Создание Url-Адресов

Yii позволяет не только маршрутизировать URL-адреса к различным действиям контроллера, но и генерировать URL-адрес, указав правильный внутренний маршрут и его параметры. Это действительно полезно, потому что вы можете сосредоточиться на внутренних маршрутах при разработке приложения, и только беспокоиться о реальных URL-адресов, прежде чем идти в прямом эфире. Никогда не указывайте URL напрямую и убедитесь, что вы используете группу инструментов YII URL. Это позволит вам изменить URL-адреса без перезаписи большого количества кода приложения.

Подготовка

1. Create a new application using the Composer package manager, as described in the official guide at <http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.
2. Find your @app/config/web. php file and replace the rules array as follows:

'urlManager' => array(

'enablePrettyUrl' => true,

'showScriptName' => false,

),

1. Configure your application server to use clean URLs. If you are using Apache with mod\_rewrite and AllowOverride turned on, then you should add the following lines to the . htaccess file under your @app/web folder:

Options +FollowSymLinks IndexIgnore \*/\*

RewriteEngine on

* if a directory or a file exists, use it directly RewriteCond %{REQUEST\_FILENAME} !-f RewriteCond %{REQUEST\_FILENAME} !-d
* otherwise forward it to index.php RewriteRule . index.php

How to do it...

1. In your @app/controllers directory, create BlogController with the following code inside:

<?php

namespace app\controllers; use yii\web\Controller;

class BlogController extends Controller {

public function actionIndex()

{

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

}

public function actionRssFeed($param)

{

return $this->renderContent('This is RSS feed for our blog and ' . $param);

}

public function actionArticle($alias)

{

return $this->renderContent('This is an article with alias ' . $alias);

}

public function actionList()

{

return $this->renderContent('Blog\'s articles here');

}

public function actionHiTech()

{

return $this->renderContent('Just a test of action which contains more than one words in the name') ;

}

}

This is our blog controller that we are going to generate custom URLs for.

1. In your @app/controllers directory, create TestController with the following code inside:

<?php

namespace app\controllers; use Yii;

use yii\web\Controller;

class TestController extends Controller {

public function actionUrls()

{

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

}

}

1. In the @app/views directory, create the test directory and the urls. php view file, and place the following code inside:

<?php

use yii\helpers\Url; use yii\helpers\Html;

?>

<h1>Generating URLs</h1>

<h3>Generating a link with URL to <i>blog</i> controller and <i>article</i> action with alias as param</h3>

<?= Html::a('Link Name', ['blog/article', 'alias' => 'someAlias']); ?>

<h3>Current url</h3>

<?=Url::to(' ')?>

<h3>Current Controller, but you can specify an action</h3>

<?=Url::toRoute(['view', 'id' => 'contact']);?>

<h3>Current module, but you can specify controller and action</h3>

<?= Url::toRoute('blog/article')?>

<h3>An absolute route to blog/list </h3>

<?= Url::toRoute('/blog/list')?>

<h3> URL for <i>blog</i> controller and action <i>HiTech</i> </h3>

<?= Url::toRoute('blog/hi-tech')?>

<h3>Canonical URL for current page</h3>

<?= Url::canonical()?>

<h3>Getting a home URL</h3>

<?= Url::home()?>

<h3>Saving a URL of the current page and getting it for re-use</h3>

<?php Url::remember()?>

<?=Url::previous()?>

<h3>Creating URL to <i>blog</i> controller and <i>rss-feed</i> action while URL helper isn't available</h3>

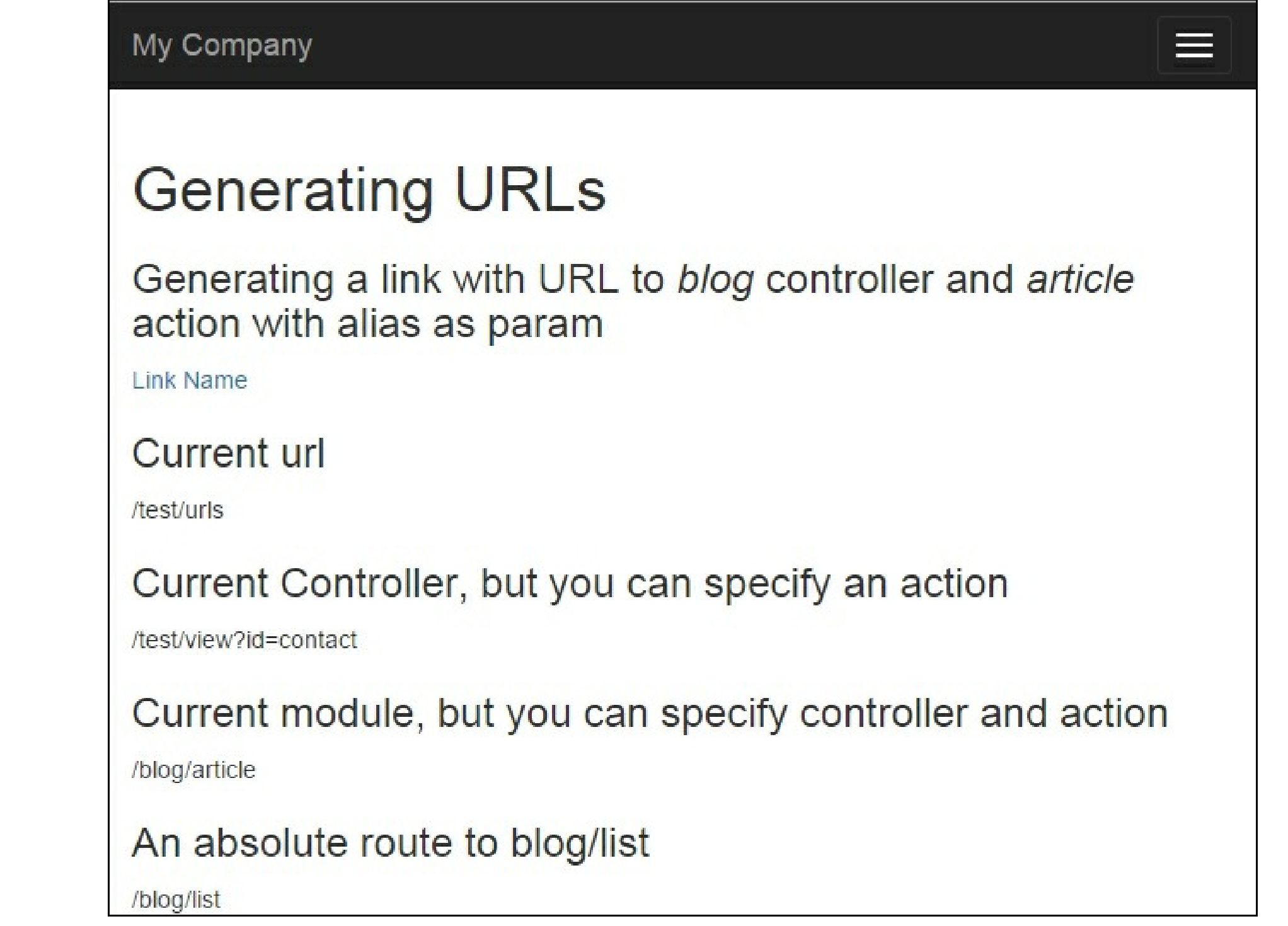
<?=Yii::$app->urlManager->createUrl(['blog/rss-feed', 'param' => 'someParam'])?>

<h3>Creating an absolute URL to <i>blog</i> controller and <i>rss-feed</i></h3> <p>It's very useful for emails and console applications</p>

<?=Yii::$app->urlManager->createAbsoluteUrl(['blog/rss-feed', 'param' =>

' someParam'])?>

4. Go to the URL http: //yii-book.app/test/urls and you will see the output. (Refer to the full list of methods in the preceding code.):



How it works...

We need to generate URLs pointing to the controller actions (RssFeed, Article, List, HiTech) of BlogController.

Depending on where we need it, there are different ways of doing it, but the basics are the same. Let’s list some methods that generate URLs.

What is an internal route? Each controller and its actions have corresponding routes. A format for a route is moduleID/controllerID/actionID. For example, the actionHiTech method of BlogController corresponds to the blog/hi-tech route.

To get a controller ID, you should take its name without the Controller postfix and make its first letter lowercase. To get an action ID, you should take the action method name without the action prefix and make the first letter in each word lowercase, and separate them with a dash (-) sign (for example, actionHiTech will be hi-tech).

The $\_get variables are the parameters that will be passed to an action with an internal route specified. For example, if we want to create a URL to a BlogController: :action article that passes the $\_get[ ' name' ] parameter to it, it can be done as follows:

<?= Html::a('Link Name', ['blog/article', 'alias' => 'someAlias']); ?>

Relative URLs can be used inside your application, while absolute ones should be used for pointing to locations outside your website (such as other websites) or for linking to resources meant to be accessed from outside (RSS feeds, e-mails, and so on).

You can do it easily with the URL manager. The URL manager is a built-in application component named urlManager. You have to use this component, which is accessible from both web and console applications via Yii::$app->urlManager.

When you cannot get a controller instance, for example, when you implement a console application, you can use the two following urlManager creation methods:

<?=Yii::$app->urlManager->createUrl(['blog/rss-feed', 'param' => 'someParam'])?> <?=Yii::$app->urlManager->createAbsoluteUrl(['blog/rss-feed', 'param' => 'someParam'])?>

There’s more.

For further information, refer to the following URLs:

* [https://en.wikipedia.org/wiki/Canonical link element](https://en.wikipedia.org/wiki/Canonical_link_element)
* <http://www.yiiframework.com/doc-2.0/guide-structure-controllers.html>
* <http://www.yiiframework.com/doc-2.0/guide-runtime-routing.html>
* [http://www.viiframework.com/doc-2.0/guide-helper-url.html](http://www.yiiframework.com/doc-2.0/guide-helper-url.html)
* <http://www.yiiframework.com/doc-2.0/yii-web-urlmanager.html>

See also

• The Configuring URL rules recipe