AdminStore Web Service Design

Version 1.0

# Description

AdminStore is Web Service to facilitate application and project administration functionality, user authentication and authorization.

# Scope

AdminStore is primary web service to encapsulate all functionality directly and indirectly related to application and project administration, user authentication and authorization. AdminStore Web Service will contain all user records, privileges and other user and application settings and properties. The following design principles are used to limit the scope of the AdminStore Web Service for the current version:

1. IIS service supports DELETE HTTP methods, alternatively X-HTTP-Method-Override header to be used in conjunction with POST method.

# Interface

## Endpoint

Service will be hosted in the path /svc/adminstore

All methods except for sign-in expect to receive header Token to identify user session. Token is used to authorize user for an action being requested.

## Methods

GET /sessions

Method to query if session exists, expect to receive session token in header Session-Token to identify user session. Method will not extend lifetime of the session by SESSION\_TIMEOUT. This method to be used for sign in sequence only. Session information is returned back.

Returns status **200 OK** if user session exists and user is permitted to perform the operation on the artifact as specified.  
Returns status **401 Unauthorized** if session exists but user is not allowed to perform the operation on the artifact as specified.  
Returns status **403 Bad Request** if session, operation or artifact identifiers are not recognized.  
Returns status **404 Not Found** if session, operation or artifact not found.  
Returns status **500 Internal Server Error** if error:  
- database connection failed;  
- database connection established but table Sessions not found.

Example: PUT /svc/accesscontrol/sessions/read\_artifact/123

GET /sessions/select?ps={ps}&pn={pn}  
- where {ps} – page size , and {pn} – page number, 1-based.

Method returns paged list of existing sessions and max page number for the page size specified in the header. Methods expect to receive session token in header Session-Token to identify admin user session.

Returns status **200 OK** if admin user session exists and user is permitted to list sessions.  
Returns status **401 Unauthorized** if session exists but user is not allowed to list sessions, as only Application Admin group members are allowed to use this method.  
Returns status **403 Bad Request** if session, operation or artifact identifiers are not recognized.  
Returns status **404 Not Found** if session, operation or artifact not found.  
Returns status **500 Internal Server Error** if error:  
- database connection failed;  
- database connection established but table Sessions not found.

Example: GET /svc/accesscontrol/sessions?ps=20&pn=1

POST /sessions  
- where username and initial password hash are passed in the body of the request.

Method for standard (database) authentication of the user.

Returns user session token via X-Token header.

Returns status **200 OK** if no issue is detected.  
Returns status **401 Unauthorized** if username/password combination is not recognized.  
Returns status **403 Bad Request** if username and/or password are missing.  
Returns status **500 Internal Server Error** if error:  
- database connection failed;  
- database connection established but table Users not found.

Example: POST /svc/adminstore/sessions

POST /sessions/sso

Method for processing SAML response from Identity Provider.

Returns user session token via X-Token header.

Returns status **200 OK** if no issue is detected.  
Returns status **401 Unauthorized** if session cannot be instantiated for the user.  
Returns status **403 Bad Request** if SAML request is missing or malformed.  
Returns status **500 Internal Server Error** if error:  
- database connection failed;  
- database connection established but table Users not found.

Example: POST /svc/adminstore/session/sso

### DELETE /sessions

Method for explicit log out of application and release the license. Method expects session token in X-Token header.

Returns status **200 OK** if no issue is detected.  
Returns status **403 Not Acceptable** if session token is missing or malformed.  
Returns status **404 Not Found** if session cannot be found.  
Returns status **500 Internal Server Error** if error:  
- database connection failed;  
- database connection established but table Users not found.

GET /status  
Method to return current status of AdminStore Web Service.

Returns status **200 OK** if no issue is detected.  
Returns status **401 Unauthorized** if session cannot be found or user is not permitted the action requested.  
Returns status **406 Not Acceptable** if Token is missing or malformed.  
Returns status **500 Internal Server Error** if error:  
- database connection failed;  
- database connection established but table Users not found.

# Technology

AdminStore Web Service will be implemented using ASP.NET Web API/C#. Database will be Microsoft SQL Server 2012.

# Behavior

## Methods

GET /sessions  
Method retrieves session information. Return value should not be cached. Method will not change the state of the sessions it returns. **This method should not be used in any regular non-administrative subroutine.**

GET /sessions/select?ps={ps}&pn={pn}  
Method retrieves paged list of current sessions. Return value should not be cached. Method will not change the state of the sessions it returns. **This method should not be used in any regular non-administrative subroutine.**

POST /sessions  
Method checks user credentials. If no credentials are supplied, service attempts to obtain windows integrated credentials, then matching username against database record of the user. If matched and user record marked as windows integrated account, then session is created.

If credentials are supplied, then If matched, AccessControl web service is called to create session. Session token received from AccessControl web service is then forwarded to the application.

POST /sessions/sso  
Method processes the SAML response from Identity Provider. Invokes AccessControl web service to create session and returns session token back to the application.

### DELETE /sessions

Method logs out user and releases the license. Invokes AccessControl and LicenseStore web services.

GET /status  
Method tries to execute SELECT COUNT(\*) FROM [USERS]; to test database connection.