北京广视通达数字网络科技有限公司

2012年8月

**版本管理**

| 版本 | 日期 | 描述 | 修订者 |
| --- | --- | --- | --- |
| 1.0 | 2012-09-06 | 新建 | 邓雪平 |
|  |  |  |  |
|  |  |  |  |

目 录

[1. 概述 4](#_Toc365795461)

[2. function.php 5](#_Toc365795462)

[3. plugin.php 6](#_Toc365795463)

[3.1. 钩子、动作与过滤器（hook、action、filer） 6](#_Toc365795464)

[3.2. do\_action函数 7](#_Toc365795465)

[4. 应用部署 9](#_Toc365795466)

[4.1. MGP代理服务器2 9](#_Toc365795467)

[4.2. 118100mgpclient2 9](#_Toc365795468)

[4.3. MGP系统 11](#_Toc365795469)

[4.4. MGP代理服务器1 （10.1.2.12） 20](#_Toc365795470)

[5. 网络配置 26](#_Toc365795471)

[5.1. 网络拓扑图 26](#_Toc365795472)

[5.2. 网络结构描述 26](#_Toc365795473)

[5.3. IP地址配置 26](#_Toc365795474)

1. 概述

文件调用路径为：index.php->wp-blog-header.php->wp-load.php

index.php：几乎啥也没干，就是调用wp-blog-header.php

wp-blog-header.php：调用wp-load.php

wp-load.php：加载wp-config.php

wp-config.php：定义数据库连接信息以及相关变量，同时加载wp-settings.php

wp-settings.php：加载wp-include目录下相关文件，这些文件是一开始就要被加载的

在wp-settings.php中有如下do\_action的调用：

do\_action( 'muplugins\_loaded' );

do\_action( 'plugins\_loaded' );

do\_action( 'sanitize\_comment\_cookies' );

do\_action( 'setup\_theme' );

do\_action( 'after\_setup\_theme' );

do\_action( 'init' );

do\_action('wp\_loaded');

在do\_action之前需要先调用add\_action，在wp-include目录下的文件，则应该有大量add\_action的调用，添加相应的action

1. function.php
2. plugin.php

文件路径为：wp-include/plugin.php，此文件定义插件API函数。用于创建“动作”和“过滤器”以及相关钩子函数和方法。这些函数和方法在“动作”和“过滤器”调用时运行。

更多信息与实例，以及如何使用这些函数，可以参考：<http://codex.wordpress.org/Plugin_API>

* 1. 钩子、动作与过滤器（hook、action、filer）

WordPress中有两类钩子：

1. 动作
2. 过滤器
   * 1. action
     2. filter

所有的wp\_filter列表如附件所示：

所有wp\_filter列表的内容见附件：

一个wp\_filter的格式如下所示：

[10] => Array

(

[sanitize\_text\_field] => Array

(

[function] => sanitize\_text\_field

[accepted\_args] => 1

)

[wp\_filter\_kses] => Array

(

[function] => wp\_filter\_kses

[accepted\_args] => 1

)

)

[30] => Array

(

[\_wp\_specialchars] => Array

(

[function] => \_wp\_specialchars

[accepted\_args] => 1

)

)

* 1. add\_action函数

add\_action实际调用add\_filter

* 1. add\_filter函数
  2. do\_action函数
     1. do\_action(‘setup\_theme’)过程分析

if ( !isset($wp\_filter['all']) )

$wp\_current\_filter[] = $tag;

执行以上代码之后，$wp\_filter的内容如下：

wp\_current\_filter is :

Array

(

[0] => setup\_theme

)

do\_action函数中关键是如下几行代码：

do {

foreach ( (array) current($wp\_filter[$tag]) as $the\_ )

if ( !is\_null($the\_['function']) )

call\_user\_func\_array($the\_['function'], array\_slice($args, 0, (int) $the\_['accepted\_args']));

} while ( next($wp\_filter[$tag]) !== false );

而wp\_filter[‘setup\_theme’]的内容如下：

[setup\_theme] => Array

(

[10] => Array

(

[preview\_theme] => Array

(

[function] => preview\_theme

[accepted\_args] => 1

)

)

)

其中preview\_theme函数在wp-incoude/theme.php中定义

1. theme
   1. index.php

wp-content/themes目录保存所有的主题，每个主题一个目录，每个主题下有index.php页面。customizr主题的index页面内容很简单：

get\_header();

do\_action(‘\_\_fp\_block’);

do\_action(‘\_\_breadcrumb’);

get\_footer();

header.php

主题中的get\_header()，实际为加载header.php，header.php的内容如下：

<?php do\_action( '\_\_head' ); ?>

<body <?php body\_class(); ?> itemscope itemtype="http://schema.org/WebPage">

<header class="tc-header clearfix" role="banner">

<?php do\_action( 'before\_menu' ); ?>

这里的do\_action(‘\_\_head’);执行\_\_head相关的钩子函数，那在do\_action之前，必须要先add\_action。在类customizr/parts/class-main-header.php中有如下定义：

class TC\_header {

function \_\_construct () {

//html > head actions

add\_action ( '\_\_head' , array( $this , 'tc\_display\_head' ));

add\_action ( 'wp\_head' , array( $this , 'tc\_write\_custom\_css' ), 20 );

add\_action ( '\_\_favicon' , array( $this , 'tc\_display\_favicon' ));

//body > header actions

add\_action ( '\_\_logo\_title' , array( $this , 'tc\_display\_logo\_title' ));

add\_action ( '\_\_tagline' , array( $this , 'tc\_display\_tagline' ));

}

…

在类TC\_header的构造函数中有add\_action的动作，但有一个疑问，什么时候对类TC\_header中进行了实例化呢？这个可以参考customizr/function.php中的相关函数。

另外一个疑问是，

wp\_filter[\_\_head] :

Array

(

[function] => Array

(

[0] => TC\_header Object

(

)

[1] => tc\_display\_head

)

[accepted\_args] => 1

)

以上是wp\_filter[\_\_head]的内容，这里的function有两个元素，其中[0]是类对象，而[1]是函数，因此，在do\_action里使用的是：

if ( !is\_null($the\_['function']) )

call\_user\_func\_array($the\_['function'], array\_slice($args, 0, (int) $the\_['accepted\_args']));

当调用的是类中的函数时，使用call\_user\_func\_array函数。