集成即将推出的模式

目录
简介
前置条件
逐步说明
在站点可见性设置更改时清除服务器缓存
在模板更改时清除服务器缓存
与其他插件同步 “即将推出 ”模式
从 WooCommerce 触发
从其他插件触发
双向同步与插件
使用选项覆盖实现单向绑定
自定义排除过滤器
在所有页面禁用 “即将推出 ”
仅在特定页面上禁用 “即将推出 ” 自定义分享链接
扩展 "仅应用于商店网页 " 设置

本指南为第三方开发者和主机提供商提供示例，说明如何将他们的系统与 WooCommerce
的即将推出的模式集成。 更多详情，请阅读 开发者博客文章 。有关站点可见性设置，请参阅 管理文档 。

简介
WooCommerce 的即将推出的模式允许您在进行开发时，暂时将您的站点对公众隐藏。
本指南将向您展示如何将此功能与您的系统集成，并在站点可见性设置更改时清除服务器缓存，以及如何将即
将推出的模式与其他插件同步。

前置条件

熟悉 PHP 和 WordPress 开发。

逐步说明

在站点可见性设置更改时清除服务器缓存

当站点的可见性设置发生更改时，可能需要清除服务器缓存，以应用更改并重新缓存面向客户的页面。
可以使用 update_option 钩子来实现此目的。

add_action('update_option_woocommerce_coming_soon', 'clear_server_cache', 10, 3); add_action('update_option_woocommerce_store_pages_only', 'clear_server_cache', 10, 3);

function clear_server_cache($old_value, $new_value, $option) {
// 实现清除服务器缓存的逻辑。
if (function_exists('your_cache_clear_function')) {
your_cache_clear_function();
}
}

在模板更改时清除服务器缓存

默认情况下，即将推出的页面设置了 Cache-Control: max-age=60 头部。 此设置允许 CDN
和其他缓存机制将页面缓存 60 秒，在高效性能和合理的更新时间之间取得平衡。

当用户更改即将推出的模板时，建议清除任何缓存，以便在客户端缓存过期时，更改立即生效。

您可以使用 save_post_wp_template、 save_post_wp_template_part 和 save_post_wp_global_styles 钩子来检测模板何时更新，并触发缓存清除。

add_action('save_post_wp_template', 'purge_cache_on_template_change', 10, 3); add_action('save_post_wp_template_part', 'purge_cache_on_template_change', 10, 3); add_action('save_post_wp_global_styles', 'purge_cache_on_template_change', 10, 3);

function purge_cache_on_template_change($post_id, $post, $update) {
// 检查模板是否与即将推出的模式相关联。
if ('coming-soon' === $post->post_name) {
// 实现清除服务器缓存的逻辑。
if (function_exists('your_cache_clear_function')) {

```
            your_cache_clear_function();
        }
    }
}
```

与其他插件同步“即将推出”模式

“即将推出”模式可以通过插件或应用程序进行编程同步。以下是一些示例用例：

- 与“维护模式”插件集成。
- 与托管服务商的“即将推出”模式集成。

从 WooCommerce 触发

您可以使用以下示例来执行操作，例如在更新“即将推出”模式选项时设置插件的“状态”：

```
add_action( 'update_option_woocommerce_coming_soon', 'sync_coming_soon_to_other_plugins', 10, 3 );

function sync_coming_soon_to_other_plugins( $old_value, $new_value, $option ) {
    $is_enabled = $new_value === 'yes';

    // 实现同步“即将推出”状态的逻辑。
    if ( function_exists( 'your_plugin_set_coming_soon' ) ) {
        your_plugin_set_coming_soon( $is_enabled );
    }
}
```

从其他插件触发

您可以使用以下示例，通过直接更新 `woocommerce_coming_soon` 选项，从另一个插件启用或禁用 WooCommerce“即将推出”模式：

```
function sync_coming_soon_from_other_plugins( $is_enabled ) {
    // 检查用户权限。
    if ( ! current_user_can( 'manage_options' ) ) {
        wp_die( '您没有足够的权限访问此页面。' );
    }

    // 设置“即将推出”模式。
    if ( isset( $is_enabled ) ) {
```

```
        update_option( 'woocommerce_coming_soon', $is_enabled ? 'yes' : 'no' );
    }
}
```

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ `update_option` □□□□□□□ `sync_coming_soon_from_other_plugins` □

```
add_action( 'update_option_woocommerce_coming_soon', 'sync_coming_soon_to_other_plugins', 10, 3 );

function sync_coming_soon_to_other_plugins( $old_value, $new_value, $option ) {
    $is_enabled = $new_value === 'yes';

    // □□□□“□□□□”□□□□□□□
    if ( function_exists( 'your_plugin_set_coming_soon' ) ) {
        your_plugin_set_coming_soon( $is_enabled );
    }
}

function sync_coming_soon_from_other_plugins( $is_enabled ) {
    // □□□□□□□□
    if ( ! current_user_can( 'manage_options' ) ) {
        wp_die( '□□□□□□□□□□□□□□□□□□' );
    }

    if ( isset( $is_enabled ) ) {
        // □□□□□□□□□□□□□□□□□□□□
        remove_action( 'update_option_woocommerce_coming_soon', 'sync_coming_soon_to_other_plugins', 10, 3 );

        // □□“□□□□”□□□
        update_option( 'woocommerce_coming_soon', $is_enabled ? 'yes' : 'no' );

        // □□□□□□□□
        add_action( 'update_option_woocommerce_coming_soon', 'sync_coming_soon_to_other_plugins', 10, 3 );
    }
}
```

使用选项覆盖实现单向绑定

我们还可以通过覆盖 `woocommerce_coming_soon` 选项，从另一个插件中以编程方式绑定 “即将推出 ”选项。
这样做具有优势，因为它简化了状态管理，并防止了可能出现的同步问题。

在以下示例中，我们通过覆盖 `woocommerce_coming_soon` 选项，从另一个插件绑定 “即将推出 ”选项。

```
add_filter( 'pre_option_woocommerce_coming_soon', 'override_option_woocommerce_coming_soon' );

function override_option_woocommerce_coming_soon( $current_value ) {
    // 实现您的逻辑以同步“即将推出”状态。
    if ( function_exists( 'your_plugin_is_coming_soon' ) ) {
        return your_plugin_is_coming_soon() ? 'yes' : 'no';
    }
    return $current_value;
}

add_filter( 'pre_update_option_woocommerce_coming_soon',
'override_update_woocommerce_coming_soon', 10, 2 );

function override_update_woocommerce_coming_soon( $new_value, $old_value ) {
    // 检查用户权限。
    if ( ! current_user_can( 'manage_options' ) ) {
        wp_die( '您没有足够的权限访问此页面。' );
    }

    // 实现您的逻辑以同步“即将推出”状态。
    if ( function_exists( 'your_plugin_set_coming_soon' ) ) {
        your_plugin_set_coming_soon( $new_value === 'yes' );
    }
}
```

自定义排除过滤器

开发者可以添加自定义排除规则，以绕过 “即将推出 ”保护。 这对于排除某些情况非常有用，例如始终在特定 IP
地址上绕过 “屏幕 ”，或者使特定的 “网页 ”可用。

在所有页面禁用 “即将推出 ”

如果存在其他功能与 WooCommerce 的“即将推出 ”模式类似，可能会导致意外冲突。
可以通过排除所有面向客户的 “页面 ”来禁用 “即将推出 ”模式。 以下是一个示例：

```
add_filter( 'woocommerce_coming_soon_exclude', function() {
    return true;
}, 10 );
```

仅在特定页面上禁用 “即将推出”

使用以下示例，根据 “页面”的 ID 排除某个 “页面”。将 `<page-id>` 替换为您的 “页面”标识符：

```
add_filter( 'woocommerce_coming_soon_exclude', function( $is_excluded ) {
    if ( get_the_ID() === <page-id> ) {
        return true;
    }
    return $is_excluded;
}, 10 );
```

自定义分享链接

以下示例显示了如何与自定义分享代码集成。我们建议使用 Cookie 或其他存储机制来持久化访问，以便用户在浏览网站时保持访问状态：

```
add_filter( 'woocommerce_coming_soon_exclude', function( $exclude ) {
    // 实现您的逻辑以获取和验证分享代码。
    if ( function_exists( 'your_plugin_get_share_code' ) && function_exists(
'your_plugin_is_valid_share_code' ) ) {
        $share_code = your_plugin_get_share_code();
        if ( your_plugin_is_valid_share_code( $share_code ) ) {
            return true;
        }
    }

    return $exclude;
} );
```

扩展 "仅应用于商店网页" 设置

当使用 `Apply to store pages only` 设置时，您可能希望将自定义网页添加到商店网页列表中，这些网页将受到 “即将推出”模式的限制。您可以使用以下示例来添加自定义网页：

```
add_filter( 'woocommerce_store_pages', function( $pages ) {
    $page = get_page_by_path( 'your-page-slug' );
```

```
    if ( $page ) {
        $pages[] = $page->ID;
    }
    return $pages;
} );
```

□□□ #2
□ □□□ □□□ 2026-03-09 11:53:58 UTC
□ □□□ □□□ 2026-03-09 11:53:58 UTC