

Wordpress自定义页面(非模版)

这种方法的自由度较高，并且可以创建非WordPress格式的URL，非常有用。比如我们要把 /test 转交给主题文件夹下的 /custom/test.php 来处理，就可以用这种方式来处理。这种方法用到 template redirect 钩子，template redirect 是 WordPress 在预处理好所有参数设置之后决定调用主题模板的时候调用的。

我们只需在主题的 function.php 文件的尾部加上：

```
function loadCustomTemplate($template) {
    global $wp_query;
    if(!file_exists($template))return;
    $wp_query->is_page = true;
    $wp_query->is_single = false;
    $wp_query->is_home = false;
    $wp_query->comments = false;
    // if we have a 404 status
    if ($wp_query->is_404) {
        // set status of 404 to false
        unset($wp_query->query["error"]);
        $wp_query->query_vars["error"]="";
        $wp_query->is_404=false;
    }
    // change the header to 200 OK
    header("HTTP/1.1 200 OK");
    //load our template
    include($template);
    exit;
}

function templateRedirect() {
    $basename = basename($_SERVER['REQUEST_URI'], '?' . $_SERVER['QUERY_STRING']);
    loadCustomTemplate(TEMPLATEPATH.'/custom/'.$basename.php);
}

add_action('template_redirect', 'templateRedirect');
```

这样就实现了 WordPress 查找 /custom 文件夹下的 php 文件，并且将相匹配的 URL 请求转交给对应的 php 文件来处理的效果，与此同时，这个 php 文件还保持了对 WordPress API 的调用，因此留给我们的空间非常大。

例如如果我们在 /custom 文件夹下创建一个 test.php 的文件如下：

```
<?php get_header();?>
<div id="main">
<div id="content">
<div id="post-999" class="page type-page status-publish hentry">
<h2 class="post-title">Iteblog page</h2>
<div class="post-content">
<?php echo "test"; ?>
<div class="clear"></div>
</div><!-- END .post-content -->
<div class="clear"></div>
</div><!-- #post-## -->
</div><!-- END #content -->
<?php get_sidebar(); ?>
<div class="clear"></div>
</div><!-- END #main -->
<?php get_footer(); ?>
```

那么只要输入/test 就能显示test，并且具有主题的框架和 CSS 样式，非常方便好用。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：[【】（）](#)