benprew / **pony**
forked from adamwiggins/pony

The express way to send mail from Ruby.

152 commits | **1** branch | **15** releases | **25** contributors

Branch: **master** ▾ | New pull request | Find file | Clone or download ▾

This branch is 122 commits ahead, 2 commits behind adamwiggins:master. Pull request Compare

benprew Update pony gem to 1.12 Latest commit f2bd337 on 13 Dec 2017

lib	Move Pony method calls outside of mail block	a year ago
spec	Make private class methods actually private	a year ago
.gitignore	Basic .gitignore for OS X/Vim.	10 years ago
.travis.yml	Remove jruby target for now, so tests can pass	a year ago
Gemfile	add bundler, move from tabs to soft tabs (2 spaces) and add reply_to …	8 years ago
Gemfile.lock	Update pony gem to 1.12	a year ago
README.md	Fix `NoMethodError` when pony is used with mail 2.7.0.	a year ago
Rakefile	update authors	3 years ago
pony.gemspec	Update pony gem to 1.12	a year ago

README.md

Pony, the express way to send email in Ruby

Overview

Ruby no longer has to be jealous of PHP's mail() function, which can send an email in a single command.

```
Pony.mail(:to => 'you@example.com', :from => 'me@example.com', :subject => 'hi', :body => 'Hello there.')
Pony.mail(:to => 'you@example.com', :html_body => '<h1>Hello there!</h1>', :body => "In case you can't read html,
Hello there.")
Pony.mail(:to => 'you@example.com', :cc => 'him@example.com', :from => 'me@example.com', :subject => 'hi', :body =>
'Howsit!')
```

Any option key may be omitted except for `:to` . See List Of Options section for the complete list.

Transport

Pony uses `/usr/sbin/sendmail` to send mail if it is available, otherwise it uses SMTP to localhost.

This can be overridden if you specify a via option:

```
Pony.mail(:to => 'you@example.com', :via => :smtp) # sends via SMTP
Pony.mail(:to => 'you@example.com', :via => :sendmail) # sends via sendmail
```

You can also specify options for SMTP:

```
Pony.mail({
  :to => 'you@example.com',
  :via => :smtp,
  :via_options => {
    :address        => 'smtp.yourserver.com',
    :port           => '25',
    :user_name      => 'user',
    :password       => 'password',
    :authentication => :plain, # :plain, :login, :cram_md5, no auth by default
    :domain         => "localhost.localdomain" # the HELO domain provided by the client to the server
  }
})
```

Gmail example (with TLS/SSL)

```
Pony.mail({
  :to => 'you@example.com',
  :via => :smtp,
  :via_options => {
    :address              => 'smtp.gmail.com',
    :port                 => '587',
    :enable_starttls_auto => true,
    :user_name            => 'user',
    :password             => 'password_see_note',
    :authentication       => :plain, # :plain, :login, :cram_md5, no auth by default
    :domain               => "localhost.localdomain" # the HELO domain provided by the client to the server
  }
})
```

Note: If you use 2 step verification, you will have to generate an application specific password and NOT use your normal password - see https://support.google.com/accounts/answer/185833?hl=en

And options for Sendmail:

```
Pony.mail({
  :to => 'you@example.com',
  :via => :sendmail,
  :via_options => {
    :location  => '/path/to/sendmail', # defaults to 'which sendmail' or '/usr/sbin/sendmail' if 'which' fails
    :arguments => '-t' # -t and -i are the defaults
  }
})
```

If you're using smtp, set `:arguments => ''` .

Attachments

You can attach a file or two with the :attachments option:

```
Pony.mail(..., :attachments => {"foo.zip" => File.read("path/to/foo.zip"), "hello.txt" => "hello!"})
```

Note: An attachment's mime-type is set based on the filename (as dictated by the ruby gem mime-types). So 'foo.pdf' has a mime-type of 'application/pdf'

Custom Headers

Pony allows you to specify custom mail headers

```
Pony.mail(
  :to => 'me@example.com',
  :headers => { "List-ID" => "...", "X-My-Custom-Header" => "what a cool custom header" }
)
```

Add additional options for headers in each part of letter (text, html)

```
Pony.mail(
  :body => 'test',
  :html_body => 'What do you know, Joe?',
  :attachments => {"foo.txt" => "content of foo.txt"},
  :body_part_header => { content_disposition: "inline" }
)
```

This will add option `'Content-Disposition: inline'` for text part header of letter.

Also you can add additional options for html part of latter, e.g.:

```
:html_body_part_header => { content_disposition: "inline" }
```

List Of Options

You can get a list of options from Pony directly:

```
Pony.permissable_options
```

Options passed pretty much directly to Mail

```
attachments # see Attachments section
bcc
body # the plain text body
body_part_header # see Custom headers section
cc
charset # In case you need to send in utf-8 or similar
content_type
from
headers # see Custom headers section
html_body # for sending html-formatted email
html_body_part_header # see Custom headers section
message_id
reply_to
sender  # Sets "envelope from" (and the Sender header)
smtp_envelope_to
subject
text_part_charset # for multipart messages, set the charset of the text part
to
```

Other options via # :smtp or :sendmail, see Transport section via_options # specify transport options, see Transport section

Default Options

Default options can be set so that they don't have to be repeated. The default options you set will be overriden by any options you pass in to Pony.mail()

```
Pony.options = { :from => 'noreply@example.com', :via => :smtp, :via_options => { :host => 'smtp.yourserver.com' }
}
Pony.mail(:to => 'foo@bar') # Sends mail to foo@bar from noreply@example.com using smtp
Pony.mail(:from => 'pony@example.com', :to => 'foo@bar') # Sends mail to foo@bar from pony@example.com using smtp
```

Override Options

Override options can be set so that the override value is always be used, even if a different value is passed in to Pony.options() or Pony.mail(). This can be used to configure a development or staging environment.

```
Pony.override_options = { :to => 'test@example.com' }
Pony.mail(:to => 'foo@bar') # Sends mail to test@example.com instead of foo@bar
```

Testing/Debugging Aids

build passing

Subject prefix

Prepends a string to the subject line. This is used to identify email sent from a specific environment.

```
Pony.subject_prefix('Prefix:')
```

Append options to body

Append the options passd into Pony.mail to the body of the email. Useful for debugging.

```
Pony.append_inputs
```

Using Pony with Testing or Spec'ing Libraries

As pony relies on mail to send the mails, you can also use its TestMailer in your tests.

```
Pony.override_options = { :via => :test }
Pony.mail(:to => 'foo@bar')
Mail::TestMailer.deliveries.length
=> 1
```

For further examples see the corresponding section of mail's readme

Help

If you need help using Pony, or it looks like you've found a bug, email ponyrb@googlegroups.com. The full forum is https://groups.google.com/forum/#!forum/ponyrb

Meta

- Released under the MIT License
- Homepage
- Mailing List: ponyrb@googlegroups.com

Authors

- Adam Wiggins @adamwiggins
- Andrea Talbot @janehax
- Ben Prew @benprew
- Cameron Matheson @cmatheson
- Carl Hörberg @carlhoerberg
- Christian Haase @krissi
- Daniel Lopes @danielvlopes
- Doug Hammond @dslh
- Hiroshi Saito @hiroshi
- Kalin Harvey
- MIKAMI Yoshiyuki @yoshuki
- Mathieu Martin @webmat
- Michael Durrant @durrantm
- Michał Kwiatkowski @mkwiatkowski
- Nami-Doc
- Neil Middleton @neilmiddleton
- Neil Mock @neilmock
- Nickolas Means @nmeans
- Othmane Benkirane @anaothmane
- Rich Meyers @richmeyers
- Roman Franko @roman-franko
- Ryan Malecky @rmalecky

- Seamus Abshere @seamusabshere
- Stephen Celis @stephencelis
- Steve Root @steveroot
- arunthampi @arunthampi
- rick @rick
- rohit @rohit

Changelog

1.12

- fix bug: NoMethodError when using mail 2.7.0

1.11

- Improved handling of mails with both text and html bodies and attachments

1.10

- Add `subject_prefix` , `append_options` and `override_options`

1.9

- Allow options to be queried from the gem

1.8

- Add additional options for headers in each part of letter

1.7

- Better default content_type with attachments

1.6

- Unknown options are passed directly to mail to handle. Remove deprecated syntax

1.5.1

- Loosen mail dependency to >= 2.0 instead of > 2.0

1.5

- Specify content-id of attachments as filename@hostname

1.4.1

- Update gemfile

1.4

- Updated docs

1.3

- Add new option `:text_part_charset` , which allows you to specify the charset for the text portion

1.2

- Remove limitations on :via, and let mail handle it (this means you can say things like :via => test)
- Add reply-to option and a bundler file

1.1

- Add default options

1.0

- Convert to using Mail as the mail-generation backend, instead of TMail

0.9.1

- provide the ability to set custom mail headers with something like: Pony.mail(:headers => {"List-ID" => "..."})
- provide the ability to set the Message-Id from Pony.mail

0.9

- merge in kalin's fixes to use tmail.destinations instead of trying to parse tmail.to, tmail.cc and tmail.bcc. New specs to test functionality

0.8

- Fix bug that was allowing nil :bcc and :cc options to be passed to smtp

0.7

- Pass :cc and :bcc options to sendmail appropriately

0.6

- Add :bcc capability
- Add :charset capability
- Add complete list of options to readme
- fix bug: readme examples

0.5

- default location of sendmail to /usr/sbin/sendmail if sendmail not in path
- fix bug: README not showing password option (listed as pass)

0.4.1

- Add :cc capability
- fix bug: resolve body not displaying when attachments sent

0.4

- Implemented file attachments option
- use TLS if :tls => true