



# Simple Ingestion API



GE VERNOVA

# Contents

<b>Chapter 1: Overview</b>	<b>1</b>
Overview	2
<b>Chapter 2: Getting Started</b>	<b>3</b>
Getting Started	4
<b>Chapter 3: Serialization Scheme</b>	<b>5</b>
Serialization Scheme	6
<b>Chapter 4: Base URL</b>	<b>7</b>
BaseURL	8
<b>Chapter 5: Authentication</b>	<b>9</b>
About Authentication	10
POST	10
Request Syntax	10
Request Body	10
Response	10
<b>Chapter 6: Whitelisting</b>	<b>11</b>
Whitelisting	12
<b>Chapter 7: Resources</b>	<b>13</b>
Create/Update an Entity	14
Create/Update a Relationship	16
<b>Chapter 8: Reference</b>	<b>20</b>
Field Definitions	21

Limitations	21
Customer Support	22

# Copyright Digital, part of GE Vernova

© 2024 GE Vernova and/or its affiliates. All rights reserved.

GE, the GE Monogram, and Predix are trademarks of General Electric Company used under trademark license.

This document may contain Confidential/Proprietary information of GE Vernova and/or its affiliates. Distribution or reproduction is prohibited without permission.

THIS DOCUMENT AND ITS CONTENTS ARE PROVIDED "AS IS," WITH NO REPRESENTATION OR WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF DESIGN, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. ALL OTHER LIABILITY ARISING FROM RELIANCE UPON ANY INFORMATION CONTAINED HEREIN IS EXPRESSLY DISCLAIMED.

Access to and use of the software described in this document is conditioned on acceptance of the End User License Agreement and compliance with its terms.

# Chapter 1

---

## Overview

### Topics:

- [Overview](#)

## Overview

This API provides option to ingest small number of records into the APM Families. This is a synchronous call and will not return until the transaction is completed or rolled back.

# Chapter 2

---

## Getting Started

### Topics:

- [Getting Started](#)

## Getting Started

This API is subject to all requirements and constraints of the existing WebApi endpoints, specifically, authentication, authorization, localization, globalization, and serialization.

- Both the records/entities and relationships/connections can be created or updated through the API
- The payload consists of two high level categories:
  - Recipe: This contains Plan sections that define the `PrimaryPlan`, `PredecessorPlan`, and `SuccessorPlan`.
  - Rows: This contains records that are to be ingested.

### Recipe

Provide the following information in the recipe section:

- `PrimaryPlan`: Defines the family or relationship for the primary records and the associated action.
- `PredecessorPlan`: Defines the predecessor family details of the relationship and the associated action.
- `SuccessorPlan`: Defines the successor family details of the relationship and the associated action.

The following items are defined for each plan:

- `Id`: Family or Relationship name
- `Action`: `ACTION_LOCATE`, `ACTION_INSERTUPDATE`. For more information, refer to the section, [About the Configuration Worksheet](#), in the Data Loaders documentation.
- `Genealogy`: Primary, Predecessor, or Successor
- `FamilyType`: Entity or Relationship
- `GedaTypeName`: This field will be blank for the on-premise application.
- `KeyFields`: Lists the family fields that need to be considered as the key fields for the insert or update. For inserting relationship rows, this field will be empty for the `PrimaryPlan`.

### Rows

In the row section, provide the data for ingestion. For data upload to a Family, the values need to be provided for each field. For the relationship, the fields need to be prefixed by either "PRED|" or "SUCC|" as a qualifier.

The response will be the Bundle ID of the ingestion along with additional information.



# Chapter 3

---

## Serialization Scheme

Topics:

- [Serialization Scheme](#)

## Serialization Scheme

Javascript Object Notation (JSON) is the only serialization scheme supported.

The Javascript language does not fully support signed 64-bit integers at the upper and lower value limits. DTO properties defined as type object that have a signed 64-bit integer value are deserialized as string. For example, `RowsetDTO.RowValueDTO.ColumnValue`.

For JSON serialization:

- Numeric and Boolean values are unquoted.
- Strings are quoted.
- Dates are quoted and are assumed to be in the time zone of the API authenticated user if not explicitly qualified as a field.
- Dates must be in the `yyyy-MM-ddTHH:mm:ss.fff` format. In this format, the following literal values are required:
  - Hyphen (-)
  - Colon (:)

# Chapter 4

---

## Base URL

Topics:

- [BaseURL](#)

## BaseURL

`/v1/simpleIngest`

# Chapter 5

---

## Authentication

### Topics:

- [About Authentication](#)
- [POST](#)
- [Request Syntax](#)
- [Request Body](#)
- [Response](#)

## About Authentication

To request for a user token, a POST call to the APM Web API URL with the following URL extension and body must be used. After the Meridium token is retrieved, it is passed as the value to the header key `MeridiumToken`.

## POST

Returns a `SessionDTO` object. From this DTO, `sessionId`, ; (semicolon), and `timezoneId` are concatenated to produce a Meridium Token value.

You must provide a request header, `meridiumtoken`, with this value on every subsequent API call.

The header `content-type` with value `application/json` is the only other header required by the application.

## Request Syntax

```
{{APM_WEB_API_URL}}/api/v1/core/security/login
```

## Request Body

```
{"DataSourceId":"APM_DATASOURCE_NAME","Id":"APM User","Password":"APM Password"}
```

## Response

Retrieve the Meridium Token from the returned headers

# Chapter 6

---

## Whitelisting

### Topics:

- [Whitelisting](#)

## **Whitelisting**

Whitelisting is optional for the endpoint URL FQDN or IP addresses.



# Chapter 7

---

## Resources

### Topics:

- [Create/Update an Entity](#)
- [Create/Update a Relationship](#)

## Create/Update an Entity

### POST

Creates an entity.

#### Consumes

This API call consumes the `application/json` media type via the Content-Type request header.

### Request Syntax

```
POST /v1/simpleIngest
```

### Request Body

JSON

### Usage Sample

```
{
  "Description": "Test Simple Ingestion",
  "Recipe":
  {
    "PrimaryPlan":
    {
      "KeyFieldIds": [
        "MI_EQUIP000_EQUIP_ID_C"
      ],
      "Id": "MI_EQUIP000",
      "Action": "ACTION_INSERTUPDATE",
      "Genealogy": "Primary",
      "FamilyType": "Entity",
      "GedaTypeName": "UNIFIED_ASSET__assettype"
    }
  },
  "Rows": [
  {
```

```

        "MI_EQUIP000_EQUIP_ID_C" : "PWR-APM-ASSET99",
        "MI_EQUIP000_SAP_SYSTEM_C" : "PRF-800",
        "MI_EQUIP000_UNIQUE_ID_C" :
"PRF-800-PWR-APM-ASSET99",
        "MI_EQUIP000_MAINT_PLANT_C" :
"PRF-800",
        "MI_SITE_NAME" : "PRF-800"
    ]]
}

```

## Return Type

200 OK

## Produces

This API call produces the `application/json` media type according to the `Accept` request header. The media type will be conveyed by the `Content-Type` response header.

## Success Response

```

{
  "bundle": {
    "id": "9d1568a0-3147-45a3-ba93-e50fc2f05411",
    "userId": "MIADMIN", "dataLoaderId": "FamilyDataLoader", "origin":
"SimpleIngestion",
    "created": "2024-02-21T22:57:32.713",
    "expires": "2024-02-23T22:57:32.713",
    "status": "Completed",
    "description": "Test Simple Ingestion", "recipe": {CJ},
    "sliceCount": 1,
    "latestTimeStamp": "2024-02-21T22:57:33.210",
    "insertedRowCount": 0,
    "updatedRowCount": 1,
    "deletedRowCount": 0,
    "rejectedRm.Count": 0, "skipSynchronization":false, "manualRun": 1 e,
    "progressPercentage": 100
  },
  "rejectedRows": [], "warningR011s": []
}

```

**Table 1: Explanation of the response properties**

Property	Description
id	Bundle Id which is represented by a GUID
userId	User that ingested the data
dataLoaderId	Data Loader used to ingest the data

Property	Description
origin	SimpleIngestion
created	DateTime of when the Bundle was created
expires	DateTime of when the Bundle will be removed from the database
status	(Started, Failed, Completed, CompletedWithRejects, CompletedWithWarnings)
description	Description provided by the user
recipe	Compilation of ingestion instructions received in the configuration plans
sliceCount	Number of Slices in the Bundle
latestTimeStamp	DateTime of the latest update
insertedRowCount	Number of rows inserted
updatedRowCount	Number of rows updated
deletedRowCount	Number of rows deleted
rejectedRowCount	Number of rows rejected
skipSynchronization	Skip the synchronization step
manualRun	Data ingested manually
progressPercentage	Bundle ingestion progress expressed as a percentage
rejectedRows	List of rejected rows
warningRows	List of warning rows

## Create/Update a Relationship

### POST

Creates or updates a relationship.

### Consumes

This API call consumes the `application/json` media types via the Content-Type request header.

### Request Syntax

```
POST /v1/simpleIngest
```

### Request Body

JSON

### Request Example

```
{
  "Description": "Test Simple Ingestion Relationship",
```

```

"Recipe":
{
  "PrimaryPlan":
  {
    "KeyFieldIds": [
      ],
    "Id": "MIR_EQHSEQ",
    "Action": "ACTION_INSERTUPDATE",
    "Genealogy": "Primary",
    "FamilyType": "Relationship",
    "GedaTypeName": "assettype_assettype"
  },
  "PredecessorPlan":
  {
    "KeyFieldIds": [
      "MI_EQUIP000_EQUIP_ID_C",
      "MI_EQUIP000_SAP_SYSTEM_C"
    ],
    "Id": "MI_EQUIP000",
    "GedaTypeName" : "UNIFIED_ASSET__assettype",
    "Action": "ACTION_INSERTUPDATE",
    "Genealogy": "Predecessor",
    "FamilyType": "Entity"
  },
  "SuccessorPlan":
  {
    "KeyFieldIds": [
      "MI_EQUIP000_EQUIP_ID_C",
      "MI_EQUIP000_SAP_SYSTEM_C"
    ],
    "Id": "MI_EQUIP000",

```

```

        "GedaTypeName" : "UNIFIED_ASSET__assettype",
        "Action": "ACTION_INSERTUPDATE",
        "Genealogy": "Successor",
        "FamilyType": "Entity"
    }
},
"Rows": [
{
    "PRED|MI_EQUIP000_EQUIP_ID_C": "PWR-APM-ASSET97",
    "PRED|MI_EQUIP000_SAP_SYSTEM_C": "PRF-800",
    "PRED|MI_EQUIP000_UNIQUE_ID_C": "PRF-800-PWR-APM-ASSET97",
    "SUCC|MI_EQUIP000_EQUIP_ID_C": "PWR-APM-ASSET99",
    "SUCC|MI_EQUIP000_SAP_SYSTEM_C": "PRF-800",
    "SUCC|MI_EQUIP000_UNIQUE_ID_C": "PRF-800-PWR-APM-ASSET99"
}
]
}

```

## Return Type

200 OK

## Produces

This API call produces the application/json media type according to the Accept request header. The media type will be conveyed by the Content-Type response header.

## Success Response

```

"bundle": {
  "id": "827782f7-2867-4176-8a38-52c44b166340", "userId": "MIADMIN",
  "dataloaderId": "FamilyDataloader", "origin": "Simpleingestion",
  "created": "2024-02-21T22:01:48.782",
  "expires": "2024-02-23T22:01:48.782",
  "status": "Completed",
  "description": "Test Simple Ingestion Relationship", "recipe": { },
  "sliceCount": 1,
  "latestTimeStamp": "2024-02-21T22:01:49.820",
  "insertedR01-1Count": 1,
  "updatedRowCount": 0,
  "deletedRowCount": 0,

```

```

"rejectedR01,lCount": 0, "skipSynchronization": false, "manualRun":
"progressPercentage": 100
},
"rejectedRm,is": [],
"warningR01-ls": []

```

**Table 2: Explanation of the response properties**

Property	Description
id	Bundle Id which is represented by a GUID
userId	User that ingested the data
dataLoaderId	Data Loader used to ingest the data
origin	SimpleIngestion
created	DateTime of when the Bundle was created
expires	DateTime of when the Bundle will be removed from the database
status	(Started, Failed, Completed, CompletedWithRejects, CompletedWithWarnings)
description	Description provided by the user
recipe	Compilation of ingestion instructions received in the configuration plans
sliceCount	Number of Slices in the Bundle
latestTimeStamp	DateTime of the latest update
insertedRowCount	Number of rows inserted
updatedRowCount	Number of rows updated
deletedRowCount	Number of rows deleted
rejectedRowCount	Number of rows rejected
skipSynchronization	Skip the synchronization step
manualRun	Data ingested manually
progressPercentage	Bundle ingestion progress expressed as a percentage
rejectedRows	List of rejected rows
warningRows	List of warning rows

# Chapter 8

---

## Reference

### Topics:

- [Field Definitions](#)
- [Limitations](#)
- [Customer Support](#)



## Field Definitions

### Field Definitions

Arguments/ Elements	Type	Format	Description	Required	Notes
Id	String	N/A	Family/Relationship	Required	The name of the Family or Relationship needs to be provided.
Action	String	N/A	Action on the ingestion	Required	ACTION_LOCATE (to be used for relationship families), ACTION_INSERTUP DATE
Genealogy	String	N/A	Type of the Plan	Required	Primary, Predecessor or Successor
FamilyType	String	N/A	Type of the Family	Required	Entity or Relationship
GedaTypeName	String	N/A	Classification for Geda facing Families	Optional	Classification required for GEDA facing families. The default value is the standard Asset/ Segment Classification
KeyFieldIDs	Array	N/A	FamilyFields	Required	Family field names that are to be considered as Key for the Ingestion separated by

## Limitations

### Limitations

The API currently supports only a few records (~50 records) per request. Consideration should also be taken for size of the content.

## **Customer Support**

### **Customer Support**

To contact support, refer <http://www.ge.com/digital/support>.