

ReSTful Endpoint Design for Lists

Can we use the typical stories we tell in designing databases to discover HTTP command/ReSTful endpoint combinations?

- Signin
 1. created by users with valid credentials (returns token on success)
 - `POST /signin`
 2. Users can sign out (removes token)
 - `DELETE /signin`
- Password
 1. can be changed by logged in users
 - `PUT /password`
- Users
 1. are created by administrative users
 - `POST /users`
 2. can get information about other users
 - `GET /users`
 - `GET /users/<uId>`
 3. can update their profile information
 - `PUT /users/<uId>`
 4. are deleted by the user themselves or administrative users
 - `DELETE /users/<uId>`

Checkpoint

- Lists
 1. created by users, who own them
 - `POST /users/<uId>/lists`
 2. Users can get details of their lists
 - `GET /users/<uId>/lists`
 - `GET /users/<uId>/lists/<lId>`
 3. Users can update their list attributes
 - `PUT /users/<uId>/lists/<lId>`
 4. deleted by the user that owns them
 - `DELETE /users/<uId>/lists/<lId>`

- ListItems

For the purposes here a user or collaborator with valid access permissions is termed a contributor.

1. can be added by a contributor

- `POST /users/<uId>/lists/<lId>/items`

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

- `GET /users/<uId>/lists/<lId>/items`

- `GET /users/<uId>/lists/<lId>/items/<iId>`

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

- `PUT /users/<uId>/lists/<lId>/items/<iId>`

4. Contributors can remove an item from a list

- `DELETE /users/<uId>/lists/<lId>/items/<iId>`

- Collaboration

A collaborator is another user given access (r, rw) by a user to a list, by the list owner.

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

- `POST /users/<uId>/lists/<lId>/collaborators`

2. Users can get the collaborators for their list

- `GET /users/<uId>/lists/<lId>/collaborators`

3. Users can edit the permissions of their list collaborators

- `PUT /users/<uId>/lists/<lId>/collaborators/<cId>`

4. Users can revoke list collaborators

- `DELETE /users/<uId>/lists/<lId>/collaborators/<cId>`

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

- `GET /users/<uId>/collaborations`

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

- `GET /users/<uId>/collaborations/{lId}`

7. A collaborator can end collaborating on a list

- `DELETE /users/<uId>/collaborations/<lId>`