Application Features

|  |
| --- |
| TODO:   1. Error Handling 2. Method not supported 405 catch all 3. Support xml 4. Implement HEAD and OPTION 5. Implement HATEOAS (hypermedia as the engine of application state) 6. Implement sorting 7. Implement paging 8. Implement filtering 9. Implement searching |

# User

1. Types:
   1. Admin
   2. Member
2. User Attribute:
   1. FirstName
   2. LastName
   3. DateOfBirth (o)
   4. Email
   5. UserName
   6. Password
3. Functions:
   1. createUser() – Anonymous
   2. getUser() -- Authenticated
   3. updateUser() - Autheticate
   4. changePassword() – Authenticate
   5. recoverPassword()
   6. login() ---------------- literature gap
   7. logout()--------------- literature gap
   8. deleteUser() – Authenticate

Accounts

|  |  |  |
| --- | --- | --- |
| Method | URL | RPC |
| POST | Api/Accounts/{UserName} | createUser() – partial shared with profile |
| POST | Api/Accounts/Login | Login() |
| POST | (just delete token from client) | Logout() |

Credentials

|  |  |  |
| --- | --- | --- |
| Method | URL | RPC |
| PUT | Api/Accounts/{UserName}/Credentials | changePassword() |

Profile

|  |  |  |
| --- | --- | --- |
| Method | URL | RPC |
| GET | Api/Accounts/{UserName}/Profile | getUser() |
| PUT | Api/Accounts/{UserName}/Profile | createUser() – partial shared with accounts |
| PATCH | Api/Accounts/{UserName}/Profile | updateUser() |
|  |  |  |

User Groups

|  |  |  |
| --- | --- | --- |
| Method | URL | RPC |
| GET | Api/Accounts/{UserName}/Groups | getAllUserGroups() |
| GET | Api/Accounts/{UserName}/Groups/{groupId} | getUserGroupById() |
| POST | Api/Accounts/{UserName}/Groups | createUserGroup() |
| PUT | Api/Accounts/{UserName}/Groups/{groupId} | updteUserGroup() |
| DELETE | Api/Accounts/{UserName}/Groups/{groupId} | deleteUserGroup() |

Group Membership

|  |  |  |
| --- | --- | --- |
| Method | URL | RPC |
| GET | Api/Accounts/{UserName}/Groups/{groupId}/memberships | getAllGroupMemberships() |
| GET | Api/Accounts/{UserName}/Groups/{groupId}/memberships/{membershipId} | getGroupMembership() |
| POST | Api/Accounts/{UserName}/Groups/{groupId}/memberships | createGroupMembership() |
| DELETE | Api/Accounts/{UserName}/Groups/{groupId}/memberships/{membershipId} | deleteGroupMembership() |