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.