Creating a custom filter

Filters are objects that run before and/or after controller actions. For example, an access control filter may run before actions to ensure that they are allowed to be accessed by particular end users; a content compression filter may run after actions to compress the response content before sending them out to end users.

A filter may consist of a prefilter (filtering logic applied before actions) and/or a postfilter (logic applied after actions). Filters are essentially a special kind of behavior. Therefore, using filters is the same as using behaviors.

Let’s assume that we have a web application, which provides a user interface for working only at specified hours, for example, from 10 AM to 6 PM.

Getting ready

Create a new application using the Composer package manager, as described in the official guide at <http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.

How to do it...

1. Create a controller, @app/controllers/TestController. php, as follows: <?php

namespace app\controllers;

use app\components\CustomFilter; use yii\helpers\Html; use yii\web\Controller;

class TestController extends Controller {

public function behaviors()

{

return [

'access' => [

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

],

];

}

public function actionIndex()

{

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

'This is a test content'

));

}

}

1. Create a new filter, @app/components/CustomFilter . php, as follows:

<?php

namespace app\components;