Creating reusable controllers

In Yii, you can create reusable controllers. If you are creating a lot of applications or controllers that are of  
the same type, moving all common code into a reusable controller will save you a lot of time.

In this recipe, we try to create a common cleanController, which will clear temporary directories and  
flush cached data.

Getting ready

Create a new yii2-app-basic application using the composer as described in the official guide at  
<http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.

How to do it...

Carry out the following steps to create reusable controllers:

1. Create the cleaner directory and add the standalone CleanController controller:

<?php

namespace app\cleaner;  
use Yii;

use yii\filters\VerbFilter;  
use yii\helpers\FileHelper;  
use yii\web\Controller;

class CleanController extends Controller  
{

public $assetPaths = ['@app/web/assets'];  
public $runtimePaths = ['©runtime'];  
public $caches = ['cache'];

public function behaviors()

{

return [

'verbs' => [

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

'actions' => [

' assets ' => [' post' ],

'runtime' => ['post'],

' cache' => [' post' ],

],

],

];

}

public function actionIndex()

{

return $this->render('@app/cleaner/views/index');

}

public function actionAssets()

{

foreach ((array)$this->assetPaths as $path) {

$this->cleanDir($path);

Yii::$app->session->addFlash(

' cleaner',

'Assets path "' . $path . '" is cleaned.'

);

}

return $this->redirect(['index']);

}

public function actionRuntime()

{

foreach ((array)$this->runtimePaths as $path) {

$this->cleanDir($path);

Yii::$app->session->addFlash(

'cleaner',

'Runtime path "' . $path . '" is cleaned.'

);

}

return $this->redirect(['index']);

}

public function actionCache()

{

foreach ((array)$this->caches as $cache) {

Yii::$app->get($cache)->flush();

Yii::$app->session->addFlash(

'cleaner',

'Cache "' . $cache . '" is cleaned.'

);

}

return $this->redirect(['index']);

}

private function cleanDir($dir)

{

$iterator = new \DirectoryIterator(Yii::getAlias($dir));  
foreach($iterator as $sub) {

if(!$sub->isDot() && $sub->isDir()) {

FileHelper::removeDirectory($sub->getPathname());

}

}

}

}

2. Create the cleaner/views/index . php view file for the actionIndex method:

<?php

use yii\helpers\Html;

/\* @var $this yii\web\View \*/

$this->title = 'Cleaner';

$this->params['breadcrumbs'][] = $this->title;

?>

<div class="clean-index">

<h1><?= Html::encode($this->title) ?></h1>

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

<?php foreach ((array)Yii::$app->session->getFlash('cleaner', []) as $message):

?>

<div class="alert alert-success">

<?= $message ?>

</div>

<?php endforeach; ?>

<?php endif; ?>

<p>

<?= Html::a('Clear Caches', ['cache'], [

'class' => 'btn btn-primary',

'data' => [

'confirm' => 'Are you sure you want to clear all cache data?',  
'method ' => 'post',

],

]) ?>

<?= Html::a('Clear Assets', ['assets'],

['class' => 'btn btn-primary',

'data' => [

'confirm' => 'Are you sure you want to  
clear all temporary assets?',

'method ' => 'post',

],

]) ?>

<?= Html::a('Clear Runtime', ['runtime'],

['class' => 'btn btn-primary',

'data' => [

'confirm' => 'Are you sure you want to clear all runtime

files?',

'method ' => 'post',

],

]) ?>

</p>

</div>

1. Attach the controller to application via the controllerMap section of the config/web. php  
   configuration file:

$config = [

'id' => 'basic',

'basePath' => dirname( DIR ),

'bootstrap' => ['log'],

'controllerMap' => [

'clean' => 'app\cleaner\CleanController',

],

'components' => [

]...

];

1. Add a new item to the main menu:

echo Nav::widget([

'options' => ['class' => 'navbar-nav navbar-right'],

'items' => [

['label' => 'Home', 'url' => ['/site/index']],

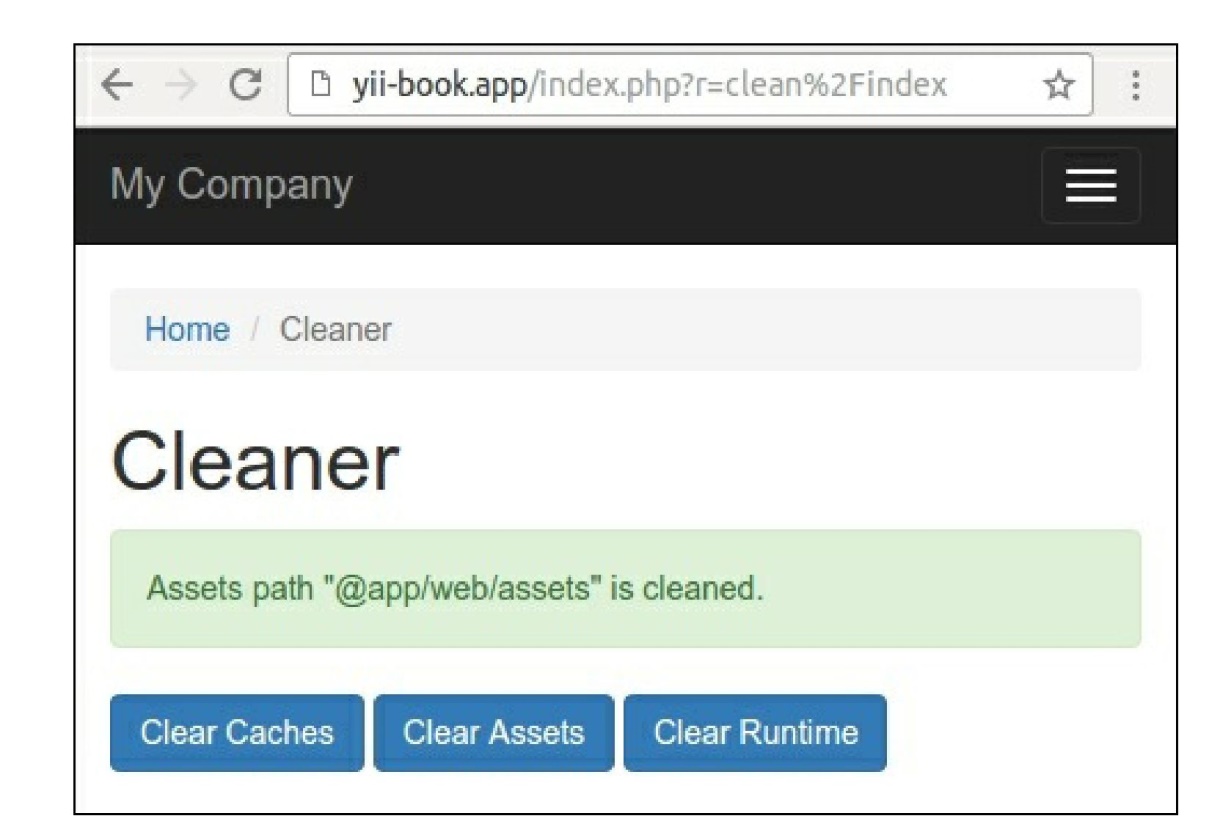
['label' => 'Cleaner', 'url' => ['/clean/index']],

['label' => 'About', 'url' => ['/site/about']],

],

]);

1. Open the controller and clear the assets:



6. In case you use the yii2-app-advanced application template, just specify the correct paths in the  
configuration:

'controllerMap' => [

'clean' => 'app\cleaner\CleanController',

'assetPaths' => [

'@backend/web/assets',

'@frontend/web/assets',

],

'runtimePaths' => [

'@backend/runtime',

'@frontend/runtime',

'@console/runtime',

],

],

Now we can attach the controller to any application.

How it works...

When you are running an application and passing a route such as clean/index, prior to executing  
cleanController: : actionIndex, Yii checks if there is controllerMap defined. Since we have a clean  
controller defined there, Yii executes it instead of going the usual way.

In the controller itself we defined the assetPaths, runtimePaths, and caches properties to be able to  
connect the controller to applications with different directory and cache structures. We are setting it when

attaching the controller.

See also

* In order to learn more about controllers and about the controllers map, refer to  
  [http://www.yiiframework.eom/doc-2.0/guide-structure-controllers.html](http://www.yiiframework.com/doc-2.0/guide-structure-controllers.html)
* The Creating reusable controllers recipe in this chapter