单例模式

2011年6月7日
9:18

单例模式（单一实例模式）

```
<?php

class db {
	private static $obj = NULL;

	private function __construct() {

	}

	private function __clone() {

	}

	public static function instance() {
		if(!is_object(self::$obj)) {
			self::$obj = new self();
		}
		return self::$obj;
	}
}

?>
```

目录结构

2011年6月7日
9:52

主要目录

api目录	外部接口目录
archiver目录	无样式论坛
config目录	配置文件目录
data目录	缓存目录
install目录	安装目录
source目录	源代码目录
static目录	图片，js目录
template目录	模板源文件目录
uc_client目录	UCenter接口目录
uc_server目录	UCenter目录

根目录文件

admin.php文件	后台管理入口
api.php文件	外部接口
connect.php文件	QQ互联平台接口
cp.php文件	应用管理面板
forum.php文件	论坛入口文件
group.php文件	群组入口文件
home.php文件	家园入口文件
index.php文件	首页文件
member.php文件	用户基本操作
misc.php文件	杂项文件
Plugin.php文件	插件的入口文件
portal.php文件	门户入口文件
search.php文件	站内搜索文件
userapp.php文件	用户应用管理文件

重要文件

source/class/class_core.php	核心类文件(单例加载)
source/function/function_core.php	核心函数文件(加载核心类时加载)
source/function/function_cache.php	缓存生成文件(更新缓存时引入)
source/class/class_template.php	模板引擎类(template()函数加载)
source/funcion/function_admincp.php	后台管理函数(根目录admin.php加载)
source/class/class_memcache.php	Memcached类封装
config/config_global.php	核心配置文件
config/config_ucenter.php	UCenter配置文件
forum.php	广场入口
home.php	家园入口

class_core.php

2011年6月7日
9:47

方法清单

注：以'_'开头的方法，为私有方法

instance方法	单例模式加载核心类方法
构造函数	初始化核心类时调用的方法(调用一个方法列表)
init方法	二次初始化调用的方法(调用另一个方法列表)
_init_env方法	初始化运行环境
_init_config方法	加载配置文件方法
_init_input方法	过滤输入参数方法
_init_output方法	初始化输出设置
_init_db方法	初始化数据库链接
_init_memory方法	初始化通用内存接口类
_init_user方法	初始化用户信息
_init_session方法	初始化Discuz! X 内置Session类
☆ _init_setting方法	加载核心类的cachelist数组中的缓存，加载网站的配置
_init_mobile方法	初始化wap版Discuz
_init_cron方法	初始化计划任务类
_init_misc方法	初始化其他内容(如：时区设定)
_init_style方法	在模板加载时，初始化网站样式

成员清单

$db	数据库类实例
$mem	内存通用接口类实例
$session	Session类实例
$config	配置文件数组
$var	全局变量($_G)
$cachelist	初始化setting时加载的缓存列表
$init_xxx	为各个核心功能的开关
$superglobal	允许在$GLOBALS存在的一维索引名称

使用方法

单例模式引入

```
$discuz = & discuz_core::instance();
```

例如在一个函数使用核心类实例

```
function show() {
	$discuz = & discuz_core::instance();  //取得实例
	echo '<pre>';
	print_r($discuz->var);  //输出全局变量
}
```

数据库操作

2011年6月14日
8:27

核心类中调用了_init_db方法，进行了db的初始化

Discuz! X 对数据库进行的二次封装

数据库查询采用了类的静态调用方式

调用方法： DB::方法名() 操作数据库

例子:

查询pre_common_member表的uid为3的用户的信息数组

代码：

```
$query = DB::query("SELECT * FROM ".DB::table('common_member')." WHERE uid = 3 LIMIT 1");
while($result = DB::fetch($query)) {

}
```

方法	作用	参数
DB::table	获取数据表真实名称	$table = 用户表名
DB::delete	删除表数据	$table = 操作表 $condition = 限制条件 $limit = 删除条数 $unbuffered = 是否缓存查询数据
DB::insert	向某个表插入数据	$table = 操作表 $data = array('字段名1' => '值1', '字段名2' => '值2'); $return_insert_id = 是否返回插入id $replace = 是否使用replace into $selient = 是否报错(默认提示)
DB::update	更新某个表	$table = 操作表 $data = 数据 $condition = 限制条件array('字段' => '值') $unbuffered = 是否缓存查询数据 $low_priority = 是否锁表更新
DB::implode_field_value	获取字段拼接串	$array = array('字段名' => '字段值') $glue = 连接条件的字符串
DB::Insert_id	获取最后插入id	
DB::fetch	获取查询集的关联数组	$resourceid = QUERY的资源集 $type = 数组的类型 1.MYSQL_BOTH 数字索引与关联索引 2.MYSQL_ASSOC 关联索引 3.MYSQL_NUM 数字索引
DB::fetch_first	获取一条查询数组	$sql = SQL语句 注：只出现一条结果数组
DB::result	获取某一个查询集的关联数组的索引	$resourceid = QUERY资源集 $row = 取得的字段的顺序
DB::result_first	获取一个查询集的关联数组的索引	$sql = SQL语句 注：只出现一条结果

DB::query	数据库查询方法	$sql = SQL语句 $type = 类型
DB::num_rows	取得结果集中行的数目	$resourceid = QUERY资源集
DB::affected_rows	获取增，删，改操作的影响行数	
DB::free_result	释放内存	$query = 需要释放的结果集
DB::error	获取错误信息	
DB::errno	获取错误错误号	
DB::_execute	执行某个类的实例	$cmd = 使用的方法 $arg1 = 参数1 $arg2 = 参数2
DB::object	获取某个类的实例	$dbclass = 二次调用的类名
DB::checkquery	检查query查询安全性	$sql = 被检查的SQL语句
DB::_do_query_safe	根据配置信息，判断查询安全性	$sql = 被检查的SQL语句

数据的交换

2011年6月14日
9:31

$_GET用于取url参数
$_POST用户取POST表单的数据
$_COOKIE用户取用户的cookie信息（带有前缀）
$_FILES用户上传文件（可以正常读取）

获取值	变量取法
$_GET['名称']	$_G['gp_名称']
$_POST['名称']	$_G['gp_名称']
$_COOKIE中的auth	$_G['cookie']['auth'] 注：去掉前缀，经过转义
$_FILES	$_FILES
网站memory配置（$_config['memory']['memcache']['pre']）	$_G['config']['memory']['memcache']['pre']
$_GET['mod']	$_G['mod']
$_GET['inajax']	$_G['inajax']
$_GET['page']	$_G['page']
$_COOKIE['sid']	$_G['sid']
用户信息	$_G['member']
用户id 注：判断用户是否登录	$_G['uid']
用户名	$_G['username']
用户组id	$_G['groupid']
管理组id	$_G['adminid']
网站的管理配置信息	$_G['setting']
当前用户组信息	$_G['group']
加载medal的缓存	loadcache('medal'); $_G['cache']['medal']
每页帖子数	$_G['tpp']
每页主题数	$_G['ppp']
表单hash值	FORMHASH $_G['formhash']
时间戳	TIMESTAMP

获取source/function/cache/cache_medal.php文件

取法：include libfile('cache/medal', 'function');

核心函数库

2011年6月7日

/source/function/function_core.php

system_error	错误报告
updatesession	更新Session
dmicrotime	获取微秒时间
setglobal	将数值存入$_G中，如：$_G['cache']['brand'] = 1，调用方法setglobal('cache/brand', 1);
getglobal	与getglobal对应，获取$_G中的数值
getgpc	获取$_GET或$_POST或$_COOKIE中的莫个索引值
getuserbyuid	通过uid获取用户信息
getuserprofile	获取用户详细信息
daddslashes	转译功能，可以转译数组(输入数据转译，防止注入)
authcode	加密cookie用的函数，uc通信中也用到
dfsockopen	模拟发包器，通过http协议发送头部信息
dhtmlspecialchars	数据的html实体化，可以防止页面的跨站漏洞
dexit	封装了output输出的函数
dheader	发送头部信息的函数
dsetcookie	根据配置文件，设置cookie到用户的浏览器
getcookie	获取cookie，封装了$_G['cookie']数组
fileext	获取文件扩展名
formhash	表单hash，一定程度的防止站外提交
checkrobot	判断是否为机器人程序
checkmobile	获取手机信息
dstrpos	字符串查找
isemail	邮箱正则匹配
quescrypt	获取安全问题的md5截取值
random	随机获得一个长度的字符串
strexists	检查字符串是否存在
avatar	获取用户头像
lang	加载语言包
checktplrefresh	检查模板是否更新
template	加载模板
modauthkey	根据用户信息生成authkey
getcurrentnav	
loaducenter	加载UCenter通信包
loadcache	加载缓存
cachedata	获取缓存数据
dgmdate	格式化时间戳为日期
dmktime	用日期生成时间戳

save_syscache	存储缓存数据进入common_syscache表
block_get	获取block数据
block_display	显示block
dimplode	打散一个数组
libfile	加载一个脚本文件
dstrlen	获取字符串长度
cutstr	截取字符串
dstripslashes	去掉转译，可以操作数组
aidencode	
getforumimg	
rewriteoutput	重写页面url链接
mobilereplace	手机浏览字符替换
mobileoutput	手机浏览输出
output	输出页面代码，包括调用url重写，静态页生成，手机版输出
output_replace	
output_ajax	
runhooks	运行钩子，取出钩子数据
hookscript	runhooks调用，用来加载数据
hookscriptoutput	模板输出前调用
pluginmodule	
updatecreditbyaction	通过action更新用户积分
checklowerlimit	
batchupdatecredit	
updatemembercount	更新用户积分数
checkusergroup	
checkformulasyntax	
checkformulacredits	
debug	页面打印变量信息
debuginfo	调用Discuz X 的调试模块，调试
getfocus_rand	
check_seccode	检查验证码
check_secqaa	检查安全问题
adshow	显示广告
showmessage	提示用户信息
submitcheck	检查用户表单提交
multi	分页
simplepage	简单分页
censor	敏感词
censormod	
space_merge	合并用户空间
runlog	运行日志
stripsearchkey	

dmkdir	创建目录
dreferer	用户点击来源
ftpcmd	ftp上传
diconv	iconv转码
renum	
getonlinenum	获取在线人数
sizecount	获取尺寸
swapclass	
writelog	写入Discuz X 系统运行日志
getcolorpalette	
getstatus	
setstatus	
notification_add	
manage_addnotify	
sendpm	发送pm信息
g_icon	获取图标
updatediytemplate	更新diy模板
space_key	获取空间问题
getposttablebytid	获取发送帖子的表（x2的分表功能主要用到的函数）
getposttable	获取帖子的表格
memory	将数据存入内存缓存中
ipaccess	
ipbanned	判断ip是否被禁止
getcount	获取指定表的count数
sysmessage	系统消息提示
forumperm	板块权限判断
checkperm	检查用户权限
periodscheck	
cknewuser	
manyoulog	更新漫游日志
useractionlog	更新用户动作日志
getuseraction	获取用户动作
getuserapp	获取用户漫游应用
getmyappiconpath	
getexpiration	获取过期时间
return_bytes	返回一个大小的字节数
get_url_list	
iswhitelist	
update_template_block	
http_build_query	
getrelatedlink	
getattachtablebyaid	

getattachtableid	
getattachtablebytid	
getattachtablebypid	
getattachnewaid	
get_seosetting	获取seo设置
strreplace_strip_split	
get_title_page	
getimgthumbname	
updatemoderate	
userappprompt	

模板标签

2011年6月7日
9:48

模板标签

标签	实现功能
$变量名	<?=$变量名?>
{$变量名}	<?=$变量名?>
{subtemplate 字模板名称}	加载字模板内容，与当前模板成为一个缓存文件
{lang 语言包索引}	加载索引对应的语言包文字
{block 调用标识}	数据调用
{blockdata }	调用数据
{ad 广告标识}	广告
{date 时间戳}	显示给定的时间戳对应的date日期
{avatar 用户id}	显示用户的头像
{eval php语句}	执行某个php语句
{csstemplate 样式标识}	加载CSS模板
{LF}	换行
{hook 钩子名称}	自定义钩子名称 注：两部分组成 mod名称_自定义名称
{template 模板名称}	加载模板
{echo 字符串}	输出一个字符串
{if 条件}	PHP中的if判断
{elseif 条件}	PHP中的elseif判断
{else}	PHP中的否则判断
{/if}	闭合if判断
{loop 变量1 变量2 变量3}	foreach(变量1 as 变量2 => 变量3)
{loop 变量1 变量2}	foreach(变量1 as 变量2)
{/loop}	foreach闭合
{常量}	输出一个常量

function_cache.php

2011年6月7日
9:48

缓存生成

2011年6月9日
9:12

调用原理

Source/function/funtion_cache.php
Updatecache()调用
Source/function/cache/cache_名称.php

如何添加一个缓存

- 1.建立source/function/cache/cache_缓存名称.php文件
- 2.写入函数build_cache_缓存名称
- 3.管理后台->工具->更新缓存

实例代码

如：建立cache_haoran.php

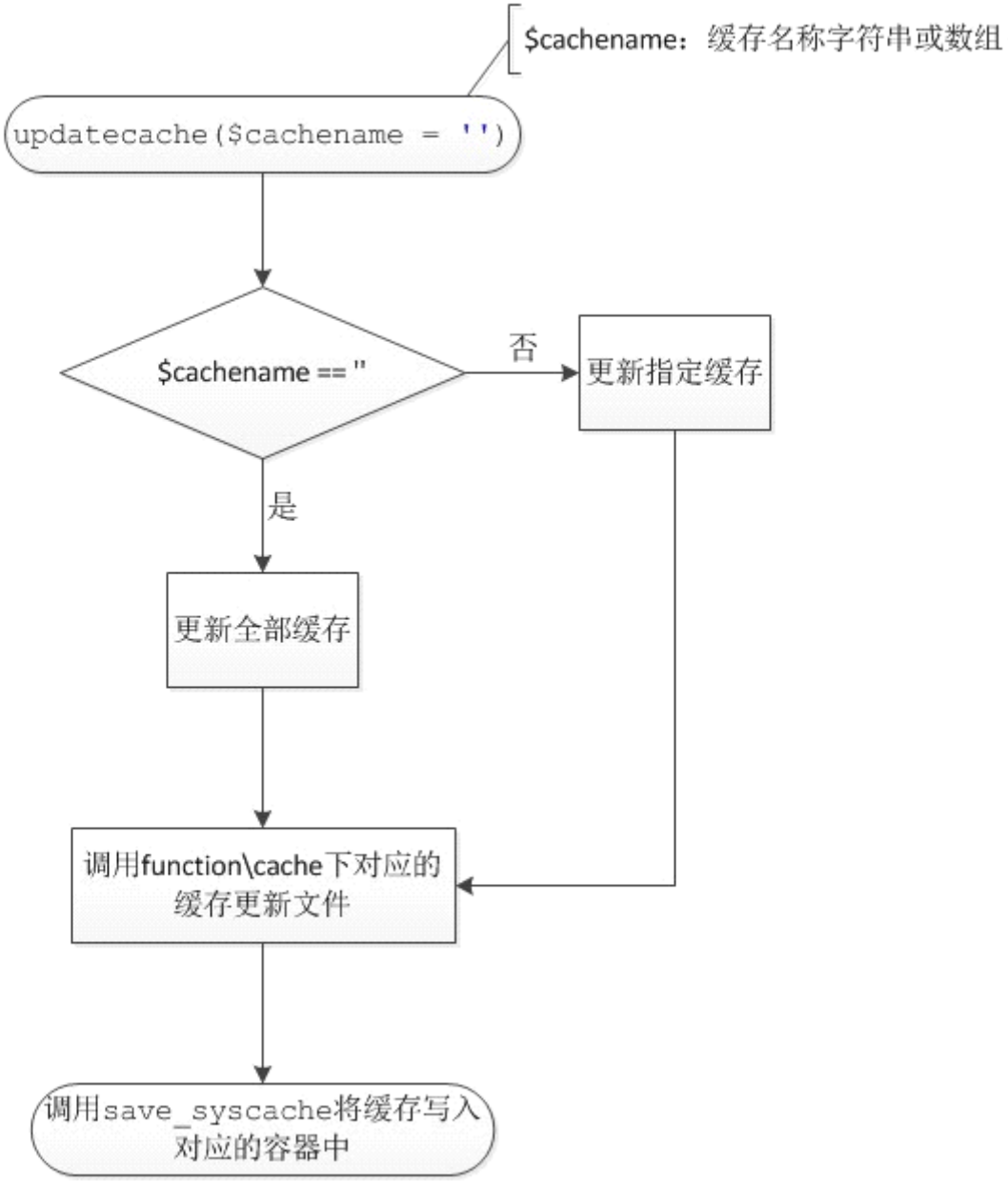
贴入以下代码

```
<?php

if(!defined('IN_DISCUZ')) {
	exit('Access Deny!');
}

function build_cache_haoran() {
	$query = DB::query('SELECT * FROM
	'.DB::table('common_member')." WHERE 1");
	while($result = DB::fetch($query)) {
		$data[$result['username']] = $result;
	}
	save_syscache('haoran', $data);
}

?>
```


$cachename：缓存名称字符串或数组
updatecache($cachename = '')
$cachename == ''
否
更新指定缓存
是
更新全部缓存
调用function\cache下对应的
缓存更新文件
调用save_syscache将缓存写入
对应的容器中

缓存自动载入

2011年6月7日
9:48

函数列举(function_core.php)

1.函数loadcache()
功能：防止多次加载，调用cachedata()函数，放入$_G中

2.函数cachedata()
功能：判断缓存类型，加载memory缓存，加载文件缓存，加载数据库缓存

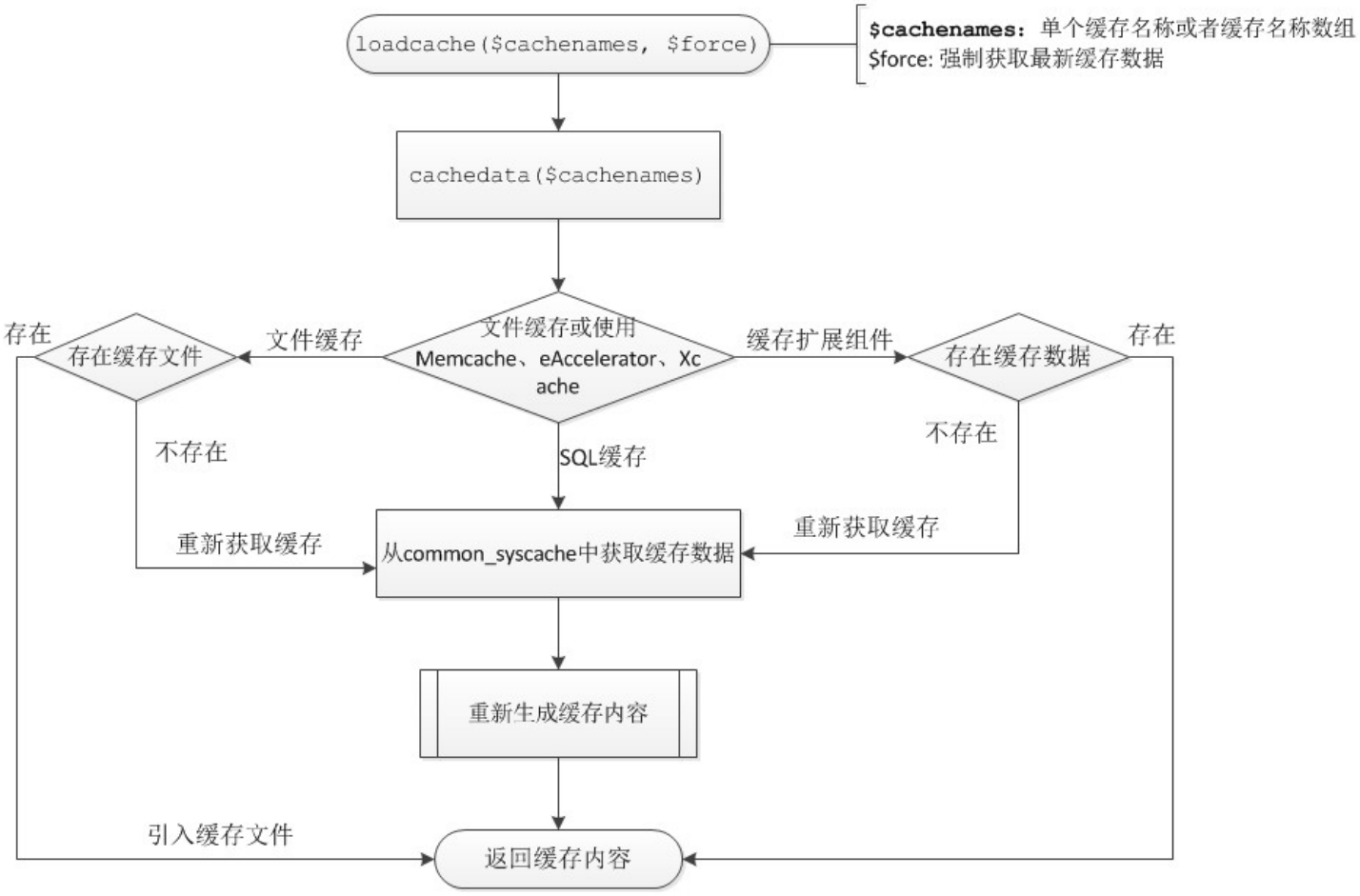
使用缓存

注：配合上面实例说的缓存的建立举例

由于，上面的代码生成的了名称为'haoran'的缓存
所以，只需要进行缓存的加载即可

实例代码：

```
<?php
    loadcache('haoran');
    print_r($_G['cache']['haoran']);
?>
```



Memcache缓存

2011年6月7日
9:49

安装准备：

1.安装php扩展memcache
2.安装memcache服务器软件

测试代码

```
<?php
	$mem = new Memcache();
	$mem->connect('127.0.0.1', 11211);

	$mem->set('haoran', '123', MEMCACHE_COMPRESSED, 0);

	$user = $mem->get('haoran');

	var_dump($user);
?>
```

plugin

2011年6月7日
9:49

后台管理

2011年6月7日
9:49

入口：根目录/admin.php
后台管理函数库：source/function/function_admincp.php

核心函数：
showsetting()
showtablerow()
showformheader()

代码举例

DIY拖动

2011年6月7日
9:49

DIY需要的知识

什么是框架？
什么是模块？

计划任务

2011年6月7日
9:50

静态页抓取

2011年6月7日
10:23

apc缓存

2011年6月19日
16:42

插件开发

2011年7月12日
16:36

1.不修改语言包的情况下，修改程序语言文字
2.曾经密码错误记录通知本人
3.楼层调整
4.前台勋章查询：帖子页面管理勋章
5.头衔用户组：根据任意一种积分，获取一个头衔，不相关权限
6.异地登录，登录成功提示：提示上次登录地
7.帖子查看密码
8.任务系统改进，自动弹出需要做新手任务的提示，普通任务可以选择是否弹出