Displaying static pages

If you have a few static pages and aren’t going to change them very frequently, then it’s not worth querying the database and implementing page management for them.

Getting ready

Create a new application using the Composer package manager, as described in the official guide at [http://www.yiiframework.eom/doc-2.0/guide-startinstallation.html](http://www.yiiframework.com/doc-2.0/guide-startinstallation.html).

How to do it...

1. Create the test controller file, @app/controllers/TestController. php, as follows:

<?php

namespace app\controllers;

use yii\web\Controller;

class TestController extends Controller {

public function actions()

{

return [

'page' => [

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

]

];

}

}

1. Now, put your pages into views/test/pages, and name them index. php and contact. php. The content of index. php is as follows:

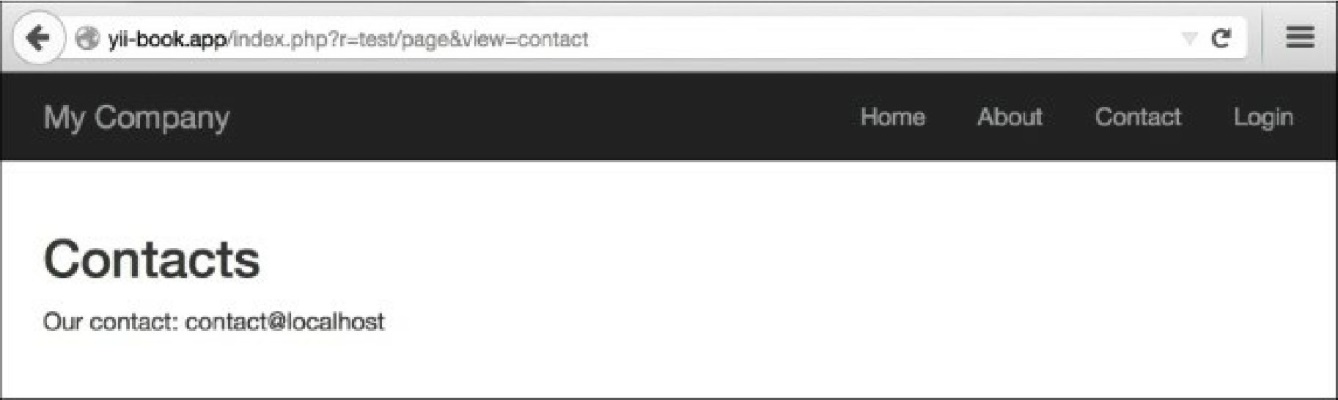
<h1>Index</h1> content of index file

Contact.php content is:

<h2>Contacts</h2>

<p>Our contact: contact@localhost</p>

1. Now you can check your pages by typing in the URL,
2. http://yii-book.app/index.php?r=test/page&view=contact:



5. Alternatively, you can type in the URL http: //yii-book. app/test/page/view/about, if you have configured clean URLs with a path format.

How it works...

We connect the external action named \yii\web\viewAction, which simply tries to find a view named the same as the $\_get parameter supplied. If it is there, it displays it. If not, then it will give you a 404 not found page. In case viewParam is not set, the defaultview value will be used.

There’s more.

**About ViewAction**

There are some useful \yii\web\ViewAction parameters we can use. These are listed in the following table:

|  |  |
| --- | --- |
| Parameter  name | Description |
| defaultView | The name of the default view when the yii\web\ViewAction : : $viewParam GET parameter is not provided by the user. Defaults to ' index'. This should be in the format of path/to/view, similar to that given in the get parameter. |
| layout | The name of the layout to be applied to the requested view. This will be assigned to yii\base\Controller: :$layout before the view is rendered. Defaults to null, meaning the controller’s layout will be used. If false, no layout will be applied. |
| viewParam | The name of the get parameter that contains the requested view name. |

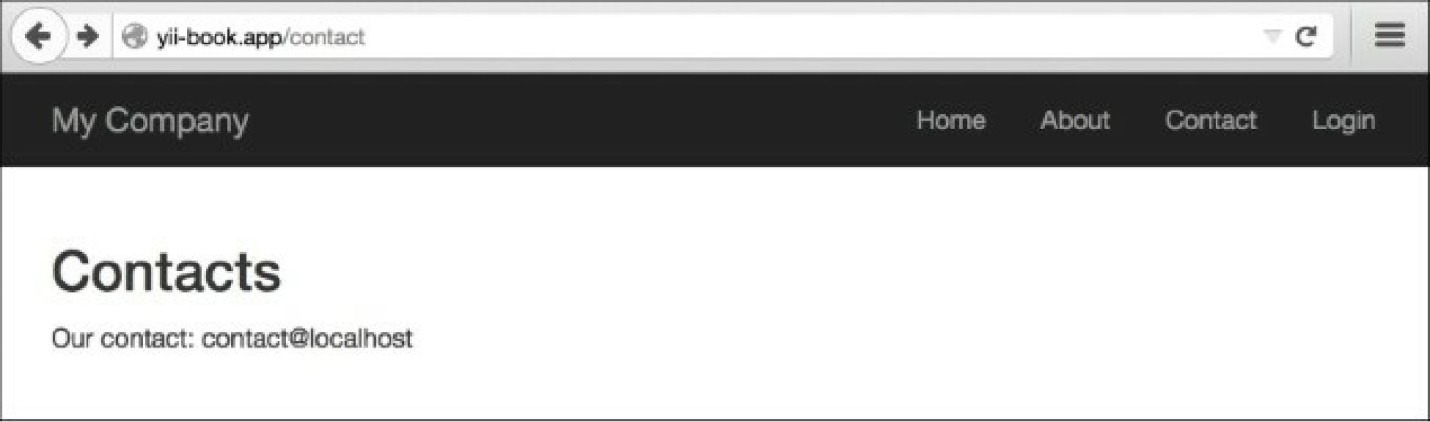
|  |  |
| --- | --- |
| viewPrefix | A string to be prefixed to the user-specified view name to form a complete view name. For example, if a user requests tutorial/chap1, the corresponding view name will be pages/tutorial/chap1, assuming that the prefix is pages. The actual view file is determined by yii\base\View: : findViewFile (). |

**Configuring URL rules**

The viewAction action provides you a way to minify your controller, but the URLs look like http: //yii- book. app/index. php?r=test/page&page=about. To make URLs short and more readable, add a URL rule to urlManager component:

'<view:about>' => 'test/page'

If the urlManager component configures properly you will get the following:



To configure the urlManager component, refer to the Configuring URL rules recipe.

See also

For further information, refer to the following URLs:

* [http://www.yiiframework.com/doc-2.0/yir-web-v'ewaction.html](http://www.yiiframework.com/doc-2.0/yii-web-viewaction.html)
* [http://www.viiframework.com/doc-2.0/guide-structure-views.hrml#rendering-static-pages](http://www.yiiframework.com/doc-2.0/guide-structure-views.html%23rendering-static-pages)
* The *Configuring URL rules* recipe

Using flash messages

When you are editing a model with a form, deleting a model, or doing any other operation, it is good to tell users if it went well or if there was an error. Typically, after some kind of action, such as editing a form, a redirect will happen and we need to display a message on the page we want to go to. However, how do we pass it from the current page to the redirect target and clean up afterwards? Flash messages will help us do this.

Getting ready

Create a new application using the Composer package manager, as described in the official guide at [http://www.viiframework.com/doc-2.0/guide-start-installation.html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html).

How to do it...

1. Let’s create a @app/controllers/TestController. php controller as follows:

<?php

namespace app\controllers; use Yii;

use yii\web\Controller;

use yii\filters\AccessControl;

class TestController extends Controller {

public function behaviors()

{

return [

'access' => [

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

'rules' => [

[

'allow' => true,

'roles ' => ['@'],

'actions' => ['user']

],

[

'allow' => true,

'roles' => ['?'],

'actions' => ['index', 'success', 'error']

],

],

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

Yii::$app->session->setFlash('error',

'This section is only for registered users.');

$this->redirect(['index']);

},

],

];

}

public function actionUser()

{

return $this->renderContent('user');

**}**

public function actionSuccess()

{

Yii::$app->session->setFlash('success', 'Everything went fine!'); $this->redirect(['index']);

}

public function actionError()

{

Yii::$app->session->setFlash('error', 'Everything went wrong!'); $this->redirect(['index']);

}

public function actionIndex()

{

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

}

}

1. Additionally, create the @app/views/common/alert. php view as follows:

<?php

use yii\bootstrap\Alert;

?>

<?php if (Yii::$app->session->hasFlash('success')):?>

<?= Alert::widget([

'options' => ['class' => 'alert-success'],

'body' => Yii::$app->session->getFlash('success'),

]);?>

<?php endif ?>

<?php if (Yii::$app->session->hasFlash('error')) :?>

<?= Alert::widget([

'options' => ['class' => 'alert-danger'],

'body' => Yii::$app->session->getFlash('error'),

]);?>

<?php endif; ?>

1. Create the @app/views/test/index. php file view as follows:

<?php

/\* @var $this yii\web\View \*/

?>

<?= $this->render('//common/alert') ?>

<h**2**>Guest page</h**2**>

<p>There's a content of guest page</p>

1. Create the @app/views/test/user. php file view as follows:

<?php

/\* @var $this yii\web\View \*/

?>

<?= $this->render('//common/alert') ?>

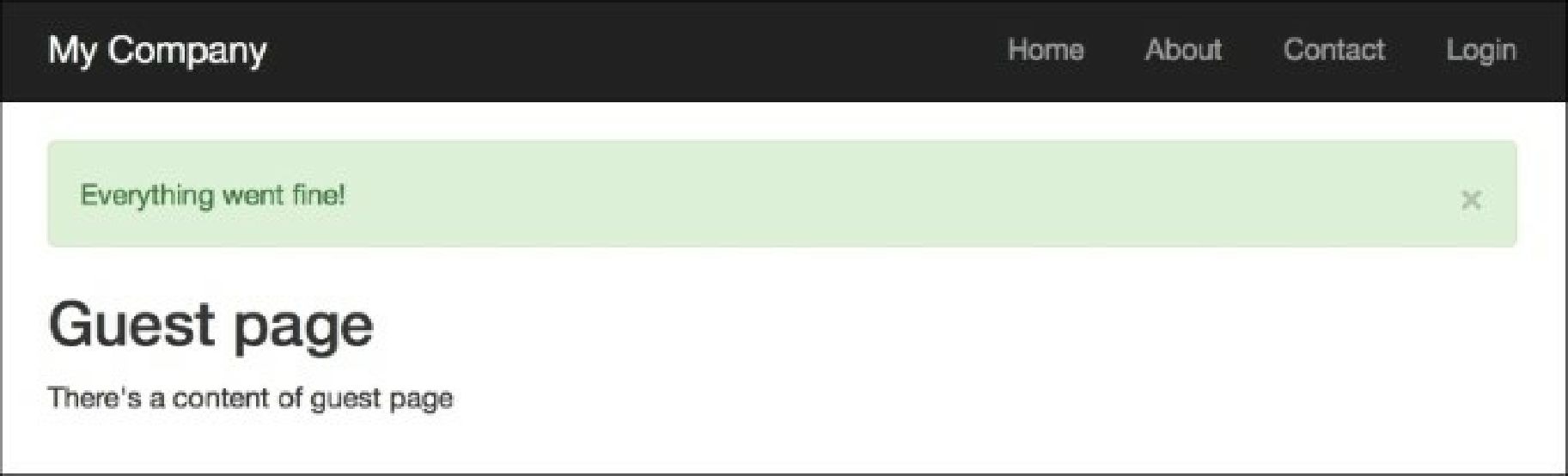
<h**2**>User page</h**2**>

<p>There's a content of user page</p>

1. Now, if you go to http: //yii-book. app/index. php?r=test/success, you will be redirected to

6.

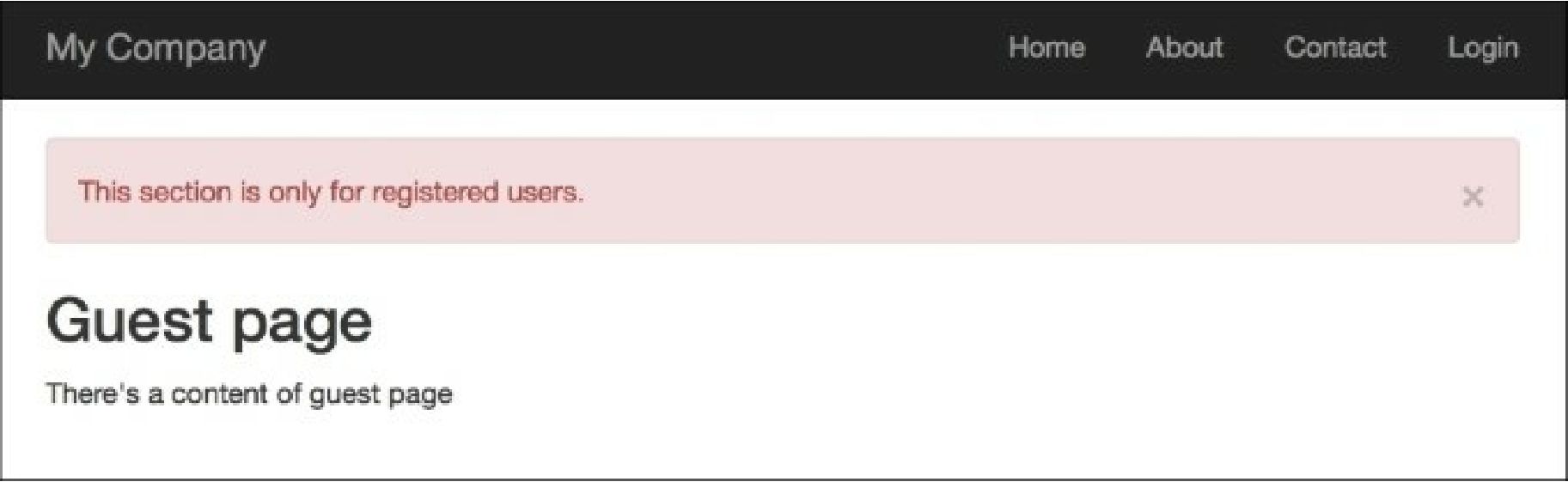
http://yii-book.app/index.php?r=test/index and a success message will be displayed as follows:



Moreover, if you go to http: //yii-book. app/index. php?r=test/error, you will be redirected to the same page, but with an error message. Refreshing the index page will hide the message:



7. Then try running http://yii-book.app/index.php?r=test/user. You will be redirected to http: //yii-book. app/index. php?r=test/index and an error message will be displayed that executed in the denyCallback function:



How it works...

We set a flash message with Yii::$app->session->('success', 'Everything went fine!'). Internally, it saves a message into a session state, so at the lowest level, our message is being kept in $\_session until Yii: :$app->session->getFlash( 'success') is called and the $\_session key is deleted.

The flash message will be automatically deleted after it is accessed in a request.

There’s more.

**The getAllFlashesO method**

Sometimes you need to handle all flashes. You can do it in a simple manner, as follows:

$flashes = Yii::$app->session->getAllFlashes();

<?php foreach ($flashes as $key => $message): ?>

<?= Alert::widget([

'options' => ['class' => 'alert-info'],

'body' => $message,

]);

?>

<?php endforeach; ?>

**The removeAHFlashes() method**

When you need to flush all your flashes, use the following:

Yii::$app->session->removeAllFlashes();

**The removeFlash() method**

When you need to remove the flash method with a specified key, use the following:

Yii::$app->session->removeFlash('success');

In this example, we added a very useful callback function, which sets up an error message and does a redirect to the test/ind ex page.

See also

For further information, refer to:

* [http://www.viiframework.com/doc-2.0/vii-web-session.html](http://www.yiiframework.com/doc-2.0/yii-web-session.html)
* <http://www.yiiframework.com/doc-2.0/yii-bootstrap-alert.html>