

How to format your recommended page paths

Last Modified on 04/07/2026 3:45 pm EDT

To provide contextual help through the widget, you set which resources are recommended on which page by entering relative page paths or query strings in the editor. On this page, we walk through how to construct those page paths.

The overall process looks like this:

1. **Copy your page path or query string:** This is whatever portion of the URL appears after the base URL of the website or app where your widget is embedded. Refer to [Copy your page path or query string](#). If this path or string has no random IDs, you can use it as-is and don't need to do any other steps.
2. **Replace variable IDs with {id} or a custom regex:** If your URL includes an ID that changes (such as an account ID, user ID, project ID, or other object ID), replace that ID with either our pre-built `{id}` variable or create your own `{custom}` variable. Refer to [Replace variable IDs with {id}](#) for more detailed instructions.

Copy your page path or query string

The widget can recommend on pages based on **page path** or based on **query string**. (This setting can be changed in **KB settings > Widget** in the **Admin Settings** section. Refer to [Widget Admin Settings](#) for more detailed information and instructions.)

You can't mix query strings and page paths in the recommended URLs; you must commit to one.



General rule of thumb

Copy the portion of the URL that comes after your base URL to use as your **recommend on paths** in the editor.

For example, let's say we embedded our widget on `https://www.knowledgeowl.com`. That URL is our **base URL**.

Page paths are the portion of the URL that appears after the base URL and a slash. For example, if our pricing page URL is `https://www.knowledgeowl.com/pricing`, then the **relative page path** for this page is `/pricing`.

Query strings are any portion of the URL that appears after a question mark after the base URL. These usually contain **key-value pairs**. For example, if our pricing page URL is `https://www.knowledgeowl.com?page=pricing`, then the **query string** for this page is `?page=pricing`.

If your page path or query string has set values like this, where there's no variable or ID that changes for different users, you can enter this portion as-is into KnowledgeOwl's **Recommend On Pages** paths. Refer to [Recommend an article or category in contextual help](#) for more detailed instructions.

If the page path or query string contains an ID that changes, follow the [Replace variable IDs with {id}](#) instructions.

Replace variable IDs with {id}

Many URLs may contain an ID that might change between accounts, users, projects, or other objects. You don't want to have to enter every single possible ID as a recommended page path. Instead, replace that ID with `{id}` in your recommend on page path.

For example, let's say we display different pricing information to different accounts, so the URL might look something like this: `https://www.knowledgeowl.com/pricing/00000aaaaaa00000`.

To recommend articles on this page:

1. Copy the page path. In our example, that's: `/pricing/00000aaaaaa00000`.
2. Replace that random account number with `{id}`. In this example, the page path I enter in KnowledgeOwl is: `/pricing/{id}`
3. Enter that path in your article's **Recommend On Pages** paths. Refer to [Recommend an article or category in contextual help](#) for more detailed instructions.



Built-in `{id}` limitations

`{id}` only works with numeric and 24-digit hexadecimal IDs. For IDs that don't match either of those patterns, use the `{custom}` option via regex in **KB settings > Widget** and configure custom regex rules to handle your ID formats. Refer to the [Regex Replace URLs](#) section in [Widget Admin Settings](#) for more details.

How to handle other URL formats or cases where the URL doesn't update

If you're using a single page application or the URLs in your website/app don't fit well within this framework, your developers can set up the widget to call a function to send a path of your choice to the widget.

Refer to [Using Widget 2.0 on Single Page Applications](#) for more information.