Ведение журнала и использование информации контекста

Иногда сообщения журнала недостаточно для исправления ошибки. Например, если вы следуете рекомендациям и разрабатываете и тестируете приложение со всеми возможными ошибками, вы можете получить сообщение об ошибке. Однако без контекста выполнения это только говорит вам, что была ошибка, и не ясно, что на самом деле вызвало ее.

Для нашего примера мы будем использовать очень простое и плохо закодированное действие, которое просто перекликается с Hello, <username>! если имя пользователя берется непосредственно из $\_get.

Подготовка

Создайте новое yii2-app-basic приложение с помощью диспетчера пакетов Composer, как описано в официальном руководстве по адресу  
<http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.   
По русски <http://yiiframework.domain-na.me/doc/guide/2.0/ru/start-installation>

Как это сделать...

Выполните следующие действия:

1. Во-первых, нам понадобится контроллер для работы. Таким образом, создайте protected/controllers/LogController.php следующим образом:

<?php

namespace app\controllers;

use yii\web\Controller;

class LogController extends Controller  
{

public function actionIndex()

{

return 'Hello, ' . $\_GET['username'];

}

}

1. Теперь, если мы запустим действие index, мы получим сообщение об ошибке, Undefined index: username. Давайте сконфигурируем регистратор для записи такого рода ошибок в файл:

config/web.php:

'components'=>array(

'log' => [

'targets' => [

[

'class' => 'yii\log\FileTarget',

'levels' => ['error'],

'logFile' => '@runtime/logs/errors.log',

],

],

],

],

1. Снова запустите действие index и проверьте runtime/logs/errors.log. В журнале должна быть следующая информация:

2016-03-06 09:27:09 [127.0.0.1][-][-][error][yii\base\ErrorException:8] exception  
'yii\base\ErrorException' with message 'Undefined index: username' in  
/controllers/LogController.php:11  
Stack trace:

#0 /yii2/base/InlineAction.php(55): ::call\_user\_func\_array()

#1 /yii2/base/Controller.php(151): yii\base\InlineAction->runWithParams()

#2 /yii2/base/Module.php(455): yii\base\Controller->runAction()

#3 /yii2/web/Application.php(84): yii\base\Module->runAction()

#4 /yii2/base/Application.php(375): yii\web\Application->handleRequest()

#5 /web/index.php(12): yii\base\Application->run()

#6 {main}

2016-03-06 09:27:09 [127.0.0.1][-][-][info][application] $\_GET = [

'r' => 'log/index'

]

$\_COOKIE = [

'\_csrf' => 'ca689043348e...a69ea:2:{i:0;s:...\"DSS...KJ\";}'

' PHPSESSID' => '30584oqhat4ek8b0hrqsapsbf4'

]

$\_SERVER = [

'USER' => 'www-data'

'HOME' => '/var/www'

'FCGI\_ROLE' => 'RESPONDER'

'QUERY\_STRING' => 'r=log/index'

'PHP\_SELF' => '/index.php'

'REQUEST\_TIME\_FLOAT' => 1459934829.3067  
'REQUEST\_TIME' => 1459934829

]

1. Теперь мы можем передать наше приложение команде тестирования и время от времени проверять журнал ошибок. По умолчанию, отчет об ошибках журнала содержат значения из всего массива $\_get, $\_post, где, переменную $\_files, $их помощью $\_SESSION и $\_server для переменных. Если не требуется отображать все значения, можно указать список настраиваемых переменных:

'log' => [

'targets' => [

[

'class' => 'yii\log\FileTarget',

'levels' => ['error'],

'logVars' => ['\_GET', '\_POST'],

'logFile' => '@runtime/logs/errors.log',

],

],

],

1. В этом случае отчет будет содержать только массивы $\_GET и $\_POST:

2016-04-06 09:49:08 [127.0.0.1][-][-][info][application] $\_GET = [ 'r' =>

'log/index'

]

Как это работает...

Yii добавляет полную информацию о контексте выполнения и среде в случае регистрации сообщений об ошибках. Если мы регистрируем сообщение вручную, то мы, вероятно, знаем, какая информация нам нужна, поэтому мы можем установить некоторые целевые параметры, чтобы писать только то, что нам действительно нужно:

'log' => [

'targets' => [

[

'class' => 'yii\log\FileTarget',

'levels' => ['error'],

'logVars' => ['\_GET', '\_POST'],

'logFile' => '@runtime/logs/errors.log',

],

],

],

В предыдущем коде ошибки записываются в файл с именем errors. В дополнение к самому сообщению, оно будет регистрировать содержимое переменных $\_GET или $\_POST, если они не пусты.

Смотрите так же

* Чтобы узнать больше о фильтрах журнала и информации о контексте, обратитесь к  
  <https://www.yiiframework.com/doc/guide/2.0/en/runtime-logging>

По русски <https://www.yiiframework.com/doc/guide/2.0/ru/runtime-logging>

* Рецепт использования различных маршрутов журнала