Profiling an application with Yii

If all of the best practices for deploying a Yii application are applied and you still do not have the performance you want, then most probably there are some bottlenecks with the application itself. The main principle while dealing with these bottlenecks is that you should never assume anything and always test and profile the code before trying to optimize it.

In this recipe, we will try to find bottlenecks in the Yii2 mini application.

Getting ready

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

1. Set up your database connection and apply the following migration:

<?php

use yii\db\Migration;

class m160308\_093233\_create\_example\_tables extends Migration {

public function up()

{

$tableOptions = null;

if ($this->db->driverName === 'mysql') {

$tableOptions = 'CHARACTER SET utf8 COLLATE utf8\_general\_ci ENGINE=InnoDB';

}

$this->createTable('{{%category}} ', [

'id' => $this->primaryKey(),

'name' => $this->string()->notNull(),

], $tableOptions);

$this->createTable('{{%article}}', [

'id' => $this->primaryKey(),

'category\_id' => $this->integer()->notNull(),

'title' => $this->string()->notNull(),

'text' => $this->text()->notNull(),

], $tableOptions);

$this->createIndex('idx-article-category\_id', '{{%article}}',

' category\_id');

$this->addForeignKey('fk-article-category\_id', '{{%article}}',

'category\_id', '{{%category}}', 'id');

}

public function down()

{

$this->dropTable('{{%article}}');

$this->dropTable('{{%category}}');

}

}

2. Generate models for each table in Yii.

3. Write the following console command: