Most of the e-mail management tools (Gmail, Hotmail, Yahoo…) don’t show  images if they are coming from an external source,  like the following link:

<img src=”[http://www.example.com/politician.jpg](http://images.nationalgeographic.com/wpf/media-live/photos/000/007/cache/spider-monkey_719_600x450.jpg)“>

The solution for this issue is to embed the images into the email. It’s  well  explained on [php.net](http://ie2.php.net/manual/en/function.mail.php), but probably it’ll take a precious time. Fortunately for the Zend Framework developers there is a faster way.  I’ve implemented the code developed by [David Nussio](http://davidnussio.wordpress.com/2008/09/21/inline-images-into-html-email-with-zend-framework/) in the following model:

|  |
| --- |
|  |

//Mail.php

<?php

class Application\_Model\_Mail extends Zend\_Mail

{

    public static function sendMail($subject=null,$body=null, array $recipients){

        if(empty($recipients)){

            throw new Exception('There is no email recipients defined');

        }

        $mail = new self();

        $mail->setBodyHtml($body,'UTF-8',Zend\_Mime::MULTIPART\_RELATED);

        $mail->setFrom('example@zenddveloper.com');

        foreach($recipients as $recipient){

            $mail->addTo($recipient);

        }

        $mail->buildHtml();

        $mail->setSubject($subject);

        $mail->send();

    }

    public function buildHtml()

    {

        // Important, without this line the example don't work!

        // The images will be attached to the email but these will be not

        // showed inline

        $this->setType(Zend\_Mime::MULTIPART\_RELATED);

        $matches = array();

        preg\_match\_all("#<img.\*?src=['\"][file://](file:///)([^'\"]+)#i",

        $this->getBodyHtml(true),

        $matches);

        $matches = array\_unique($matches[1]);

        if (count($matches ) > 0) {

            foreach ($matches as $key => $filename) {

                if (is\_readable($filename)) {

                    $at = $this->createAttachment(file\_get\_contents($filename));

                    $at->type = $this->mimeByExtension($filename);

                    $at->disposition = Zend\_Mime::DISPOSITION\_INLINE;

                    $at->encoding = Zend\_Mime::ENCODING\_BASE64;

                    $at->id = 'cid\_' . md5\_file($filename);

                    $this->setBodyHtml(str\_replace('[file://](file:///)' . $filename,

                                       'cid:' . $at->id,

                    $this->getBodyHtml(true)),

                                       'UTF-8',

                    Zend\_Mime::ENCODING\_8BIT);

                }

            }

        }

    }

    public function mimeByExtension($filename)

    {

        if (is\_readable($filename) ) {

            $extension = pathinfo($filename, PATHINFO\_EXTENSION);

            switch ($extension) {

                case 'gif':

                    $type = 'image/gif';

                    break;

                case 'jpg':

                case 'jpeg':

                    $type = 'image/jpg';

                    break;

                case 'png':

                    $type = 'image/png';

                    break;

                default:

                    $type = 'application/octet-stream';

            }

        }

        return $type;

    }

}

For this example I’m using a mail template that is located in ”application/views/scripts/mail-templates/“,  take a look at the  src attribute  of the images. All the paths  ” file://imgs/imageName” will be replaced for the embed  content of the image located on that path.

//Mail template

//example.phtml

<h1>This is a mail example</h1>

<img src="[file://imgs/image1.jpg](smb://imgs/image1.jpg)">

<img src="[file://imgs/image2.jpg](smb://imgs/image2.jpg)">

At this point we only need to call the static method sendMail with the appropiate params  (topic, body and recipients).

//Piece of code of SendMailController.php

$this->html = new Zend\_View();//Create view

$this->html->setScriptPath('../application/views/scripts/mail-templates/');

$topic='Mail topic'; // set email subject

$body=$this->html->render('example.phtml'); //get mail template content

$recipients=array('example@gmail.com'); //Set list of recipients

Application\_Model\_Mail::sendMail($topic, $body, $recipients);