This is a simple authentication and authorization service. The service allows users to be authenticated, and authorizes different behavior. The following is a brief design description**.**

**Data Structure**

User:

The entity to be authenticated and authorized different behavior. Here is data structure description for it.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| userName | String | unique identifier of one user |
| password | String | user’s password |
| roles | List<Role> | roles the user has |

Role:

The role of the user means what kind of behavior the user is authorized. A user can have many roles. The role is represented by a string. Here is data structure description for it.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| roleName | String | represent the role |

Session:

The entity that contains information associated with the token. Token is a string generated for the authenticated user. Here is the data structure for session.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| user | User | associated with token |
| time | long | time of creating or updating of the token |

**Interface**

All APIs are defined in a java interface. The interface is as follows:

public interface IUserService {

boolean createUser(String userName, String password);

boolean deleteUser(User user);

boolean createRole(String roleName);

boolean deleteRole(Role role);

void addRoleToUser(User user, Role role);

String authenticateUser(String userName, String password);

void invalidateToken(String token);

boolean checkRole(String token, Role role);

List<Role> allRoles(String token);

}

The implementation of this interface is UserServiceImpl. ScheduledExecutorService is used to periodically clear expired tokens in order to avoid too many expired tokens in memory.