Team Membership Sync

How to set up Stack Overflow Enterprise to automate team membership.

Document generated 12/06/2024

PDF VERSION

Tags | **Private teams** | **API** | **automation** |

Applies to: Free Basic Business Enterprise

ADMIN PRIVILEGES REQUIRED

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

Overview

The introduction of Private Teams in Stack Overflow for Teams Enterprise (SOE) allows site administrators to restrict the sharing of sensitive information to only those who should have access. To streamline and automate management of Private Team rosters, SOE offers team membership sync with SOE's API v2.

NOTE: *These instructions are for Stack Overflow API v2, not API v3. API v3 does not support team membership sync.*

Syncing team membership via API push

To enable API team membership sync:

1. Click **Admin settings** in the left-hand menu, then **Teams sync**.
2. Click Push **data via API**.
3. Click **Use API push**.

With team membership sync enabled, your site will update Private Teams membership each time you push a properly-formatted JSON file to https:[your_site]/api/2.3/enterprise/usersync. This is your site's user sync API endpoint.

NOTE: *This special endpoint is not documented on your site at https://[your_site]/api/docs or in the API v2 support article.*

Team membership sync authorization

To use the API to sync team memberships, you need an OAuth access token *owned by a site admininstrator*. The API team sync endpoint will reject access tokens owned by regular users, or community service keys. Learn more about acquiring an OAuth token at https://[your_site]/api/docs/authentication.

To call the user sync API, make a form-type POST request (encoded "application/x-www-form-urlencoded") to your site's usersync URL (https:[your_site]/api/2.3/enterprise/usersync). Use the following form fields.

Form field	Definition
`access_token`	The OAuth **access token** (see "Write-enabled API" in the API Docs)
`key`	The **key** from your API Access Keys page
`requestsJson`	A JSON array of user sync requests, with one entry for each team. ***NOTE:*** *Encode this field with "percent-encoding".*
`dryRun`	(boolean) If **true**, will do a preview of the sync process. No actual changes to users will be made, but you can see if accounts resolve properly and which changes would be done in a real run. If **false** (or not present), no changes will be made.

Example JSON request

```
[{"Team":"TEAM-NAME","Members":[{"UserIdentifier":"USER-ID-1","Level":"USER-LEVEL-1"},{"UserIdentifier":"
```

Include the following fields in your JSON request:

- **TEAM-NAME** The team to sync membership with
- **USER-ID** The user ID of the user
- **USER-LEVEL** The level (role) of the user in the team ("Member", "Admin", or "Moderator")

NOTE: *The team sync API call is a full update operation of the membership roster (not an append operation).* ***You must include all the members of the team with each API call****. Team sync will remove any existing team members not included in the JSON data file.*

Example HTTP request

```
POST https://stackoverflowenterprise.example.com/api/2.3/enterprise/usersync HTTP/1.1
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 964
Host: stackoverflowenterprise.example.com

access_token=c9GJIziqUtqx0uBr7t4mdw))&key=iyGkn*C5yqRXRzinhFwkdw((&requestsJson=%5b%0d%0a++%7b%0d%0a++++%
```

Example HTTP response

```
{
    "items": [
        {
            "HasErrors": true,
            "Results": [
                {
                    "StatusCode": "TeamNotFound",
                    "Team": "notexisting"
                },
                {
                    "SyncResult": {
                        "IntendedChanges": [],
                        "ActualChanges": [],
                        "Status": "Success",
                        "SiteName": "productone",
                        "Log": "[PushAsync] START User Sync from Push for Team productone (6/17/2019 3:00:22 PM -04:00)\
                    },
                    "StatusCode": "Success",
                    "Team": "productone"
                },
                {
                    "SyncResult": {
                        "IntendedChanges": [
                            {
                                "IsDeactivated": false,
                                "AccountId": 13645930,
                                "Change": "NoChange",
                                "NewUserType": "Admin",
                                "CurrentUserType": "Admin",
                                "SiteUserId": 1
                            },
```

Status codes—overall sync process

The HTTP response will include a status code for each Private Team updated.

Status code	Definition
Success	The entire sync process succeeded
PartialSyncFailure	The sync process started, achieved partial success, then failed
SuccessfulDryRun	The sync process completed as a dry run (preview mode)
OtherError	An unspecified error occurred
UserSyncNotEnabled	The Private Team is not set to be managed by API user sync
FailedToGetMembersFromSource	There was some problem with parsing the JSON
ErrorTryingToResolveUsers	There was a system error when trying to find users for the given identifiers
FailedToDetermineChanges	There was a system error when trying to determine intended changes
ErrorApplyingChanges	There was a system error when trying to make the actual changes
TeamNotFound	No team with the given URL could be found

The basic API response matches other API methods. It will always be a single page with a single item:

```
{
    "items": [{...}],
    "has_more": false,
    "quota_max": 10000,
    "quota_remaining": 9999,
    "page": 1,
    "page_size": 1,
    "total": 1,
    "type": "user_sync_api_response"
}
```

The data item itself is a single element of type `items-array`, a root object with two properties:

- `Results` is an array containing one item per individual team that you requested.
- `HasErrors` is a boolean that indicates if any result has an error. This is a quick way to see if the entire bulk operation/batch succeeded, thus saving having to dig deep into individual results.

```
{
    "HasErrors": true,
    "Results": [...]
}
```

Each result item contains information about a single Private Team:

- `Team` is the URL slug of the team (https://[your_site]/c/[team_name])
- `SyncResult` is information about the actual result of the team sync. See below for a full list.

SyncResult values

SyncResult value	Definition
Status	Same as `StatusCode` on the result object
SiteName	The display name of the Private Team
IntendedChanges and ActualChanges	Shows which changes the sync process determined should be done, and which ones were actually done
Change	May return "NoChange", "AddToSite", "RemoveFromSite", or "ChangeUserType"
CurrentUserType and NewUserType	Returns either "Admin" for team owners, or "Registered" for regular members
AccountId	The AccountId of the user
SiteUserId	The User Id within that team
IsDeactivated	Returns "true" if the account is currently deactivated and will reject login
ActualChanges	Indicates which changes were made (blank on a dry run)

SyncResult value	Definition
Log	A human-readable log file that contains verbose information about the sync operation. Log format may change, and is not meant to be machine-parseable.

NOTE: `SyncResult` *may be absent for some status codes.*

Log example

```
{
    "SyncResult": {
        "IntendedChanges": [
            {
                "IsDeactivated": false,
                "AccountId": 13645930,
                "Change": "NoChange",
                "NewUserType": "Admin",
                "CurrentUserType": "Admin",
                "SiteUserId": 1
            },
            {
                "IsDeactivated": false,
                "AccountId": 13645931,
                "Change": "NoChange",
                "NewUserType": "Registered",
                "CurrentUserType": "Registered",
                "SiteUserId": 2
            }
        ],
        "ActualChanges": [],
        "Status": "Success",
        "SiteName": "internalchat",
        "Log": "[PushAsync] START User Sync from Push for Team internalchat (6/17/2019 3:00:22 PM -04:00)\r\nSwitched Sites
    },
    "StatusCode": "Success",
    "Team": "productone"
}
```