Следуя передовой практике

В этом рецепте вы увидите, как настроить Yii2 для лучшей производительности и некоторые дополнительные принципы построения адаптивных приложений. Эти принципы являются как общими, так и связанными с Yii. Поэтому, мы будем таблицей для того чтобы приложить некоторые из этих даже без использования Yii2.

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...

1. Обновить PHP до последней стабильной версии. Основные выпуски PHP могут принести значительные улучшения производительности. Выключите режим отладки и установите среду prod. Это можно сделать путем редактирования web/index.php следующим образом:

defined('YII\_DEBUG') or define('YII\_DEBUG', false);  
defined('YII\_ENV') or define('YII\_ENV', 'prod');

Замечание: В скелете приложения yii2-app-advanced можно использовать команду оболочки php init и производственную среду opt для загрузки оптимизированного index.php и конфигурационных файлов.

1. Включить компонент кэша:

'components' => [

'cache' => [

'class' => 'yii\caching\FileCache',

],

],

Вы можете использовать любой кэш-памяти вместо FileCache. Кроме того, вы можете зарегистрировать несколько компонентов приложения кэша и использовать Yii::$app ->cache и Yii::$app ->cache2 для различных типов данных:

'components' => [

'cache' => [

'class' => 'yii\caching\MemCache',

'useMemcached' => true,

],

'cache2' => [

'class' => 'yii\caching\FileCache',

],

],

Фреймворк использует компонент кэша по умолчанию в своих собственных классах.

1. Включить кэширование схемы таблицы для компонента БД следующим образом:

return [

// ...

'components' => [

// ...

'cache' => [

'class' => 'yii\caching\FileCache',

],

'db' => [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=localhost;dbname=mydatabase',

'username' => 'root',

'password' => '',

'enableSchemaCache' => true,

// Optional. Default value is 3600 seconds  
schemaCacheDuration' => 3600,

// Optional. Default value is 'cache'

'schemaCache' => 'cache',

],

],

];

1. Используйте простые массивы вместо активных объектов записей для перечисления наборов элементов:

$categoriesArray = Categories::find()->asArray()->all();

1. Используйте each () вместо all () в foreach для большого количества результатов:

foreach (Post::find()->each() as $post) {

// ...

}

1. Поскольку autoloader Composer используется для включения большинства файлов сторонних классов, рекомендуется оптимизировать его, выполнив следующую команду:

composer dump-autoload -o

Как это работает...

Если YII\_DEBUG имеет значение false, Yii отключает ведение журнала на уровне трассировки и использует меньше кода обработки ошибок. Кроме того, при установке YII\_ENV в prod приложение не загружает модули панели Yii и отладки.

Установка schemaCachingDuration в несколько секунд позволяет кэшировать схему базы данных, используемую активной записью Yii. Это настоятельно рекомендуется для рабочих серверов и значительно повышает производительность активных записей. Чтобы он работал, необходимо правильно настроить компонент кэша следующим образом:

'cache' => [

'class' => 'yii\cache\FileCache',

],

Включение кэша также положительно влияет на другие компоненты Yii. Например, маршрутизатор Yii или urlManager начинает кэшировать маршруты.

Конечно, можно попасть в ситуацию, когда предыдущие настройки не помогут достичь достаточного уровня производительности. В большинстве случаев это означает, что либо само приложение является узким местом, либо вам нужно больше оборудования.

* **Производительность на стороне сервера-это только часть общей картины**: производительность на стороне сервера-это только одна из вещей, которые влияют на общую производительность. Благодаря оптимизации клиентской части, такой как обслуживание CSS, изображений и файлов JavaScript, правильное кэширование и минимизация количества HTTP-запросов может дать хороший визуальный прирост производительности даже без оптимизации кода PHP.
* **Вещи, которые нужно сделать без использования Yii**: некоторые вещи лучше всего делать без Yii. Например, изменение размера изображения на лету лучше в отдельном PHP-скрипте, чтобы избежать дополнительных затрат.
* **Active Record versus Query Builder и SQL**: использование Построителя запросов или SQL в критически важных для производительности частях приложения. Как правило, AR наиболее полезен при добавлении и редактировании записей, поскольку он добавляет удобный слой проверки и менее полезен при выборе записей.
* **Всегда сначала проверяйте медленные запросы**: база данных может стать узким местом за секунду, если Разработчик случайно забудет добавить индекс в таблицу, которая читается часто или наоборот, или добавит слишком много индексов в таблицу, в которую мы пишем очень часто. То же самое касается выбора ненужных данных и ненужных объединений.
* **Кэширование или сохранение результатов тяжелых процессов**: если вы можете избежать запуска тяжелого процесса при каждой загрузке страницы, лучше сделать это. Например, рекомендуется сохранить или кэшировать результаты разбора текста markdown, очистить его (это очень ресурсоемкий процесс) один раз, а затем использовать готовый к отображению HTML.
* Обработка слишком много обработки: иногда слишком много обработки, чтобы быть обработаны немедленно. Это может быть построение сложных отчетов или просто отправка электронной почты (если ваш проект сильно загружен). В этом случае лучше поставить его в очередь и обработать позже с помощью cron или других специализированных инструментов.

Смотрите так же

Дополнительные сведения о настройке производительности и кэшировании см. по следующим URL-адресам:

* [http://www.yiiframework.eom/doc-2.0/guide-tutorial-performance-tnning.html](http://www.yiiframework.com/doc-2.0/guide-tutorial-performance-tuning.html)
  1. по русски https://www.yiiframework.com/doc/guide/2.0/ru/tutorial-performance-tuning
* [http://www.yiiframework.com/doc-2.0/guide-cachjng-overv'ew.html](http://www.yiiframework.com/doc-2.0/guide-caching-overview.html)
  1. по русски https://www.yiiframework.com/doc/guide/2.0/ru/caching-overview