CakePHP-MailPreview Documentation

Release 1.0.0

Jose Diaz-Gonzalez

December 21, 2016

Contents:

CHAPTER 1

Introduction

1.1 CakePHP MailPreview Plugin

A simple CakePHP plugin for use with previewing emails during development.

This plugin was inspired by the Rails 4.1 feature, Action Mailer Previews, as well as the 37signals `mail_view` gem.

Alternatives to this plugin include any of the following:

- CatchMe: A nodejs app to catch, display and validate emails
- MailCatcher: A super simple SMTP server which catches any message sent to it to display in a web interface.
- MailHog: A Web and API based SMTP testing service.
- MailTrap: An email testing service

This plugin is *specifically* meant to bring Rapid Application Development to email development by enabling developers to simply update a template and reload the browser. Developers are encouraged to use other tools as their needs change.

1.2 Requirements

- CakePHP 3.x
- PHP 5.5+

1.3 What does this plugin do?

- preview HTML or plain text emails from within your web browser
- emails are reloaded with each view so you can tweak/save/refresh for instant verification
- integrates perfectly with existing test fixtures
- only exposes routes in development mode to prevent leaking into production mode

CHAPTER 2

Installation

The only officialy supported method of installing this plugin is via composer.

2.1 Using Composer

View on Packagist, and copy the json snippet for the latest version into your project's `composer.json`. Eg, v. 0.0.1 would look like this:

```
{
    "require": {
        "josegonzalez/cakephp-mail-preview": "0.0.1"
    }
}
```

2.2 Enable plugin

You need to enable the plugin your `config/bootstrap.php` file:

```
<?php
Plugin::load('Josegonzalez/MailPreview', ['routes' => true]);
```

If you are already using `Plugin::loadAll();`, then this is not necessary.

CHAPTER 3

Usage

3.1 Basic example

MailPreview integrates with CakePHP's Mailer class. All mailers should use the Josegonzalez\Mailer\PreviewTrait trait. Below is an example UserMailer with a welcome email method:

```
<?php
namespace App\Mailer;

use Cake\Mailer\Mailer;
use Josegonzalez\MailPreview\Mailer\PreviewTrait;

class UserMailer extends Mailer
{
    use PreviewTrait;

    public function welcome($user)
    {
        $this
            ->to($user->email)
            ->subject(sprintf('Welcome %s', $user->name))
            ->template('welcome_mail') // By default template with same name as method name is used.
            ->layout('custom');
    }
}
```

Next, you'll want to create a Preview class for your mailer. As mailers can have multiple methods, the associated Preview class will provide an integration point for each method. All Preview classes should extend Josegonzalez\MailPreview\Mailer\Preview\MailPreview. Here is a UserMailPreview class to accompany our above UserMailer.

```
<?php
namespace App\Mailer\Preview;

use Josegonzalez\MailPreview\Mailer\Preview\MailPreview;

class UserMailPreview extends MailPreview
{
    public function welcome()
    {
        $this->loadModel('Users');
        $user = $this->Users->find()->first();
        return $this->getMailer('User')
                    ->preview('welcome', [$user]);
    }
}
```

A few things to note here:

- `MailPreview` classes are in the `App\Mailer\Preview` namespace, and **must** extend the `Josegonzalez\MailPreview\Mailer\Preview\MailPreview` class. The path to the MailPreview class is `src/Mailer/Preview/ClassName.php`.
- The return function of each mailer **must** be the result of the `->preview()` call on the `Mailer` object. This is injected into the class by our aforementioned `Josegonzalez\MailPreview\Mailer\PreviewTrait`.
- The `->preview()` call uses the `Cake\Mailer\Transport\DebugTransport` email transport to retrieve the results of the sent email without actually sending it, and also injects some extra metadata for use in the ui.
- The `->preview()` call has the same api as the `->send()` call from the `Mailer` class.

Once we have our `UserMailPreview` class in place, we can view them at the `/mail-preview` url of your application. This route is loaded by the plugin routes, so be sure to have those enabled when you install the plugin.

Plugin configuration options

This plugin can be configured via your `config/app.php`. Here is an example config stanza:

```
/**
 * Configures the MailPreview plugin
 */
'MailPreview' => [
    'Routes' => [
        // the router class for the MailPreview plugin
        'class' => 'Cake\Routing\Route\DashedRoute',
        // prefix to use for accessing the MailPreview plugin routes
        'prefix' => '/mail-preview',
    ],
    'Previews' => [
        // A list of classNames to override the automatically detected classes
        // Useful when loading previews from plugins
        'classNames' => [
            'App\Mailer\Preview\UserMailPreview',
        ],
    ],
],
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search