Adding and customizing CaptchaWidget

Nowadays, on the Internet, if you leave a form without spam protection, you will get a ton of spam data  
entered in a short time. Yii includes a Captcha component that makes adding such protection a breeze.  
The only problem is that there is no systematic guide on how to use it.

In the following example, we will add Captcha protection to a simple form.

Getting ready

1. Create a new application by using the Composer package manager, as described in the official  
   guide at [http://www. yiiframework. c om/doc-2.0/guide -start-installation .html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html).
2. Create a form model, @app/models/EmailForm.php, as follows:

<?php

namespace app\models;  
use yii\base\Model;  
class EmailForm extends Model  
{

public $email;

public function rules()

{

return [

['email', 'email']

];

}

}

1. Create a controller, @app/controllers/EmailController . php, as follows:

<?php

namespace app\controllers;  
use Yii;

use yii\web\Controller;  
use app\models\EmailForm;  
class EmailController extends Controller  
{

public function actionIndex(){

$success = false;

$model = new EmailForm();

if ($model->load(Yii::$app->request->post()) && $model->validate()) {

Yii::$app->session->setFlash('success', 'Success!');

}

return $this->render('index', [

'model' => $model,

'success' => $success,

]);

}

}

1. Create a view, @app/views/email/index.php, as follows:

<?php

use yii\helpers\Html;  
use yii\captcha\Captcha;  
use yii\widgets\ActiveForm;

?>

<?php if (Yii::$app->session->hasFlash('success')): ?>

<div class="alert alert-success"><?=Yii::$app->session->getFlash('success')?>  
</div>

<?php else: ?>

<?php $form = ActiveForm::begin()?>

<div class="control-group">

<div class="controls">

<?= $form->field($model, 'email')->textInput(['class' => 'form-

control ']); ?>

<?php echo Html::error($model, 'email', ['class' => 'help-block'])?

>

</div>

</div>

<?php if (Captcha::checkRequirements() && Yii::$app->user->isGuest): ?>

<div class="control-group">

<?= $form->field($model, 'verifyCode')-  
>widget(\yii\captcha\Captcha::classname(), [

'captchaAction' => 'email/captcha'

]) ?>

</div>

<?php endif; ?>

<div class="control-group">

<label class="control-label" for=""></label>

<div class="controls">

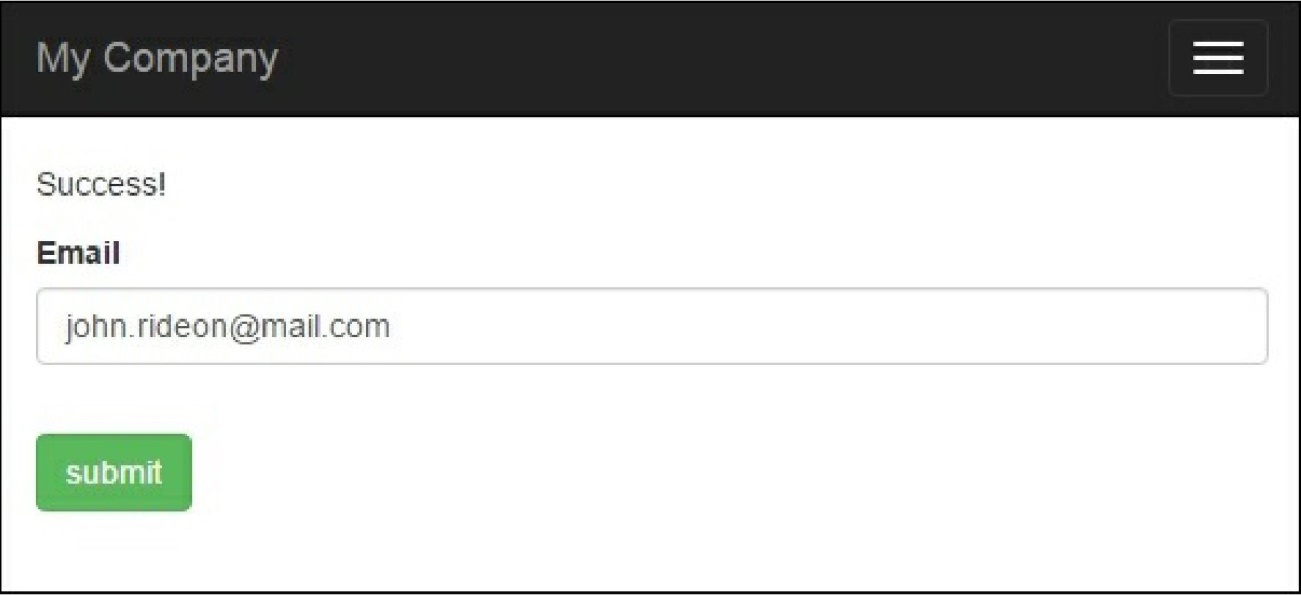
<?=Html::submitButton('Submit', ['class' => 'btn btn-success'])?>  
</div>

</div>

<?php ActiveForm::end()?>

<?php endif;?>

5. Now, we have an e-mail submission form, as shown in the following screenshot, which validates the  
e-mail field. Let’s add Captcha:



How to do it...

1. First, we need to customize the form model. We need to add $verifyCode, which will hold the  
verification code entered and add a validation rule for it:

<?php

namespace app\models;  
use yii\base\Model;  
use yii\captcha\Captcha;  
class EmailForm extends Model  
{

public $email;  
public $verifyCode;  
public function rules()

{

return [

['email', 'email' ],

['verifyCode', 'captcha', 'skipOnEmpty' =>  
!Captcha::checkRequirements(), 'captchaAction' => 'email/captcha']

];

}

}

1. We then need to add an external action to the controller. Add the following code to it:

public function actions()

{

return [

'captcha' => [

'class' => 'yii\captcha\CaptchaAction',

],

];

}

1. In a view, we need to show an additional field and the Captcha image. The following code will do  
   this for us:

<?php if (Captcha::checkRequirements() && Yii::$app->user->isGuest): ?>

<div class="control-group">

<?=Captcha::widget([

'model' => $model,

'attribute' => 'verifyCode',

]);?>

<?php echo Html::error($model, 'verifyCode')?>

</div>

<?php endif; ?>

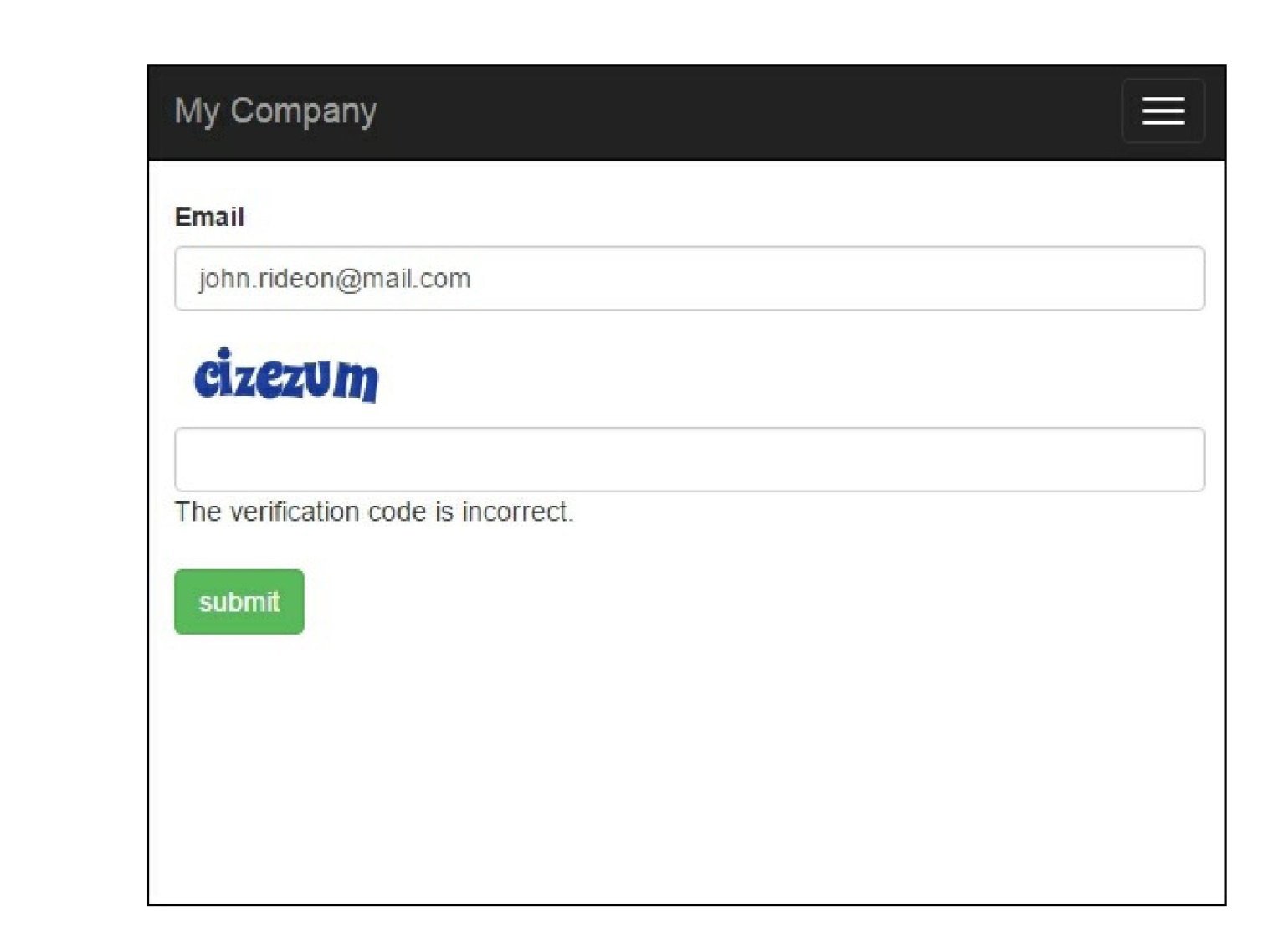
1. Also, do not forget to add the Captcha import in the header section of the view:

<?php

use yii\helpers\Html;  
use yii\captcha\Captcha;

?>

1. That is it. Now, you can run the e-mail controller and see Captcha in action, as shown in the  
   following screenshot:



If there are no errors on the screen and no Captcha field on the form, most probably, you don’t have the  
GD PHP or Imagick extensions installed and configured. Imagick or GD is required for Captcha because  
it generates images. We have added several Captcha: :checkRequirements() checks, so the application  
will not use Captcha if the image cannot be displayed, but it will still work.

How it works...

In a view, we call the Captcha widget that renders the img tag with a src attribute pointing to the Captcha  
action we added to the controller. In this action, an image with a random word is generated. The word  
generated is a code that the user should enter into the form. It is stored in a user session and an image is  
displayed to the user. When the user enters the e-mail and verification code into the form, we assign these  
values to the form model and then validate it. For the verification of the code field, we use  
Captchavalidator. It gets the code from the user session and compares it to the code entered. If they  
don’t match, the model data is considered invalid.

There’s more...

If you restrict access to controller actions by using the accessRules controller method, don’t forget to  
grant everyone access to them:

public function behaviors()

{

return [

'access' => [

'class' => AccessControl::className(),

'rules' => [

[

'actions' => ['index', 'captcha'],

'allow' => true,

]

],

],

];

}