GitHub Provisioning Plugin

The GitHub Provisioning Plugin synchronizes Registry data with GitHub.

Operations

Registry CO Person Transaction	Changelog Action
Add	Add CO Person's group memberships to GitHub team memberships
Edit	Synchronize CO Person's group memberships with GitHub team memberships
Enter Grace Period	No changes (unless attributes change as part of grace period)
Expiration / Becomes Inactive	Remove CO Person's group memberships from corresponding GitHub team memberships
Unexpire / Becomes Active	Synchronize CO Person's group memberships with GitHub team memberships
Delete	Remove CO Person's group memberships from corresponding GitHub team memberships
Manual Provision	Synchronize CO Person's group memberships with GitHub team memberships

Registry CO Group Transaction	Changelog Action
Add	Add CO Group's memberships to corresponding GitHub team
Edit	Synchronize CO Group's memberships with corresponding GitHub team
Delete	Remove CO Group's memberships from corresponding GitHub team
Manual Provision	Write CO Group record (including memberships) to changelog

Configuration

1. This is a non-core plugin, see Installing and Enabling Registry Plugins for more information.
2. In order to get started, COmanage must first be registered as an application with GitHub. In order to manage group/team memberships, this must be done using the account of an owner of the GitHub organization containing the teams to be managed.
    a. Start by creating a new Provisioning Target of type *GithubProvisioner*.
    b. After it is created, the configuration page will appear. This page has the callback URL that you must use when registering with GitHub.
    c. After registering, populate the *GitHub Username*, *GitHub Client ID*, and *GitHub Client Secret*.
3. Select the operations for the provisioner to handle.
    a. If removal of unknown members from GitHub teams is enabled, COmanage will keep the team membership exactly in sync with the corresponding COmanage group. This means Team members added directly via GitHub will (eventually) be removed if they do not correspond to a CO Group member.
    b. However, if removal of unknown members from GitHub teams is *not* enabled, (1) re-provisioning a group or (2) setting a group to Suspended will *not* remove any members from the corresponding GitHub team -- even for members who were formerly part of the CO Group. (Under normal operations, however, CO People *will* be removed from the GitHub team when a group membership is removed.)
4. Upon clicking Save, GitHub may ask to authenticate and/or authorize COmanage.
5. After completing authentication and authorization, COmanage will ask which GitHub organization to manage.
6. COmanage will manage the memberships of GitHub Teams in the selected organization when those Teams have names that exactly match the names of COmanage groups within the CO.
    a. ⚠ In order to be provisioned to a GitHub Team, a CO Person must have an identifier of type *GitHub* (case sensitive) attached to their record, exactly matching the name of the CO Person's GitHub login.
    b. ⚠ If a Team member is not already a member of any Team within the GitHub Organization, GitHub will first send an invitation to the person before they become an active member of the team. (Soon, anyway. CO-942)

⚠ **Request Caching**

In order to reduce the number of queries that count towards GitHub's API rate limit, the GitHub Provisioner uses a file based caching mechanism, with cache files written to `APP/tmp/cache/github-api-cache`. (`APP/tmp` is wherever it was initially set it up to point to.)

This means that if multiple web servers are running COmanage Registry, each will have its own cache, effectively doubling the number of uncached queries. Under typical usage, this should not be a problem, though for heavier usage something like a database cache (CO-945) may be required.

Connecting GitHub Usernames to CO People

There are several ways to attach GitHub usernames to CO Person records. All require the extended identifier type *GitHub* to be created first.

1. The GitHub identifier can be collected as part of an enrollment flow.
2. An administrator can manually add the identifier.

3. Self service is not currently supported (other than as part of an enrollment flow). (CO-943)

See Also

- cm_co_github_provisioner_targets