magmodules



Installation using Composer (recommended)

Alternate Hreflang Tags for Magento 2

Installation using Composer (recommended)

1 Step-by-Step Installation of Magento 2 Extensions via Composer

2 Add the Magmodules Composer Repository via SSH

3 Install and Activate the Extension via Composer

4 Update or Remove the Extension via Composer

5 Troubleshooting

Step-by-Step Installation of Magento 2 Extensions via Composer

Important: If the plugin was purchased from the Adobe Marketplace, this installation method does not apply. The steps outlined below are only for orders placed directly via our site.

Installing Magento extensions using Composer is the preferred method due to its ability to manage dependencies, ensure version compatibility, and streamline the update process. Composer simplifies the installation and management of PHP packages, making it an essential tool for Magento development.

Get the Details

1. Log in to your account to obtain your Composer Authentication Keys and Credentials, which are automatically generated after purchasing one of our extensions.

2. You can also create new Composer keys with a description if you want to change environments and keys.

Backup (Optional but Recommended)

We strongly recommend testing your installation in a development environment prior to releasing it to production. Additionally, ensure you have a full backup of your store available in case you need to perform a rollback.

Add the Magmodules Composer Repository via SSH

The next step is to add the **packages.magmodules.eu** repository to your Composer configuration.

Add Repository by SSH

Through an SSH connection, run the following command from your Magento installation directory to connect with our private repository:

```
composer config repositories.magmodules composer https://packages.magmodules.eu/
```

After executing this command, a blank line will appear, indicating that the repository has been successfully added to your `composer.json` file.

If you prefer to add the Composer repository manually, please refer to your account for the exact commands.

Install and Activate the Extension via Composer

Once the Magmodules repository has been added to your `composer.json`, you can install any packages available through it. If preferred, you may enable maintenance mode before continuing.

Download Extension

To download the extension into your Magento installation, run the following command from the Magento root directory:

```
composer require magmodules/m2-alternate-hreflang
```

If no Composer authentication keys are configured yet, you will be prompted to enter the username and password. These Composer Auth Keys can be obtained from your Magmodules account: Magmodules Auth Keys

Install Extension

After the extension has been downloaded, install and activate the `Alternate Hreflang Tags` extension by executing the following commands:

```
php bin/magento module:enable Magmodules_AlternateHreflang
php bin/magento setup:upgrade
```

If your Magento store is running in production mode, you must recompile and deploy static content:

```
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy -f
```

It is recommended to clear the Magento cache after installation:

```
php bin/magento cache:flush
```

Once these commands have been executed successfully, the extension is installed and ready for configuration. You can now configure the extension via: **Stores → Configuration → Magmodules → Alternate Hreflang**

Tags

Update or Remove the Extension via Composer

Once the extension is installed via Composer, you can easily update it, switch to a specific version, or remove it entirely. Composer ensures that dependencies are handled correctly, making updates safe and consistent across your Magento installation.

Update to the Latest Version

```
composer update magmodules/m2-alternate-hreflang
```

Update to a Specific Version

```
composer require magmodules/m2-alternate-hreflang:x.y.z
```

Remove the Extension

```
composer remove magmodules/m2-alternate-hreflang
```

After updating or removing an extension, run the following commands:

```
php bin/magento setup:upgrade
php bin/magento cache:flush
```

If you are in production mode, also execute:

```
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy -f
```

Troubleshooting

503 Service Temporarily Unavailable after installation

Cause: Maintenance mode still active.

Solution:

Disable maintenance mode:

```
php bin/magento maintenance:disable
```

Or manually remove the maintenance file:

```
rm var/.maintenance.flag
```

Memory Limit errors during compilation

Cause: PHP memory limit set too low.

Solution:

Temporarily increase memory limit:

```
php -d memory_limit=2G bin/magento setup:di:compile
```

Or permanently update `php.ini`:

```
memory_limit = 2G
```

Still having issues?

If you continue to experience problems:

1. Check the Magento error logs: `var/log/system.log` and `var/log/exception.log`.
2. Verify that all prerequisites are met (PHP version, required extensions, file permissions).
3. Check that all module files have been uploaded correctly (compare with the original zip file).
4. Contact Magmodules Support and provide:
    - Your Magento version
    - PHP version
    - Installation method (FTP/SSH)
    - Complete error messages
    - Steps you've already tried

For a complete overview of features and configuration options, see the Alternate Hreflang Tags extension on magmodules.eu

All articles for Alternate Hreflang Tags

Installation

1	**Installation using Composer (recommended) (current)**
2	Installation using the Adobe Marketplace
3	Install through FTP and SSH

Configuration

1	Quick Start Guide
2	Configuration Guide
3	CMS Pages and Homepage Setup

Troubleshooting

1	Troubleshooting

Background

1	Alternate Link Groups for Products and Categories
2	Best Practices
3	Understanding Hreflang Tags

