

Web page reader

Last Modified on 17/07/2025 9:43 pm AEST

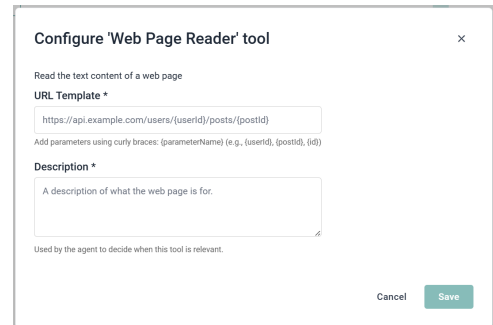
← [Go back to Agent Home Page](#)

i This tool is currently in Alpha and has been released for early access and feedback

It may include incomplete features, limited functionality, or be subject to change. Alpha tools are intended for evaluation and validation purposes only and should **not** be used in production environments.

Adding the tool to an Agent

1. Open the **Agent Builder**.
2. From the left-hand menu, go to the **Tools** section.
3. Click **Add Tool**.
4. Select and click **Web Page Reader** from the list of available tool types.
5. The **Configure 'Web Page Reader' Tool** dialog will appear.



The screenshot shows a dialog box titled "Configure 'Web Page Reader' tool" with a close button (X) in the top right corner. The dialog contains the following fields and text:

- Read the text content of a web page**
- URL Template ***
A text input field containing the URL: `https://api.example.com/users/{userId}/posts/{postId}`
- Add parameters using curly braces: {parameterName} (e.g., {userId}, {postId}, {id})
- Description ***
A text input field containing the text: "A description of what the web page is for."
- Used by the agent to decide when this tool is relevant.
- At the bottom right, there are two buttons: "Cancel" and "Save".

Configuring the tool

Follow these steps to complete the configuration:

1. **Enter a URL Template**
 1. Enter the full URL of the web page the agent should read.
 2. You can include dynamic parameters by wrapping them in curly braces (e.g. `{parameter}`)
 3. These parameters will be automatically populated by the agent using user input or other contextual information.
2. **Enter a Description**
 - Provide a description of what the web page contains or its purpose.
 - This helps the agent determine when to use this tool based on the nature of the user's question.
3. **Click Save to finalise the configuration.**
 - Once configured, the agent will be able to dynamically construct the URL, fetch the web page content, and use that information to inform its response.

Limitations. Some web pages are not designed to be scrapped and others rely of execution of javascript to generate content. In these cases it may be more appropriate to use Call API to get content.