Customizing Captcha

A standard Yii Captcha is good enough to protect you from spam, but there are situations where you may  
want to customize it, such as the following:

* You face a spam bot that can read image text and you need to add more security
* You want to make it more interesting or easier to enter the Captcha text

In our example, we will modify Yii’s Captcha so it will require the user to solve a really simple arithmetic  
puzzle instead of just repeating the text in an image.

Getting ready

As a starting point for this example, we will take the result of the Adding and customizing  
CaptchaWidget recipe. Alternatively, you can take any form that uses Captcha, as we are not modifying  
the existing code a lot.

How to do it...

We need to customize captchaAction, which generates the code and renders its image representation.  
The code should be a random number and the representation should be an arithmetic expression that gives  
the same result:

1. Create an @app/components/MathCaptchaAction. php action as follows:

<?php

namespace app\components;  
use \Yii;

use yii\captcha\CaptchaAction;  
class MathCaptchaAction extends CaptchaAction  
{

protected function renderImage($code)

{

return parent::renderImage($this->getText($code));

}

protected function generateVerifyCode()

{

return mt\_rand((int)$this->minLength,

(int)$this->maxLength);

}

protected function getText($code)

{

$code = (int) $code;

$rand = mt\_rand(1, $code-1);

$op = mt\_rand(0, 1);  
if ($op) {

return $code - $rand . " + " . $rand;

}

else {

return $code + $rand . " - " . " " . $rand;

}

}

}

1. Now, in our controller’s actions method, we need to replace captchaAction with our own  
   Captcha action, as follows:

public function actions()

{

return [

'captcha' => [

'class' => 'app\components\MathCaptchaAction',

'minLength' => 1,

'maxLength' => 10,

],

];

}

1. Now, run your form and try the new Captcha. It will show arithmetic expressions with numbers  
   from 1 to 10 and will require entering an answer, as shown in the following screenshot:



We override two CaptchaAction methods. In generateVerifyCode(), we generate a random number  
instead of text. Then, as we need to render an expression instead of just showing text, we override  
renderImage. The expression itself is generated in our custom getText () method.

The $minLength and $maxLenght properties are already defined in CaptchaAction, so we don’t have to  
add them to our Math CaptchaAction class.

See also

For further information, refer to the following:

* [http://www.yiiframework.eom/doc-2.0/yii-captcha-captcha.html](http://www.yiiframework.com/doc-2.0/yii-captcha-captcha.html)
* [http://www.viiframework.com/doc-2.0/vii-captcha-captchaartion.html](http://www.yiiframework.com/doc-2.0/yii-captcha-captchaaction.html)
* The Using standalone actions recipe in Chapter 2, Routing, Controllers, and Views