Using a base controller

In many frameworks, the concept of a base controller that is being extended by other ones is described right in the guide. In Yii, it is not in the guide, as you can achieve flexibility in many other ways. Still, using a base controller is possible and can be useful.

Let’s say we want to add some controllers that will be accessible only when the user is logged in. We can certainly set this constraint for each controller separately, but we will do it in a better way.

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-startinstallation.html>.

How to do it...

1. First, we will need a base controller that our user-only controllers will use. Let’s create @app/components/BaseController . php with the following code:

<?php

namespace app\components; use Yii;

use yii\web\Controller;

use yii\filters\AccessControl;

class BaseController extends Controller {

public function actions()

{

return [

'error' => ['class' => 'yii\web\ErrorAction'],

];

}

public function behaviors()

{

return [

'access' => [

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

'rules' => [

[

'allow' => true,

'actions' => 'error'

],

[

'allow' => true,

'roles' => ['@'],

],

],

]

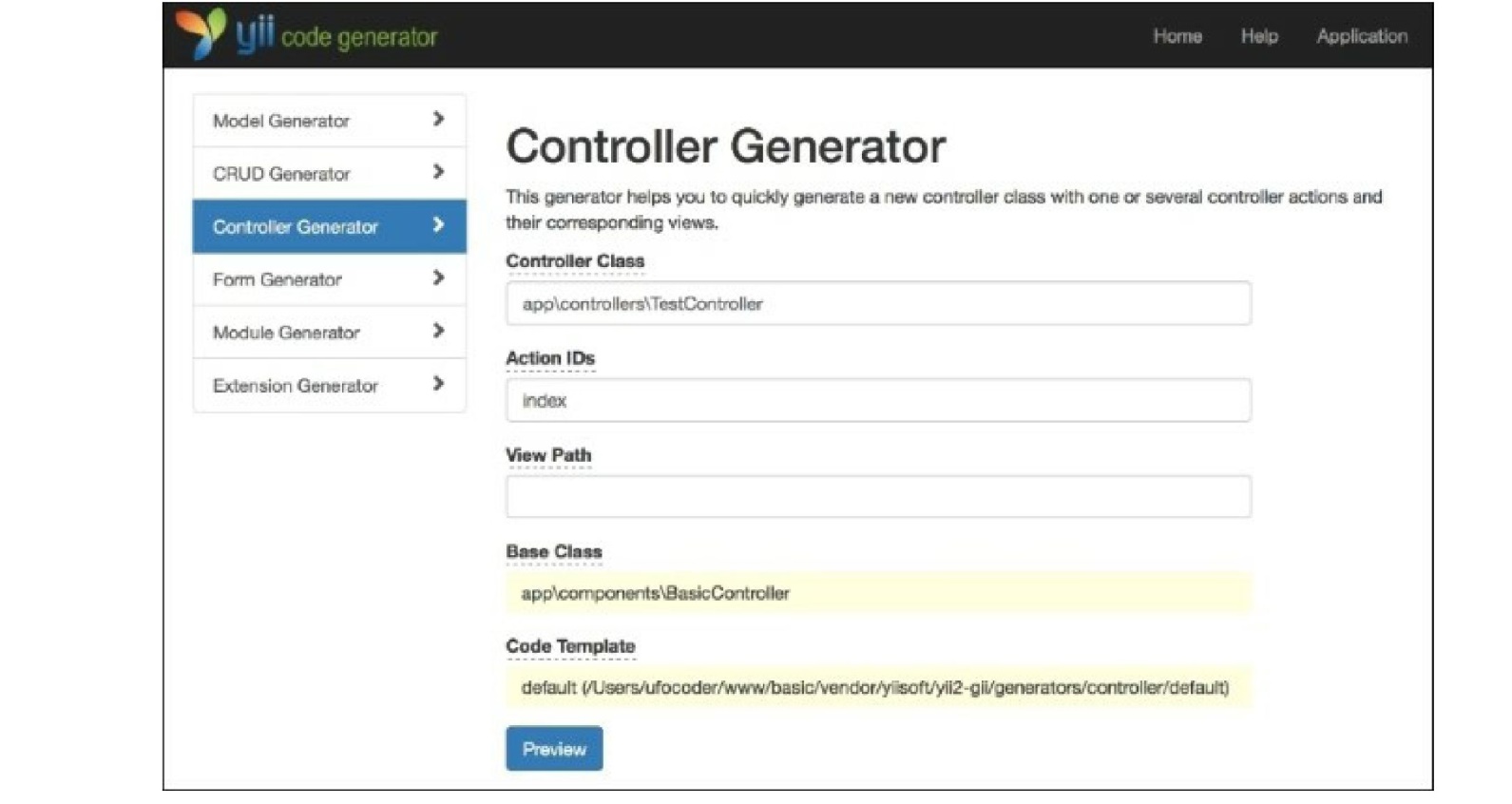
];

}

}

This controller has an action map with an error action also.

2. Now, create Testcontroller by Gii, but set the value of the base class field as app/components/BaseController:



You will get something similar to the following:

<?php

namespace app\controllers;

class Testcontroller extends \app\components\BaseController {

public function actionIndex()

{

return $this->render('index');

}

}

3. Now, your TestController will be only accessible if the user is logged in, even though we have not declared it explicitly in the TestController class. You can check it by visiting [http://yii- book](http://yii-book) . app/index . php?r=test/index while logged out.

How it works...

The trick is nothing more than a basic class inheritance. If filters or access control rules are not found in TestController, then they will be called from SecureController.

There’s more...

If you need to extend the base controller’s method, keep in the mind that it must not be overridden. For example, we need to add a page action to the controller’s action map:

<?php

namespace app\controllers;

use yii\helpers\ArrayHelper; use app\components\BaseController;

class TestController extends BaseController {

public function actions()

{

return ArrayHelper::merge(parent::actions(), [

'page' => [

'class' => 'yii\web\ViewAction',

],

]);

}

public function behaviors()

{

$behaviors = parent::behaviors();

$rules = $behaviors['access']['rules'];

$rules = ArrayHelper::merge(

$rules,

[

[

'allow' => true,

'actions' => ['page']

]

]

);

$behaviors['access']['rules'] = $rules; return $behaviors;

}

public function actionIndex()

{

return $this->render('index');

}

}

For further information, refer to [http://www.yijframework.com/doc-2.0/yii-base-controller.html](http://www.yiiframework.com/doc-2.0/yii-base-controller.html).