Menu Builder

This Module allows you to easily create custom menus/navigation lists in the ProcessWire Admin Panel using drag and drop. In the backend, it uses the nestedSortable jQueryUI plugin by Manuele J Sarfatti.

Features

- Visual menu builder
- Ability to create menus that do not mirror your ProcessWire Page Tree hierarchy/structure
- Menus can contain both ProcessWire pages and custom (external) links
- Create menu hierarchies and nesting via drag and drop
- Lock down menus for editing
- Easily apply CSS IDs and Classes to each and every menu item if you wish
- Optionally set custom links to open in a new tab
- Readily view the structure and settings for each menu and menu item
- For each menu, multiple configurable ways to add menu items from ProcessWire pages - PageAutocomplete, PageListSelectMultiple OR AsmSelect[default] AND ProcessWire Selector
- Using a Selector, you can search for pages to add, for example, template=basic-page, limit=20, sort=title.
- Batch edit menus
- Menus stored as pages (note: just the menu, not the items!)
- Menu items stored as JSON in a field in the menu pages (empty values not stored)
- For page fields, you can specify a selector to return only those specified pages for selection in the page field (only applicable to Asm and Autocomplete)
- For page fields, you can also add CSS classes and IDs as you add the items (similar to custom menu items)
- Menu settings for nestedSortable - e.g. maxLevels (limit nesting levels)
- Advanced features (e.g. add pages via selector, menu settings) permissible to superadmins only
- Delete single or all menu items without deleting the menu itself
- Easily render menus and breadcrumbs in the frontend using MarkupMenuBuilder
- Optionally create complex menu structures by returning only menu items using MarkupMenuBuilder and passing these to your custom recursive menu function
- Fully multi-lingual

How to Install

The module has two components:

- ProcessMenuBuilder: for creating menus in the ProcessWire Admin
- MarkupMenuBuilder: for displaying menus in the frontend (unstyled, unordered list by default)

1. Install the module from within the ProcessWire admin or download the module and copy the file contents to **/site/modules/MenuBuilder/**
2. In Admin, click Modules > check for new modules
3. Click install ProcessMenuBuilder. The module will automatically install MarkupMenuBuilder
4. Go to Setup > Menu Builder and start creating your menus

Note

- The module installs three fields: **'menu_items'**, **'menu_pages'** and **'menu_settings'** and one template **'menus'**. If any similarly named fields/template are already present on your site, the module will not install but throw an error instead. You would need to rename your fields/template first.
- To allow access to the Menu Builder admin, a non-superuser must have the permission **menu-builder**. The permission is created on install.
- Some Menu Builder admin options are only available to Superusers by default. Other users would require specific permissions as described below.

API

MarkupMenuBuilder has three methods available to users.

render()

This method renders a menu/navigation list of a specified menu. The method accepts two arguments/parameters:

```
render($menu, $options);
```

The first argument is not optional and can be a Page object, a title, name or id of a menu or an array of menu items returned from a menu's menu_items field. **Note that using titles or names in multi-lingual environments also works irrespective of the current user's language**. The

second argument is an optional array and will fall back to defaults if no user configurations are passed to the method.

The available **render()** options are. **Please note the possible values for each**.

```
$defaultOptions = array(

    'wrapper_list_type' => 'ul',// ul, ol, nav, div, etc.
    'list_type' => 'li',// li, a, span, etc.
    'menu_css_id' => '',// a CSS ID for the menu
    'menu_css_class' => '',// a CSS Class for the menu
    'submenu_css_class' => '',// CSS Class for sub-menus
    'has_children_class' => '',// CSS Class for any menu item that has children
    'first_class'=>'',// CSS Class for the first item in
    'last_class' => '',
    'current_class' => '',
    'default_title' => 0,// 0=show saved titles;1=show actual/current titles
    'include_children' => 4, // whether to include natural/pw children of menu items
in navigation; 0=never;1=in menu only;2=in breadcrumbs only;3=in both;4=do not show
    'm_max_level' => 1,// how deep to fetch 'include_children'
    'current_class_level' => 1,// how high up the ancestral tree to apply
'current_class'
    'default_class' => '',// a CSS class to apply to all menu items

);
```

renderBreadcrumbs()

This method renders a breadcrumb navigation of a specified menu. The method also accepts two arguments/parameters:

```
renderBreadcrumbs($menu, $options);
```

Similar to **render()**, the first argument is not optional and can be a Page object, a title, name or id of a menu or an array of menu items returned from a menu's menu_items field. This means that you only have to retrieve a menu once and pass that to both **render()** and **renderBreadcrumbs()**. **Note that using titles or names in multi-lingual environments also works irrespective of the current user's language**. The second argument is an optional array and will fall back to defaults if no user configurations are passed to the method. The options are very similar to those of **render()**. Hence, as applicable, you can create one array of options and pass it to both **render()** and **renderBreadcrumbs()**. The methods will pick up what's of relevance to them.

The available **renderBreadcrumbs()** options are shown below. **Please note the possible values for each**.

```
$defaultOptions = array(

    'wrapper_list_type' => 'ul',// ul, ol, nav, div, etc.
    'list_type' => 'li',// li, a, span, etc.
    'menu_css_id' => '',
    'menu_css_class' => '',
    'current_css_id' => '',
    'current_class' => '',
    'divider' => '»',// e.g. Home >> About Us >> Leadership
    // prepend home page at the as topmost item even if it isn't part of the
breadcrumb
    'prepend_home' => 0,// 0=no;1=yes
    'default_title' => 0,// 0=show saved titles;1=show actual/current titles
    'include_children' => 4,// whether to include natural/pw children of menu items in
navigation; 0=never;1=in menu only;2=in breadcrumbs only;3=in both;4=do not show
    'b_max_level' => 1,// how deep to fetch 'include_children'

);
```

getMenuItems()

This is a new method since version 0.1.5. The method greatly simplifies the creation of custom complex menus. Use this method instead of **render()** if you wish to have total control over your menu logic and markup. Examples of how to use this method can be found in these gists as well as in this post in the support forum. The method accepts three arguments:

```
getMenuItems($menu, $type, $options);
```

Similar to **render()**, the first argument is not optional and can be a Page object, a title, name or id of a menu or an array of menu items returned from a menu's menu_items field.

The second argument determines the type of items that the method will return. A value of 1 will return an array and one of 2 (the default) will return a WireArray Menu Object.

The third argument is similar to the $options passed to **render()** but accepts slightly only a limited number of options as noted below.

getMenuItems() Options

- From the options available to **render()** above, 5 also apply to **getMenuItems()**: *default_title, default_class, include_children, m_max_level and current_class_level*. In addition,

getMenuItems() also takes a number of options only applicable to it as outlined below.

- In this section the term navigation is used in the context of menus only.
- The term 'Class(es)' indicates that multiple CSS Classes can be applied, separated by space.

1. **default_title**: Controls whether to display Menu Builder saved menu item titles/labels versus displaying pages' actual/current titles. This is useful in scenarios where, for example, you need dynamic titles such as in a multi-lingual environment where you would want navigation labels to change depending on the current language. The default option is to display saved titles. To instead display actual titles, set option to **'default_title' => 1** in your options array.
2. **default_class**: **For menus only** and at all menu levels (i.e. nesting), a CSS Class(es) applied to each menu item (e.g. a Bootstrap or Foundation CSS Class).
3. **include_children**: Controls whether to display viewable descendant ProcessWire pages (children, grandchildren, etc.) of Menu Builder ProcessWire pages' navigation items at runtime as opposed to editing and saving them in Menu Builder admin. This is useful in a number of cases. For instance, you may wish to limit the number of menu items appearing in a particular menu in the the Menu Builder admin, e.g. in cases where these are many. This option can be applied globally (i.e. at a menu-level/$options level) and locally (at a menu-item-level). Item-level settings mean that only children of the specified item will be included. The 'include_children' option is disabled by default. Please see below for further details on how to use the feature.
4. **m_max_level**: This is a menu-only option related to the **include children** feature. It limits the depth from within which viewable descendant pages can be retrieved for display in a menu. The default is 1. This means that only fetch immediate children. A value of 2 means fetch both children and grandchildren, etc. The option can be applied globally and locally as explained previously.
5. **current_class_level**: Using this option, you can specify how high up the menu tree you want to apply the **'current_class'** to the current item's ancestors. The default is 1, meaning (if specified) apply the **'current_class'** to only the current item. A setting of 3 implies apply it to the current item, its parent and grandparent, etc. An option of 0 will apply **'current_class'** to all ancestors of the current page being viewed irrespective if that current page is part of the menu. In short, the option can be used to show some or all 'active/current' menu items at various levels in your menu. This option only applies to menus and not breadcrumbs.
6. **check_listable**: If set to 1, will not display items that are not (ProcessWire-) listable to the current user. The default is to show all items.
7. **maximum_children_per_parent**: If set to greater than 0, it will limit the number of child items fetched for each parent menu item using the **include children** option. The default is 0, meaning, all visible child items are fetched for a parent. Please note that child items set via the GUI are not included in the count. For instance, if **maximum_children_per_parent** is set

to *3* and a parent item has *1* child item set via the GUI, a total of *4* items will be returned for that parent, i.e. *3+4* items.

8. **get_total_children**: If set to 1, it will return a property with the number of child items each parent menu item has. The property accounts for both natural (ProcessWire) children as well any added via the GUI. The default is not to return counts.
9. **show_more_text**: This is used in conjuction with **maximum_children_per_parent**. It returns a string property with the text that should be displayed if a parent menu item has more (visible) child items than the limit set by **maximum_children_per_parent**. The default (English) string is **View More**.
10. **extra_fields**: Accepts an array of named ProcessWire fields and respective options whose values to return for each menu item as applicable. fault is not to return counts. Please see dedicated section below for how to use this feature.

Properties

Menu items returned via **getMenuItems()** have the following properties, as applicable. As previously mentioned, if *$type=1* (see above), **getMenuItems()** will return an array. The array index equivalents of the menu items' properties are indicated in parentheses. **Please note that if a propety is empty (including zero value), it will have no equivalent array index**. For menu items returned as objects, the properties can be accessed using $menuItem->propertName. It is also possible to query menu items using their property names. For instance:

```
$items = $menu->find("parentID=10");

$item = $menu->get("id=5");
```

1. **parentID(parent_id)**: The ID of the parent of a menu item. Please note that these IDs are for internal reference and do not correspond to ProcessWire page IDs! .An ID of **0** means the menu item has no parent and is therefore a top-tier menu item.
2. **pagesID(pages_id)**: The ProcessWire page ID of the menu item. If **0** it means the menu item is a custom menu item with a link external to ProcessWire.
3. **title(title)**: The title of the menu item. Please note that this may vary depending on the setting **default_title** above.
4. **url(url)**: The URL the menu item points to.
5. **newtab(new_tab)**: For custom menu items only. If value is 1, custom links will open in a new browser tab.
6. **cssID(css_itemid)**: A single menu item's CSS ID.
7. **cssClass(css_itemclass)**: A single menu item's CSS class(es).

8. **includeChildren(include_children)**: A value of 1 denotes that the menu item will be including its native (ProcessWire) children as menu items.
9. **menuMaxLevel(m_max_level)**: Specifies how deep child menu items are fetched using **include_children**.
10. **isParent(is_parent)**: Returns 1 if a menu item is a parent (i.e., has children menu items).
11. **isFirst(is_first)**: Returns 1 if a menu item is the first child of its parent or for top-tier menu items, if the item is the first item.
12. **isLast(is_last)**: Returns 1 if a menu item is the last child of its parent or for top-tier menu items, if the item is the last item.
13. **isCurrent(is_current)**: Returns 1 if a menu item matches the current page being viewed in the browser.
14. **numChildren(num_children)**: Shows the number of visible (in the ProcessWire sense) natural (ProcessWire pages) children that the menu item has.
15. **totalChildren(total_children)**: Denotes the total number of child items a menu item has. This includes both natural (see **numChildren**) and any children added via the GUI (backend), both natural and non-natural. By default, this property is not included. It only applies if the option **get_total_children** (see above) is set to 1.
16. **showMoreText(show_more_text)**: This is only applicable if the option **maximum_children_per_parent** (see above) is in effect. It will be applied to the last shown child item of a parent in case that parent has more children than the limit set in **maximum_children_per_parent**.
17. **named_field** (e.g., *description*) specified in **extra_fields**. See notes below.

Extra Fields

This option provides a powerful way to build complex navigation systems, for instance, mega menus, that provide more information to your web users than a simple menu. The option is only supported in **getMenuItems()**. To use it, pass an array to your $options as shown below.

Supported Fieldtypes

The following Fieldtypes can be used in **extra_fields**:

- FieldtypeURL
- FieldtypeCheckbox
- FieldtypeEmail
- FieldtypeInteger
- FieldtypeFloat
- FieldtypePageTitle

- FieldtypeText
- FieldtypeTextarea
- FieldtypePageTitleLanguage
- FieldtypeTextLanguage
- FieldtypeTextareaLanguage
- FieldtypeDatetime
- FieldtypeImage
- FieldtypeFile
- FieldtypeOptions
- FieldtypePage

Using Extra Fields

The option **extra_fields** accepts an associative multi-dimensional array with named ProcessWire fields with options if applicable. Each inner array must have a *name* index whose value corresponds to the name of the ProcessWire field whose value to return for a given menu item.

getMenuItems() will return values of the named fields for each menu where applicable. For multi-lingual sites, the correct values for the current user's language are returned. For simpler fields, the single value is returned. For more complex fields (such as image fields), an array of items with values is returned.

Examples and options for all supported Fieldtypes are shown below.

Datetime Fields

FieldtypeDatetime fields can accept one string option, *format*. If this is not specified, the value of the field as returned by ProcessWire will be returned. This will be in accordance with the field's date output format. If specified, the value returned will be formatted as per the string in *format*, for instance *d-m-Y*.

Example

```
$options = array(
    'extra_fields' =>array(
        // example FieldtypeDatetime field
        array('name'=>'date'),// e.g. $m->date
        // example FieldtypeDatetime field with option
        array('name'=>'date', 'format'=>'d/m/Y'),// e.g. ['date'] or $m->date
    )
);
```

Image and File Fields

FieldtypeImage and FieldtypeFile fields will have their values returned as arrays irrespective of whether the fields are of type single or multiple files/images. The names (of the original image/file) and urls of images/files are always returned irrespective of the options below.

Available options (none need to be declared) for FieldtypeImage fields are:

- **eq**: This will return the image at the specified index. This has precedence over count below. Needs to be an integer. Only applicable to multi-image fields.
- **count**: This will return the first *n* images as per the integer specified in this option. This has lower precedence compared to eq above. Only applicable to multi-image fields. **Please note that if both eq and count are not specified, the first image will be returned**.
- **width**: Images requested will be of the specified width. The value needs to be an integer. If not specified and a height is specified, the image width will be set to auto.
- **height**: Images requested will be of the specified height. The value needs to be an integer. If not specified and a width is specified, the image height will be set to auto. **Please note that if both width and height are not specified,images will be returned at their original size**.
- **description**: Determines if to return image descriptions. If this is not desired, just leave out the option. If descriptions are needed, the value specified needs to be a boolean or a *truthy* integer. In multi-lingual sites, the value for the current user's language is returned.
- **tags**: Determines if to return image tags. If this is not desired, just leave out the option. If tags are needed, the value specified needs to be a boolean or a *truthy* integer.

Example

```
$options = array(
    'extra_fields' =>array(
        // example FieldtypeImage field with no options
        array('name'=>'images'),// e.g. $m->images. (will return array with values for the first image)
        // example FieldtypeImage field with options
        array('name'=>'images','eq'=>2,'height'=>250),// returns the image at index 2 and sized at height 250px, width auto
        // example FieldtypeImage field with options
        array('name'=>'images','count'=>3,'height'=>200,'width'=>200),// returns the first 3 images and sized at height and width 200px
    )
);
```

Available options (none need to be declared) for FieldtypeFile fields are:

- **eq**: This will return the file at the specified index. This has precedence over count below. Needs to be an integer. Only applicable to multi-files fields.
- **count**: This will return the first *n* files as per the integer specified in this option. This has lower precedence compared to eq above. Only applicable to multi-file fields. **Please note that if both eq and count are not specified, the first file will be returned**.
- **description**: Determines if to return file descriptions. If this is not desired, just leave out the option. If descriptions are needed, the value specified needs to be a boolean *true* or a *truthy* integer. In multi-lingual sites, the value for the current user's language is returned.
- **tags**: Determines if to return file tags. If this is not desired, just leave out the option. If tags are needed, the value specified needs to be a boolean or a *truthy* integer.
- **filesize_str**: Determines if to return a file's size in human readable format, e.g. 20 kB. If this is not desired, just leave out the option. If the file size is needed, the value specified needs to be a boolean *true* or a *truthy* integer.

Example

```
$options = array(
    'extra_fields' =>array(
        // example FieldtypeFile field (multi) with no options
        array('name'=>'documents','count'=>5),// e.g. $m->documents. (will return
array with values for the first 5 documents)
        // example FieldtypeFile field (single) with options

array('name'=>'download','filesize_str'=>true,'description'=>1,'tags'=>true),//
returns the file together with the file size, description and tags.
    )
);
```

Page Reference Fields

FieldtypePage will have their values returned as arrays irrespective of whether the fields are of type single or multiple pages. The titles and urls of pages are always returned irrespective of the options below.

Caution: Special attention should be paid to both the number of page items to return as well as the fields returned for those page items. The page items' fields may themselves be *page reference fields*. Failure to carefully consider the count of pages and/or the pages' fields can result in site performance issues (high memory usage, etc).

Available options (none need to be declared) for FieldtypePage fields are:

- **eq**: This will return the page at the specified index. This has precedence over count below. Needs to be an integer. Only applicable to multi-page fields.

- **count**: This will return the first *n* pages as per the integer specified in this option. This has lower precedence compared to eq above. Only applicable to multi-page fields. **Please note that if both eq and count are not specified, the first page will be returned**.
- **fields**: A simple array of names of fields whose values to return for pages. Use with care as per caution above.

Example

```
$options = array(
    'extra_fields' =>array(
        // example FieldtypePage field (multi) without options
        array('name'=>'countries'),// e.g. $m->countries (will return the first page)
            // example FieldtypePage field (multi) with options
        array(
            'name'=>'countries',
            'count' => 3,
            // values of fields inside 'countries' pages to return
            'fields' => array('population','gdp','summary'),
            ),// e.g. $m->countries (will return the first 3 pages values)

            // example FieldtypePage field (single) without options
            array('name'=>'city'),// e.g. ['city'] or $m->city
    )
);
```

Options Fields

FieldtypeOptions will have their values returned as arrays irrespective of whether the fields are of type single or multiple options. The titles and ids of options are always returned irrespective of the options below.

Available options (none need to be declared) for FieldtypeOptions fields are:

- **all_options**: Determines whether to return all selectable options for the field. If this is not desired, just leave out the setting. If all all selectable options are needed, the value specified for *all_options* needs to be a boolean or a *truthy* integer.
- **value**: Determines if to return an option's *value* property as well as the *title*. If this is not desired, just leave out this setting. If the value property is needed, this needs to be specified as a boolean *true* or a *truthy* integer. This setting is only applicable if you have set up your FieldtypeOptions field to separately store *titles* and *values* (see FieldtypeOptions documenation).

Example

```
$options = array(
    'extra_fields' =>array(
        // example FieldtypeOptions field (single), no options
        array('name'=>'option'),// e.g. $m->options (will return the id and the title
of the selected option)
        // example FieldtypeOptions field (multi) with option
        array('name'=>'sizes', 'all_options'=>true, 'value'=>true),// e.g. $m->sizes
(will return all selectable options as well as their value properties. The selected
option will have an index [selected] with value 1)
    )
);
```

Language Fields

FieldtypeTextLanguage, FieldtypeTextareaLanguage and FieldtypePageTitleLanguage fields will have their respective values returned according to the current user's langauge. Only the field names need to be specified. No other options are applicable. The values will be accessible as a property of a menu object (or an index in an array, see **Properties** above) in **getMenuItems()** as $m->nameOfField.

Example

```
$options = array(
    'extra_fields' =>array(
        // example FieldtypeTextLanguage field
        array('name'=>'headline'),// e.g. $m->headline
        // example FieldtypeTextareaLanguage field
        array('name'=>'body'),// e.g. ['body'] or $m->body
    )
);
```

Other Fields

Fields of type FieldtypeURL, FieldtypeCheckbox, FieldtypeEmail, FieldtypeInteger, FieldtypeFloat, FieldtypePageTitle, FieldtypeText and FieldtypeTextarea will have their values returned in their respective native formats.

Example

```
$options = array(
    'extra_fields' =>array(
        // example FieldtypeURL field
        array('name'=>'web'),// e.g. $m->web
        // example FieldtypeCheckbox field
        array('name'=>'selected'),// e.g. ['selected'] or $m->selected
    )
);
```

Example Extra Fields Option Array

Below is an example *countries trivia* navigation using *extra_fields* in its $option. The output of the navigation is shown afterward.

```
$options = array(
    'extra_fields' =>array(
        // datetime
        array('name'=>'visit','format'=>'d M Y'),
        // checkbox
        array('name'=>'safari'),
        // float
        array('name'=>'gdp'),
        // image
        array(
            'name' => 'images',
            'width' => 500,
            'count' => 3,
            'description' =>true,// or a truth value, e.g. 1
            'tags' =>1,// or a truthy value, e.g. true
        ),
        // page field
        array(
            'name'=>'cities',
            'count'=>2,
            'fields'=>array('summary','population','web')),
        // options
        array('name'=>'languages')

    )
);
```

Below is the array output of the above example menu using **extra_fields** option.

```
(
    [1] => Array
        (
            [pages_id] => 1279
            [title] => South Africa
            [url] => /mb-tests/country-facts/country-facts-countries/south-africa/
            [is_first] => 1
            [visit] => 16 Jan 2017
            [safari] => 1
            [gdp] => 6160.73
            [images] => Array
                (
                    [south_africa-1.jpg] => Array
                        (
                            [name] => south_africa-1.jpg
                            [description] => South Africa is a country on the
southernmost tip of the African continent, marked by several distinct ecosystems.
Inland safari destination Kruger National Park is populated by big game. The Western
Cape offers beaches, lush winelands around Stellenbosch and Paarl, craggy cliffs at
the Cape of Good Hope, forest and lagoons along the Garden Route, and the city of Cape
Town, beneath flat-topped Table Mountain.
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1279/south_africa-
1.500x0.jpg
                        )

                    [south_africa-2.jpg] => Array
                        (
                            [name] => south_africa-2.jpg
                            [description] =>
                            [tags] => Safari Delightful
                            [url] => /site-menubuilder/assets/files/1279/south_africa-
2.500x0.jpg
                        )

                    [south_africa-3.jpg] => Array
                        (
                            [name] => south_africa-3.jpg
                            [description] =>
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1279/south_africa-
3.500x0.jpg
                        )

                )

            [cities] => Array
                (
                    [port-elizabeth] => Array
                        (
```

```
                            [title] => Port Elizabeth
                            [url] => /mb-tests/country-facts/country-facts-
cities/port-elizabeth/
                            [population] => 312392
                            [web] => https://www.nmbt.co.za/
                        )

                    [soweto] => Array
                        (
                            [title] => Soweto
                            [url] => /mb-tests/country-facts/country-facts-
cities/soweto/
                            [population] => 1271628
                            [web] => http://www.soweto.gov.za/
                        )

                )

            [languages] => Array
                (
                    [1] => Array
                        (
                            [id] => 1
                            [title] => English
                        )

                    [2] => Array
                        (
                            [id] => 2
                            [title] => Tswana
                        )

                    [3] => Array
                        (
                            [id] => 3
                            [title] => Afrikaans
                        )

                    [4] => Array
                        (
                            [id] => 4
                            [title] => Zulu
                        )

                    [5] => Array
                        (
                            [id] => 5
                            [title] => Xhosa
                        )

                    [6] => Array
```

```
                    (
                        [id] => 6
                        [title] => Northern Sotho
                    )

                [7] => Array
                    (
                        [id] => 7
                        [title] => Venda
                    )

                [8] => Array
                    (
                        [id] => 8
                        [title] => Southern Sotho
                    )

                [9] => Array
                    (
                        [id] => 9
                        [title] => Tsonga
                    )

                [10] => Array
                    (
                        [id] => 10
                        [title] => Swati
                    )

                [11] => Array
                    (
                        [id] => 11
                        [title] => Ndebele
                    )

            )

    )

[2] => Array
    (
        [pages_id] => 1262
        [title] => China
        [url] => /mb-tests/country-facts/country-facts-countries/china/
        [visit] => 23 Sep 2021
        [safari] => 1
        [gdp] => 8826.99
        [images] => Array
            (
                [china-1.jpg] => Array
                    (
```

```
                            [name] => china-1.jpg
                            [description] => China, officially the People's Republic
of China, is a country in East Asia and is the world's most populous country, with a
population of around 1.428 billion in 2017. Covering approximately 9,600,000 square
kilometers, it is the third-largest or the fourth-largest country by area.
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1262/china-
1.500x0.jpg
                        )

                    [china-2.jpg] => Array
                        (
                            [name] => china-2.jpg
                            [description] =>
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1262/china-
2.500x0.jpg
                        )

                    [china-3.jpg] => Array
                        (
                            [name] => china-3.jpg
                            [description] =>
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1262/china-
3.500x0.jpg
                        )

                )

            [cities] => Array
                (
                    [wuhan] => Array
                        (
                            [title] => Wuhan
                            [url] => /mb-tests/country-facts/country-facts-
cities/wuhan/
                            [population] => 11081000
                            [web] => http://www.wuhan.gov.cn/
                        )

                    [suzhou] => Array
                        (
                            [title] => Suzhou
                            [url] => /mb-tests/country-facts/country-facts-
cities/suzhou/
                            [population] => 10721700
                            [web] => http://www.suzhou.gov.cn/
                        )

                )
```

```
            [languages] => Array
                (
                    [12] => Array
                        (
                            [id] => 12
                            [title] => Mandarin
                        )

                )

        )

    [3] => Array
        (
            [pages_id] => 1260
            [title] => Canada
            [url] => /mb-tests/country-facts/country-facts-countries/canada/
            [visit] => 11 Jan 2020
            [gdp] => 45032.1
            [images] => Array
                (
                    [canada-1.jpg] => Array
                        (
                            [name] => canada-1.jpg
                            [description] => The beauty that is Canada
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1260/canada-
1.500x0.jpg
                        )

                    [canada-2.jpg] => Array
                        (
                            [name] => canada-2.jpg
                            [description] =>
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1260/canada-
2.500x0.jpg
                        )

                    [canada-3.jpg] => Array
                        (
                            [name] => canada-3.jpg
                            [description] =>
                            [tags] => Travel Wow
                            [url] => /site-menubuilder/assets/files/1260/canada-
3.500x0.jpg
                        )

                )
```

```
            [cities] => Array
                (
                    [quebec-city] => Array
                        (
                            [title] => Quebec City
                            [url] => /mb-tests/country-facts/country-facts-
cities/quebec-city/
                            [population] => 531902
                            [web] => https://www.ville.quebec.qc.ca/en/
                        )

                    [vancouver] => Array
                        (
                            [title] => Vancouver
                            [url] => /mb-tests/country-facts/country-facts-
cities/vancouver/
                            [population] => 631486
                            [web] => https://vancouver.ca/
                        )

                )

            [languages] => Array
                (
                    [1] => Array
                        (
                            [id] => 1
                            [title] => English
                        )

                    [13] => Array
                        (
                            [id] => 13
                            [title] => French
                        )

                )

        )

    [4] => Array
        (
            [pages_id] => 1263
            [title] => Switzerland
            [url] => /mb-tests/country-facts/country-facts-countries/switzerland/
            [gdp] => 80189.7
            [images] => Array
                (
                    [switzerland-2.jpg] => Array
                        (
                            [name] => switzerland-2.jpg
```

```
                            [description] =>
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1263/switzerland-
2.500x0.jpg
                        )

                    [switzerland-3.jpg] => Array
                        (
                            [name] => switzerland-3.jpg
                            [description] =>
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1263/switzerland-
3.500x0.jpg
                        )

                    [switzerland-1.jpg] => Array
                        (
                            [name] => switzerland-1.jpg
                            [description] => Switzerland is a mountainous Central
European country, home to numerous lakes, villages and the high peaks of the Alps. Its
cities contain medieval quarters, with landmarks like capital Bern’s Zytglogge clock
tower and Lucerne’s wooden chapel bridge. The country is also known for its ski
resorts and hiking trails. Banking and finance are key industries, and Swiss watches
and chocolate are world renowned.
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1263/switzerland-
1.500x0.jpg
                        )

                )

            [cities] => Array
                (
                    [chur] => Array
                        (
                            [title] => Chur
                            [url] => /mb-tests/country-facts/country-facts-
cities/chur/
                            [population] => 33373
                            [web] => http://www.chur.ch/
                        )

                    [st-gallen] => Array
                        (
                            [title] => St. Gallen
                            [url] => /mb-tests/country-facts/country-facts-cities/st-
gallen/
                            [population] => 75806
                            [web] => http://www.stadt.sg.ch/
                        )
```

```
                )

            [languages] => Array
                (
                    [13] => Array
                        (
                            [id] => 13
                            [title] => French
                        )

                    [14] => Array
                        (
                            [id] => 14
                            [title] => German
                        )

                    [15] => Array
                        (
                            [id] => 15
                            [title] => Italian
                        )

                    [16] => Array
                        (
                            [id] => 16
                            [title] => Romansh
                        )

                )

        )

    [5] => Array
        (
            [pages_id] => 1261
            [title] => USA
            [url] => /mb-tests/country-facts/country-facts-countries/usa/
            [is_last] => 1
            [visit] => 08 Aug 2020
            [gdp] => 59531.7
            [images] => Array
                (
                    [usa-1.jpg] => Array
                        (
                            [name] => usa-1.jpg
                            [description] => The U.S. is a country of 50 states
covering a vast swath of North America, with Alaska in the northwest and Hawaii
extending the nation’s presence into the Pacific Ocean.
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1261/usa-
1.500x0.jpg
```

```
                        )

                    [usa-2.jpg] => Array
                        (
                            [name] => usa-2.jpg
                            [description] =>
                            [tags] =>
                            [url] => /site-menubuilder/assets/files/1261/usa-
2.500x0.jpg
                        )

                    [usa-3.jpeg] => Array
                        (
                            [name] => usa-3.jpeg
                            [description] =>
                            [tags] => Mountains
                            [url] => /site-menubuilder/assets/files/1261/usa-
3.500x0.jpeg
                        )

                )

            [cities] => Array
                (
                    [seattle] => Array
                        (
                            [title] => Seattle
                            [url] => /mb-tests/country-facts/country-facts-
cities/seattle/
                            [population] => 608660
                            [web] => http://www.seattle.gov/
                        )

                    [chicago] => Array
                        (
                            [title] => Chicago
                            [url] => /mb-tests/country-facts/country-facts-
cities/chicago/
                            [population] => 2695598
                            [web] => http://chicago.gov/
                        )

                )

            [languages] => Array
                (
                    [1] => Array
                        (
                            [id] => 1
                            [title] => English
                        )
```

```
                )
        )
)
```

How to Use Menu Builder

- Access Menu Builder in your ProcessWire admin and create a menu.
- Edit the menu and add items to it, dragging and dropping them in different positions as you wish.
- Once you've created a menu, you can view it in the frontend by loading it using MarkupMenuBuilder in a template file as follows.

Rendering a Menu

The simplest way to render a menu is to use the method **render()**. It is straightforward and can be passed various options to customise your menu.

However, for the ultimate flexibility and total freedom, especially for complex menu structures, we recommend that you use the method **getMenuItems()**. Please note that the method **getMenuItems()** will not return a menu out of the box. Instead, it returns menu items that you can optionally manipulate, traverse using any recursive function (or for simpler menus, nested foreachs) and output within a HTML structure of your choosing. It means that you can apply logic within your chosen recursive function (or loop) to output extra details with your menu, for instance grab some data from a field within the pages that your menu items represent. Another example would be to show some parts of the menu only when a user is logged in, etc. If working with the Menu object, it means you can easily add properties to some or all of the menu items at runtime. It also means you have all the powerful WireArray methods at your disposal (don't touch sort though!).

```
// load the module
$menu = $modules->get('MarkupMenuBuilder');// $menu is an example

/**you can render by menu Page object, name, title, id or properly formatted array of
menu items**/

// render by name, title or id
echo $menu->render('Title of Your Menu');// render the menu by title
echo $menu->render('name-of-your-menu');// render the menu by name
echo $menu->render('1234');//render by ID

// render by passing a Page object
$m = $pages->get(1234);
echo $menu->render($m);//render by Page object

// render by passing an array
// get the Menu Builder field menu_items for this menu. That is where your menu items
JSON string is stored
$json = $pages->get(1234)->menu_items;
// convert the JSON string to an array. Here we assume the JSON string is not empty
$array = json_decode($json, true);
echo $menu->render($array);//render by array
```

You can render additional menus as well, e.g.

```
echo $menu->render('sidenav');
```

You can additionally pass CSS class/id options to the method. See above for available options.

```
$options = array(
    'has_children_class' => 'has_children',
    'current_class' => 'current',
    'menu_css_id' => 'main',
    'menu_css_class' => 'nav',
);

echo $menu->render('sidenav', $options);
```

Examples of more complex menus utilising the method **getMenuItems()** using various recursive menu functions can be seen at these gists.

Rendering Breadcrumbs

Rendering breadcrumbs is quite similar to the above, the only difference being the method you use and some of the options that can be used to configure the output.

```
// load the module
$menu = $modules->get('MarkupMenuBuilder');// $menu is an example

/**you can render by menu Page object, name, title, id or properly formatted array of
menu items**/

// render by name, title or id
echo $menu->renderBreadcrumbs('Title of Your Menu');// render the menu by title
echo $menu->renderBreadcrumbs('name-of-your-menu');// render the menu by name
echo $menu->renderBreadcrumbs('1234');// render by ID

// render by passing a Page object
$m = $pages->get(1234);
echo $menu->renderBreadcrumbs($m);

// render by passing an array
// get the Menu Builder field menu_items for this menu. That is where your menu items
JSON string is stored
$json = $pages->get(1234)->menu_items;
// convert the JSON string to an array. Here we assume the JSON string is not empty
$array = json_decode($json, true);
echo $menu->renderBreadcrumbs($array);// render by array
```

Additionally, you can pass some options to the method. See above for available options.

```
$options = array(
    'wrapper_list_type' => 'div',
    'list_type' => 'span',
    //'list_type' => '',// if empty, no tag will be applied + no CSS ID
    'menu_css_id' => 'crumbs',
    'menu_css_class' => 'trail',
    'current_css_id' => 'current',
    'divider' => '&ast;',
    'prepend_home' => 1
);

echo $menu->renderBreadcrumbs(1234, $options);
```

Using the Include Children Feature

This is a flexible and powerful feature that you can leverage to produce all sorts of dynamic navigation for your ProcessWire website. With lots of power comes responsibility. The features should only be enabled (see relevant permission below) for users who know what they are doing. For example, a user could unknowingly use the feature to include descendant pages of a menu item that has hundreds of descendants, leading to undesirable effects.

Menu-Level Values

At the global/menu/API-level, the following values can be passed with 'include_children'. This is useful if you want to override some item-level settings. The priority order of the settings and in comparison to item-level settings are explained further below.

- **0**: Overrides all settings, global and local and suppresses output of 'include_children'.
- **1**: Include children of all navigation items but only in the menu unless specified otherwise in item-level setting.
- **2**: Include children of all navigation items but only in the breadcrumbs unless specified otherwise in item-level setting.
- **3**: Include children of all navigation items in both the menu and breadcrumbs unless specified otherwise in item-level setting. This assumes you are passing the same options to both **render()** and **renderBreadcrumbs()**.
- **4**: Do not include children of any navigation item unless specified otherwise in item-level setting. This is the default setting (automatically applied) and does not need to be specified in your $options.

Item-Level Values

At the local/admin/item-level, the following values can be saved with each navigation item either when creating or editing the menu item in the Menu Builder admin. This is useful when you want fine-grained control over 'include_children' output. The priority order of the settings and in comparison to menu-level settings are explained later below.

- **No**: This is the default value; It means do not include children of this menu item unless overridden by a menu-level setting.
- **Menu**: Include children of this navigation item but only in the menu unless overridden by a **0** menu-level setting.
- **Breadcrumbs**: Include children of this navigation item but only in the breadcrumbs unless overridden by a **0** menu-level setting.
- **Both**: Include children of this navigation item in both the menu and breadcrumbs overridden only by a **0** menu-level setting.
- **Never**: Never include children of this navigation item.

The combined (menu- and item-level) order of descending priority (i.e. a setting cannot override the setting above it) for these 'include_children' settings is as follows:

- **0** / **Never**
- **Menu** / **Breadcrumbs** / **Both**
- **1** / **2** / **3**

- **4 / No**

As a sanity check, in the Menu Builder admin, a navigation item made up of your ProcessWire 'homepage' cannot have its children included/excluded since it is the parent of all pages. In addition, you will only see 'include_children' inputs in the Menu Builder admin when both the feature is enabled and you have the permission **menu-builder-include-children** (or are a superuser; more below under Permissions). Finally, note that if a user without this permission edits a menu that was previously edited by a user with the permission, any 'include_children' item-level settings will be lost.

Other Usage Notes

Note that you are not limited to using MarkupMenuBuilder. You can also use your own PHP (or even JavaScript) recursive function to display your menu by decoding the saved JSON string containing menu items and passing the resulting array to your function for traversal. The CSS is up to you, of course. Here's a nice example from a forum member. However, It is preferable to use **getMenuItems()** as shown above.

By default, if using AsmSelect or PageAutocomplete to select pages to add to your menu, the module will return a maximum of 50 pages. In that case, you might run into the 'not all my selectable pages are displayed issue'. You have two choices to resolve this. One, use PageListSelectMultiple instead. Alternatively, add a custom selector in the text field 'Pages selectable in menu' (main tab). For instance, *limit=50*. Or *limit=50, sort=title* or any other valid ProcessWire selector. You will either need to be logged in as a superuser or have the permission **menu-builder-selectable** in order to do this.

Permissions

You can use the following permissions to control visibility and access to various advanced settings of Menu Builder by non-superusers. In ProcessWire, by default, Superusers inherit all permissions. **Note that you will have to create and apply the permissions yourself using the normal ProcessWire way**, i.e.

- Create a role, e.g. **menu-editor**.
- Create **permissions** and add assign them (when editing your role) to the role you created.
- Create a **user** and assign them the role with the Menu Builder permissions.

There are 9 permissions at your disposal for fine-grained access control of your Menu Builder admin.

1. **menu-builder-lock** This permission allows your non-superusers to lock and unlock menus using the 'Actions Panel' in Menu Builder's admin main page. **Note two things**: (i) This

permission takes a whitelist approach. If this permission does not exist on your system (i.e. has not yet been created or is present but unpublished), by default, all users will be able to lock/unlock menus. (ii) The permission only kicks in if it is found. In that case, users without the permission will not be able to lock/unlock menus.

2. **menu-builder-delete** This permission allows your non-superusers to trash and delete menus using either the 'Actions Panel' in Menu Builder's admin main page or, when editing a single menu, via the 'Delete' tab. **Note two things**: (i) This permission takes a whitelist approach. If this permission does not exist on your system (i.e. has not yet been created or is present but unpublished), by default, all users will be able to trash and delete menus. (ii) The permission only kicks in if it is found. In that case, users without the permission will not be able to trash or delete menus.

3. **menu-builder-selector** This permission allows your non-superusers to add ProcessWire pages as menu items using a ProcessWire selector. This permission (**and all the rest below**) takes a blacklist approach. The permission does not need to exist in your system to kick in and by default all non-superusers are not able to add pages as menu items using selectors. This means that if you intend that only superusers will be allowed to use selectors in this manner, *there is no need to create the permission*. Only create and apply it if you wish to grant a particular non-superuser this feature. **The same applies to all the following permissions**.

4. **menu-builder-selectable** This permission allows non-superusers to specify ProcessWire pages that are selectable as menu items in either of two configurable page fields in Menu Builder, i.e. AsmSelect or PageAutocomplete (BUT not PageListSelectMultiple).

5. **menu-builder-markup** This permission allows your non-superusers to allow/disallow the use of markup/HTML in menu item titles/labels.

6. **menu-builder-settings** This permission allows non-superusers to edit nestedSortable settings, for instance set maxLevels (that controls number of nesting levels in a menu).

7. **menu-builder-page-field** This permission allows non-superusers to change the page field type used to select ProcessWire pages to add as menu items, i.e. toggle between AsmSelect [default], PageAutocomplete or PageListSelectMultiple.

8. **menu-builder-include-children** This permission allows non-superusers to set and use the include children feature.

9. **menu-builder-disable-items** This permission allows non-superusers to set and use the disable menu items feature.

10. **menu-builder-multi-lingual-items** This permission allows non-superusers to set and use the multi-lingual menu items feature (available only on multi-lingual sites).

Uninstall

Uninstall like any other ProcessWire module. Note that **All your menus will be deleted on uninstall!**. The associated fields and template above will also be deleted.

Resources

- Support Forum

License

GPL2

Changelog

Version 0.2.7

1. In multi-lingual environments, menus and breadcrumbs can retrieved using their titles or names in any language irrespective of the current user's language.
2. For getMenuItems() usage only, added option extra_fields to return values of some specific fields on the menu item pages. See documentation for compatible Fieldtypes.
3. Changed menu builder GUI and process for adding Menus. Menus are now added one at a time and can have multi-lingual titles.
4. Locked menu GUI improved.
5. Fixed bug that allowed access to unpublished menus for frontend rendering.

Version 0.2.6

1. Added the properties numChildren, totalChildren and showMoreText for use with getMenuItems().
2. Added option maximum_children_per_parent to limit the maximum number of (included) children a menu item can return.
3. Fixed CSS issue that affected menu items' trash cans in the backend. Thanks @duncan.
4. Refactored code to improve efficiency.

Version 0.2.5

1. Fixed typos and minor bugs where we needed to check if a variable was an array first before counting it.

Version 0.2.4

1. Fixed a bug where default_class was not getting applied to included children in getMenuItems() context.
2. Fixed a bug where last_class was not getting applied correctly/at all to some menu items.

Version 0.2.3

1. Fixed a bug where inactive pages in multi-language sites would still be displayed in menus.

Version 0.2.2

1. Support for namespaced ProcessWire only (ProcessWire 3.x).
2. Various bug fixes in getMenuItems() and breadcrumbs.
3. Fixed bug that prevented menus in multi lingual environments from saving.

Version 0.2.1

1. Fixed bug in 'current_class_level' where 'current_class' was being applied to 'Home' menu item when level was set to '0'.
2. getMenuItems() now honours both locally and globally set 'include_children'.
3. Added 'cached_menu' option to MarkupMenuBuilder to save and fetch menus from cache to speed up menu rendering if required.
4. Added 'cached_menu_time' to set cached menu expiration time.
5. Fixed bug in ProcessMenuBuilder where 'include_children' and 'm_max_level' inputs would be displayed when the page selected to add to menu was 'Home'.

Version 0.2.0

1. Modified 'current_class_level' option to apply 'current_class' to a menu item when viewing any of its descendant child pages (child, grandchild, etc.) in cases where those descendant pages are neither part of the menu nor included via 'include_children' if its value is set to '0'.

Version 0.1.9

1. All (custom and pages) menu items now optionally multi-lingual.

2. Added setting to use permission 'menu-builder-multi-lingual-items'.
3. Code refactoring.

Version 0.1.8

1. Added enable/disable menu items settings.
2. Added setting to use permission 'menu-builder-disable-items'.
3. Added 'check_listable' option not to display menu items not listable to current user.
4. 'current_class' now also applies to breadcrumbs.
5. Code cleanup.

Version 0.1.7

1. Fixed minor bug that affected display of titles of menu items with apostrophes in menu settings.
2. Code cleanup.

Version 0.1.6

1. Fixed JS Bug where select all checkbox was not working in Menu Builder dashboard.
2. Fixed issue where InputfieldMarkup would not properly render HTML in descriptions.

Version 0.1.5

1. Added method getMenuItems that greatly simplifies the creation of complex custom menus.

Version 0.1.4

1. Moved first tab 'Main' to become the third tab and renamed it to 'Settings' for better UX.

Version 0.1.3

1. Fixed bug where on uninstalling the module, menu pages in the trash would not get deleted, throwing an error when attempting to delete their template.

Version 0.1.2

1. As per a request, added ability to use 'Page List Select Multiple' page field to select pages to add to menu items. This is in addition to the existing AsmSelect and PageAutocomplete.

2. Fixed bug where 'new_tab' setting would not be reliably applied to new custom links menu items.

Version 0.1.1

1. Fixed bug relating to getLanguageValue() (issues #22 and #25).
2. Fixed bug where 'has_children' would not be applied to a menu item whose children were included 'natively'.
3. Switched to $this->wire('pages') from wire('pages') (and similar).
4. As per request at issue #18, added option 'default_class' to enable application of a default CSS class to every menu item at API level.

Version 0.1.0

1. Cleaned-up HTML output by MarkupMenuBuilder.

Version 0.0.9

1. Fixed bug in 'current_class_level' when 'include_children' is active.

Version 0.0.8

1. Added new feature 'include children'. Allows global (menu-level) or item-level inclusion of a ProcessWire page menu item's 'natural' descendants (children, grandchildren, etc) in menus/breadcrumbs at runtime. This means such pages do not have to be added as part of the navigation during editing in Menu Builder.
2. Added a 'current_class_level' to specify how far up menus the 'current_class' should be applied (i.e. current/active parents).
3. Permission 'menu-builder-included-children' added to limit access to the new feature 'include children'.

Version 0.0.7

1. Added multi-lingual support for saving and displaying menus/breadcrumbs.

Version 0.0.6

1. Added method in MarkupMenuBuilder for rendering breadcrumbs - renderBreadcrumbs().
2. Can now also pass Page object to method render() to render menu items/breadcrumbs

Version 0.0.5

1. Corrected some markup errors.

Version 0.0.4

1. Added various permissions to control visibility and editing of advanced settings

Version 0.0.3

1. Option to allow markup in menu item title (label) - superusers only
2. Added more options to render() method, e.g. first, last classes

Version 0.0.2

1. First Beta version
2. Menus saved as ProcessWire pages
3. Menu items saved as JSON in a field in Menu pages
4. Add menu items from ProcessWire pages using page fields (option to choose between PageAutocomplete and AsmSelect [default]) or a Selector (e.g. template=basic-page, limit-20).
5. Specify a selector to return only those specified pages for selection in the page field (i.e. asm and autocomplete)
6. For page fields, similar to custom menu items, add CSS classes and IDs as you add the items
7. Menu settings for nestedSortable - e.g. maxLevels (limit nesting levels)
8. Lock down menus for editing
9. Highly configurable MarkupMenuBuilder

Version 0.0.1

Initial alpha release