AMAZZING FILTER

Installation and configuration guide

Installation

Module is installed in a regular way: go to **Module Manager** in BackOffice and upload the archive. We recommend uninstalling other layered navigation modules in order to avoid possible interference.

Product indexation

After installing the module you should index all products. Later, when you update products <u>in a native way</u>, they should be re-indexed automatically. So you don't have to re-index them additionally.

If products are updated in a <u>non-native way</u> (such as direct database modification), then you have to re-index them manually. On the **Indexation** tab you will find 2 buttons:

1. **INDEX MISSING PRODUCTS** - When this button is clicked, only products that haven't been indexed before will be indexed. Other products will not be processed.
2. **REINDEX ALL PRODUCTS** - When this button is clicked all products will be re-indexed, no matter if they have been indexed before or not.

Alternatively you can reindex products via **CRON TASKS:**

You can use standard **cron** commands like this one: `curl -L "cron_url"` or this: `wget -O /dev/null -q "cron_url"`

Don't forget to wrap `cron_url` in quotes. If you use curl, allow following redirects by adding `-L`: `curl -L "https://cron_url?token=xxx&id_shop=1&action=index-all"`

NOTE: indexation is a resource consuming task, so we DON'T recommend calling cron tasks very often, like "every 5 minutes", or so. 1-2 times per day should be enough. If you need to call cron tasks more often, you can <u>contact us</u>, and together we will try to find an optimal way of keeping indexation data up to date.

NOTE 2: When you add new shops in multishop system, or when you add bulk catalog price rules, you should reindex all products

Filter templates

After installing the module you will find a general filter template for all existing categories. If you don't associate any categories with this template, it will be displayed on <u>all</u> available category pages, including new ones, that may be created later.

If you want to display different filters on some specific categories, you can create another template and associate it with those categories.

 You can add/remove/update filter criteria and arrange them in required order by drag-n-dropping

Next to the **FILTERS** tab you will find **ADDITIONAL SETTINGS**. On that tab you can define custom settings that will override values specified in **General** settings. For example you can set custom *sorting* only for the current template, or you can set a different number of *products per page*, etc…

Except category pages, you can also display filter blocks on the following pages: ***New products, Specials, Bestsellers, Search results, Products by manufacturer, Products by supplier and Main page***.

All of these pages have adjustable templates that are activated automatically on module installation. If you don't want to display filters on some of those pages, you can simply deactivate the corresponding templates by clicking the green check mark next to the **Edit** button.

Module position and hook settings

Initially the filter is hooked to **displayLeftColumn** and it is positioned at the top of that hook. You can easily change the hook or update positions of other modules within that hook.
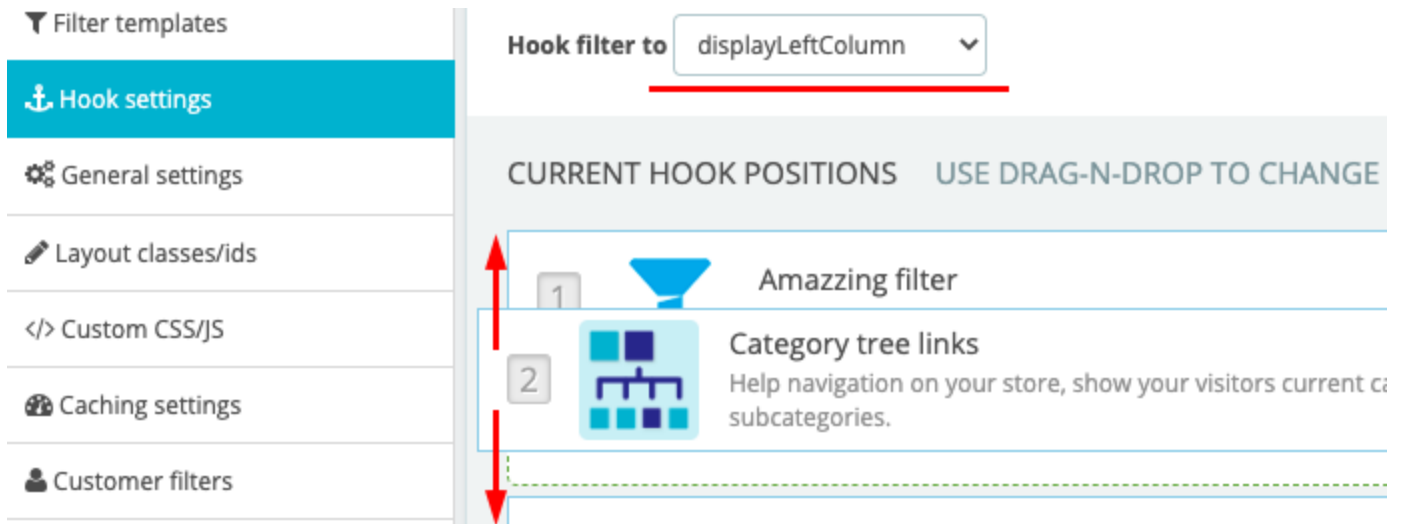
If you want to display the filter in a specific position that doesn't have predefined hooks, you can use custom hook `displayAmazzingFilter`.

That hook allows you to display the filter in any position in your theme. In order to start using that hook you should find the required tpl file and add the following code in the position where you want to display filter block: `{hook h='displayAmazzingFilter'}`

General settings

Most General Settings are self explanatory. Here is the description of some specific options:

Check availability based on selected attributes: If this option is enabled, availability is defined based on selected attributes. For example, a product has only 3 combinations in stock:
- Color: **RED +** Size: **M**
- Color: **RED +** Size: **L**
- Color: **GREEN +** Size: **M**
Basically this product has 2 colors and 2 sizes. But if a user selects Color: **GREEN** + Size: **L** then this product will be processed as non-available, because combination **GREEN + L** is not available. So, if you choose the option to "Exclude non-available products from the list", this product will be excluded.

NOTE: this option requires additional processing time. If you have a powerful server, you won't notice a significant time difference. However, it is recommended to use it only if you really need it.

Activate merged attributes/features: These options allow you to merge multiple attributes (or features) into one common value. For example you have many similar colors: BLUE, LIGHT BLUE, DARK BLUE, AZURE, NAVY BLUE, etc. You can merge them into one common value called BLUE, that will be displayed in the filter block as a single option.

After activating Merged attributes/features there will appear a new tab on the side panel. You can configure merged values in that tab.

Layout classes/ids

Here you can specify theme selectors that are required for seamless integration with the module. In most cases these will work out of the box without extra modifications.

Also, in this tab you can configure classes that are used to display icons in filter block, like ▼ or ✕
If your theme doesn't include an icon-font library, then you should activate the option **Load icon font.**
Otherwise don't activate it.

In some cases you might need to update icon classes. For example, your theme can use class `fa fa-plus` instead of `icon-plus`. Feel free to play with icon classes. You can easily **RESET** them later if required.

Adjusting the design of filter block

In most cases the design and behavior of filter block can be adjusted by adding <u>custom CSS/JS</u>:
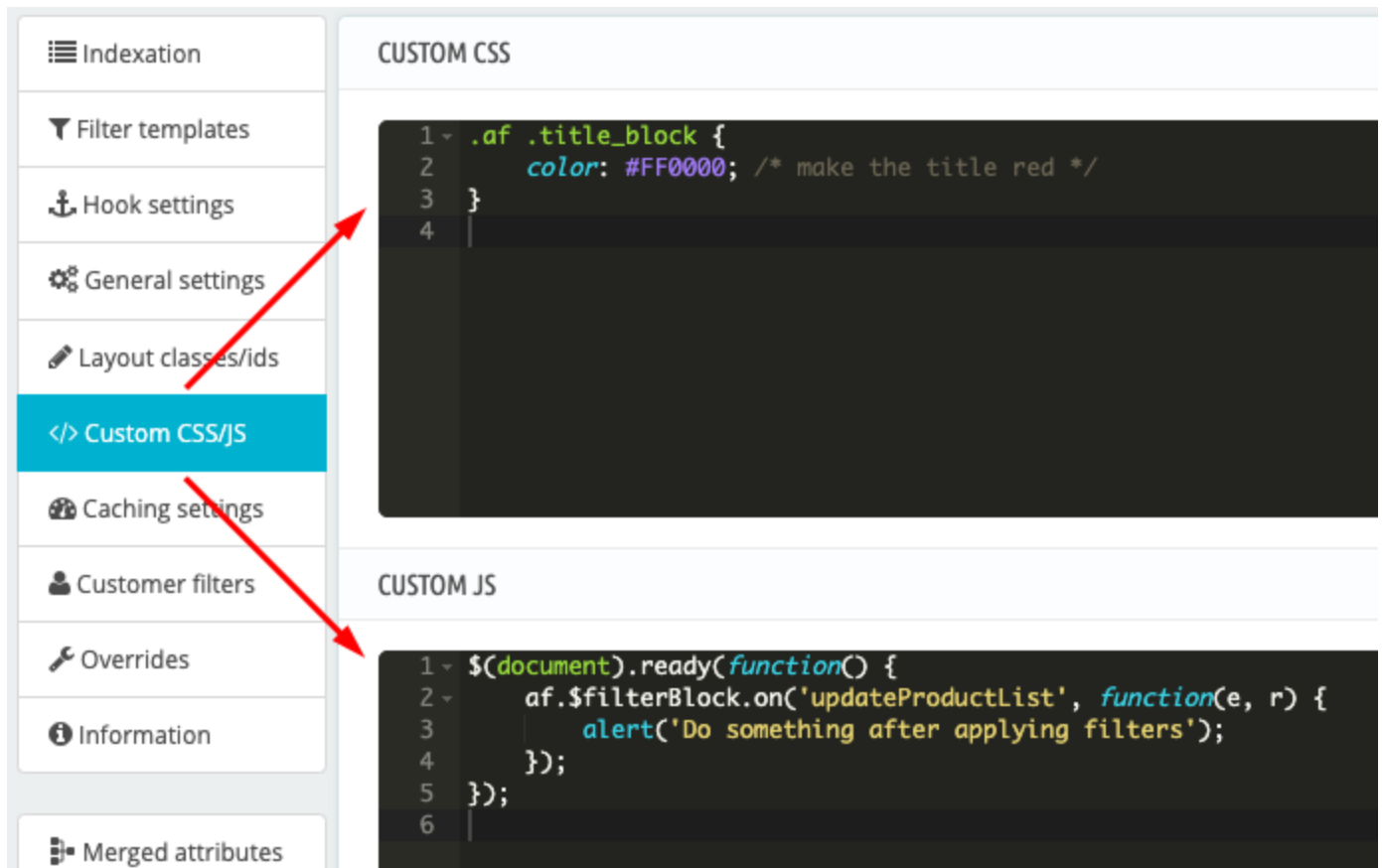
We DON'T recommend modifying any module files directly, or overriding them in your theme.

If you decide to override some files, then you have to check them every time after upgrading the module. For example, you override this file: `/views/templates/hook/amazzingfilter.tpl`

In the next module version some new essential variables and layout elements can be added/updated in original `/amazzingfilter.tpl`. But those variables and elements will not be updated in your custom `/amazzingfilter.tpl`. So after upgrading the module you have to make sure that overriden file has all required variables and layout elements.

You will understand that your custom `/amazzingfilter.tpl` is out of date when you get this warning on module configuration page:

Some of your customized files have been updated in the new version
- /amazzingfilter.tpl Update this file, and insert the following code to the last line: {* since 3.2.2 *}

It means that original `/amazzingfilter.tpl` was updated in the new module version. So you should update that file in your theme, and then add comment `{* since 3.2.2 *}` to the last line, as the warning says.

Caching settings

Caching is used to minimize repetitive requests to the database and complex calculations. You can activate caching for selected resources like **category options**, **attribute options**, **feature options** and **combinations availability**.

Here is how it works in a general way: when any user opens a page, some data is fetched from the database and processed in a regular way. After this data is saved in a cache-file. So, when the same data is requested next time by another user, there are no repeating requests to the database, and no processing involved, data is fetched from cache-file "as is" and it is ready to use.

In the current module version, caching is updated every hour. So once an hour data is processed in real time and saved in cache. During the next hour data is fetched directly from the cache, without requests to the database or complex calculations. When the caching hour is over, data is processed in real time again, caching is updated and used for the next hour, and so on…

This approach can decrease page loading time and filtration time on stores with many attributes, combinations and features. But you should be careful if data in your store is updated very often, because obviously caching doesn't use real-time data, it uses data that was saved within the last hour.

Besides resetting every hour, caching data it is automatically erased on the following native actions:

Category options - automatically erased, when a category is added/updated/deleted
Attribute options - automatically erased, when an attribute is added/updated/deleted
Feature options - automatically erased, when a predefined feature value is added/updated/deleted
Combinations availability - automatically erased, when new combinations are added/deleted, or product quantities are updated, or an order is placed.

If you need to reset cached data for any reason, you can always use the **Clear cache** button:

Customer filters

This functionality allows users to save predefined filters for more efficient browsing. For example, if someone is interested only in vehicle parts for **BRAND:** TOYOTA, **MODEL:** CAMRY, **YEAR: 2023**, they can select and save the required criteria in the "My filters" section. Once saved, these criteria will be automatically applied as filters on all category pages, eliminating the need for repeated selections.

In module settings you can define what filters can be saved by users, along with other related options.

Translations

Module translations can be edited by clicking on standard **Translate** button in the top right corner on module configuration page:

Additionally, you can define custom multilingual names for each filter group:

TECHNICAL TIPS

AUTOMATIC RE-INDEXATION ON MASS PRODUCT IMPORT/UPDATE:

If you add/update products using a bulk importing tool, you have to make sure that they are re-indexed properly after saving. Normally products should be re-indexed automatically if your bulk importing tool uses one of native programming methods for saving them, like `$product_obj->save()` or `$product_obj->add()` or `$product_obj->update()`.

If your tool doesn't use native methods for saving products, then you should find a way to re-index them after updating. You can choose one of the following alternatives:

1. Run indexation manually on module configuration page
2. Or call a cron task to re-index selected products
3. Or integrate native methods like `$product_obj->update()` in bulk importing tool
4. Or integrate the following hook in bulk importing tool:
   `Hook::exec('actionIndexProduct', ['product' => $ids_to_reindex]);`

Here is a simplified example of code demonstrating the use of `'actionIndexProduct'`:

```
$ids_to_reindex = [];
foreach ($products as $product) { // typical loop for bulk importing tools
    $reindex = false;
    // Perform bulk import actions. Set $reindex to true if
    // a product value that might be used in filters is updated
    // (attributes, features, categories, etc.)
    // ...
    if ($reindex) {
        $ids_to_reindex[] = $product['id_product'];
    }
}
Hook::exec('actionIndexProduct', ['product' => $ids_to_reindex]);
// $ids_to_reindex can be [1,2,3] or '1,2,3' or 3
```

NOTE: Indexation is a resource-consuming task. Therefore, we recommend re-indexing products only after updating parameters that might be used in the filter block or sorting selector. These parameters include price, weight, attributes, features, category associations, manufacturers, suppliers, etc.

Re-indexation is not required after updating other parameters such as description, meta title, friendly URL, etc.