

CakePHP 4 Sending Email

Asyraf Wahi Anuar - February 01, 2022

<https://codethepixel.com/CakePHP-4-Sending-Email>



Sending email in CakePHP is simple and easy. Mailer class makes the email sending process very convenient. Using Mailer, the developer can configure and send email from any place inside your application. Email can be used to send a notification to a user or moderator regarding the new registration, contact support, mailing list etc. The developer needs to have a valid email to ensure that the sent email is not going to the spam folder.

Email Transport Configuration

File location: ...\\config\\app_local.php

Set your email account as follows. This email configuration will be used to send emails to the newly registered user and password reset requests.

```
'EmailTransport' => [
    'smtp' => [
        'host' => 'YourHostName', //eg: codethepixel.com
        'port' => 26,
        'username' => 'EmailAddress', //eg: admin[at]codethepixel.com
        'password' => 'Secret', //email password
        'className' => 'Smtplib',
        'client' => null,
        'url' => env('EMAIL_TRANSPORT_DEFAULT_URL', null),
    ],
],
```

or you can use a Gmail account. However, it required you to [allow less secure app access in your Google account](#) setting to allow your Gmail to be accessed from your localhost. Refer to the [Google announcement here](#).-

```
'EmailTransport' => [
    'gmail' => [
        'host' => 'smtp.gmail.com',
        'port' => 587,
        'username' => 'YourGmailAddress', //eg: sample[at]gmail.com
        'password' => 'Secret', //email password
        'className' => 'Smtplib',
        'tls' => true,
        'client' => null,
        'url' => env('EMAIL_TRANSPORT_DEFAULT_URL', null),
    ],
],
```

```
],
```

Configure Email Class

File location: ...\\src\\Controller\\UsersController.php

Load the email class at the beginning of the controller. This tutorial used the 'user' table as a reference.

```
use Cake\\Mailer\\Email;
use Cake\\Mailer\\Mailer;
use Cake\\Mailer\\TransportFactory;
```

Send Email Without Template

File location: ...\\src\\Controller\\UsersController.php

In this tutorial, we assumed that when the user completed the registration process, an email will be sent to their registered email account. The name and email variable are requested to be used in the email content. Then using the \$mailer, the developer needs to identify the email transport (SMTP or Gmail or others transport that has been configured), sender, receiver, format, subject and content of the email as shown below:

```
public function add()
{
    $user = $this->Users->newEmptyEntity();

    if($this->request->is('post')){
        $userTable = TableRegistry::get('Users');
        $name = $this->request->getData('name');
        $email = $this->request->getData('email');
        $user = $userTable->newEntity($this->request->getData());
        if($userTable->save($user)){
            $user->name = $name;
            $user->email = $email;

            $mailer = new Mailer('default');
            $mailer
                ->setTransport('smtp'); //your email configuration name
                ->setFrom(['noreply[at]codethepixel.com' => 'Code The Pixel');
                ->setTo($email)
                ->setEmailFormat('html')
                ->setSubject('Verify New Account')
                ->deliver('Hi $name<br/>Welcome to Code The Pixel.');

            $this->Flash->success(__('Your account has been registered.));
            return $this->redirect(['action' => 'index']);
        }
        else
        {
            $this->Flash->error(__('Registration failed, please try again.));
        }
    }
    $this->set(compact('user'));
}
```

Send Email Using Template

Developer can create a email template in ...\\templates\\email\\html\\new_user.php and used the value from \$setViewVars. The configuration is shown as follows:

```
public function add()
{
    $user = $this->Users->newEmptyEntity();
    if($this->request->is('post')){
        $userTable = TableRegistry::get('Users');
        $name = $this->request->getData('name');
        $email = $this->request->getData('email');
        $user = $userTable->newEntity($this->request->getData());
        if($userTable->save($user)){
            $user->name = $name;
            $user->email = $email;

            $mailer = new Mailer('default');
            $mailer
                ->setTransport('smtp'); //your email configuration name
                ->setViewVars([ //email view variables
                    'name' => $name,
                    'email' => $email,
                ])
                ->setFrom(['noreply[at]codethepixel.com' => 'Code The Pixel'])
                ->setTo($email)
                ->setEmailFormat('html')
                ->setSubject('Verify New Account')
                ->viewBuilder()
                    ->setTemplate('new_user');
            $mailer->deliver();

            $this->Flash->success(__('Your account has been registered.'));
            return $this->redirect(['action' => 'index']);
        }
        else
        {
            $this->Flash->error(__('Registration failed, please try again.
        });
    }
    $this->set(compact('user'));
}
```

Example of Email Template (new_user.php)

Hi <?= \$name; ?>,
Welcome to Code The Pixel.

You also can refer [CakePHP cookbook](#) for more details. That's all. Happy coding :)