

How to contribute to Ruby on Rails

Nicolas Cavigneaux



About me

- Software Engineer
- Fond of Ruby since 2003
- Using Rails since 2005
- Works professionally with Rails since 2006
- Blogs on <http://www.bounga.org>



Why contributing?

- Rails is an Open Source Software
- Solve bugs that bother you and other users
- Add new features and share it
- Give back to the community
- Better understanding of how Rails really works



What to contribute?

What kind of contribution?

- code
- tests
- documentation

What for?

- bug fixes
- enhancements
- refactoring
- open tickets (write a failing test then write a fix)



Where to go?

- **Tracker:** <http://rails.lighthouseapp.com/projects/8994-ruby-on-rails/tickets>
- **Source code:** <http://github.com/rails/rails/tree/master>
- **Discussions:**
 - rails core: rubyonrails-core@googlegroups.com
 - contribution: #rails-contrib on irc.freenode.net



How to contribute?

- Set up a Rails development environment
- Code your modification
- Generate a patch file
- Create a ticket
- Wait for replies



Set up development environment

- Create a new project for working on Rails core
 - `$ rails contribute`
- Retrieve Edge Rails using Subversion
 - `$ cd contribute/vendor`
 - `$ git clone git://github.com/rails/rails.git`
- Setup
 - Read `RUNNING_UNIT_TEST` file for module you want to work on
- Run tests before any changes to ensure that all is ok



Implement your changes

- Start with the tests
- Follow the style guidelines
- Document your code
- Make sure all tests pass and docs build



Generate a patch

- Use Git diff command
 - `$ cd contribute/vendor/rails`
 - `$ git format-patch master --stdout > well_named_patch.diff`
- Examine diff file content



Size matters

- Small changes are easier to get accepted
 - Use “tiny” keyword for really small ones
- Got a big patch?
 - Discuss change on rails-core list first
 - Try out patch as plugin - makes it easy to check stability of change



Submit your patch

- Register on Rails Trac
- Create a ticket (<http://rails.lighthouseapp.com/projects/8994-ruby-on-rails/tickets/new>)
- Tag it with patch
- Upload your diff file
- Describe problem and solution, show usage
- Use good keywords : doc, tiny, test



Example

My first patch

The Setup

```
link_to(:action => :list)
```

```
# => /list
```

```
link_to(:action => :list, :host => "www.bounga.org")
```

```
# => /list
```

```
link_to(:action => :list, :host => "www.bounga.org", :only_path  
=> false)
```

```
# => http://www.bounga.org/list
```



What I wanted

```
link_to(:action => :list)
```

```
# => /list
```

```
link_to(:action => :list, :host => "www.bounga.org")
```

```
# => http://www.bounga.org/list
```

Setting the :host option in url_for should automatically turn off :only_path (since :host would otherwise not be shown)



The whole patch

Index: actionpack/lib/action_view/helpers/url_helper.rb

```
=====
--- actionpack/lib/action_view/helpers/url_helper.rb      (révision 7513)
+++ actionpack/lib/action_view/helpers/url_helper.rb      (copie de travail)
@@ -20,7 +20,7 @@
  #
  # ==== Options
  # * <tt>:anchor</tt> -- specifies the anchor name to be appended to
the path.
-   # * <tt>:only_path</tt> -- if true, returns the relative URL
(omitting the protocol, host name, and port) (<tt>true</tt> by default)
+   # * <tt>:only_path</tt> -- if true, returns the relative URL
(omitting the protocol, host name, and port) (<tt>true</tt> by default
unless <tt>:host</tt> is specified)
  # * <tt>:trailing_slash</tt> -- if true, adds a trailing slash, as
in "/archive/2005/". Note that this
  #   is currently not recommended since it breaks caching.
  # * <tt>:host</tt> -- overrides the default (current) host if
provided
@@ -65,7 +65,8 @@
  def url_for(options = {})
    case options
    when Hash
-      options = { :only_path => true }.update(options.symbolize_keys)
+      show_path = options[:host].nil? ? true : false
+      options = { :only_path => show_path }.update
(options.symbolize_keys)
      escape = options.key?(:escape) ? options.delete(:escape) : true
      url = @controller.send(:url_for, options)
    when String
```

Index: actionpack/test/template/url_helper_test.rb

```
=====
--- actionpack/test/template/url_helper_test.rb (révision 7513)
+++ actionpack/test/template/url_helper_test.rb (copie de travail)
@@ -82,7 +82,25 @@
  def test_link_tag_with_straight_url
    assert_dom_equal "<a href=\"http://www.example.com\">Hello</a>", link_to
("Hello", "http://www.example.com")
  end
+
+  def test_link_tag_without_host_option
+    ActionController::Base.class_eval { attr_accessor :url }
+    url = { :controller => 'weblog', :action => 'show' }
+    @controller = ActionController::Base.new
+    @controller.request = ActionController::TestRequest.new
+    @controller.url = ActionController::UrlRewriter.new(@controller.request,
url)
+    assert_dom_equal(%q{<a href="/weblog/show">Test Link</a>}, link_to('Test
Link', url))
+  end
+
+  def test_link_tag_with_host_option
+    ActionController::Base.class_eval { attr_accessor :url }
+    url = { :controller => 'weblog', :action => 'show', :host =>
'www.example.com' }
+    @controller = ActionController::Base.new
+    @controller.request = ActionController::TestRequest.new
+    @controller.url = ActionController::UrlRewriter.new(@controller.request,
url)
+    assert_dom_equal(%q{<a href="http://www.example.com/weblog/show">Test
Link</a>}, link_to('Test Link', url))
+  end
+
  def test_link_tag_with_query
    assert_dom_equal "<a href=\"http://www.example.com?q1=v1&q2=v2\">Hello</a>", link_to("Hello", "http://www.example.com?q1=v1&q2=v2")
  end
```



And now what?

- Got an email after ticket creation?
- Respond to ticket updates
- Wait for feedbacks
- Discuss about your patch on rubyonrails-core@googlegroups.com and [#rails-contrib](https://www.rubyonrails.org/en/contributing-to-rubyonrails/#contributing-to-the-discussion) on [irc.freenode.net](https://www.rubyonrails.org/en/contributing-to-rubyonrails/#contributing-to-the-discussion)
- Blog about it to popularize your change



Licence

<http://www.bounga.org>



Attribution-NonCommercial-ShareAlike 2.5

