Rate limiting

To prevent abuse, you should consider adding rate limiting to your APIs. For example, you may want to limit the API usage of each user to be, at most, five API calls within a period of one minute. If too many requests are received from a user within the stated period of time, a response with the status code 429 (Too Many Requests) should be returned.

Getting ready

Repeat all the steps from the Creating a REST server recipe’s Getting ready and How to do it... sections.

1. Create a migration for creating a user allowance table with the following command:

./yii migrate/create create\_user\_allowance\_table

2. Then, update the just-created migration method, up, with the following code:

public function up()

{

$tableOptions = null;

if ($this->db->driverName === 'mysql') {

$tableOptions = 'CHARACTER SET utf8 COLLATEutf8\_general\_ci ENGINE=InnoDB';

}

$this->createTable('{{%user\_allowance}}', [

'user\_id' => $this->primaryKey(),

'allowed\_number\_requests' => $this->integer(10)->notNull(), 'last\_check\_time' => $this->integer(10)->notNull()

], $tableOptions);

}

3. Update the down methodwith the following code:

public function down()

{

$this->dropTable('{{%user\_allowance}}');

}

4. Run the created create\_film\_table migration.

5. Generate the UserAllowance model with the Gii module.

How to do it...

First, you have to update @app/controllers/FilmController. php with the following code: <?php

namespace app\controllers;

use yii\rest\ActiveController;

use yii\filters\RateLimiter;

use yii\filters\auth\QueryParamAuth;

class FilmController extends ActiveController {