
Stampie Documentation

Release 1.0.0-dev

Henrik Bjrnkov

July 25, 2014

1	Documentation	3
1.1	Setup	3
1.2	Sending Messages	3
1.3	Running the Examples	4
2	Carriers	5
2.1	Postmark	5
2.2	Mandrill	5
2.3	MailGun	5

Stampie is a small library that makes it easy to send emails through online services. It does this by implementing specific handlers for each of them.

Documentation

1.1 Setup

Stampie is made up of a couple of different things. The first is a “Carrier” the carrier is responsible for creating and formatting a Message into a Request. The Request is then given to an Adapter which responsibility is to actually do the call to the service. These are all bound together by the Mailer.

In order to setup you need a Mailer, an Adapter and a Carrier. The following uses Buzz as an example but Stampie comes with an Adapter for Guzzle aswell.

```
<?php

use Buzz\Browser;
use Stampie\Adapter\BuzzAdapter;
use Stampie\Carrier\PostmarkCarrier;
use Stampie\DirectMailer;
use Symfony\Component\EventDispatcher\EventDispatcher;

$mailer = new DirectMailer(new PostmarkCarrier('my-api-key'), new BuzzAdapter(new Browser), new EventDispatcher);
```

The DirectMailer sends the message right away. There can be use cases where this is not desired, therefor Spooler is a mailer implementation that stores your Messages until it is flushed.

```
<?php

use Stampie\Spooler;
use Stampie\Spool\MemoryStorage;

$mailer = new Spooler(new DirectMailer, new MemoryStorage);

// send some messages

// flush
$mailer->flushSpool();
```

1.2 Sending Messages

Stampie comes with an implemetation of the Message interface called DefaultMessage. DefaultMessage makes it easy to start sending out messages. DefaultMessage is a simple ValueObject and takes every possible argument in its constructor.

It is important to know that a Message does not contain the recipient as it did in version 0.x.x. This is because the recipient is almost always a dynamic value, and therefore it does not make sense to have it on the message.

The recipient is wrapped as an Identity. An Identity always consists of the recipient email and sometimes also the name. If you use the name it will show up as “My Name <noreply@email.tld>” in your inbox.

```
<?php
```

```
use Stampie\Message\DefaultMessage;
use Stampie\Identity;

// assuming $mailer is the same as the previous example.
$from = new Identity('from@domain.tld', 'Optional From Name');

$message = new DefaultMessage($from, 'Subject', '<b>Html</b>', 'Text');

$mailer->send(new Identity('to@domain.tld', 'Optional To Name'), $message);
```

The send method returns a MessageHeader. The MessageHeader contains a identifier given by the carrier. This identifier is different from carrier to carrier. Also if an event stopped the sending the identifier will be null.

1.3 Running the Examples

Stampie comes with a example for each carrier. This will send a email through the carrier to the email you provide.

```
$ php example/insert-carrier.php "my-api-key" "to@domain.tld" "from@domain.tld"
```

Carriers

2.1 Postmark

Using Postmark is simple, but first you have to obtain a key from [their website](<http://postmarkapp.com>) and configure your Server and Sender Profile.

```
<?php
```

```
$carrier = new Stampie\Carrier\PostmarkCarrier('my-api-key');
```

2.2 Mandrill

Coming soon

2.3 MailGun

Using MailGun you first have to create an account and setup you domain and so on. The key used for configuring MailGun is `{domain}:{API-Key}`. An example of that would be `bjrnskov.mailgun.org:my-secret-api-key`.

```
<?php
```

```
$carrier = new Stampie\Carrier\MailGunCarrier('domain:secret-api-key');
```