Creating a custom view renderer

There are many PHP template engines out there. Yii2 only offers native PHP templates. If you want to use one of the existing template engines or create your own one, you have to implement it—of course, if it’s not yet implemented by the Yii community.

In this recipe we’ll re-implement the Smarty templates support.

Getting ready

1. Create a new yii2-app-basic application using the composer, as described in the official guide at [http://www.viiframework.com/doc-2.0/guide-start-installation.html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html).
2. Install the Smarty library:

composer require smarty/smarty

How to do it...

Carry out the following steps for creating a custom view renderer:

1. Create the smarty/ViewRenderer . php file:

<?php

namespace app\smarty;

use Smarty; use Yii;

class ViewRenderer extends \yii\base\ViewRenderer {

public $cachePath = '@runtime/smarty/cache'; public $compilePath = '@runtime/smarty/compile';

/\*\*

\* @var Smarty \*/

private $smarty;

public function init()

{

$this->smarty = new Smarty();

$this->smarty->setCompileDir(Yii::getAlias($this->compilePath)); $this->smarty->setCacheDir(Yii::getAlias($this->cachePath)); $this->smarty->setTemplateDir([

dirname(Yii::$app->getView()->getViewFile()),

Yii::$app->getViewPath(),

]);

}

public function render($view, $file, $params)

{

$templateParams = empty($params) ? null : $params;

$template = $this->smarty->createTemplate($file, null, null, $templateParams, false);

$template->assign('app', \Yii::$app);

$template->assign('this', $view);