

Sponsored By



Microsoft

AUTOMATTIC



pluralsight **JetBRAINS**



Braintree_PayPal



Swift Mailer

Configuring, Testing and Using with Symfony

What is Swift Mailer?

A library for sending emails from your PHP application

Open source

Default for Symfony since 1.3 (2009)

Chris Corbyn started in 2005

SensioLabs project since 2009



Swift Mailer is not...

SMTP server software, like...

Postfix

Sendmail

qmail

An app with a GUI, like...

Thunderbird

Retail email marketing platform, like...

Constant Contact

Listrak

Why not just use PHP's native `mail()` ?

Headers correctly configured

- prevent false positives for spam
- all recipients get properly formed emails

Efficiency

- `mail()` opens SMTP socket for each message

Security

- sanitize input
- prevent spoofing

Reliability

- some email providers reject mail sent via `mail()`

Added functionality

- support for attachments, HTML-formatted mail

Use cases

Send emails from your application

Contact forms

Automatic notifications

Use in conjunction with an SMTP relay service, like...

Mandrill

SendGrid

Amazon SES

Google Apps SMTP relay



MANDRILL



Setting up

Included in Symfony

In `app/AppKernel.php`

```
new Symfony\Bundle\SwiftmailerBundle\SwiftmailerBundle(),
```


Check for these lines

In `composer.json`

```
"require": {  
    .  
    .  
    "swiftmailer/swiftmailer": "@stable"  
    "symfony/swiftmailer-bundle": "~2.3"  
},
```

Ensure dependencies

Check for `sendmail`

```
sendmail -v
```

PHP `proc_*` program execution functions

```
<?php echo function_exists('proc_open') ? "Yep, that  
will work" : "Sorry, that won't work";
```

Configure test environment!

Add to: `application/config/config_test.yml`

```
swiftmailer:  
  spool:  
    type:memory  
  path:/tmp
```

Test

Unit testing

```
$this->mailer = $this->getMock('Swift_Mailer', ['send'], [],  
'', false, false);
```

```
$this->fixture = new MyClassThatSendsMail($this->mailer);
```

```
public function testMailerIsCalledWithMessage()
```

```
{
```

```
    $message = uniqid();
```

```
    $this->mailer->expects($this->once())
```

```
        ->method('send')
```

```
        ->with($message);
```

```
    $this->fixture->process($message);
```

```
}
```

Unit testing

Sample values need to be valid for their MIME type, or else your tests won't pass.

For example, note `$to` value:

```
public function testMessageHasTo()  
{  
    $to = uniqid('abc');  
  
    $actual = $this->fixture->create($to);  
  
    $this->assertEquals($to, $actual->getTo());  
}
```

There was 1 error:

1) AppBundle\Tests\Mail\MessageFactoryTest::testMessageHasToSwift_RfcComplianceException: Address in mailbox given [abc563e7baef0d35] does not comply with RFC 2822, 3.6.2.

/vagrant/vendor/swiftmailer/swiftmailer/lib/classes/Swift/Mime/Header/MailboxHeader.php:348

/vagrant/vendor/swiftmailer/swiftmailer/lib/classes/Swift/Mime/Header/MailboxHeader.php:263

/vagrant/vendor/swiftmailer/swiftmailer/lib/classes/Swift/Mime/Header/MailboxHeader.php:106

/vagrant/vendor/swiftmailer/swiftmailer/lib/classes/Swift/Mime/Header/MailboxHeader.php:63

/vagrant/vendor/swiftmailer/swiftmailer/lib/classes/Swift/Mime/SimpleHeaderFactory.php:58

/vagrant/vendor/swiftmailer/swiftmailer/lib/classes/Swift/Mime/SimpleHeaderSet.php:68

/vagrant/vendor/swiftmailer/swiftmailer/lib/classes/Swift/Mime/SimpleMessage.php:323

/vagrant/src/AppBundle/Mail/MessageFactory.php:12

/vagrant/src/AppBundle/Tests/Mail/MessageFactoryTest.php:33

FAILURES!

Unit testing

```
$this->mailer = $this->getMock('Swift_Mailer', [], []  
'', false, false);
```

```
$this->fixture = new MyClassThatSendsMail($this->mailer);
```

```
public function testMailNotSentWhenShouldNotBe()
```

```
{
```

```
    $this->mailer->expects($this-> never())
```

```
        ->method('send');
```

```
    $this->assertFalse($this->fixture->process());
```

```
}
```


Use in your application

```
<service class="MyApp\SomeBundle\MyClassThatSendsMail">  
    <argument type="service" id="mailer" />  
</service>
```

```
Class MyClassThatSendsMail()  
{  
    public function __construct(\Swift_Mailer $mailer)  
    { $this->mailer = $mailer; }  
}
```

Use in your application

```
protected function sendEmail()  
{  
    $message = \Swift_Mailer::newInstance()  
        ->setSubject('Nifty subject')  
        ->setFrom('jones@example.com')  
        ->setTo('gonzales@example.com')  
        ->setBody('Is it not a lovely day?');  
    $this->mailer->send($message);  
}
```

Front-end testing

Best way: install MailCatcher

“Catches mail and serves it through a dream.”

– MailCatcher website



MailCatcher

Other way: dump to /tmp and pick through output for your message.

Configure development environment

In `application/config/config_dev.yml`

Add swiftmailer config:

```
swiftmailer:
  spool:
    type:file
    path:/tmp
```

Front-end testing: config for MailCatcher

In `application/config/config_dev.yml`

```
swiftmailer:
```

```
  transport: smtp
```

```
  host: localhost
```

```
  port: 1025
```

```
  sender_address: apache@example.com
```

```
  logging: %kernel.debug%
```

```
<!-- swiftmailer:
      pool:
        type: file
        path: /tmp -->
```

Front-end testing: MailCatcher setup

Install sqlite libraries:

```
$ sudo apt-get install libsqlite3-dev
```

Install MailCatcher without ruby docs or `ri` (API info at command line)

```
$ sudo gem install --no-rdoc --no-ri mailcatcher
```

Front-end testing: using MailCatcher

Run Mailcatcher:

```
$ mailcatcher
```

If you're using a virtual machine, and a web browser on the host, run this to listen to any machine that can access your machine on port 1025:

```
$ mailcatcher --http-ip 0.0.0.0
```

or

```
$ ip addr
```

```
$ mailcatcher --http-ip 192.168.56.0 //use whatever IP you find
```

Front-end testing: using MailCatcher

If you're using a virtual machine, and a web browser on the host:

```
$ ip addr
```

```
$ mailcatcher --http-ip 192.168.56.0 //use whatever IP you find
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 08:00:27:f5:44:4d brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:fef5:444d/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 08:00:27:38:8c:f3 brd ff:ff:ff:ff:ff:ff
   inet 192.168.56.10/24 brd 192.168.56.255 scope global eth1
       valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:fe38:8cf3/64 scope link
       valid_lft forever preferred_lft forever
```


MailCatcher troubleshooting

```
~~> ERROR: Something's using port 1025. Are you already  
running MailCatcher?
```

Try this to

- List open listening tcp ports, without resolving ip names / reverse lookups
- include which process ID and program name is using the port.

```
sudo netstat -ltvnp
```

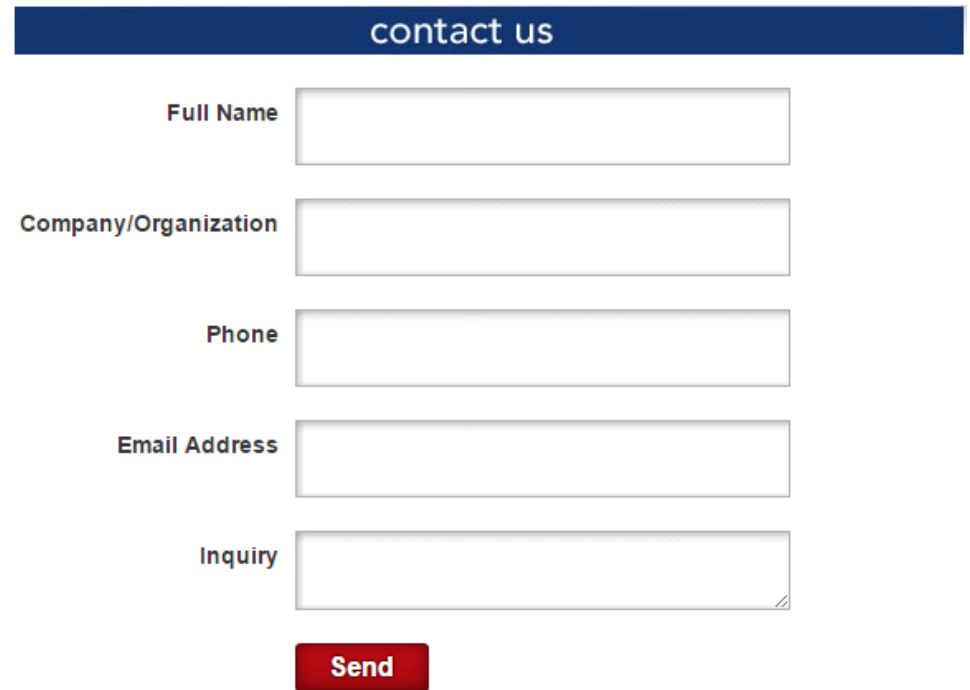
Front-end testing: using MailCatcher

On your website, send the email.

In your web browser, go to
(for example):

`192.168.56.10:1080`

You'll see the email you sent.



contact us

Full Name

Company/Organization

Phone

Email Address

Inquiry

Send

From	To	Subject	Received
------	----	---------	----------

Received
From
To
Subject

```
Return-Path: <cron@officesupply.com>  
Message-ID: <fef62597eda34092f9e5becf83b3383d@vagrant-ubuntu-trusty-64>  
Date: Wed, 21 Oct 2015 14:56:19 -0500  
Subject: GSA Contact Information Submitted  
From: General Service Administration <gsa@officesupply.com>  
To: J[REDACTED] <j[REDACTED]@officesupply.com>  
MIME-Version: 1.0  
Content-Type: multipart/alternative;  
  boundary="=_swift_v4_1445457379_43b29e1f29e8a7b6c085b4d202fbabd6_=_"  
  
--=_swift_v4_1445457379_43b29e1f29e8a7b6c085b4d202fbabd6_=_  
Content-Type: text/html; charset=utf-8  
Content-Transfer-Encoding: quoted-printable
```

Production

Production settings: Gmail

These will be specific to your SMTP relay service

```
swiftmailer:
```

```
    transport:      smtp
    host:           smtp-relay.gmail.com
    port:           25
    encryption:     tls
    sender_address: apache@example.com
    logging:        %kernel.debug%
```

Production settings: Mandrill

These will be specific to your SMTP relay service

```
swiftmailer:
```

```
    transport:      smtp
    host:           smtp.mandrillapp.com
    port:           587
    encryption:
    sender_address: admin@example.com
    logging:        %kernel.debug%
```

Production Settings

...

```
spool:
```

```
  type: file
```

```
  path: %kernel.cache_dir%/swiftmailer/spool
```

```
antiflood:
```

```
  sleep:      20
```

```
  threshold: 20
```

More ideas

Abstracting

```
public function __construct(  
    Swift_Mailer $swift,  
    Mapper $mapper  
) { $this->swift = $swift; $this->mapper = $mapper; }
```

```
public function send(Message $message)  
{  
    $swiftMessage = $this->mapper  
        ->createSwiftMessage($message)  
  
    $this->swift->send($swiftMessage);  
}
```

Ideas for mapping

```
protected function mapBasicFields(Swift_Mime_Message
$swift, Message $message)
{
    $swift->setFrom($message->getFrom());
    $swift->setTo($message->getTo());
    $swift->setCc($message->getCc());
    $swift->setBcc($message->getBcc());
    $swift->setReplyTo($message->getReplyTo());
    $swift->setSubject($message->getSubject());
}
```

Map a text message

```
protected function mapTextMessage(  
    Swift_Mime_Message $swift,  
    Message $message)  
{  
    $swift->setBody($message->getText(), 'text/plain');  
}
```

Send HTML mail

```
protected function mapHtmlMessage(  
    Swift_Mime_Message $swift,  
    Message $message)  
{  
    $html = $message->getHtml();  
    if (!empty($html)) {  
        $swift->addPart($html, 'text/html');  
    }  
}
```

Map attachments

```
Swift_Mime_Message $swift = // ...
```

```
    $swift->attach(  
        Swift_Attachment::newInstance(  
            $attachment->getContentBlob(),  
            $attachment->getName(),  
            $attachment->getContentType()  
        )    );
```

References and Resources

Swift Mailer home

<http://swiftmailer.org>

SwiftMailerBundle Configuration

<https://symfony.com/doc/current/reference/configuration/swiftmailer.html>

MailCatcher

<http://mailcatcher.me/>

Official Swift Mailer group: use [StackOverflow.com](https://stackoverflow.com) as of July 2015

Old Google Group archive: “Swift Mailer (DISABLED)”

<https://groups.google.com/forum/#!forum/swiftmailer>

Gmail SMTP configuration

<http://www.serversmtp.com/en/smtp-gmail-configuration>

How to Create a Contact Form Using Symfony

<http://www.lucas.courot.com/how-to-create-a-contact-form-using-symfony2.html>

More References and Resources

Setting-up Swift Mailer to send through Google Apps SMTP servers

<http://nuclearprojects.com/blog/setting-up-swift-mailer-to-send-through-google-apps-smtp-servers/>

PHP package of the week: SwiftMailer (shows Mandrill setup)

<https://www.youtube.com/watch?v=UEubfhMast8>

Sending email using swiftmailer symfony2 (In French, but understandable for non-Francophones)

<https://www.youtube.com/watch?v=DqUZHQn--y0>

PHP's mail() (for comparison)

<http://php.net/manual/en/function.mail.php>

Swift Mailer: The Missing Manual by Norio Suzuki (presentation deck)

<https://speakerdeck.com/suzuki/swift-mailer-the-missing-manual-and-more>

Sending email with Swift Mailer: Sitepoint (not Symfony-specific):

<http://www.sitepoint.com/sending-email-with-swift-mailer/>

The Hostile Email Landscape, by Jody Ribton

<http://liminality.xyz/the-hostile-email-landscape/>

About me



Please rate this talk!

<https://joind.in/talk/view/14776>

Vesna Vuynovich Kovach runs DevDivas.com, a website celebrating the rich history of women in technology.

She is part of the agile PHP development team that powers OfficeSupply.com, an ecommerce retail site headquartered in Columbus, Wisconsin.

She tweets at [@Vesna V K](https://twitter.com/Vesna_V_K).