Client API Specifications

API to get user preferences based on User ID: End user has already set up preferences to get the data.

API Name	api/v1/UserPreference/preferences
API Type	POST
Authentication	Halliburton Okta Token (Okta token is required in Request Header "Authorization" Key)

Request	{
	"email": "sangeetha.puthiyaveetil@halliburton.com",
	"wellID": "4623e417-1da6-4155-9d21-9f06d738b3d4", //This is optional. If user will not set wellID then API will return all wellID which you have access with preferences.
	"application": "DSRT365", //This is optional. If user will not send App name, then API will try to apply Default well preference which user has configured in User Pref App.
	}

Response	{
	"preferences": [
	{
	"wellID": "4623e417-1da6-4155-9d21-9f06d738b3d4",
	"application": "DSRT365", //This is an option. It's not mandatory.
	"decimalPlaces": 1,
	"depthPrecision": 4,
	"timeZone": "(GMT-06:00) Central Time (US & Canada)",
	"uom": "Metric_Canada-2021-04-14"
	}
	]
	}

API to get well defaults as preference.

API Name	api/v1/UserPreference/wellpreferences
API Type	POST
Authentication	Halliburton Okta Token (Okta token is required in Request Header "Authorization" Key)

Request	{
	"wellUIDs": [
	"4623e417-1da6-4155-9d21-9f06d738b3d4",
	"ec34ea6d-8ec7-47e2-b873-1a49d99f43f2",
	"dee6efd0-d220-4933-9cb9-3e4f49a7f284"
	]
	}

Response	{
	"preferences": [
	{
	"wellID": "4623e417-1da6-4155-9d21-9f06d738b3d4",
	"application": "",
	"decimalPlaces": 1,
	"depthPrecision": 4,
	"timeZone": "(GMT-05:00) Eastern Time (US & Canada)",
	"uom": "English"
	},
	{
	"wellID": "ec34ea6d-8ec7-47e2-b873-1a49d99f43f2",
	"application": "",
	"decimalPlaces": 1,
	"depthPrecision": 4,
	"timeZone": "(GMT-06:00) Central Time (US & Canada)",
	"uom": "Metric_Canada-2021-04-14"
	},
	{
	"wellID": "dee6efd0-d220-4933-9cb9-3e4f49a7f284",
	"application": "",
	"decimalPlaces": 1,
	"depthPrecision": 4,
	"timeZone": "(GMT-06:00) Central Time (US & Canada)",
	"uom": "Metric_Canada-2021-04-14"
	}
	]
	}

User Preference API URLs

Dev: https://dev.rtsuserpref.ienergy.halliburton.com/service/
Stage: https://stg.rtsuserpref.ienergy.halliburton.com/service/
Production: https://rtsuserpref.ienergy.halliburton.com/service/

Note: User Preference also provide an UI interface that can be integrated with any application. Please request more details if needed.

API Authorization Details

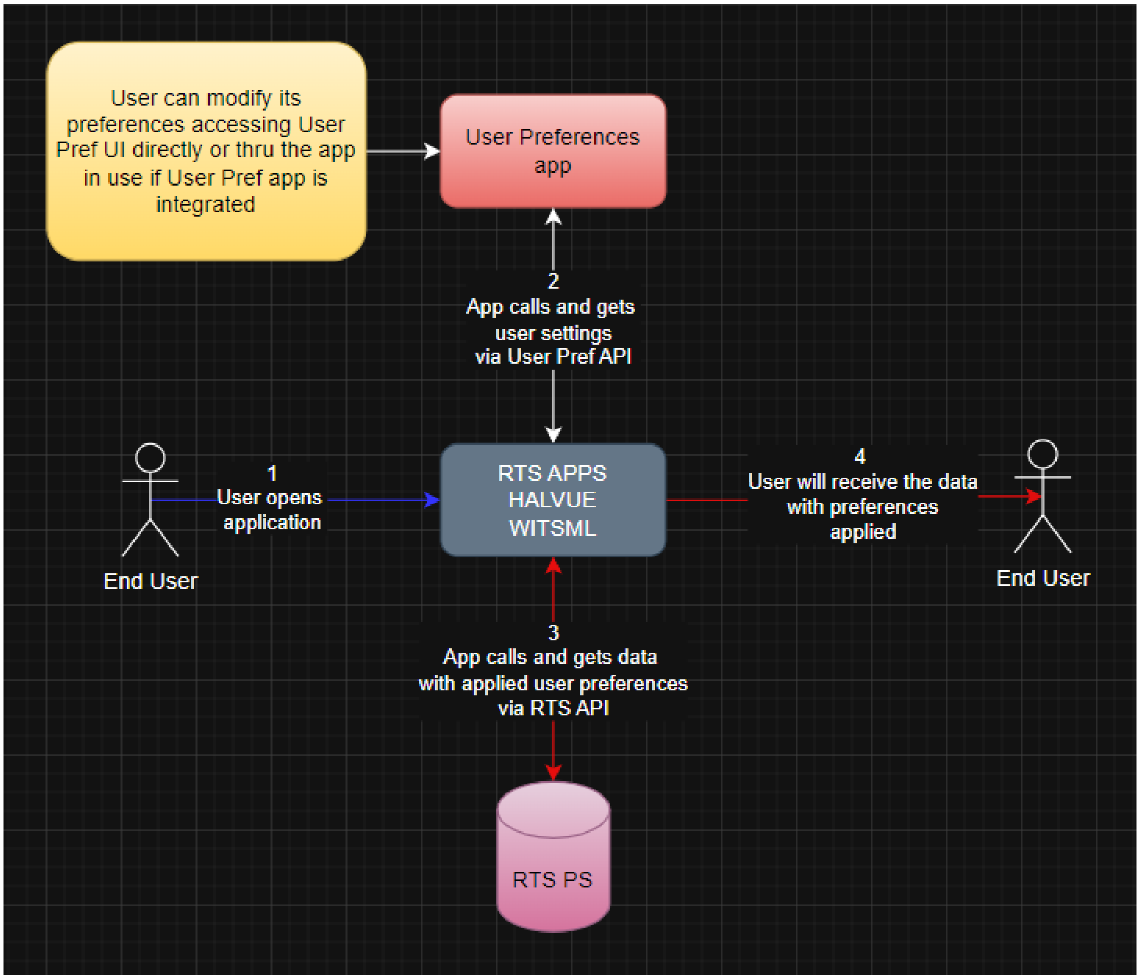
User Preferences API Authentication is using OKTA Bearer Token.

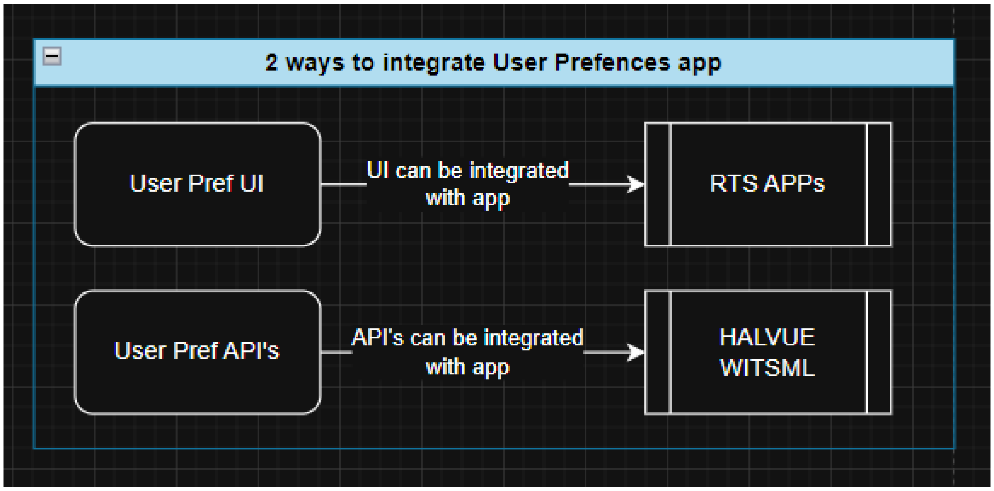
The authentication process follows these steps:

1. Client Requests Access:
    a. The client (user or application) requests access to a protected API endpoint by including an 'Authorization' header with a Bearer token.
2. Extract and Validate Token:
    b. The middleware extracts the token from the request header.
    c. If the token is missing or improperly formatted, the request is rejected with a '401 Unauthorized' response.
3. Token Validation Process:
    d. The token is checked against an internal validation method (InternalTokenValidation).
    e. If validation fails, an additional check is performed using Azure Active Directory (AadValidator).
    f. If neither validation method succeeds, the request is denied.
4. Claims Extraction and User Identity Setup:
    g. If the token is valid, user claims (such as email) are extracted.
    h. A `ClaimsPrincipal` is created and associated with the request.
5. Authorized Access:
    i. If the user is authorized, the request is processed further.
    j. If authorization fails, the response returns '401 Unauthorized' with an appropriate message.

This authentication mechanism ensures secure access control using OKTA Bearer tokens, validating them via JWKS or Azure AD before granting access to API endpoints.

Workflow Diagram


User can modify its preferences accessing User Pref UI directly or thru the app in use if User Pref app is integrated
User Preferences app
2
App calls and gets user settings via User Pref API
End User
1
User opens application
RTS APPS HALVUE WITSML
4
User will receive the data with preferences applied
End User
3
App calls and gets data with applied user preferences via RTS API
RTS PS


2 ways to integrate User Prefences app
User Pref UI
UI can be integrated with app
RTS APPs
User Pref API's
API's can be integrated with app
HALVUE WITSML