



Set up Widget

Last Modified on 06/25/2025 12:02 pm EDT

Detailed instructions to set up and install the Widget in your website or web app.

Widget Settings section

Use the **Widget Settings** section to choose how you want an end-user to open the widget from your website or application.

We offer two widget launch methods:

- An **automatic circular button** with the "?" icon -- no coding required!
- **Attach the widget to an element of your choice** (such as a menu option, top navigation link, hyperlink, icon, image, etc.) by Element ID

Refer to the more detailed instructions below to set up either option.



Settings changes may require refresh due to browser caching

If you make changes to these settings, it may take up to an hour to see them in your browser due to caching. You can bypass the caching by doing a [hard refresh](#).

Automatic button

With the automatic button option, you don't need to adjust anything else on the website you're adding the code to. The script will automatically create the button and icon and place it based on the location setting you select.

To use this configuration, select any of the **Button Page Position** options other than Attach to Element (e.g. Bottom Right, Top Center, etc.):

Widget Settings when an icon display position is selected

1. Select the display location for the icon in the **Button Page Position** dropdown.
2. Enter a hex color in the **Button Text Color** to set the color of the question mark on the button.
3. Enter a hex color in the **Button Background Color** to set the color of the button.
4. Be sure to **Save** any changes you made here.

Here's what a widget button looks like if it's configured with:

- **Button Page Position:** Bottom Right
- **Button Text Color:** #ffffff (white)
- **Button Background Color:** #8080ff 

Sample website with embedded widget configured using the settings above

Attach to element

If you don't want to use the built-in button, you can launch the widget by attaching it to an element of your choice-- this can be a menu option, an icon, a hyperlink, etc.

For example, we use this option within app.knowledgeowl.com when you use the **Help > In-app help** menu.

To use this option, select **Attach to Element** from the **Button Page Position** dropdown:

Widget Settings when Attach To Element is selected

1. In the **Button Page Position** dropdown, select **Attach to Element**.
2. Once you select **Attach To Element**, the page hides the other Button Page fields and displays an **Element ID**. Set the Element ID to match the html ID attribute of the element you want the widget attached to in your website or application. When someone clicks on this element, the widget will open. If your developers are creating an element from scratch, we recommend using the default `ko-launch` Element ID or another obviously KnowledgeOwl-related Element ID. (You can't use `ko-widget`; we prevent that ID from being used, and you'll see an error message if you try to use it.)
3. Be sure to **Save** your changes.

For example, if we used "ko-launch" as our Element ID, we might add this HTML element to a top navigation option in our other website:

```
<li><a class="header-link" id="ko-launch" href="">Get Help</a></li>
```

Anytime an end-user selects the Get Help link, the widget will open.

Widget Admin Settings

Use the Widget's **Admin Settings** to control the functionality of the widget behind the scenes, including if/how it suggests content and if/how it's authenticated. Go to **KB Settings > Widget** to access these settings.



Settings changes may require refresh due to browser caching

If you make changes to these settings, it may take up to an hour to see them in your browser due to

caching. You can bypass the caching by doing a hard refresh.

Here's what each of the settings options does:

Recommended tab



Only for Contextual Help

This setting only applies if you're using the widget to automatically recommend content, as outlined in [Use Widget 2.0 for Contextual Help: Recommend on Pages](#).

If you check the box to **Display article when there is only one recommended**, when the widget opens and there's only a single recommended article or category, the widget will automatically open that article or category.

If you leave the box unchecked, the Recommended tab displays the list containing the single article or category.

This setting doesn't impact widget behavior when there are multiple recommended articles or categories for a given page.

Be sure to **Save** any changes you make.

Regex Replace URLs



Only for Contextual Help

This setting only applies if you're using the widget to automatically recommend content, as outlined in [Use Widget 2.0 for Contextual Help: Recommend on Pages](#).

In order for Contextual Help recommendations to work, the widget has to parse the URLs it finds to determine which content to show. Since URLs often contain ID values that might vary from user-to-user but the same overall path, Widget 2.0 includes a built-in regex that uses an `{id}` variable. `{id}` only works with numeric and 24-digit hexadecimal IDs.

If your website or application URLs have IDs that don't match those patterns:

1. Define the regex you need the widget to use in **Regex Replace URLs**.
 - a. [regex101](#) is a handy site that allows you to try out Regex rules against example words and phrases. When testing regex for use in KnowledgeOwl, select **PCRE2 (PHP >= 7.3)** under **FLAVOR**.
2. Be sure to **Save** any changes you make.
3. Advise your authors to use the `{custom}` variable in your **Recommend on Pages URLs**, instead of `{id}`.

Suggested content



Only for Contextual Help



This setting only applies if you're using the widget to automatically recommend content, as outlined in [Use Widget 2.0 for Contextual Help: Recommend on Pages](#).

Use the **Suggested content** settings to control which portion of the URL the widget uses to identify recommended content and whether the widget should "learn" recommended resources based on what end-users on this page ultimately open in the widget.

First, select one of these options to set which portion of the URL the widget should use to identify recommended content:

- **Suggest content based off of the page path (default):** Use this setting if the website where you're embedding the widget uses the URL or page path to differentiate pages.
 - For example, we use this setting for the widget in this Support knowledge base, since our page URLs are formatted without query strings and with no special characters.
- **Use the page query strings to suggest content:** Use this setting if you want to recommend based on query strings, which appear in the URL as `?variable=`, such as `?s=` or `?client=`.
 - If your website's URLs don't have query strings--or the query strings aren't meaningful for recommendation distinctions--don't use this setting.
- **Disable automatic page suggestions:** This option effectively turns off automatic page suggestions. If you select this option, the widget will only provide suggestions when the `_ko19.updateRecommended` method is being used. Use this setting if:
 - You plan to consistently override the page path or query string with our [Single Page Application](#) methods.
 - or
 - You don't want the widget to suggest content at all.

Next, decide whether you want the widget to "learn" additional articles to recommend.

As people open the widget, we track which articles or categories people end up viewing from a given page. We save this information as the **Learned Bias** in [Widget reporting](#). Once the Learned Bias of a resource gets high enough, the widget includes that article or category in the Recommended tab list automatically. This is the behavior your widget has if you leave the **Exclude articles that have a zero starting weight** box unchecked.

If you want tight control over the content that you're recommending, you'll want to disable the learning behavior. To prevent the widget from displaying these "learned" recommended pages, check the box to **Exclude articles that have a zero starting weight**. When this box is checked, the widget only recommends articles and categories you've explicitly assigned in the Recommend On Pages (it basically ignores anything that only has a learned bias).

Be sure to **Save** any changes you make.

OAuth Authentication

If you want or need to authenticate your widget using OAuth2, use this section to generate the OAuth client secret, enable OAuth authentication, or regenerate the client secret.

Refer to [Widget 2.0 OAuth2 authentication](#) for full instructions. Refer to [Widget authentication--when and how to use it](#) to figure out if you need authentication.

JWT Authentication

If you want or need to authenticate your widget using JWT, use this section to grab your JWT client secret, enable JWT authentication, or regenerate the client secret.

Refer to [Widget 2.0 JWT authentication](#) for full instructions. Refer to [Widget authentication--when and how to use it](#) to figure out if you need authentication.

Widget Customize Text section

In KB Settings > Widget, the Customize Text section allows you to change a lot of the text in the widget:

- The Knowledge, Recommended, and Contact tab labels
- The header text displayed in the Knowledge and Recommended tabs
- The Open in full site and Scroll to top actions in the footer
- The search bar placeholder and the header text displayed above Quick Search and Full Search results
- Add a custom footer that will display below the open in full site and scroll to top footer



Settings changes may require refresh due to browser caching

If you make changes to these settings, it may take up to an hour to see them in your browser due to caching. You can bypass the caching by doing a hard refresh.

If you want to change any of this text, check out the screenshot and descriptions below so you know you're changing the right field(s)!

Knowledge tab

The widget's Knowledge tab displays your full knowledge base and lets your readers browse and navigate your knowledge base. All articles selected in other tabs open in the Knowledge tab.

On the Knowledge tab, you can customize:

1. The tab's "Knowledge" label: Edit the **Knowledge Tab** text.
2. The text displayed between the search bar and your knowledge base's table of contents [default: "How can we help you?"]
3. Search bar text--refer to [Search box](#) below for more information.



Recommended tab

If you're using [Contextual Help](#) and Pages to Recommend On, the widget displays recommended articles for the current page in the Recommended tab.

On the Recommended tab, you can customize:

1. The tab's "Recommended" label: Edit the **Recommended Tab** text.

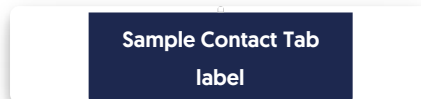
2. The text displayed between the search bar and your list of recommended articles (default: "Recommended Articles for You")
3. Search bar text--refer to [Search box](#) below for more information.



Contact tab

If you're using the [Contact Form](#) in your knowledge base, the widget displays a Contact tab for your readers to submit requests.

On the Contact tab, you can customize the tab's "Contact" label by editing the [Contact Tab](#).



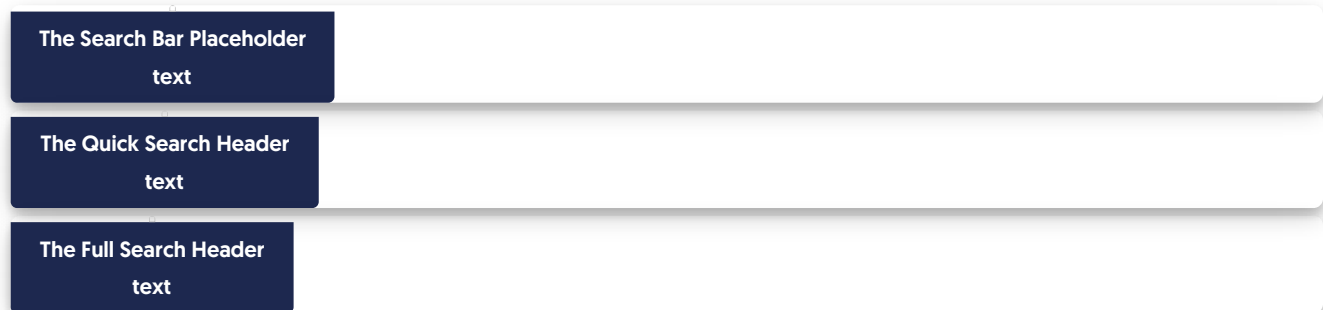
Search box

The search box displays on both the Knowledge tab and the Recommended tab. Once someone enters a search term, quick search displays autosuggested resources along with guidance on how to complete a full search. Once someone hits Enter to complete the full search, more detailed search results display.

You can customize this search-related text:

1. The search bar's "Search for articles..." placeholder: Edit the [Search Bar Placeholder](#) to change this text.
2. The header text displayed between the search bar and the autosuggested search results: Edit the [Quick Search Header](#)
3. The header text displayed between the search bar and the full search results list: Edit the [Full Search Header](#) to change this text.

Here are three sample screenshots showing where each text string displays in the widget:



Widget footer

All tabs of the widget display a standard footer containing two options:

1. **Open in full site:** This link opens a new browser tab with the widget's current page opened in your full knowledge base. Change the text for this hyperlink by editing the **Open in Site**.
 - a. Knowledge tab: Opens to whatever page you're currently viewing, or the homepage if you're just viewing the table of contents.
 - b. Recommended tab: Opens to the homepage.
 - c. Contact tab: Opens to the [Contact Form](#).
2. **Scroll to Top:** This link is shown on non-touch devices so that if someone is scrolling through a lengthy article in the widget, they can quickly jump back to the top of the article. Change the text for this function by editing the **Scroll to Top**.
3. The widget also supports a custom footer, which displays below the standard two links. Add raw HTML to the **Custom Footer Content** to add, edit, or remove that custom footer. In the screenshot below, we've entered this HTML in the **Custom Footer Content**:

```
<span style="text-align:center;">This custom footer powered by KnowledgeOwl</span>
```

Sample default widget footer

text

Sample widget footer with Custom Footer Content

entered

Widget Contact Form section

In **Settings > Widget**, the **Contact Form** section allows you to change some of the text labels and placeholder text in the widget's Contact tab, including:

- Field labels
- Search results header
- Thank you message
- Add a custom footer



If you make changes to these settings, it may take up to an hour to see them in your browser due to caching. You can bypass the caching by doing a hard refresh.

Contact Form

Subject 1

Details 2

Name 3

Email Address 4

Attachment 5

Button 6

Suggested Resources 7

Expand Suggested Resources 8

Thank You Message 9

Custom Footer HTML 10

Widget Contact Form settings

The image displays four sequential screenshots of the 'Widget 2.0 (beta)' contact form, illustrating various settings and states. Each screenshot has a header with tabs: 'Knowledge', 'Recommended', 'Contact', and a close button 'X'.

- Screenshot 1:** Shows the initial form. Callouts indicate: 1. Subject field, 2. Details field, 3. Your name field, 4. Your email address field, 5. Upload an Attachment button, 6. Send email button, and 10. Custom Footer HTML (contact information).
- Screenshot 2:** Shows the form after the Subject field is filled with 'subs'. Callout 7 points to the 'Suggested content' section, which lists links like 'Subscribing to a category' and 'Requirements for enabling subscriptions'.
- Screenshot 3:** Shows the form after submission. Callout 8 points to the 'Suggested content' section, which now displays a 'Thanks! Your request has been sent.' message.
- Screenshot 4:** Shows the final state of the form, identical to the previous one, with the 'Thanks! Your request has been sent.' message.

At the bottom of the screenshots, a dark blue bar contains the text: 'Widget Contact Form settings in widget'.

1. **Subject:** Text entered here changes the label for the Subject field.
2. **Details:** Text entered here changes the label for the Details field.
3. **Name:** Text entered here changes the label for the Name field.
4. **Email Address:** Text entered here changes the label for the Email address field.
5. **Attachment:** Text entered here changes the label for the Upload an Attachment field.
6. **Button:** Text entered here changes the label for the button that submits the contact form.
7. **Suggested Resources:** Text entered here changes the header above the Suggested Articles once a reader types into the Subject field.
8. **Expand Suggested Resources:** Text entered here changes the text next to the expand arrow when the suggested resources are collapsed.
9. **Thank You Message:** Text entered here changes the Thank you message that appears after a reader submits the contact form.
10. **Custom Footer HTML:** You can add a custom footer which will appear below the main body of the contact form. This field accepts HTML, so you can include links to a privacy policy, data retention policy, etc.



As of July 2023, the Widget 2.0 Contact Form does not incorporate Custom Fields or ZenDesk fields in the main Knowledge Base > Contact Form settings. [Contact us](#) if you'd like to be added to the feature request for any of these fields.

Custom Widget Styles

In addition to changing the widget [tabs](#), [labels](#), and [headers](#) by using the built-in settings, you can also customize its look and feel using CSS and Javascript.

Use the **Custom Widget Styles** section of **KB settings > Widget** to add custom CSS to style your widget.



Quick view

If you **Save** changes to these settings, they should show up nearly instantly in your live widget.

You can play around with a right-click and inspect on various fields or areas in the widget to try to identify the classes being used. Here are a few of the most frequently changed styles. You can copy and paste these in to your Custom Widget Styles section and then change the appropriate text to make adjustments.

Change the background color of the tabs

To use this code, update the value used for background-color in row 3 with the color of your choice. Here, we use a hex code but you can use rgb values, like `rgb(0, 151, 167)` .

```
/* Tab color */
.widget .widget__nav a {
  background-color: #0097a7;
}
```

Change the font on the tab labels

To use this code, update the font-family in row 3 from Neuton to a family of your choice.

```
/* Tab label font used (must be defined on the site where widget is being used) */
.widget .widget__nav a {
  font-family: Neuton;
}
```

Change the color of all hyperlinks

To use this code, update the value used for color in row 7 with color of your choice. Here, we use a hex code but you can use rgb values, like `rgb(0, 151, 167)` .

```
/* Change the color of hyperlinks */
.article-links a,
.list-links a,
.list-links i,
.btn-upload,
a {
  color: #1f9baa;
}
```

Change the contact form submit button color

To use this code, update the value used for background-color in row 3 with the color of your choice. Here, we use a hex code but you can use rgb values, like `rgb(0, 151, 167)` .

```
/* Contact Form submit button color */
.form .form-submit {
  background-color: #fc9f00;
}
```

Change the > icon next to articles in the Knowledge tab and search results to a different icon

We use a FontAwesome chevron-right icon next to the titles of articles in the Knowledge tab and the search results.

If you'd like to change this to use a different Font Awesome icon, you'll need to know the unicode for the [Font Awesome](#) icon you'd like to use. Replace the `f15c` in row 5 below with that unicode. Be sure you keep the `"\"` around it:

```
/* Change the default ">" icon in the Knowledge tab navigation (chevron-right)
and the Search results (angle-right), including Contact Form search results */
.list-links .fas.fa-chevron-right:before,
.list-links .fas.fa-angle-right:before {
  content: "\f15c" !important;
}
```

Hide one of the widget's tabs

You can use CSS to hide one or more of the tabs from the widget. Refer to [Hide one of the Widget 2.0 tabs](#) for full instructions.

Custom article styles in the Widget

Some authors want to style article elements differently in the widget from their main knowledge base (such as article title, etc.).

For these customizations, instead of using **KB settings > Widget**, you'll add the Custom CSS directly to **Customize > Style (HTML & CSS)**. Preface your class selectors with `.w-documentation-article` , which targets articles displayed in the widget rather than in the knowledge base.

1. Go to **Customize > Style (HTML & CSS)**.
2. In the **Customize HTML, CSS, and JS** section, select **Custom CSS**.
3. Add the Custom CSS for your widget article styles.
4. **Save**.

For example, `h1.hg-article-title` is the CSS selector for the article title in your full knowledge base. This sample code will let you set the article title to a 20px font-size in the widget only (you can update the font-size in row 2 to any size you'd like):

```
.w-documentation-article h1.hg-article-title {  
  font-size: 20px;  
}
```

Custom Widget Javascript

For additional customizations of behavior in the widget (such as removing some fields entirely, changing the order, etc.), you'll need to use Javascript (JS) in the widget.



If you make changes to these settings, they should show up nearly instantly in your live widget.

To add widget JS:

1. Go to KB settings > Widget.
2. Add your JS to the Custom Widget Javascript section.
3. Save your changes.



No script tags needed

You don't need to add `<script>` tags around code entered in the Custom Widget Javascript section-
-we automatically wrap this box in script tags!

Working with the Contact tab

Javascript for the contact form needs to go inside the `onTabChange` function. Inside this function, you have access to `data.oldTab` and `data.newTab` to hook onto the `tabChange` events:

```
//JS for the contact form needs to go inside onTabChange function  
var onTabChange = function(data) {  
  //you have access to the data.oldTab and data.newTab to hook onto these events if it helps. Example:  
  console.log("Tab changed from " + data.oldTab + " to " + data.newTab);  
  //do some code down here  
}
```

For example, this code will hide the "Upload file attachment" option in the contact form:

```
//JS for the contact form needs to go inside onTabChange function  
var onTabChange = function(data) {  
  //simple example of hiding an option  
  //hide file attachment option  
  $('form__upload').hide();  
}
```

Add spam protection to Widget 2.0 Contact Tab

If you're using Contextual Help Widget (2.0) on a publicly-available site , we recommend adding spam protection to your contact form and contextual help widget.

You'll need to follow these steps if all the following conditions apply:

- You've enabled the [Contact Form](#) in your knowledge base.
- You've embedded the [Contextual Help Widget \(2.0\)](#) in a public website or application.
- You want to protect the Contextual Help Widget's Contact tab with [spam protection](#).

Find your setup in the table below, then select the link to jump to relevant instructions.

Contact Form spam protection enabled?	Knowledge base domain	Spam protection setting	Link to instructions	reCAPTCHA Key/Secret V3 status	Widget end-user experience	Contact tab spam protection
No	KO subdomain or private domain	Any	N/A	Not required	No different; tab submits	None
Yes	KO subdomain	Honeypot	KO subdomains	N/A	No different; tab submits unless end-user violates honeypot restrictions	Honeypot
Yes	Private domain	Honeypot	Private domains and honeypot	N/A	No different; tab submits unless end-user violates honeypot restrictions	Honeypot
Yes	KO subdomain	Google reCAPTCHA	KO subdomains	Not required; automatically applied	Contact tab displays reCAPTCHA footer statement; tab submits	Google reCAPTCHA
Yes	Private domain	Google reCAPTCHA	Private domains and reCAPTCHA	Valid key/secret required	Contact tab displays reCAPTCHA footer statement; tab submits	Google reCAPTCHA
Yes	Private domain	Google reCAPTCHA	Haven't added V3 yet	Key/secret required but not yet added	No different; tab submits	None

Contact Form spam protection enabled?	Knowledge base domain	Spam protection setting	Link to instructions	reCAPTCHA Key/Secret V3 status	Widget end-user experience	Contact tab spam protection
Yes	Private domain	Google reCAPTCHA	V3 added but receiving error	Key/secret required but invalid	Contact tab displays reCAPTCHA statement; reCAPTCHA error thrown when reader tries to submit	Yes, but submissions are prevented

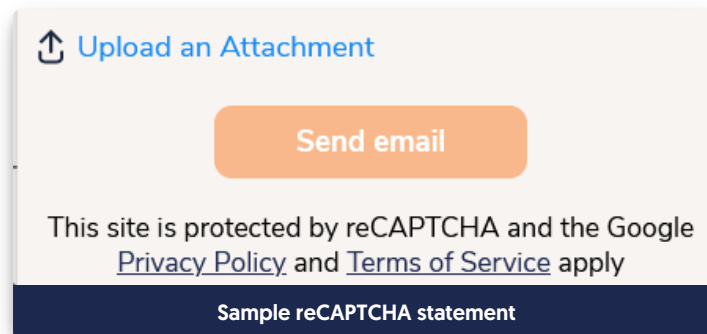
How it works

The Contextual Help Widget's Contact tab is aware of the spam protection setting in **Customize > Contact form** and will use your **Default method** set in **Security and access > Spam protection**, just as it's used in the Contact form.

For spam protection, you can choose between Google [reCAPTCHA](#) and KnowledgeOwl's homegrown honeypot solution. Refer to [Spam protection](#) for more information.

If your knowledge base uses honeypot spam protection, no change to the widget display or behavior is visible to your end-users.

If your knowledge base uses Google reCAPTCHA spam protection, the Widget Contact tab uses reCAPTCHA V3, which is a silent/invisible reCAPTCHA--no boxes to check or fire hydrants to find. reCAPTCHA calculates a score behind-the-scenes to determine if your reader is a bot or a human. Widget readers who fail the invisible V3 will not be able to submit the Contact tab. The Contact tab also displays a required statement just below the Submit button:



This statement is part of Google reCAPTCHA's terms of service. You can't edit or remove it as long as you're using reCAPTCHA to protect your widget.

KnowledgeOwl subdomains

As with all spam protection settings, if you're using a KnowledgeOwl subdomain (like [mykb.knowledgeowl.com](#)), we handle the spam protection maintenance for you.

You'll need to change one setting to protect your widget with honeypot or Google reCAPTCHA spam protection:

1. Go to **Customize > Contact form**.
2. Check the **Admin Settings** box to **Use spam protection on ticket submissions**.
3. Be sure to **Save** your changes.

We'll apply your knowledge base's default [Spam protection](#) method to the contextual help widget.

If your default spam protection method is set to **Google reCAPTCHA**, the widget's Contact tab displays a required Google reCAPTCHA statement.

If your default spam protection method is set to honeypot, no visible message or alert displays in the widget's Contact tab.

Private domains and honeypot

If you're using your own domain (like [help.owlsareawesome.com](#)) and your **Security and access > Spam protection > Default method** is set to **Honeypot**, you'll need to change one setting to protect your widget with honeypot:

1. Go to **Customize > Contact form**.
2. Check the **Admin Settings** box to **Use spam protection on ticket submissions**.
3. Be sure to **Save** your changes.

Private domains and Google reCAPTCHA

If you're using your own domain (like [help.owlsareawesome.com](#)) and your **Security and access > Spam protection > Default method** is set to **Google reCAPTCHA**, you must add your own V3 reCAPTCHA key and secret to protect your widget's Contact tab:

1. Go to **Customize > Contact form**.
2. Check the **Admin Settings** box to **Use spam protection on ticket submissions**.
3. Be sure to **Save** your changes.
4. Now go to **Security and access > Spam protection**.
5. In the **Custom Google reCAPTCHA integration** section, add a **reCAPTCHA Key V3** and **reCAPTCHA Secret V3**. Refer to the more detailed instructions in [Add reCAPTCHA](#).

Once you've added your V3 key, the widget displays a required Google reCAPTCHA message near the footer:

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Contact Form has spam protection but I haven't added my V3 reCAPTCHA key yet

If you haven't yet entered a V3 reCAPTCHA key, **don't worry!** Your widget Contact tab will still work. In this state, you won't see the reCAPTCHA statement in the footer and you'll have no reCAPTCHA protections. But the widget can still accept Contact form submissions.

Follow the instructions in [Private domains and Google reCAPTCHA](#) above to set up your key.

I have a V3 reCAPTCHA key but my widget displays a reCAPTCHA error

If you enter an invalid V3 reCAPTCHA key/secret, when a reader selects **Submit** in the Contact tab, they'll receive an error instead of the confirmation message that their request was submitted. The text reads something like: "reCAPTCHA error, could not submit request". Here's a sample of what it looks like to a reader:

The screenshot shows a contact form with a purple header bar containing four tabs: 'Knowledge', 'Recommended', 'Contact', and 'X'. The 'Contact' tab is active. The form contains the following fields and elements:

- Subject:** A text input field containing 'invalid key test'.
- Suggested content:** A dropdown menu with a downward arrow.
- Details:** A large text area containing 'invalid key test'.
- Your name:** A text input field containing 'invalid key test'.
- Your email address:** A text input field containing 'support@knowledgeowl.com'.
- Upload an Attachment:** A link with an upward arrow icon.
- Error Message:** A red text line stating 'reCAPTCHA error, could not submit request.'
- Send email:** A purple button.
- Footer:** Text stating 'This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply'.
- Bottom Bar:** Two links: 'Open in full site' (with an external link icon) and 'Scroll to Top' (with an upward arrow icon).

To resolve this issue, re-enter your reCAPTCHA V3 Key and Secret and be sure you enter the correct values in the correct fields.

Open widget to a specific article or category

Some customers like to be able to open the widget directly to a specific article or category. The steps for each are quite similar.

To open the widget to a specific article, you'll need to use two methods:

- `_ko19.openArticle('article-identifier');` -- this loads the specific article you want into the widget. For the *article-identifier*, you can use one of two options:
 - The article's relative permalink (for example, for this article, it would be `open-widget-to-a-specific-article`)
 - The article's ID (found by opening the article in edit mode: the ID is the number that comes after `/aid/` in the URL)
- `_ko19.open();` --this call opens the widget

Using only the `openArticle` method will prime the widget to open to a specific article the next time it's opened by your reader.

Similarly, to open the widget to a specific category, you'll need to use two methods:

- `_ko19.openCategory('category-identifier');` -- this loads the specific article you want into the widget. For the *category-identifier*, you can use one of two options:
 - The category's relative permalink (for example, for the Widget 2.0 category, it would be `widget-20`)
 - The category's ID (found by opening the category in edit mode: the ID is the number that comes after `/cid/` in the URL)
- `_ko19.open();` --this call opens the widget

A good way to use these methods is on a specific hyperlink in your documentation, to open the widget to a specific article. In this case, you'll insert a hyperlink with an empty URL, and add two onclick events--one for each of the methods above.

So, for example, this link [opens the widget directly to the article](#) you're reading right now. Here's the code for it:

```
<a href="#" name="" onclick="_ko19.openArticle('open-widget-to-a-specific-article'); _ko19.open();return false;" title="Open widget to a specific article or category">Open widget directly to an article</a>
```

So, for example, this link [opens the widget directly to the Widget 2.0 category](#) you're reading right now. Here's the code for it:

```
<a href="#" name="" onclick="_ko19.openCategory('widget-20'); _ko19.open();return false;" title="Widget 2.0">opens the widget directly to the Widget 2.0 category</a>
```

Scrolling behavior

For the hyperlinks, you can have different scrolling behavior on the main page when the link is clicked by formatting the link differently:

- Use an `href="#"` by itself to force the main page to scroll to the top when the widget is opened, as in this link [opens the widget directly to the article](#) you're reading right now.

```
<a href="#" name="" onclick="_ko19.openArticle('open-widget-to-a-specific-article'); _ko19.open();" title="Open widget to a specific article or category">Open widget directly to an article</a>
```

- Use an `href="#"` with a `return false;` at the end of the onclick to prevent the scroll (as used in the examples above)
- You can also use `href=""` to achieve the no-scroll behavior:

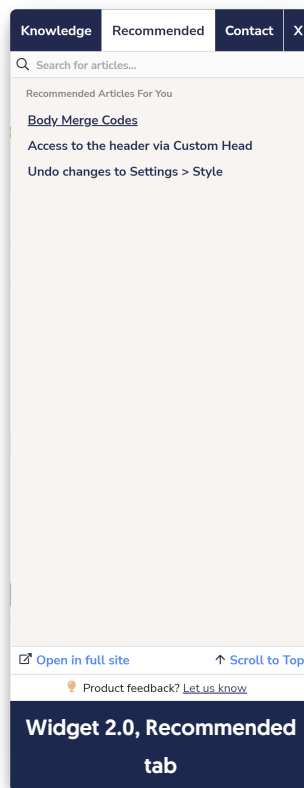
```
<a href="" name="" onclick="_ko19.openArticle('open-widget-to-a-specific-article'); _ko19.open();" title="Open widget to a specific article or category">Open widget directly to an article</a>
```

Use Widget 2.0 for Contextual Help: Recommend on Pages

Recommend On Pages allows you to provide contextual help to your readers through the embeddable [Contextual Help Widget \(2.0\)](#) or the API. You can provide a list of relevant articles based on the page the person opened the widget from, which may eliminate the need for them to search the knowledge base or contact you.

You can relate individual articles to pages in your knowledge base, website, or application so that they are suggested when the widget loads on the page, or you use the API to return articles for a particular page in your site. For example, if you associate articles with the `"/pricing"` page on your website, these articles appear when the contextual help widget is opened on that page. Alternatively, if you associate articles with the `"/account"` page in your application and use the API to pull articles for that page, these articles will be returned for your `"/account"` page.

In Widget 2.0, the Recommend on Pages will update the Recommended tab list:



In order to start using this feature, you'll need to:

- Set the Widget Settings to use the appropriate **Suggested Content** setting
- Add URLs from your main application or website to the articles or categories you'd like to recommend in the widget or API

Configuration - Widget settings

First, in **Settings > Widget**, check your widget's [Admin Settings](#) and be sure that the **Suggested Content** option best matches how your website or application is set up. This is usually a one-time set-up.

- Suggested Content**
- ☒ Suggest content based off of the page path
 - ☐ Use the page query strings to suggest content
 - ☐ Disable automatic page suggestions

Using the manual updateRecommended(page) function will continue to work.

The Suggested Content options in the Admin Settings section

Formatting your page paths

Always use relative URLs. If we are configuring articles for <https://www.knowledgeowl.com>, take the portion of the URL that comes *after* that URL. For example, <https://www.knowledgeowl.com/pricing> becomes `"/pricing"`. Specifically, the pathname of the URL is sent. The fragment and query string are not sent (unless you've selected the option to use query strings! Then you'd want to use query strings here.)

If your website or application includes URLs that have a variable ID field in them, you can replace that ID with `{id}` in your URL.



`{id}` only works with numeric and 24-digit hexadecimal IDs. For IDs that don't match either of those patterns, you would need to use the `{custom}` option via regex in **Settings > Widget** and configure custom regex rules appropriately.

You may also need to use some custom regex to handle ID values in your URLs; see the **Regex Replace URLs** section in [Widget Admin Settings](#) for more details.

Once you have the relative URL for pages you want to specify recommended articles or categories for, you can get started.

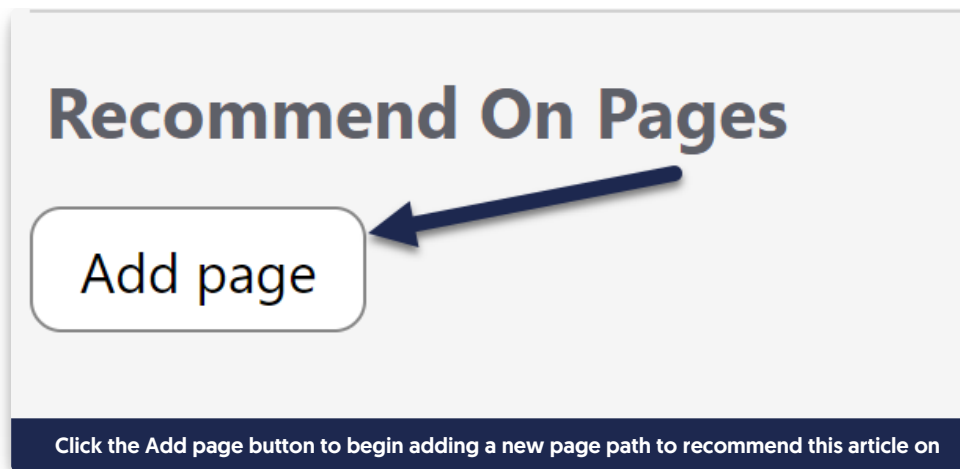
If you are using a single page application or the URLs in your website/app don't fit well within this framework, you can call a function to send a path of your choice to the widget. See [Using Widget 2.0 on Single Page Applications](#) for more information.

The interface is a little different depending on whether you're adding articles or categories, so let's look at each in turn:

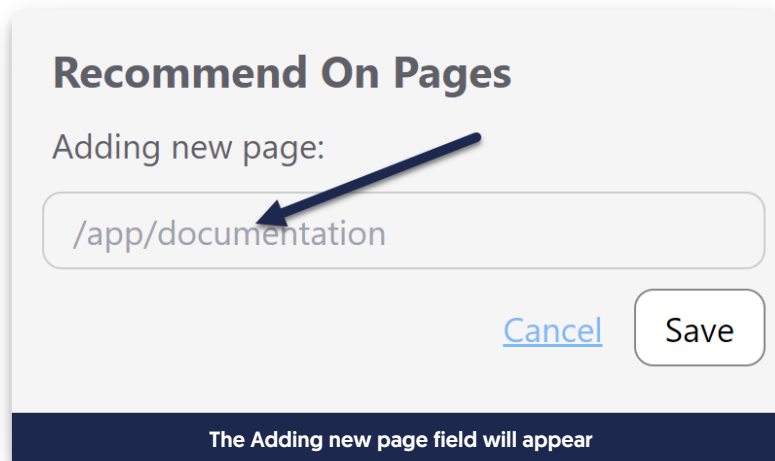
Recommending Articles

To set up an article as being recommended:

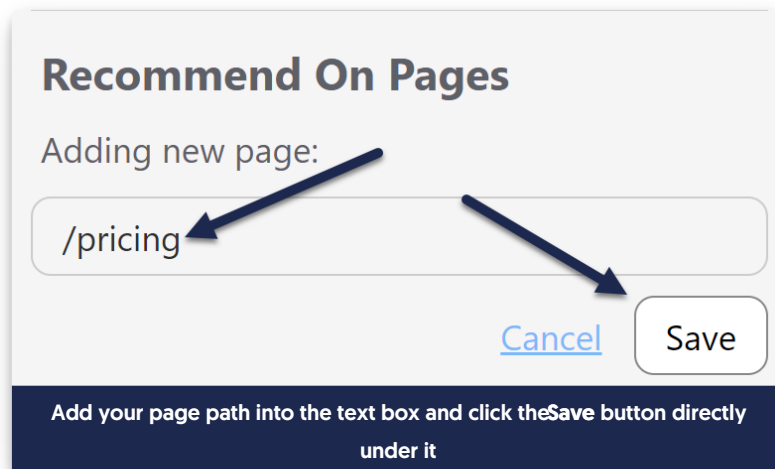
1. Open the article for editing.
2. Find the **Recommend on Pages** section in the righthand column.
3. Select **Add page** to begin adding a new page.



4. This will open an "Adding new page" section. Add your page path into the text field with the /app/documentation placeholder:



5. Once you've finished adding your page path, click the Save button directly under it.



6. The interface will update to show the page path you added:





7. Repeat this process for any other pages you want to recommend this article on.

8. Be sure to **Save** your article once you're done adding paths.

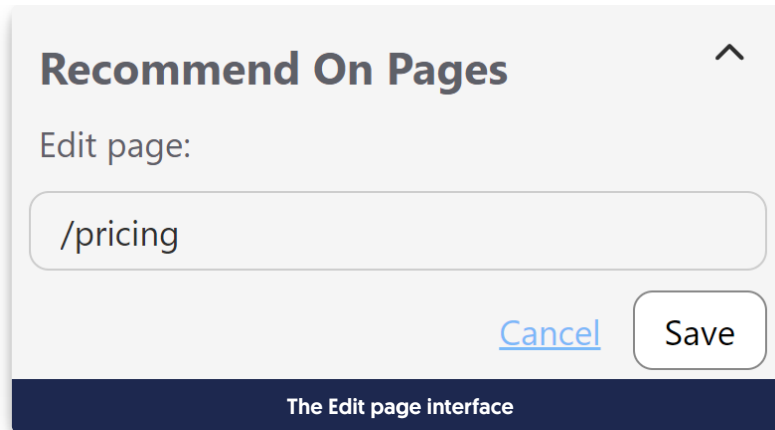
Update an existing recommended article path

To update an existing **Recommend On Pages** path:

1. Open the article for editing.
2. In the **Recommend On Pages** section, click on the text of the path you want to edit. Here, we'll edit our /pricing page path:



3. Once you click on the path, the interface adjusts to say "Edit page". You can now edit the path in the text box.



Recommend On Pages ^

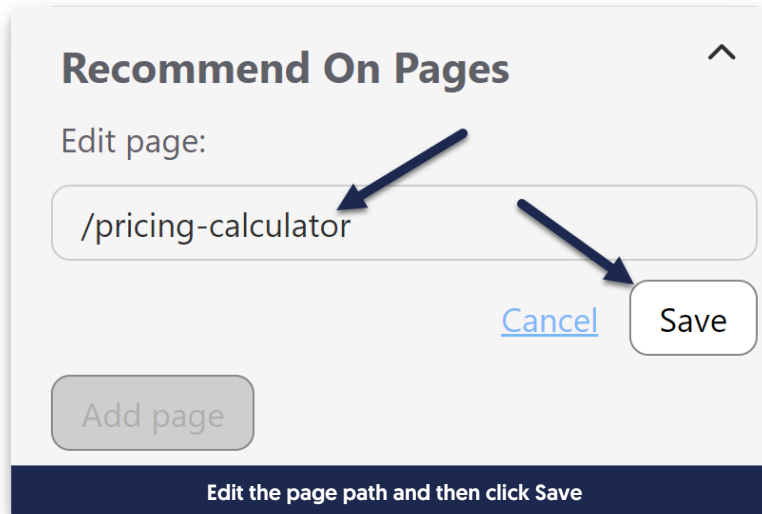
Edit page:

/pricing

[Cancel](#) **Save**

The Edit page interface

4. Once you've finished making your edits, click the **Save** button directly under the page path:



Recommend On Pages ^

Edit page:

/pricing-calculator

[Cancel](#) **Save**

Add page

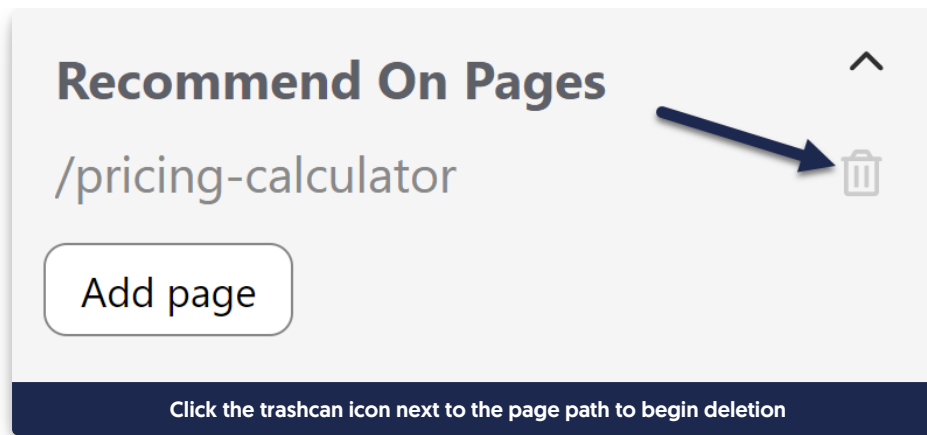
Edit the page path and then click Save

5. Be sure to **Save** your article once you're done making your edits!

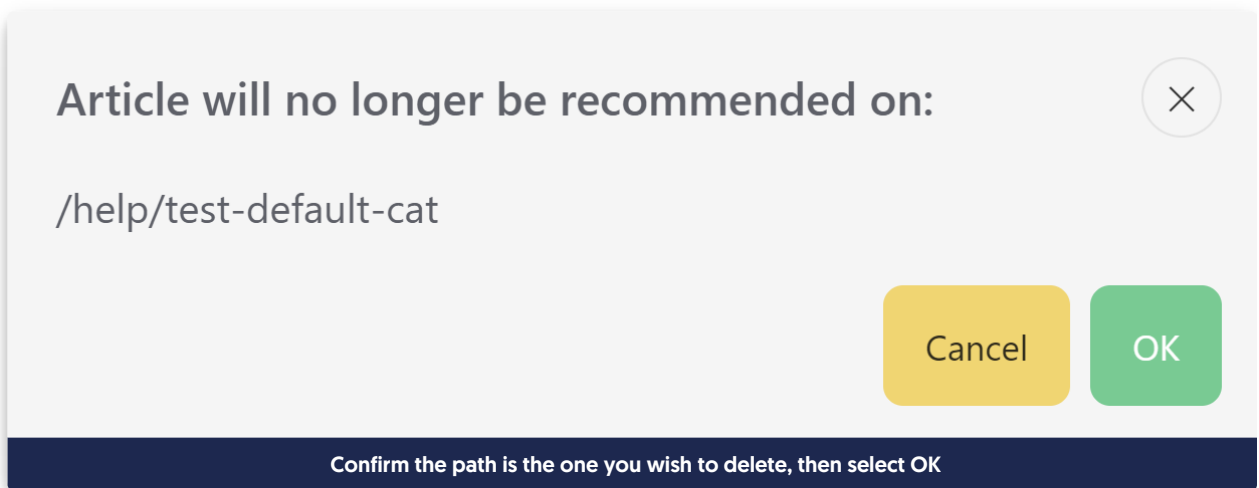
Delete an existing recommended article path

To remove a page path you no longer want to recommend this article on:

1. Open the article for editing.
2. In the **Recommend On Pages** section, click the trashcan icon to the right of the page path you want to remove:



3. In the confirmation message that appears, confirm that the page path is the one you meant to delete.
4. If it's the correct page path, click **OK** to complete the deletion.



5. The interface updates to remove the path.
6. Once you're done making changes, be sure to **Save** your article!

Recommending Categories

To set up a category as being recommended:

1. Open the category for editing.
2. Find the **Recommend on Pages** section in the righthand column.
3. Add the relative URL to the **Recommend On Pages** section, then select **Save**.

Recommend On Pages: ?

/pricing

Add one path per line

4. If you want this article to be recommended on multiple pages in your application or website, add each relative URL on a separate line:

Recommend On Pages: ?

/pricing
/product

Add each new page path on a new line

5. Be sure to **Save** the article. The next time someone opens the widget from one of those pages in your site or application, it will suggest this category.

Update an existing recommended category path

To update an existing **Recommend On Pages** path in a category:

1. Open the category for editing.
2. Find the **Recommend on Pages** section in the righthand column.
3. Click in the text box there and edit the page path you want to adjust.
4. Be sure to keep each page path on its own line in the box.
5. Once you're done making edits, be sure to **Save** the article.

Delete an existing recommended category path

To remove a page path you no longer want to recommend this category on:

1. Open the category for editing.

2. Find the **Recommend on Pages** section in the righthand column.
3. Click in the text box there and delete the page path you want to adjust.
4. Be sure the remaining page paths are all on their own lines in the box.
5. Once you're done making edits, be sure to **Save** the category.

Ordering recommended articles

If you've added several recommended articles and you'd like to set the order they display in, you can change the order in by editing the article weights for that page in **Widget > Reporting**. See [Recommended Article Weights](#) for more information.

Hide one of the Widget 2.0 tabs

If you don't want to use one of the tabs in Widget 2.0, you can hide it:

1. Go to **Settings > Widget**.
2. Copy the relevant code below and paste it into the **Custom Widget Styles** section:
 - a. To hide the Knowledge tab, copy and paste this code:

```
/* Hide the Knowledge tab */
.widget .widget__nav li:nth-child(1) {
  display:none !important;
}
```

- b. To hide the Recommended tab, copy and paste this code:

```
/* Hide the Recommended tab */
.widget .widget__nav li:nth-child(2) {
  display:none !important;
}
```

- c. The Contact tab will only display if you have enabled Contact Form in your knowledge base. If you have Contact Form enabled but you want to hide the Contact tab, copy and paste this code:

```
/* Hide the Contact tab */
.widget .widget__nav li:nth-child(3) {
  display:none !important;
}
```

3. Be sure to **Save** your changes to the Widget settings.

Widget 2.0 installation

Once you've [configured and saved your widget settings](#), you'll need to add the widget embed code into your website or web-based application:

1. Go to KB settings > **Widget**.
2. Copy the code in the **Widget Embed Code** section.
3. Paste that embed code before the closing `</body>` tag in your website or application.

Depending on your configuration and requirements, you may need to perform additional setup steps, such as:

- If your knowledge base requires authentication, review whether you need to [authenticate your widget](#) and configure [OAuth2 authentication](#) or [JWT authentication](#).
 - If your knowledge base uses a private domain and is publicly available, make sure your [Contact Form](#) has the box checked to **Use spam protection on ticket submissions** to prevent spam and potential DDoS attacks. Refer to [Protect Widget 2.0 Contact Form with reCAPTCHA](#) for more information.
 - If you're using the widget within a single page application, refer to [Using Widget 2.0 on Single Page Applications](#).
 - If your URLs contain IDs, figure out if you can use the built-in `{id}` variable (for numeric and 24-digit hexadecimal IDs) or if you need to write your own **Regex Replace URLs** regex and use the `{custom}` variable instead. See [Widget Admin Settings](#) for more information.
 - Refer to [Advanced widget configuration & methods](#) for additional methods you can use to further control the widget opening behavior, such as opening to specific articles or categories, opening to a specific tab, and more.
-