Flow SwiftMailer Documentation

Release 7.0.1

The Neos Team

Apr 12, 2022

Contents

This package allows to easily use the Swift Mailer library with Flow applications.

This version of the documentation covering release 7.0.1 has been rendered at: Apr 12, 2022

CHAPTER 1

Installation

The package can be installed via composer:

```
composer require neos/swiftmailer
```

CHAPTER 2

Configuration

To set up the mail transport to be used, adjust the settings as needed. Without any further configuration, mails will be sent using `Swift_SendmailTransport` which uses `sendmail` on the server. To adjust the sendmail command, you can use:

```
Neos:
  SwiftMailer:
    transport:
      type: 'Swift_SendmailTransport'
      options:
        command: '/usr/sbin/sendmail -bs'
```

To use SMTP for sending, follow the following example:

```
Neos:
  SwiftMailer:
    transport:
      type: 'Swift_SmtpTransport'
      options:
        host: 'smtp.example.com'
        port: '465'
        encryption: 'ssl'
        username: 'myaccount@example.com'
        password: 'shoobidoo'
        localDomain: 'example.com'
```

The encryption property supports values `ssl` and `tls`.

Further transports are available with Swift Mailer and can be used as well. Their options can be looked up the Swift Mailer documentation and they can be set by extrapolating from their setter method names (as in: `setUsername()` becomes `username` in the options.)

The `Swift_MailTransport` that was available in the past has been removed with the Swift Mailer library 6.0 release, see https://github.com/swiftmailer/swiftmailer/issues/866 for background information.

CHAPTER 3

Sending mail

If a transport is configured, sending mail is as simple as this:

- create new `Neos\SwiftMailer\Message` instance
- set your sender address with `setFrom()`
- set a subject line with `setSubject()`
- set recipients with `setTo()`, `setCc()`, `setBcc()`
- set a body with `setBody()`
- add attachments with `attach()`
- send with `send()`

Here is an example:

```
$mail = new \Neos\SwiftMailer\Message();

$mail
    ->setFrom(array($senderAddress => $senderName))
    ->setTo(array($recipientAddress => $recipientName))
    ->setSubject($subject);

$mail->setBody($messageTxt, 'text/plain');
$mail->addPart($messageHtml, 'text/html');

$mail->send();
```

CHAPTER 4

Debugging sent mail

To debug sent mails, an easy way is to configure the transport to the the mbox handler of this package. If this is done in a context-specific configuration (for *Development* or a sub-context on a staging server), it can be safely committed to a VCS:

```
Neos:
  SwiftMailer:
    transport:
      type: 'Neos\SwiftMailer\Transport\MboxTransport'
      options:
        mboxPathAndFilename: '%FLOW_PATH_DATA%/Persistent/sent-mail'
```

All sent mails will be added to the configured mbox file and can be read with any client that can handle the mbox file format.

A second option is to use the `LoggingTransport`, which logs all mails to the *SystemLog* of Flow:

```
Neos:
  SwiftMailer:
    transport:
      type: 'Neos\SwiftMailer\Transport\LoggingTransport'
```