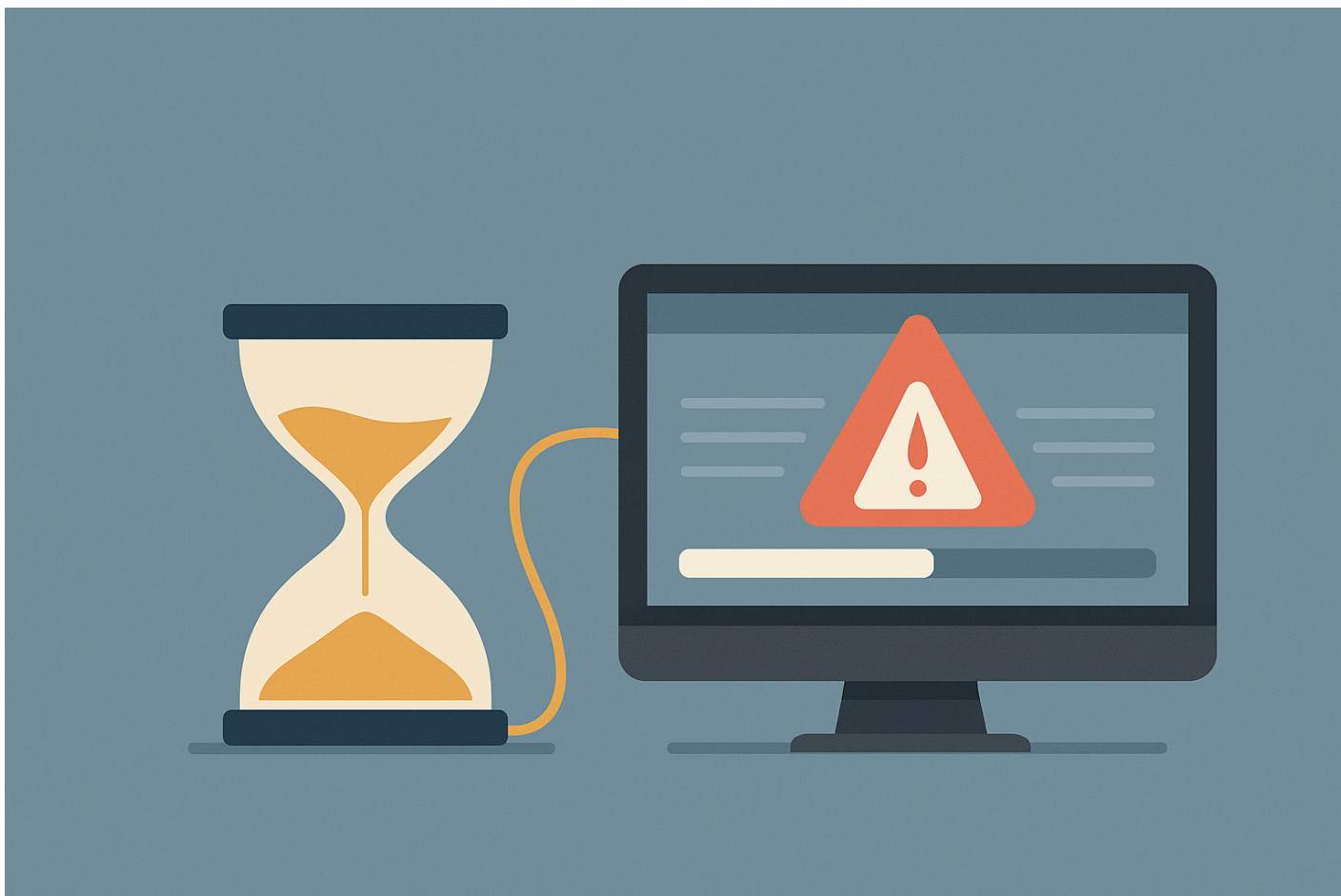


# Как да увеличим HTTP request timeout в WordPress

<https://go359.com/bg/uvelichi-http-request-timeout-wordpress-p1119>



## Как да увеличим HTTP request timeout в WordPress

Когато разработвате плъгини или теми, които комуникират с външни API—независимо дали става дума за плащания, доставки или импортиране на данни—често може да се сблъскате с проблеми с *timeout*. Това се случва, когато външен сървър отговаря прекалено бавно и WordPress прекъсва заявката, преди да е завършила.

Добрата новина? WordPress ви дава няколко начина да контролирате *timeout* стойността и да избегнете ненужни грешки при заявки към бавни, но функциониращи външни услуги.

В тази статия ще разгледаме как да увеличите *timeout* с помощта на два филтъра: `http_request_timeout` и `http_request_args`.

---

## Какво е Timeout при HTTP заявки в WordPress?

Когато WordPress прави HTTP заявка към външен сървър (например чрез `wp_remote_get()` или `wp_remote_post()`), той изчаква определено време за отговор. Ако този лимит бъде надвишен, заявката *изтича* (*times out*) и WordPress спира да чака.

Това е полезен защитен механизъм—предпазва сайта ви от забиване при проблеми с външни услуги. Но понякога стандартният *timeout* (5–15 секунди) просто не е достатъчен, особено ако API-то е натоварено или обработва голям обем от данни.

---

## Как WordPress изпраща HTTP заявки

WordPress предлага няколко лесни функции за изпращане на външни HTTP заявки:

```
$response = wp_remote_get('https://example.com/data');
```

```
$response = wp_remote_post('https://example.com/submit', [  
    'body' => [  
        'name' => 'John',  
        'email' => 'john@example.com',  
    ],  
]);
```

Тези функции използват WordPress HTTP API, който поддържа филтри за модифициране на параметри—включително *timeout*.

---

## Метод 1: Увеличаване на timeout с `http_request_timeout` (глобално или условно)

Ако искате **само да увеличите стойността на *timeout***, този филтър е най-чистият и лесен вариант.

```
add_filter('http_request_timeout', 'go359_tutorial_custom_timeout', 10, 2);
```

```
/**
 * Задаване на timeout според URL
 */
function go359_tutorial_custom_timeout($timeout, $url) {
    if (strpos($url, 'example.com') !== false) {
        return max(60, $timeout); // поне 60 секунди
    }

    return $timeout;
}
```

Този филтър:

- Се изпълнява преди заявката
- Проверява дали URL-ът съдържа example.com
- Увеличава timeout стойността, ако е прекалено ниска

Полезен е при прости случаи, където не са нужни допълнителни настройки.

---

## Метод 2: Прецизен контрол с http\_request\_args

Ако трябва да промените и други параметри (например connect\_timeout, заглавки, тяло на заявката и др.), използвайте http\_request\_args.

```
add_filter('http_request_args', 'go359_tutorial_change_timeout', 10, 2);
```

```
/**
 * Промяна на timeout и други настройки при заявки към example.com
 */
function go359_tutorial_change_timeout($args, $url) {
    if (strpos($url, 'example.com') !== false) {
        $current_timeout = !empty($args['timeout']) ? (int) $args['timeout'] : 0;
        $args['timeout'] = max(60, $current_timeout);

        if (empty($args['connect_timeout'])) {
            $args['connect_timeout'] = 30;
        }
    }

    return $args;
}
```

Този метод:

- Позволява настройка на `timeout` и `connect_timeout`
  - Може да се разшири с логика за POST/GET, заглавки и др.
  - Подходящ е при интеграции с по-сложни или по-бавни API услуги
- 

## Кой метод да използвате?

| Ситуация                                                            | Използвайте<br><code>http_request_timeout</code> | Използвайте<br><code>http_request_args</code> |
|---------------------------------------------------------------------|--------------------------------------------------|-----------------------------------------------|
| Само искате да увеличите <code>timeout</code>                       | Да                                               | Не е нужно                                    |
| Искате да таргетирате конкретен домейн                              | Да                                               | Да                                            |
| Искате да зададете <code>connect_timeout</code> или други параметри | Не                                               | Да                                            |
| Изграждате сложна логика за заявки                                  | Ограничено                                       | Пълен контрол                                 |

---

## Финални съвети

- Увеличаването на `timeout` **не решава** проблеми с бавно API, но предотвратява ненужно прекъсване при легитимно забавяне.
- Винаги тествайте с реални заявки, особено ако проектът ви зависи от външна услуга.
- Не използвайте дълги `timeout`-и като заместител на диагностика при проблеми с DNS или недостъпни сървъри.