SWARCO TRAFFIC SYSTEMS

Querying static and dynamic PLS data via the SMI.

SWARCO I First in Traffic Solutions.

INHALT

STATUS OF THE DOCUMENT

Version	Date	Author	Description
1.0	16.03.2023	Rosario Scicolone	Base version
1.1	02.06.2023	Andreas Pfitzner	Translated to English.
2.0	08.02.2024	Andreas Pfitzner	Add cahpter 3 - Querying the SMI via WebSocket

1 Introduction

The SWARCO Mobility Interface (SMI) is used to transmit all static and dynamic data from a PLS/Cloud PGS via a REST interface or provide update via websocket connection

This document aims to describe the individual steps for querying the mentioned data sets.

The current interface description of SMI can be requested through GF Parken.

2 Querying via the SMI via REST-API Call

2.1 Access to SMI REST-API

The SMI URLs required for querying in Cambridge are as follows:

- dynamic:
  - https://mycity.swarco.com/api/swarco.pgs.smi.v8/getDynamicPOIDataByPgs/CLD_PGS_a868a172-e1d9-4087-80ca-814bdd9a10c2
- static:
  - https://mycity.swarco.com/api/swarco.pgs.smi.v8/getStaticPOIDataByPgs/CLD_PGS_a868a172-e1d9-4087-80ca-814bdd9a10c2

User: cambridge_smi@swarco.com

2.2 Authentication – generate token

To use the API calls, authentication using a Bearer token is required. Below is an example PowerShell script that demonstrates how to obtain the Bearer token and use it in the request.

Powershell:

```
#set username & password
$password = "PASS"
$username = "USER"

# OAuth 2.0- authentication server -URL
$oauth_url = "https://sso.swarco.com/auth/realms/swarco/protocol/openid-connect/token"

# Client ID
$client_id = "swarco.ui.base"

# access token
$token_url = "$oauth_url"
$body = @{
    grant_type = "password"
    client_id = $client_id
    username = $username
    password = $password
}
$headers = @{ 'Content-Type' = 'application/x-www-form-urlencoded' }
$response = Invoke-RestMethod -Uri $token_url -Method Post -Body $body -Headers $headers

# Extract access token from the response
if ($response) {
    $access_token = $response.access_token
    # Use the token for further API requests
} else {
```

```
    Write-Host " Error while retrieving the token "
}
```

2.3 Quering the SMI via REST-API calls

```
# Perform REST API request using the access token
$api_url =
"https://mycity.swarco.com/api/swarco.pgs.smi.v8/getDynamicPOIDataByPgs/CLD
_PGS_d4cca5b5-5903-41bb-bfb0-fbec919c42fb"
$query_params = @{}
$headers = @{
    "Content-Type" = "application/json"
    'Authorization' = "Bearer $access_token"
}
$response = Invoke-RestMethod -Uri $api_url -Headers $headers
```

Please replace USER and PASS with the appropriate access credentials.

The cURL request would look like this::

```
curl --location --request POST
'https://sso.swarco.com/auth/realms/swarco/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'client_id=swarco.ui.base' \
--data-urlencode 'username=USER' \
--data-urlencode 'password=PASS'
```

The response will be a JSON in the following format::

```
{
    "access_token": "TOKEN",
    "expires_in": 7200,
    "refresh_expires_in": 7200,
    "refresh_token": "REFRESH_TOKEN",
    "token_type": "Bearer",
    "not-before-policy": 0,
    "session_state": "51447665-40d6-4293-bc64-9343a8e3f443",
    "scope": "email profile"
}
```

The Bearer token for communication can be found in the "access_token" field.

3 Querying the SMI via WebSocket

3.1 Access to SMI WebSocket

To access the SMI via WebSocket, use the following URL for queries in Cambridge:

- WebSocket URL: wss://prod4-parking-dach.swarco.com

User credentials remain the same as for Rest-API calls (refer to Chapter 2.1)

3.2 Quering SMI via WebSocket

To perform the websocket request you have to use the access token see 2.2)

- **URL Call and Connection Establishment:**
  - Connect to the WebSocket URL.

- **Sending a Message to the WebSocket to Receive Data:**
  - Send a message in the following format to the WebSocket to receive data:

```
{
        "command": "getDynamicPOIDataByPgs",
        "id": "CLD_PGS_a868a172-e1d9-4087-80ca-814bdd9a10c2",
        "token": "token from Chapter 2.2."
}
```

After successfully establishing the connection and validating the token, you will receive the initial data showing the current occupancy status at the time of retrieval. Subsequent updates will only be for individual facilities.

3.2.1 Example for initial message:


```
{
  "total": 3,
  "dynamicPOIData": [
    {
      "timeStamp": "2024-02-06T13:55:55.000Z",
      "occupancyTotal": [
        {
          "parkingSpaceType": "total",
          "capacity": 50,
          "vacantSpaces": 20,
          "occupiedSpaces": 30,
          "colourIndicator": "green"
        },
        {
          "parkingSpaceType": "shortterm",
          "capacity": 30,
          "vacantSpaces": 15,
          "occupiedSpaces": 15,
          "colourIndicator": "yellow"
```

```
        }
      ],
      "objectID": "PGS_EXAMPLE_1",
      "name": "Location A"
    },
    {
      "timeStamp": "2024-02-06T13:56:25.000Z",
      "occupancyTotal": [
        {
          "parkingSpaceType": "total",
          "capacity": 75,
          "vacantSpaces": 25,
          "occupiedSpaces": 50,
          "colourIndicator": "yellow"
        },
        {
          "parkingSpaceType": "shortterm",
          "capacity": 40,
          "vacantSpaces": 20,
          "occupiedSpaces": 20,
          "colourIndicator": "green"
        }
      ],
      "objectID": "PGS_EXAMPLE_1",
      "name": "Location B"
    },
    {
      "timeStamp": "2024-02-06T13:56:27.000Z",
      "occupancyTotal": [
        {
          "parkingSpaceType": "total",
          "capacity": 100,
          "vacantSpaces": 10,
          "occupiedSpaces": 90,
          "colourIndicator": "red"
        },
        {
          "parkingSpaceType": "shortterm",
          "capacity": 60,
          "vacantSpaces": 10,
          "occupiedSpaces": 50,
          "colourIndicator": "red"
        },
        {
          "parkingSpaceType": "longterm",
          "capacity": 40,
          "vacantSpaces": 5,
          "occupiedSpaces": 35,
          "colourIndicator": "yellow"
        }
      ],
      "objectID": "PGS_EXAMPLE_1",
      "name": "Location C"
    }
  ]
}
```

3.2.2 Example for update message:


```
{
  "timeStamp": "2024-02-06T14:06:12.000Z",
  "occupancyTotal": [
    {
      "parkingSpaceType": "total",
      "capacity": 50,
      "vacantSpaces": null,
      "occupiedSpaces": 45,
      "colourIndicator": null
    },
    {
      "parkingSpaceType": "shortterm",
      "capacity": 30,
      "vacantSpaces": null,
      "occupiedSpaces": 28,
      "colourIndicator": null
    },
    {
      "parkingSpaceType": "longterm",
      "capacity": 40,
      "vacantSpaces": null,
      "occupiedSpaces": 38,
      "colourIndicator": null
    }
  ],
  "occupancyAreas": [],
  "objectID": "PGS_EXAMPLE_1",
  "guID": null,
  "name": "Location C"
}
```