



Complex Mobile & Web Software Development

Smart, flexible, secure, & reliable solutions that match high tech's quality standards.

[giantleaplab.com](http://giantleaplab.com)

---

# Brainy Filter User Guide

Demo version: [demo.brainyfilter.com](http://demo.brainyfilter.com)

Control panel: [demo.brainyfilter.com/admin](http://demo.brainyfilter.com/admin) (User: demo, Password: demo)

## Basic Settings

**Basic Settings** section is a set of settings of the basic module instance (a module instance defines settings of a particular layout, e.g. category page, search page, etc.) The basic settings are inherited by all other instances that can be managed in the Module Instances tab. In other words, the Basic Settings can be used as a template that can be overridden for particular pages by adding Module Instances with their own settings.

## Module Instances

In the **Module Instances** section layouts for various pages can be created. They can be used if there is a need to have different filter settings for different pages. If there is no instance for a layout it uses the Basic Settings configuration by default.

Please note, that the default layout Brainy Filter Layout should not be removed. The related page can be used to redirect to in some cases.

When creating a new instance the parameters below should be set.

- Layout — a layout to link the current instance to.
- Position — a position of the filter in the related page.
- Sort Order — sort order of the filter module relative to other modules in the same position.

In addition to the main parameters enumerated above each module instance can be configured with many different options explained below.

# Configuration of Basic Settings & Module Instances

*Please note, that if you are using Brainy Filter Pro only the Embedding tab will be shown. The full set of options is available only in the Brainy Filter Ultimate version.*

- **Categories** tab (available only for Module Instances) — allows to enable or disable the current filter module instance for particular categories. This way it is possible to have module instances with different settings linked to different categories.
- **Embedding** tab — includes CSS-selectors that are used by the filter to find and update the list of products and the pagination blocks in your theme without reloading the currently open page. The CSS-selectors may vary depending on the installed theme.
- **Filter Blocks Display** tab — allows to define the order in which different filters are displayed in the module. The order can be changed by dragging the filter items. There are also a few additional settings for each filter to collapse or expand it by default and select the type of display where applicable.
- **Filter Layout View** — offers settings for the module layout display.
  - Show Attribute Groups — Enables displaying of attribute group titles in the filter.
  - Enable Product Counts — shows the number of products for each attribute.
  - Hide Empty Attribute Values — hides attributes that have no products available. Attributes that have products but for which there are no products that would satisfy all other filtering options are shown with the product quantity set to “+0”.
  - Limit Number of Attributes Shown by Default — limits the number of attributes that can be shown by default. If the limit is reached, the Show More button is displayed instead of all extra attributes exceeding the limit.
  - Limit Blocks Height — limits height of every attribute block. In this case All attributes can be scrolled within blocks. If this option is enabled, the Limit Number of Attributes Shown by Default is disabled and cannot be used.
- **Data Submission** tab — defines when filtering should be triggered (and the search results updated) and includes an extra option for search results updating animation.
  - Any Filter Setting Change — search results will be updated instantly after any change in the filter.
  - Several Filter Settings Change — search results will be updated after a sequence of changes in the filter is completed. There is a timer that will trigger updating the filter search results updating after a user is inactive for certain amount of time.
  - The Apply Button is Pressed — search results will be updated only when the Apply button is pressed. The button will be shown only when this option is enabled.
  - Fade Out the Filter Panel During Requests — if enabled, the filter panel will be semi-transparent at the time of sending and receiving data to the server.
- **Attributes** tab — offers various settings for individual attributes.

- Default Settings for All Attributes — allows to hide or show all attributes and set a type for each of them.
- Individual Attribute Settings — allows to override the default attribute settings (see above) for particular attributes. It is possible to hide or show an attribute, change its type, or make it collapsed by default.
- **Options** tab — the same as the Attributes tab described above but is applicable to options.
- **Responsive View** tab — allows to show the filter in a popup on certain devices.
  - Show the Filter in a Popup in Responsive Mode — enables showing the filter in a popup when the screen width is less than the chosen number (see below).
  - Collapse Attribute Sections — collapses or expands attribute sections by default.
  - Max Filter Popup Width — sets the maximum width of the filter popup window.
  - Screen Width at Which the Filter is Moved into a Popup — the minimal screen width to show the filter in a popup (when the screen width is less or equals this setting the filter is shown in a popup).
- **Theme** tab — allows to select the filter title and the filter color palette.
- **Global Settings** tab (not available in the Module Instances tab) —
  - Hide Out of Stock Products — hides products that are out of stock.
  - CRON Secret Key — a secret key that should be used with the cron task link to update the module cache using a cron schedule. Please note, that the cron task is disabled when OpenCart is in the Maintenance Mode.
- **Values Ordering** tab — allows to sort attribute values.

## Installation

1. Make sure that the version of the Brainy Filter module you are going to install is compatible with your OpenCart version.
2. Install the Brainy Filter module zip-file through the Extension Installer section of OpenCart. If during the installation you face an error (usually due to lack of write permissions on the server) unzip the module file and copy the content of the upload folder into the root of your OpenCart site.
3. Go to the Extensions section (the Modules section if it's OpenCart 2.3 or later), find Brainy Filter in the list of extensions and click on the "Install" icon.
4. Open the filter control panel and make sure the default Brainy Filter layout has been automatically created (the Module Instances main menu item). If the default layout hasn't been created double check that you are using the correct version of the extension that suits your OpenCart and try to re-install.
5. Create an instance for the layout you need (Categories, Manufacturers, etc.).
6. Update modifications in the Modifications section of your OpenCart.

7. If after the steps above were performed but the filter doesn't update the list of products or pagination (but both work after reloading the entire page) you should manually set CSS selectors of your theme. See below.

## How to Create a CSS Selector

In order to find the right product list CSS selector click with the right mouse button at a product and choose Inspect Element (may be called slightly differently depending on your browser). This will open the developer tools. The element that is inspected (a product) will be highlighted. Go up through the tree of elements and find the element that wraps all products. Take that element Id or its class to use in the selector (see example below). Before this element you can also add its parent that wraps the entire content of the page. The same approach should be used for the pagination block as well.

### Example

Assume we have the following structure of a products page:

```
<div id="content">
  <div class="child products">
    <div class="individual-product">
      <div class="individual-product">
        ...
      </div>
    </div>
  <div class="child pagination"></div>
</div>
```

As described above, you should click on a product and go from an individual product to its parent element "products". Note that:

Use #content with "#" at the beginning if there is "id="

```
<div id="content"></div>
```

Use .content with "." at the beginning if there is "class="

```
<div class="content"></div> -
```

Thus we get the following selectors for a list of products and pagination:

1. #content .products
2. #content .pagination

Please note that the selectors you create should be unique. To check this use the console in the developer tools and type `$('#content .products')` — there is your selector inside. After

you press Enter it should return only one element. Otherwise you should create a more explicit selector.

For example, in the case of the default OpenCart theme, following the described steps you will get `#content .row` as a selector for products. If you execute `$('#content .row')` in console, you will receive several elements because in that theme they use the class `.row` for several different blocks including the product list and pagination. In cases like this one, it's necessary to find more explicit ids or names for the required blocks if there are any, or use CSS pseudo-classes ([https://www.w3schools.com/cssref/selector\\_nth-last-of-type.asp](https://www.w3schools.com/cssref/selector_nth-last-of-type.asp)). E.g. `#content .row:nth-last-of-type(1)` would be the last of the blocks.

## Signs of Incorrectly Chosen CSS Selectors

If you see that the filter causes the issues described above, it means you've incorrectly set the CSS selectors:

- Each time the filter is used the entire page is reloaded.
- Products or the pagination blocks are displayed incorrectly but after reloading the entire page all gets back to normal.

In this case just repeat the process of finding the selectors again and test via the developer tools console.



**trackabi**

Free time tracking &  
leave scheduling software