public $modelClass = 'app\models\Film';

public function behaviors()

{

$behaviors = parent::behaviors();

$behaviors['authenticator'] = [

'class' => QueryParamAuth::className(), ];

$behaviors['rateLimiter'] = [

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

'enableRateLimitHeaders' => true ];

return $behaviors;

}

}

To enable rate limiting, the User model class should implement yii\filters\RateLimitInterface and requires the implementation of three methods: getRateLimit(), loadAllowance(), and saveAllowance(). You have to add them with rate\_limit\_number and rate\_limit\_reset constants:

<?php

namespace app\models;

class User extends \yii\base\Object implements \yii\web\IdentityInterface, \yii\filters\RateLimitInterface {

public $id; public $username; public $password; public $authKey; public $accessToken;

const RATE\_LIMIT\_NUMBER = 5; const RATE\_LIMIT\_RESET = 60;

// it means that user allowed only 5 requests per one minute public function getRateLimit($request, $action)

{

return [self::RATE\_LIMIT\_NUMBER,self::RATE\_LIMIT\_RESET];

}

public function loadAllowance($request, $action)

{

$userAllowance = UserAllowance::findOne($this->id); return $userAllowance ?

[$userAllowance->allowed\_number\_requests,$userAllowance->last\_check\_time] $this->getRateLimit($request, $action);

}

public function saveAllowance($request, $action,$allowance, $timestamp)

{

$userAllowance = ($allowanceModel =UserAllowance::findOne($this->id)) ? $allowanceModel : new UserAllowance();

$userAllowance->user\_id = $this->id;

$userAllowance->last\_check\_time = $timestamp; $userAllowance->allowed\_number\_requests =$allowance;

$userAllowance->save();

}