

## Кэширование с помощью фреймворка Yii2

*Шайдуров Александр Александрович*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

*Клинский Станислав Дмитриевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

*Глаголев Владимир Александрович*

*Приамурский государственный университет имени Шолом-Алейхема*

*к.г.н., доцент кафедры информационных систем, математики и правовой информатики*

### **Аннотация**

В данной работе продемонстрирован метод кэширования данных. Описана реализация функции Memcache в фреймворке Yii2. Также в работе описаны исследования по использованию фреймворка Yii2.

**Ключевые слова:** Кэширование, Memcache, Yii2.

## Caching with the Yii2 Framework

*Shaidurov Aleksandr Aleksandrovich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Klinsky Stanislav Dmitrievich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Glagolev Vladimir Aleksandrovich*

*Sholom-Aleichem Priamursky State University*

*candidate of geographical Sciences, Associate Professor of the Department of Information Systems, Mathematics and Legal Informatics*

### **Abstract**

In this work, we use the data caching method. The implementation of the Memcache function in the Yii2 framework is described. Also in the work of research on the use of the Yii2 framework.

**Keywords:** Caching, Memcache, Yii2.

В настоящее время, с развитием сети интернет, актуальна веб-разработка. У большинства компаний появилась необходимость в создании собственного сайта. Самым удобным и эффективным способом является создание сайтов с помощью фреймворка. Одна из функций, которая позволяет ускорить отзывчивость сайта, кэширование. В любом приложении встречаются медленные операции, это может быть: SQL запросы или запросы к внешним API, результаты которых можно сохранить на некоторое время. Это позволит выполнять меньше таких операций, а большинству пользователей показывать заранее сохраненные данные.

Кэш — это память с большей скоростью доступа, предназначенная для ускорения обращения к данным, содержащимся постоянно в памяти с меньшей скоростью доступа (далее «основная память»). Кэширование применяется ЦПУ, жёсткими дисками, браузерами, веб-серверами, службами DNS и WINS. Кэш состоит из набора записей. Каждая запись ассоциирована с элементом данных или блоком данных (небольшой части данных), которая является копией элемента данных в основной памяти. Каждая запись имеет идентификатор, часто называемый тегом, определяющий соответствие между элементами данных в кэше и их копиями в основной памяти.

Когда клиент кэша (ЦПУ, веб-браузер, операционная система) обращается к данным, прежде всего исследуется кэш. Если в кэше найдена запись с идентификатором, совпадающим с идентификатором затребованного элемента данных, то используются элементы данных в кэше. Такой случай называется попаданием кэша. Если в кэше не найдена запись, содержащая затребованный элемент данных, то он читается из основной памяти в кэш, и становится доступным для последующих обращений. Такой случай называется промахом кэша. Процент обращений к кэшу, когда в нём найден результат, называется уровнем попаданий, или коэффициентом попаданий в кэш. Например, веб-браузер проверяет локальный кэш на диске на наличие локальной копии веб-страницы, соответствующей запрошенному URL. В этом примере URL — это идентификатор, а содержимое веб-страницы — это элементы данных. Если кэш ограничен в объёме, то при промахе может быть принято решение отбросить некоторую запись для освобождения пространства. Для выбора отбрасываемой записи используются разные алгоритмы вытеснения. При модификации элементов данных в кэше выполняется их обновление в основной памяти. Задержка во времени между модификацией данных в кэше и обновлением основной памяти управляется так называемой политикой записи. В кэше с немедленной записью каждое изменение вызывает синхронное обновление данных в основной памяти. В кэше с отложенной записью (или обратной записью) обновление происходит в случае вытеснения элемента данных, периодически или по запросу клиента. Для отслеживания модифицированных элементов данных записи кэша хранят признак модификации (изменённый или «грязный»). Промах в кэше с отложенной записью может потребовать два обращения к основной памяти: первое для записи заменяемых данных из кэша, второе для чтения необходимого

элемента данных. В случае, если данные в основной памяти могут быть изменены независимо от кэша, то запись кэша может стать неактуальной. Протоколы взаимодействия между кэшами, которые сохраняют согласованность данных, называют протоколами когерентности кэша.

Кэшировать необходимо данные, которые медленно генерируются и часто запрашиваются. На практике это обычно:

- Результаты запросов к внешним сервисам (RSS, SOAP, REST и т.п.).
- Результаты медленных выборок из базы данных.
- Сгенерированные html блоки либо целые страницы.

В данной работе будет продемонстрирован способ кэширования с помощью функции Memcache в фреймворке Yii2.

Yii2 — это высокоэффективный основанный на компонентной структуре PHP-фреймворк для разработки масштабных веб-приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки. Название Yii означает простой, эффективный и расширяемый.

В электронных ресурсах можно обнаружить множество литературы, связанной с фреймворком Yii2. Так работа Д.С. Жаворонкова, А.А. Бабкиной и Е.Ю. Довгия раскрывает особенности и достоинства Yii2 при разработке REST систем. Цель работы была, выделить ключевые плюсы данного фреймворка в контексте проблемы. Все исследования в рамках работы проводились на личном опыте авторов [1]. И.В. Филимонов, И.Б. Разин в своей статье описали разработку платформы для создания сайта типа интернет-магазин на базе Yii2 framework [2]. Статья И.Г. Игнатовой, О.Н. Савёловой и И.А. Сидорова посвящена разработке подсистемы по формированию и выдаче справок студентам в институте. Данная подсистема является частью всей информационной системы института. Для разработки требовалось провести интервьюирование сотрудников института, выявить требования, спроектировать подсистему с использованием языка UML, разработать с использованием фреймворка Yii2, провести тестирование и внедрить в рабочий процесс [3].

Для демонстрации работы используется Open Server 5.2.2 с PHP 7.0

Для подключения Memcache необходимо открыть настройки Open Server в разделе «Модули», подключить имеющуюся версию memcached. В данной работе рассматривается версия 1.4.5.

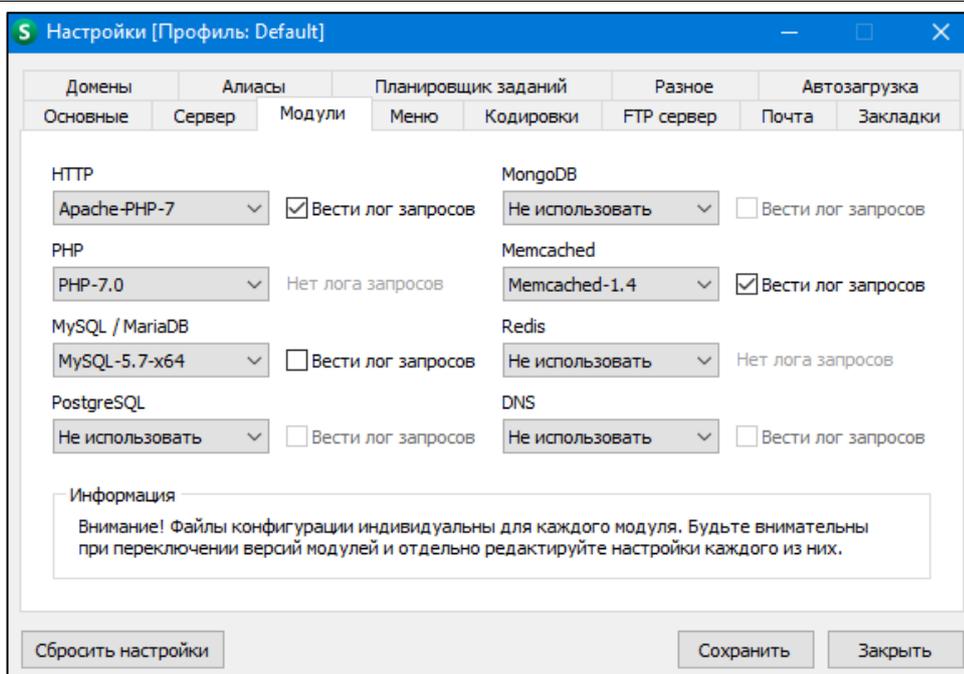


Рис.1 - Настройки Open Server

Далее необходимо скачать библиотеку для PHP с наименованием «php7-memcache-dll», которая подойдет к конфигурациям сервера. Для просмотра всех установок сервера, необходимо создать страницу с командой «echo phpinfo();». Библиотеку необходимо разместить в папке расширений для PHP «C:\OSPanel\modules\php\PHP-7.0\ext», после чего необходимо перезапустить сервер и страницу с информацией о настройках сервера. Ниже на рис. 2 представлено расширение Memcache в команде phpinfo, что означает подключенное расширение.

memcache		
memcache support	enabled	
Version	3.0.9-dev	
Revision	\$Revision\$	
Directive	Local Value	Master Value
memcache.allow_failover	1	1
memcache.chunk_size	32768	32768
memcache.compress_threshold	20000	20000
memcache.default_port	11211	11211
memcache.hash_function	crc32	crc32
memcache.hash_strategy	consistent	consistent
memcache.lock_timeout	15	15
memcache.max_failover_attempts	20	20
memcache.protocol	ascii	ascii
memcache.redundancy	1	1
memcache.session_redundancy	2	2
mongodb		
MongoDB support	enabled	

Рис.2 - Страница PHPinfo

Подключиться к модулю Memcache можно через командную строку с помощью следующей строки:

```
telnet 127.0.0.1 11211
```

Для получения текущей статистики модуля используется команда «stats». Для вывода информации о закешированных данных необходима строка «stats items», но так как модуль ещё нигде не используется, то выведется сразу «END».

Для демонстрации работы модуля была написана веб-страница.



Рис. 3 - Веб-страница

В примере происходит сохранение объекта в кэше и его последующее чтение. Объекты и другие не скалярные типы сериализуются перед сохранением, что делает невозможным хранение ресурсов (например, идентификаторов подключений и др.) в кэше.

```

17 $memcache = new Memcache;
18 $memcache->connect('localhost', 11211) or die ("Не могу подключиться");
19
20 $version = $memcache->getVersion();
21 echo "Версия сервера: ".$version."<br/>\n";
22
23 $tmp_object = new stdClass;
24 $tmp_object->str_attr = 'test';
25 $tmp_object->int_attr = 123;
26
27 $memcache->set('key', $tmp_object, false, 10) or die ("Ошибка при сохранении данных на сервере");
28 echo "Данные сохранены в кэше. (время жизни данных 10 секунд)<br/>\n";
29
30 $get_result = $memcache->get('key');
31 echo "Данные из кэша:<br/>\n";
32
33 var_dump($get_result);

```

Рис. 4 - Код примера на PHP

Результатом выполнения кода будет кэширование примерного элемента и вывод отчёта на веб-странице.

```

Версия сервера: 1.4.5_4_gaa7839e
Данные сохранены в кэше. (время жизни данных 10 секунд)
Данные из кэша:
object(stdClass)#121 (2) ( ["str_attr"]=> string(4) "test" ["int_attr"]=> int(123) )

```

Рис. 5 - Отчёт о выполнении

Также в командной строке в подключении к модулю через telnet после введения строки «stats items» отобразится информация о данных в кэше.

```
stats items
STAT items:1:number 1
STAT items:1:age 1526
STAT items:1:evicted 0
STAT items:1:evicted_nonzero 0
STAT items:1:evicted_time 0
STAT items:1:outofmemory 0
STAT items:1:tailrepairs 0
STAT items:1:reclaimed 0
STAT items:2:number 3
STAT items:2:age 1526
STAT items:2:evicted 0
STAT items:2:evicted_nonzero 0
STAT items:2:evicted_time 0
STAT items:2:outofmemory 0
STAT items:2:tailrepairs 0
STAT items:2:reclaimed 0
STAT items:3:number 1
STAT items:3:age 1885
STAT items:3:evicted 0
STAT items:3:evicted_nonzero 0
STAT items:3:evicted_time 0
STAT items:3:outofmemory 0
STAT items:3:tailrepairs 0
STAT items:3:reclaimed 1
END
```

Рис. 6 - Информация о кэше

Благодаря гибкости и простоте в использовании данного модуля можно получить хорошую систему кэширования данных и более сложных веб-ресурсов без применения огромных усилий.

### Библиографический список

1. Жаворонков Д.С., Бабкина А.А., Довгий Е.Ю. Достоинства фреймворка Yii2 в разработке rest систем // "Чистая наука" на службе научно-технического прогресса, 2018. С. 9-11.
2. Филимонов И.В., Разин И.Б. Разработка платформы для создания сайта типа интернет-магазин на базе Yii2 framework // Инновационное развитие легкой и текстильной промышленности, 2018. С.111-113.
3. Игнатова И.Г., Савёлова О.Н., Сидоров И.А. Разработка подсистемы формирования справок в информационной системе института // Аллея науки. 2017. №11. С. 156-159.