com/blog/2009/12/inside-ruby-on-rails-delegate – tlbrack Sep 30, 2011 at 15:41

2 I get a `undefined local variable or method 'url_helpers' for Event:Class` error... :(– Augustin Riedinger Feb 14, 2014 at 10:56

I get `undefined method url_helpers`. What I gonna do? – Alex Antonov Apr 25, 2015 at 6:28

@asiniy, are You sure You've used delegate option to url_helpers that is written after class declaration? – Krzysztof Witczak Feb 10, 2016 at 10:53

Doesn't work, but it might be that this only works if you create your own `class`, as shown in the answer. If your Model class already extends `< ApplicationRecord` this won't work? – skplunkerin Feb 7, 2019 at 23:58

15

Any logic having to do with what is displayed in the view should be delegated to a helper method, as methods in the model are strictly for handling data.

Here is what you could do:

```
# In the helper...
```

```
def link_to_thing(text, thing)
  (thing.url?) ? link_to(text, thing_path(thing)) : link_to(text, thing.url)
end

# In the view...

<%= link_to_thing("text", @thing) %>
```

Share

Improve this answer

Follow

edited Jun 17, 2014 at 3:00

Mark Swardstrom

18k ● 6 ● 71 ● 72

answered Dec 4, 2008 at 16:14

Josh Delsman

3,042 ● 1 ● 19 ● 28

1 What I don't like about helper methods in this case: When I look at link_to_thing(), I have to decide whether that's a thing-specific or application-wide helper (it easily could be either). I need to consider checking 2 files for the source. thing.link leaves no question about the source file. – Aaron Longwell Dec 4, 2008 at 16:31

1 Also, what if I need to use this functionality in a Rake task (to export a CSV file of thing URLs perhaps)... going directly to the model would be much better in that case as well. – Aaron Longwell Dec 4, 2008 at 16:32

But the difference is that the link_to wouldn't be available in the model, since it is an ActionView helper. So, this would not work. You could hack some attribute defaults in the model, so if it isn't set, it defaults to something, but that's up to you. – Josh Delsman Dec 4, 2008 at 18:17

13 With web services APIs growing, often times models need to contact external resources with their own data and provide callback URLs. For example, a photo object needs to post itself to Socialmod, which will call back to that photo's URL when moderation is performed. An after_create() hook would make sense to post to Socialmod, but how do you know the callback URL? I think the author's own answer makes good sense. – JasonSmith Sep 25, 2009 at 4:01

4 Whilst you are right in strictly technical sense, there are times when people just need things to work without having to shave every yak there is to avoid violating some sacred design pattern or what have you, maybe they need to get the url, and actually store it in the database, would be handy then for a model to just know how to do that instead of passing things back and forth, sometimes rules can be broken. – nitecoder Jun 28, 2012 at 8:24

12

I really like following clean solution.

```
class Router
  include Rails.application.routes.url_helpers

  def self.default_url_options
    ActionMailer::Base.default_url_options
  end
end

router = Router.new
```

```
router.posts_url  # http://localhost:3000/posts
router.posts_path # /posts
```

It's from
https://web.archive.org/web/20200221203515/https://www.hawkins.io/2012/03/generating_urls_whenever_and_wherever_you_want/

Share
Improve this answer
Follow

edited Sep 11, 2023 at 3:20

answered Feb 5, 2019 at 20:59

Swar Shah
173 ● 3 ● 9

1 Link is broken FYI – fatfrog Sep 10, 2023 at 2:03

1 Updated to link from archive. – Swar Shah Sep 11, 2023 at 3:20

Absolutely great, but is it possible to get hostname not from mailer but the way Rails itself does? – sekrett Mar 7 at 15:23

1

While there might be a way I would tend to keep that kind of logic out of the Model. I agree that you shouldn't put that in the view (keep it skinny) but unless the model is returning a url as a piece of data to the controller, the routing stuff should be in the controller.

Share Improve this answer Follow

answered Dec 4, 2008 at 16:15

rmontgomery429
14.9k ● 17 ● 62 ● 68

4 Re: "Unless the model is returning a URL as a piece of data". That's exactly what is happening here... the Model "owns" this piece of data, which is either a link to an on-site or off-site URL. In some cases the URL is generated from Rails Routing. In other cases, the URL is user-supplied data. – Aaron Longwell Dec 4, 2008 at 16:19

0

(Edit: Forget my previous babble...)

Ok, there might be situations where you would go either to the model or to some other url... But I don't really think this belongs in the model, the view (or maybe the model) sounds more apropriate.

About the routes, as far as I know the routes is for the actions in controllers (wich usually "magically" uses a view), not directly to views. The controller should handle all requests, the view should present the results and the model should handle the data and serve it to the view or controller. I've heard a lot of people here talking about routes to models (to the point I'm allmost starting to beleave it), but as I understand it:

routes goes to controllers. Of course a lot of controllers are controllers for one model and is often called `<modelname>sController` (e.g. "UsersController" is the controller of the model "User").

If you find yourself writing nasty amounts of logic in a view, try to move the logic somewhere more appropriate; request and internal communication logic probably belongs in the controller, data related logic may be placed in the model (but not display logic, which includes link tags etc.) and logic that is purely display related would be placed in a helper.

Share

Improve this answer

Follow

edited Dec 4, 2008 at 16:47

answered Dec 4, 2008 at 16:15

Stein G. Strindhaug

5,119 ● 2 ● 30 ● 42

Say you have an Image model. If the Image has an associated outside URL with it, then point to that URL. Otherwise, point to the Image's show page to upload an image? Just an idea. – Josh Delsman Dec 4, 2008 at 16:17

How about this less generic example: link_to "Full Size Image", image.link The link method in the model would either link to the on-site URL (image_path), or to, say, a Flickr URL if the user provided one and .url was set on the image. – Aaron Longwell Dec 4, 2008 at 16:21