Stored Procedure Support for **Lists**

This set of stories is derived from the HTTP command/ReSTful endpoint combinations document in the main Lists directory.

- Signin

  1. created by users with valid credentials (returns token on success)

  - `POST /signin`
  - Supported by sql proc `verifyUser()`

  2. Users can sign out (removes token)

  - `DELETE /signin`

- Password

  1. can be changed by logged in users

  - `PUT /password`
  - Supported by sql proc `putUserPassword()`

- Users

  1. are created by administrative users

  - `POST /users`
  - Supported by sql proc `addUser()`

  2. can get information about other users

  - `GET /users`
  - `GET /users/<uId>`
  - Supported by sql proc `getUsersAll()`
  - Supported by sql proc `getUser()`

  3. can update their profile information

  - `PUT /users/<uId>`
  - Supported by sql proc `putUser()`

  4. are deleted by the user themselves or administrative users

  - `DELETE /users/<uId>`
  - Supported by sql proc `delUser()`

  5. can have administrator role assigned/removed

  - `POST /users/<uId>/admin`
  - `DELETE /users/<uId>/admin`
  - Supported by sql proc `putUserAdmin()`

- Lists

  1. created by users, who own them

    - `POST /users/<uId>/lists`
    - Supported by sql proc `addList()`
        - Supported by sql proc `addCollaborator()`

  2. Users can get details of their lists

    - `GET /users/<uId>/lists`
    - `GET /users/<uId>/lists/<lId>`
    - Supported by sql proc `getUserLists()`
    - Supported by sql proc `getUserList()`

  3. Users can update their list attributes

    - `PUT /users/<uId>/lists/<lId>`
    - Supported by sql proc `putUserList()`

  4. deleted by the user that owns them

    - `DELETE /users/<uId>/lists/<lId>`
    - Supported by sql proc `delUserList()`

  5. can have collaborators with access permissions

    - `GET /users/<uId>/lists/<lId>/collaborators`
    - Supported by sql proc `getCollaborators()`

  Note: Collaborators cannot alter lists, only the items in the list

- ListItems

  1. can be added by a collaborator with 'W' or 'O' access

    - `POST /users/<uId>/lists/<lId>/items`
    - Supported by sql proc `getCollaborators()`
    - Supported by sql proc `addItem()`

  2. collaborators can get all or a item for a list they collaborate on

    - `GET /users/<uId>/lists/<lId>/items`
    - `GET /users/<uId>/lists/<lId>/items/<tId>`
    - Supported by sql proc `getCollaborators()`
    - Supported by sql proc `getListItems()`
    - Supported by sql proc `getItem()`

  3. Contributors can edit a item. Contributors with read-only can modify the status (completed/not)

    - `PUT /users/<uId>/lists/<lId>/items/<tId>`
    - Supported by sql proc `getCollaborators()`
    - Supported by sql proc `putItem()`

  4. Contributors can remove a item from a list

    - `DELETE /users/<uId>/lists/<lId>/items/<tId>`
    - Supported by sql proc `getCollaborators()`
    - Supported by sql proc `delItem()`

- Collaboration

A collaborator is a user with access (O, R, W) to a list, issued by the list owner.

- O: owner
- R: read-only
- W: read-write

1. Users can allow other users to read or read/write items in their Lists

- `POST /users/<uId>/lists/<lId>/collaborators`
- Supported by sql proc `addCollaborator()`

2. Users can get the collaborators for their list

- `GET /users/<uId>/lists/<lId>/collaborators`
- Supported by sql proc `getCollaborators()`

3. Users can edit the permissions of their list collaborators

- `PUT /users/<uId>/lists/<lId>/collaborators/<cId>`
- Supported by sql proc `putCollaborator()`

4. Users can revoke list collaborators

- `DELETE /users/<uId>/lists/<lId>/collaborators/<cId>`
- Supported by sql proc `delCollaborator()`

5. A user can find the lists/accesses they collaborate on

- `GET /users/<uId>/collaborations`
- Supported by sql proc `getCollaborations()`

6. A user can find access permissions for a list they collaborate on

- `GET /users/<uId>/collaborations/{lId}`
- Supported by sql proc `getCollaborationAccess()`

7. A collaborator can end collaborating on a list

- `DELETE /users/<uId>/collaborations/<lId>`
- Supported by sql proc `delCollaboration()`