

Кэширование информации с использованием языка программирования PHP

Ересь Артём Владимирович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассмотрен процесс кэширования информации с использованием языка программирования PHP. Подробно будет раскрыт смысл этого понятия и даны правила для работы с ним.

Ключевые слова: Сайт, данные, PHP, сервер, система

Caching of information with use of the PHP programming language

Yeres Artem Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article process of caching of information with use of the PHP programming language is considered. The sense of this concept will be in detail revealed and rules for work with it are given.

Keywords: Website, data, PHP, server, system

Большинство крупных компании по работе с масштабными проектами в сфере информационных технологий зачастую несут огромные трудозатраты по обработке информации. И используемые серверы значительно перегружаются данными, что существенно снижает скорость работы. И здесь на помощь приходит процесс кэширования данных, позволяющий уменьшить серверные нагрузки.

Целью данной статьи является раскрытие смысла понятия кэширования в PHP относительно целей снижения нагрузок на сервер.

В сфере научно-исследовательской деятельности тема данной статьи имеет достаточное количество работ, представленных в свободный доступ. Авторы А.А. Нуров и А.В. Кистрин в своей работе выделили подходы к теме кэширования в приложениях с высокой нагрузкой. Работа написана с примерами и подробными методами [1]. Исследователь К.А. Кондаков знакомит пользователей с программным продуктом для кэширования данных и дает полное понимание этого процесса с точки зрения программных сессий [2]. В.П. Носов разобрал данную тему с точки зрения образовательных ресурсов, затронул кэширование приложений информационного характера [3]. В ресурсах сети Интернет рассматривается кэширование на языке

программирования PHP, выделяются основные аспекты и методы для проведения операций с данными [4-5].

В начале работы рассмотрим такой элемент как меню с несколькими уровнями. Именно оно хорошо показывает на своем примере важность функции кэширования. Множество очень тяжелых процессов использует меню. Кэширование поможет обойтись лишь одним файлом.

Функция `view cat` позволяет вывести меню на экран, но результат получить мы не можем. Так как необходимо, чтобы функция давала нам код меню, а не просто отправляла его на экран, то используем кэширование. Изменим наш код следующим образом:

```
1 //вывод каталога с помощью рекурсии
2 function view_cat($arr,$parent_id = 0) {
3
4 //Условия выхода из рекурсии
5 if(empty($arr[$parent_id])) {
6 return;
7 }
8 $menu .= '<ul>';
9 //перебираем в цикле массив и выводим на экран
10 for($i = 0; $i < count($arr[$parent_id]);$i++) {
11 $menu .= '<li><a href="?category_id='.$arr[$parent_id][$i]['id'].
12 '&parent_id='.$parent_id.'">'
13 . $arr[$parent_id][$i]['title'].'</a>';
14 //рекурсия - проверяем нет ли дочерних категорий
15 $menu .= view_cat($arr,$arr[$parent_id][$i]['id']);
16 $menu .= '</li>';
17 }
18 $menu .= '</ul>';
19
20 return $menu;
21 }
```

Рис. 1. Исходный код меню

Далее увидим результат функции в `$ menu`, раскроем данные переменной:

```
1 // получаем массив каталога
2 $result = get_cat();
3 $menu = view_cat($result);
4
5 //Выводи каталог на экран с помощью рекурсивной функции
6 echo '<div style="width:300px;float:left; padding:10px; border:1px solid #074776">';
7 echo $menu;
8 echo '</div>';
```

Рис. 2. Menu

Следующим шагом будет вставка функции кэширования.

Обращаемся к `getcat` (база данных) и `viewcat` (рекурсия) для вывода нашего меню. В работе есть условие: отсутствие кэш-файла. Если ситуация обратная, то есть возможность просто прочесть содержание без наших функций выше.

В `functions` добавим функционал для создания и прочтения кэш-файла.

```
1 // создание файла кэша
2 function cache_file_create($data, $file){
3
4 }
5
6 // чтение файла кэша
7 function cache_file_read($file){
8
9 }
```

Рис. 3. Кэш-операции

Отметим, что параметрами для операции создания файла кэша являются данные для этого файла, а для чтения – его расположение на компьютере.

Для хранения кэш-файлов необходима папка, создадим ее. Файлу кэширования дадим имя «menu.html». Укажем необходимый файл и добавим его:

```
1 $menu_cache = "cache/menu.html";
```

Рис. 4. Файл кэша

Создадим необходим нам условие для получения информации в кэш-файл:

```
1 $menu_cache = "cache/menu.html";
2
3 if( file_exists($menu_cache) ){
4     $menu = cache_file_read($menu_cache);
5 }else{
6     // получаем массив каталога
7     $result = get_cat();
8     $menu = view_cat($result);
9
10 cache_file_create($menu, $menu_cache);
11 }
```

Рис. 5. Условие для кэша

Условие работает так: если кэш-файл существует – получаем результат, если отсутствует – из базы данных читаем данные добавляем в файл.

Опишем функции для работы с файлом.

```
1 // создание файла кэша
2 function cache_file_create($data, $file){
3     $handle = fopen($file, "w"); // открываем файл на запись или создаем, если его нет
4     flock($handle, LOCK_EX); // блокируем файл для единоличного пользования
5     fwrite($handle, $data); // записываем в него данные
6     flock($handle, LOCK_UN); // разблокируем файл
7     fclose($handle); // закрываем файл
8 }
```

Рис. 6. Функции создания

```
1 // чтение файла кэша
2 function cache_file_read($file){
3     if( !$menu = file_get_contents($file) ){
4         // если файл пуст
5         $result = get_cat(); // получаем данные из БД
6         $menu = view_cat($result); // формируем меню
7         cache_file_create($menu, $file); // записываем меню в кэш
8     }
9     return $menu; // возвращаем меню
10 }
```

Рис. 7. Функции чтения

Таким образом, в работе было рассмотрено кэширование с использованием языка программирования PHP. Раскрыты основные процессы этой процедуры относительно целей снижения нагрузок на сервер.

Библиографический список

1. Нуров А.А., Кистрин А.В. Кэширование данных в web-приложениях // Методы и средства обработки и хранения информации. 2018. С. 89-93. URL: <https://elibrary.ru/item.asp?id=32583917> (Дата обращения: 03.01.2019)
2. Кондаков К.А. MEMCACHED. Эффективно работаем с кэшированием // Системный администратор. 2013. №11. С. 18-21. URL: <https://elibrary.ru/item.asp?id=20466316> (Дата обращения: 03.01.2019)
3. Носов В.П. Кэширование в web-приложениях, используемых для создания образовательных информационных ресурсов // Телекоммуникации и информатизация образования. 2005. №6. С. 77-81. URL: <https://elibrary.ru/item.asp?id=9230157> (Дата обращения: 03.01.2019)
4. Кэширование в PHP URL: <https://www.internet-technologies.ru/articles/keshirovanie-stranic-v-php.html> (Дата обращения: 03.01.2019)
5. Простая и эффективная система кэширования в PHP URL: <https://ruseller.com/lessons.php?id=1555> (Дата обращения: 03.01.2019)