Smart Sidebars Slider

Developers Guide

Home: www.smartplugins.info/plugin/wordpress/smart-sidebars-slider
CodeCanyon: codecanyon.net/item/smart-sidebars-slider/7864196
Author: Milan Petrovic www.dev4press.com

Plugin Version: **2.5** - 2014.09.06
Documentation Version: **2.5** - 2014.09.06

Disclaimer

This guide assumes that you are familiar with PHP and WordPress development and API's used in WordPress, and that you have some development experience working with WordPress. There will be no explanations on how to use specific filters or actions, that is basic WordPress development practice.

Sidebar Activation Filter

This is most important filter that will be different for each sidebar.

sss_sidebar_activation_$id

This filter is used to hide or show sidebar. $id in the filter name is equivalent to Sidebar ID, and you can see full filter name for each sidebar on the rules page. This filter expects TRUE/FALSE bool values and plugin sends current visibility status, and sidebar object.
Example
If you want to show sidebar on category archive for term 'blog', year archive for year 2013 and still use the custom rules check here is the example. Example is for sidebar with ID 1.

```
add_filter('sss_sidebar_activation_1', 'sidebar_1_activation');
function 'sidebar_1_activation'($active) {
  if (!$active) {
    $active = is_category('blog') || is_year(2013);
  }

  return $active;
}
```

If the sidebar is already active (based on custom rules you might used), $active will be TRUE. If it is not TRUE, we then check for catrgory or year archive. Finally, we return new value for the activation variable.
For this activation check you can use any WordPress conditional you want to determine

current page.

Other Filters

There are several filters you can use to modify plugin behavior.

sss_sidebar_arguments

This filter is used to filter settings used for Smart Tab Drawer jQuery plugin used to display each slider sidebar added to the page. Extra parameter is current sidebar object.

sss_sidebar_registration_args

This filter is used to modify sidebar registration arguments. Extra parameter for this filter is current sidebar object. Plugin default styling is made for certain registration arguments, and changing this might break widgets layout in the sidebar.

sss_custom_style_classes

This filter is used to add extra CSS classes for slider styling outside of normal simple styler method for adding styles. Each style added this way needs style code, name and border width. Your actual class name for the CSS styles need to use this prefix: **std-style-**.
Here is the full example for adding one style (based on default styling). Full style class is **.std-style-my-custom-style**. But, to register our style, **.std-style-** is removed, and only **my-custom-style** is used for registration purposes.

CSS :

```
.std-style-my-custom-style .std-drawer {
    background-color: #fefefe;
    color: #333333;
    border: 2px solid #111111;
}

.std-style-my-custom-style .std-drawer a {
    color: #990000;
}

.std-style-my-custom-style .std-tab {
    background-color: #fefefe;
    border: 2px solid #111111;
    color: #990000;
    font-family: inherit;
    font-weight: 700;
    font-style: normal;
    font-size: 1.1em;
}

.std-style-my-custom-style .nano .nano-pane {
    background: rgba(51, 51, 51, 0.1);
}

.std-style-my-custom-style .nano .nano-slider {
    background: rgba(17, 17, 17, 0.7);
}
```

PHP:

```
add_filter('sss_custom_style_classes', 'my_sss_styles');
function my_sss_styles($styles) {
  $styles['my-custom-style'] = array('name' => 'My Style', 'border' => 2);
  return $styles;
}
```

Note, where we used our style code, and how we set border to 2. If your style defines border for tab and container, same border width number must be set in the style.