Logging and using the context information

Sometimes a log message is not enough to fix an error. For example, if you are following best practices  
and developing and testing an application with all possible errors reported, you can get an error message.  
However, without the execution context, it is only telling you that there was an error and it is not clear  
what actually caused it.

For our example, we will use a very simple and poorly coded action that just echoes Hello, <username>!  
where the username is taken directly from $\_get.

Getting ready

Create a new yii2 -app-basic application by using the Composer package manager, as described in the  
official guide at [http://www.viiframework.com/doc-2.0/gurdestart-installation.html](http://www.yiiframework.com/doc-2.0/guidestart-installation.html).

How to do it...

Carry out the following steps:

1. First, we will need a controller to work with. Therefore, create  
   protected/controllers/LogController . php as follows:

<?php

namespace app\controllers;

use yii\web\Controller;

class LogController extends Controller  
{

public function actionIndex()

{

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

}

}

1. Now, if we run the index action, we will get the error message, Undefined index: username.  
   Let’s configure the logger to write this kind of error to a file:

config/web.php:

'components'=>array(

'log' => [

'targets' => [

[

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

'levels' => ['error'],

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

],

],

],

],

1. Run the index action again and check runtime/logs/errors. log. There should be log information

like the following:

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. Now we can give our application to a testing team and check the errors log from time to time. By  
   default, error report log contain values from all the $\_get, $\_post, $\_files, $\_cookie, $\_session,  
   and $\_server variables. If you do not want to display all values, you can specify a custom variable  
   list:

'log' => [

'targets' => [

[

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

'levels' => ['error'],

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

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

],

],

],

1. In this case, the report will contain only the $\_get and $\_post arrays:

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

'log/index'

]

How it works...

Yii adds complete information about the execution context and environment in the case of logging error  
messages. If we are logging a message manually, then we probably know what information we need, so

we can set some target options to write only what we really need:

'log' => [

'targets' => [

[

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

'levels' => ['error'],

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

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

],

],

],

The preceding code will log errors to a file named errors. Additionally to a message itself, it will log  
contents of the $\_get or $\_post variables if they are not empty.

See also

* In order to learn more about log filters and context information, refer to  
  [http://www.yiiframework.com/doc-2.0/guide-rnntime-logging.html](http://www.yiiframework.com/doc-2.0/guide-runtime-logging.html)
* The Using different log routes recipe