Using different log routes

Logging is the key to understanding what your application actually does when you have no chance to debug it. Believe it or not, even if you are 100% sure that the application will behave as expected, in production, it can do many things you were not aware of. This is OK, as no one can be aware of everything. Therefore, if we are expecting unusual behavior, we need to know about it as soon as possible and have enough details to reproduce it. This is where logging comes in handy.

Yii allows a developer not only to log messages but also to handle them differently depending on the message level and category. You can, for example, write a message to the database, send an e-mail, or just show it in the browser.

In this recipe, we will handle log messages in a wise manner: the most important message will be sent through an e-mail, less important messages will be saved in files A and B, and the profiling will be routed to Firebug. Additionally, in a development mode, all messages and profiling information will be displayed on the screen.

Getting ready

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

How to do it...

Carry out the following steps:

1. Configure logging using config/web .php:

'components' => [

'log' => [

'traceLevel' => 0,

'targets' => [

[

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

'categories' => ['example'],

'levels' => ['error'],

'message' => [

'from' => ['log@example.com'],

'to' => ['developer1@example.com', 'developer2@example.com'], 'subject' => 'Log message',

],

],

[

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

'levels' => ['error'],

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

],

[

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

'levels' => ['warning'],

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

],

[