'actions' => ['auth-only'],

' roles' => ['@'],

],

[

'allow' => true,

'actions' => ['ip'],

'ips' => ['127.0.0.1'],

],

[

'allow' => true,

'actions' => ['user'],

' roles ' => ['admin'],

],

[

'allow' => true,

'actions' => ['user'],

'matchCallback' => function ($rule, $action) {

return preg\_match('/MSIE 9/',$\_SERVER['HTTP\_USER\_AGENT']) !==

false;

}

],

['allow' => false]

],

]

];

}

2. Now try to run controller actions using Internet Explorer and other browsers by using both the admin and demo usernames.

How it works...

We will start with limiting access to the controller action to authorized users only. See the following code in the rules array:

[

'allow' => true,

'actions' => ['auth-only'],

'roles' => [User::ROLE\_USER]

],

Each array here is an access rule. You can either use allow=true or allow=false for a deny rule. For each rule, there are several parameters.

By default, Yii does not deny everything, so consider adding [' allow' => false] to the end of your rules list if you need maximum security.

In our rule, we use two parameters. The first is the actions parameter, which takes an array of actions to which the rule will be applied. The second is the roles parameter, which takes an array of user roles to determine the users this rule applies to.

Yii2’s built in Access Control supports only two roles by default: guest (not logged in), represented by ?, and authenticated, represented by @.

With simple access controls, we can just limit access to specific pages or controller actions based on the