



ThinkPHP Framework 1.5 Configurable Static Cache

ThinkPHP 1.5 可配置静态缓存设计

编写：ThinkPHP 文档组

最后更新：2008-12-16

目录

1	概述.....	3
2	启用静态缓存	3
3	静态规则定义	3
4	相关设置参数	5

1 概述

ThinkPHP 内置了静态缓存类，通过静态缓存规则定义来实现了可配置的静态缓存。本文描述了 ThinkPHP 的静态缓存的定义和使用。

2 启用静态缓存

ThinkPHP 内置了静态缓存的功能，并且支持静态缓存的规则定义。

要使用静态缓存功能，需要开启 `HTML_CACHE_ON` 参数，并且在项目配置目录下面增加静态缓存规则文件 `htmls.php`，两者缺一不可。否则静态缓存不会生效。

3 静态规则定义

静态规则文件的定义方式如下：

```
Return Array(  
  
    'ActionName'=>array('静态规则','静态缓存有效期','附加规则'),  
  
    'ModuleName:ActionName'=>array('静态规则','静态缓存有效期','附加规则'),  
  
    '*'=>array('静态规则','静态缓存有效期','附加规则'),  
  
    ...更多操作的静态规则  
)
```

静态缓存文件的根目录在 `HTML_PATH` 定义的路径下面，并且只有定义了静态规则的操作才会进行静态缓存，注意，静态规则的定义有三种方式，

第一种是定义全局的操作静态规则，例如定义所有的 `read` 操作的静态规则为

```
'read'=>array('{id}','60')
```

其中，{id} 表示取\$_GET['id'] 为静态缓存文件名，第二个参数表示缓存 60 秒

第二种是定义某个模块的操作的静态规则，例如，我们需要定义 Blog 模块的 read 操作进行静态缓存

```
'Blog:read'=>array('{id}',-1)
```

第三种方式是定义全局的静态缓存规则，这个属于特殊情况下的使用，任何模块的操作都适用，例如

```
'*'=>array('{$_SERVER.REQUEST_URI|md5}'), 根据当前的 URL 进行缓存
```

静态规则的写法可以包括以下情况

1、使用系统变量 包括 _GET _REQUEST _SERVER _SESSION _COOKIE

格式：{\$_xxx|function}

例如：{\$_GET.name} {\$_SERVER.REQUEST_URI}

2、使用框架特定的变量

例如：{:module} {:action} 表示当前模块名和操作名

3、使用_GET 变量

{var|function}

也就是说 {id} 其实等效于 {\$_GET.id}

4、直接使用函数

{|function}

例如：{|time}

5、支持混合定义，例如我们可以定义一个静态规则为：

```
'{id},{name|md5}'
```

在{}之外的字符作为字符串对待，如果包含有"/"，会自动创建目录。

例如，定义下面的静态规则：

```
{:module}/{:action}_{id}
```

则会在静态目录下面创建模块名称的子目录，然后写入操作名_id.shtml 文件。

静态有效时间 单位为秒如果不定义，则会获取配置参数 HTML_CACHE_TIME 的设置值

附加规则通常用于对静态规则进行函数运算，例如

```
'read'=>array('Think{id},{name}','60','md5')
```

翻译后的静态规则是 md5('Think'._GET['id'].'._GET['name']);

4 相关设置参数

和静态缓存相关的配置参数包括：

HTML_CACHE_ON 是否开启静态缓存功能

HTML_FILE_SUFFIX 静态文件后缀 惯例配置的值是 .shtml

HTML_CACHE_TIME 默认的静态缓存有效期 默认 60 秒 可以在静态规则定义覆盖

HTML_READ_TYPE 页面静态化后读取的规则

一种是直接读取缓存文件输出（readfile 方式 HTML_READ_TYPE 为 0）这是系统默认的方式，属于隐含静态化，用户看到的 URL 地址是没有变化的。

另外一种方式是重定向到静态文件的方式（HTML_READ_TYPE 为 1），这种方式下面，用户可以看到 URL 的地址属于静态页面地址，比较直观。