

Smart Audio Playlists

Developers Guide

Home: www.smartplugins.info/plugin/wordpress/smart-audio-playlists

Home: codecanyon.net/item/smart-audio-playlists/7654919

Author: Milan Petrovic www.dev4press.com

Plugin Version: **2.0** - 2014.08.07

Documentation Version: **2.0** - 2014.08.07

Disclaimer

This guide is for developers only and it assumes that you are familiar with PHP and WordPress development and API's used in WordPress, and that you have development experience working with WordPress. There will be no explanations on how to use every filter or action, that is basic WordPress development practice. For some filters and actions, you can get simple examples in this documentation.

Functions

There are functions you can override and replace with your own. This is usually done by placing your function version in the theme **functions.php** file. For your reference, default functions that can be overridden are located in file **override.php** in plugins **core** directory.

sap_archive_pagination()

This function renders pager for the archives page. It relies on the \$wp_query to get number of pages and whatever is needed.

sap_get_related_playlists(\$playlist_ID, \$count = 4)

This function is used to get list of related playlists for selected playlist. Default function does this using relations based on artists and genres terms. Result of this function must be array of WP_Post objects.

Actions

There are several actions available allowing hooking into plugin execution.

Initialization

These actions are executed while the plugin is initializing.

sap_init_plugin_settings

This action is executed after plugin initialize all its settings.

sap_init_plugin_objects

This action is executed after all plugin objects and image sizes are set.

sap_init_plugin_visuals

This action is executed after all plugin visuals are initialized: playlists styles, player skins and types and editions lists.

Playlists Player Widget

It has two more classes allowing you to add content before and after the list is displayed.

Both actions have 3 parameters: **\$instance** (settings for the current widget), **\$results** (list of all playlists objects) and **\$widget_id** (internal WordPress ID for the widget).

sap_widget_playlist_player_before

Run before playlists player.

sap_widget_playlist_player_after

Run after playlists player.

Track & Playlist Played

This actions are executed when track or playlist is played.

sap_played_track

When track is played, this action is run before the new stats are saved into database. It has two parameters: track ID and current stats array.

sap_played_playlist

When playlists is played, this action is run before the new stats are saved into database. It has two parameters: playlist ID and current stats array.

Filters

Plugin has many filters that you can use to modify various data.

Add Extra Track/Playlists fields

There are two filters to add new fields for tracks and playlist meta data. All added fields must be strings.

sap_fields_track

With this filter you can add meta fields for tracks. Each added field needs name and type (this must be set to string).

Example:

```
add_filter('sap_fields_track', 'my_fields_track');
function my_fields_track($fields) {
    $fields['my_field'] = array('name' => 'My Field', 'type' => 'string');
    return $fields;
}
```

sap_fields_playlist

With this filter you can add meta fields for playlists. Each added field needs name and type (this must be set to string).

Example:

```
add_filter('sap_fields_playlist', 'my_fields_playlist');
function my_fields_playlist($fields) {
    $fields['my_field'] = array('name' => 'My Field', 'type' => 'string');
    return $fields;
}
```

Playlist Release Date Format

Depending on the release date elements there are two filters available. If only year is set (month and day are set to 0), only year is returned. This is all for PHP date format.

sap_playlist_release_date_format_full

This is format for full date: year, month and day are all set. By default plugin uses date format from WordPress settings.

sap_playlist_release_date_format_month

This is format for date with month and year. By default shows month and year.

Control Playlists Output

These filters allow you to modify tracks sent to the playlists rendering, whole playlists setup object and CSS classes applied to the playlist player.

sap-playlist-player-tracks

This filter can be used to modify all tracks before they are sent for rendering. It has two parameters: **\$tracks** (array with tracks, each is associated array) and **\$playlist** (Playlist object for the current playlist page). This filter can be used to modify URL for each track ('src' element for each track), and maybe replace URL to point to CDN instead of local file.

sap-playlist-player-object

This filter is used to modify whole playlist object before rendering. This includes settings for the rendering with all tracks. It has two parameters: **\$object** and **\$playlist**.

sap-playlist-player-css

This filter is used to modify CSS classes applied to the Playlist control. It has two parameters: **\$classes** (array with CSS classes) and **\$playlist**.

Playlist templates classes

All templates made to render playlists and related content have filters that allow modification to classes added to the rendered output. Each filter here passes variable **\$class**, string with the current classes and one more variable with the name of the file it is used in (without .php extension, just name).

sap_template_single_classes

This is used in templates to display single playlist.

sap_template_archive_classes

This is used in templates to display playlists archive.

sap_template_index_classes

This is used in templates to display terms index.

Playlist widgets classes

This is similar to previous filters, but it has 4 parameters: **\$class**, **\$instance** (settings for the current widget), **\$results** (list of all playlists objects) and **\$widget_id** (internal WordPress ID for the widget).

sap_template_widget_playlists_classes

This one is a bit different. It is used for widget Playlists Lists.

sap_template_widget_playlist_player_classes

Same format as previous filter, but this is used for widget Playlist Player.

Theme Templates locating

There are two functions plugin uses to find main theme templates, and to find template parts. To override these functions, plugin uses two filters (one for each function).

sap_locate_theme_template

This is used in function that has to find theme templates. This include one of these files:

- archive-sapplaylist.php
- single-sapplaylist.php
- taxindex-sapplaylist.php

sap_locate_template

This is used in function that has to locate template parts (loop parts) that are used to display data, and can have multiple alternatives. All default templates for this purpose are in plugin directory **templates**. You can copy all of these files into your theme and customize it, and plugin will load them.

Registration of post types and taxonomies

This filters allow modification of data used to register plugins custom post types and custom taxonomies. Be careful with this, and use only if you are sure what this will do. If you make any changes to any of these many things can break, since plugin depends on the default settings for all of these. If you make changes here, you need to modify other things as well.

Very Important:

We can't provide any support for the plugin if these objects don't use default settings.

sap_register_taxonomy_genre

Filter settings to register Genre taxonomy.

sap_register_taxonomy_artist

Filter settings to register Artist taxonomy.

sap_register_post_type_track

Filter settings to register hidden Track post type.

sap_register_post_type_playlist

Filter settings to register main Playlist post type.