

## Stack Overflow for Teams API v3

### How to access and update your SOE data with API v3.

Document generated 07/28/2025

[PDF VERSION](#)

Tags | [API](#) |

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 Stack Overflow for Teams API enables you to interact with your Stack Overflow Enterprise (SOE) site programmatically with external scripts, reporting tools and automated workflows. You can read about general API v3 features in the [API v3 Overview](#) article.

---

**THIS ARTICLE APPLIES TO STACK OVERFLOW FOR TEAMS ENTERPRISE ONLY.**

Other Stack Overflow for Teams users should read [this article](#) instead. [Find your plan](#).

## Access API v3

You'll access API v3 within your SOE instance at `https://[your_site]/api/v3`.

In addition to this document, your site has interactive documentation with definitions for all API v3 methods and data models. This interactive documentation system ([Swagger UI](#)) allows you to test API calls and view returned data in a convenient web interface with no API implementation required. See the "Interactive API v3 documentation" section below for more information. You can access the interactive API documentation at `https://[your_site]/api/v3`.

**NOTE:** To avoid having the API block or throttle your requests, make sure each API request includes a `User-Agent` header. Learn more in our [API v3 Overview](#) article.

## Authentication and authorization

The starting point for API v3 authentication is the API application. An [OAuth authentication](#) process uses the API application information to create an API access token. This token then accompanies and authorizes every API v3 request. Access tokens that are read-only can read data only; read-write tokens allow API calls to read, update, and delete data. Only users authenticated into SOE can create API Applications and tokens.

For more information about the explicit OAuth process, read the [Authorization Code Grant \(Explicit OAuth\) with PKCE](#) article.

**NOTE:** The implicit OAuth process is a legacy method previously used for quickly obtaining access tokens in native apps and JavaScript server-side apps. We do NOT recommend using this method due to security risks. For more information, visit [OAuth.net](https://oauth.net).

## Manage API v3 applications and service keys

API v3 supports two types of API applications: user API applications and service keys. A user API application is owned by the user that created it. Any access token created with that API application ceases to work if the user is deleted or deactivated.

A service key is a special type of API application created by an SOE administrator and owned by the "Community" user. Unlike user-specific API applications, service keys are persistent and will never be deactivated along with any user account. Only admins can create and access service keys.

This table lists the key differences between service keys and regular user API applications.

	API Application	Service Key
Created by	Any user (including admins)	Admins only
Owned by	Specific user	"Community" user
Accessible by	User who created it	Any admin
Best for	Single-user applications	3rd-party integrations Persistent access
Deactivated on user deletion	Yes	No

Site admins can create both service keys and their own API applications, and should consider the following factors when deciding which to use. Access tokens generated by a regular API application are dependent on that API application. If the admin deletes the API application, or if the admin account is deleted or deactivated, the tokens will become invalid. This could interrupt API integrations, potentially compromising site performance or user access.

A service key is not tied to (or affected by) the admin's user account, and provides a persistent and stable access point for API integrations. Administrators should use service keys when enabling an integration that's independent of any individual user account. If a third-party integration generates dynamic tokens or requires persistent access to the SOE site, for example, a service key is the appropriate choice. Using an API application instead would create an unwanted dependency on a specific user account. If the user is deleted or deactivated, the integration will lose API access and stop functioning.

To manage all API applications and service keys from the admin API settings page, click **Admin Settings** then **API** under the "INTEGRATION AND APPS" heading. This page shows a table of existing API applications for specific users, as well as service keys (owned by the "Community" user).

# API

Manage API applications and Swagger integration.

## API applications

Control all API applications across the site.

Create new service key



Showing 3 of 3 applications

Read-only 2

Read-write 1

All 3

All applications 3


Name ▾	Owner ⇅	Created On ⇅	Status ⇅	
<b>Swagger UI</b> Client Id: 2	Community	Sep 15 at 11:57	Read-write	Show details  
<b>Service key 3</b> Client Id: 3	Community	2 hours ago	Read-only	Show details  
<b>MablKey</b> Client Id: 1	Smitty	Jul 13 at 19:23	Read-only	Show details  

Interactive API v3 documentation

Click on any column heading to sort the table by that column. To change which applications appear in the list, select **All applications**, **Service keys**, or **User applications** from the upper-right pull-down men.

## Service keys

Only admins can create API v3 service keys. To create a new service key, click **Create new service key**. The Community user page will appear.



# Community

Moderator

Member for 5 years, 9 months  
on the server farm

Mod Dashboard Account info

Profile Activity

PERSONAL INFORMATION

Edit profile

COMMUNICATION SETTINGS

Edit email settings  
Tag watching & ignoring  
Community digests

SITE SETTINGS

Preferences

ACCESS

Your logins

APPS & INTEGRATIONS

API applications

Microsoft Teams integrations

## API applications

### Create an API Application (service key)

You can create an API application for use with the Stack Overflow for Teams API. Users will be able to log in to this application using the OAuth flow. This will also create a service key.

[View API v2 documentation](#)  
[View API v3 documentation](#)


**API application name**

**Domain** Optional

The permitted domain for access tokens and codes to be sent to as part of the OAuth flow. A domain is required for use with API v3.

Create API application

### Active API applications

Name	Created on	Domain	Status	
Service key 4 Client Id 3	3 hours ago	site.stackenterprise.co	Read-only	<a href="#">Show details</a> 
Swapper III				

Create the new service key with this process:

1. Enter a distinct **API application name** that helps you distinguish and organize your API applications.
2. Enter the permitted **Domain**. This should be the domain of the external application you intend to connect with SOE, if you don't intend to connect an external application just use your SOE site as a Domain.
3. Click **Create API application** to create the new API application.

Once done, the new service key will appear in the "Active API applications" list for the Community user.

## API application secrets

When you create an API application and provide an optional domain, SOE generates a secret as well as a key. You'll use the secret to generate access tokens in some OAuth processes. SOE shows the API application's secret only once on initial creation, so be sure to copy it for future use. Later, the secret appears as a string of asterisks for security.

## Application Created

**Client Id:** 212

**Name:** Service Key 123


**Domain:** site123.stackenterprise.com

**Secret:** cNBW ... 9sL1jRoA(( 

**Key:** rl\_V4CD ... 4Y5T3iW 

You'll see each API application in the Active API applications list. Click **Show details** to see any application's key and (obscured) secret.




### Active API applications

Name	Created on	Domain	Status	
<b>API Application</b> Client Id 64	Feb 15	support-test-teams.stackenterprise.co	Read-only ▾	<a href="#">Hide details</a> 
<div><b>Key</b> sw_9dlks ... 8pCbE</div> <div><b>Secret</b> ***** <a href="#">Rotate Secret</a></div>				

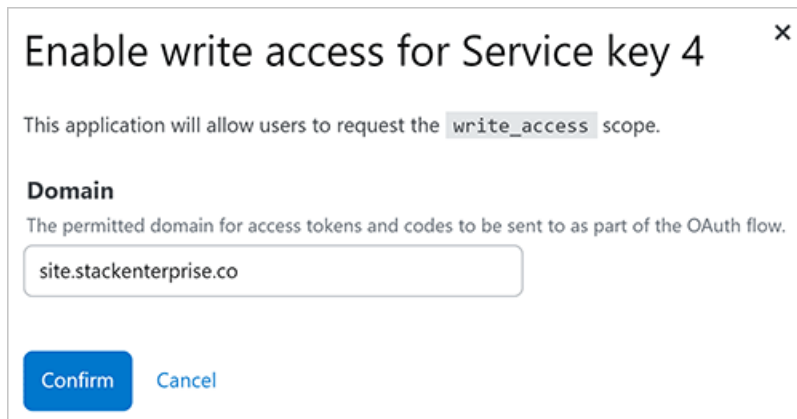
If you created an API application but didn't copy its secret, there's no way to access that secret again. Instead, click **Rotate secret** to generate a new secret you can copy and save. If you generate a new secret, you'll need to use it for all future access tokens for that API application. Access tokens created with earlier secrets will also continue to work.

## API application permissions

All new API applications, including service keys, start with read-only permissions. To change the permissions of a service key, click its select menu in the **Status** column.

Name	Created on	Domain	Status	
<b>Service key 4</b> Client Id 3	1 min ago	site.stackenterprise.co	Read-only ▾	<a href="#">Show details</a> 
<b>Swagger UI</b> Client Id 2	Sep 15	spm1.dev.stackenterprise.co	Read-only Read-write Read-write ▾	<a href="#">Show details</a> 
<b>MablKey</b> Client Id 1	Jul 13	<a href="#">Add +</a>	Read-only ▾	<a href="#">Show details</a> 

When changing a service key's status to Read-write, SOE will prompt you for a permitted **Domain**. Enter (or verify) the domain of the application that will be accessing the API and click **Confirm**.



Enable write access for Service key 4

This application will allow users to request the `write_access` scope.

**Domain**

The permitted domain for access tokens and codes to be sent to as part of the OAuth flow.

site.stackenterprise.co

Confirm Cancel

To change a read-write service key to read-only status, click its **Status** select menu and select **Read-only**. Click **Disable**. To remove a service key entirely, click its **trash can** (delete) button.

## User API applications

Unlike service keys, user API applications are owned by the user that created them. If the user is removed from the site by deactivation or deletion, their API applications and access tokens will no longer work.

An API application's access token has access to the same data as the user that created it. An access token with read-write permissions, for example, can update the same content (questions, answers, comments) as the user.

To access the user API application management page, click **your avatar** (profile picture) at the top of the page. Click **Settings**, then **API applications** (under the "APPS & INTEGRATIONS" heading). The API application page has two parts: an upper section to create new API applications, and a table of existing API applications below.

## API applications

### Create an API Application

You can create a personal API application for use with the Stack Overflow for Teams API.

[View API v2 documentation](#)

[View API v3 documentation](#)





#### API application name

#### Domain Optional

The permitted domain for access tokens and codes to be sent to as part of the OAuth flow. A domain is required for use with API v3.

Create API application

## Active API applications

Name	Created on	Domain	Status	
<b>API v3 reporting</b> Client Id 6	2 days ago	site.stackenterprise.co	Read-only 	<a href="#">Show details</a> 
<b>My first API app</b> Client Id 5	13 days ago	<a href="#">Add +</a>	Read-only 	<a href="#">Show details</a> 

To create a new user API application, enter an **API application name** and your site's **Domain**. Click **Create API application**. Your new API application will appear in the "Active API applications" table.

To change the permissions of an API application, click the select menu in that API application's "Status" column. When you enable write access, you'll be prompted to enter or confirm the domain of the application that will be accessing the API. To delete an API application, click its **trash can** (delete) button. The API will then reject any future calls that use the deleted API application's access token(s).

The admin settings API application page is similar to the user API applications page, but with additional controls. The admin page allows control of service keys as well as all user API applications. Admins can delete user API applications, change their read-only/read-write scope, and update domains.

## Write API access

If you authorise an API access token with write access, you can use it to perform write operations. This includes submitting and editing questions and answers, as well as voting and editing. The site credits these actions to the user that authorised the access token, so don't share your write-enabled access token with other users.

## Interactive API v3 documentation


Admins can enable or disable the interactive API v3 documentation (Swagger UI) by scrolling to the bottom of the API applications page and clicking **On** or **Off**. When on, SOE creates a dedicated service key for Swagger UI. If you disable Swagger UI, SOE will delete the Swagger UI service key.

### Interactive API v3 documentation

**Swagger UI**

OnOff

Enable Swagger UI to allow your users to access your site's data through an interactive API documentation interface. This will generate a new API application. [Learn more](#) about API v3.

 Set the Swagger UI API application **Status** to **Read-write** to allow the interface to both read and alter your site's data.

Go to Swagger UI

---

### API V3 Endpoints via Swagger

Download `swagger.json` to view the API endpoints.

Download

To access the interactive documentation, click **Go to Swagger UI** (or navigate to [https://\[your\\_site\]/api/v3](https://[your_site]/api/v3)). To download a JSON file of all API v3 endpoints and models, click **Download**.

## Swagger UI access tokens

The Swagger UI interface functions as a service key with full access to API v3 as the authenticated user. When authorized, Swagger UI uses its service key to create an access token. This token expires after 24 hours.

**NOTE:** You don't have to wait for the 24-hour period to end to create a new Swagger UI access token. To delete the existing token and create a new one, click **Authorize**, then **Logout**, then **Authorize** again.

To authorize Swagger UI and create a new access token:

1. Go to [https://\[your\\_site\]/api/v3](https://[your_site]/api/v3).
2. Click **Authorize**. An OAuth explicit flow (with PKCE) prompt will appear.
3. Leave **client\_id** unchanged.
4. Leave **client\_secret** blank.
5. Check **write\_access** if the token should have write permissions.
6. Check **access\_team** if the token is for a private team and not the main site. Choose the private team and click **Continue**.
7. Click **Approve**.



## Available authorizations



Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

### oauth2 (OAuth2, authorizationCode with PKCE)

Authorization URL: `https://support-test-teams.stackenterprise.co/oauth`

Token URL: `https://support-test-teams.stackenterprise.co/oauth/access_token/json`

Flow: `authorizationCode` with `PKCE`

client\_id:

client\_secret:

Scopes: [select all](#) [select none](#)

☐ `write_access`  
Allow write access

☐ `access_team`  
Access a team

Authorize

Close

When you use Swagger UI to access API v3, Swagger UI reveals the access token as part of the underlying API call. You can find the access token in the "Curl" section of the Swagger UI execution output. Look for the following line:

```
-H 'Authorization: Bearer [access_token]'
```

You are free to use this token outside of the Swagger UI, but only temporarily. The token will expire and become unusable 24 hours after creation.

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