Preventing XSS

XSS stands for cross-site scripting and is a type of vulnerability that allows one to inject a client-side script (typically JavaScript) in a page viewed by other users. Considering the power of client-side scripting, this can lead to very serious consequences such as bypassing security checks, getting other user’s credentials, or data leaks.

In this recipe, we will see how to prevent XSS by escaping the output with both \yii\helpers\Html and \yii\helpers\HtmlPurifier.

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 controllers/XssController.php:

<?php

namespace app\controllers; use Yii;

use yii\helpers\Html; use yii\web\Controller;

/\*\*

* Class SiteController.
* @package app\controllers \*/

class XssController extends Controller {

/\*\*

\* @return string \*/

public function actionIndex()

{

$username = Yii::$app->request->get('username', 'nobody');

return $this->renderContent(Html::tag('h1',

'Hello, ' . $username . '!'

));

}

}

1. Normally, it will be used as /xss/simple?username=Administrator. However, as the main security principle filter input, escape output was not taken into account, malicious users will be able to use it in the following way:

/xss/simple?username=<script>alert('XSS');</script>

1. The previous code will result in a script execution, as shown in the following screenshot: