GE VERNOVA

Asset Performance Management
On-Premises APM
V5.0.7.0.0

CustDev API

Contents

Copyright Digital, part of GE Vernova

Chapter

1

Overview

Topics:

- Overview

Overview

This API provides customer developers a transaction-based interface to access the Core families. This API supports the following families:

- Equipment
- Equipment Has Equipment
- Functional Location
- Functional Location Has Equipment
- Has Event Detail
- Has Work History
- Work History
- Work History Detail

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.

- Family Key is required for each call.
- Data Transfer Objects (DTOs) are required content for Insert (`POST`) and Update (`PUT`) API calls. They are returned for the Retrieve (`GET`) calls. You must maintain an array of Family DTOs to provide referential data for each call to the API.
- This API provides option to retrieve. The API does not support search. You can, however, use the Query execution.
- Entity Id is not required to be unique in APM. When you retrieve by the Entity Id, an array is returned.
- Entity Key or Relationship Key must be zero for Insert.
- You must retrieve an entity before updating the entity.

    ◦ Fields on relationships were obsoleted prior to 4.0. You can update a relationship only if you want to change legacy relationships which have fields.
    ◦ Entity Key or Relationship Key cannot be zero for Update.

- If you delete an entity, any relationship the entity has is also deleted. The related entities, however, do not change.

    ◦ Deleting a relationship does not change the related entities.
    ◦ Entity Key or Relationship Key cannot be zero for Delete.

- PUT and POST methods will always have a required request content, a DTO, depending on whether it is an entity or relationship.
- The HTTP DELETE verb supports a request content, but this API does not.

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 logged in user.
- Dates must be in the `yyyy-MM-ddTHH:mm:ss.fff` format. In this format, the following literal values and required:
  - Hyphen (-)
  - T
  - Colon (:)

Chapter
4

Base URL

Topics:

- BaseURL

BaseURL

```
api/custdev/core/
```

Chapter
5

Security

Topics:

- Authentication

Authentication

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

```
/core/security/login
```

Request Body

```
{"DatasourceId":"V4050100_QA","Id":"miadmin","Password":"Usability"}
```

Chapter
6

Resources

Topics:

Retrieve all Metadata Entity Families

GET

Retrieves all metadata entity families.

Request Syntax

```
GET/custdev/core/family/ent
```

Return Type

```
array[Meridium.Api.CustDev.Core.FmlyDevDTO]
```

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

200

Response Example

```
[
{
"key":"64263570621",
"id":"AAA_HEALTH_TEST_READING",
"fields":
[{
"key":"64263570623",
"id":"AAA_HEALTH_T_READING_VALUE_C"
},{
"key":"64263570625",
"id":"AAA_HEALTH_T_READING_VALUE_N"
},{
"key":"64263570627",
"id":"AAA_HEALTH_T_TIMESTAMP_D"
},{
"key":"64263570629",
"id":"AAA_HEALTH_T_NOTES"
}]
}
]
```

Retrieve all Metadata Relationship Families

GET

Retrieves all metadata relationship families.

Request Syntax

```
GET/custdev/core/relationship/{relFmlyKey}/pred/{predEntKey}/
{predFmlyKey}/succ/{succEntKey}/{succFmlyKey}
```

Return Type

```
array[Meridium.Api.CustDev.Core.RelDevDTO]
```

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

```
200
```

```
Success
```

Response Example

```
{
    "key":"64263626272",
    "familyKey":"64251800815",
    "familyId":"MIR_FLHSEQ",
    "predFamilyKey":"64251789393",
    "predEntityKey":"64263626261",
    "succFamilyKey":"64251784124",
    "succEntityKey":"64263626258",
    "updateDate":"2021-07-12T14:30:11.900"
}
```

Retrieve an Entity

Request Syntax

```
GET/custdev/core/entity/{entKey}/{entFmlyKey}
```

Return Type

```
Meridium.Api.CustDev.Core.EntDevDTO
```

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

```
200
```

```
Success Meridium.Api.CustDev.Core.EntDevDTO
```

Response Example

```
{
  "key": "64251821844",
  "id": " ~ SLACK WAX HEATER ~ HXST 1",
  "seqNum": "4",
  "familyKey": "64251784124",
  "familyId": "MI_EQUIP000",
  "updateDate": "2017-05-25T16:56:06.000",
  "fields": [
    {
      "key": "64251784126",
      "id": "MI_EQUIP000_EQUIP_ID_C",
      "value": "HXST 1"
    }
  ]
}
```

Retrieve an Array

GET

Retrieves an array of entities by Id.

Request Syntax

```
GET/custdev/core/entity/id/{entId}/{entFmlyKey}
```

Return Type

```
array[Meridium.Api.CustDev.Core.EntDevDTO]
```

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

```
200
```

```
Success
```

Response Example

```
[
  {
    "key": "64251821844",
    "id": " ~ SLACK WAX HEATER ~ HXST 1",
    "seqNum": "4",
    "familyKey": "64251784124",
    "familyId": "MI_EQUIP000",
```

```
      "updateDate": "2017-05-25T16:56:06.000",
      "fields": [
        {
          "key": "64251784126",
          "id": "MI_EQUIP000_EQUIP_ID_C",
          "value": "HXST 1"
        }
```

Create 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/custdev/core/entity
```

Request Body

```
body Meridium.Api.CustDev.Core.EntDevDTO
```

Usage Sample

```
{
    "FamilyKey": "64251784124",
    "Fields": [
        {
            "key": "64251784137",
            "value": "tech number"
        },
        {
            "id": "MI_EQUIP000_EQUIP_SHRT_DESC_C",
            "value": "description"
        },
        {
            "key": "64251784126",
            "id": "MI_EQUIP000_EQUIP_ID_C",
            "value": "equip id"
        }
    ]
}
```

Return Type

```
array[Meridium.Api.CustDev.Core.EntDevDTO]
```

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

```
200

Success
```

Update an Entity

PUT

Updates an entity.

Consumes

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

Request Syntax

```
PUT/custdev/core/entity
```

Request Body

```
body Meridium.Api.CustDev.Core.EntDevDTO(optional)
```

Request Body Example

```
{
    "Key": "64263626247",
    "FamilyKey": "64251784124",
    "SeqNum": 1,
    "Fields": [
        {
            "key": "64251784148",
            "value": "description changed"
        },
        {
            "id": "MI_EQUIP000_EQUIP_ID_C",
            "value": "equip id changed"
        }
    ]
}
```

Return Type

None

Success Response

```
204
Success
```

Delete an Entity

DELETE

Deletes an entity.

Request Syntax

```
DELETE/custdev/core/entity/{entKey}/{entFmlyKey}
```

Success Response

```
204
Success
```

Retrieve an Array of Relationships

GET

Retrieves an array of relationships by relationship family key and either predecessor or successor entity key.

Request Syntax

```
GET/custdev/core/relationships/{relFmlyKey}/pred/{predFmlyKey}/succ/
{succFmlyKey}/ent/{entKey}
```

Return Type

```
array[Meridium.Api.CustDev.Core.RelDevDTO]
```

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

```
204
Success
```

Response Example

```
[{
"key":"64263626272",
"familyKey":"64251800815",
"familyId":"MIR_FLHSEQ",
"predFamilyKey":"64251789393",
"predEntityKey":"64263626261",
"succFamilyKey":"64251784124",
"succEntityKey":"64263626258",
"updateDate":"2021-07-12T14:30:11.900"
}]
```

Retrieve a Relationship

GET

Retrieves a relationship by relationship family key, predecessor entity and family keys, successor entity and family keys.

Request Syntax

```
GET/custdev/core/relationship/{relFmlyKey}/pred/{predEntKey}/
{predFmlyKey}/succ/{succEntKey}/{succFmlyKey}
```

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

```
200
```

```
Success
```

Response Example

```
{
    "key":"64263626272",
    "familyKey":"64251800815",
    "familyId":"MIR_FLHSEQ",
    "predFamilyKey":"64251789393",
    "predEntityKey":"64263626261",
    "succFamilyKey":"64251784124",
    "succEntityKey":"64263626258",
    "updateDate":"2021-07-12T14:30:11.900"
}
```

Create a Relationship

POST

Creates a relationship.

Consumes

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

Request Syntax

`POST/custdev/core/relationship`

Request Body

```
body Meridium.Api.CustDev.Core.RelDevDTO(optional)
```

Request Body Example

```
{
    "familyKey": "64251800815",
    "predFamilyKey": "64251789393",
    "predEntityKey": "64263626261",
    "succFamilyKey": "64251784124",
    "succEntityKey": "64263626258"
}
```

Return Type

`array[Meridium.Api.CustDev.Core.RelDevDTO]`

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

```
200
```

```
Success
```

Delete a Relationship

DELETE

Deletes a relationship.

Request Syntax

```
DELETE /custdev/core/relationship/{relFmlyKey}/pred/{predEntKey}/
{predFmlyKey}/succ/{succEntKey}/{succFmlyKey}
```

Return Type

None

Success Response

```
204

Success
```

Chapter
7

Reference

Topics:

Parameters

Parameters

Name	In	Type	Format	Description	Required	Notes
entId	Path	String	N/A	Entity ID	Required	Entity Id is not required to be unique in APM
entKey	Path	Integer	int64	Entity Key	Required	• Must be zero for Insert • Cannot be zero for Delete or Update
entFmlyKey	Path	Integer	int64	Entity family key	Required	None
relFmlyKey	Path	Integer	int64	Relationship family key	Required	• Must be zero for Insert • Cannot be zero for Delete or Update
predEntKey	Path	Integer	int64	Predecessor entity key	Required	None
predFmlyKey	Path	Integer	int64	Predecessor family key	Required	None
succEntKey	Path	Integer	int64	Successor Entity Key	Required	None
succFmlyKey	Path	Integer	int64	Successor entity key	Required	None

Data Transfer Objects (DTOs)

EntDevDTO

Arguments/ Elements	Type	Format	Description	Required	Notes
Meridium.Api.CustDev.Core.EntDevDTO	object	N/A	Entity DTO	Required	None
Key	Integer/Long	Int64	Entity Key	Optional	Must be zero for insert, non-zero for update and cannot be changed.
id	String	N/A	Entity ID	Optional	Read only, computed value
seqNum	Integer/Long	Int64	Sequence Number	Optional	Required for insert and update. Enables consistency and concurrency for database transactions
familyKey	Integer/Long	Int64	Family Key of the entity	Optional	Required for insert and update.
familyId	String	N/A	Family Id of the entity	Optional	Nullable: true Not required for insert and update
updateDate	String	Date-time	Date and time of last update	Optional	Read only. This value will be in the local time zone of the session user
fields	Array	N/A	Family Fields	Optional	Only those field whose values are to be changed are required.

FldDevDTO

Arguments/Elements	Type	Format	Description	Required	Notes
Meridium.Api.CustDev.Core.EntDevDTO	object	N/A	Entity field	Required	When creating or updating a field using FldDevDTO, either Key or Id are required but both.
Key	Integer	int64	Metadata field key.	Optional	Either Id or Key are required for insert and update
id	String	N/A	Metadata field key.	Optional	Either Id or Key are required for insert and update
value	Object	Date-time	Field value	Optional	Required for insert and update.

FmlyDevDTO

Table 1:

Arguments/Elements	Type	Format	Description	Required	Notes
Meridium.Api.CustDev.Core.FmlyFLDDevDTO	Object	N/A	Metadata family field	Required	None
key	Integer	int64	Metadata family field	Optional	None
id	String	N/A	Metadata family field	Optional	None

FmlyFLDDevDTO

Table 2:

Arguments/ Elements	Type	Format	Description	Required	Notes
Meridium.Api.Cus tDev.Core.FmlyF LDDevDTO	Object	N/A	Metadata family field.	Required	None
key	Integer	int64	Metadata family field.	Optional	None
id	String	N/A	Metadata family field.	Optional	None

RelDevDTO

Arguments/ Elements	Type	Format	Description	Required	Notes
Meridium.Api.Cus tDev.Core.RelDev DTO	Object	N/A	Relationship DTO	Required	A relationship is uniquely identified by FamilyKey, PredEntityKey and SuccEntity key.
Key	Integer/Long	Int64	Relationship key	Optional	Read only, included for legacy purposes only.
familyKey	Integer/Long	Int64	Metadata family key of the relationship	Optional	Required for insert.
familyID	String	N/A	Metadata family Id	Optional	Nullable: true

Not required for insert. |
PredFmlyKey	Integer/Long	Int64	Predecessor entity family key	Optional	Required for insert
PredEntityKey	Integer/Long	Int64	Predecessor entity key	Optional	Required for insert
SuccFmlyKey	Integer/Long	Int64	Successor entity family key	Optional	Required for insert
SuccEntityKey	Integer/Long	Int64	Successor entity key	Optional	Required for insert

Arguments/Elements	Type	Format	Description	Required	Notes
updateDate	String	Date-time	Date and time of last update	Optional	Read only. This value will be in the local time zone of the session user.
fields	Array	N/A	Obsolete	Optional	Included for legacy purposes only. There are no relationship families supported by this API which have fields.

Limitations

Limitations

This API supports the following entity families:

- Equipment
- Functional Location
- Work History
- Work History Details

Customer Support

Customer Support

Contact APM Support Team https://digitalsupport.ge.com/.