Masahiro Nagano / Cache-Memcached-Fast-Safe-0.06 2 ++ ⭐ Starred 2 **/ Cache::Memcached::Fast::Safe**

Contents [hide]

NAME

Cache::Memcached::Fast::Safe - Cache::Memcached::Fast with sanitizing keys and fork-safe

SYNOPSIS

```
use Cache::Memcached::Fast::Safe;

my $memd = Cache::Memcached::Fast::Safe->new({
  servers => [..]
});

#This module supports all method that Cache::Memcached::Fast has.
```

DESCRIPTION

Cache::Memcached::Fast::Safe is subclass of Cache::Memcached::Fast.
Cache::Memcached::Fast::Safe sanitizes all requested keys for against memcached injection problem. and call disconnect_all automatically after fork for fork-safe.

ADDITIONAL METHOD

get_or_set($key:Str, $callback:CodeRef [,$expires:Num])

Get a cache value for $key if it's already cached. If can not retrieve cache values, execute $callback and cache with $expires seconds.

```
$memcached->get_or_set('key:941',sub {
  DB->retrieve(941)
},10);
```

callback can also return expires sec.

```
$memcached->get_or_set('key:941',sub {
  my $val = DB->retrieve(941);
  return ($val, 10)
});
```

CUSTOMIZE Sanitizer

This module allow to change sanitizing behavior through $Cache::Memcached::Fast::Safe::SANITIZE_METHOD. Default sanitizer is

```
use bytes;
my %escapes = map { chr($_) => sprintf('%%%02X', $_) } (0x00..0x20, 0x7f..0xff);
local $Cache::Memcached::Fast::Safe::SANITIZE_METHOD = sub {
    my $key = shift;
    $key =~ s/([\x00-\x20\x7f-\xff])/$escapes{$1}/ge;
    if ( length $key > 200 ) {
        $key = sha1_hex($key);
    }
    $key;
};
```

AUTHOR

Masahiro Nagano <kazeburo {at} gmail.com>

SEE ALSO

Cache::Memcached::Fast, http://gihyo.jp/dev/feature/01/memcached_advanced/0002 (Japanese)

LICENSE

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.