Displaying custom errors

In Yii, the error handling is very flexible, so you can create your own error handler for errors of a specific  
type. In this recipe, we will handle a 404 not found error in a smart way. We will show a custom 404 page  
that will suggest the content based on what was entered in the address bar.

Getting ready

1. Create a new yii2 -app-basic application by using the Composer package manager, as described  
   in the official guide at [http://www. yiiframework. c om/doc-2.0/guide -start-installation .html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html).
2. Add the fail action to your siteController:

class SiteController extends Controller  
{

// ...

public function actionFail()

{

throw new ServerErrorHttpException('Error message example.');

}

}

1. Add the web/.htaccess file with the following content:

RewriteEngine on

RewriteCond %{REQUEST\_FILENAME} !-f  
RewriteCond %{REQUEST\_FILENAME} !-d  
RewriteRule . index.php

1. Configure pretty URLs for the urlManager component in your config/web. php file:

'components' => [

// .

'urlManager' => [

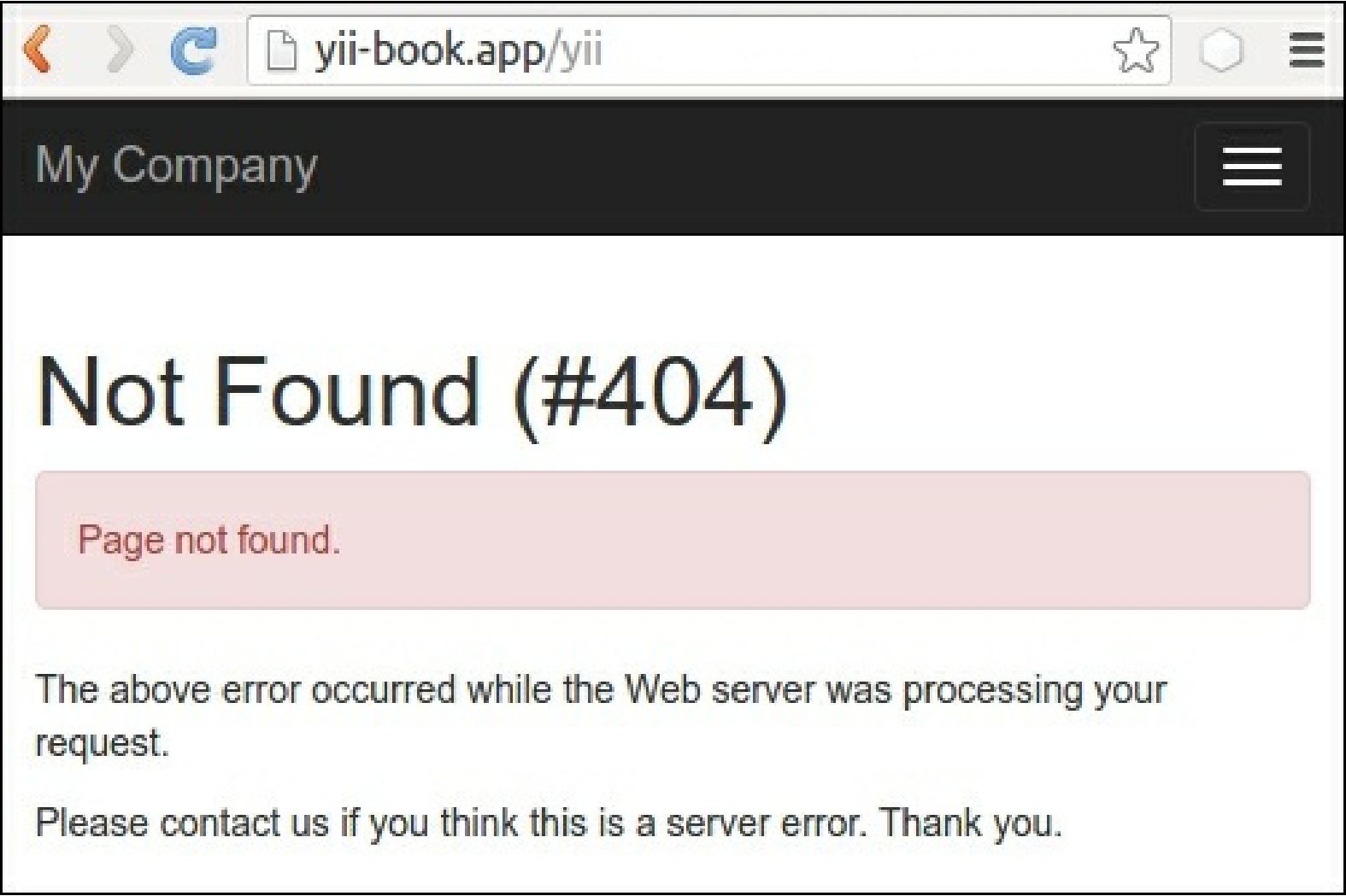
'enablePrettyUrl' => true,

'showScriptName' => false,

],

],

1. Check that framework displays the Not found exception for URLs that are not existing:



6. Also, check that the framework displays the Internal Server Error exception for our  
actionFail:



7. Now we want to create a custom page for the Not Found page. Let’s start it.

How to do it...

Now we need to change the Not Found page content, but leave it as it is for other error types. In order to  
achieve this, follow these steps:

1. Open the siteController class and look for the actions() method:

class SiteController extends Controller  
{

// ...

public function actions()

{

return [

'error' => [

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

],

'captcha' => [

'class' => 'yii\captcha\CaptchaAction',

'fixedVerifyCode' => YII\_ENV\_TEST ? 'testme' : null,

],

];

}

// ...

}

1. Remove the default error section and leave actions() as follows:

class SiteController extends Controller  
{

// ...

public function actions()

{

return [

'captcha' => [

'class' => 'yii\captcha\CaptchaAction',

'fixedVerifyCode' => YII\_ENV\_TEST ? 'testme' : null,

],

];

}

// ...

}

1. Add the own actionError() method:

class SiteController extends Controller  
{

// ...

public function actionError()

{

}

}

1. Open the original \yii\web\ErrorAction class and copy its action content into our actionError()  
   and customize it for the render custom error-404 view for the Not Found error with the 404 code:

// ...

use yii\base\Exception;

use yii\base\UserException;

class SiteController extends Controller  
{

// ...

public function actionError()

{

if (($exception = Yii::$app->getErrorHandler()->exception)== null) {

$exception = new HttpException(404, Yii::t('yii', 'Page not found.'));

}

if ($exception instanceof HttpException) {

$code = $exception->statusCode;

} else {

$code = $exception->getCode();

}

if ($exception instanceof Exception) {

$name = $exception->getName();

} else {

$name = Yii::t('yii', 'Error');

}

if ($code) {

$name .= " (#$code)";

}

if ($exception instanceof UserException) {

$message = $exception->getMessage();

} else {

$message = Yii::t('yii', 'An internal server error occurred.');

}

if (Yii::$app->getRequest()->getIsAjax()) {  
return "$name: $message";

} else {

if ($code == 404) {

return $this->render('error-404');

} else {

return $this->render('error', [

'name' => $name,

'message' => $message,

'exception' => $exception,

]);

}

}

}

}

5. Add the views/site/error-404.php view file with a custom message:

<?php

use yii\helpers\Html;

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

$this->title = 'Not Found!'

?>

<div class="site-error-404">

<h1>Oops!</h1>

<p>Sorry, but requested page not found.</p>

<p>

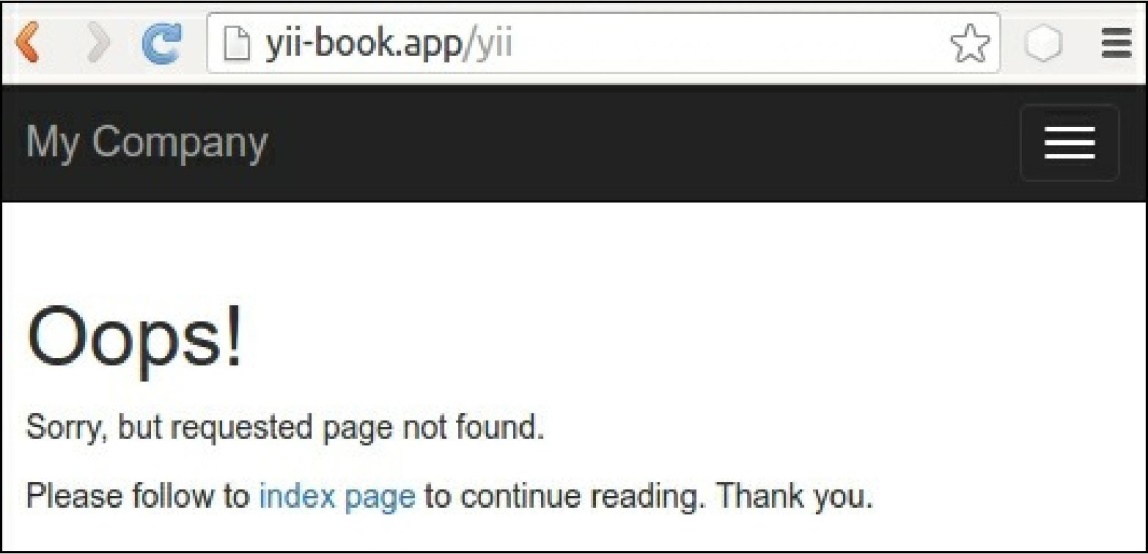
Please follow to <?= Html::a('index page', ['site/index'])?>

to continue reading. Thank you.

</p>

</div>

6. That is it. Now try to follow to the non-existing URL and see our content from the error-404.php  
view:



7. However, for a fail action we must see the default content from the error.php file:



How it works...

By default, in the yii2-app-basic application, we configure errorAction for the errorHandler  
component in our configuration file config/web .php as site/error. It means that the framework will use  
this route for displaying every handled exception:

'components' => [

'errorHandler' => [

'errorAction' => 'site/error',

],

],

In the SiteController class, we use the built-in standalone yii\web\ErrorAction class, which renders  
the so-called error. php view:

class SiteController extends Controller  
{

// ...

public function actions()

{

return [

'error' => [

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

],

'captcha' => [

'class' => 'yii\captcha\CaptchaAction',

'fixedVerifyCode' => YII\_ENV\_TEST ? 'testme' : null,

],

];

}

// ...

}

If we want to override its implementation, we can replace it in an inline actionError() method with our  
own custom content.

In this recipe, we add our own if statement for rendering a specific view on the base of error code:

if ($code == 404) {

return $this->render('error-404');

} else {

return $this->render('error', [

'name' => $name,

'message' => $message,

'exception' => $exception,

]);

}

Also, we can use a custom design for the Not Found page.

See also

[In order to learn more about handling errors in Yii, refer to http://www.yiiframework.com/doc-2.0/guide-](http://www.yiiframework.com/doc-2.0/guide-runtime-handling-errors.html)  
runtime-handling-errors.html.