phpFastCache V2

by khoaofgod@yahoo.com | http://www.facebook.com/khoab

Website: http://www.phpfastcache.com
Github: https://github.com/khoaofgod/phpfastcache

1. What's new in version 2.0?

To take advantage of saving more memory and CPU, phpFastCache will not auto load all classes and functions as version 1.3. The version 2 supported instances, new class() , easier to edit, update and use with a lot of methods.

Faster, and easier. Feel free to travel between all caching methods.

Go ahead and get started.

2. Caching SQL | Simple Example

This example use to reduce database calls. For example, you have 10,000 visitors, this code will send query to DB only 1 time, and use the cache to serv other 9,999 visitors in 5 minutes before send another query.

```
<?php

Required_once("phpfastcache/phpfastcache.php");
// change auto to: files, sqlite, memcache, memcached, xcache, wincache, apc or simple with "auto"…etc
phpFastCache::setup("storage","auto");

$cache = phpFastCache();

// GET FROM CACHE FIRST
$products = $cache->get("my_products");

If($products == null) {
        $products = "MY QUERY RESULTS | MY FUNCTIONS GET PRODUCTS";
        // write to cache 300 seconds = 5 minutes
        $cache->set("my_products", $products, 300);
}

// RENDER PAGE
echo $products;
```

3. Reduce API Call, Render Your Page

I know you're using Bing API, Google API, Facebook API, eBay API, Amazon API... etc. And they are limit your requests. You can't continue send request to their server in short time. This example will show your how to save your API Transactions.

```
<?php

Required_once("phpfastcache/phpfastcache.php");

// check from cache first
$images = __c()->my_images;


if($images == null) {
        $images = "BING API GET IMAGES LIST | GOOGLE API SEARCH IMAGES";
        // Write to cache in 3600, if any one searching same keywords, will use cache.
        // Your Server will not use API again, until cache is expired
        __c()->my_images = array($images, 3600);
}

// Render Your Page
Foreach($images as $img) {
        // output img
}
```

4. Learn how to use it

Keep it simple, you only need **remember 2 methods:**

1. $products = $cache->get("keyword");
2. $cache->set("keyword", "VALUE | PRODUCTS | ARRAY | OBJECT" , 300);

Now you are ready to speed up your slowly website, and rock the world.

5. Want to learn it seriously ?

Cool, man. Take at look at how easy on caching with phpFastCache. Here is the list of all the functions.

1. require_once("phpfastcache/phpfastcache.php");
2. $cache = phpFastCache();
3. $data = $cache->get("keyword");
4. $cache->set("keyword", $data, $time_in_second);
5. $cache->delete("keyword");
6. $cache->clean();

7. $stats = $cache->stats();
8. $cache->increment("keyword", $step = 1);
9. $cache->decrement("keyword", $step = 1);

"Step 2" is $cache = phpFastCache(); to create or use Existing instance for whole php script. Use $cache = new phpFastcache(); the instance will be destroy after your function finish.

6. "Shortcut", some of us like it!

The shortcut is new, and only available on this version. Enjoy it!

- $data = __c()->get("keyword");
- __c()->set("keyword");
- $data = $cache->keyword2;
- $data = __c()->keyword3;
- $cache->keyword5 = array($data_need_set , $time_in_second);
- $cache->my_products = array($products, 300);
- __c()->enjoy = array($array_listing, 10);

Look cool huh? You now can use __c() on whatever function, classes in your scripts without any limit.

7. Create New "Instances"?

To create new instances you can use, those instances will be destroy after your function finished. However, the cache will still here until they are expired.

1. $cache = new phpFastCache();
2. $cache = new phpFastCache("memcache");
3. $cache = new phpFastCache("apc");
4. $cache = new phpFastCache("files");

To create only 1 instance, and keep using it until your script finish:

1. $cache = phpFastCache();
2. $cache = __c();
3. $cache = __c("memcached");

8. How To "Get" From Cache?

Use $cache->get("keyword_identity") or any methods:

1. $data = $cache->get("keyword");
2. $data = $cache->keyword;
3. $data = __c()->get("keyword");
4. $data = __c()->keyword;
5. $data = $cache->memcache->keyword;
6. $data = __c()->files->get("keyword");
7. $data = __c("apc")->keyword;

Now, pick one that easy for you.

9. "Set" Something to Cache

Method set("keyword", "data | array | object" , "time_in_second");

1. $cache->set("products", $products_list, 300);
2. $cache->my_images = array($images_list, 3600*24);
3. __c()->set("something", "some data", 600);

As "Get" method, you can use shortcut or travel between any caching method easily.

10. Setup & Configuration

By default $cache = phpFastCache(); will setup as "auto" , and it will set sqlite, files, memcache , apc or whatever you have in server. To specific a caching method, in your config.inc.php or whatever your config file. You use:

Require_once("phpfastcache/phpfastcache.php");
phpFastCache::$storage = "files";

After this line, any where you use $cache = phpFastCache() or __c(); it will use "files" caching method.
For detail of config, please look at examples/2.setup.php ← this is full setup;

Setup MemCache Default Server

```
$server = array(array("127.0.0.1",11211,100),
                  array("server2",12212,80),
                );
```

Default Config for all phpFastCache("memcache"); and __c(); // default will be 127.0.0.1 with port 11211

Require_once("phpfastcache/phpfastcache.php");
phpFastCache::setup("server", $server);

// After this config, all memcache will be setup to default server

$memcache = __c();
$memcache->get("something");

// Setup Server for 1 instance
$cache = phpFastCache("memcache");
$cache->option("server", $server);

Please do not use $cache = new phpFastCache("memcache");
Because we don't want PHP try to connect to memcache server every time. We only want keep 1 connection by 1 instance for whole php script.

Setup Specific Path For Files Cache

As you know that Files & Sqlite require writable permission to rewrite files and create folders.

If you're using CGI, SUPHP, or ANY PHP run your USER ACCOUNT, then PHP will auto create folder and setup files for you without any problem.

If you're using PHP as Apache Module, PHP as Nobody, then you will get the problem with Files Permission. However, don't worry about this, phpFastCache will auto detect and use one of your "tmp" directory or "upload" directory that you have permission to write to setup your Files Caching there.

In case that phpFastCache couldn't look for any folders have write permission and You don't have /tmp folder in your server, then phpFastCache will tell you that you have to setup one PATH for your Files Caching.

Here is how to setup the PATH to custom place:

Require_once("phpfastcache/phpfastcache.php");
phpFastCache::setup("path","/home/your/path");

OR

$cache = phpFastCache("files");
$cache->option("path","/home/username/caching/");
$cache->option("securityKey","cache.folder");

And then remember create and chmod 0777 for that folders.
For Security, phpFastCache will automatic create .htaccess and block any request from outside to that folder, but I suggest you setup the path outside of your public_html for more security.

Fallback Driver

For example, you use travelling between many drivers, and $cache->memcached, $cache->files, $cache->apc

And then when you moved your Web Hosting, the new server doesn't have memcached or apc. So, the traveling links will be break and you got errors.

In Your config files.

Require_once("phpfastcache/phpfastcache.php");
phpFastCache::setup("fallback", array("memcached" => "files", "apc" => "files"));

From this line, any traveling links to memcached and apc will use "files" driver. And you don't need to re-code your php files again.

11. Travel Between Multiple Caching Method

In the real world, we use Memcache | APC | XCache | WinCache for caching DB only. If we have to cache a part content of website, do files, apc and sqlite is good and fast enough.

Why? Because caching whole webpage, and a part of content are cost a lot of "bytes". And Memcache or any Memory Caching method will use your Server Memory, and that is the limit. Just thinking a website after render will have around 200kb html code, and then you cache that to your memory. By the times, your memory will cost like 2GB , 4GB, 8GB and 16GB or a Data Center like Facebook, with thounsand of Server to caching with Memcache.

But 200KB to a file or sqlite, will have no problem, your Hard Drive have ton of space, 1TB, 2TB, 4TB.

If you're using shared web hosting, not dedicated server, then I suggest you use files, sqlite only. If you're using VPS, Dedicated Server, then go for APC, MemCache, XCache.

You can mix all of them together with phpFastCache:

```
<?php

Required_once("phpfastcache/phpfastcache.php");

$cache1 = phpFastCache("files");
$cache2 = phpFastCache("memcache");
$cache3 = phpFastCache("apc");

// Now this is the magic
$cache1->memcache->get("keyword");
$cache2->files->get("keyword");
$cache3->sqlite->get("keyword");


__c("xcache")->get("keyword");
__c()->wincache->get("keyword");
```

Feel FREE to travel between all caching methods without any limit with phpFastCache

12. Write An Extension

Very simple if you want write an extension for phpFastCache.

First, please look at files "phpfastcache/ext/files.php"

And an example is "phpfastcache/ext/example.php"

You can add your own cache method, new methods and share it in our caching community http://www.codehelper.io

And I am sure that I will add it into my class when I see you submit your extension.

13. Multiple Set & Get

Use methodMulti with array():

1. $cache->getMulti(array("keyword1","keyword2","keyword3"));
2. $cache->setMulti(array("keyword", "value, time), array("keyword","value", time));
3. $cache->deleteMulti(array("key1","key2","key3");
4. $cache->isExistingMulti(array("key1","key2","key3");

14. Bugs, Issues And Contact

As I said, any questions please visit our forum at http://www.codehelper.io

You don't need to register, just login with your Facebook Account.

Post any issues here and we will fix, improve it together.