blar/mail Documentation

Release dev

The blar/mail devel team

Sep. 27, 2017

Inhaltsverzeichnis

Mail für PHP.

KAPITEL 1

Inhaltsverzeichnis

Requirements

- PHP 5.5+
- blar/mime @dev
- blar/curl @dev
- blar/common @dev

Installation

Es wird die Installation per Composer empfohlen, da die Abhängigkeiten automatisch aufgelöst werden können. Bei der Installation per Git müssen die Abhängigkeiten manuell gelöst werden.

Installation mit Composer

```
$ composer require blar/mail @dev
```

Installation mit Git

```
$ git clone https://github.com/blar/mail.git
```

Mail

Transports

Es werden verschiedene Transports bereitgestellt um E-Mails zu versenden.

Sendmail-Transport

```
$transport = new SendmailTransport();
$transport->sendMail($mail);
```

Bemerkung: Der Versand von E-Mails mit dem Sendmail-Transport ist meist einfacher, da Sendmail vom Administrator vorkonfiguriert sein sollte und keine Zugangsdaten benötigt werden.

Curl-Transport

```
$transport = new CurlTransport('smtp.example.com');
$transport->setCredentials('username', 'password');
$transport->sendMail($mail);
```

Bemerkung: Der Versand von E-Mails per Curl im selben Script ist deutlich schneller (~1.000%), da nicht für jede E-Mail ein neuer Prozess von Sendmail gestartet wird, sondern alle E-Mails mit der selben SMTP-Verbindung versendet werden.

Beispiel

```
$mail = new Mail();
$mail->setFrom('foobar@example.com');
$mail->setTo('foo@example.com, bar@example.com');
```

Bemerkung: Mail::setTo() setzt den Envelope-To. In dieser Liste sind alle Empfänger enthalten. Es werden jedoch nur Empfänger in der E-Mail angezeigt, die in den Headern **To** oder **BCC** angegeben sind.

E-Mail-Adressen als Objekte

```
$mail = new Mail();
$mail->setFrom(new MailAddress('foobar@example.com', 'Foo Bar'));

$to = new AddressCollection();
$to->push('foo@example.com');
$to->push('bar@example.com');
$mail->setTo($to);
```

Zusätzliche Header

```
$mail = new Mail();
$mail->setFrom(new MailAddress('foobar@example.com', 'Foo Bar'));

$to = new AddressCollection();
$to->push('foo@example.com');
$to->push('bar@example.com');
$mail->setTo($to);

$headers = $mail->getHeaders();
$headers->set('Content-Type', 'text/html');
$headers->set('Subject', 'Hello World');

$mail->push('<p>Hello World</p>');
```

Fluid Interface

```
$to = new AddressCollection();
$to
    ->push('foo@example.com')
    ->push('bar@example.com');

$mail = new Mail();

$mail
    ->setFrom(new MailAddress('foobar@example.com', 'Foo Bar'))
    ->setTo($to);

$mail
     ->getHeaders()
     ->set('Content-Type', 'text/html')
     ->set('Subject', 'Hello World')
     ->push('<p>Hello World</p>');
```