Toggly WordPress

Plugin · host setup · templates · context · WP-Cron

Install from WordPress root

```
composer require toggly/wordpress
mkdir -p wp-content/plugins wp-content/mu-plugins
ln -s ../../vendor/toggly/wordpress \
  wp-content/plugins/toggly
```

Only create the symlink if its destination is absent. Entry point: wp-content/plugins/toggly/toggly.php. Prepare the MU plugin before activation.

MU host setup

WP-CONTENT/MU-PLUGINS/TOGGLY-HOST.PHP

```
<?php
require_once ABSPATH . 'vendor/autoload.php';
add_filter('cron_schedules',
  static function (array $s): array {
    $s['toggly_refresh_interval'] = [
      'interval' => 300,
      'display' => 'Every 5 Minutes',
    ];
    $s['toggly_send_interval'] = [
      'interval' => 60, 'display' => 'Every Minute',
    ];
    return $s;
});
add_filter('pre_option_toggly_snapshot_provider',
  static fn () => 'none');
add_action('plugins_loaded',
  static function (): void {
  $options = get_option('toggly_settings', []);
  if (!empty($options['app_key'])) {
    do_action('toggly_refresh_features');
  }
}, 20);
```

Root autoload path assumes the layout above. The SDK initializes on plugins_loaded. Do not call init twice.

Activate and configure

Activate Toggly Feature Management under Plugins. Settings → Toggly: set App Key, exact Environment and https://definitions.toggly.io/ Base URL. Select signed definitions to match your app, save, then make a new request.

Snapshot choice

The MU recipe selects none, bypassing optional transient caching with PSR Simple Cache 3. It gives up persistence: each new request needs refresh; API failure leaves no saved definitions in a new process. Default transient storage requires a compatible cache interface.

First flag

```
<?php
$on = toggly_is_enabled(
  'new-header');
?>
<?php if ($on): ?>
  <p>New header</p>
<?php else: ?>
  <p>Original header</p>
<?php endif; ?>
```

Place in a theme template after initialization. Create new-header in the selected app/environment. AlwaysOn shows New header; AlwaysOff shows Original header after refresh. Unknown flags are disabled by default.

Shortcode

```
[toggly_feature name="new-banner"]
  <p>Shown when enabled.</p>
[/toggly_feature]
```

name is required. Disabled or missing flags render nothing. JSON context is literal author data, not the authenticated viewer.

Toggly WordPress

Plugin · host setup · templates · context · WP-Cron

Authenticated request context

```
<?php
$user = wp_get_current_user();
$ctx = [
  'identity' => $user->exists()
    ? (string) $user->ID : null,
  'groups' => $user->exists()
    ? array_values($user->roles) : [],
];
$enabled = toggly_is_enabled(
  'premium-content', $ctx);
?>
<?php if ($enabled): ?>
  <p>Premium content</p>
<?php else: ?>
  <p>Standard content</p>
<?php endif; ?>
```

Use the same identity convention in Targeting. Ambient WordPress telemetry identity does not populate evaluation context. A top-level email is not a targeting identity. Flags do not replace WordPress capability checks.

Plugin API

```
$plugin = \Toggly\WordPress\TogglyPlugin
  ::getInstance();
$enabled = $plugin->isEnabled(
  'premium-content', $ctx);
```

Use after native initialization. Explicit context is forwarded to core evaluation. PHP has no ContextProperty or entity registration API.

WP-Cron

Early intervals let the SDK schedule refresh every 300 seconds and stats every 60 seconds. Verify with `wp cron event list` and run normal WP-Cron on the host.

```
do_action('toggly_refresh_features');
do_action('toggly_send_stats');
```

These actions refresh/send in the current process. Cron does not initialize the current page or drain another worker's counters. Request shutdown flushes owning telemetry.

Common pitfalls

Use the global toggly_is_enabled helper after plugins_loaded. Register cron intervals before native initialization. Without snapshots, keep explicit startup refresh. Do not use state-action names as automatic flag-change notifications or clear unrelated transients.

Resources

Complete WordPress guide · PHP core · PHP sheet · Samples catalog

PHP 7.4+ through 8.5 and WordPress 5.0+ (tested: PHP 8.5 / WordPress 7.1). Run composer check-platform-reqs for your dependency resolution.