Using controller filters

In many cases, we need to filter the incoming data or perform some actions based on the data. For example, with custom filters, we can filter visitors by IP, force users to use HTTPS, or redirect the user to an installation page prior to using the application.

In Yii2, filters are essentially a special kind of behavior, so using filters is the same as using behaviors.

Yii has a lot of built-in usable filters, which include:

• Core

• Custom

• Authentication

• Content Negotiator

• HttpCache

• PageCache

• RateLimiter

• Verb

• Cors

In this recipe, we will implement the following:

• Limiting access to the controller action to authorized users only

• Limiting access to the controller action to specified IPs

• Limiting access to specific user roles

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 app/components/AccessRule.php:

<?php

namespace app\components; use app\models\User;

class AccessRule extends \yii\filters\AccessRule {

/\* \*

\* @inheritdoc \*/

protected function matchRole($user)

{

if (empty($this->roles)) { return true;

}

$isGuest = $user->getIsGuest(); foreach ($this->roles as $role) { switch($role) { case '?':

return ($isGuest) ? true : false;