What is the schema?

The schema defines all the components that are available to the theme:

Component	Description
Templates	The major page types (e.g. homepage, product page).
Sections	Reusable layout components that can contain blocks.
Blocks	Reusable items available for use inside sections.
Settings	Configurable options for the entire theme, specific sections or specific blocks.
Presets	Configurable presets for options to avoid repetition.

Location

The schema should be created at the following location in your local storage disk:

```
"themes/{$theme}/blueprint/schema.json";
```

When you have made unsaved changes to the schema a copy will be temporarily made at:

```
"blueprint/{$theme}/temp/versions/{$version}/schema.json"
```

And when you save any changes a new schema will be stored at:

```
"blueprint/{$theme}/versions/{$version}/schema.json"
```

Where `$theme` is your store's theme and `$version` is the current active version (defaults to `main`).

Templates

Templates define the major page types that your theme supports.

Types

These are the supported template types as of `10/27/2025`:

- `homepage`
- `listings`
- `products`
- `pages`
- `cart`
- `blogs` (if `aerocargo/blog` `1.x` or `2.x` is installed)
- `blog-listings` (if `aerocargo/blog` `2.x` is installed)
- `error-404`

Structure

Key	Description	Required?
`id`	A unique identifier for the template.	Yes
`type`	The type of template (e.g. `homepage`, `products`, `pages`).	Yes
`variants`	An array of variations for this template.	No
`variants.*.id`	A unique identifier for the template variant.	Yes
`variants.*.name`	The name of the template variants.	Yes

Example

```
{
    "templates": {
        "homepage": {
            "id": "homepage",
            "type": "homepage"
        },
        "products": {
            "id": "products",
            "type": "products",
            "variants": [
```

```
                {
                    "id": "default",
                    "name": "Default Product"
                }
            ]
        }
    }
}
```

Sections

Sections are reusable building blocks that can be added to templates.

Structure

Key	Description	Required?
`id`	A unique alpha-numeric identifier for the section.	Yes
`preview`	Filename of an image located in public storage at `/images/blueprint/previews` (used in the section picker).	No
`icon`	The key of an icon to use (overrides the predicted icon determined by what settings the section contains).	No
`name`	The display name of the section.	Yes
`max_blocks`	The maximum number of blocks allowed in this section.	No
`settings`	An array of Setting objects.	No
`blocks`	An array of Block objects.	No
`scopes`	An array of Scope objects.	No

`variations`	An array of Variation objects.	No

Example

```
{
    "id": "announcement-bar",
    "preview": "announcement-bar.png",
    "name": "Announcement Bar",
    "max_blocks": 4
}
```

Blocks

Blocks represent repeatable content items within a section.

Structure

Key	Description	Required?
`type`	The type of block (e.g. `message`, `image`, `button`)	Yes
`display`	The title of the block (overrides the auto-generated one determined from the block's settings)	No
`name`	The name of the block (displayed in the block picker)	Yes
`settings`	An array of Setting objects	Yes

Example

```
{
  "type": "message",
  "name": "Message",
  "settings": []
}
```

Settings

Settings are configurable options for the entire theme, specific sections or specific blocks

Types

These are the supported setting types as of `10/27/2025`:

- `checkbox`
- `color`
- `css`
- `eloquent`
- `html`
- `image`
- `link`
- `number`
- `pill_select`
- `radio`
- `range`
- `richtext`
- `select`
- `text`
- `video`

Structure

Base Setting

Key	Description	Required?
`id`	A unique identifier for the setting	Yes
`type`	The setting type (e.g. `color`, `select`, `checkbox`, `eloquent`)	Yes
`label`	The name of the setting shown in the editor	Yes
`default`	The default value for this setting	No

`group`	A group name (e.g. `custom_styling`) to organize related settings	No
`info`	Additional helper text or hint	No
`rules`	A rule string that determines when the setting should show/hide	No

Eloquent Setting

Key	Description	Required?
`model`	The model that the eloquent setting is for, e.g. `product`	Yes

Number Setting

Key	Description	Required?
`unit`	The unit of the number, e.g. `mph`	No
`min`	The minimum value the number setting can be	No
`max`	The maximum value the number setting can be	No
`step`	The step size used	No

Radio Setting

Key	Description	Required?
`preset`	An id of an options preset to use	`*`
`options`	An array of options	`*`
`options.*.value`	The value of the option	Yes
`options.*.label`	The label for the option	Yes

`*` One of `preset` and `options` is required, if both are present `preset` will take priority.

Range Setting

Key	Description	Required?
`unit`	The unit of the number, e.g. `mph`	No
`min`	The minimum value the number setting can be	No
`max`	The maximum value the number setting can be	No
`step`	The step size used	No

Select Setting

Key	Description	Required?
`preset`	An id of an options preset to use	`*`
`options`	An array of options	`*`
`options.*.value`	The value of the option	Yes
`options.*.label`	The label for the option	Yes

`*` One of `preset` and `options` is required, if both are present `preset` will take priority.

Pill Select Setting

Key	Description	Required?
`preset`	An id of an options preset to use	`*`
`options`	An array of options	`*`
`options.*.value`	The value of the option	Yes
`options.*.label`	The label for the option	Yes

`*` One of `preset` and `options` is required, if both are present `preset` will take priority.

Examples

```
[
    {
        "id": "background_color",
        "type": "color",
        "label": "Background colour",
        "default": "#ffffff"
```

```
    },
    {
        "id": "border",
        "type": "checkbox",
        "label": "Show separator line",
        "default": true
    },
    {
        "id": "padding_bottom",
        "group": "custom_styling",
        "label": "Padding bottom",
        "type": "select",
        "default": "xxs",
        "options": [
            {
                "label": "None",
                "value": "0"
            },
            {
                "label": "Small",
                "value": "sm"
            },
            {
                "label": "Medium",
                "value": "md"
            },
            {
                "label": "Large",
                "value": "lg"
            }
        ]
    },
    {
        "id": "scoped_css",
        "group": "custom_styling",
        "label": "CSS",
        "type": "css",
        "info": "Add custom styles to this section only."
    },
    {
```

```
        "id": "text",
        "type": "richtext",
        "label": "Text",
        "default": "Welcome to our store"
    },
    {
        "id": "link",
        "type": "link",
        "label": "Link"
    }
]
```

Scopes

Scopes are used in sections to define which configs/templates they are available for use in.

Structure

Key	Description	Required?
`id`	The config or template that the scope applies for	Yes
`allowed`	If the section is allowed or not in that config/template	Yes

Example

```
[
    {
        "id": "header",
        "allowed": true
    }
]
```

> A section which the above scope will only be available for use inside the `header` config.

Presets

Presets are configurable options to avoid repetition.

Structure

Key	Description	Required?
`options`	An array of options	No
`options.*.value`	The value of the option	Yes
`options.*.label`	The label for the option	Yes

Example

```
{
    "options": {
        "padding": [
            {
                "label": "None",
                "value": "0"
            },
            {
                "label": "Extra extra small",
                "value": "xxs"
            },
            {
                "label": "Extra small",
                "value": "xs"
            },
            {
                "label": "Small",
                "value": "sm"
            },
            {
                "label": "Medium",
```

```
                "value": "md"
            },
            {
                "label": "Large",
                "value": "lg"
            },
            {
                "label": "Extra large",
                "value": "xl"
            }
        ]
    }
}
```

> Adding this preset to a setting will allow it to inherit these options without having to explicitly define them on it.

Variations

Variations are used in sections to define layout presets for blocks.

Structure

Key	Description	Required?
`id`	A unique identifier for the variation	Yes
`name`	The name of the variation	Yes
`blocks`	An array of Block objects	Yes

Example

```
{
    "id": "cms-grid",
    "name": "Content Grid",
    "settings": [],
```

```
    "blocks": [
        {
            "type": "image",
            "name": "Image",
            "settings": []
        },
        {
            "type": "text",
            "name": "Text",
            "settings": []
        }
    ],
    "variations": [
        {
            "id": "image-text",
            "name": "Image (Left), Text (Right)",
            "blocks": [
                {
                    "type": "image"
                },
                {
                    "type": "text"
                }
            ]
        },
        {
            "id": "text-image",
            "name": "Text (Left), Image (Right)",
            "blocks": [
                {
                    "type": "text"
                },
                {
                    "type": "image"
                }
            ]
        }
    ]
}
```

Revision #2
Created 2026-09-21 12:24:37 UTC by Ethan Payne
Updated 2026-10-02 13:06:36 UTC by Ethan Payne