Using blocks

One of the Yii features you can use in your views is blocks. The basic idea is that you can record some output and then reuse it later in a view. A good example would be to define additional content regions for your layout and filling them elsewhere.

In the previous version, Yii 1.1, blocks were called clips.

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-start-installation.html>.

How to do it...

1. For our example, we need to define two regions in our layout—beforeContent and footer.
2. Open @app/views/layouts/main. php and insert the following code line just before the content output:

<?php if(!empty($this->blocks['beforeContent'])) echo $this- >blocks['beforeContent']; ?>

1. Then, replace the footer code with the following code:

<footer class="footer">

<div class="container">

<?php if (!empty($this->blocks['footer'])): echo $this->blocks['footer'] ?>

<?php else: ?>

<p class="pull-left">&copy; My Company <?= date('Y') ?></p>

<p class="pull-right"><?= Yii::powered() ?></p>

<?php endif; ?>

</div>

</footer>

1. That is it! Then, add a new action to controllers/SiteController. php, named blocks:

public function actionBlocks()

{

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

}

1. Now, create a view file, views/site/blocks. php, with the following content:

<?php

use \yii\Helpers\Html;

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

?>

<?php $this->beginBlock('beforeContent');

echo Html::tag('pre', 'Your IP is ' . Yii::$app->request->userIP);

$this->endBlock(); ?>

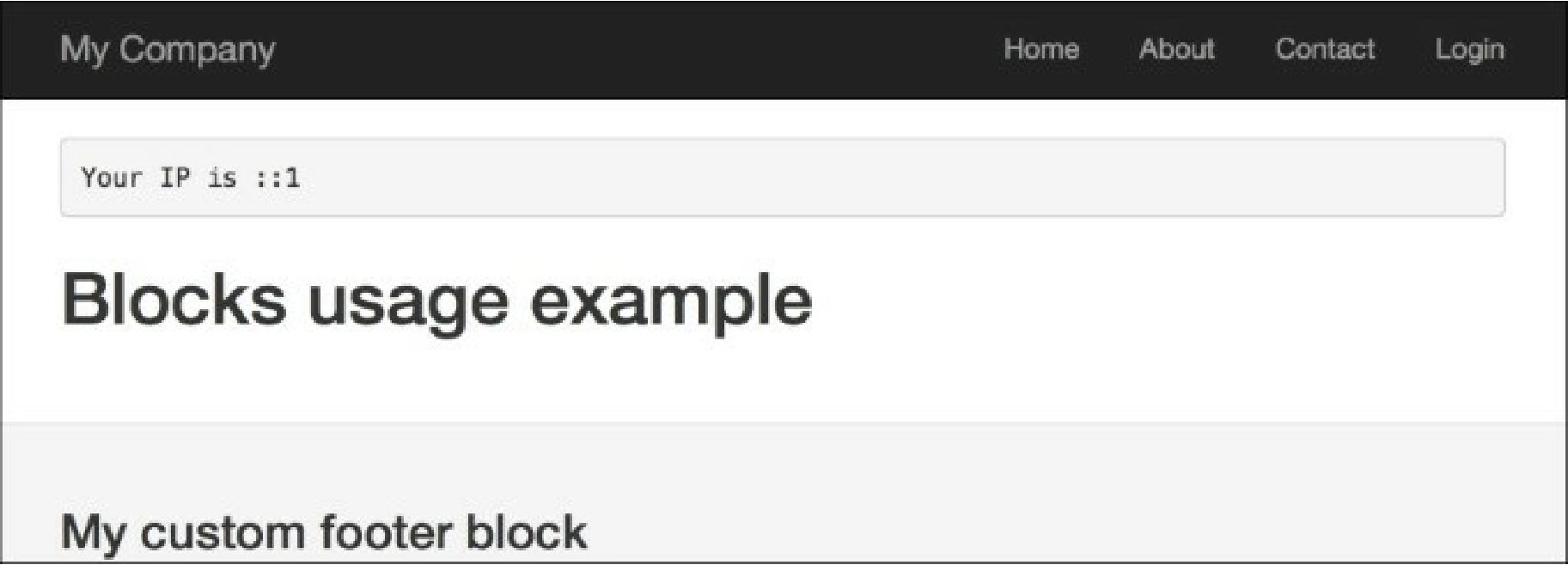
<?php $this->beginBlock('footer');

echo Html::tag('h3', 'My custom footer block');

$this->endBlock(); ?>

<h1>Blocks usage example</h1>

6. Now, when you open your /index. php?r=site/blocks page, you should get your IP just before the page content and a built-with note in the footer:



How it works...

We mark regions with the code that just checks for the existence of a specific block, and if the block exists, the code outputs it. Then, we record content for blocks we defined using the special controller methods named beginBlock and endBlock.

From controller, you can easily access our block’s variables via $this->view->blocks[' blockID' ].

There’s more.

* The Using the controller context in a view recipe
* [http://www.yiiframework.com/doc-2.0/guide-structure-views.html#using-blocks](http://www.yiiframework.com/doc-2.0/guide-structure-views.html%23using-blocks)