Implementing and executing cron jobs

Sometimes, an application requires some background tasks, such as regenerating a site map or refreshing statistics. A common way to implement this is by using cron jobs. When using Yii, there is a way to use a command to run as a job.

In this recipe, we will see how to implement both. For our recipe, we will implement writing the current timestamp into a t imestamp. txt file under the protected directory.

Getting ready

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

How to do it...

**Running the Hello command**

Let us try to run app\commands\HelloController : : actionIndex as a shell command:

<?php

namespace app\commands; use yii\console\Controller;

/\*\*

\* This command echoes the first argument that you have entered.

\*/

class HelloController extends Controller {

/\*\*

\* This command echoes what you have entered as the message.

\* @param string $message the message to be echoed.

\*/

public function actionIndex($message = 'hello world')

{

echo $message . "\n";

}

}

1. Open the shell in your application directory and execute this command:

php yii

Alternatively, you also can call the following and ensure that the shell works:

. /yii

2. Type the following command for the display hello:

./yii help hello

3. The framework must display some information:

DESCRIPTION

This command echoes what you have entered as the message.