aw2.register

This shortcode is used to register custom post type, taxonomy, sidebar, menu positions, meta boxes, rewrite rules, users, app settings and site options. aw2.register needs to be used within triggers with correct "*when*" taxonomy selection. It accepts a JSON object of args array that are needed by WordPress.

attributes:

- **main:**
  post | taxonomy | sidebar | menu_location | metabox | site_option | custom_metabox | app_option | rewrite_rule|less_variables |query_vars |user

  You have to specify the type, to register respective options

- **slug:**
  It is required for registering custom post type and taxonomy. Specified slug is used to register the CPT and taxonomy slug.
- **post_type:**
  You can specify the comma-separated list of post types, to which you want to attach your custom taxonomy. It is only used by taxonomy type.
- **id:**
  you need to specify *id* when you want to register metabox and site_options, this id is used to match the fields
- **part:**
  main | field

  When registering meta box or site options, you need to specify the part. part equal to 'main' is used for defining the properties of options panel or meta box. Part 'field' is used to add various fields to the part 'main'.

Check following links for getting various parameters that are available for use

1. Custom Post Type
2. Custom Taxonomy
3. Sidebars
4. Menu Locations
5. Metaboxes
6. Site Options

Examples:

Creating custom post type with Awesome UI 2 using *aw2.register*, select *init* in *When* taxonomy.

```
[aw2.set module.singular_label='Form Entry' /]
[aw2.set module.plural_label='Form Entries' /]
[aw2.register post slug='as_entries']
{   "label" : "[aw2.get module.plural_label /]",
    "public" : true,
    "exclude_from_search":true,
    "publicly_queryable":true,
    "show_in_nav_menus":false,
    "show_ui" : true,
    "show_in_menu" : true,
    "capability_type" : "post",
    "map_meta_cap" : true,
    "rewrite" : [ "slug":"entry_type", "with_front":false  ],
    "hierarchical" : false,
    "query_var" : true,
    "supports" : [ "title","editor","page-attributes","custom-fields" ],
    "labels" : {
      "name" : "[aw2.get module.plural_label /]",
      "singular_name" : "[aw2.get module.singular_label /]",
      "menu_name" : "[aw2.get module.plural_label /]",
      "add_new" : "Add [aw2.get module.singular_label /]",
      "add_new_item" : "Add New [aw2.get module.singular_label /]",
      "edit" : "Edit",
      "edit_item" : "Edit [aw2.get module.singular_label /]",
      "new_item" : "New [aw2.get module.singular_label /]",
      "view" : "View [aw2.get module.singular_label /]",
      "view_item" : "View [aw2.get module.singular_label /]",
      "search_items" : "Search [aw2.get module.plural_label /]",
      "not_found" : "No [aw2.get module.plural_label /] Found",
      "not_found_in_trash" : "No [aw2.get module.plural_label /] Found in Trash",
      "parent" : "Parent [aw2.get module.singular_label /]"
      }
}
[/aw2.register]
```

Registering Taxonomy is awesome studio framework.

```
[aw2.set module.type_singular_label='Entry Type' /]
[aw2.set module.type_plural_label='Entry Types' /]
[aw2.register taxonomy slug='entry_type' post_type='as_entries']
{
    "label" :"[aw2_param type_plural_label /]",
    "hierarchical" : true,
    "show_ui":true,
    "show_admin_column":false,
    "show_in_nav_menus":false,
    "query_var" : true,
    "rewrite" : [ "slug":"entry_type", "with_front":false  ],
    "labels" : {
        "name" : "[aw2.get module.type_plural_label /]",
        "singular_name" : "[aw2.get module.type_singular_label /]",
        "menu_name" : "[aw2.get module.type_plural_label /]",
        "add_new" : "Add [aw2.get module.type_singular_label /]",
        "add_new_item" : "Add New [aw2.get module.type_singular_label /]",
        "edit" : "Edit",
        "edit_item" : "Edit [aw2.get module.type_singular_label /]",
        "new_item" : "New [aw2.get module.type_singular_label /]",
        "view" : "View [aw2.get module.type_singular_label /]",
        "view_item" : "View [aw2.get module.type_singular_label /]",
        "search_items" : "Search [aw2.get module.type_plural_label /]",
        "not_found" : "No [aw2.get module.type_plural_label /] Found"
    }
}
[/aw2.register]
```

Example of creating a sidebar

```
[aw2.register sidebar]
    {
      "name" : "Sidebar name",
      "id"   : "unique-sidebar-id",
        "description"  :" Descriptions",
        "class"      :"",
 "before_widget" : "<li id='%1$s' class='widget %2$s'>",
```

```
 "after_widget"  : "</li>",
 "before_title"  : "<h2 class='widgettitle'>",
 "after_title"   : "</h2>"
    }
[/aw2.register]
```

creating custom menu location for site

```
[aw2.register menu_locations]
{
  "footer_menu_one":"Footer Menu First Column",
  "footer_menu_two":"Footer Menu Second Column",
  "footer_menu_three":"Footer Menu Third Column"
  "footer_menu_four":"Footer Menu Forth Column"
}
[/aw2.register]
```

Example of creating metaboxes in WordPress, please use *metabox* term in the *when* taxonomy.

```
[aw2.register metabox part=start]
{
 "id" :"author_module_data",
 "title" :"Author Data",
 "object_types" : [ "product" ],
 "context" :"normal",
 "priority" :"high"
}
[/aw2.register]

[aw2.register metabox part=field]
{
  "id"      :"sms-api-key",
 "type"    :"text",
 "name"    :"API Key",
 "desc"    :"Key to connect to the SMS Integration Servers."
}
[/aw2.register]

[aw2.register metabox part=field]
{
  "id"      :"sms-sender-id",
 "type"    :"text",
 "name"    :"Sender ID",
 "desc"    :"The 6 characters senderid which has been setup on your KooKoo Account."
}
[/aw2.register]
```

Custom metabox using a module, *module* is slug that will be run to show the fields on metabox section

```
[aw2.register custom_metabox]
{
"id" :"author_module_data",
"title" :"Author Data",
"object_types" : [ "product" ],
"context" :"normal",
"priority" :"high",
"module" : "author-data"
}
[/aw2.register]
```

Creating site options/app options panel using we use CMB2 fields library, you can use that to check definitions

```
[aw2.register site_option part=start]
 {
     "title"           :"Footer contact Settings",
      "id"              :"footer-contact-settings",
       "show_names"      :true,
 "show_on"          :{ "key": "options-page", "value": [ "footer-contact-settings" ] }
 }
[/aw2.register]

[aw2.register site_option part=field]
 {
 "id"      :"opt-address",
 "type"    :"textarea",
 "name"    :"Address Text",
 "subtitle" :"HTML Allowed (wp_kses)",
 "desc"    :"This is the Address field"

 }
[/aw2.register]

[aw2.register site_option part=field]
 {
 "id"      :"opt-landline-one",
 "type"    :"text",
 "name"    :"Landline Number",
 "desc"     :"Add Your Contact Details"

 }
[/aw2.register]
```

Example of registering less variables, just use init hook to trigger it, afterwards, these variables will be available on all the less blocks.

```
[aw2.register less_variables]
  @brand-primary:   #ff5722;
  @brand-success:   #8ec63f;
  @brand-info:      #01aaad;
  @brand-warning:   #f09a09;
  @brand-danger:    #d9534f;
  @brand-secondary: #7cb342;
  @brand-third:    #601848;
  @brand-fourth:  #fabe28;
  @brand-fifth:    #03a9f4;
  @brand-sixth:    #00ffff;
[/aw2.register]
```