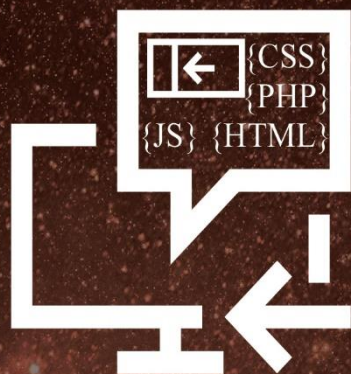


Tag Injector



TAG INJECTOR GIVES MERCHANTS THE ABILITY TO INJECT CUSTOM HTML, SCRIPTS AND STYLES INTO THEIR WEB PAGES.



NUBLUE
EXPERTS IN MAGENTO ECOMMERCE



4.87 OUT OF 5 STARS

BASED ON 180 REVIEWS
FROM REVIEWS.CO.UK



Contents

Installation Process	3
Installation via composer	4
Manual Installation	5
Setup & Configuration	6
Module Enable/Disable.....	6
Managing Tags	6
Variables & Handlebars.....	6
Additional Features.....	6
Magento Version	6
Support.....	6

Installation Process



Instructions for Installation

1. Installation via composer

You can install the extension via *System > Web Setup Wizard > Extension Manager* like other extensions in the Marketplace. You will need to have your access keys on hand as you will be prompted to enter them.

1. Select *Review and Install*, select the module to install and follow the Setup Wizard
2. In the CLI (Command Line Interface) the following needs to be entered
composer require nublue/taginjector
3. Enable Module

To get the module to appear in your Magento instance it needs to be enabled. To do this via the CLI you need to run:

```
php bin/magento module:enable Nublue_TagInjector
```

Then, to upgrade to module you need to run:

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

Finally, to compile run:

```
php bin/magento setup:di:compile
```

2. Manual Installation

1. Download the Files

Download the files from Magento Marketplace.

2. Place the Downloaded Files in *app > code* Folder

In the *app > code* folder (If you do not have the *code* folder, then you will need to create it), place the extension folder and files into this. If the files are zipped, they will need to be extracted before placing in *app > code*.

When successful, you should see the following structure:

app > code > Nublu > TagInjector > Module Folders

3. Enable Module

To get the module to appear in your Magento instance it needs to be enabled. To do this via the CLI you need to run:

```
php bin/magento module:enable Nublu_TagInjector
```

Then, install the necessary dependencies using composer:

```
composer require zordius/lightncandy:1.2.2
```

Then, to upgrade to module you need to run:

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

Finally, to compile run:

```
php bin/magento setup:di:compile
```

Database

During installation the extension creates default data in the table *Nublu_taginjector_tags*. Configuration for the extension can be found in the *core_config_data* table for row *taginjector/options/enable*.

Setup & Configuration

1. Module Enable/Disable

To enable or disable the extension go to *Stores > Configuration > Nublu > TagInjector* in the admin panel.

Note: Disabling the module via the admin does not remove the module files from *app > code*. For this you will need to use the `module:disable` command via the CLI. If you installed the extension via composer you can use the `module:uninstall` command. If you then need to enable the module again you will need to follow the Installation Process again.

2. Managing Tags

Tags can be added, edited and removed from the *Content > Nublu Tag Injection* menu in the admin panel.

Adding New Tags

To add a new tag, click on the *Create New Tab* button to view the following form:

The form consists of the following fields and controls:

- Enable Tag:** A toggle switch currently set to 'Yes'.
- Tags Name:** A text input field with a red asterisk. Below it is a hint: "Enter a recognisable name for the tag. This is only used in the admin panel."
- Page Type:** A dropdown menu with a red asterisk, currently showing "-- Please Select --". Below it is a hint: "Select which page the tag will show on."
- Inject Location:** A dropdown menu with a red asterisk, currently showing "-- Please Select --". Below it is a hint: "Select where on the page to apply the tag."
- Tag Type:** A dropdown menu with a red asterisk, currently showing "-- Please Select --". Below it is a hint: "Select the type of the tag you wish to apply."
- Priority:** A text input field with a red asterisk. Below it is a hint: "Defines the order at which tags are injected into the page."

Enable Tag: By default this is 'Yes'. Click this to set it to 'No' and disable the extension.

Tags Name: A recognisable name you can use to identify the tag in the admin grid.

Page Type: Specify which page type the tag is injected on. You can select *All* pages or a specific page type such as *Product*, *Category*, *Checkout*, *Cart*, *CMS* and *Account* pages.

Inject Location: Specify where on the selected page/s the tag is injected. For example, injecting the tag into *Head* will inject the tag into the end of the `<head></head>` section. You can also inject it into the pages *Body Top*, *Body Header* or *Body Footer*.

Tag Type: Specify the type of tag from *Script*, *Style*, *External Script*, *External Style* or *Custom HTML*. For example, *Script* will wrap the tag in `<script></script>` HTML tags.

Priority: Specify at what point to inject the tag. A higher number will inject tags later than tags with a lower priority.

Contents: This field shows depending on which Tag Type you selected. For example, for *External Scripts*, this will be the URL of the script to be injected, for *Style*, this will be CSS definitions and for *Custom HTML*, this will be raw HTML content.

Editing Tags

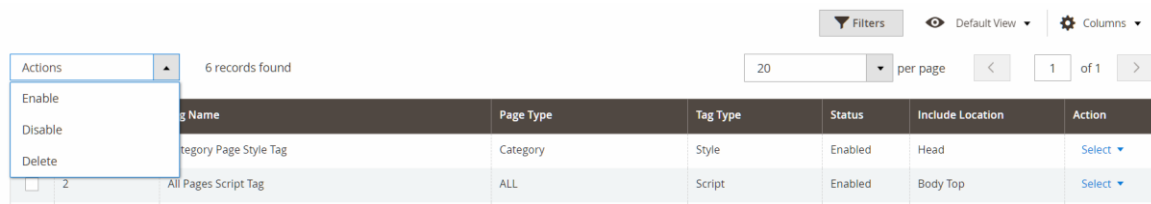
From the 'Action' column of the main Tag Injector grid, click *Select > Edit*.

Removing Tags

From the 'Action' column of the main Tag Injector grid, click *Select > Delete*. Then click 'OK' on the confirmation pop-up. You can also mass delete a set of tags by using the checkboxes in the first column of the table and then choosing *Disable* from the *Action* drop-down above the table.

Enabling and Disabling Tags

Tags can be enabled or disabled individually from the edit form tag – see previous section titled 'Adding New Tags'. Alternatively, tags can be enabled or disabled in mass by using the *Action* drop-down above the table.



6 records found

20 per page

1 of 1

	Tag Name	Page Type	Tag Type	Status	Include Location	Action
<input type="checkbox"/>	Category Page Style Tag	Category	Style	Enabled	Head	Select
<input type="checkbox"/>	All Pages Script Tag	ALL	Script	Enabled	Body Top	Select

3. Variables & Handlebars

Tag Injector allows you to use 'Smarty' variables to get dynamic information that has specific context of the page you are on. For example, on the Product pages you can get access to product SKU, name, price etc. and on the Checkout you can get access to order details, order total, discounts etc.

Variables and Handlebars help pages are provided as part of the extension. These can be accessed at the bottom of the *Edit Tag* or *Create New Tag* form when adding *Scripts* or *Styles*. You can read more about handlebars on their official page [here](#).

Additional Features



Magento Version

This extension works with Magento v2.1.x, v2.2.x and v2.3.x despite v2.1.x and v2.2.x and above using different formats for storing data in the database (v2.1.x *serialises* data whilst v2.2.x and above uses *JSON*). A *Helper* is used to compare the Magento version in the *composer.json* file against a string, with the data being parsed according to how it is stored.

Support

If you require support, please contact our support team at extensions@nublu.co.uk