Grouper local entities

Description

Grouper users sometimes need to create and manage entities in Grouper that are not part of a central subject source. An example is a small set of service accounts that need access to Grouper, when a configured subject source for such accounts isn't an option. Another example is when Grouper integrates with an external database that has schemas needed for access management. These schemas must be represented in Grouper so they can be assigned to Groups/Roles/Permissions. A "local entity" can be created in the folder structure. Local entities are not intended to be used to represent people; those should be in your subject source.

A local entity is modeled in Grouper as a specialized type of group object. Like a group, it exists in a folder, has a uuid as the subject id, and full path as the subject identifier0. However, you cannot add members to it, and you can only assign admin, view, attribute read, and attribute update privileges. It also behaves similar to a normal subject, in that an admin can use GSH to set a password for it for UI and WS access. Alternatively, admins for the entity can assign a WS JWT Key for access, which does not require assistance from a system administrator.

Description	Group	Local entity
subject ID label in UI	UUID	Unique ID
Subject identifier0 in UI	ID path	Name
subject source	g:gsa	grouperEntities
subject type	group	application
icon	multiple users icon	cloud with downward arrow
assignable privileges	admin, update, read, view, optin, optout, attribute read, attribute update	admin, view, attribute read, attribute update

Technical description

A local entity has the following attributes, comparable to groups:

- id - uuid, doesn't change
- extension - system name in the folder, shouldn't change
- display extension - display name in the folder, can change
- description - free form text documentation about the entity
- name - fully qualified (including parent folders) system name
- display name - fully qualified (including parent folders) display name

Entities have a subject source different than the Grouper subject source (though similar). The following subject attributes exist in addition to the group subject attributes:

Attribute name	Meaning
entityId	same as name attribute
entityExtension	same as extension attribute
entityIdAttribute	if a secondary identifier has been configured via an attribute (see below, but deprecated), this is the value

(Deprecated) alternate identifier - not working since 2.5

~~If the identifier of the entity is not valid for the extension (e.g. if it could contain a colon, or other invalid character in the grouper extension namespace), then you can put any fully qualified (including folder names) identifier here. Note, no two entities can have the same subjectIdentifier. Also, this attribute is public, meaning anyone can read (if they can VIEW the entity), or update it (if the can ADMIN the entity). Note, this security to be maintained, this assumes a hierarchical security model for folders... i.e. you must trust the owners of parent folders where the entities are stored since they can have a subjectIdentifier with a colon inside. The attribute must start with the folder where the entity is stored. This is autocreated for you, depending on your config, might be here: etc:attribute:entities:entitySubjectIdentifier Assign this to the local entity (e.g. with UI), and give the string value which is the identifier. Note: the assignment to the local entity is done with a "group attribute assignment" not an "entity attribute assignment"~~

Local entity privileges

There are only two privileges for local entities: VIEW and ADMIN.

- VIEW means you can see it, its name, description, etc. With VIEW you could add it to a group or assign permissions to it in a role.
- ADMIN means you can edit it, delete it, assign attributes to it, etc.

In the grouper.properties you can designate if entities are viewable by all by default. Property entities.create.grant.all.view = false by default for security reasons. If set to true, then the ALL subject will be granted view on new entities. This occurs on local entity create, and can be unassigned.

If you try to assign READ, UPDATE, OPTIN, OPTOUT to a local entity, you will get an error

Local entity auditing, change log, point in time

Entities are audited like groups, but the categories are: entity, and the actions are addEntity, updateEntity, and deleteEntity.

There are three change log types for entities: ENTITY_ADD, ENTITY_UPDATE, ENTITY_DELETE. All other actions will appear under groups. e.g. if you add a privilege to an entity it will appear like a privilege is added to a group.

The point in time information is available, similar to point in time information on groups.

Authentication

Built-in authentication

if grouper.hibernate properties grouper.is.ws.basicAuthn is set, or environment variable GROUPER_WS_GROUPER_AUTH is set, Basic authentication can be used to allow specific entities to log in. The password is set on the local entity using GSH and the GrouperPasswordSave() method. As with other subjects, either the subject id (the entity uuid) or the subject identifier (the entity full path) can be used to authenticate. Note that the identifier will have colons. This normally conflicts with Basic auth's username:password syntax. Grouper handles Basic auth in a non-standard way by splitting on the last colon of the username:password string instead of the first (grouper.properties grouper.authentication.splitBasicAuthOnFirstColon = false by default). You can also escape the colons (in either the username or password) by replacing with `&#x3a;` (grouper.properties, grouper.authentication. basicAuthUnescapeColon is true by default).

JWT private key

Whether or not Basic authentication is set, You can also generate a JWT private key, and authenticate that way.

API

You can create a local entity with the `EntitySave` class:

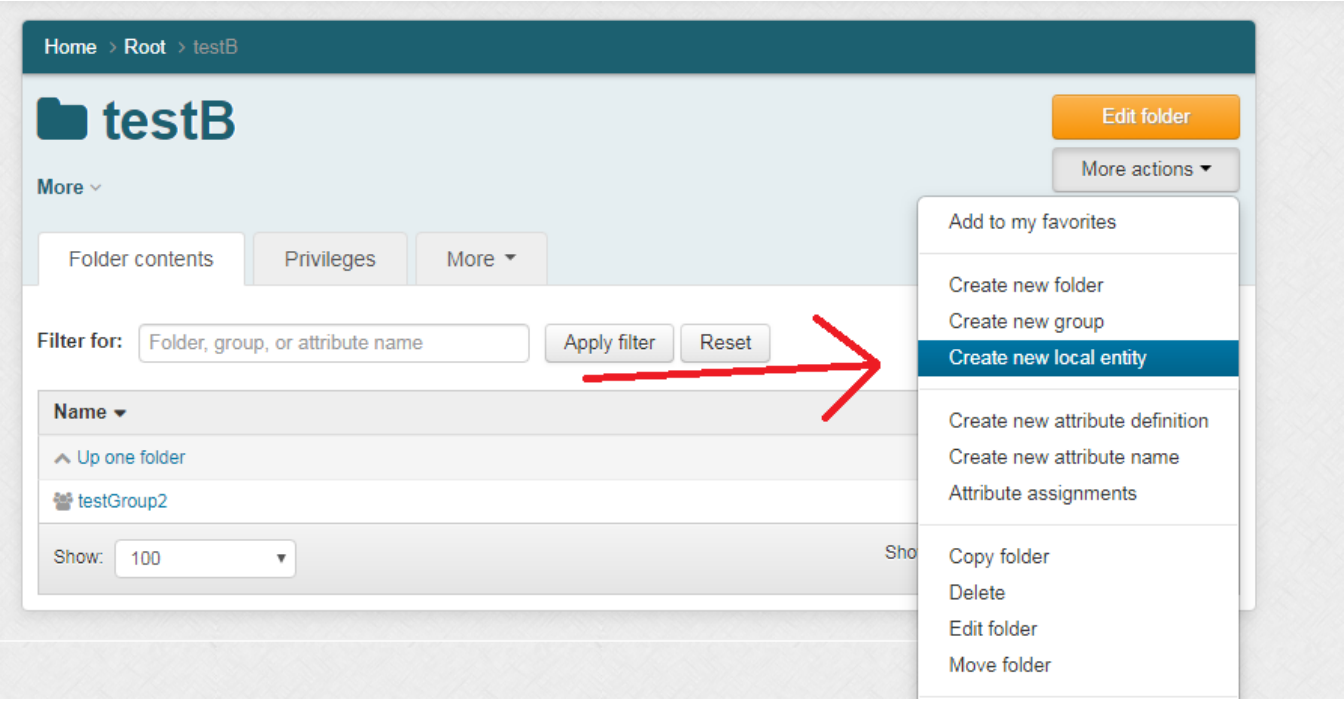
```
Entity testEntity = new EntitySave(grouperSession).assignCreateParentStemsIfNotExist(true) .assignName("test:
testEntity").save();
```

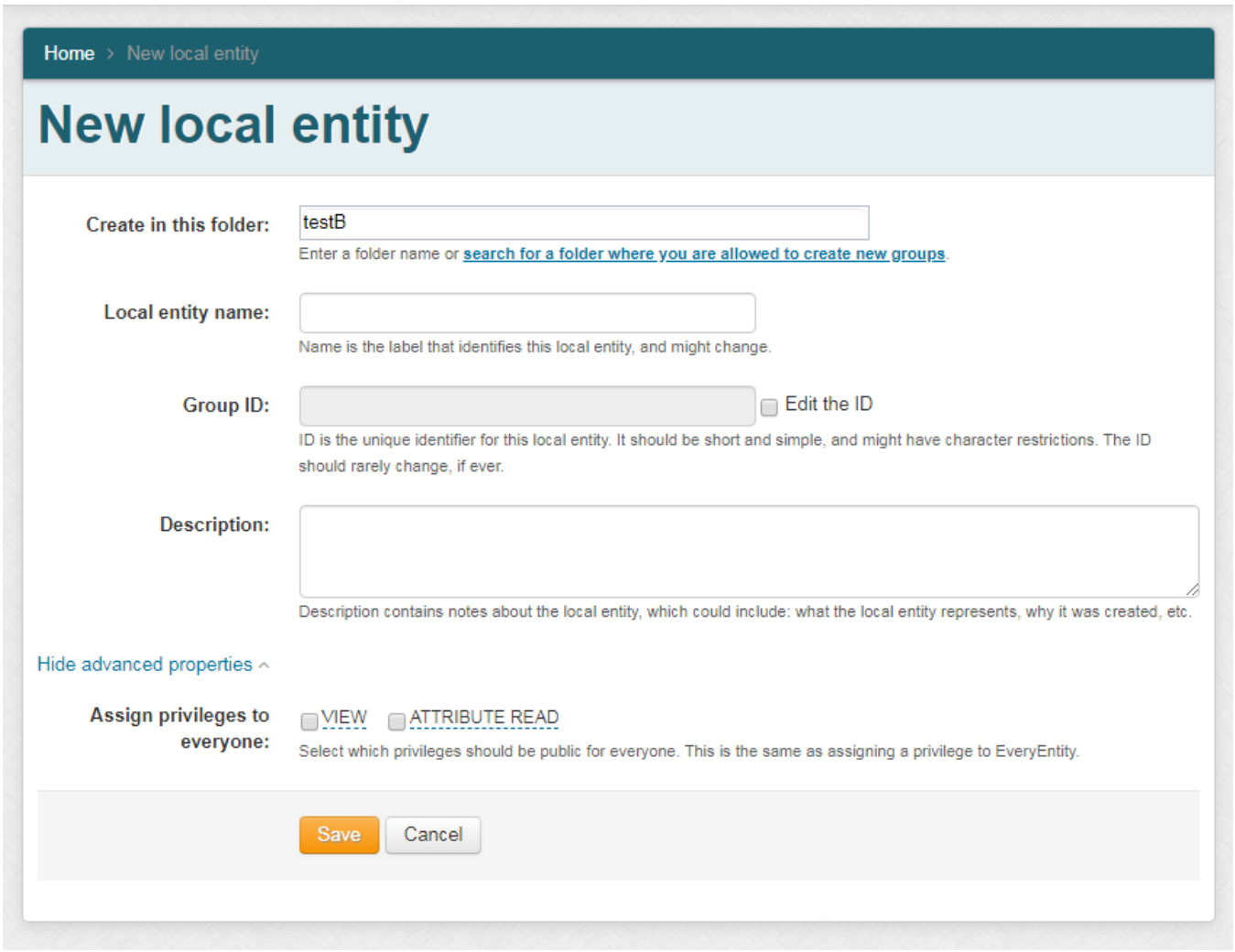
You can find local entities with the `EntityFinder` class (note a grouper session must be open, and the grouper session user must have VIEW or ADMIN on the entity to show the result):

```
Set entities = new EntityFinder().addName("test:testEntity").findEntities();
```

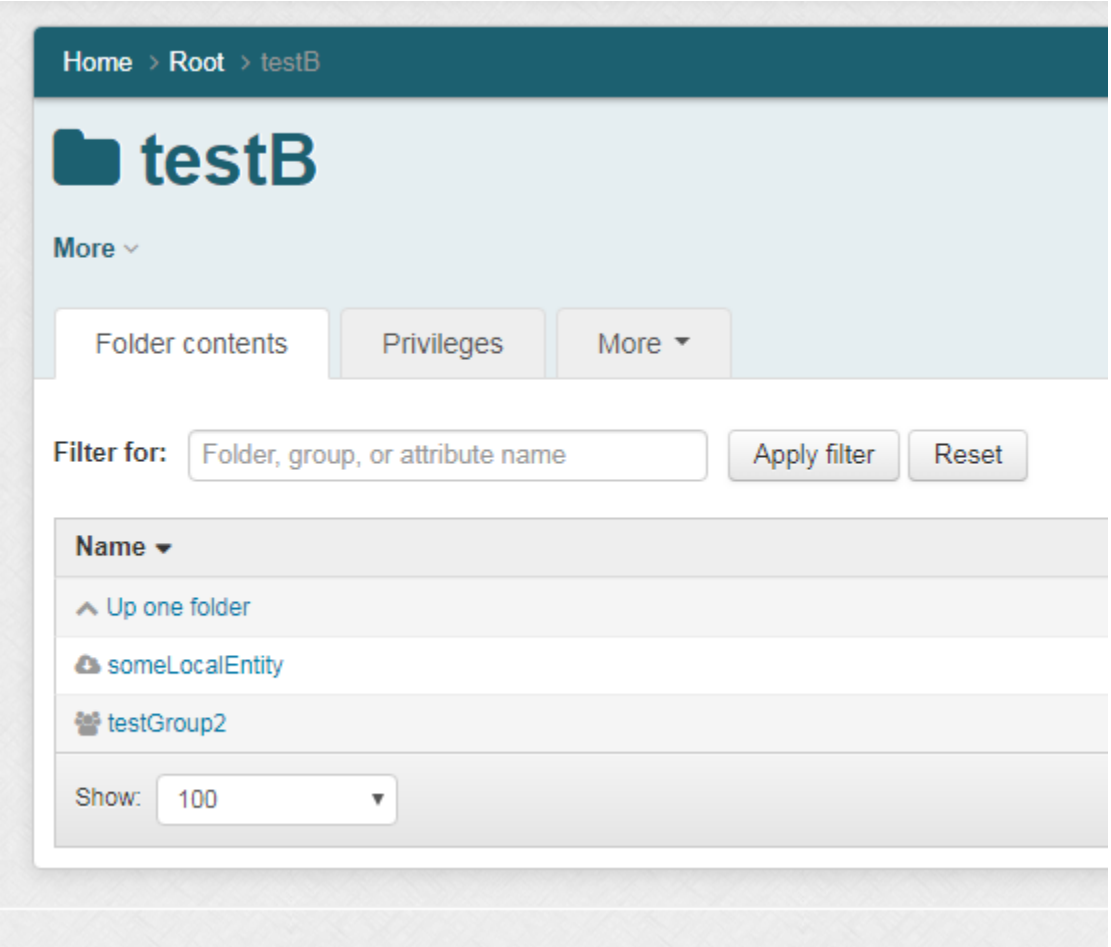
UI

You can create/edit/delete local entities on the UI in a folder you have CREATE on.

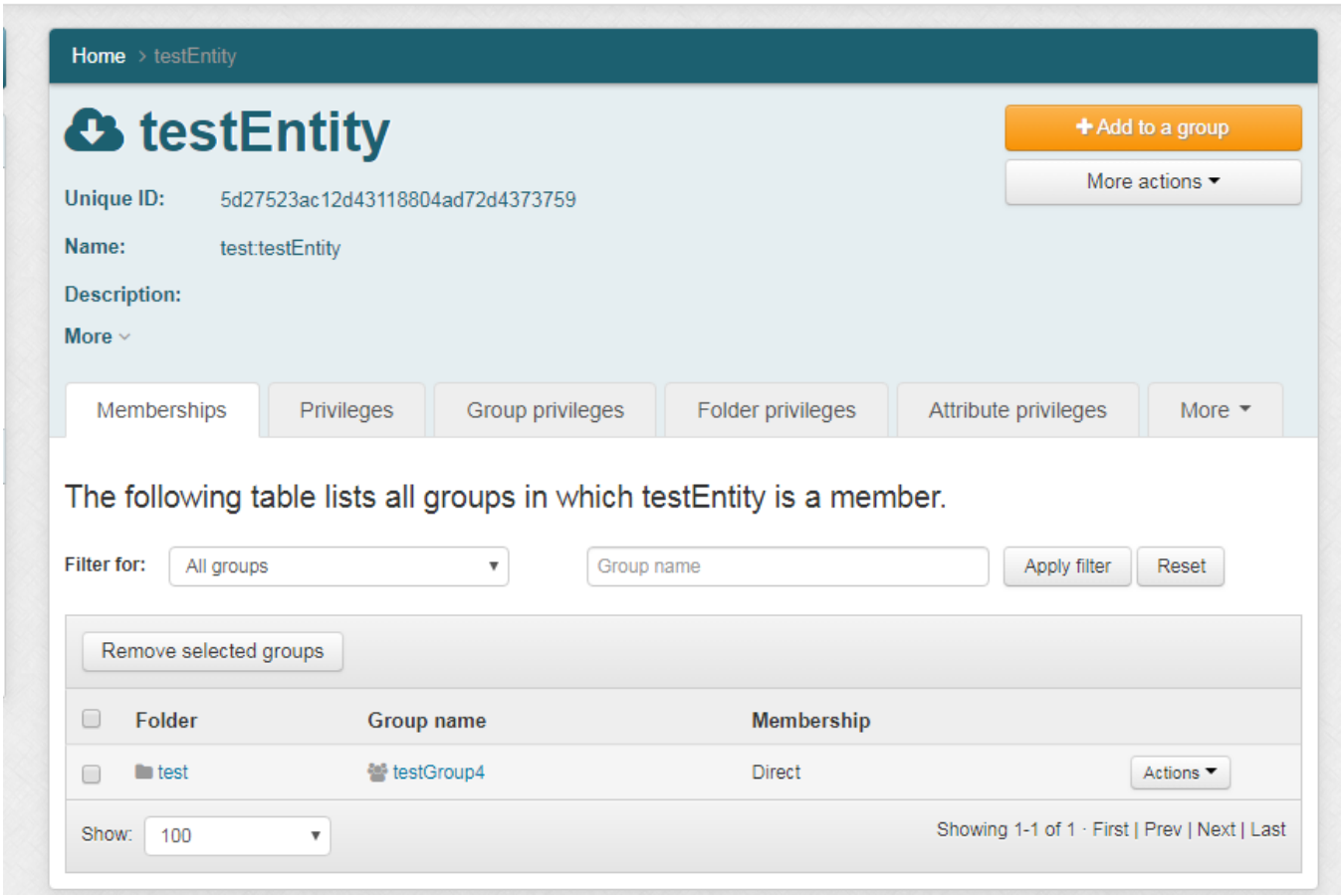

Home › Root › testB
testB
Edit folder
More actions
More
Add to my favorites
Folder contents
Privileges
More
Create new folder
Create new group
Filter for:
Folder, group, or attribute name
Apply filter
Reset
Create new local entity
Create new attribute definition
Name
Create new attribute name
Up one folder
Attribute assignments
testGroup2
Copy folder
Show:
100
Delete
Edit folder
Move folder



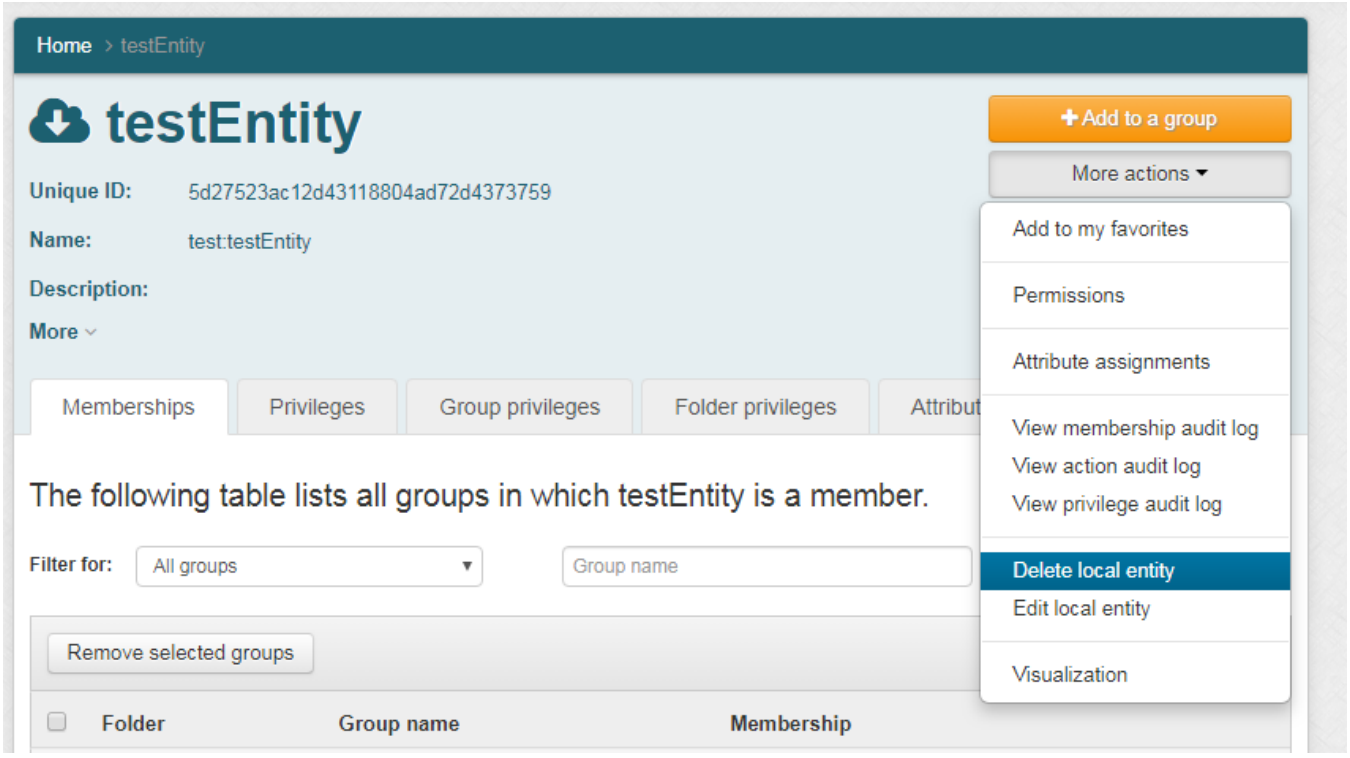
Local entity icon:



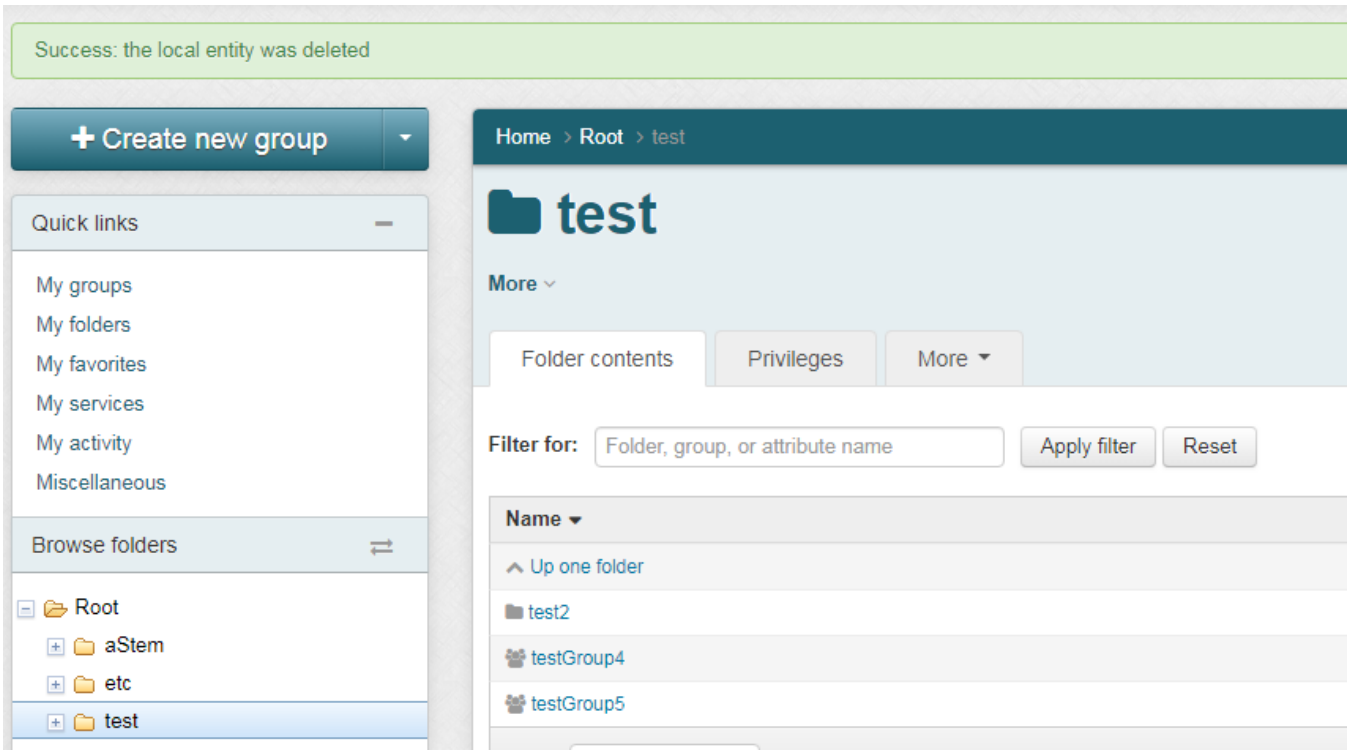
View an entity



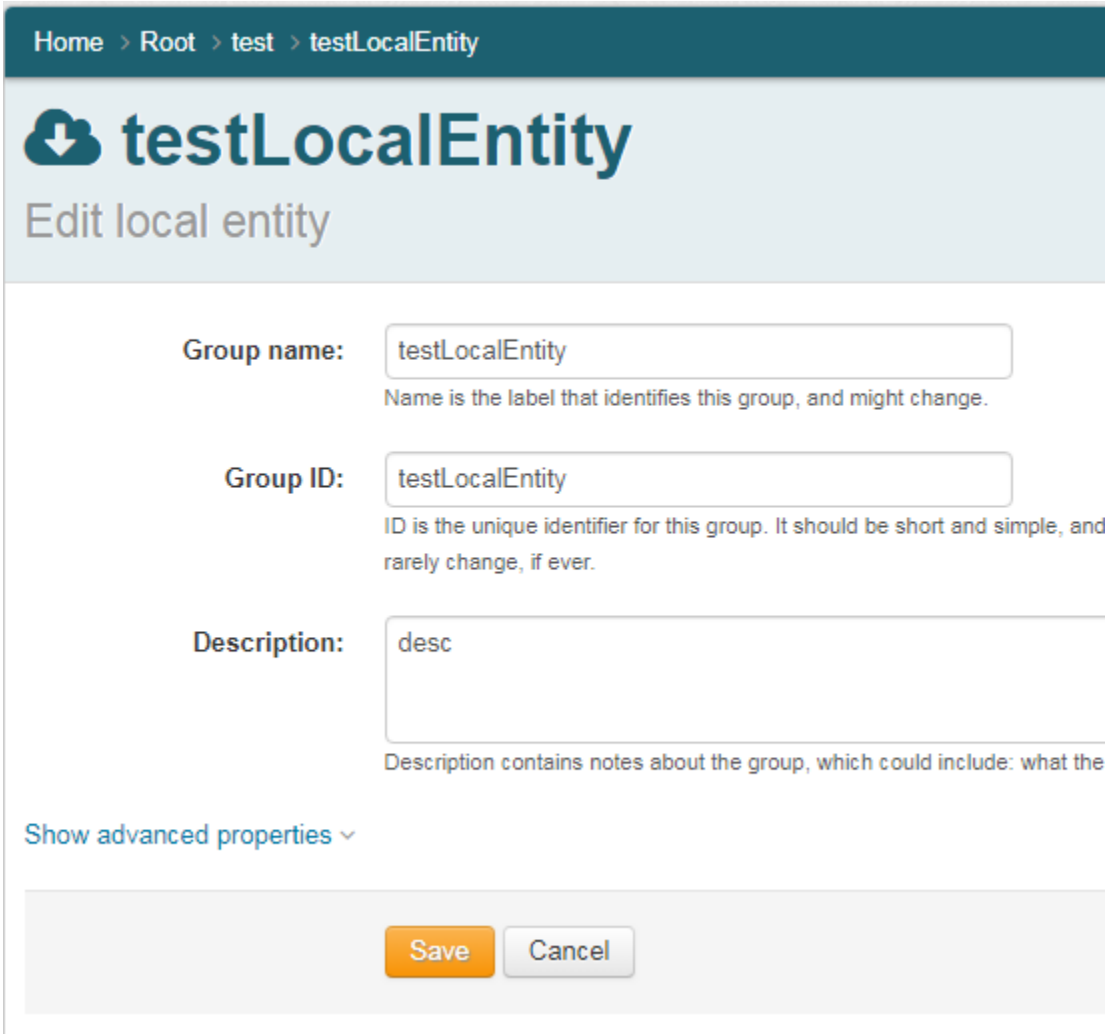
Menu has entity options



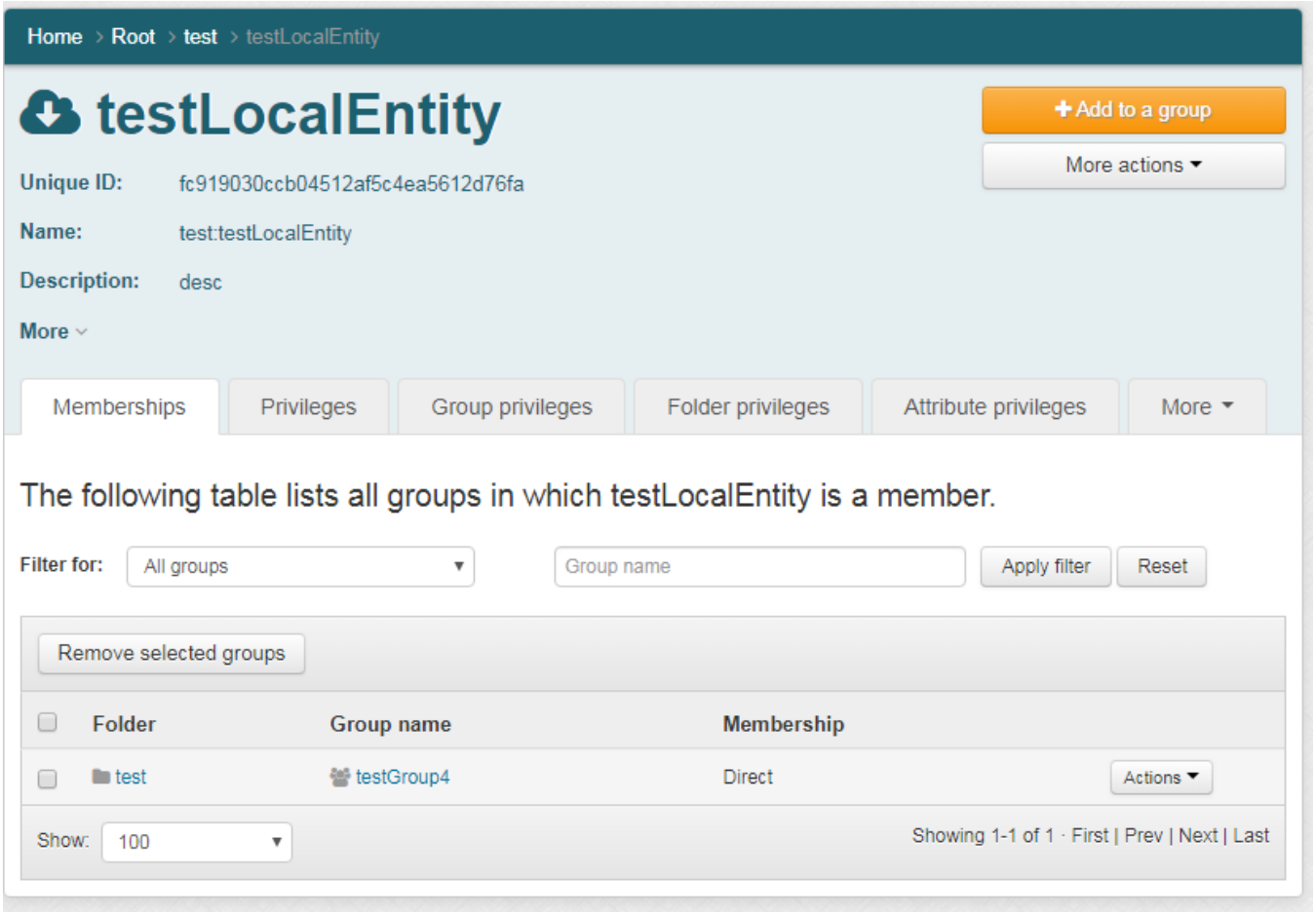
Delete a local entity



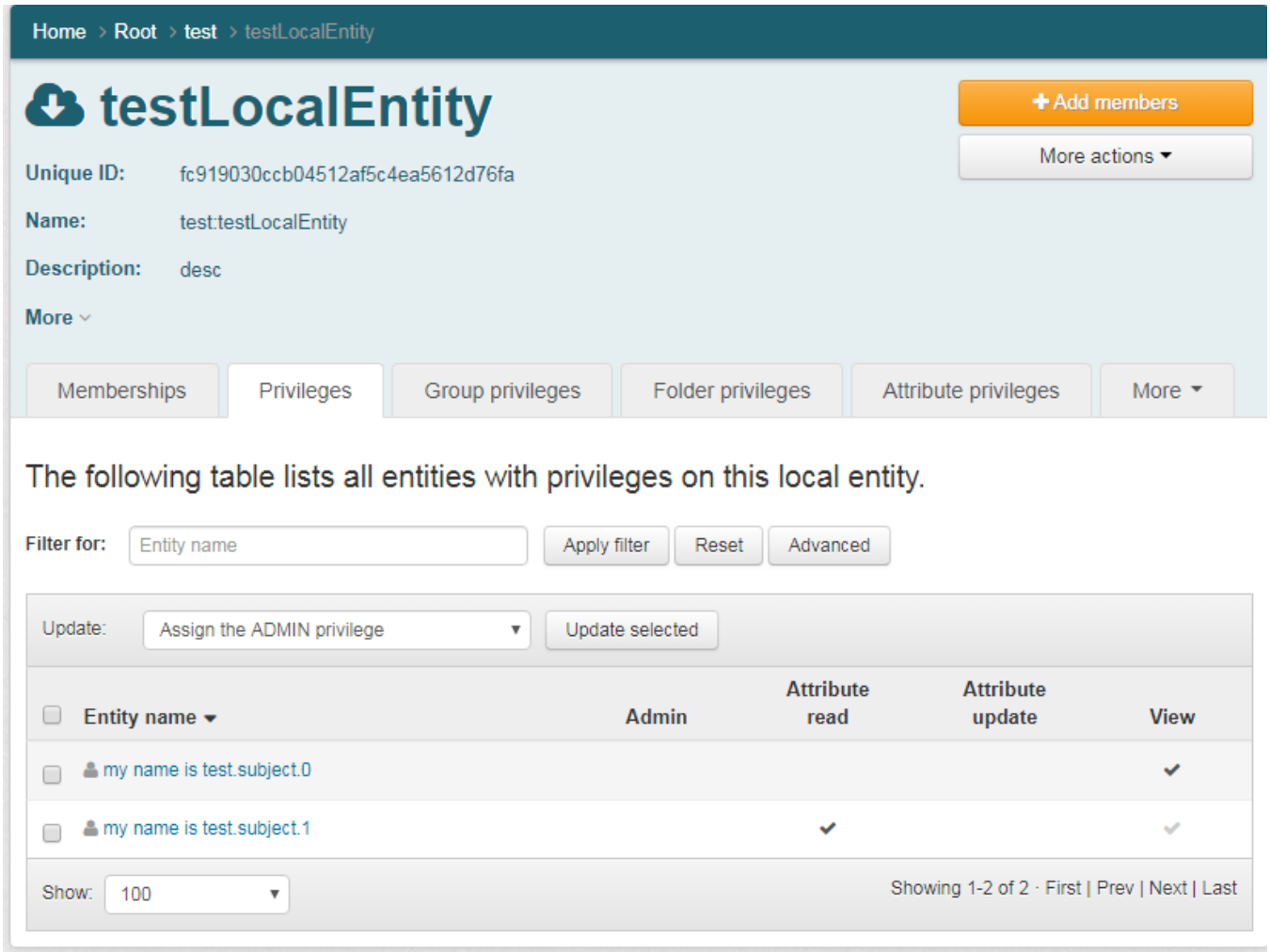
Edit a local entity



There is a privilege tab



Only entity privileges can be assigned



Web services

Note: all web service changes are also available in the Grouper client.

You can create a local entity (or edit, delete), with the GroupSave web service and add typeOfGroup.

```
{
  "WsRestGroupSaveRequest":{
    "wsGroupToSaves":[
      {
        "wsGroup":{
          "description":"desc1",
          "displayExtension":"disp1",
          "name":"aStem:whateverGroup",
          "typeOfGroups":"entity"
        },
        "wsGroupLookup":{
          "groupName":"aStem:whateverGroup"
        }

      }
    ]
  }
}
```

You can also use the FindGroups web service to find entities:

```
{
  "WsRestFindGroupsRequest":{
    "wsQueryFilter":{
      "typeOfGroups":"entity",
      "queryFilterType":"FIND_BY_GROUP_NAME_APPROXIMATE",
      "stemName":"aStem",
      "groupName":"aGr"
    }
  }
}
```
