

GREENRIVERS Maintenance Mode

Magento module maintenance mode

Table of Contents

- I Requirements.....3
- II Installation.....3
 - 1) Module.....3
- III Usage.....3
 - 1) Config.....3
- III Dev notes.....6
 - 1) Backend access.....6
 - 2) Commands.....6
 - 3) Log files.....7

I Requirements

- ✗ PHP >= 7.1
- ✗ Magento >= 2.3

II Installation

1) Module

```
composer require greenrivers/maintenance-mode  
php bin/magento module:enable GreenRivers_MaintenanceMode  
php bin/magento setup:upgrade  
php bin/magento setup:di:compile  
php bin/magento setup:static-content:deploy -f
```

III Usage

1) Config

Stores → Configuration → GREENRIVERS → Maintenance Mode

General

Enabled  Yes


General → Enabled - module activation

Settings

Frontend  Yes

Backend  Yes

Whitelist ips
[store view]

IP	Action
<input type="text"/>	
<input type="button" value="Add"/>	

Settings → Frontend – enable maintenance mode in frontend

Settings → Backend – enable maintenance mode in backend (warning: if You toggle this option, then admin dashboard can't be accessible)

Settings → Whitelist ips – list of addresses which will be excluded from maintenance mode (working on both areas: frontend & backend)

Custom page

Enabled ☒ Yes

Styles
[store view]

Content
[store view]

Show / Hide Editor

Paragraph **B** *I* U (x)

POWERED BY TINY

Custom page → Enabled – enable if You want to override default maintenance template

Custom page → Styles – add custom css rules for maintenance page (don't wrap content in <style> tag)

Custom page → Content – content of Your custom maintenance page

Cron

Frontend

Enabled  Yes

Frequency [global] ▼

Time start [global] ▼ : ▼ : ▼

Time end [global] ▼ : ▼ : ▼

Backend


Enabled  Yes

Frequency [global] ▼

Time start [global] ▼ : ▼ : ▼

Time end [global] ▼ : ▼ : ▼

Whitelist ips [store view]

IP	Action
<input type="text"/>	
<input type="button" value="Add"/>	

Cron → Frontend → Enabled – enable cronjob for maintenance mode in frontend

Cron → Frontend → Frequency – frontend cron frequency

Cron → Frontend → Time start – enable frontend maintenance time

Cron → Frontend → Time end – disable frontend maintenance time

Cron → Backend → Enabled - enable cronjob for maintenance mode in backend

Cron → Backend → Frequency - backend cron frequency

Cron → Backend → Time start - enable backend maintenance time

Cron → Backend → Time end - disable backend maintenance time

Cron → Whitelist ips - list of addresses which will be excluded from cronjobs maintenance mode (working on both areas: frontend & backend)

You must flush Magento cache after all of those operations.

III Dev notes

1) Backend access

Backend = Adminhtml = Admin dashboard

Keep in mind that after enable maintenance in backend and save config site should showing 503 page (maintenance mode: on).

There are some ways to again access admin:

- ➔ commands: maintenance:disable / greenrivers:maintenance:disable
- ➔ delete var/maintenance.flag file
- ➔ waiting for backend cron maintenance disabled time

2) Commands

There are 2 commands which works the same as operations in settings config section.

```
greenrivers
greenrivers:maintenance:disable      Disable maintenance mode in frontend/backend
greenrivers:maintenance:enable      Enable maintenance mode in frontend/backend
```

greenrivers:maintenance:enable

It has one argument:

- a) **whitelist-ips** – specify excluded addresses for maintenance mode (separated by comma)

It has also two flags:

- a) **frontend** – enable maintenance in frontend
- b) **backend** – enable maintenance in backend

To enable maintenance mode on both areas run command without flags.

Example:

We want to enable maintenance in frontend excluded localhost.

To achieve this run below command:

greenrivers:maintenance:enable --frontend --whitelist-ips=127.0.0.1

greenrivers:maintenance:disable

It has two flags:

- a) **frontend** – disable maintenance in frontend
- b) **backend** – disable maintenance in backend

To disable maintenance mode on both areas run command without flags.

Example:

We want to disable maintenance in backend.

To achieve this run below command:

greenrivers:maintenance:disable --backend

You must flush Magento cache after all of those operations.

3) Log files

Errors are logged in *var/log/greenrivers/maintenance_mode.log* file.