**[php] TP3.2入口分析**

1. // 检测PHP环境，这里可以看出想要使用ThinkPHP3.2.1版本，PHP版本必须大于5.3，直接说明ThinkPHP3.2.1框架使用了大量PHP5.3以上版本的新特新
2. **if**(version\_compare(PHP\_VERSION,'5.3.0','<'))  **die**('require PHP > 5.3.0 !');
3. // 开启调试模式 建议开发阶段开启 部署阶段注释或者设为false
4. define('APP\_DEBUG',True);
5. // 定义应用目录
6. define('APP\_PATH','./Application/');
7. // 引入ThinkPHP入口文件
8. **require** './ThinkPHP/ThinkPHP.php';

从上面的例子中可以看出ThinkPHP的项目入口文件当真简单，这只是针对应用者而已，其实ThinkPHP框架正常运行需要定义大量常量。下面分析框架入口文件，并记录所有使用的配置和常量，如果出现bug也会做出记录，在整个项目分析完成后一起归纳分享。

**[php]** [view plain](http://blog.csdn.net/thinkercode/article/details/21332457" \o "view plain) [copy](http://blog.csdn.net/thinkercode/article/details/21332457" \o "copy)

1. // ThinkPHP公共入口文件
2. //以下两个记录为调试模式和记录日志时使用的。
3. //将当前时间记录到全局变量\_beginTime中
4. $GLOBALS['\_beginTime'] = microtime(TRUE);
5. //判断memory\_get\_usage是否启用，并定义一个常量
6. define('MEMORY\_LIMIT\_ON',function\_exists('memory\_get\_usage'));
7. //将获得当前内存使用值存入全局变量\_startUseMems中
8. **if**(MEMORY\_LIMIT\_ON) $GLOBALS['\_startUseMems'] = memory\_get\_usage();
10. /\*
11. \*这里有一个PHP5.3版本的新特新，const在5.3之后就变为可全局使用了，并且在使用命名空间时只能用const来定义
12. \*define为全局常量，const可以使用命名空间来避免不同命名空间的常量名称冲突。
13. \*const关键字定义常量的几个规则：
14. \*  1.建议使用大写字母定义
15. \*  2.常量不能使用$符号去声明
16. \*  3.必须符合define定义常量的命名规则
17. \*  4.常量值必须是一个常量表达式，而不能是一个不确定的结果，如变量或者函数等
18. \*  5.其值必须是标量类型（字符串、布尔值、整型、浮点型）
19. \*  6.如果在一个命名空间定义，在其它命名空间必须是命名空间方式访问常量
20. \*/
21. // 版本信息
22. **const** THINK\_VERSION     =   '3.2.1';
24. // URL 模式定义
25. **const** URL\_COMMON        =   0;  //普通模式
26. **const** URL\_PATHINFO      =   1;  //PATHINFO模式
27. **const** URL\_REWRITE       =   2;  //REWRITE模式
28. **const** URL\_COMPAT        =   3;  // 兼容模式
30. // 类文件后缀
31. **const** EXT               =   '.class.php';
33. // 系统常量定义,系统常量都做了判断，这也是我们自己可以在项目入口文件配置的常量值
34. //THINK\_PATH定义THINK\_PATH框架路径，\_\_DIR\_\_返回当前脚本所在目录
35. defined('THINK\_PATH')   **or** define('THINK\_PATH',     \_\_DIR\_\_.'/');
36. //APP\_PATH项目路径，dirname($\_SERVER['SCRIPT\_FILENAME']).'/')获取当前应用入口文件所在的目录
37. defined('APP\_PATH')     **or** define('APP\_PATH',       dirname($\_SERVER['SCRIPT\_FILENAME']).'/');
38. //APP\_STATUS应用环境常量，用来决定加载的主配置文件
39. defined('APP\_STATUS')   **or** define('APP\_STATUS',     '');
40. //APP\_DEBUG调试模式常量
41. defined('APP\_DEBUG')    **or** define('APP\_DEBUG',      false);
43. //下面这两个常量比较有意思，比之前的版本更加简单化支持sae环境
44. //APP\_MODE项目环境,sae新浪云，common普通模式
45. //STORAGE\_TYPE定义默认缓存文件类型
46. **if**(function\_exists('saeAutoLoader')){// 自动识别SAE环境
47. defined('APP\_MODE')     **or** define('APP\_MODE',      'sae');
48. defined('STORAGE\_TYPE') **or** define('STORAGE\_TYPE',  'Sae');
49. }**else**{
50. defined('APP\_MODE')     **or** define('APP\_MODE',       'common'); // 应用模式 默认为普通模式
51. defined('STORAGE\_TYPE') **or** define('STORAGE\_TYPE',   'File'); // 存储类型 默认为File
52. }
54. //RUNTIME\_PATH，项目运行时缓存主目录，可以更改其它常量实现与主缓存目录分离
55. defined('RUNTIME\_PATH') **or** define('RUNTIME\_PATH',   APP\_PATH.'Runtime/');   // 系统运行时目录
56. //LIB\_PATH定义系统核心类库主目录
57. defined('LIB\_PATH')     **or** define('LIB\_PATH',       realpath(THINK\_PATH.'Library').'/'); // 系统核心类库目录
58. //CORE\_PATH定义ThinkPHP框架类库目录
59. defined('CORE\_PATH')    **or** define('CORE\_PATH',      LIB\_PATH.'Think/'); // Think类库目录
60. //BEHAVIOR\_PATH定义ThinkPHP框架行为类库目录
61. defined('BEHAVIOR\_PATH')**or** define('BEHAVIOR\_PATH',  LIB\_PATH.'Behavior/'); // 行为类库目录
62. //MODE\_PATH定义系统应用模式目录，目前支持api、sae和common模式
63. defined('MODE\_PATH')    **or** define('MODE\_PATH',      THINK\_PATH.'Mode/'); // 系统应用模式目录
64. //Vendor定义第三方类库目录
65. defined('VENDOR\_PATH')  **or** define('VENDOR\_PATH',    LIB\_PATH.'Vendor/'); // 第三方类库目录
66. //Common应用公共函数库目录
67. defined('COMMON\_PATH')  **or** define('COMMON\_PATH',    APP\_PATH.'Common/'); // 应用公共目录
68. //CONF\_PATH定义应用配置文件目录
69. defined('CONF\_PATH')    **or** define('CONF\_PATH',      COMMON\_PATH.'Conf/'); // 应用配置目录
70. //LANG\_PATH定义应用语言目录
71. defined('LANG\_PATH')    **or** define('LANG\_PATH',      COMMON\_PATH.'Lang/'); // 应用语言目录
72. //HTML\_PATH定义应用静态文件目录
73. defined('HTML\_PATH')    **or** define('HTML\_PATH',      APP\_PATH.'Html/'); // 应用静态目录
74. //LOG\_PATH定义应用日志目录
75. defined('LOG\_PATH')     **or** define('LOG\_PATH',       RUNTIME\_PATH.'Logs/'); // 应用日志目录
76. //TEMP\_PATH定义缓存目录
77. defined('TEMP\_PATH')    **or** define('TEMP\_PATH',      RUNTIME\_PATH.'Temp/'); // 应用缓存目录
78. //DATA\_PATH定义应用数据缓存目录
79. defined('DATA\_PATH')    **or** define('DATA\_PATH',      RUNTIME\_PATH.'Data/'); // 应用数据目录
80. //CACHE\_PATH定义模版编译缓存目录
81. defined('CACHE\_PATH')   **or** define('CACHE\_PATH',     RUNTIME\_PATH.'Cache/'); // 应用模板缓存目录
83. //设置magic\_quotes\_runtime自动安全过滤状态，
84. //这个在php5.4及以上版本中移除了该设置
85. **if**(version\_compare(PHP\_VERSION,'5.4.0','<')) {
86. ini\_set('magic\_quotes\_runtime',0);
87. define('MAGIC\_QUOTES\_GPC',get\_magic\_quotes\_gpc()?True:False);
88. }**else**{
89. define('MAGIC\_QUOTES\_GPC',false);
90. }
91. //判断php与服务器之间所使用的接口类型
92. define('IS\_CGI',substr(PHP\_SAPI, 0,3)=='cgi' ? 1 : 0 );
93. //判断是否在windows系统下运行
94. define('IS\_WIN',strstr(PHP\_OS, 'WIN') ? 1 : 0 );
95. //判断是否在命令行模式下运行
96. define('IS\_CLI',PHP\_SAPI=='cli'? 1   :   0);
98. **if**(!IS\_CLI) {
99. // 当前入口文件路径文件名
100. **if**(!defined('\_PHP\_FILE\_')) {
101. echo $\_SERVER['HTTP\_HOST'];
102. **if**(IS\_CGI) {
103. //CGI/FASTCGI模式下
104. $\_temp  = explode('.php',$\_SERVER['PHP\_SELF']);
105. define('\_PHP\_FILE\_',    rtrim(str\_replace($\_SERVER['HTTP\_HOST'],'',$\_temp[0].'.php'),'/'));
106. }**else** {
107. define('\_PHP\_FILE\_',    rtrim($\_SERVER['SCRIPT\_NAME'],'/'));
108. }
109. }
110. //获取当前入口所在目录
111. **if**(!defined('\_\_ROOT\_\_')) {
112. $\_root  =   rtrim(dirname(\_PHP\_FILE\_),'/');
113. define('\_\_ROOT\_\_',  (($\_root=='/' || $\_root=='\\')?'':$\_root));
114. }
115. }
117. // 加载核心Think类
118. **require** CORE\_PATH.'Think'.EXT;
119. // 应用初始化
120. Think\Think::start();