Security, Users, and Roles

Overview

Documents the security and user management features.

Details

The security service provides authentication, authorization, and logout services. A user timeout also supported so users do not remained logged in indefinitely. The default timeout is set to 2 hours and can be easily changed. All REST services are secured by the use of an authorization token that is obtained from the security service and passed to other calls. For a truly secured environment, it is recommended that the system be run under HTTPS instead of HTTP. Otherwise usernames, passwords, and authorization tokens will be passed in plain text with HTTP requests.

When a user authenticates, an authToken (a random UUID) is passed back to the client which can be used on subsequent calls to the server. This authToken is checked against a map of valid tokens and can be used to identify the user who was assigned this token.   The time of an authentication request is recorded by the authToken so that it can be determined when the last request with this token was made. If upon checking an authToken the last request was longer ago than the "ihtsdo.security.timeout" setting, then the authorization is rejected and the user is redirected to log in again.

Restarting the server clears all authTokens and users must re-authenticate. The application will redirect users to the login page if their authToken is no longer valid (either because of inactivity timeout or server restart).

The sequence of events looks like this

- A username and password is supplied to the security service "authenticate" method.
    - The username and password are checked against the IHTSDO user management system
        - If it passes, a local application user (e.g. MapUserJpa) is either created or updated based on the full name and email address information supplied by the ITHSDO user management system.
        - If it fails, a 401 error is returned to the client
    - Once authenticated an authToken is generated and both the user and time-of-access are stored with the token.
    - The token is then returned to the client.
- A subsequent call to an application service includes an authToken
    - The application service requests the role of the user using that token (for authorization checks)
    - This internally requests the user using that token which performs a check that the authToken is itself valid and that the date of the last use of the token is not prior to the timeout period from the current time.
        - If either the authToken is not valid or the timeout period has been exceeded, an HTTP 401 error is returned to the client
        - Otherwise, the role is returned to the application
    - The application service then authorizes that role
        - If the role has privileges to access the capability, the request proceeds
        - Otherwise, an HTTP 401 error is returned to the client.
- A period of 2 hours passes without activity on a particular authToken
    - Any requests with this authToken will result in an HTTP 401 error.

User Management

While authentication of users is offloaded to the IHTSDO user management services, there is some user management within the application itself.   The system comes with a few "stock" users that never require authentication and are used to track the output of certain admin tasks. These users are

- guest - the default user that does not require specific authentication.
- loader - used for loading data from files to indicate that editors did not make these changes.
- qa - used as the original owner of QA records that enter the workflow - because they have not yet been assigned.

For any other user that authenticates normally, a local application user object is created with a default "VIEWER" application role. If the user already exists, the name and email address are requested from the user management service and are updated so the application tracking of user info is always kept up-to-date.

Once a "map user" exists, that user can be assigned as a specialist or lead or administrator on a particular project. Any user not specifically assigned to a project is considered a VIEWER.   VIEWERS do not have access to private (e.g. "non public") projects at all.

User Roles

The system make use of "application roles" and "map-project-specific roles". There are 4 roles:

- VIEWER - has read-only access to public projects, including map project details, and published map records.
- SPECIALIST- editor for a map project, can be assigned work, can edit work, can interact with other features in limited ways.
    - This is only a 'map-project-specific role', not an 'application role'.
- LEAD - lead editor for a map project, has all SPECIALIST capabilities and can also assign work, resolve conflicts, and interact with other features in more advanced ways.
    - This is only a 'map-project-specific role', not an 'application role'.
- ADMINISTRATOR - has full access to all operations and a dashboard with admin utilities not available to other users. In particular, can create and edit the various project metadata.

Each REST call not only verifies the authToken is valid but also authorizes the user's role against the task being attempted. Read-only services generally only require VIEWER role where as things like map project creation and metadata editing require ADMINISTRATOR role.   Any REST service that takes a map project or map project id requires a 'map-project-specific role' by that user on that project. See REST documentation for more details.

An admin tool exists to create a user with an application role of ADMINISTRATOR, and if needed an empty project to bootstrapp the rest of application configuration. See Admin Tools for more info.

Configuration

The following configuration settings affect security.

Property	Default Value	Notes
ihtsdo.security.activated	true	Indicates whether security is enabled or disabled. UAT and dev environments have security disabled by default. When security is disabled, an authorization token matching the username is used. NOTE: even when security is disabled, a user must still login but any password will authenticate any user.
ihtsdo.security.url	https://usermanagement.ihtsdotools.org/security-web/query/	The URL of the IHTSDO user management server. NOTE: Only IHTSDO security is supported. Ideally, this would be made more general in the future to support injection of a handler to support security.
ihtsdo.security.timeout	7200000	Inactivity timeout, in milliseconds.

References/Links

- User Management