Please note that GitHub no longer supports old versions of Safari.
We recommend upgrading to the latest Safari, Google Chrome, or Firefox.

Learn more Ignore

Search Explore Features Enterprise Blog Sign up Sign in

FriendsOfSymfony / **FOSUserBundle**
forked from KnpLabs/KnpUserBundle

Star 1,748 Fork 1,007

branch: **master** FOSUserBundle / Resources / doc / **emails.md**

armetiz July 17, 2013 The format is required by FrameworkBundle

5
contributors

218 lines (167 sloc) | 6.385 kb

Raw Blame History

FOSUserBundle Emails

The FOSUserBundle has built-in support for sending emails in two different instances.

Registration Confirmation

The first is when a new user registers and the bundle is configured to require email confirmation before the user registration is complete. The email that is sent to the new user contains a link that, when visited, will verify the registration and enable the user account.

Requiring email confirmation for a new account is turned off by default. To enable it, update your configuration as follows:

```
# app/config/config.yml

fos_user:
    # ...
    registration:
        confirmation:
            enabled:    true
```

Password Reset

An email is also sent when a user has requested a password reset. The FOSUserBundle provides password reset functionality in a two-step process. First the user must request a password reset. After the request has been made, an email is sent containing a link to visit. Upon visiting the link, the user will be identified by the token contained in the url. When the user visits the link and the token is confirmed, the user will be presented with a form to enter in a new password.

Default Mailer Implementations

The bundle comes with three mailer implementations. They are listed below by service id:

- `fos_user.mailer.default` is the default implementation, and uses Swiftmailer to send emails.
- `fos_user.mailer.twig_swift` uses Swiftmailer to send emails and Twig blocks to render the message.
- `fos_user.mailer.noop` is a mailer implementation which performs no operation, so no emails are sent.

Note:

> The `fos_user.mailer.noop` mailer service should be used in the case where you do not want the bundle to send emails and you do not want to include the SwiftmailerBundle in your app. If you leave the default implementation configured as the mailer and do not have the SwiftmailerBundle registered, you will receive an exception because of a missing dependency.

Configuring the Sender Email Address

The FOSUserBundle default mailer allows you to configure the sender email address of the emails sent out by the bundle. You can configure the address globally or on a per email basis.

To configure the sender email address for all emails sent out by the bundle, simply update your `fos_user` config as follows:

```
# app/config/config.yml
fos_user:
    #...
    from_email:
        address:        noreply@acmedemo.com
        sender_name:    Acme Demo App
```

The bundle also provides the flexibility of allowing you to configure the sender email address for the emails individually.

To configure the sender email address for the user registration confirmation email update your `fos_user` config as follows:

```
# app/config/config.yml
fos_user:
    #...
    registration:
        confirmation:
            from_email:
                address:        registration@acmedemo.com
                sender_name:    Acme Demo Registration
```

You can similarly update the `fos_user` config to change the sender email address for the password reset request email:

```
# app/config/config.yml
fos_user:
    #...
    resetting:
        email:
            from_email:
                address:        resetting@acmedemo.com
                sender_name:    Acme Demo Resetting
```

Sending HTML mails

The default mailer only supports sending plain text messages. If you want to send multipart messages, the easiest solution is to use the TwigSwiftMailer implementation instead. It expects your twig template to define 3 blocks:

- `subject` containing the email subject
- `body_text` rendering the plain text version of the message
- `body_html` rendering the html mail

Here is how you can use it:

```yaml
# app/config/config.yml

fos_user:
    # ...
    service:
        mailer: fos_user.mailer.twig_swift
    resetting:
        email:
            template: AcmeDemoBundle:User:resetting.email.html.twig
```

```twig
{# src/Acme/DemoBundle/Resources/views/User/resetting.email.html.twig #}

{% block subject %}Resetting your password{% endblock %}

{% block body_text %}
{% autoescape false %}
Hello {{ user.username }} !

You can reset your email by accessing {{ confirmationUrl }}

Greetings,
the Acme team
{% endautoescape %}
{% endblock %}

{% block body_html %}
{#
    You can of course render the html directly here.
    Including a template as done here allows keeping things DRY by using
    the template inheritance in it
#}
{% include 'AcmeDemoBundle:User:resetting_email.html.twig' %}
{% endblock %}
```

Note:

> The HTML part is set in the message only when the `body_html` block is not empty.

Using A Custom Mailer

The default mailer service used by FOSUserBundle relies on the Swiftmailer library to send mail. If you would like to use a different library to send emails, want to send HTML emails or simply change the content of the email you may do so by defining your own service.

First you must create a new class which implements `FOS\UserBundle\Mailer\MailerInterface` which is listed below.

```php
<?php

namespace FOS\UserBundle\Mailer;

use FOS\UserBundle\Model\UserInterface;

/**
 * @author Thibault Duplessis <thibault.duplessis@gmail.com>
 */
interface MailerInterface
{
    /**
     * Send an email to a user to confirm the account creation
     *
     * @param UserInterface $user
     */
    function sendConfirmationEmailMessage(UserInterface $user);

    /**
     * Send an email to a user to confirm the password reset
     *
     * @param UserInterface $user
     */
    function sendResettingEmailMessage(UserInterface $user);
}
```

After you have implemented your custom mailer class and defined it as a service, you must update your bundle configuration so that FOSUserBundle will use it. Simply set the `mailer` configuration parameter under the `service` section. An example is listed below.

In YAML:

```yaml
# app/config/config.yml

fos_user:
    # ...
    service:
        mailer: acme.mailer
```

To see an example of a working implementation of the `MailerInterface` see the ZetaMailer class of the ZetaWebmailBundle. This implementation uses the Zeta Components Mail to send emails instead of Swiftmailer.