Registration System

Facility API Specification

SPEC-249

Issue: 1.0

Issue Date: September 1, 2021

Public

Connecting Today.
Powering Tomorrow.

This document uses IESO Internal General Template IESO_TPL_0066 version 6.

Document Change History

Issue	Reason for Issue	Date
1.0	First Release	September 1, 2021

Table of Changes

Reference (Section and Paragraph)	Description of Change

Table of Contents

Contents

List of Tables

1. Introduction

1.1 Purpose

1. The purpose of this document is to provide technical specification for the Registration System Application Programing Interface (API) web service.

2. These specifications are reviewed by the IESO solution analyst, IESO business analyst and providers of systems that will interact with the Registration system through an automated interface. These specifications will be approved by the solution steward for the Registration System.

1.2 Assumptions and Limitations

3. The technical specification document describes the API provided as part of the Registration System. The API is provided to enable participant to retrieve registration information from the IESO Registration System.

4. The technical specification document is a living document and will be updated as new web services are added or existing web services are changed.

1.3 Conventions

5. The standard conventions followed for this document are as follows:

- Quotation marks are used to highlight process or component names;
- Italics are used to highlight publication, titles of procedures, letters and forms; and
- All time mentioned in this document is in East Standard Time (EST).

1.4 Glossary

6. **Custodian** – The person Assigned in Online IESO who maintains the API account used to access the Registration API.

1.5 How This Document Is Organized

7. Section 2 describes the general information of the Registration API.

8. Section 3 describes the specific details associated with each ***web service retrieve function***.

– End of Section –

2. Registration API General Information

2.1 Registration REST API

9. The Registration API is a Representational State Transfer (REST) Application Programming Interface (API). The REST API is a web service architecture that focuses on resources for a specific service and their representations. A client sends a stateless request to a server element located at a Uniform Resource Identifier (URI) and performs operations with standard HTTP methods, such as **GET**, **POST**, **PUT**, and **DELETE**.

10. The default request and response type of the Registration API should be in JSON (JavaScript Object Notation) format.

2.2 Authentication and Authorization

11. The Registration API leverages API keys for authentication. There are multiple methods to authenticate using an API key. With basic authentication, a client is required to send a username and password with all requests to correctly authenticate. The API encodes these credentials in Base64 format and transmits in the HTTP authorization header. If a request does not include an appropriate Authorization header, the API returns a 401 Authorization Required error.

12. To authenticate using Basic Authentication you must supply the API key as either the username with a blank password, or as the password with a blank username.

Basic Authentication Header (with null username):

```
curl -u :${API_KEY} https://online.ieso.ca/suite/webapi/endpoint
```

Basic Authentication Header (with null password):

```
curl -u ${API_KEY}: https://online.ieso.ca/suite/webapi/endpoint
```

13. To authenticate using an Appian-API-Key header without Basic Authentication you must supply a header called Appian-API-Key as follows:

Appian-API-Key Header:

```
curl https://online.ieso.ca/suite/webapi/endpoint -H "Appian-API-Key: ${API_KEY}"
```

14. To authenticate using a Bearer Token header without Basic Authentication you must supply an Authorization header as follows:

Bearer Token Header:

```
curl https://online.ieso.ca/suite/webapi/endpoint -H "Authorization: Bearer ${API_KEY}"
```

15.

16. HTTPS is required by the Registration API to provide end-to-end encryption when sending credential and business data information across the open internet.

17. The API (machine) account should be used to access the Registration API. The Rights Administrator registered for your organization needs to request an API account with the “Registration API” access role or apply this role to an existing API account in Online IESO by selecting “Manage System Access” under the ACTION tab. The Rights Administrator for your organization must grant the “Registration API” access role to a Custodian machine account. If a Custodian has not yet been

registered, then the Rights Administrator must request a new machine account and provide the Custodian details.

18. The API account will have the ability to execute the web services for multiple participants. Participants will be responsible for approving access to their information and also that the service provider meets all of their requirements with respect to information protection and privacy.

2.3 API Endpoint

19. Interaction with the Registration API begins with a request to the URL. The format of the Registration API URL is made up of a root endpoint and a resource URL. For example, to get the list of pre-project applications from the Registration system, the API endpoint is https://online.ieso.ca/suite/webapi/reg-v1-facilities/. https://online.ieso.ca/suite/webapi/ is the root endpoint and /reg-v1-facilities/ is the resource URL. The following is the list of the Registration API root endpoints.

Table 2-1 API Root Endpoints

Environment	Root Endpoint
Sandbox	https://onlinesandbox.ieso.ca/suite/webapi/
Production	https://online.ieso.ca/suite/webapi/

2.4 API HTTP Methods

20. The Registration API uses the standard GET, PUT, DELETE and POST HTTP methods for the different type of request to a resource. The Registration API supports the following list of HTTP methods.

Table 2-2 API HTTP Methods

Method	Description
GET	Retrieval of a single resource or search for resources. For example, retrieve an facility with details or a list of facilities.

2.5 API Status Codes and Error Messages

21. The Registration API returns standard HTTP status codes for the API successful or failed calls. The following is a list of most common status codes with a brief description of each code.

Table 2-3 API Status Codes

Code	Text	Description
200	OK	Standard response for successful API requests
201	Created	The resource was successfully created
400	Bad Request	Invalid request due to bad syntax or validation failed.
401	Unauthorized	Authentication has failed or was not provided
403	Forbidden	The request has insufficient permissions to perform the action or read the resource, or the action is unsupported in general
404	Not Found	The requested resource cannot be found

Code	Text	Description
408	Request Timeout	The server did not receive a complete request message within the time that was prepared to wait.
429	Too Many Requests	Too many requests have been made in a given amount of time
500	Internal Server Error	A generic error message when something is broken
503	Service Unavailable	The server is currently unavailable; possibly down for maintenance

22. In general, the status codes follow the following rules:

    a. 200+ means the request has succeeded

    b. 400+ means an error that originates from the client has occurred

    c. 500+ means an error that originates from the server has occurred

23. Status code 429 will be returned if the number of requests made by an API account exceeds the following thresholds:

Period	**Total Requests (per 5 minutes)**
Business Hours (7:00 AM – 6:00 PM EST) Monday-Friday excluding holidays	150
All other hours	600

2.6 API General Validation Rules

24. The Registration API solution should be designed and implemented with respect to information protection and privacy. The API should return “Bad Request” with status code 400 if the validation of an API account and Organization relationship failed.

25. The mandatory request and data input parameters such as Organization ID and Facility ID parameters shall be validated before executing web service calls. The API should return “Bad Request” with status code 400 if validations failed.

26. The API shall return “Record Not Found” with status code 404 if the requested record is not found in the Registration system.

Table 2-4 API General Validations Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is required	400	{
		“message”: “Validation Failed”
		“details” : “organization_id is required”
		}

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		“message”: “Validation Failed”
		“details” : “The API account doesn’t have permissions to access Registration system for organization “ + organization_id
		}

Facility is not found	404	{
		“message”: “Record Not Found”
		“details” : “Couldn’t find a facility with the input parameters in the Registration system ” + substation_id
		}

2.7 Testing the Requests to the API

27. There are numerous tools you can use to test your requests to the Registration API. Below is a list of some common tools:

    a. Postman - Chrome Extension

    b. SOAP UI - Automated testing tools for SOAP and REST APIs

2.8 API Assumptions and Limitations

28. The Registration API has the following system default settings.

Table 2-5 API System Default Settings

Attribute	Description
Versioning	The Registration API has not stated versioning; however this attribute has been included in the URL structure for potential usage at a later time. The current defaulted version of the API is “v1”
UTF-8 Encoding	Every string passed to and from the Registration API needs to be UTF-8 encoded.
Date Format	All date and time field values are in the ISO standard format (YYYY-MM-DDThh:mm:ssZ)
Accept Header	The “Accept” Header must be specified on all **GET** requests in order to control the accepted output format of the request. The default Accept header is application/json.
Content-Type Header	The “Content-Type” header must be specified on **POST** and **PUT** requests in order to control the accepted input and output format of the request. The default Content-Type Header is application/json.

Attribute	Description
batch-size	The maximum number of records returned per request. The default value of batch-size is 20.
total-count	The total number of records available for the given input parameters.
start-index	The index of the first record to be returned, the default value of the start index is 1. If an invalid value is passed the system will use the default (instead of returning a validation error.

29. Unnecessary information in the sample JSON response has been left out for brevity.

– End of Section –

3. Registration API Retrieve Functions

3.1 Retrieve Registered Facilities

30. The participants shall be able to retrieve the list of registered facilities (substations) associated with an organization.

31.

Attribute	Details
URI Suffix	/reg-v1-facilities
HTTP Method GET/POST/PUT	GET

3.1.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization ID
start_index	Number	Optional	Response will be limited to a predefined batch size starting at record 1 if the start index is not provided. If the number of total records exceeds the batch size the requester must provide a start index greater than the batch size to retrieve the subsequent batch of records.
last_updated_dt_from	Date	Optional	Facility Last Updated Date From
last_updated_dt_to	Date	Optional	Facility Last Updated Date To

3.1.2 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		"message": "Validation Failed"
		"details": "organization_id is invalid or was not provided"
		}

Last Updated Date From is invalid	400	{ "message": "Validation Failed" "details": "Last Updated Date From is invalid. Last Updated Date From is required if Last Updated Date To is present."

Validation Rule	HTTP Status Code	Examples/Explanations
		}

Last Updated Date To is invalid	400	{
		“message”: “Validation Failed”
		“details”: “Last Updated Date To is invalid. Last Updated Date To is required if Last Updated Date From is present.”
		}

Last Updated Date To is invalid	400	{
		“message”: “Validation Failed”
		“details”: “Last Updated Date To should be greater than Last Updated Date From.”
		}

Start index exceeds total record count	400	{
		“message”: “Start Index Invalid”
		“details”: “The requested start index exceeds the total number of records ” + start_index
		}

3.1.3 Response

Field	Data Type	Data Size	Cardinality	Explanations
substation_id	Number	15,0	0..*	Substation ID
substation_name	String	255	0..*	Substation Name
registration_status_name	String	50	0..*	Registration Status
substation_sub_class_name	Number	(5,0)	0..*	Substation Sub-Class
sub_region_name	Number	(5,0)	0..*	Substation Region
electrical_zone_name	Number	(5,0)	0..*	Electrical Zone
hydro_one_sector	String	15	0..*	Hydro One Sector
settlement_taxation_area	String	15	0..*	Settlement Taxation Area
gps_north	Number	(15,5)	0..*	Global Positioning System North
gps_west	Number	(15,5)	0..*	Global Positioning System West
connection_type	String	20	0..*	Connection Type
telemetry_perf_class	String	20	0..*	Telemetry Performance Classification
telemetry_size_class	String	20	0..*	Telemetry Size Classification
sps_ras_participation	String	10	0..*	Special Protection system Participant
double_contingency_flag	String	1	0..*	Double Contingency Flag
key_facility	String	1	0..*	Key Restoration Facility
address1	String	50	0..*	Address 1
address2	String	50	0..*	Address 2
address3	String	50	0..*	Address 3
postal_code	String	20	0..*	Postal Code/Zip Code
city	String	20	0..*	City
province	String	2	0..*	Province
country	String	20	0..*	Country
effective_date	Date		0..*	Effective Date
last_update_dt	Date		0..*	Last Updated Date
last_update_user	String	50	0..*	Last Update User ID
start_index	Number	3	0..*	Start Index
batch_size	Number	3	0..*	Batch Size
total_count	Number	3	0..*	Total Count

3.1.4 Sample Request

```
{
" organization_id": 123456,
“last_updated_dt_from”: "2018-01-01T00:00:00Z"
“last_updated_dt_to”: "2018-01-15T00:00:00Z"
}
```

3.1.5 Sample Response

```
{
  "start_index": 1,
  "batch_size": 50,
  "total_count": 4,
“facilities” : [
          {
          "substation_id": 123456,
          "substation_name ": “Test 1”,
          "registration_status_name": “Registered”,
          ……
          },
          ……
          {
          "substation_id": 123457,
          "substation_name ": “Test  2”,
          "registration_status_name": “Registered”,
          ……
          }
    ]
}
```

3.2 Retrieve Facility Registered Equipment List

32. The participants shall be able to retrieve the list of registered equipment information associated with a facility.

33.

Attribute	Details
URI Suffix	/reg-v1-equipmentList
HTTP Method GET/POST/PUT	GET

3.2.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization Id
substation_id	Number	Optional	Substation Id
equipment_type	Number	Optional	Equipment Type
last_updated_dt_from	Date	Optional	Equipment Last Updated Date From
last_updated_dt_to	Date	Optional	Equipment Last Updated Date To

3.2.2 Equipment Type Values

Equipment Type	Equipment Type Name
2	Synchronous Machine
3	Wind Turbine Generator
4	PV Inverter
7	Load
14	Circuit Switcher
15	Disconnect Switch
16	Circuit Breaker
17	Shunt Capacitor
18	Shunt Reactor
19	Bus
20	Transformer
21	Circuit
22	Static VAR Compensator (SVC)
23	STATCOM

Equipment Type	Equipment Type Name
24	Collector System
25	Meteorological Tower
26	Nacelle Mounted
28	Sodar Unit
30	Opener
31	Series Capacitor
38	Induction Machine
41	Lidar Unit
42	UFLS Relay Group

3.2.3 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		“message”: “Validation Failed”
		“details**”: “**organization_id is invalid or was not provided”
		}

substation_id is invalid	400	{
		“message”: “Validation Failed”
		“details”: “substation_id is invalid”
		}

Substation / Equipment is not found	404	{
		“message”: “Record Not Found”
		“details”: “Could not find equipment with the input parameters in the Registration system ”
		}

equipment_type **is** invalid	400	{
		“message”: “Validation Failed”
		“details”: “equipment_type is invalid “ + equipment_type
		}

Last Updated Date From is invalid	400	{ “message”: “Validation Failed”

Validation Rule	HTTP Status Code	Examples/Explanations
		“details”: “Last Updated Date From is invalid. Last Updated Date From is required if Last Updated Date To is present.” }

Last Updated Date To is invalid	400	{
		“message”: “Validation Failed”
		“details”: “Last Updated Date To is invalid. Last Updated Date To is required if Last Updated Date From is present.”
		}

Last Updated Date To is invalid	400	{
		“message”: “Validation Failed”
		“details”: “Last Updated Date To should be greater than Last Updated Date From.”
		}

3.2.4 Response

Field	Data Type	Data Size	Cardinality	Explanations
substation_id	Number	15,0	0..*	Substation ID
substation_name	String	255	0..*	Substation Name
equipment_id	Number	(15,0)	0..*	Equipment Id
equipment_name	String	100	0..*	Equipment Name
equipment_type	Number	(5,0)	0..*	Equipment Type
equipment_type_name	String	50		Equipment Type Name
operating_nomenclature	String	100	0..*	Operating Nomenclature
voltage_level	String	10	0..*	Voltage Level
registration_status	Number	(5,0)	0..*	Registration Status
effective_date	Date		0..*	Date when the equipment is registered
last_update_dt	Date		0..*	Date when the equipment is last updated
last_update_user	String	50	0..*	Last Update User

3.2.5 Sample Request

```
{
" organization_id": 123456,
" substation_id": 234567
}
```

3.2.6 Sample Response

```
"equipment": [{
        "equipment_id" : 129839,
         "equipment_name " : "",
        "operating_nomenclature" : "OCT-SM-GAS-STEAM",
        "equipment_type" : 2,
        "equipment_type_name" : "Synchronous Machine",
        "effective_date" : "2016-06-15Z",
        "last_update_dt" : "2016-06-14T15:09:33Z",
        "last_update_user" : "user1"
        ……
        },

        ……

]
```

3.3 Retrieve Facility Registered Equipment Details

34. The participants shall be able to retrieve the registered equipment detail information.

35.

Attribute	Details
URI Suffix	/reg-v1- equipmentDetails
HTTP Method GET/POST/PUT	GET

3.3.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization ID
equipment_id	Number	Mandatory	Equipment Id

3.3.2 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		"message": "Validation Failed"
		"details": "organization_id is invalid or was not provided"
		}

equipment_id is invalid	400	{
		"message": "Validation Failed"
		"details": "equipment_id is invalid or was not provided"
		}

equipment_id is not found	404	{
		"message": "Record Not Found"
		"details": "Could not find the equipment with the input parameters in the Registration system."
		}

3.3.3 Response

Field	Data Type	Data Size	Cardinality	Explanations
equipment_details	Structure			JSON object structure
equipment_id	Number	(15,0)	1	Equipment Id
operating_nomenclature	String	100	1	Operating Nomenclature
equipment_name	String	100	1	Equipment Name
equipment_type	Number	(5,0)	1	Equipment Type
equipment_type_name	String	50	1	Equipment Type Name
effective_date	Date		1	Effective Date
creation_dt	Date		1	Creation Date
last_update_dt	Date		1	Last Update Date
last_update_user	String	50	1	Last Update User
field_values	Structure		0..*	JSON object structure
field_name	String			Field Name
filed_display_name	String			Field Display Name
field_value	String			Field Value
children_equipment	Structure		0..*	JSON object structure same as equipment_details
dynamic_model	Structure		0..*	JSON object structure
dynamic_model_id	Number			Dynamic model Id
model_name	String			Dynamic model Name
description	String			Description
model_type_name	String			Dynamic model type name

3.3.4 Sample Request

```
{
" organization_id": 123456,
" equipment_Id ": 234567
}
```

3.3.5 Sample Response

```
{
 "total_count" : 1,
 "equipment_details" : [ {
  "equipment_id" : 148297,
  "operating_nomenclature" : "GLOBEX COLLECTOR SYSTEM",
  "equipment_name" : "",
  "equipment_type" : 24,
    "equipment_type_name" : "Collector System",
    "effective_date" : "2019-06-07Z",
    "last_update_dt" : "2019-06-06T00:00:00Z",
    "last_update_user" : "user1",
    "field_values" : [ {
      "field_name" : "operatingNomenclature",
      "field_display" : "Operating Nomenclature",
      "field_value" : "GLOBEX COLLECTOR SYSTEM"
        }, {
      "field_name" : "voltageLevel",
      "field_display" : "Voltage Level (kV)",
      "field_value" : "230"
        },
        ….
       ],
    "children_equipment" : [ {
      "equipment_id" : 148296,
      "operating_nomenclature" : "PV INVERTER 2019",
      "equipment_name" : "",
      "equipment_type" : 4,
      "equipment_type_name" : "PV Inverter",
      "effective_date" : "2019-06-07Z",
      "last_update_dt" : "2019-06-06T10:47:01Z",
      "last_update_user" : "user1",
      "field_values" : [ ]
        } ],
    "dynamic_model" : [ {
      "dynamic_model_id" : 148298,
      "model_name" : "REGCAU1",
      "description" : "Renewable Energy Generator/Converter Model",
      "model_type_name" : "Generator"
    } ]
  } ]
}
```

3.4 Retrieve Facility Dynamic Model Details

36. The participants shall be able to retrieve the dynamic model details associated with an equipment.

37.

Attribute	Details
URI Suffix	/reg-v1- dynamicModelDetails
HTTP Method GET/POST/PUT	GET

3.4.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization ID
equipment_id	Number	Mandatory	Equipment Id

3.4.2 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		"message": "Validation Failed"
		"details": "organization_id is invalid or was not provided"
		}

equipment_id is invalid	400	{
		"message": "Validation Failed"
		"details": "equipment_id is invalid or was not provided"
		}

the dynamic model details are not found	404	{
		"message": "Record Not Found"
		"details": "Could not find the dynamic model details associated with equipment in the Registration system "
		}

3.4.3 Response

Field	Data Type	Data Size	Cardinality	Explanations
dynamic_model	**Structure**		0..*	JSON object structure
dynamic_model_id	Number			Dynamic model Id
model_name	String			Dynamic model Name
description	String			Description
model_type_name	String			Dynamic model type name
parameter_values	**Structure**		1..*	JSON object structure
model_parameter_name	String			Parameter Name
model_parameter_value	Number			Parameter Value

3.4.4 Sample Request

```
{

" organization_id": 123456,
" equipment_Id ": 234567
}
```

3.4.5 Sample Response

```
{
  "dynamic_model" : [ {
   "dynamic_model_id" : 148298,
   "model_name" : "REGCAU1",
   "description" : "Renewable Energy Generator/Converter Model",
   "model_type_name" : "Generator",
   "parameter_values" : [ {
     "model_parameter_name" : "Lvplsw (Low Voltage Power Logic) switch",
     "model_parameter_value" : "2"
      }, {
     "model_parameter_name" : "Lvplsw (Low Voltage Power Logic) switch",
     "model_parameter_value" : "1"
     },
      ….
  } ]
   ….
 } ]
```

3.5 Retrieve In-Flight Facilities

38. The participants shall be able to retrieve the list of in-flight facilities (substations) associated with an organization.

39.

Attribute	Details
URI Suffix	/reg-v1-if-facilities
HTTP Method GET/POST/PUT	GET

3.5.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization ID
start_index	Number	Optional	Response will be limited to a predefined batch size starting at record 1 if the start index is not provided. If the number of total records exceeds the batch size the requester must provide a start_index greater than the batch size to retrieve the subsequent batch of records.

3.5.2 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		"message": "Validation Failed"
		"details": "organization_id is invalid or was not provided"
		}

Start index exceeds total record count	400	{
		"message": "Start Index Invalid"
		"details" : "The requested start index exceeds the total number of records " + start_index
		}

3.5.3 Response

Field	Data Type	Data Size	Cardinality	Description
substation_id	Number	15,0	0..*	Substation ID
substation_name	String	255	0..*	Substation Name
request_number	Number		0..*	In-flight Request Number
registration_status_name	Number	(5,0)	0..*	Registration Status
substation_sub_class_name	Number	(5,0)	0..*	Substation Sub-Class
sub_region_name	Number	(5,0)	0..*	Substation Region
electrical_zone_name	String		0..*	Electrical Zone
hydro_one_sector	String	15	0..*	Hydro One Sector
settlement_taxation_area	String	15	0..*	Settlement Taxation Area
gps_north	Number	(15,5)	0..*	Global Positioning System North
gps_west	Number	(15,5)	0..*	Global Positioning System West
connection_type	String	20	0..*	Connection Type
telemetry_perf_class	String	20	0..*	Telemetry Performance Classification
telemetry_size_class	String	20	0..*	Telemetry Size Classification
sps_ras_participation	String	10	0..*	Special Protection system Participant
double_contingency_flag	String	1	0..*	Double Contingency Flag
key_facility	String	1	0..*	Key Restoration Facility
address1	String	50	0..*	Address 1
address2	String	50	0..*	Address 2
address3	String	50	0..*	Address 3
postal_code	String	20	0..*	Postal Code/Zip Code
city	String	20	0..*	City
province	String	2	0..*	Province
country	String	20	0..*	Country
last_update_dt	Date		0..*	Last Updated Date
last_update_user	String	50	0..*	Last Update User ID
start_index	Number	3	0..*	Start Index
batch_size	Number	3	0..*	Batch Size
total_count	Number	3	0..*	Total Count

3.5.4 Sample Request

```
{
" organization_id": 123456
}
```

3.5.5 Sample Response

```
{
  "start_index": 1,
  "batch_size": 50,
  "total_count": 4,
“facilities”: [
          {
          " request_number": 123456,
          " substation_id": 678910,
          " substation_name ": “Test 1”,
          ……
          },
          ……
    ]
}
```

3.6 Retrieve In-Flight Facility Registered Equipment List

40. The participants shall be able to retrieve the list of in-flight equipment information associated with a facility.

41.

Attribute	Details
URI Suffix	/reg-v1-if-equipmentList
HTTP Method GET/POST/PUT	GET

3.6.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization Id
substation_Id	Number	Optional	substation_Id
last_updated_dt_from	Date	Optional	Equipment Last Updated Date From
last_updated_dt_to	Date	Optional	Equipment Last Updated Date To
equipment_type	Number	Optional	When provided, the response will be limited to this type.

3.6.2 Equipment Type Values

Equipment Type	Equipment Type Name
2	Synchronous Machine
3	Wind Turbine Generator
4	PV Inverter
7	Load
14	Circuit Switcher
15	Disconnect Switch
16	Circuit Breaker
17	Shunt Capacitor
18	Shunt Reactor
19	Bus
20	Transformer
21	Circuit
22	Static VAR Compensator (SVC)

Equipment Type	Equipment Type Name
23	STATCOM
24	Collector System
25	Meteorological Tower
26	Nacelle Mounted
28	Sodar Unit
30	Opener
31	Series Capacitor
38	Induction Machine
41	Lidar Unit
42	UFLS Relay Group

3.6.3 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		“message”: “Validation Failed”
		“details”: “organization_id is invalid or was not provided”
		}

substation_Id is invalid	400	{
		“message”: “Validation Failed”
		“details” : “substation_Id is invalid or was not provided”
		}

Substation is not found	404	{
		“message”: “Record Not Found”
		“details” : “Could not find the substation with the input parameters in the Registration system ” + substation_id
		}

Last Updated Date From is invalid	400	{
		“message”: “Validation Failed”
		“details”: “Last Updated Date From is invalid. Last Updated Date From is required if Last Updated Date To is present.”
		}

Last Updated Date To is invalid	400	{

Validation Rule	HTTP Status Code	Examples/Explanations
		“message”: “Validation Failed” “details”: “Last Updated Date To is invalid. Last Updated Date To is required if Last Updated Date From is present.” }

Last Updated Date To is invalid	400	{
		“message”: “Validation Failed”
		“details”: “Last Updated Date To should be greater than Last Updated Date From.”
		}

3.6.4 Response

Field	Data Type	Data Size	Cardinality	Explanations
request_number	Number	15,0	0..*	Request Number
equipment_id	Number	(15,0)	0..*	Equipment Id
equipment_name	String	100	0..*	Equipment Name
equipment_type	Number	(5,0)	0..*	Equipment Type
equipment_type_name	String	50	0..*	Equipment Type Name
operating_nomenclature	String	100	0..*	Operating Nomenclature
voltage_level	String	10	0..*	Voltage Level
registration_status	Number	(5,0)	0..*	Registration Status
action	String	10	0..*	Indicates if the equipment row is addition, update, removal, Existing, Decommission, Add New, Update

3.6.4 Sample Request

```
{
" organization_id": 123456,
" substation_id": 234567
}
```

3.6.5 Sample Response

```
“equipment”: [{
          "request_number" : 678910,
          "equipment_id" : 129839,
          "equipment_name " : "",
         "operating_nomenclature" : "OCT-SM-GAS-STEAM",
         "equipment_type" : 2,
         "equipment_type_name" : "Synchronous Machine",
           ……
         },
…….]
```

3.7 Retrieve Facility In-Flight Equipment Details

42. The participants shall be able to retrieve the in-flight equipment detail information.

43.

Attribute	Details
URI Suffix	/reg-v1-if- equipmentDetails
HTTP Method GET/POST/PUT	GET

3.7.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization ID
equipment_id	Number	Mandatory	Equipment ID

3.7.2 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		"message": "Validation Failed"
		"details": "organization_id is invalid or was not provided"
		}

equipment_id is invalid	400	{
		"message": "Validation Failed"
		"details" : "equipment_id is invalid or was not provided"
		}

equipment is not found	404	{
		"message": "Record Not Found"
		"details" : "Could not find the equipment with the input parameters in the Registration system
		}

3.7.3 Response

Field	Data Type	Data Size	Cardinality	Explanations
equipment_details	**Structure**			JSON object structure
equipment_id	Number	(15,0)	1	Equipment Id
request_number	Number	15,0	1	In-flight request number
operating_nomenclature	String	100	1	Operating Nomenclature
equipment_name	String	100	1	Equipment Name
equipment_type	Number	(5,0)	1	Equipment Type
equipment_type_name	String	50	1	Equipment Type Name
last_update_dt	Date		1	Last Update Date
last_update_user	String	50	1	Last Update User
field_values	**Structure**		0..*	JSON object structure
field_name	String			Field Name
filed_display_name	String			Field Display Name
field_value	String			Field Value
children_equipment	**Structure**		0..*	JSON object structure same as equipment_details
dynamic_model	**Structure**		0..*	JSON object structure
dynamic_model_id	Number			Dynamic Model Id
model_name	String			Dynamic Model Name
description	String			Description
model_type_name	String			Dynamic Model Type

3.7.4 Sample Request

```
{
" organization_id": 123456,
" equipment_Id ": 234567
}
```

3.7.5 Sample Response

```
{
 "total_count" : 1,
 "equipment_details" : [ {
  "equipment_id" : 148297,
  “request_number” :32423,
    "operating_nomenclature" : "GLOBEX COLLECTOR SYSTEM",
    "equipment_name" : "",
    "equipment_type" : 24,
    "equipment_type_name" : "Collector System",
    "last_update_dt" : "2019-06-06T00:00:00Z",
    "last_update_user" : "user1",
    "field_values" : [ {
      "field_name" : "operatingNomenclature",
      "field_display" : "Operating Nomenclature",
      "field_value" : "GLOBEX COLLECTOR SYSTEM"
        }, {
      "field_name" : "voltageLevel",
      "field_display" : "Voltage Level (kV)",
      "field_value" : "230"
       },
    ….
 ],
    "children_equipment" : [ {
      "equipmen_iId" : 148296,
      "operating_nomenclature" : "PV INVERTER 2019",
      "equipment_name" : "",
      "equipment_type" : 4,
      "equipment_type_name" : "PV Inverter",
      "last_update_dt" : "2019-06-06T10:47:01Z",
      "last_update_user" : "user1",
      "field_values" : [ ]
    } ],
    "dynamic_model" : [ {
      "dynamic_model_id" : 148298,
      "model_name" : "REGCAU1",
      "description" : "Renewable Energy Generator/Converter Model",
      "model_type_name" : "Generator"
    } ]
  } ]
}
```

3.8 Retrieve Facility In-Flight Dynamic Model Details

44. The participants shall be able to retrieve the in-flight dynamic model details associated with an equipment.

45.

Attribute	Details
URI Suffix	/reg-v1-if-dynamicModelDetails
HTTP Method GET/POST/PUT	GET

3.8.1 Request Parameters

Parameter	Data Type	Optionality	Explanations
organization_id	Number	Mandatory	Organization ID
equipment_id	Number	Mandatory	Equipment Id

3.8.2 Request Validation Rules

Validation Rule	HTTP Status Code	Examples/Explanations

organization_id is invalid	400	{
		“message”: “Validation Failed”
		“details**”: “**organization_id is invalid or was not provided”
		}

equipment_id is invalid	400	{
		“message”: “Validation Failed”
		“details”: “equipment_id is invalid or was not provided”
		}

equipment_id is not found	404	{
		“message”: “Record Not Found”
		“details”: “Could not find the equipment with the input parameters in the Registration system ” + equipment_id
		}

3.8.3 Response

Field	Data Type	Data Size	Cardinality	Explanations
dynamic_model	**Structure**		0..*	JSON object structure
dynamic_model_id	Number			Dynamic model Id
model_name	String			Dynamic model Name
description	String			Description
model_type_name	String			Dynamic model type name
parameter_values	**Structure**		1..*	JSON object structure
model_parameter_name	String			Parameter Name
model_parameter_value	Number			Parameter Value

3.8.4 Sample Request

```
{
" organization_id": 123456,
" equipment_Id ": 234567
}
```

3.8.5 Sample Response

```
{
  "dynamic_model" : [ {
   "dynamic_model_id" : 148298,
   "model_name" : "REGCAU1",
   "description" : "Renewable Energy Generator/Converter Model",
   "model_type_name" : "Generator",
   "parameter_values" : [ {
     "model_parameter_name" : "Lvplsw (Low Voltage Power Logic) switch",
     "model_parameter_value" : "2"
   }, {
     "model_parameter_name" : "Lvplsw (Low Voltage Power Logic) switch",
     "model_parameter_value" : "1"
   },
   ….
  } ]
 }
```

– End of Document –