Socrata was acquired by Tyler Technologies in 2018 and is now the Data and Insights division of Tyler. The platform is still powered by the same software formerly known as Socrata but you will see references to Data & Insights going forward.

Learn more... (https://www.tylertech.com/solutions/transformative-technology/data-insights)

Socrata Permissions API 1.0.0

The permissions API can be used to inspect and update permissions.

In general, the usage pattern for the API is:

- `GET` the permissions JSON object for an asset
- Change the permissions JSON object
- `PUT` the JSON object back

The permissions JSON object

The permissions JSON object includes all of the info about an asset's permissions.

`scope` string

This is the current visibility level of the asset.

Possible `scope` values are:

- `private` scope: This means that the asset can only be seen by its owner, and any users inside of the `users` array.
- `public` scope: This means that the asset can be seen by any user who is able to access the site. **NOTE:** Changing an asset to public may result in the asset going into the approvals queue.
- `site` scope, a.k.a. "internal" scope: This means that all site members are able to see the asset. This is only relevant for Socrata Connected Government Cloud sites.

Root `accessLevels` array

!!! `accessLevels` is *NOT* required when doing a `PUT` !!!

The `accessLevels` array is the list of **available** access levels that can be set for `users` using the permissions API on the current domain.

`accessLevels` is intended mainly for guidance about what permissions can be set.

`accessLevels`

For an SCGC internal site, `accessLevels` will be:

- `{ "name": "viewer", "version": "published" }`
  - a.k.a. "published viewer": users with this access level can only see the published version of an asset and cannot change the asset at all (unless their site role allows them to).
- `{ "name": "owner", "version": "all" }`
  - a.k.a. "contributor" (in SCGC) and "co-owner" (non-SCGC): users with this access level can do *anything* to the asset except delete it and transfer ownership of it.

For an Open Data site, `accessLevels` will be:

- `{ "name": "viewer", "version": "all" }`
  - a.k.a. "legacy viewer": users with this access level can see both the published *and* the draft versions of an asset
- `{ "name": "contributor", "version": "all" }`
  - a.k.a. "legacy contributor": users with this access level can see both the published *and* the draft versions of an asset; they can also add/remove data but cannot change schema.
- `{ "name": "owner", "version": "all" }`
  - a.k.a. "contributor" (in SCGC) and "co-owner" (non-SCGC): users with this access level can do *anything* to the asset except delete it and transfer ownership of it.

`users` array

This array lists every user/team that has been explicitly given access to the asset.

Identifying a user/team

There are two ways of granting access to users, by `id` and by `email`.

If a user with the given `id` does not exist, you will get back a 404 error. The `email` key takes precedence over the `id` key.

Teams are allowed in the `users` array as well but *only* have an `id` and *never* have an `email`

Email shares to users who don't have accounts yet

It is possible, in some situations, to share to a user via their email address even if they don't have an account. These users will come back from the API **without** an `id` field but with an `email` field. This scenario is only possible for Open Data sites that are open to community users.

`accessLevels` array

While this is an array, **currently only 1 access level per user is supported**.

The object in this array *must* match one of the `accessLevels` that was returned in the root `accessLevels` array, or can be the `current_owner` access level.

In the `users` array, there **must** be **one and only one** user who has an `accessLevel` with the `name` of `current_owner` and the `version` of `all`. That user **can not** be a team. Changing the `id` of the user who has the `current_owner` access level will transfer ownership to the new user. The `current_owner` *must* be a user who has a Socrata account (not an email share to a user who doesn't have an acocunt yet).

`displayName`, `type`, and other fields

!!! Any fields other than `id`, `email`, and `accessLevels` are solely for informational purposes and are ignored by the API when doing a `PUT` **!!!**

(cc) BY-NC-SA (http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US) Licensed by Tyler Technologies (https://www.tylertech.com/solutions/transformative-technology/data-insights) under CC BY-NC-SA 3.0 (http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US). Learn how you can contribute! (/contributing)