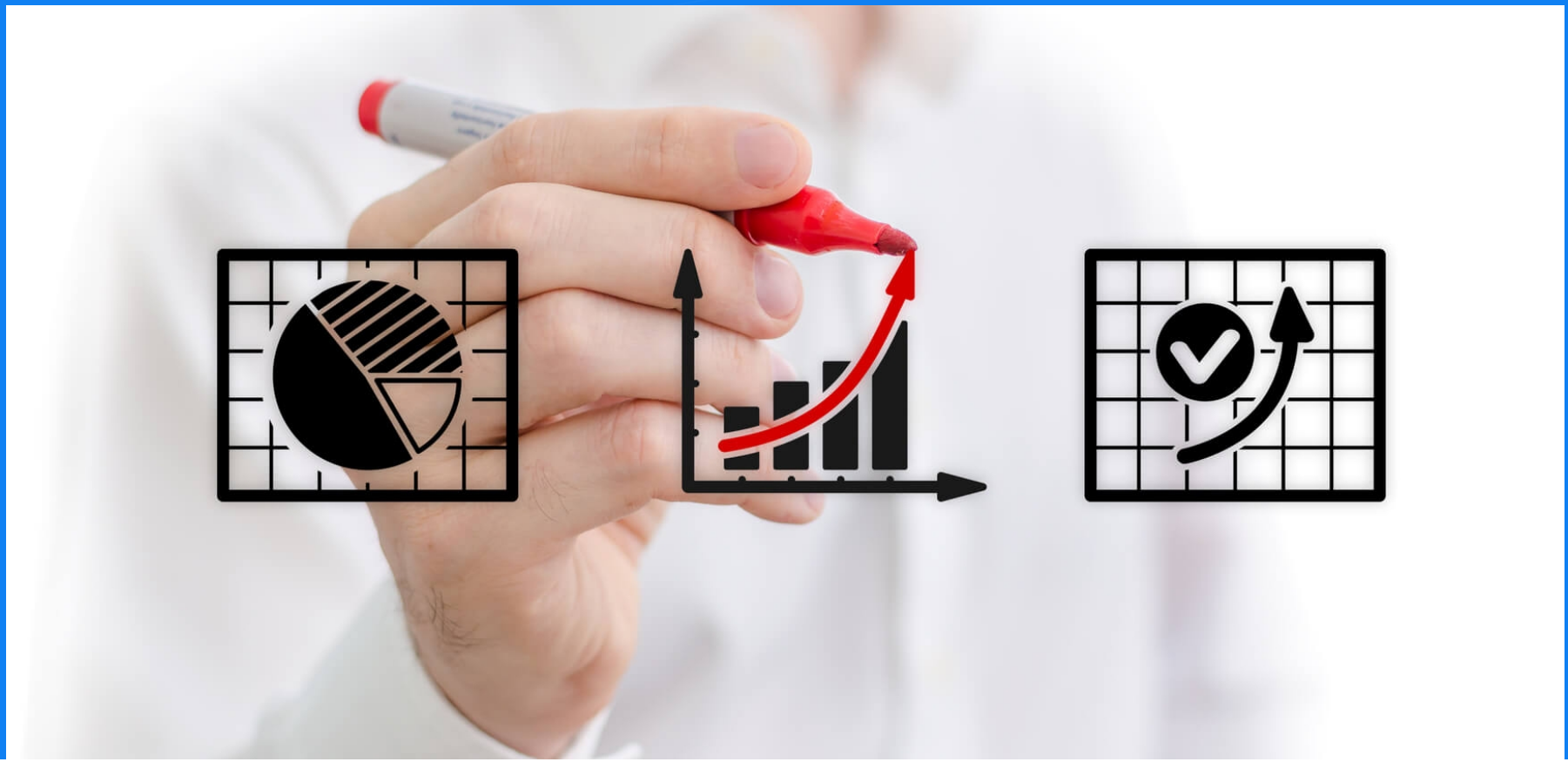
chaos LABORATORY

HOME ABOUT US PROJECTS CLIENTS BLOG CONTACT US

Simple. Robust. Reliable.

Spend less time worrying about front-end and more focusing on your product

LEARN MORE



AJAX based Currency Converter

CakePHP Securimage Component

On SecurImage

> Securimage is an open-source free PHP CAPTCHA script for generating complex images and CAPTCHA codes to protect forms from spam and abuse. It can be easily added into existing forms on your website to provide protection from spam bots. It can run on most any webserver as long as you have PHP installed, and GD support within PHP. Securimage does everything from generating the CAPTCHA images to validating the typed code. Audible codes can be streamed to the browser with Flash for the vision impaired.
> *SecurImage CAPTCHA*

So what is the CakePHP SecurImage Component

It is a component for the **CakePHP MVC framework** that makes it easy to implement CAPTCHA codes in your pages or forms or as a matter of fact, wherever you want to enforce a check against automated bots.
This component was first published at **The Bakery** for CakePHP 1.2.x and then republished at the **World of Baud** as an upgraded version compatible with CakePHP 1.3.x. This isn't an original work and I take no credit for creation of the original component. However, the project seems to have been ditched for a good while, whereas both CakePHP and SecurImage keep evolving. Certain configuration options have become deprecated while a whole bunch of new ones have been introduced.

This is an attempt to keep up with the evolution of SecurImage and make the correct set of options available in the component.
Moreover, certain values passed to the component (e.g. Color Codes – foreground or background) now need to be converted to a native **SecureImage_Color** format. The component handles this task, letting the end-user specify the colors in standard hexademical notation (e.g. #cccccc).

Getting the component

You can grab this component by visiting its **GitHub Repository** or use the following links to grab the Zip or Tarball directly.
[download id="11"]
[download id="12"]

Installation

First and foremost, you are required to obtain the **SecurImage library** and unzip it into your **app/vendors** folder.
If you are familiar with the CakePHP framework, usage of this component should be fairly evident to you.
The zip file contains:

1. A component file that should be placed in **controllers/components** folder of your application, and
2. A view file, that is to be placed in **views/elements** folder.

If effect, the folder structure should be as follows:

```
app
 |-controllers
 |   |-components
 |      |-securimage.php (Component)
 |-vendors
 |   |-securimage (The folder you get from unzipping the Securimage library)
 |-views
    |-elements
    |-securimage.ctp
```

Usage

The component should be included like a standard CakePHP component, in whichever controller you wish to use it. In my examples, I'm using a controller named **ContactsController**. The corresponding model's name is **Contact**.

Simple example:

```php
class ContactsController extends AppController {

    // Components
    var $components = array(
        'Securimage'
    );

}
```

Example with parameters:

```php
class ContactsController extends AppController {

    // Components
    var $components = array(
        'Securimage' => array(
            'code_length' => 6,
            'font_size' => 60,
            'image_width' => 200,
            'image_height' => 70,
            'image_bg_color' => '#F7F7F7',
            'line_color' => '#D3D3D3',
            'multi_text_color' => '#8E67D6,#B98B83,#529071,#7C3E39,#E07E6A,#46765D',
            'num_lines' => 8,
            'perturbation' => 0.5,
            'text_transparency_percentage' => 20,
            'use_multi_text' => true,
            'use_wordlist' => true,
        ),
    );

}
```

Note: For a full list of available parameters (configuration options) please take a look into the component file. Details of each option are included in it in PHP Doc format. Alternatively you can view them below:

```
Full list of parameters
```

In your view file the CAPTCHA image can be displayed in the following manner:

```php
<div id="captcha_container">
    <img id="captcha_img" src="/contacts/securimage/0" alt="CAPTCHA image" />
    <img id="captcha_reload" src="/img/icon-reload.png" title="Refresh CAPTCHA" />
    <input id="captcha_text" name="data[Contact][captcha_text]" value="" />
</div>
```

The purpose of the second image is to act like a refresh button, in case your captcha is unintelligible. A little bit of javascript can aid in refreshing the image dynamically.

```
// Using jQuery
$( '#captcha_reload' ).click( function() {
    $( '#captcha_img' ).attr('src', '/contacts/securimage/' + Math.random()); // Append rand
    $( '#captcha_text' ).val('');
});
```

If you have any questions, feel free to write back and I'll try my best to respond ASAP.

Last, but not the least

While working on the component, I came by a cool site named **CaptchaFails**, with a whole bunch of really funny captchas that people tend to encounter on the net. Have Fun!!

About Us

Boost your Sales with CRMs

Web-applications

Mobile applications

Cross-Platform Integrations

Finding it difficult to move

We offer consultancy & implementation services for major Open Source & Enterprise CRMs. Our custom crafted Performance Dashboards help you maximize on your ROI and make the best out of your Marketing and Sales campaigns.

No two businesses are equal. Develope your system to match your business processes. The possibilities are endless!

Be visible to your customers at all times. Step into the new paradigm of building brand and recognition, while improving customer engagement – all at the same time.
Stand out from the competition while cultivating customer loyalty.

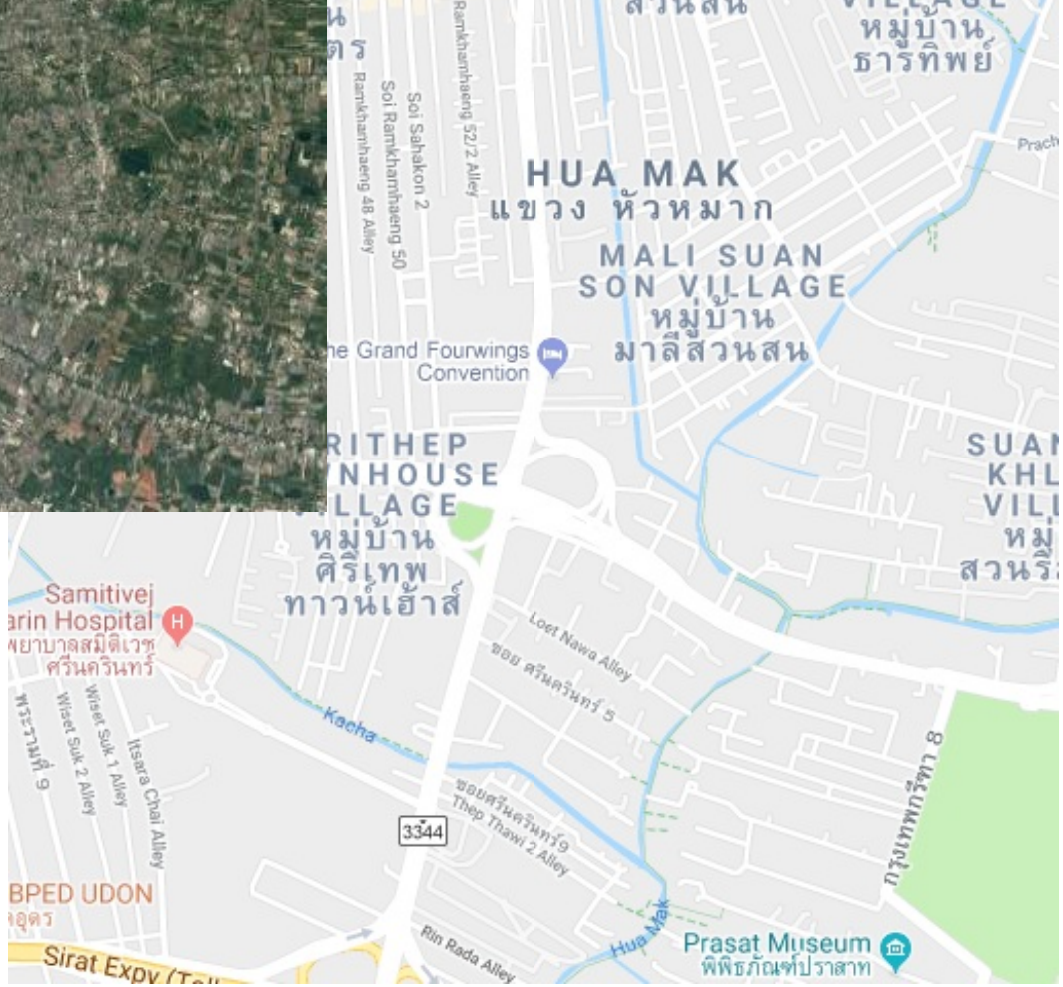
from legacy systems? Let us build custom integrations with the latest and the best in the SaaS world.

Clients

Fields marked with an * are required

Name *

Email *



Message *

Send