

## MCP Server Quickstart Guide

### How to enable and use the MCP server.

Document generated 09/11/2025

[PDF VERSION](#)

Tags | [AI](#) | [LLM](#) |

Applies to: Free Basic Business Enterprise

This documentation is for **Stack Overflow for Teams Enterprise**. Free, Basic, and Business users can access their documentation [here](#). [Find your plan](#).

## Overview

The Model Context Protocol (MCP) standard offers an efficient, flexible way for AI agents and assistants to access data sources of various types. With an MCP server on your Stack Overflow for Teams Enterprise (SOE) site, you can give an AI agent access to trusted content in real time. This results in accurate, context-aware AI support with responses grounded in enterprise knowledge.

The Stack Overflow for Teams MCP server makes your enterprise knowledge available to any MCP-compatible AI agent. It provides a structured interface for secure read and write access to your team's content. By adding an MCP server, you enable AI coding agents, AI assistants, and custom agents to:

- Read trusted, high-quality enterprise knowledge to improve their outputs.
- Draft, create, and edit questions, answers, and articles from within their own interface, seamlessly storing enterprise knowledge back to your Stack Overflow for Teams site.

The MCP server is built with enterprise-grade governance, attribution, and usage logging, ensuring secure and scalable adoption across your organization.

**NOTE:** The MCP server for SOE is currently in closed beta. If you're an SOE customer, please contact your sales team (or [reach out to support](#)) for access. Write functionality is in limited rollout and may require enabling by your administrator.

## Enable the MCP server

To enable their site's MCP server, SOE site admins should follow these steps:

1. Go to [https://\[your\\_site\].stackenterprise.co/enterprise/mcp](https://[your_site].stackenterprise.co/enterprise/mcp).

## MCP Server

Enable the MCP server to allow enterprise AI agents to access your Stack Overflow for Teams content through the Model Context Protocol. This will make your knowledge base available to supported agents.

### Enable MCP Server

Allow users to authenticate with the MCP Server.

On Off

2. Set **Enable MCP Server** to **On**.

When you enable the MCP server, your site automatically registers a new "Stack Overflow MCP Server" API service application. You can see this new service application by clicking **Admin settings**, then **API**.

## API

Manage API applications and Swagger integration.

### API applications

Control all API applications across the site.

Register new service application







Showing 38 of 38 applications

Read-only 17

Read-write 21

All 38

All applications 38

Name ▲	Owner ▼	Registered on ▼	Status ▼	
<b>Stack Overflow MCP Server</b> Client Id: 88	Community	Aug 22 at 16:20	Read-write	Show details  
<b>API Application</b> Client Id: 33	F. Cummings	Nov 28, 2023 at 17:05	Read-only	Show details  
<b>Backstage</b>	Community	Mar 7 at 15:38	Read-write	Show details  

## Access the MCP server

After you've enabled the MCP server, developers can add the MCP server to any MCP-compatible tool including:

- AI Assistants (ChatGPT, Claude Code, Gemini)
- AI Coding Assistants (Cursor, GitHub Copilot via VS Code, Windsurf, JetBrains AI Assistant, Junie, Devin)
- Custom internal agents or chatbots

**NOTE:** All MCP clients or agents must support the 3/26/2025 MCP spec.

If you're building a custom agent, [reach out to support](#) for implementation details.

## Remote MCP location

You can access one-click add-ons for Cursor and VS Code at [https://\[your\\_site\].stackenterprise.co/mcp](https://[your_site].stackenterprise.co/mcp).

You can also connect to the software of your choice using the following JSON:

```
{
  "mcpServers": {
    "teams": {
      "url": "https://[your_site].stackenterprise.co/mcp",
      "headers": {}
    }
  }
}
```

**NOTE:** The Teams MCP Server uses streamable HTTP as its transport method. It does not support Server-Sent Events (SSE).

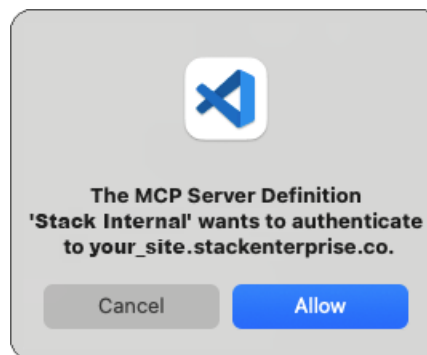
## Authentication

The SOE MCP server uses OAuth 2.0 with PKCE for secure authentication.

- End users authenticate with their Stack Overflow for Teams account. The MCP client (for example: Copilot, Cursor, JetBrains Assistant) will open a browser window for login and token exchange.
- After SOE creates a new API service application for the MCP server, admins manage client registration and API access at [https://\[your\\_site\].stackenterprise.co/enterprise/api](https://[your_site].stackenterprise.co/enterprise/api).
- Tokens are short-lived and scoped to the authenticated user. As the MCP server doesn't support long-lived static tokens, agents must refresh tokens as needed.
- The MCP server enforces security at both the user and tenant level, and logs all requests for governance and attribution.

## Example connection flow (VS Code)

1. MCP client requests authorization, redirects user to SOE login.



2. User signs in with SSO or Stack Overflow for Teams credentials.

Sign-in successful! Returning to Visual Studio Code...

If you're not redirected automatically, [click here](#) or close this page.

3. Authorization server returns a short-lived access token and refresh token.

4. MCP client exchanges the token for access to the MCP server endpoint at [https://\[your\\_site\].stackexchange.co/mcp](https://[your_site].stackexchange.co/mcp).

**NOTE:** If you're building a custom internal agent, you'll need to implement the standard OAuth 2.0 with PKCE flow. Your agent will use the MCP server URL AT [https://\[your\\_site\].stackexchange.co/mcp](https://[your_site].stackexchange.co/mcp). If you require different authentication methods, reach out to your contact at Stack Overflow.

## Prompts

Prompts provide pre-built instructions to guide agents on how to use the MCP server. Prompts provide instructions for how to use the tools and resources available.

PROMPT	DESCRIPTION
<code>search</code>	Query SOE for existing questions, answers, or articles.
<code>create_question</code>	Draft a question for the user to review.
<code>create_article</code>	Draft a knowledge article for the user to review.
<code>create_QA</code>	Draft a question and answer pair for the user to review.

In addition to using natural language, you can invoke prompts by preceding them with the '/' character.

## Prompt examples

- `How do I configure PKCE authentication?` (invokes search prompt)
- `/create_question "Error when connecting to remote MCP server"`

## Tools

The MCP server exposes the following tools to MCP clients. Clients can invoke tools through normal conversations with the agent, or through '#' commands.

TOOL	DESCRIPTION
<code>search</code>	Search for questions, answers, and articles.
<code>get_article</code>	Retrieve a specific article.
<code>get_comments_by_article</code>	Retrieve comments associated with an article.
<code>get_answers_by_question</code>	Retrieve answers for a question.

TOOL	DESCRIPTION
<code>get_comments_by_question_answer</code>	Retrieve comments on a specific answer.
<code>get_comments_by_question</code>	Retrieve comments on a specific question.
<code>get_question</code>	Retrieve a specific question.
<code>create_article</code>	Create a new article.
<code>create_question</code>	Create a new question.
<code>create_qa</code>	Create a question and answer pair.

### Tool examples

- `How do I configure PKCE authentication?` (invokes search tool)
- `#get_question 3172`

### MCP resources (advanced)

Resources are the “data” available to the client. Below is a comprehensive list of resources available.

NAME	RESOURCE URI FORMAT	DESCRIPTION
Question	<code>questions://{questionId}</code>	Retrieve an individual question.
Answers List	<code>questions://{questionId}/answers</code>	Retrieve all answers for a question.
Answer Detail	<code>questions://{questionId}/answers/{answerId}</code>	Retrieve a specific answer.
Article	<code>articles://{articleId}</code>	Retrieve an individual article.
Comments on Question	<code>questions://{questionId}/comments</code>	Retrieve all comments on a question.
Tags	<code>tags://{tagName}</code>	Retrieve information about a specific tag.

### How it works

To use an analogy of building furniture, you can think of prompts as the directions, tools as your hammer and nails, and resources as your wood.

#### Prompts = Directions

Prompts are natural-language instructions your agent follows (for example: “Find related answers” or “Draft an article”). Prompts are great for everyday users in chat/IDE.

#### Tools = Hammer and Nails

Tools are the specific actions the agent can take (for example: search, create content, retrieve details). Tools are ideal when you're creating repeatable, automated workflows.

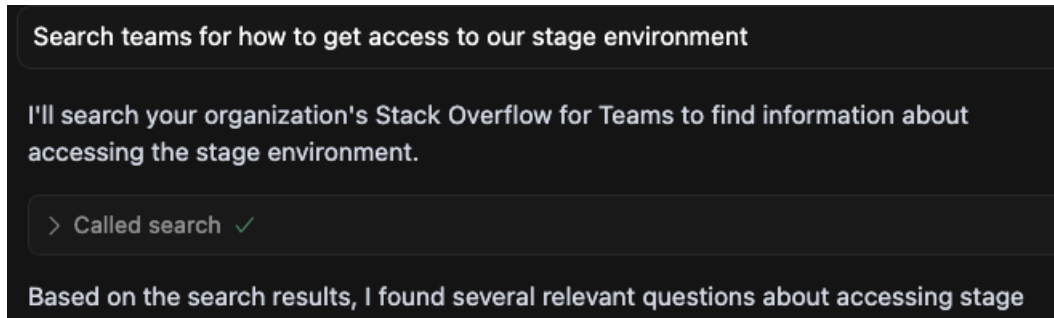
## Resources = Wood

Resources are the materials, the actual pieces of knowledge (for example: questions, answers, articles) the agent retrieves and cites to keep responses trusted and auditable.

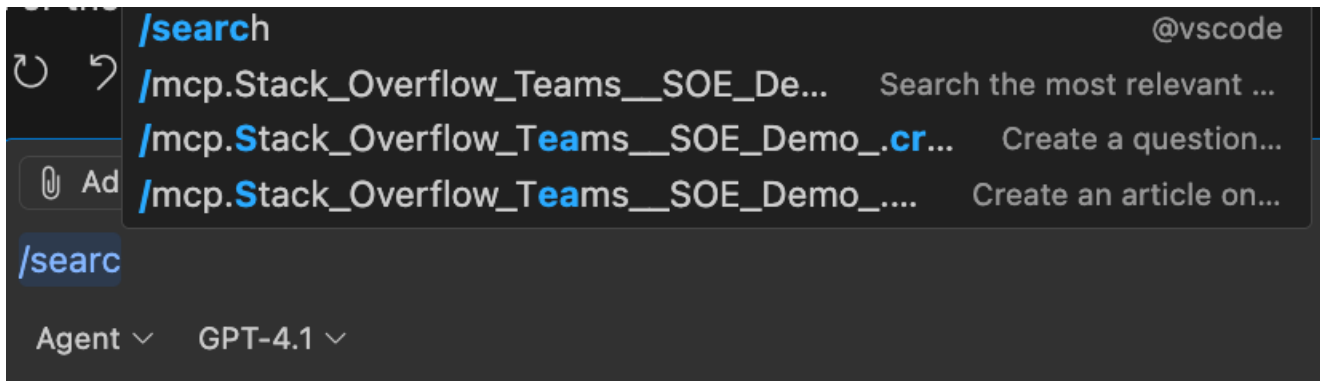
Prompts guide the work, tools do the work, resources prove the work.

## Usage examples

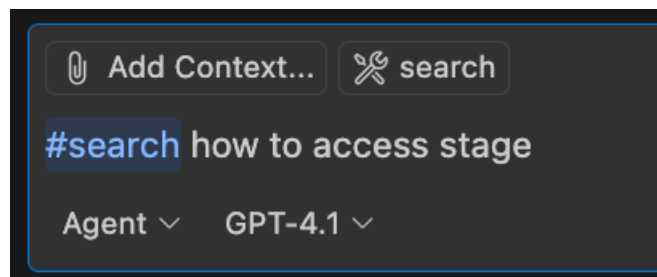
Agents will automatically invoke MCP server functions based on key words you use. The example below shows how the natural language prompt “Search teams...” caused the agent to invoke the search prompt.



You can also invoke prompts by preceding them with the '/' character.



You can explicitly invoke tools by preceding them with the '#' character.



If you need further support or have questions, contact your site administrator.