

Widgetfinger: Simple content management for simple websites.

Come “ride the toad” on **Hoptoad**, the app error app.

Tee-Bot, funny shirts your friends won't understand!

Umbrella Today: “It’s like totally the simplest weather report ever, Julie.”

thoughtbot is a technology consulting firm that provides web application development and design services. We focus on building modern systems, embracing good ideas and delivering elegant solutions.

Interested in learning Rails?
Sign up for our training.

Development	186
Business	57
Design	23
Technology	58

APR
04

RESTful contact forms

Posted by Matt Jankowski

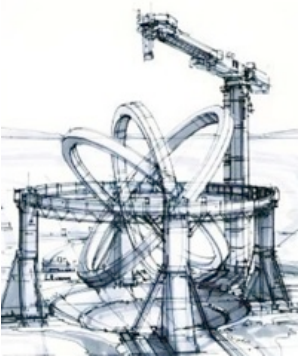
Just like [static content pages](#) – simple “contact form” pages are another thing that can very easily fall into the RESTful pattern if you want them to. And, in the same way that seeing implicit actions in static page controllers keeps me up at night, so does seeing something like this in a controller...

```
1
2 class SiteController < ApplicationController
3
4   def contact
5     if request.post?
6       # Send some mail
7       # Set a flash, and redirect back somewhere
8     else
9       # Render a contact form
10    end
11  end
12
13 end
```

There are least three things wrong with this small code fragment:

- The name `SiteController` – in this case, does not correspond to a “site resource”, and there’s no `Site` model in the app. I feel like a controller name should meet at least one if not both of those requirements, and this controller doesn’t. It’s just being used as a bucket to drop stuff that doesn’t go anywhere else.
- The use of `contact` for action naming. “This action does a ‘contact’ for the site” only marginally makes sense.
- The use of “`if request.post?`” as the outermost code in an action. Come on. This style can almost always be refactored into two actions.

So how do we want to change this? Well, what’s the resource here? I’m going to say the resource is “the contact” or “a contacting event” maybe. I suppose you could go with “`Message`” as well, or some other naming that reflects the passing of information from a site user back to the team



running the site. That deals with the first problem.

We can deal with the action naming problem and the “if request.post?” problem at the same time. In terms of action naming, I think the form we show the user to submit the information should be “new” (although I’ve entertained fascinating arguments for why it should be “show!”), and the action which actually “creates the contact” (ie, sends an email back to the team, or logs the info in the db maybe?) should be “create”. Both are standard RESTful named actions, so that requirement is met.

We end up with a controller like this...

```
1
2 class ContactController < ApplicationController
3
4   before_filter :check_message, :only => :create
5
6   def new
7   end
8
9   def create
10    Notifier.deliver_contact_form params[:name], params[:email], params[:message], params[:refe
11    flash[:success] = 'Feedback sent successfully'
12    redirect_to root_url
13  end
14
15  protected
16
17  def check_message
18    if params[:message].blank?
19      flash.now[:failure] = 'Please supply a message'
20      render :action => :new and return false
21    end
22  end
23
24 end
```

(Note the correct [flash.now](#) usage in the before_filter. Again, without that in place, I’d lose sleep.)

...and a new route, like this...

```
1
2 map.resource :contact, :controller => 'contact'
```

(Note the correct [singleton resource](#) usage. If I didn’t do that I’d once again lose more sleep and this refactoring wouldn’t be even be worth it anymore because I’d be up all night.)

I like that much better – and as a side effect, I’m going to be able to write separate functional tests for #new and #create, which should clean up the tests a bit as well.

Does anyone else do “RESTful contact forming”? Any better ideas for this simple problem?

Comments on this post



[Dan Croak](#) said,

That request.post? makes me feel dirty.

Apr 04