

Instantiation

set

exists

get

clear

reset

load

Cache

This class represents F3's multi protocol Cache engine. It supports Memcache (<http://memcached.org/>), WinCache (<https://www.iis.net/downloads/microsoft/wincache-extension>), APC (<http://php.net/manual/en/book.apc.php>), XCache (<http://xcache.lighttpd.net/>), Redis (<http://redis.io/>) and filesystem based caching.

Caching is a powerful way to get more performance out of your application. It is seamlessly integrated to all core and plugin features like setting hive keys (`base#set`), HTTP responses (`base# caching`), DB mappers and queries and even to JS/CSS minification (`optimization#keeping-javascript-and-css-on-a-healthy-diet`).

There is a good Cache Engine User Guide (`optimization#cache-engine`) that covers how the cache engine works and gives you tips to improve your application and your database queries, as they can be cached by F3 as well. You really should have read it.

Namespace: `\`

File location: `lib/base.php`

Instantiation

Return class instance

```
$cache = \Cache::instance();
```

The Cache engine is deactivated by default. To activate it using an auto-detected cache backend, simply set the `CACHE` (`quick-reference#cache`) system var to `TRUE`. See the `load()` function (`cache#load`) below for additional configuration.

The Cache class uses the Prefab (`prefab-registry`) factory wrapper, so you are able to grab the same instance of that class at any point of your code.

set

Store a key/value pair in the cache

```
mixed|FALSE set( string $key, mixed $val [, int $ttl = 0 ] )
```

If `$ttl` is 0 then the entry is saved for an infinite time. Otherwise, the specified time, in seconds, is used as TTL.

NB: F3 versions older than 3.6.5 are reusing old expiration times. Since 3.6.5 only old TTL values are reused. See [issue #237](https://github.com/bcosca/fatfree-core/issues/237) for details.

exists

Check if a cache entry exists: Return timestamp and TTL of cache entry or FALSE if not found

```
array|FALSE exists( string $key [, mixed &$amp;val = NULL ] )
```

Returns an array containing the elements [0] creation timestamp and [1] time-to-live (TTL, in seconds) of the cache entry; or `FALSE` if it was not found.

```
$cache->set('foo','bar',5);

var_dump($cache->exists('foo'));
/* returns
array(2) {
  [0]=> float(1369257446.4874)
  [1]=> int(5)
}
*/
```

You can use the `$val` argument to fetch the cache entry content as well. This could save an additional `get` call to the cache backend.

```
if ($cache->exists('foo',$value)) {
    echo $value; // bar
}
```

get

Retrieve value of a cache entry

```
mixed|FALSE get( string $key )
```

clear

Delete a cache entry

```
bool clear( string $key )
```

reset

Clear contents of the cache backend

```
bool reset( [ string $suffix = NULL ] )
```

You can use `$suffix` to only clear cache entries with a suffix that matches this string.

You can also use `$f3->clear('CACHE')` as a shortcut to this.

Limitations:

- when using XCache, all entries are cleared, no matter the value of `$suffix`.
- when using Memcached via the memcache (<https://pecl.php.net/package/memcache>) extension, the *cachedump* stat should be enabled (no `-X` flag), otherwise the command won't work. The memcached (<https://pecl.php.net/package/memcached>) extension doesn't have this limitation.

load

Load/auto-detect cache backend. Return the cache DSN

```
string load( string|bool $dsn )
```

Possible configurations for `$dsn` are:

- `apc`
- `apcu`
- `wincache`
- `xcache`
- `memcache=localhost:11211`
- `memcached=localhost:11211`
- `redis=localhost`
- `folder=tmp/cache/`

You can set `$dsn` to `TRUE` to trigger the auto-detection feature, using the filesystem as a fallback when no shared memory engine has been detected or is available. Setting `$dsn` to `FALSE` disables caching.

The `memcache` and `memcached` backends support multiple servers. The servers have to be separated by `,`, `;` or `|`.

Returns the cache DSN of the cache in use (specified or auto-detected).