

# Heartbeat Manager

Technical & User Documentation: Optimizing WordPress Performance via  
Heartbeat API Control

---

|                        |     |
|------------------------|-----|
| <b>Current Version</b> | 1.1 |
|------------------------|-----|

---

|                  |                                 |
|------------------|---------------------------------|
| <b>Developer</b> | Level Network (Nicola Loglisci) |
|------------------|---------------------------------|

---

|                            |               |
|----------------------------|---------------|
| <b>Min WP Requirements</b> | 5.0 or higher |
|----------------------------|---------------|

---

|                             |               |
|-----------------------------|---------------|
| <b>Min PHP Requirements</b> | 7.0 or higher |
|-----------------------------|---------------|

---

|                |                |
|----------------|----------------|
| <b>License</b> | GPLv2 or later |
|----------------|----------------|

## 1. Introduction

---

### 1.1 What is the WordPress Heartbeat API?

The **Heartbeat API** is a native WordPress feature introduced in version 3.6. It allows WordPress to establish continuous and periodic communication between the user's browser and the server, simulating a real-time connection.

It works by sending AJAX requests (via the `admin-ajax.php` file) at regular intervals. This technology is essential for features such as:

- **Autosave:** Automatically saving posts while you are writing.
- **Post Locking:** Notifying when another admin is editing a post to prevent overwriting.
- **Dashboard Notifications:** Real-time updates on sales (WooCommerce) or comments.

- **Session Management:** Keeping the user session active.

## 1.2 The Performance Problem

While useful, the default request interval is often very frequent (every 15-60 seconds). On high-traffic sites or servers with limited resources (shared hosting), this constant flow of AJAX requests can saturate the server's CPU resources.

A common symptom is abnormal usage of `admin-ajax.php`, which slows down the entire website for both administrators and visitors.

## 1.3 The Solution: Heartbeat Manager

**Heartbeat Manager** provides a simple and elegant solution to manage this behavior. It allows administrators to:

1. **Increase the interval** (reduce frequency) of the "heartbeat" in the Backend.
2. **Disable or reduce** the "heartbeat" in the Frontend, where it often consumes precious visitor resources without real necessity.

## 2. Key Features

---

- **Intuitive Configuration Panel:** Located under *Settings > Heartbeat Manager*.
- **Independent Control:** Separate management for the administration area (Backend) and the public site (Frontend).
- **Secure Implementation:** The plugin uses standard WordPress filters for the backend and a clean JavaScript method ( `wp.heartbeat.disable()` ) for the frontend. This approach drastically reduces the risk of conflicts.
- **Resource Optimization:** Immediate reduction of CPU load related to `admin-ajax.php`.
- **Security and Sanitization:** Code written following WordPress security best practices.
- **Internationalization:** Fully translation-ready (Text Domain: `heartbeat-manager`).

## 3. Installation and Activation

---

## 3.1 Standard Installation

1. Log in to the WordPress Dashboard.
2. Navigate to **Plugins > Add New**.
3. Click on **Upload Plugin** and select the plugin's .zip file.
4. Click **Install Now**.

## 3.2 Activation

After installation, click **Activate Plugin**.

Once activated, you can access the settings through the main menu: **Settings > Heartbeat Manager**.

# 4. Configuration Guide and Best Practices

---

The plugin is designed to be "set and forget." Below is the analysis of the available options.

### 4.1 Backend Frequency (Administration Panel)

This option adjusts the frequency of Heartbeat requests when a logged-in user is inside the control panel ( /wp-admin/ ).

- **Field Type:** Number (interval in seconds).
- **Range:** 15 - 300 seconds.

*Effect:* A higher value reduces server load but slows down autosaving and real-time notification updates.

#### Recommendations:

- Single-author sites: Set to **120** or **180** seconds.
- Multi-author sites or active E-commerce: Set to **60** seconds.

### 4.2 Frontend Frequency (Public Site)

This option regulates the API behavior when visitors navigate the public site. This is often the primary cause of slowdowns.

- **Field Type:** Dropdown menu (Select).
- **Options:** Disabled (Recommended), 60 Seconds, 120 Seconds.

**Recommendation: Select "Disabled".**

Most sites do not require real-time features on the frontend for visitors. By disabling it, you totally eliminate these AJAX calls, freeing up immense resources for the server. The method used preserves the stability of dependent plugins (such as WooCommerce).

## 5. Technical Notes for Developers

---

### Main Hooks Used

- `add_filter( 'heartbeat_settings', ... )` : Used to modify the interval in the backend.
- `add_action( 'wp_enqueue_scripts', ... )` : Used to inject the `wp.heartbeat.disable()` inline script in the frontend.

## 6. Changelog

---

### Version 1.1

- Drastically improved frontend management logic (disabling via secure inline script).
- Added escaping procedures ( `esc_attr` , `esc_html` ) and sanitization ( `intval` ).
- Fully internationalized (i18n ready).

### Version 1.0

- Initial release.

---

Heartbeat Manager is developed and maintained by [Level Network](#).

If you find this tool useful, please consider a  [donation](#).