

# 64位ID生成器说明

我们在libfastcommon中实现了64位（8字节整数）ID生成器，针对PHP这样的多进程方式，生成的64位ID也可以做到全局唯一。

提供的php扩展 php-fastcommon，封装了64位ID生成器。

64位ID生成规则（注：生成的ID可以小于64位）

32位Unix时间戳 + X位机器ID + Y位extra data + Z位序号

其中  $X + Y + Z \leq 32$

\* 机器ID (machine\_id, 缩写为mid) 可以在初始化时指定，如果设置为0表示获取本地IP地址的后X位作为机器ID

\* extra data用来存储额外信息，例如订单分库的库号。如果不需要这个特性，将Y设置为0即可

\* 序号 (sn) 会保存在本地文件中，建议序号的位数Z至少为14，其对应的最大数值为16383 (16K)

php-fastcommon扩展提供的4个PHP函数

```
resource fastcommon_id_generator_init([string $filename = "/tmp/fastcommon_id_generator.sn",  
int $machine_id = 0, int $mid_bits = 16, int $extra_bits = 0, int $sn_bits = 16])
```

return resource handle for success, false for fail

\* 这个函数只需要在初始化的时候调用一次即可，建议不同的实例采用不同的文件来保存序列号。

\* php程序运行用户对这个文件必须有读写权限，文件不存在会自动创建。

\* 返回的resoure需要保存到php变量，否则该初始化工作会自动撤销

```
long/string fastcommon_id_generator_next([int $extra = 0, resource $handle = null])
```

return id for success, false for fail

return long in 64 bits OS, return string in 32 bits OS

\* 如果不需要存储额外信息，extra传0即可。

\* 其中\$handle参数为 fastcommon\_id\_generator\_init返回值，不传递该参数表示使用最后一次调用 fastcommon\_id\_generator\_init返回的handle。

```
int fastcommon_id_generator_get_extra(long id [, resource $handle = null])
```

return the extra data

\* 使用了额外信息的情况下，可以使用这个函数获取ID中包含的extra data

```
bool fastcommon_id_generator_destroy([resource $handle = null])
```

return true for success

\* 这个函数通常不需要显式调用