

MultiDrive Web API Specification



MultiDrive Web API

Documentation

Release 1.37

EDMI Limited

47 Yishun Industrial Park A

Singapore, 768724

Document History

Date	Synopsis of change	Author	Version
11/07/2013	Added notes on how comms server is assigned for new sites	TP	1.10
31/07/2013	Added MultiDrive permission information	DY	1.20
15/08/2013	Modify Register Data touRate from "T" to "TOTAL"	DY	1.21
19/08/2013	Making sure all IDs are Integer (deviceId, deviceConfigurationId, siteId, deviceGroupId, parentDeviceGroupId, deliveryId, scheduleId, executionId, executionJobId, eventLogId)	DY	1.22
28/08/2013	<ul style="list-style-type: none"> Data Delivery API <ul style="list-style-type: none"> Added deviceGroupName URI to all functions Diagnostic API <ul style="list-style-type: none"> Added plantNumber URI to "Read diagnostic register" and "write diagnostic register" Fleet Management API/Commissioning API <ul style="list-style-type: none"> Added plantNumber URI to "Commission a device in the system", "Decommission a device in the system" Fleet Management API/Device Configurations API <ul style="list-style-type: none"> Added deviceConfigurationName URI to "Get Device Configuration" Fleet Management API/Device Group API <ul style="list-style-type: none"> Added deviceGroupName URI to "Get Device Group", "Update Device Group", "Delete DeviceGroup", "List / Search Devices in Device Group", "Add Devices to Device Group", "Remove Device From Device Group" Fleet Management API/Devices API <ul style="list-style-type: none"> Corrected URI for "create device" Added plantNumber URI to "update device", "delete device", "Last Device Access Time", "Next Schedule Device Access Time" Added limit and offset to "List / Search Devices" Fleet Management API/Sites API <ul style="list-style-type: none"> Added siteCode URI to "Delete Site" Added limit and offset to "List / Search Sites" Measured Data API <ul style="list-style-type: none"> Added plantNumber URI to "GetProfileData", "Get Time of Use Register Data", "Get Event Log Data", "Get Voltage Sag/Swell, THD or Unbalance Event Data", "Rollback data to a specific date/time" Remove deviceId parameter from "GetProfileData", "Get Time of Use Register Data", "Get Event Log Data", "Get Voltage Sag/Swell, THD or Unbalance Event Data", "Rollback data to a specific date/time" Schedule API <ul style="list-style-type: none"> Added limit and offset to "List Schedule Executions", "Search for Schedule Execution Jobs" Added scheduleName URI to "Get Schedule", "Trigger Regular Schedule As Configured", "Trigger Adhoc Schedule for particular devices", "List Schedule Executions", "Search for Schedule Execution Jobs" Remove name parameter from "List Schedule Executions" Supply Control API <ul style="list-style-type: none"> Added deviceGroupName URI to "Get Main Supply status for a device group" 	DY	1.23
11/10/2013	<ul style="list-style-type: none"> Fleet Management API <ul style="list-style-type: none"> Added Modem API and SIM Card API Fleet Management API/Commissioning API <ul style="list-style-type: none"> Added "Commission a SMETS Device in the System", "Request ZigBee Pairing for SMETS Gas Device", "Request ZigBee Pairing for SMETS IHD Device", "Request ZigBee Unpairing for SMETS Device" Added deviceGroupId parameter to "Commission a Device in the System", Decommission a Device in the System" Fleet Management API/Device Groups API <ul style="list-style-type: none"> Added plantNumber payload to "Add Devices to Device Group" Added deviceId and plantNumber payload to "Remove Device from Device Group" Prepayment API <ul style="list-style-type: none"> Modified items in payload for "Write Misc. Prepayment Info to Meter" Added 400 Bad Request return value for "Write Misc. Prepayment Info to Meter", "Request Prepayment Information for a Device" 	DY	1.24
12/10/2013	<ul style="list-style-type: none"> Fleet Management API <ul style="list-style-type: none"> Added Gas Calorific Value API Include Calorific Value to Device Object 	DY	1.25

08/11/2013	<ul style="list-style-type: none"> Fleet Management <ul style="list-style-type: none"> Added "List / Search Gas Device Pairing Events", "List / Search IHD Device Pairing Events" Modified "List / Search Commissioning Events" Data Delivery API <ul style="list-style-type: none"> Modified "Trigger export of existing Load Profile data", "Trigger export of existing Time of Use/Register data", "Trigger export of existing Event Log data" parameters and return value Removed "Trigger export of existing voltage sag/swell data" Added MDX Object - Data Deliveries 	CB	1.26
12/11/2013	<ul style="list-style-type: none"> Fleet Management API <ul style="list-style-type: none"> Added "Request ZigBee Un-pairing for SMETS Gas Device", "Request ZigBee Un-pairing for SMETS IHD Device" 	JC	1.27
16/12/2013	<ul style="list-style-type: none"> Data Delivery API <ul style="list-style-type: none"> Updated all data export requests to return a 202 instead of 200 on success 	AF	1.28
20/05/2014	<ul style="list-style-type: none"> System API <ul style="list-style-type: none"> Added "Decrypt Device Password" 	WT	1.29
27/05/2014	<ul style="list-style-type: none"> Fleet Management API <ul style="list-style-type: none"> Updated device password to be encrypted for mdx files 	AF	1.30
23/07/2014	<ul style="list-style-type: none"> System API <ul style="list-style-type: none"> Added permission requirement for "Decrypt Device Password" 	WT	1.31
04/09/2014	<ul style="list-style-type: none"> Measured Data API <ul style="list-style-type: none"> Added optional parameter "logNumber" in "Get Event Log Data" 	WT	1.32
15/09/2014	<ul style="list-style-type: none"> Fleet Management API <ul style="list-style-type: none"> Updated payload format for "Get Device Configuration" 	WT	1.33
22/09/2014	<ul style="list-style-type: none"> Measured Data API <ul style="list-style-type: none"> Updated "Valid values" for "logNumber" parameter in "Get Event Log Data" Added "billingResetTime" property to "registerDataSets" payload of "Get Time of Use Register Data" Added optional parameters "loadSurveyNumber" and "dataTypeId" in "Get Profile Data" Updated comments on "Data Gaps" under "Get Profile Data" 	WT	1.34
10/11/2014	<ul style="list-style-type: none"> Measured Data API <ul style="list-style-type: none"> Removed "Rollback data to a specific date/time". (http://mantis.edmi.com.au/view.php?id=7812#c30232) 	WT	1.35
05/02/2015	<ul style="list-style-type: none"> Prepayment API <ul style="list-style-type: none"> Updated payload for "Pass prepayment token to meter" to support sending of multiple tokens to device 	WT	1.36
11/02/2015	<ul style="list-style-type: none"> Fleet Management API <ul style="list-style-type: none"> Updated return payload for "Last Device Access Time (Scheduled and Adhoc Communications)" 	WT	1.37

Table of Contents

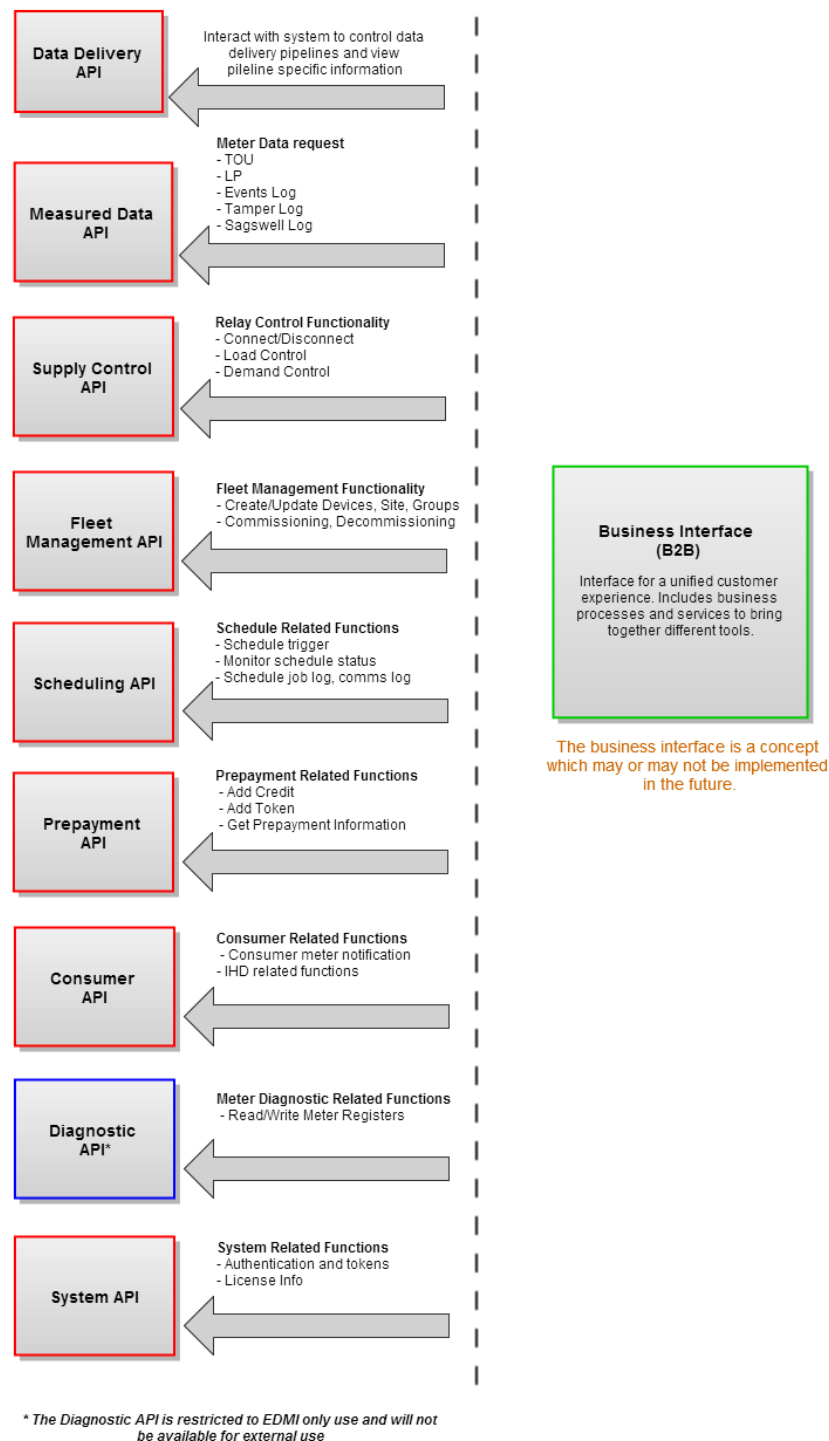
- 1. Overview
 - 1.1. MD Web Service APIs
 - 1.2. REST Background
 - 1.3. API Patterns
- 2. API Versioning
- 3. API Security
- 4. API Success Responses
 - 4.1. 200 OK
 - 4.2. 201 Created
 - 4.3. 202 Accepted
- 5. API Error Responses
 - 5.1. 400 Bad Request
 - 5.2. 401 Unauthorized
 - 5.3. 403 Forbidden
 - 5.4. 404 Not Found
 - 5.5. 405 Method Not Allowed
 - 5.6. 409 Conflict
 - 5.7. 500 Internal Server Error
 - 5.8. 501 Not Implemented
 - 5.9. 503 Service Not Available
- 6. Data Delivery API
- 7. Measured Data API
 - 7.1.1. Get Data Types
 - 7.1.2. Get Profile Data
 - 7.1.3. Data Gaps
 - 7.1.4. Calendar Interval Length Units
 - 7.1.5. Get Time of Use Register Data
 - 7.1.6. Period Types
 - 7.1.7. Supported Register Channels
 - 7.1.8. Get Event Log Data
 - 7.1.9. Get Voltage Sag/Swell, THD or Unbalance Event Data
- 8. Supply Control API
- 9. Fleet Management API
 - 9.1.1. Devices
 - 9.1.2. Protocols
 - 9.1.3. Device Statuses
 - 9.1.4. Meter Series
 - 9.1.5. Sites
 - 9.1.6. List / Search Device Configurations
 - 9.1.7. List / Search Device Groups
 - 9.1.8. Get Device Group
 - 9.1.9. Create Device Group
 - 9.1.10. Update Device Group
 - 9.1.11. Delete DeviceGroup
 - 9.1.12. List / Search Devices in Device Group
 - 9.1.13. Add Devices to Device Group
 - 9.1.14. Remove Device From Device Group
 - 9.1.15. Commission a SMETS device in the system
 - 9.1.16. Request ZigBee pairing for SMETS Gas Device
 - 9.1.17. Request ZigBee pairing for SMETS IHD device
 - 9.1.18. Request ZigBee un-pairing for SMETS Gas Device
 - 9.1.19. Request ZigBee un-pairing for SMETS IHD device
 - 9.1.20. Commission a device in the system
 - 9.1.21. Decommission a device in the system
 - 9.1.22. List / Search Commissioning Events
 - 9.1.23. List / Search Gas Device Pairing Events
 - 9.1.24. List / Search IHD Device Pairing Events
 - 9.1.25. Modem
 - 9.1.26. SIM Cards
 - 9.1.27. Gas Calorific Value
- 10. Scheduling API
 - 10.1.1. Schedules

- 10.1.2. Executions
 - 10.1.3. Execution Jobs
- 11. Prepayment API
- 12. Consumer API
- 13. Diagnostic API
 - 13.1.2. Data types
- 14. System API

1. Overview

1.1. MD Web Service APIs

The MultiDrive Web Service APIs are divided into different components for different roles. APIs are intended to be consistent in design and return responses in the [MDX](#) format.



1.2. REST Background

MDXU Web Service API will be adapting "standard" RESTful web service concept. A RESTful web service (also called a RESTful [web API](#)) is a web service implemented using HTTP and the principles of REST. It is a collection of resources, with four defined aspects:

- the base URI for the web service, such as <http://example.com/resources/>
- the **Internet media type** of the data supported by the web service. This is often **JSON** but can be any other valid Internet media type provided that it is a valid hypertext standard.
- the set of operations supported by the web service using **HTTP methods** (e.g., GET, PUT, POST, or DELETE).
- The API must be hypertext driven.

The following table shows how the HTTP methods are typically used to implement a web service.

Resource	GET	PUT	POST	DELETE
Collection URI, such as http://example.com/resources/	List the URIs and perhaps other details of the collection's members.	Replace the entire collection with another collection.	Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.	Delete the entire collection.
Element URI, such as http://example.com/resources/item17	Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	Replace the addressed member of the collection, or if it doesn't exist, create it	Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it.	Delete the addressed member of the collection.

The PUT and DELETE methods are **idempotent methods**. The GET method is a **safe method** (or nullipotent), meaning that calling it produces no **side-effects**.

Reference:

- <http://www.codeproject.com/Articles/21258/Everything-about-REST-web-services-what-and-how-Pa> (👉)
- <http://rest.elkstein.org/> (👉)
- <http://www.peej.co.uk/articles/restfully-delicious.html>
- http://en.wikipedia.org/wiki/Representational_State_Transfer#RESTful_web_services
- [http://msdn.microsoft.com/en-us/library/system.web.mvc.httpverbs\(v=vs.108\).aspx](http://msdn.microsoft.com/en-us/library/system.web.mvc.httpverbs(v=vs.108).aspx)
- <http://www.thebuzzmedia.com/designing-a-secure-rest-api-without-oauth-authentication/>

1.3. API Patterns

- **TimeZone** - All dates returned will be in UTC. All dates as parameters must be in ISO8601. As per this format, if no time zone is specified then it will interpret dates in the server's time zone.
- **Collection end points** - Most documents will need to be represented within a collection element for listing and searching. The document will often contain a subset of document attributes to reduce the response. Collection results may need to be paginated when there are a large number of results. If the number of documents returned in a collection will always be minimal, the collection document may be sufficient to display the complete contents of all child documents. For ease of use, the collection endpoint may be provided, without providing a separate endpoint for displaying child elements.
- **Update / Create with PUT, Delete with DELETE** - Where the URI is known in advance then creation is performed using PUT. This is convenient when the element has a unique code of reference provided by the client, as the server does not need to pass back a reference to the new document. Modification and deletion is performed with PUT and DELETE respectively to the same endpoint address. When the URI contains a component which is not predefined by the user (e.g. when an ID is included within the URI), then a POST is required, and a 201 Created response is returned along with a reference to the new document.
- **Processing functions with POST** - If processing functions are required an endpoint using POST is required.
- **Including collection elements within documents**. For ease of use, when possible the contents of a collection for a document should be included within it's parent's document. In this case when creating or modifying the document with PUT the contents of the collection can also be included.
- **Modifying big collections with POST**. If a collection could be too large to be included within a parent's document then it is separated into a collection document. To modify elements the user can POST a set of elements to a processing end point to add or remove items. The processing end point could be at the collection documents end point itself.
- **Id** - where present, the data base ID is named `\{x\}Id` where `\{x\}` is the type of object.
- **Including codes / names when referencing objects** - When including the ID of a referenced element, the related name or code is included for ease of the consumer.
- **Creating objects** - Objects are typically created with a PUT to the location of the desired document. For example a site which has a unique site code would be PUT to `/sites/{siteCode}`. The object is passed as JSON, XML or HTML and must correspond to the MDX specification for the object. However when creating and modifying objects there will be attributes which cannot be set, for example IDs and collections of related objects which are returned for ease of use. These attributes are marked as read-only in the API.

2. API Versioning

All APIs will be versioned.

Versioning is handled at the URI level. All resources will have a version number before them and will be indicated as the version number alone. For example:

Measured Data API, loadProfile data, API version 1:

/measuredData/v1/loadProfile

Anything after the /v1/ is considered version 1 of the API.

There is only one version per base resource, even if accessing sub-resources, for instance.

If you are accessing list of devices in a device group:

This is a valid URI:

/fleetManagement/v1/deviceGroups/{deviceId}/devices

This is **not** a valid URI:

/fleetManagement/v2/deviceGroups/{deviceId}/v1/devices

In this case trying to use version 2 of the device groups API and then version 1 of the devices objects **is not valid**. You must accept the version of devices that is used in version 2 of the entire API for device groups.

Versioning will follow these rules:

Changes That Don't Require a New Version

- New resources
- New HTTP methods on existing resources
- New data formats
- New attributes or elements on existing data types

Change That Require a New Version

- Removed or renamed URIs
- Different data returned for same URI
- Removal of support for HTTP methods on existing URIs

Any older version may be marked for deprecation once it becomes unused.

It will be general practice to do this once it becomes two versions out of date (so once get to version 3, mark version 1 as deprecated).

Any APIs marked as deprecated will be removed in the next major release.

3. API Security

The API is secured through the use of SSL and proper certificates to secure the connection and basic HTTP authentication to authorize users.

Basic HTTP authentication is done using the username/password of the user as current stored in MultiDrive. This also provides the access capabilities for the user.

All users must first request an access token from the System Management API. That token then must be passed in with all API calls.

The token is entered as part of the HTTP Authorization header:

Authorization: AuthToken "Token = <token>"

Example: Authorization: AuthToken "Token = c1a0d63e-18b6-46df-bbda-5cb0282cecd8"

All API calls must validate the token in the system before allowing any request.

- Any request for which the token does not exist (or has been deleted) will respond with a "HTTP 401 Unauthorised" response.
- Any request for which the token indicates that the user is not allowed to access will respond with a "HTTP 403 Forbidden" response.

4. API Success Responses

For successful operations, the API can respond with a number of different success responses. These responses and their payloads are detailed below.

4.1. 200 OK

This indicates that the request was completed successfully.

For a GET operation, this will return the object requested.

For a PUT operation, this will return the updated object.

For a DELETE operation, this will return an empty payload.

4.2. 201 Created

This indicates that the request to create a new resource was successful.

The URN/URI to the new resource is indicated in the HTTP header **Location** field.

The payload will be the newly created object.

4.3. 202 Accepted

This indicates that the request has been accepted for processing.

A URN/URI to access the status of the request is indicated in the HTTP header **Location** field.

The payload will contain the current copy of the status (same format as would be requested via the URI given in the location field). This is different depending on the resource.

5. API Error Responses

The API can respond with a number of errors. The format of the responses provided with the errors is detailed below.

5.1. 400 Bad Request

This error indicates that the request was invalid. The errors list will contain details on any element which is invalid.

```
{
  "errorMessage": {
    "status" : "409",
    "dateTime" : "2013-01-21T10:00:00",
    "message" : "The device you try to create already exists",
    "errorCode" : "Device-400",
    "errors" : [
      {
        "message" : "The device already exists",
        "errorCode" : "Device-400",
        "detail" : "fleetManagement/device/123",
      }
    ],
  },
}
```

5.1.1.1. Description

Property			MDX export status	Format	Description
errorMessage					
	status		Mandatory	Text	HTTP status code associated with error
	dateTime		Mandatory	Text	Date and Time of error
	message		Mandatory	Text	User friendly error message
	errorCode		Mandatory	Text	A unique numeric code for the error. See API Error Codes for more info.

	errors				<p>This contains any items which are associated with the error.</p> <p>For example a 409 error would contain a list of the ids which conflict here.</p>
		message	Mandatory	Text	This contains a individual error message
		errorCode	Mandatory	Text	A unique numeric code for the individual error. See API Error Codes for more info.
		detail	Optional	Text	This contains any detail (ie. URI, identifiers etc) associated with the individual error

5.2. 401 Unauthorized

This error indicates that the token provided is indicated or not found. No error message is given.

For operations which request or delete a token, this indicates that the username or password provided via basic authentication was not found or invalid.

5.3. 403 Forbidden

This error indicates that the user is not allowed to access this resource via this operation. No error message is given.

For operations which request or delete a token, this indicates that the username provided has been disabled in the system.

5.4. 404 Not Found

This error indicates that the requested resource was not found. The errors list will contain details on any element which has not been found with the detail value containing the URI of the not found resource.

```
{
  "errorMessage": {
    "status" : "409",
    "dateTime" : "2013-01-21T10:00:00",
    "message" : "The device you try to create already exists",
    "errorCode" : "Device-400",
    "errors" : [
      {
        "message" : "The device already exists",
        "errorCode" : "Device-400",
        "detail" : "fleetManagement/device/123",
      }
    ],
  },
}
```

5.4.1.1. Description

Property			MDX export status	Format	Description
errorMessage					
	status		Mandatory	Text	HTTP status code associated with error
	dateTime		Mandatory	Text	Date and Time of error
	message		Mandatory	Text	User friendly error message
	errorCode		Mandatory	Text	A unique numeric code for the error. See API Error Codes for more info.

	errors				<p>This contains any items which are associated with the error.</p> <p>For example a 409 error would contain a list of the ids which conflict here.</p>
		message	Mandatory	Text	This contains a individual error message
		errorCode	Mandatory	Text	A unique numeric code for the individual error. See API Error Codes for more info.
		detail	Optional	Text	This contains any detail (ie. URI, identifiers etc) associated with the individual error

5.5. 405 Method Not Allowed

This indicates that the resource does not support the HTTP method specified by the request e.g. if you try a PUT on a resource that only supports GETs. No error message is given with this error.

5.6. 409 Conflict

This error indicates that the resource the user is trying to create conflicts with an existing resource. The errors list will contain details on any element which is conflicting with the detail value containing the URI of the conflicting resource.

```
{
  "errorMessage": {
    "status" : "409",
    "dateTime" : "2013-01-21T10:00:00",
    "message" : "The device you try to create already exists",
    "errorCode" : "Device-400",
    "errors" : [
      {
        "message" : "The device already exists",
        "errorCode" : "Device-400",
        "detail" : "fleetManagement/device/123",
      }
    ],
  },
}
```

5.6.1.1. Description

Property			MDX export status	Format	Description
errorMessage					
	status		Mandatory	Text	HTTP status code associated with error
	dateTime		Mandatory	Text	Date and Time of error
	message		Mandatory	Text	User friendly error message
	errorCode		Mandatory	Text	A unique numeric code for the error. See API Error Codes for more info.
	errors				<p>This contains any items which are associated with the error.</p> <p>For example a 409 error would contain a list of the ids which conflict here.</p>
		message	Mandatory	Text	This contains a individual error message
		errorCode	Mandatory	Text	A unique numeric code for the individual error. See API Error Codes for more info.

		detail		Optional	Text	This contains any detail (ie. URI, identifiers etc) associated with the individual error
--	--	--------	--	----------	------	--

5.7. 500 Internal Server Error

This error indicates an internal problem with the system and the request cannot be completed. The error response contains the related error message and any individual errors (if multiple errors occur). The detail may contain developer information.

```
{
  "errorMessage": {
    "status" : "409",
    "dateTime" : "2013-01-21T10:00:00",
    "message" : "The device you try to create already exists",
    "errorCode" : "Device-400",
    "errors" : [
      {
        "message" : "The device already exists",
        "errorCode" : "Device-400",
        "detail" : "fleetManagement/device/123",
      }
    ],
  },
}
```

5.7.1.1. Description

Property				MDX export status	Format	Description
errorMessage						
	status			Mandatory	Text	HTTP status code associated with error
	dateTime			Mandatory	Text	Date and Time of error
	message			Mandatory	Text	User friendly error message
	errorCode			Mandatory	Text	A unique numeric code for the error. See API Error Codes for more info.
	errors					This contains any items which are associated with the error. For example a 409 error would contain a list of the ids which conflict here.
		message		Mandatory	Text	This contains a individual error message
		errorCode		Mandatory	Text	A unique numeric code for the individual error. See API Error Codes for more info.
		detail		Optional	Text	This contains any detail (ie. URI, identifiers etc) associated with the individual error

5.8. 501 Not Implemented

This error indicates that API called is not implemented with the current version of MultiDrive deployed. The error response contains the related error message and any individual errors (if multiple errors occur). The detail may contain developer information.

```
{
  "errorMessage": {
    "status" : "409",
    "dateTime" : "2013-01-21T10:00:00",
    "message" : "The device you try to create already exists",
    "errorCode" : "Device-400",
    "errors" : [
      {
        "message" : "The device already exists",
        "errorCode" : "Device-400",
        "detail" : "fleetManagement/device/123",
      }
    ],
  },
}
```

5.8.1.1. Description

Property			MDX export status	Format	Description
errorMessage					
	status		Mandatory	Text	HTTP status code associated with error
	dateTime		Mandatory	Text	Date and Time of error
	message		Mandatory	Text	User friendly error message
	errorCode		Mandatory	Text	A unique numeric code for the error. See API Error Codes for more info.
	errors				This contains any items which are associated with the error. For example a 409 error would contain a list of the ids which conflict here.
		message	Mandatory	Text	This contains a individual error message
		errorCode	Mandatory	Text	A unique numeric code for the individual error. See API Error Codes for more info.
		detail	Optional	Text	This contains any detail (ie. URI, identifiers etc) associated with the individual error

5.9. 503 Service Not Available

This error indicates a timeout has occurred. No error message is given with this error.

A retry time may be indicated in the Retry-After field in the HTTP header.

6. Data Delivery API

6.1.1.1. Trigger export of any available data

URL	/dataDelivery/all/{deviceId} /dataDelivery/all/plantNumber/{plantNumber} /dataDelivery/all/deviceGroup/{deviceGroupId} /dataDelivery/all/deviceGroup/name/{deviceGroupName}
Method	POST

Permission	Permission checking will vary depending on the dataType specified: LOADPROFILE : mdpProfileData [View] TOU : mdpTOUData [View] EVENTLOG : mdpDeviceEventLog [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	startDate	Inclusive start date of load profile data to export	date	Mandatory		
	endDate	Inclusive end date of load profile data to export	date	Mandatory		
	dataType	The types of data to export, values seperated by commas. If not specified, then any data from the list is exported	string	Optional	LOAD_PROFILE, TOU, EVENT_LOG	*
Returns	<i>202 Accepted</i> <i>401 Unauthorized</i> (Token invalid) <i>403 Forbidden</i> (Access not allowed) <i>404 Not Found</i> (Device Id, plant number or device group Id does not exist) <i>500 Internal Server Error</i>					

```

{
  "dataDeliveries" : [
    {
      "deviceId" : 1,
      "deliveryId" : 8867,
      "dataType" : "LOAD_PROFILE",
      "data": "LOAD_PROFILE1"
    },
    {
      "deviceId" : 2,
      "deliveryId" : 8868,
      "dataType" : "TOU",
      "data": "BILLING_TOTAL"
    },
    {
      "deviceId" : 3,
      "dataType" : "LOAD_PROFILE",
      "data": "LOAD_PROFILE2",
      "errorMessage":
        {
          "status":"400",
          "dateTime":"2013-11-08T08:18:32.6348671Z",
          "message":"No associated data mapping for selected Load Profiles",
          "errorCode":"DataDelivery-012"
        }
    },
    {
      "deviceId" : 4,
      "dataType": "EVENT_LOG",
      "data" : "TAMPER",
      "errorMessage":
        {
          "status":"400",
          "dateTime":"2013-11-08T08:24:01.796Z",
          "message":"Selected Event Type is not supported",
          "errorCode":"DataDelivery-009"
        }
    }
  ]
}

```

6.1.1.2. Trigger export of existing load profile data

URL	/dataDelivery/loadProfile/{deviceId} /dataDelivery/loadProfile/plantNumber/{plantNumber} /dataDelivery/loadProfile/deviceGroup/{deviceGroupId} /dataDelivery/loadProfile/deviceGroup/name/{deviceGroupName}					
Method	POST					
Permission	mdpProfileData [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	startDate	Inclusive start date of load profile data to export	date	Mandatory		
	endDate	Inclusive end date of load profile data to export	date	Mandatory		

	dataType	The type of data to export	string	Optional		*
	profileNumber	Load Profile(s) to export, values separated by commas	string	Optional	SPECIAL_PROFILE, LOAD_PROFILE1, LOAD_PROFILE2 ... LOAD_PROFILE10	
Returns	<p>202 Accepted - same as in Export of Any Available Data operation</p> <p>400 Bad Request (no associated data mapping for selected Load Profiles, more than 1 associated data mapping for selected Load Profiles, no associated manual mapping, no Quantity Channels for selected Load Profiles, no devices under the selected group)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number or device group Id does not exist)</p> <p>500 Internal Server Error</p>					

6.1.1.3. Trigger export of existing Time of Use/Register data

URL	/dataDelivery/timeOfUse/{deviceId} /dataDelivery/timeOfUse/plantNumber/{plantNumber} /dataDelivery/timeOfUse/deviceGroup/{deviceGroupId} /dataDelivery/timeOfUse/deviceGroup/name/{deviceGroupName}					
Method	POST					
Permission	mdpTOUData [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	startDate	Inclusive start date of load profile data to export	date	Mandatory		
	endDate	Inclusive end date of load profile data to export	date	Mandatory		
	dataType	The type of data to export	string	Optional		*
	periodType	The type of Time of Use register to fetch, values separated by commas	string	Optional	TOTAL, BILLING_TOTAL, CURRENT, PREVIOUS_1, PREVIOUS_2 ... PREVIOUS_13	*
Returns	<p>202 Accepted - same as in Export of Any Available Data operation</p> <p>400 Bad Request (selected Period Type is not supported, no associated data mapping for selected TOU, no associated manual mapping, no Quantity Channels for selected TOU, no devices under the selected group)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number or device group Id does not exist)</p> <p>500 Internal Server Error</p>					

6.1.1.4. Trigger export of existing event log data

URL	/dataDelivery/eventLog/{deviceId} /dataDelivery/eventLog/plantNumber/{plantNumber} /dataDelivery/eventLog/deviceGroup/{deviceGroupId} /dataDelivery/eventLog/deviceGroup/name/{deviceGroupName}					
-----	--	--	--	--	--	--

Method	POST					
Permission	Permission checking will vary depending on the eventType specified: mdpDeviceEventLog [View] SAGSWELL : mdpSagSwellLog [View] TAMPER : mdpDeviceTamperLog [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	startDate	Inclusive start date of load profile data to export	date	Mandatory		
	endDate	Inclusive end date of load profile data to export	date	Mandatory		
	eventType	The type of event to export	string	Optional	SYSTEM, ACCESS, TAMPER, SAG_SWELL, ALARM	
Returns	202 Accepted - same as in Export of Any Available Data operation 400 Bad Request (selected Event Type is not supported, no devices under the selected group) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Device Id, plant number or device group Id does not exist) 500 Internal Server Error					

6.1.1.5. Get data delivery status

URL	/dataDelivery/status/{deliveryId}
Method	GET
Permission	NA

Returns

200 OK

401 Unauthorized (Token invalid)

403 Forbidden (Access not allowed)

404 Not Found (Delivery Id does not exist)

500 Internal Server Error

```
{
  "dataDeliveryStatus" :
  {
    "deliveryId" : 14,
    "createTime" : "2013-01-21T10:00:00",
    "statusTime" : "2013-01-21T10:00:00",
    "priority" : 1,
    "status" : "Successful",
    "log" : "Did some stuff\nWent better than expected",
  },
}
```

6.1.1.6. Description

Property		Format	Description
dataDeliveryStatus			
	deliveryId	Integer	A unique identifier for the data delivery (Map to tblProcessJobs.JobId)
	createTime	ISO8601	Date/Time job was created
	statusTime	ISO8601	Date/Time of last status update
	priority	Integer	The current priority of the job
	status	String	Current status of the job. Valid statuses are: <ul style="list-style-type: none"> • Unassigned • Assigned • Processing • Successful • Failed • Aborted
	log	String	Log string

7. Measured Data API

7.1.1. Get Data Types						
URL	/measuredData/dataTypes					
Method	GET					
Permission	None					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	name	Filter by a match in the data type name	string	Optional		-

Returns

200 OK

400 Bad Request (Hash is not valid or timestamp is too old)

401 Unauthorized (Username/password is invalid)

403 Forbidden (Username has been disabled)

500 Internal Server Error

Returns

```
{
  "dataTypes" : [
    {
      "bitRepresentation" : 4112,
      "name" : "Import Wh Phase A",
      "units" : "Wh",
      "flowDirection" : "Consumption",
      "phase" : "A"
      "apportionPolicy" : "SUM" ,
      "decimalPlaces" : 3,
      "multiplier" : 1000
    }, ...
  ]
}
```

Property	Status	Format	Description
bitRepresentation	Mandatory	Integer	Integer for binary representation of data type
name	Mandatory	Text ()	Name of the data type
units	Mandatory	See appendix C	Unit of measure.
flowDirection	Mandatory	Text ()	Flow direction, e.g. "Consumption", "Absolute" or "Q1" Flow direction must be valid for UOM. See Appendix C.
phase	Mandatory	See Appendix E	Phase, e.g. "A" or "AB".
apportionPolicy	Mandatory	See Appendix F	The rule to follow when apportioning (for example, when viewing 15-minute day apportioned to 1-day intervals).
decimalPlaces	Mandatory		
multiplier	Mandatory		

7.1.2. Get Profile Data

URL	/measuredData/loadProfile/{deviceId} /measuredData/loadProfile/plantNumber/{plantNumber}					
Method	GET					
Permission	mdpProfileData [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default

	startDate	The date to start the data period	string	Mandatory		-
	endDate	The date to end the data period (max of 1 month from startDate)	string	Mandatory		-
	limit	Maximum number of records to return status for in response	Integer	Optional	0 - 5000	5000
	offset	Offset in devices to start returning records from	Integer	Optional	*	0
	loadSurveyNumber	Identifier of load survey from the channels of which the profile data is retrieved (0 = all load surveys)	Integer	Optional	*	0
	dataTypeId	Identifier of data type of the channel from which the profile data is retrieved	Integer	Optional	*	-
<p>Notes:</p> <ol style="list-style-type: none"> 1. If both loadSurveyNumber and dataTypeId are specified, data will only be returned if a channel of the specified data type is defined in the specified load survey. 2. If only dataTypeId is specified, data will be returned for a channel of the specified data type as long as it is defined in any of the load surveys. 3. If only loadSurveyNumber is specified, data will be returned for all the channels defined in the specified load survey. 						
Returns	<p>200 OK</p> <p>400 Bad Request (length of data is invalid or too long)</p> <p>401 Unauthorised (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id Does not exist)</p>					

MDX Export Example

```
...
<intervalDataSets>
  <intervalDataSet>
    <code>LS1</code>
    <description>Continuous read load survey</description>
    <intervalLength>30</intervalLength>
    <intervalLengthUnits>MINUTE</intervalLengthUnits>
    <channels>
      <channel>
        <units>WH</units>
        <flowDirection>DELIVERED</flowDirection>
        <phase>NONE</phase>
        <apportionPolicy>SUM</apportionPolicy>
      </channel>
      <channel>
        <units>VARH</units>
        <flowDirection>RECEIVED</flowDirection>
        <phase>NONE</phase>
        <apportionPolicy>SUM</apportionPolicy>
      </channel>
      <channel>
        <units>VA</units>
        <flowDirection>ABSOLUTE</flowDirection>
        <phase>A</phase>
        <apportionPolicy>AVG</apportionPolicy>
      </channel>
    </channels>
    <intervals>
      <interval>
        <endOfInterval>2012-06-10T12:30:00.000+0000</endOfInterval>
        <values>
          <value>1319.21</value>
          <value>230.58</value>
          <value>1319.21</value>
        </values>
      </interval>
      <interval>
        <endOfInterval>2012-06-10T13:00:00.000+0000</endOfInterval>
        <values>
          <value>53.92</value>
          <value>12.10</value>
          <value>53.92</value>
        </values>
      <statuses><status>PL</status><status>II</status></statuses> //
Power loss (PL) - interval marked as incomplete (II)
    </interval>
    <interval>
      <endOfInterval>2012-06-10T13:30:00.000+0000</endOfInterval>
      <values>
        <value></value>
        <value></value>
        <value></value>
      </values>
      <statuses><status>AR</status></statuses> // AR status means
Absent Read, so values should be treated as "null"
    </interval>
  </intervals>
</intervalDataSet>
</intervalDataSets>
...
```

Full interval data details

```
{
  "intervalDataSets" : [{
    "code" : "LS1",
    "description" : "Load survey 1",
    "intervalLength" : 30,
    "intervalLengthUnits" : "MINUTE",
    "channels" : [{
      "name": "E1",
      "units": "Wh",
      "flowDirection": "DELIVERED",
      "apportionPolicy": "SUM",
      "phase": "NONE"
    }], {
      "name": "Q1",
      "units": "Varh",
      "flowDirection": "DELIVERED",
      "apportionPolicy": "SUM",
      "phase": "NONE"
    }, {
      "name": "E2",
      "units": "Wh",
      "flowDirection": "DELIVERED",
      "apportionPolicy": "SUM",
      "phase": "NONE"
    }
  ]],
  "intervals" : [{
    "endOfInterval" : "2013-01-22T00:00:00+0000",
    "values" : [ 2042.32, 3401.3, 34 ]
  }, {
    "endOfInterval" : "2013-01-22T00:00:00+0000",
    "values" : [ 2142.32, 2201.2, 23 ]
  }, {
    "endOfInterval" : "2013-01-22T00:00:00+0000",
    "values" : [ 3352, 3301.3, 56.6 ]
  }, {
    "endOfInterval" : "2013-01-22T00:00:00+0000",
    "values" : [ 2022.11, 3014.3, 23 ]
  }, {
    "endOfInterval" : "2013-01-22T00:00:00+0000",
    "values" : [ 2342.2, 3441.3, 23.1 ]
  }, {
    "endOfInterval" : "2013-01-22T00:00:00+0000",
    "values" : [ 2344.31, 2591.07, 80.33 ],
    "statuses" : [ "AR" ]
  }
  ]
}
```

7.1.2.1. Description

	Property			MDX Usage	Format	Description
--	----------	--	--	--------------	--------	-------------

intervalDataSet				Mandatory	-	Each intervalDataSet normally corresponds to the data (values and per-interval status flags) read for a particular load survey. All intervals within it must share the same length. However, different <i>intervalDataSet</i> elements may have different interval lengths, different channels, etc. Where a meter has more than one load survey, the data for each load survey should be supplied in a separate intervalDataSet. See MDX File Format Import Rules#MultipleLoadSurveys for behaviour when there are multiple load surveys.
	code			Optional	Text (20)	Optional short identifier for the data set, e.g. "LS2"
	description			Optional	Text (100)	Optional text description of the intervalDataSet, e.g. "Load Survey 2".
	intervalLength			Mandatory	Integer	Length of each interval in the set. Supported values (for minutes or seconds) are 1, 5, 10, 15, 20, 30, 60.
	intervalLengthUnits			Mandatory	See Calendar Interval Length Units	The unit the interval length is expressed in. At the moment only SECOND and MINUTE are supported.
	channel			Mandatory		Defines an interval (load survey) channel as accurately as possible.
		name		Optional	Text (50)	A custom label for the channel.
		units		Mandatory	See MDX - Unit Types (UOM)	Unit of measure.
		flowDirection		Mandatory	See MDX - Flow Direction	Flow direction, e.g. "DELIVERED", "ABSOLUTE" or "Q1". Flow direction must be valid for UOM (See MDX - Unit Types (UOM))
		phase		Mandatory	See MDX - Phase	Phase, e.g. "A" or "AB".
		apportionPolicy		Mandatory	See MDX - Apportion Policy	The rule to follow when apportioning (for example, when viewing 15-minute day apportioned to 1-day intervals).
	interval			Mandatory	-	Contains one value for each channel for a particular interval.
		endOfInterval		Mandatory	ISO8601	The end date & time of the interval. Timezones accept only numeric values. 1200 equates to GMT+12:00.
		values		Mandatory	-	All channel values for this interval

			value	Mandatory	Numeric	The value of the channel for the containing interval. For missing data, set the status flag to "AR", or use a blank value, i.e. <value> or null in json
		statuses		Optional	-	Holds the collection of statuses that apply to the interval. Examples of statuses include power cut and equipment failure alarms. Statuses are not specific to individual channels.
			status	Optional	See Interval Status Codes	This holds a code for a particular interval status that applies to the containing interval, for example "AP" for "Asymmetric Power".

7.1.3. Data Gaps

MultiDrive may export reads containing gaps - these will NOT be filled in automatically. Gaps between reads (i.e. between data in separate files) can occur when a meter has been reset or when there is a sufficiently large gap between reads.













7.1.4. Calendar Interval Length Units

Units that time is measured in, as opposed to rates that events occur at (see PeriodType below).

Unit	Description	iE2 Notes
MILLISECOND	Milliseconds.	<i>Not currently supported by the importer.</i>
SECOND	Seconds.	
MINUTE	Minutes.	
HOURL	Hours.	<i>Not currently supported by the importer.</i>
DAY	Days.	<i>Not currently supported by the importer.</i>
WEEK	Weeks.	<i>Not currently supported by the importer.</i>
MONTH	Months.	<i>Not currently supported by the importer.</i>
YEAR	Years.	<i>Not currently supported by the importer.</i>

Interval Status Codes

Device Flag in MDM (GUI)	Description	iE2 Data Quality	iE2 Value Preserved	iE2 Power Cut Flag	iE2 Exception Type	iE2 Exception Text	Atlas Profile Status	Genius Profile Status	MD Profile status Bit Field
AR	Absent Reading - Filled entry - no data. All data values to be ignored (they will generally be zero)	Missing	—	—	Absent Reading	Missing data between 29/08/2004 04:00:00 and 30/08/2004 (41 intervals)	A	A	10

II	<p>Incomplete interval.</p> <p>Normally this specifies any interval where the interval was short or long, whether due to a time change or power outage, or the very first interval.</p> <p>When the Load Survey I Flag on all Intervals with a Time Change option is set (See Figure 16-2) then the I flag only flags intervals where a time change occurred (or that a time change skipped over). Usually used with the P flag option on the same page, this separates the P and I flags to indicate short/long intervals due to time changes, or short/long intervals due to power outage.</p>	Actual			Incomplete Interval	Incomplete intervals between 29/08/2004 04:00:00 and 30/08/2004 (41 intervals)	I	I	12
PL	<p>Power failed during interval. This is set in the interval where power was lost, not where it just continued to be lost or was restored.</p> <p>The operation can be changed to indicate the P flag in all intervals where power was lost, restored, or continued to be lost, using the Load Survey P Flag on all Intervals with a Power Outage setting.</p>	Actual			Power Outage	Power outage between 29/08/2004 04:00:00 and 30/08/2004 (41 intervals)	P	P	9
UD1	<p>User programmable flag derived from the active EFA status.</p> <p>MV90 labels this as a critical error using interval bit 5.</p>	Actual			?	?	L	L	13
UD2	<p>User programmable flag derived from the active EFA status.</p> <p>MV90 labels this as a clock error using channel bit 7.</p>	Actual			?	?	W		14
UD3	<p>User programmable flag derived from the active EFA status.</p> <p>MV90 labels this as a VT failure using channel bit 9.</p> <p>May also be configured to indicate a setup change using the Load Survey U Flag Records Setup Changes.</p>	Actual			?	?	U		15
MF	<p>Memory Failure or data check-sum error. This record should be treated with caution.</p>	Actual						B	8

MD	Data was manually modified using MultiDrive System								0
ED	Data was manually estimated using MultiDrive System								1
SD	Data was manually substituted using MultiDrive System								2
ID	Data was marked invalid in MultiDrive System								3
UF1	Data was marked as user flag 1 in MultiDrive System								4
UF2	Data was marked as user flag 2 in MultiDrive System								5
UF3	Data was marked as user flag 3 in MultiDrive System								6
UF4	Data was marked as user flag 4 in MultiDrive System								7

Note: given the limited number of statuses, we may want to extend DQM to show more flags (currently it just shows Power Cut).

7.1.5. Get Time of Use Register Data

URL	/measuredData/timeOfUse/{deviceId}					
	/measuredData/timeOfUse/plantNumber/{plantNumber} ⚠					
Method	GET					
Permission	mdpTOUData [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	startDate	The date to start the data period	string	Mandatory		-
	endDate	The date to end the data period (max of 1 month from startDate)	string	Mandatory		-
	periodType	The type of TOU register to fetch, values separated by commas	string	Optional	TOTAL, CURRENT, BILLING_TOTAL, PREVIOUS, PREVIOUS_{X}	*
	limit	Maximum number of records to return status for in response	Integer	Optional	0 - 5000	5000
	offset	Offset in devices to start returning records from	Integer	Optional	*	0
Returns	200 OK 400 Bad Request (parameters are not valid) 401 Unauthorised (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Device Id Does not exist)					
	Fetches Time of Use register values between the query dates. May filter on the periodType if supplied, if not then all period types returned. <div><div>MDX export example</div><pre>... <registerDataSets></pre></div>					

```

<registerDataSet>
  <recordDateTime>2012-06-10T12:30:00.000+0000</recordDateTime>
  <periodType>PREVIOUS</periodType>
  <billingResetCount>2</billingResetCount>
  <billingResetTime>2012-06-09T12:30:00.000+0000</billingResetTime>
  <registers>
    <register>
      <units>WH</units>
      <flowDirection>DELIVERED</flowDirection>
      <phase>TOTAL</phase>
      <apportionPolicy>SUM</apportionPolicy>
      <touRate>TOTAL</touRate>
      <registerValue>
        <value>123</value>
      </registerValue>
    </register>
    <register>
      <units>WH</units>
      <flowDirection>DELIVERED</flowDirection>
      <phase>TOTAL</phase>
      <apportionPolicy>SUM</apportionPolicy>
      <touRate>1</touRate>
      <registerValue>
        <value>100</value>
      </registerValue>
    </register>
    <register>
      <units>WH</units>
      <flowDirection>DELIVERED</flowDirection>
      <phase>TOTAL</phase>
      <apportionPolicy>SUM</apportionPolicy>
      <touRate>2</touRate>
      <registerValue>
        <value>23</value>
      </registerValue>
    </register>
    <register>
      <units>varh</units>
      <flowDirection>DELIVERED</flowDirection>
      <phase>TOTAL</phase>
      <apportionPolicy>SUM</apportionPolicy>
      <touRate>TOTAL</touRate>
      <registerValue>
        <value>345</value>
      </registerValue>
    </register>
    <register>
      <units>W</units>
      <flowDirection>NONE</flowDirection>
      <phase>TOTAL</phase>
      <apportionPolicy>MAX</apportionPolicy> // value recorded is MAXIMUM
demand
      <touRate>TOTAL</touRate>
      <registerValue>
        <value>567</value>
        <dateTime>2012-06-10T12:45:00.000+0000</dateTime> // time of max
demand
      </registerValue>
    </register>
    <register>
      <units>var</units>
      <flowDirection>NONE</flowDirection>
      <phase>TOTAL</phase>
      <apportionPolicy>MAX</apportionPolicy>
      <touRate>TOTAL</touRate>

```

```
        <registerValue>
          <value>890</value>
          <dateTime>2012-06-10T12:45:00.000+0000</dateTime> // time of max
demand
        </registerValue>
      </register>
    </registers>
```

```

    </registerDataSet>
  </registerDataSets>
  ...

```

Full registerDataSets example

```

{
  "registerDataSets": [{
    "recordDateTime": "2012-06-10T12:30:00.000+0000",
    "periodType": "PREVIOUS",
    "billingResetCount": 2,
    "billingResetTime": "2012-06-09T12:30:00.000+0000",
    "registers": [{
      "units": "WH",
      "flowDirection": "DELIVERED",
      "phase": "TOTAL",
      "apportionPolicy": "SUM",
      "touRate": "TOTAL",
      "registerValue": {
        "value": 123,
      }
    }, {
      "units": "WH",
      "flowDirection": "DELIVERED",
      "phase": "TOTAL",
      "apportionPolicy": "SUM",
      "touRate": "1",
      "registerValue": {
        "value": 100
      }
    }, {
      "units": "WH",
      "flowDirection": "DELIVERED",
      "phase": "TOTAL",
      "apportionPolicy": "SUM",
      "touRate": "2",
      "registerValue": {
        "value": 23
      }
    }, {
      "units": "varh",
      "flowDirection": "DELIVERED",
      "phase": "TOTAL",
      "apportionPolicy": "SUM",
      "touRate": "TOTAL",
      "registerValue": {
        "value": 345
      }
    }, {
      "units": "W",
      "flowDirection": "NONE",
      "phase": "TOTAL",
      "apportionPolicy": "MAX", // value recorded is MAXIMUM demand
      "touRate": "TOTAL",
      "registerValue": {
        "value": 567,
        "dateTime": "2012-06-10T12:42:00.000+0000", // time of max demand
      }
    }, {
      "units": "var",
      "flowDirection": "NONE",
      "phase": "TOTAL",

```

```
"apportionPolicy": "MAX",  
"touRate": "TOTAL",  
"registerValue": {  
  "value": 890,  
  "dateTime": "2012-06-10T12:45:04.000+0000", // time of max demand  
}
```

```
}  
  }  
}
```

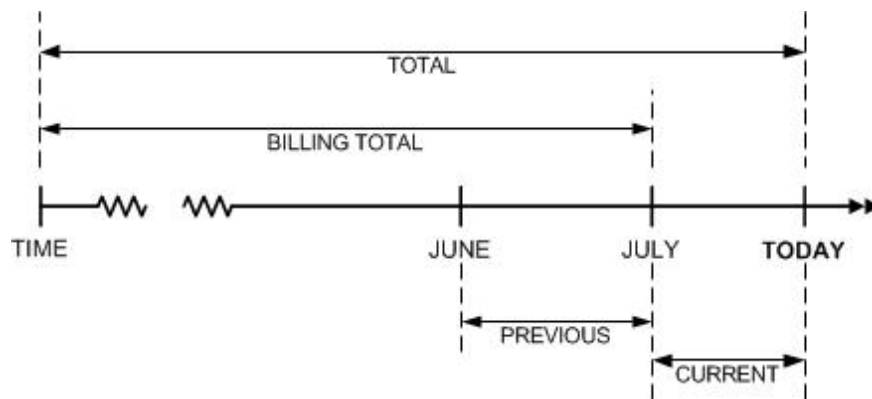
7.1.5.1. Description

Property				iE2 Import Usage	Format	Description
registerDataSet				Mandatory	-	All the (tou) registers for a single billing reset
	recordDateTime			Mandatory	ISO8601	The date and time at which the register data was recorded.
	periodType			Mandatory	See <i>Period Types</i>	Type of period the energy and/or maximum demand channels have been computed over.
	billingResetCount			Optional	Numeric	The number of billing resets that have been performed since the meter was installed.
	billingResetTime			Optional	ISO8601	The date and time at which the last billing reset was performed.
	register			Mandatory (1..* Occurrences)	-	Describes a single TOU register and its value
		units		Mandatory	See <i>MDX - Unit Types (UOM)</i>	Unit of measure.
		flowDirection		Mandatory	See <i>MDX - Flow Direction</i>	Flow direction, e.g. "DELIVERED", "ABSOLUTE" or "Q1" Flow direction must be valid for UOM. (See <i>MDX - Unit Types (UOM)</i>)
		phase		Mandatory	See <i>MDX - Phase</i>	Phase, e.g. "A" or "AB".
		apportion policy		Mandatory	See <i>MDX - Apportion Policy</i>	The rule to follow when apportioning (for example, when viewing 15-minute day apportioned to 1-day intervals).
		touRate		Mandatory	Text	The number of the TOU rate the value is for, or TOTAL if it is not for a specific TOU band.
		block			Integer	The block number for use when block tariffs are defined in the meter
		registerValue		Mandatory	-	Wrapper for a register value and its attributes.
			value	Mandatory	Mandatory	Numeric value for the channel, e.g. energy or demand value.
			dateTime	Optional	ISO8601	Where the value is for a maximum demand channel, this is the date recorded for that value.

7.1.6. Period Types

Type of period a register value relates to.

Period Type	Energy Values	Maximum Demand Values
TOTAL	Values are the total amount of energy accumulated since the meter was installed or its registers were reset to 0.	Values are the sum of the individual maximum demand values recorded since the meter was installed or its registers were reset to 0 until now.
BILLING_TOTAL	Values are a snapshot of the TOTAL energy values at the time of the last billing reset.	Values are the sum of the individual maximum demand values recorded since the meter was installed or its registers were reset to 0 up and to the last billing reset.
CURRENT	Values are the amount of energy accumulated between the last billing reset and the current time. (i.e. the amount of energy consumed in the current billing period.) $CURRENT = TOTAL - BILLING_TOTAL$	Values are the maximum demand values recorded since the last billing reset until now.
PREVIOUS	Previous can be two different values depending on the meter configuration: <ul style="list-style-type: none"> Interval - Values are the amount of energy accumulated in a specific billing period. (as depicted in the diagram above) If a meter is configured this way, then at a Billing Reset time, the $PREVIOUS_1 = CURRENT$ and $CURRENT$ will be reset to 0 ready for the next period. Cumulative - Values are the total amount of energy accumulated since the meter was installed until the billing reset of the specific Previous period. (i.e. It is a 'Previous' value of the Billing Total. So $PREVIOUS_1 = BILLING_TOTAL$, $PREVIOUS_2 = BILLING_TOTAL$ from the period prior) 	Values are the maximum demand value recorded for the billing period represented by the PREVIOUS period. $PREVIOUS_1 = BILLING_TOTAL$ maximum demand values.



Example showing monthly billing reset for an interval based meter configuration

7.1.7. Supported Register Channels

The following Register Channel attributes will be supported by default:

Units	Directions	Apportion Policies	Phases	Rates	Input Number	Period Type
Wh, varh, VAh	DELIVERED, RECEIVED	SUM	NONE	TOTAL, 1-8	0, 1, 2	PREVIOUS, CURRENT, BILLING TOTAL ,TOTAL
Wh, varh, VAh	DELIVERED, RECEIVED	SUM	A, B, C	TOTAL	0	PREVIOUS, CURRENT, BILLING TOTAL ,TOTAL
W, VA, var	NONE, DELIVERED, RECEIVED	MIN, MAX, AVG, INST	NONE	TOTAL, 1-8	0, 1, 2	PREVIOUS, CURRENT, BILLING TOTAL ,TOTAL
W, VA, var	NONE, DELIVERED, RECEIVED	MIN, MAX, AVG, INST	A, B, C	TOTAL	0	PREVIOUS, CURRENT, BILLING TOTAL ,TOTAL

7.1.8. Get Event Log Data

URL	/measuredData/eventLog/{deviceId} /measuredData/eventLog/plantNumber/{plantNumber}				
Method	GET				
Permission	mdpDeviceEventLog [View]				
Parameters	Name	Description	Type	Optional/Mandatory	Valid values
	startDate	The date to start the data period	string	Mandatory	
	endDate	The date to end the data period (max of 1 month from startDate)	string	Mandatory	
	logNumber	The identifier for the required event log (0 = all event logs)	Integer	Optional	*
	limit	Maximum number of records to return status for in response	Integer	Optional	0 - 5000
	offset	Offset in devices to start returning records from	Integer	Optional	*
Returns	200 OK 400 Bad Request (parameters are invalid) 401 Unauthorised (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Device Id Does not exist)				
	All device events are encapsulated into this object. This includes normal device events, voltage sag/swell events, THD/Unbalance events and tau events.				

MDX export example

```
...
<events>
  <event>
    <code>1.0.1.1</code>
    <logNumber>0</logNumber>
    <description>Power on: connected</description>
    <eventDateTime>2012-06-11T00:00:00.000+0000</eventDateTime>
    <parameters>
      <parameter>
        <name>extremeAmplitude</name>
      </parameter>
    </parameters>
    <value>1.00</value>
  </event>
  <event>
    <code>3.4.17.42</code>
    <description>Relay connected</description>
    <parameters>
      <parameter>
        <name>relayNumber</name>
        <value>Relay 3</value>
      </parameter>
      <parameter>
        <name>relayOperationType</name>
        <value>Disconnect Button</value>
      </parameter>
    </parameters>
  </event>
</events>
...
```

Full event details

```
{
  "events": [{
    "code": "1.0.1.1",
    "logNumber": 0,
    "description": "Power on: connected",
    "eventDateTime": "2012-06-11T00:00:00.000+0000",
    "parameters": [{
      "name": "extremeAmplitude",
      "value": "1.00"
    }]
  },
  {
    "code": "3.4.17.42",
    "description": "Relay connected",
    "parameters": [{
      "name": "relayNumber",
      "value": "Relay 3"
    }, {
      "name": "relayOperationType",
      "value": "Disconnect Button"
    }]
  }
]
```

Description

Property			iE2 Import Status	Format	Description	Notes
code			Mandatory	As per the 'EndDeviceEvent' definition in IEC 61968-9: "<domain>.<domainpart>.<type>.<index>" (each of these 4 fields are numeric only)	EndDeviceEvent category	Codes will be sup covering all mete types. iBO will h translation from 1 codes to iBO's int event types. Parameters will b meter specific to richer detail wher exists
description			Optional	Text	Description of type of event, e.g. "Power on"	This is useful if tl is not understood/m at importion for c
logNumber			Optional	Numeric	The log in which this event was recorded in.	This will be igno the importer
eventDateTime			Mandatory	ISO8601	Date and time event was recorded by the meter.	
parameters			Optional		Defines an event parameter name/value pair	
	parameter		Optional (0..* occurrences)			There could be 0, multiple paramet each event
		name	Mandatory	Text	The name of the parameter	
		value	Mandatory	Text	The value of the parameter	

7.1.9. Get Voltage Sag/Swell, THD or Unbalance Event Data

URL	/measuredData/sagSwell/{deviceId} /measuredData/sagSwell/plantNumber/{plantNumber}				
Method	GET				
Permission	mdpSagSwellLog [View]				
Parameters	Name	Description	Type	Optional/Mandatory	Valid values
	startDate	The date to start the data period	string	Mandatory	
	endDate	The date to end the data period (max of 1 month from startDate)	string	Mandatory	
	limit	Maximum number of records to return status for in response	Integer	Optional	0 - 5000
	offset	Offset in devices to start returning records from	Integer	Optional	*

Returns	<p>200 OK</p> <p>400 Bad Request (parameters are invalid)</p> <p>401 Unauthorised (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id Does not exist)</p>
	<p>All device events are encapsulated into this object. This includes normal device events, voltage sag/swell events, THD/Unbalance events and tau events.</p> <div> <div>MDX export example</div> <pre> ... <events> <event> <code>1.0.1.1</code> <logNumber>0</logNumber> <description>Power on: connected</description> <eventDateTime>2012-06-11T00:00:00.000+0000</eventDateTime> <parameters> <parameter> <name>extremeAmplitude</name> </parameter> </parameters> <value>1.00</value> </event> <event> <code>3.4.17.42</code> <description>Relay connected</description> <parameters> <parameter> <name>relayNumber</name> <value>Relay 3</value> </parameter> <parameter> <name>relayOperationType</name> <value>Disconnect Button</value> </parameter> </parameters> </event> </events> ... </pre> </div>

Full event details

```
{
  "events": [{
    "code": "1.0.1.1",
    "logNumber": 0,
    "description": "Power on: connected",
    "eventDateTime": "2012-06-11T00:00:00.000+0000",
    "parameters": [{
      "name": "extremeAmplitude",
      "value": "1.00"
    }]
  },
  {
    "code": "3.4.17.42",
    "description": "Relay connected",
    "parameters": [{
      "name": "relayNumber",
      "value": "Relay 3"
    }, {
      "name": "relayOperationType",
      "value": "Disconnect Button"
    }]
  }
]
```

Description

Property			iE2 Import Status	Format	Description	Notes
code			Mandatory	As per the 'EndDeviceEvent' definition in IEC 61968-9: "<domain>.<domainpart>.<type>.<index>" (each of these 4 fields are numeric only)	EndDeviceEvent category	Codes will be supplied covering all meter types. iBO will handle translation from IEC codes to iBO's internal event types. Parameters will be meter specific to provide richer detail where exists
description			Optional	Text	Description of type of event, e.g. "Power on"	This is useful if the event is not understood/misinterpreted at import for clarification
logNumber			Optional	Numeric	The log in which this event was recorded in.	This will be ignored by the importer
eventDateTime			Mandatory	ISO8601	Date and time event was recorded by the meter.	
parameters			Optional		Defines an event parameter name/value pair	
	parameter		Optional (0..* occurrences)			There could be 0, multiple parameters for each event
		name	Mandatory	Text	The name of the parameter	
		value	Mandatory	Text	The value of the parameter	

Specific example of a sag/volt, THD or unbalance event.


Full event details

```
{
  "events": [{
    "code": "6.0.1.1",
    "logNumber": 3,
    "description": "Sag event",
    "eventDateTime": "2012-06-11T00:00:00.000+0000",
    "parameters": [
      {
        "name": "eventType",
        "value": "Voltage"
      },
      {
        "name": "phase",
        "value": "phaseA"
      },
      {
        "name": "duration",
        "value": 1.00
      },
      {
        "name": "extremeAmplitude",
        "value": 270.00
      },
      {
        "name": "averageAmplitude",
        "value": 250.00
      }
    ]
  }
]
```

Description of parameters

Property	Format	Description
eventType	Text	The type of sag/swell event this is. Valid values: <ul style="list-style-type: none">• Voltage• THD• Unbalance
startTime	ISO8601	Date and time the sag/swell event started.
phase	Text	The phase the sag/swell occurred on. Valid values <ul style="list-style-type: none">• PhaseA• PhaseB• PhaseC• Main• Load• Load1
duration	Double	Total number of seconds the event occurred for
extremeAmplitude	Double	The worst amplitude for the event (lowest or highest value before returning)
averageAmplitude	Double	The average amplitude for the event over the duration

8. Supply Control API

8.1.1.1. Change Main Supply Status (connect / disconnect)						
URL	/supplyControl/mainSupply/{deviceid} /supplyControl/mainSupply/plantNumber/{plantNumber}					
Method	POST					
Permission	mdpMainsControl [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	startDate	The date the device should change it's relay to the new state	date	Optional		None
	endDate 	The date the device's relay should revert to it's previous state. If empty, then do not revert.	date	Optional		Blank
	controlCode	The code which will be issued to the meter to change the relay state	string	Mandatory	Connect, EnableAndDisconnect, DisableAndDisconnect	

Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } } </pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } } </pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter</p> <p>/scheduling/executions/{executionId}, This /scheduling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id does not exist)</p> <p>500 Internal Server Error</p>

8.1.1.2. Get Main Supply status

URL	/supplyControl/mainSupply/{deviceid} /supplyControl/mainSupply/plantNumber/{plantNumber}					
Notes	This returns the last known relay status. To get the latest/current status from the meter, a schedule must be triggered first.					
Method	GET					
Permission	mdpDevice [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default

Returns

200 OK

401 Unauthorized (Token invalid)

403 Forbidden (Access not allowed)


404 Not Found (Device Id or plant number does not exist)

5500 Internal Server Error

```
{
  "relayStatus" :
  {
    "deviceId" : 14,
    "statusTime" : "2013-01-21T10:00:00",
    "lastCmdStartTime" : "2013-01-21T10:00:00",
    "lastCmdEndTime" : "2013-01-21T10:00:00",
    "changePending" : "False",
    "status" : "Connected",
    "info" : "",
  },
}
```

8.1.1.3. Description

Property				iE2 Import Usage	Format	Description
relayStatus						
	deviceId			Mandatory	Integer	Device Id
	statusTime			Mandatory	ISO8601	Date/Time of when status was read
	status			Mandatory	String	Current (or pending) status of the relay. Valid statuses are: <ul style="list-style-type: none"> • Connected • EnabledAndDisconnected • DisabledAndDisconnected • Error
	changePending			Optional	Boolean	True if change is still pending False is already changed
	info			Optional	Text	Misc information

8.1.1.4. Get Main Supply status for a device group						
URL	/supplyControl/mainSupply/deviceGroup/{deviceGroupId} /supplyControl/mainSupply/deviceGroup/name/{deviceGroupName} 					
Notes	This returns the last known relay status for all the devices. To get the latest/current status from the meters, a schedule must be triggered first.					
Method	GET					
Permission	mdpDevice [View] and mdpMeterGroup [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	limit	Maximum number of devices to return status for in response	Integer	Optional	0 - 5000	5000
	offset	Offset in devices to start returning statuses for in response	Integer	Optional	*	0

Returns

200 OK

401 Unauthorized (Token invalid)

403 Forbidden (Access not allowed)

404 Not Found (Device Group Id Does not exist)

500 Internal Server Error

```
{
  "relayStatus" :
  {
    "deviceId" : 14,
    "statusTime" : "2013-01-21T10:00:00",
    "lastCmdStartTime" : "2013-01-21T10:00:00",
    "lastCmdEndTime" : "2013-01-21T10:00:00",
    "changePending" : "False",
    "status" : "Connected",
    "info" : "",
  },
}
```



8.1.1.5. Description

Property				iE2 Import Usage	Format	Description
relayStatus						
	deviceId			Mandatory	Integer	Device Id
	statusTime			Mandatory	ISO8601	Date/Time of when status was read
	status			Mandatory	String	Current (or pending) status of the relay. Valid statuses are: <ul style="list-style-type: none"> • Connected • EnabledAndDisconnected • DisabledAndDisconnected • Error
	changePending			Optional	Boolean	True if change is still pending False is already changed
	info			Optional	Text	Misc information

9. Fleet Management API

9.1.1. Devices

9.1.1.1. List / Search Devices						
URL	/fleetManagement/devices					
Method	GET					
Permission	mdpDevice [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	serialNumber	Filter by serial number	string	Optional		*
	plantNumber	Filter by plant number	string	Optional		*
	deviceTypeName	Filter by device type name	string	Optional		*

	status	Filter by the meter status	string	Optional	AWAITING_INSTALLATION, INSTALLED, COMMISSIONED, NOT_IN_SERVICE	*
	siteAssigned	Whether the site is assigned.	boolean	Optional	true, false	*
	siteCode	Filter by the site code	string	Optional		*
	installDate	Date meter was started to be read from	Date	Optional		*
	commissionDate	Date device was commissioned	Date	Optional		*
	decommissionDate	Date device was decommissioned	Date	Optional		*
	limit 	Maximum number of devices to return status for in response	Integer	Optional	0 - 5000	5000
	offset 	Offset in devices to start returning statuses for in response	Integer	Optional	*	0
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					

```

{
  "devices" : [
    {
      "deviceId" : 156312,
      "plantNumber" : "ABC1234"
      "serialNumber" : "41545456",
      "series" : "Atlas 2.0"
      "model" : "EDMI Mk10E",
      "installDate" : "2013-01-22T10:50:32",
      "commissionDate" : "2013-01-25T12:02:03",
      "decommissionDate" : "",
      "status" : "INSTALLED",
      "siteCode" : "SiteABC",
      "siteId" : 134980,
      "deviceConfiguration" : "Mk 10 Import 2 Ch",
      "deviceConfigurationId" : 33
    },...
  ]
}

```

9.1.1.2. Description

Property	iE2 Import Usage	Format	Valid Values	Description
deviceId	Mandatory	Integer		A unique identifier for the device
plantNumber	Mandatory	String		The manufacturer serial number of the device
serialNumber	Mandatory	String		The market identifier for the device
series	Mandatory	String		The name of the series this meter is from, e.g. "Atlas"
model	Mandatory	String		The name of the meter's model within its series, e.g. "Mk 10A"
installDate	Mandatory	String		The date and time the meter was installed
commissionDate	Mandatory	String		The date the meter was commissioned
decommissionDate	Mandatory	String		The date the meter was decommissioned in the system
status	Mandatory	String	AWAITING_INSTALLATION, INSTALLED, COMMISSIONED, NOT_IN_SERVICE	The status of the device
siteCode	Mandatory	String		The site code of the site used to communicate with the meter
siteId	Mandatory	Integer		The siteId of the site used to communicate with the meter
deviceConfiguration	Mandatory	String		The name of the device configuration used by the meter
deviceConfigurationId	Mandatory	Integer		The id of the device configuration used by the meter

Note: *commissionDate and decommissionDate come from the actual CommissionedDate, DecommissionedDate fields in the database, not InstallDate or*

9.1.1.3. Get Device		
URL	/fleetManagement/devices/{ deviceId }	
	/fleetManagement/devices/plantNumber/{ plantNumber } ⚠	
Method	GET	
Permission	mdpDevice [View]	
Returns	200 OK	
	401 Unauthorized (Token invalid)	
	403 Forbidden (Access not allowed)	
	404 Not Found	
		<div>MDX example</div>

```

...
<device>
  <deviceId>8072584</deviceId>
  <plantNumber>8072584</plantNumber>
  <serialNumber>8072584</serialNumber>
  <series>Atlas</series>
  <model>Mk10B</model>
  <deviceConfiguration>Mk 10 Import 2 Ch</deviceConfiguration>
  <installDate>2013-01-22T10:53:03</installDate>

  <status>INSTALLED</status>
  <siteId>12345678</siteId>
  <calorificValueId>0</calorificValueId>
  <ct>
    <primary>2000</primary>
    <secondary>5</secondary>
  </ct>
  <pt>
    <primary>110</primary>
    <secondary>110</secondary>
  </pt>
  <timeOffsetMins>600</timeOffsetMins>
  <userLogins>
    <userLogin>
      <role>SUPER</role>
      <username>Bob</username>
      <password>kb3rsf</password>
    </userLogin>
    <userLogin>
      <role>SUPER</role>
      <username>Supal</username>
      <password>kb3rsf</password>
    </userLogin>
    <userLogin>
      <role>NORMAL</role>
      <username>Bob</username>
      <password>jd83hf</password>
    </userLogin>
  </userLogins>
  <deviceCommsSettings>
    <commsSettings>
      <order>1</order>
      <protocol>Edmi MiniE</protocol>
      <loginTimeoutSecs>10</loginTimeoutSecs>
      <timeoutSecs>90</timeoutSecs>
      <maxRetries>3</maxRetries>
      <devicePort>0</devicePort>
    </commsSettings>
    <commsSettings>
      <order>2</order>
      <protocol>Edmi MiniE</protocol>
      <loginTimeoutSecs>10</loginTimeoutSecs>
      <timeoutSecs>75</timeoutSecs>
      <maxRetries>3</maxRetries>
    </commsSettings>
  </deviceCommsSettings>
  <deviceGroups>
    <deviceGroup>
      <deviceGroupId>3</deviceGroupId>
      <name>Inbound Daily Read</name>
    </deviceGroup>
  </deviceGroups>
</device>
...

```


Full device details

```
{
  "device": {
    "deviceId" : 8072584,
    "plantNumber" : "ABC23421",
    "serialNumber" : "3410293A",
    "series" : "Atlas",
    "model" : "Mk10B",
    "deviceConfiguration" : "Mk 10 Import 2 Ch",
    "deviceConfigurationId" : 33,
    "siteId" : 15,
    "calorificValueId" : 0,
    "installDate" : "2013-01-22T00:00:00",
    "commissionDate" : "2013-01-23T00:00:00",
    "decommissionDate" : "",
    "status" : "INSTALLED",
    "misc" : "",
    "meterAppliesTransformer" : "true",
    "timeOffsetMins": 1,
    "userLogins" : [{
      "role" : "SUPER",
      "username" : "Bob",
      "password" : "kb3rsf"
    }, {
      "role" : "SUPER",
      "username" : "Bob",
      "password" : "jd83hf"
    }],
    "deviceCommsSettings" : [{
      "order" : 1,
      "protocol" : "Edmi MiniE",
      "loginTimeoutSecs" : 90,
      "maxRetries" : 3,
      "unitAddress" : 0,
      "devicePort" : 0
    }, {
      "order" : 2,
      "protocol" : "Edmi MiniE",
      "loginTimeoutSecs" : 75,
      "timeoutSecs" : 90,
      "maxRetries" : 3,
      "unitAddress" : 0
    }],
    "ct" : {
      "primary" : 2000,
      "secondary" : 5
    },
    "pt" : {
      "primary" : 110,
      "secondary" : 110
    },
    "deviceGroups" : [{
      "deviceGroupId" : 33,
      "name" : "Inbound Daily Read"
    }]
  }
}
```

9.1.1.4. Description

Property				MD MDX usage	E6U MDX usage	Storm MDX Usage	F
device							
	deviceId			Mandatory	Mandatory	Returned by Storm when Device is created Mandatory for update	S
	serialNumber			Mandatory	Mandatory	Mandatory	S
	plantNumber			Mandatory	Unused	Mandatory	S
	series			Mandatory	Mandatory	Mandatory	S S sc b
	model			Mandatory	Mandatory	Mandatory	S S sc b
	deviceConfiguration			Either of the two needs to be provided but cannot be both (Mandatory)	Unused	Unused - hard-coded to "default"	S
	deviceConfigurationId				Unused	Unused	It
	installDate			Mandatory	Mandatory	Mandatory	It
	commissionDate			Unused	Unused	Unused	It
	decommissionDate			Unused	Unused	Unused	It

	status			Mandatory	Mandatory	Mandatory	S S S
	misc			Unused	Unused	Unused	S
	meterAppliesTransformer			Unused	Unused	Unused	B
	siteId			Optional	Optional	Returned by Storm when Device is created Mandatory for update	S
	calorificValueId			Optional	Optional	Optional	Ir
	ct			Optional	Unused	Unused	
		primary		Optional	Unused	Unused	Ir
		secondary		Optional	Unused	Unused	Ir
	pt			Optional	Unused	Unused	
		primary		Optional	Unused	Unused	Ir
		secondary		Optional	Unused	Unused	Ir
	timeOffsetMins			Mandatory	Unused	Unused	Ir
	userLogins			Optional	Optional	Optional	
		userLogin		Optional	Optional	Optional	
			role	Mandatory	Mandatory	Mandatory	S N
			username	Mandatory	Mandatory	Mandatory	S

			password	Mandatory <i>Encrypted for MDX files</i> <i>Decrypted for REST API create/update payloads</i> <i>Encrypted for return value from REST API get/create/updates</i>	Mandatory <i>Decrypted for MDX files</i> <i>N/A for REST API create/updates</i>	Mandatory <i>N/A for MDX files</i> <i>Decrypted for REST API create/updates</i>	S
	deviceCommsSettings			Mandatory	Mandatory	Mandatory (currently not used by Storm)	
		commsSettings		Mandatory	Mandatory	Mandatory	
			order	Mandatory	Mandatory	Mandatory	It
			protocol	Mandatory	Mandatory	Mandatory	S P sc
			loginTimeoutSecs	Mandatory	Mandatory	Mandatory	It
			timeoutSecs	Mandatory	Mandatory	Mandatory	It
			maxRetries	Mandatory	Mandatory	Mandatory	It
			devicePort	Optional	Optional	Unused	It
			unitAddress	Unused	Unused	Unused	It
	deviceGroups			Optional	Unused	Optional (currently not used by Storm)	
		deviceGroup		Optional	Unused	Optional	
			deviceGroupId	Mandatory	Unused	Mandatory	It
			name	Optional	Unused	Optional	S

9.1.2. Protocols

Default
Edmi Command Line
Edmi Command Line Packet
Edmi MiniE
Edmi MiniE Packet
IEC 1107/EDMI Command Line
LON DLC

9.1.3. Device Statuses

AWAITING_INSTALLATION
INSTALLED
COMMISSIONED
NOT_IN_SERVICE

9.1.4. Meter Series


Series	Model
Atlas Gas	G6000
Atlas	Mk7
	Mk7A
	Mk7B
	Mk7C
	Mk10
	Mk10A
	Mk10D
	Mk10E
	Mk10UPS
	Mk10China
	Mk10Mini
	Mk10Allix
Atlas 2	Mk7
	Mk7A
	Mk7B
	Mk7C
	Mk10
	Mk10A
	Mk10D
	Mk10E
	Mk10UPS
	Mk10China
	Mk10Mini
	Mk10Allix
Atlas Storm	Mk7
	Mk7A
	Mk7B
	Mk7C
	Mk10
	Mk10A
	Mk10D
	Mk10E
	Mk10UPS
	Mk10China
	Mk10Mini
	Mk10Allix

Note that username/passwords and security keys are not returned in the MDX. They are only allowed in the

9.1.4.1. Create Device

URL	/fleetManagement/devices
Method	POST
Permission	mdpDevice [Create]
Payload	MDX devices object same as in Get Device operation
Notes	<p><i>Either deviceConfiguration (name of the device configuration) or deviceConfigurationId can be used to update the device configuration. However both cannot be present at the same time (will respond with 400 error).</i></p> <p>For deviceGroups:deviceGroup:name, this does nothing if present during this operation and is ignored.</p>
Read only attributes	deviceId, deviceGroups:deviceGroup:name
Returns	<p>201 Created - includes created object and location header containing the URN of the new resource (fleetManagement/devices/{DeviceId})</p> <p>400 Bad Request (device is invalid)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>409 Conflict (Plant Number already exists)</p> <p>500 Internal Server Error</p> <p>*Note that username/passwords and security keys are sent in the clear here.</p>

9.1.4.2. Update Device

URL	/fleetManagement/devices/{deviceId} /fleetManagement/devices/plantNumber/{plantNumber} 
Method	PUT
Permission	mdpDevice [Modify]
Payload	MDX devices object same as in Get Device operation, see MDX Object - Device
Notes	<ol style="list-style-type: none"> 1. This method is used to de activate devices, by setting the status to NOT_IN_SERVICE 2. Either deviceConfiguration (name of the device configuration) or deviceConfigurationId can be used to update the device configuration. However both cannot be present at the same time (will respond with 400 error) 3. For deviceGroups:deviceGroup:name, this does nothing if present during this operation and is ignored.
Read only attributes	deviceId, deviceGroups:deviceGroup:name
Returns	<p>200 OK (returns updated object)</p> <p>400 Bad Request (changes are invalid)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found</p> <p>500 Internal Server Error</p>

9.1.4.3. Delete Device

URL	/fleetManagement/devices/{deviceId}
	/fleetManagement/devices/plantNumber/{plantNumber} ⚠
Method	DELETE
Permission	mdpDevice [Delete]
Returns	200 OK
	401 Unauthorized (Token invalid)
	403 Forbidden (Access not allowed)
	404 Not Found
	500 Internal Server Error

9.1.4.4. Last Device Access Time (Scheduled and Adhoc Communications)

URL	/fleetManagement/devices/{deviceId}/lastAccessTime
	/fleetManagement/devices/plantNumber/{plantNumber}/lastAccessTime ⚠
Method	GET
Permission	mdpDevice [View]
Returns	200 OK
	401 Unauthorized (Token invalid)
	403 Forbidden (Access not allowed)
	404 Not Found
	500 Internal Server Error

Returns



```
{
  "executionJob" : {
    "deviceId" : 32,
    "plantNumber" : "Device01",
    "scheduleId" : 12,
    "executionId" : 125,
    "executionJobId" : 9875,
    "date" : "2013-01-21T10:00:00",
    "commsServer" : "Communications Pool",
    "status" : "Success",
    "hasWarning" : "true",
    "lastError" : "Communications Device Failure: System
error:Connection refused, Unable to connect to 192.168.3.20 on port 10002",
  }
}
```

9.1.4.5. Next Schedule Device Access Time

URL	/fleetManagement/devices/{deviceId}/nextAccessTime
	/fleetManagement/devices/plantNumber/{plantNumber}/nextAccessTime ⚠

Method	GET
Permission	mdpDevice [View]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 500 Internal Server Error
	<div> <div>Returns</div> <pre> { "schedule" : { "scheduleId" : 34, "name" : "Important Read Schedule", "recurrence" : "Everyday at 10:00" "lastRun" : "2013-01-21T10:00:00", "nextRun" : "2013-01-22T10:00:00", "status" : "Active" } } </pre> </div>

9.1.5. Sites

9.1.5.1. List / Search Sites						
URL	/fleetManagement/sites					
Method	GET					
Permission	mdpSite [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	siteCode	Filter by match in Site code	string	Optional		*
	isPersistentComms	Limits returned sites to those which have a comms device with persistence of the specified type	boolean	Optional	True or False	*
	limit 	Maximum number of devices to return status for in response	Integer	Optional	0 - 5000	5000
	offset 	Offset in devices to start returning statuses for in response	Integer	Optional	*	0
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					

```

{
  "sites" : [
    {
      "siteId" : 1,
      "siteCode" : "SiteOne"
    },...
  ]
}

```

9.1.5.2. Get Site

URL	/fleetManagement/sites/{siteId}
	/fleetManagement/sites/siteCode/{siteCode}
Method	GET
Permission	mdpSite [View]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 500 Internal Server Error

MDX example

```
...
<site>
  <siteId>12345678</siteId>
  <code>Queensgate Mall</code>
  <communicationsType>GPRS</communicationsType>
  <modemid>1</modemid>
  <simid>12</simid>
  <signalStrength>6</signalStrength>
  <baseStation>4w3,234,2h8</baseStation>
  <siteCommsSettings>
    <commsSettings>
      <order>1</order>
      <deviceType>IP</serverDeviceType>
      <deviceChaining>MULTIDROP</deviceChaining>
      <persistentComms>true</persistentComms>
      <ipSettings>
        <protocol>TCP</protocol>
        <ipAddress>208.23.259.34</ipAddress>
        <port>1899</port>
      </ipSettings>
    </commsSettings>
    <commsSettings>
      <order>2</order>
      <deviceType>MODEM</deviceType>
      <deviceChaining>SINGLE_SITE</deviceChaining>
      <modemSettings>
        <phoneNumber>0212023334</phoneNumber>
      </modemSettings>
    </commsSettings>
  </siteCommsSettings>
</site>
...
```

Full site details

```
{
  "site" : {
    "siteId" : 1,
    "code" : "SiteOne",
    "communicationsType" : "OTHER",
    "modemid": 1,
    "simid": 12,
    "signalStrength": 6,
    "baseStation": "4w3,234,2h8",
    "siteCommsSettings" : [
      {
        "order": 1,
        "deviceType" : "IP",
        "deviceChaining" : "MULTIDROP",
        "persistentComms" : true,
        "ipSettings" : {
          "ipAddress" : "10.1.2.3",
          "port" : 5975,
          "protocol" : "TCP"
        }
      },
      {
        "order" : 2,
        "deviceType" : "Modem",
        "deviceChaining" : "SINGLE_SITE",
        "persistentComms" : true,
        "modemSettings" : { // Modem settings would be included for devices connected
and for non persistent IP (for SMS).
          "phoneNumber" : "648957123"
        }
      }
    ]
  }
}
```

9.1.5.3. Description

Property				MD MDX usage	E6U MDX usage	Storm MDX usage	Format
siteId				Mandatory	Mandatory	Mandatory	Integer
code				Mandatory	Mandatory	Mandatory	Text(50)
communicationsType				Mandatory	Mandatory	Mandatory	STANDARD, GPRS, RF_MESH, ZIGBEE, P OTHER
modemid				Optional	Unused	Unused	Integer
simid				Optional	Unused	Unused	Integer

signalStrength				Optional	Unused	Unused	Integer
baseStation				Optional	Unused	Unused	String
siteCommsSettings				Mandatory	Mandatory	Mandatory	
	commsSettings			Mandatory	Mandatory	Mandatory	
		order		Mandatory	Mandatory	Mandatory	1 or 2
		deviceType		Mandatory	Mandatory	Mandatory - An IP device type is required for Storm	IP, MODEM DIRECT
		deviceChaining		Mandatory	Mandatory	Mandatory - MULTIDROP is not supported yet	SINGLE_S MULTIDRO
		persistentComms		Mandatory	Mandatory	Mandatory	boolean

		ipSettings		Optional	Mandatory	Mandatory	
			protocol	Optional	Mandatory	Mandatory	TCP / UDP
			port	Optional	Mandatory	Mandatory	Integer
			ipAddress	Mandatory	Mandatory	Mandatory	Text(50)
		modemSettings		Optional	Unused	Optional	
			phoneNumber	Mandatory	Unused	Mandatory	Text

9.1.5.4. Create Site

URL	/fleetManagement/sites
Method	POST
Permission	mdpSite [Create]
Payload	MDX sites object same as in Get Site operation
Read only attributes	siteId

Notes	<p>New sites are created in the MultiDrive system as indicated:</p> <ol style="list-style-type: none"> For new persistent IP meters <ol style="list-style-type: none"> Communications server is assigned as Pool Communications device is assigned as appropriate <ul style="list-style-type: none"> If UDP, finds first UDP one If TCP, finds first TCP one For non-persistent IP meters <ol style="list-style-type: none"> Assigns to non-pooled communications server with least number of meters assigned to it Communications device is assigned as appropriate <ul style="list-style-type: none"> If UDP, finds first UDP one If TCP, finds first TCP one For meters with modems <ol style="list-style-type: none"> Assigns to modem with least number of meters assigned Flags error if no modem com devices available (412 error)
Returns	<p>201 Created - includes created object and location header containing the URN of the new resource (/sites/{siteId})</p> <p>400 Bad Request (New site is not valid, device type is not valid)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>409 Conflict (Site code already exists). Site Id of conflicting resource will be returned here.</p> <p>412 Precondition Failed (No modems in the system for communications pool)</p> <p>500 Internal Server Error</p>

9.1.5.5. Delete Site

URL	/fleetManagement/sites/{siteId} /fleetManagement/sites/siteCode/{siteCode}
Method	DELETE
Permission	mdpSite [Delete]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found

9.1.5.6. Update Site

URL	/fleetManagement/sites/{siteId} /fleetManagement/sites/siteCode/{siteCode}
Method	PUT
Permission	mdpSite [Modify]
Read only attributes	siteId
Parameters/Payload	As per create

Returns	200 OK (this will return the updated site details)
	400 Bad Request (site changes are not valid)
	401 Unauthorized (Token invalid)
	403 Forbidden (Access not allowed)
	404 Not Found
	500 Internal Server Error

9.1.6. List / Search Device Configurations						
URL	/fleetManagement/deviceConfigurations					
Method	GET					
Permission	mdpDeviceConfig [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	name	Filter by match in device configuration name	string	Optional		*
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					
	<div> <pre> { "deviceConfigurations" : [{ "deviceConfigurationId" : 1, "name" : "config1" "series" : "Atlas" "model" : "Mk 10A" }, { "deviceConfigurationId" : 2, "name" : "config2" "series" : "Atlas" "model" : "Mk 10B" }] }</pre> </div>					

9.1.6.1. Get Device Configuration	
URL	/fleetManagement/deviceConfigurations/{deviceConfigurationId} /fleetManagement/deviceConfigurations/name/{deviceConfigurationName} ⚠
Method	GET
Permission	mdpDeviceConfig [View]

Returns	<p>200 OK</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found</p> <p>500 Internal Server Error</p>
	<p>For auto channel mapping, an empty channels list should be returned.</p> <pre> { "deviceConfiguration": { "deviceConfigurationId": 3, "name": "EDMI Mk10E (Default)", "series": "Atlas", "model": "Mk10E", "description": "The default device configuration for the EDMl MK10E metering device.", "channelSets": [{ "code": "LS1", "description": "_Default Profile Mapping", "channels": [{ "name": "E1", "units": "Wh", "flowDirection": "DELIVERED", "apportionPolicy": "SUM", "phase": "NONE" }, ...] }, { "code": "LS2", "description": "Profile Mapping 2 (Auto)", "channels": [] }, ...], "registerSets": [{ "code": "RegisterSet1", "registers": [{ "name": "Wh Consumption Total", "units": "WH", "flowDirection": "DELIVERED", "phase": "TOTAL", "apportionPolicy": "SUM", "touRate": "TOTAL" }, ...] }], "eventLogs": [{ "eventLogId": 1, "name": "Event Log 1(Default)" }, ...] } } </pre>

9.1.7. List / Search Device Groups

URL	/fleetManagement/deviceGroups					
Method	GET					
Permission	mdpMeterGroup [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	name	Filter by match in device group name	string	Optional		*
	status	Filter by status	string	Optional	Active, Deactivated	*
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					
	<div>Returns</div> <pre>{ "deviceGroups" : [{ "deviceId" : 34, "name" : "Some Meters", "status" : "Active" }, { "deviceId" : 35, "name" : "Some More Meters", "status" : "Deactivated" }, { "deviceId" : 36, "name" : "Parent Meter Group", "status" : "Active" }] }</pre>					

9.1.8. Get Device Group

URL	/fleetManagement/deviceGroups/{deviceId} /fleetManagement/deviceGroups/name/{deviceGroupName}
Method	GET
Permission	mdpMeterGroup [View]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 500 Internal Server Error

```

{
  "deviceGroup" : {
    "deviceGroupId" : 23,
    "name" : "Some Meters",
    "parentDeviceGroupId" : 34,
    "status" : "Active"
  }
}

```

9.1.9. Create Device Group

URL	/fleetManagement/deviceGroups/
Method	POST
Read only attributes	deviceGroupId
Permission	mdpMeterGroup [Create]
Returns	201 Created - includes new device group object and location header containing the URN of the new resource 400 Bad Request (returned if new group is not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 409 Conflict (Device group already exists). Id of conflicting resource will be returned here. 500 Internal Server Error

9.1.10. Update Device Group

URL	/fleetManagement/deviceGroups/{deviceGroupId} /fleetManagement/deviceGroups/name/{deviceGroupName}
Method	PUT
Read only attributes	As per create
Permission	mdpMeterGroup [Modify]
Returns	200 OK (updated object is returned as well) 400 Bad Request (returned if new group is not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 500 Internal Server Error

9.1.11. Delete DeviceGroup

URL	/fleetManagement/deviceGroups/{deviceGroupId} /fleetManagement/deviceGroups/name/{deviceGroupName}
Method	DELETE
Permission	mdpMeterGroup [Delete]

Returns	200 OK
	401 Unauthorized (Token invalid)
	403 Forbidden (Access not allowed)
	404 Not Found
	500 Internal Server Error

9.1.12. List / Search Devices in Device Group

URL	/fleetManagement/deviceGroups/{deviceGroupId}/devices /fleetManagement/deviceGroups/name/{deviceGroupName}/devices					
Method	GET					
Permission	mdpMeterGroup [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	deviceId	Filter by match in device id	Integer	Optional		
	serialNumber	Filter by match in serial number	string	Optional		*
	plantNumber	Filter by match in plant number	string	Optional		*
	limit	Maximum number of devices to return status for in response	Integer	Optional	0 - 5000	5000
	offset	Offset in devices to start returning statuses for in response	Integer	Optional	*	0
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					
	<pre>{ "devices" : [{ "plantNumber" : "Device A", "SerialNumber" : "01234", "deviceId" : 12, }, { "plantNumber" : "Device B", "SerialNumber" : "456F45", "deviceId" : 13, }, { "plantNumber" : "123453", "SerialNumber" : "01222234", "deviceId" : 14, }, ...] }</pre>					

9.1.13. Add Devices to Device Group

URL	/fleetManagement/deviceGroups/{deviceGroupId}/devices /fleetManagement/deviceGroups/name/{deviceGroupName}/devices
-----	---

Method	POST
Permission	mdpMeterGroup [Modify]
Payload	<pre>{ "devices" : [{ "deviceId" : 12, }, { "deviceId" : 13, }, { "deviceId" : 14, }, ...] }</pre> <p>or by device Plant Number</p> <pre>{ "devices" : [{ "plantNumber" : "Meter at Level 1", }, { "plantNumber" : "Meter at Level 2", }, { "plantNumber" : "Meter at Level 3", }, ...] }</pre>
Returns	<p>201 Created</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device group does not exist, or plantNumber not found)</p> <p>500 Internal Error</p>

9.1.14. Remove Device From Device Group	
URL	/fleetManagement/deviceGroups/{ deviceGroupId}/devices /fleetManagement/deviceGroups/name/{ deviceGroupName}/devices
Method	DELETE
Permission	mdpMeterGroup [Modify]

Payload	<pre> { "devices" : [{ "deviceId" : 12, }, { "deviceId" : 13, }, { "deviceId" : 14, }, ...] } or by device Plant Number { "devices" : [{ "plantNumber" : "Meter at Level 1", }, { "plantNumber" : "Meter at Level 2", }, { "plantNumber" : "Meter at Level 3", }, ...] } </pre>
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Device group does not exist, or plantNumber not found) 500 Internal Error

9.1.15. Commission a SMETS device in the system

URL	/fleetManagement/smets/commission/					
Method	POST					
Permission	mdpDevice [Create]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	SerialNumber	Electrical / Gas Meter Serial Number	String	Mandatory		
	IMEI	HUB IMEI Number	String	Mandatory		
	ICCID	SIM ICCID (In case the original linked SIM card doesn't work)	String	Optional		
	NotifyPhoneNumber	Phone number to notify on commissioning success/fail	String	Optional		
Returns	200 OK 400 Bad Request (empty serial number or IMEI) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Serial Number, IMEI, ICCID does not exist) 500 Internal Server Error					

9.1.16. Request ZigBee pairing for SMETS Gas Device

URL	/fleetManagement/smetis/zigbee/pair/gas					
Method	POST					
Permission	mdpDevice [Create]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	IMEI	Master HUB IMEI Number	String	Mandatory		
	SlaveMeterSerialNumber	Slave Meter Serial Number	String	Mandatory		
	InstallCode	HUB Install code number. Min 8 bytes and Max 18 bytes (16 - 36 Hexadecimal values - 0 to F)	String	Optional		
	ZigBeeJoinTime	Open time for ZigBee device to allow HAN device joining (In seconds).	Integer	Optional	Value between 1 and 65534	300
	NotifyPhoneNumber	Phone number to notify on commissioning success/fail	String	Optional		
Returns	200 OK 400 Bad Request (empty IMEI, empty SlaveMeterSerialNumber, invalid InstallCode, out of range ZigBee Join Time) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed or MAC Address was blacklisted) 404 Not Found (IMEI / SlaveMeterSerialNumber not found in the system) 500 Internal Server Error					

9.1.17. Request ZigBee pairing for SMETS IHD device

URL	/fleetManagement/smetis/zigbee/pair/ihd					
Method	POST					
Permission	mdpDevice [Create]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	IMEI	Master HUB IMEI Number	String	Mandatory		
	MACAddress	EUI64/MAC Address of IHD	String	Mandatory		
	InstallCode	HUB Install code number. Min 8 bytes and Max 18 bytes (16 - 36 Hexadecimal values - 0 to F)	String	Optional		
	ZigBeeJoinTime	Open time for ZigBee device to allow HAN device joining (In seconds).	Integer	Optional	Value between 1 and 65534	300
	NotifyPhoneNumber	Phone number to notify on commissioning success/fail	String	Optional		
Returns	200 OK 400 Bad Request (empty IMEI, empty MAC Address, invalid InstallCode, out of range ZigBeeJoinTime) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed or MAC Address was blacklisted) 404 Not Found (IMEI / MACAddress not found in the system) 500 Internal Server Error					

9.1.18. Request ZigBee un-pairing for SMETS Gas Device

URL	/fleetManagement/smets/zigbee/unpair/gas					
Method	POST					
Permission	mdpDevice [Update]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	IMEI	Master HUB IMEI Number	String	Mandatory		
	SlaveMeterSerialNumber	Slave Meter Serial Number	String	Mandatory		
	NotifyPhoneNumber	Phone number to notify on commissioning success/fail	String	Optional		
	Returns	200 OK 400 Bad Request (empty IMEI) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (IMEI / SlaveMeterSerialNumber not found in the system) 500 Internal Server Error				

9.1.19. Request ZigBee un-pairing for SMETS IHD device

URL	/fleetManagement/smets/zigbee/un-pair/ihd					
Method	POST					
Permission	mdpDevice [Update]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	IMEI	Master HUB IMEI Number	String	Mandatory		
	MACAddress	EUI64/MAC Address of IHD	String	Mandatory		
	NotifyPhoneNumber	Phone number to notify on commissioning success/fail	String	Optional		
	Returns	200 OK 400 Bad Request (empty IMEI) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (IMEI / MACAddress not found in the system) 500 Internal Server Error				

9.1.20. Commission a device in the system

URL	/fleetManagement/commission/{deviceId}					
	/fleetManagement/commission/plantNumber/{plantNumber}					
Method	POST					
Permission	mdpDevice [Create]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	siteId	Link device to this site	String	Optional		*

	deviceGroupId	Assign device to this group	Integer	Optional		*
	scheduleId	Run this schedule against the device	String	Optional		*
Returns	200 OK 201 Created - includes location header containing the URN of the created execution (/scheduling/executions/{executionId}) if a schedule id is included 400 Bad Request (Site id, deviceGroupId or schedule id is not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Device id does not exist) 500 Internal Server Error					

9.1.21. Decomission a device in the system

URL	/fleetManagement/decommission/{deviceId} /fleetManagement/decommission/plantNumber/{plantNumber}					
Method	POST					
Permission	mdpDevice [Modify]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	removeFromGroup	Remove device from all groups	Boolean	Optional		*
	deviceGroupId	Assign device to this group	Integer	Optional		*
Returns	200 OK 400 Bad Request (deviceGroup id is not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Device id does not exist) 500 Internal Server Error					

9.1.22. List / Search Commissioning Events

URL	/fleetManagement/events/commissioning					
Method	GET					
Permission	mdpDevice [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	serialNumber	Filter by match in manufacturer serial number	string	Optional		*
	plantNumber	Filter by match in plant number	string	Optional		*
	commissioningCode	Filter by match in Commissioning Code	string	Optional		*
	startDate		date	Mandatory		-
	endDate		date	Mandatory		-
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					

```

{
  "commissioningEvents": [
    {
      "deviceId" : 32,
      "serialNumber" : "211650803",
      "plantNumber" : "ABS33241OR",
        "deviceConfig" : "",
        "modemIMEI" : "",
        "firmwareVersion" : "",
        "firmwareEdition" : 0,
        "firmwareRevision" : 0,
        "IPAddress" : "",
        "ICCID" : "",
        "MSISDN" : "",
        "baseStation" : "",
        "signalStrength" : 0,
      "date" : "2013-01-21T03:34:00",
      "status" : "SUCCESS",
      "commissioningCode" : "3585" ,
      "errorMessage" : ""
    },
    {
      "deviceId" : 33,
      "serialNumber" : "41545456",
      "plantNumber" : "SD4TG93824E",
        "deviceConfig" : "",
        "modemIMEI" : "",
        "firmwareVersion" : "",
        "firmwareEdition" : 0,
        "firmwareRevision" : 0,
        "IPAddress" : "",
        "ICCID" : "",
        "MSISDN" : "",
        "baseStation" : "",
        "signalStrength" : 0,
      "date" : "2013-01-22T12:14:33",
      "status" : "FAILED",
      "commissioningCode" : "" ,
      "errorMessage" : "Signal Strength is too low"
    },...
  ]
}

```

Property	iE2 Import Usage	Format	Valid Values	Description
deviceId	Mandatory	Integer		Multidrive identifier
serialNumber	Mandatory	String		Manufacturer serial number
plantNumber	Mandatory	String		Customer assigned market identifier
modemIMEI	Mandatory	String		Modem IMEI number
firmwareVersion	Mandatory	String		Firmware Version
firmwareEdition	Mandatory	Integer		Firmware Edition
firmwareRevision	Mandatory	Integer		Firmware Revision
IPAddress	Mandatory	String		IP address and port number
ICCID	Mandatory	String		SIM ICC-ID
MSISDN	Mandatory	String		SIM MSISDN (Phone number)
baseStation	Mandatory	String		Base Station information
signalStrength	Mandatory	Integer		Signal Strength
date	Mandatory	String		Date of commissioning event
status	Mandatory	String	PROCESSING, FAILED, SUCCESS	Status of commissioning
commissioningCode	Mandatory	String		Commissioning code assigned by multidrive
errorMessage	Mandatory	String		Error message which caused the commission to fail

9.1.23. List / Search Gas Device Pairing Events

URL	/fleetManagement/events/gasPairing					
Method	GET					
Permission	mdpDevice [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	serialNumber	Filter by match in manufacturer serial number	string	Optional		*
	plantNumber	Filter by match in plant number	string	Optional		*
	commissioningCode	Filter by match in Commissioning Code	string	Optional		*
	startDate		date	Mandatory		-
	endDate		date	Mandatory		-
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					

```
{
  "events": [
    {
      "serialNumber" : "897625142",
      "date" : "2013-01-22T12:14:33",
      "status" : "FAILED",
      "commissioningCode" : "" ,
      "errorMessage" : "Signal Strength is too low"
    },...
  ]
}
```

Property	iE2 Import Usage	Format	Valid Values	Description
serialNumber	Mandatory	String		Serial Number of Gas Meter involved
date	Mandatory	String		Date of commissioning event
status	Mandatory	String	PROCESSING, FAILED, SUCCESS	Status of commissioning
commissioningCode	Mandatory	String		Commissioning code assigned by multidrive
errorMessage	Mandatory	String		Error message which caused the commission to fail

9.1.24. List / Search IHD Device Pairing Events

URL	/fleetManagement/events/IHDPairing					
Method	GET					
Permission	mdpDevice [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	MACAddress	Filter by match in MAC Address	string	Optional		*
	commissioningCode	Filter by match in Commissioning Code	string	Optional		*
	startDate		date	Mandatory		-
	endDate		date	Mandatory		-
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					

```
{
  "events": [
    {
      "MACAddress" : "00905C0000003FF3",
      "date" : "2013-01-22T12:14:33",
      "status" : "FAILED",
      "commissioningCode" : "" ,
      "errorMessage" : "Signal Strength is too low"
    },...
  ]
}
```

Property	iE2 Import Usage	Format	Valid Values	Description
MACAddress	Mandatory	String		MAC Address of In Home Display Device
date	Mandatory	String		Date of commissioning event
status	Mandatory	String	PROCESSING, FAILED, SUCCESS	Status of commissioning
commissioningCode	Mandatory	String		Commissioning code assigned by multidrive
errorMessage	Mandatory	String		Error message which caused the commission to fail

9.1.25. Modem

9.1.25.1. List / Search Modems

URL	/fleetManagement/modem /fleetManagement/modem/{modemid}				
Method	GET				
Permission	mdpModem [View]				
Parameters	Name	Description	Type	Optional/Mandatory	Valid values
	imei	Filter by IMEI number	String	Optional	*
	modemModelName	Filter by modem Model Name	String	Optional	*
	serialNumber	Filter by Serial Number	String	Optional	*
	firmwareVersion	Filter by Firmware Version	String	Optional	*
	status	Filter by Modem Status	String	Optional	WAITING INSTALLATION / INSTALLED / OUT OF SERVICE
	limit	Maximum number of objects to return status for in response	Integer	Optional	0 - 5000
	offset	Offset in objects to start returning statuses for in response	Integer	Optional	*
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error				

```
{
  "modem" : [
    {
      "modemId" : 123,
      "imei" : "356187030007426",
      "modemModelName" : "EWM100",
      "serialNumber" : "AK9390012501300",
      "firmwareVersion" : "R7.45.1",
      "status" : "Installed"
    }, ...
  ]
}
```

Description

Property	iE2 Import Usage	Format	Valid Values	Description
modemId	Mandatory	Integer		Modem Auto Id
imei	Mandatory	String		modem IMEI (http://en.wikipedia.org/wiki/International_Mobile_Station_Identity)
status	Mandatory	String	WAITING INSTALLATION / INSTALLED / OUT OF SERVICE	Status of Modem
modemModelName	Optional	String		Modem Model Name
serialNumber	Optional	String		Modem Serial Number
firmwareVersion	Optional	String		Modem Firmware Version

9.1.25.2. Create Modem

URL	/fleetManagement/modem
Method	POST
Permission	mdpModem [Create]
Payload	MDX Modem object same as in Get Modem operation
Read only attributes	modemId
Returns	201 Created - includes created object and location header containing the URN of the new resource (/modem/{modemid}) 400 Bad Request (returned if new modem is not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 409 Conflict (IMEI already exists) 500 Internal Server Error

9.1.25.3. Delete Modem

URL	/fleetManagement/modem/{modemId} /fleetManagement/modem/imei/{imei}
-----	--

Method	DELETE
Permission	mdpModem [Delete]
Returns	200 OK
	401 Unauthorized (Token invalid)
	403 Forbidden (Access not allowed)
	404 Not Found

9.1.25.4. Update Modem

URL	/fleetManagement/modem/{modemId} /fleetManagement/modem/imei/{imei}
Method	PUT
Permission	mdpModem [Modify]
Read only attributes	modemId
Parameters/Payload	As per create
Returns	200 OK (this will return the updated modem details) 400 Bad Request (modem changes are not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 409 Conflict (IMEI already exists) 500 Internal Server Error

9.1.26. SIM Cards

9.1.26.1. List / Search SIMs

URL	/fleetManagement/sim /fleetManagement/sim/{simId}					
Method	GET					
Permission	mdpSIM [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	iccid	Filter by ICCID	String	Optional	*	
	imsi	Filter by IMSI	String	Optional	*	
	msisdn	Filter by MSISDN	String	Optional	*	
	status	Filter by SIM Status	String	Optional	ACTIVATED, DEACTIVATED	
	limit	Maximum number of objects to return status for in response	Integer	Optional	0 - 5000	5000
	offset	Offset in objects to start returning statuses for in response	Integer	Optional	*	0

Returns	<div>200 OK</div> <div>401 Unauthorized (Token invalid)</div> <div>403 Forbidden (Access not allowed)</div> <div>500 Internal Server Error</div>																																																							
	<div><pre>{ "sim" : [{ "simId" : 23, "networkProviderName": "Singtel", "dataPlanName" : "4G Plus Package", "iccid" : "1234", "imsi" : "5678", "msisdn" : "6567567538", "mcc" : "65", "mnc" : "02", "country" : "Singapore", "status" : "ACTIVATED" }, ...] }</pre></div> <div>Description</div> <table><tr><th>Property</th><th>iE2 Import Usage</th><th>Format</th><th>Valid Values</th><th>Description</th></tr><tr><td>simId</td><td>Mandatory</td><td>Integer</td><td></td><td>SIM Auto Id</td></tr><tr><td>iccid</td><td>Mandatory</td><td>String</td><td></td><td>Unique Serial Number of SIM Card (http://en.wikipedia.org/wiki/ICCID#ICCID)</td></tr><tr><td>status</td><td>Mandatory</td><td>String</td><td>ACTIVATED, DEACTIVATED</td><td>Status of SIM Card</td></tr><tr><td>networkProviderName</td><td>Optional</td><td>String</td><td></td><td>Name of Network Provider / Telco</td></tr><tr><td>dataPlanName</td><td>Optional</td><td>String</td><td></td><td>Data Subscription Plan Name</td></tr><tr><td>imsi</td><td>Optional</td><td>String</td><td></td><td>International Mobile Subscriber Identity (http://en.wikipedia.org/wiki/IMSI)</td></tr><tr><td>msisdn</td><td>Optional</td><td>String</td><td></td><td>MSISDN (http://en.wikipedia.org/wiki/MSISDN)</td></tr><tr><td>mcc</td><td>Optional</td><td>String</td><td></td><td>Mobile Country Code (http://en.wikipedia.org/wiki/Mobile_country_code)</td></tr><tr><td>mnc</td><td>Optional</td><td>String</td><td></td><td>Mobile Network Code (http://en.wikipedia.org/wiki/Mobile_country_code)</td></tr><tr><td>country</td><td>Optional</td><td>String</td><td></td><td>Country</td></tr></table>	Property	iE2 Import Usage	Format	Valid Values	Description	simId	Mandatory	Integer		SIM Auto Id	iccid	Mandatory	String		Unique Serial Number of SIM Card (http://en.wikipedia.org/wiki/ICCID#ICCID)	status	Mandatory	String	ACTIVATED, DEACTIVATED	Status of SIM Card	networkProviderName	Optional	String		Name of Network Provider / Telco	dataPlanName	Optional	String		Data Subscription Plan Name	imsi	Optional	String		International Mobile Subscriber Identity (http://en.wikipedia.org/wiki/IMSI)	msisdn	Optional	String		MSISDN (http://en.wikipedia.org/wiki/MSISDN)	mcc	Optional	String		Mobile Country Code (http://en.wikipedia.org/wiki/Mobile_country_code)	mnc	Optional	String		Mobile Network Code (http://en.wikipedia.org/wiki/Mobile_country_code)	country	Optional	String		Country
Property	iE2 Import Usage	Format	Valid Values	Description																																																				
simId	Mandatory	Integer		SIM Auto Id																																																				
iccid	Mandatory	String		Unique Serial Number of SIM Card (http://en.wikipedia.org/wiki/ICCID#ICCID)																																																				
status	Mandatory	String	ACTIVATED, DEACTIVATED	Status of SIM Card																																																				
networkProviderName	Optional	String		Name of Network Provider / Telco																																																				
dataPlanName	Optional	String		Data Subscription Plan Name																																																				
imsi	Optional	String		International Mobile Subscriber Identity (http://en.wikipedia.org/wiki/IMSI)																																																				
msisdn	Optional	String		MSISDN (http://en.wikipedia.org/wiki/MSISDN)																																																				
mcc	Optional	String		Mobile Country Code (http://en.wikipedia.org/wiki/Mobile_country_code)																																																				
mnc	Optional	String		Mobile Network Code (http://en.wikipedia.org/wiki/Mobile_country_code)																																																				
country	Optional	String		Country																																																				

9.1.26.2. Create SIM

URL	/fleetManagement/sim
Method	POST
Permission	mdpSIM [Create]
Payload	MDX SIM object same as in Get SIM operation
Read only attributes	simId

Returns	201 Created - includes created object and location header containing the URN of the new resource (/sim/{simid})
	400 Bad Request (returned if new sim is not valid)
	401 Unauthorized (Token invalid)
	403 Forbidden (Access not allowed)
	409 Conflict (ICCID already exists)
	500 Internal Server Error

9.1.26.3. Delete SIM

URL	/fleetManagement/sim/{simId} /fleetManagement/sim/iccid/{iccid}
Method	DELETE
Permission	mdpSIM [Delete]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found

9.1.26.4. Update SIM

URL	/fleetManagement/sim/{simId} /fleetManagement/sim/iccid/{iccid}
Method	PUT
Permission	mdpSIM [Modify]
Read only attributes	simId
Parameters/Payload	As per create
Returns	200 OK (this will return the updated sim details) 400 Bad Request (sim changes are not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 409 Conflict (ICCID already exists) 500 Internal Server Error

9.1.27. Gas Calorific Value

9.1.27.1. List / Search Gas Calorific Value

URL	/fleetManagement/device/gas/calorificvalue /fleetManagement/device/gas/calorificvalue/{calorificValueId}					
Method	GET					
Permission	mdpDevice [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default

	name	Unique Identifier of Gas Calorific Value	String	Optional	*																										
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error																														
	<div><pre>{ "calorificValue" : [{ "calorificValueId" : 123, "name": "Calorific Group 1", "conversionFactor " : 2.83, "calorificValue" : 39.47333 }, ...] }</pre></div> <p>Description</p> <table><tr><th>Property</th><th>iE2 Import Usage</th><th>Format</th><th>Valid Values</th><th>Description</th></tr><tr><td>calorificValueId</td><td>Mandatory</td><td>Integer</td><td></td><td>Calorific Value Auto Id</td></tr><tr><td>name</td><td>Mandatory</td><td>String</td><td></td><td>Unique Identifier of the calorific group</td></tr><tr><td>conversionFactor</td><td>Mandatory</td><td>Float</td><td></td><td>Conversion Factor</td></tr><tr><td>calorificValue</td><td>Mandatory</td><td>Float</td><td></td><td>Calorific Value</td></tr></table>						Property	iE2 Import Usage	Format	Valid Values	Description	calorificValueId	Mandatory	Integer		Calorific Value Auto Id	name	Mandatory	String		Unique Identifier of the calorific group	conversionFactor	Mandatory	Float		Conversion Factor	calorificValue	Mandatory	Float		Calorific Value
Property	iE2 Import Usage	Format	Valid Values	Description																											
calorificValueId	Mandatory	Integer		Calorific Value Auto Id																											
name	Mandatory	String		Unique Identifier of the calorific group																											
conversionFactor	Mandatory	Float		Conversion Factor																											
calorificValue	Mandatory	Float		Calorific Value																											

9.1.27.2. Create Gas Calorific Value

URL	/fleetManagement/device/gas/calorificvalue
Method	POST
Permission	mdpDevice [Create]
Payload	MDX Gas Calorific Value object same as in List / Search Gas Calorific Value
Read only attributes	calorificValueId
Returns	201 Created - includes created object and location header containing the URN of the new resource (device/gas/calorificvalue/{calorificValueId}) 400 Bad Request (returned if new Gas Calorific Value is not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 409 Conflict (name already exists) 500 Internal Server Error

9.1.27.3. Delete Gas Calorific Value

URL	/fleetManagement/device/gas/calorificvalue/{calorificValueId} /fleetManagement/device/gas/calorificvalue/name/{name}
Method	DELETE

Permission	mdpDevice [Delete]
Returns	200 OK
	401 Unauthorized (Token invalid)
	403 Forbidden (Access not allowed)
	404 Not Found

9.1.27.4. Update Gas Calorific Value

URL	/fleetManagement/device/gas/calorificvalue/{calorificValueId} /fleetManagement/device/gas/calorificvalue/name/{name}
Method	PUT
Permission	mdpDevice [Modify]
Read only attributes	calorificValueId
Parameters/Payload	As per create
Returns	200 OK (this will return the updated calorific value details) 400 Bad Request (calorific value changes are not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 409 Conflict (name already exists) 500 Internal Server Error

10. Scheduling API

10.1.1. Schedules

10.1.1.1. List / Search Schedules

URL	/scheduling/schedules					
Method	GET					
Permission	mdpSchedule [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	name	Filter by match in schedule name	string	Optional		*
	status	Filter by schedule status, whether the schedule is activated or needs to be manually triggered	string	Optional	Active, Manual	*
	deviceId	Filter by schedules associated with this device	Integer	Optional		

	scheduleIncludesFeatures	Limits inclusion of schedules which perform.	string	Optional	<i>List of comma separated operations</i> TOU, LOAD_PROFILE, SETTIME, SETUP, FIRMWARE	*
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					

```

{
  "schedules" : [
    {
      "scheduleId" : 12,
      "name" : "Important Read Schedule 1",
      "lastExecution" :
      {
        "executionId" : 14,
        "startDate" : "2013-01-21T10:00:00",
        "endDate" : "2013-01-21T12:00:00",
        "jobsWithSuccess" : 3,
        "jobsWithFailure" : 5,
        "status" : "Complete",
        "hasWarning" : "true",
        "connectAttempts" : 1
      }
      "nextRun" : "2013-01-22T10:00:00",
      "status" : "Active",
      "maxRetries" : 2
    },
    {
      "scheduleId" : 14,
      "name" : "Important Read Schedule 2",
      "lastExecution" :
      {
        "executionId" : 0,
        "startDate" : "",
        "endDate" : "",
        "jobsWithSuccess" : 0,
        "jobsWithFailure" : 0,
        "status" : "",
        "hasWarning" : "false",
        "connectAttempts" : 0
      }
      "nextRun" : "",
      "status" : "Manual Trigger",
      "maxRetries" : 3
    },
    {
      "scheduleId" : 15,
      "name" : "Important Read Schedule 3",
      "lastExecution" :
      {
        "executionId" : 14,
        "startDate" : "2013-01-21T10:00:00",
        "endDate" : "2013-01-21T12:00:00",
        "jobsWithSuccess" : 3,
        "jobsWithFailure" : 5,
        "status" : "Failed",
        "hasWarning" : "true",
        "connectAttempts" : 2,
      }
      "nextRun" : "2013-01-22T10:00:00",
      "status" : "Active",
      "maxRetries" : 3
    }
  ]
}

```

10.1.1.2. Get Schedule

URL	<div>/scheduling/schedules/{scheduleId}</div> <div>/scheduling/schedules/name/{scheduleName} ⚠</div>																																
Method	GET																																
Permission	mdpSchedule [View] or mdpScheduleRunNow[View]																																
Returns	<div>200 OK</div> <div>401 Unauthorized (Token invalid)</div> <div>403 Forbidden (Access not allowed)</div> <div>404 Not Found</div> <div>500 Internal Server Error</div>																																
	<div><pre>{ "schedule" : { "scheduleId" : 34, "name" : "Important Read Schedule", "recurrence" : "Everyday at 10:00", "lastRun" : "2013-01-21T10:00:00", "nextRun" : "2013-01-22T10:00:00", "status" : "Active" } }</pre></div> <table><tr><th>Property</th><th>Usage</th><th>Format</th><th>Description</th></tr><tr><td>scheduleId</td><td>Mandatory</td><td>Integer</td><td>A unique identifier for the schedule.</td></tr><tr><td>name</td><td>Mandatory</td><td>Text</td><td>A descriptive name for the schedule.</td></tr><tr><td>recurrence</td><td>Optional</td><td>?</td><td>Description? of recurrence rule for schedules that are set to run automatically. Omitted for schedules that are only triggered manually/via the API.</td></tr><tr><td>lastRun</td><td>Optional</td><td>ISO8601</td><td>The date and time the schedule was last run. Omitted if the schedule has never been triggered.</td></tr><tr><td>nextRun</td><td>Optional</td><td>ISO8601</td><td>The next date and time the schedule is set to run (if the schedule is set to run automatically).</td></tr><tr><td>status</td><td>Mandatory</td><td>Active, Manual</td><td><i>Active</i> indicates the schedule is set to run automatically by the system. <i>Manual</i> indicates the schedule can only be triggered by an external request.</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	Property	Usage	Format	Description	scheduleId	Mandatory	Integer	A unique identifier for the schedule.	name	Mandatory	Text	A descriptive name for the schedule.	recurrence	Optional	?	Description? of recurrence rule for schedules that are set to run automatically. Omitted for schedules that are only triggered manually/via the API.	lastRun	Optional	ISO8601	The date and time the schedule was last run. Omitted if the schedule has never been triggered.	nextRun	Optional	ISO8601	The next date and time the schedule is set to run (if the schedule is set to run automatically).	status	Mandatory	Active, Manual	<i>Active</i> indicates the schedule is set to run automatically by the system. <i>Manual</i> indicates the schedule can only be triggered by an external request.				
Property	Usage	Format	Description																														
scheduleId	Mandatory	Integer	A unique identifier for the schedule.																														
name	Mandatory	Text	A descriptive name for the schedule.																														
recurrence	Optional	?	Description? of recurrence rule for schedules that are set to run automatically. Omitted for schedules that are only triggered manually/via the API.																														
lastRun	Optional	ISO8601	The date and time the schedule was last run. Omitted if the schedule has never been triggered.																														
nextRun	Optional	ISO8601	The next date and time the schedule is set to run (if the schedule is set to run automatically).																														
status	Mandatory	Active, Manual	<i>Active</i> indicates the schedule is set to run automatically by the system. <i>Manual</i> indicates the schedule can only be triggered by an external request.																														

10.1.2. Executions

10.1.2.1. Trigger Regular Schedule As Configured

URL	/scheduling/schedules/{scheduleId}/executions /scheduling/schedules/name/{scheduleName}/executions ⚠
Method	POST
Permission	mdpScheduleRunNow [View]
Description	<p>Triggers a schedule to run as configured in the system using the device groups it has been configured for. This does not run the schedule on any user defined devices or groups.</p> <p>Notification is an optional element and when present will mean that a notification will be POSTed to the URI provided based on the notificationEvent requested. If not requiring notification (using polling instead) then payload can be Empty.</p>

Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } } </pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } } </pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>201 Created - includes location header containing the URN of the created execution (/scheduling/executions/{executionId})</p> <p>400 Bad Request (Payload is invalid or not well formed or device ids are be invalid)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found</p> <p>500 Internal Server Error</p>

10.1.2.2. Trigger Adhoc Schedule for particular devices

URL	/scheduling/schedules/{scheduleId}/executions /scheduling/schedules/name/{scheduleName}/executions ⚠
Method	POST
Permission	mdpScheduleRunNow [View]

Description	<p>Triggers a schedule to run on a particular set of devices (from payload below) and ignores the preconfigured device groups setup in the system.</p> <p>Notification is an optional element and when present will mean that a notification will be POSTed to the URI provided based on the notificationEvent requested. If not requiring notification (using polling instead) then payload can be Empty.</p>
Payload	<pre> { "devices" : [{ "deviceId" : 1, }, { "deviceId" : 2, },...] } </pre>
Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } } </pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } } </pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>

Returns	201 Created - includes location header containing the URN of the created execution (/scheduling/executions/{executionId}) 400 Bad Request (Payload is invalid or not well formed or device ids are be invalid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Schedule id does not exist) 500 Internal Server Error
---------	--

10.1.2.3. List Schedule Executions

URL	/scheduling/executions /scheduling/schedules/{scheduleID}/executions (filtered by schedule Id) /scheduling/schedules/name/{scheduleName}/executions (filtered by schedule Name) ⚠					
Method	GET					
Permission	mdpScheduleExecutions [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	hasWarning	Filter by has warning status	boolean	Optional	True, False	*
	limit ⚠	Maximum number of devices to return status for in response	Integer	Optional	0 - 5000	5000
	offset ⚠	Offset in devices to start returning statuses for in response	Integer	Optional	*	0
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error					

```

{
  "executions" : [
    {
      "executionId" : 14,
      "scheduleId" : 12,
      "scheduleName" : "Important Read Schedule",
      "startDate" : "2013-01-21T10:00:00",
      "endDate" : "2013-01-21T12:00:00",
      "jobsWithSuccess" : 3,
      "jobsWithFailure" : 5,
      "status" : "Complete",
      "hasWarning" : "true"
    },
    {
      "executionId" : 15,
      "scheduleId" : 54,
      "scheduleName" : "Another Important Read Schedule",
      "startDate" : "2013-01-21T10:00:00",
      "endDate" : "2013-01-21T12:00:00",
      "jobsWithSuccess" : 0,
      "jobsWithFailure" : 5,
      "status" : "Complete",
      "hasWarning" : "true"
    },
    {
      "executionId" : 83,
      "scheduleId" : 54,
      "scheduleName" : "Unimportant schedule...",
      "startDate" : "2013-01-21T10:00:00",
      "endDate" : "",
      "jobsWithSuccess" : 0,
      "jobsWithFailure" : 0,
      "status" : "Pending",
      "hasWarning" : "true"
    },
    ...
  ]
}

```

10.1.2.4. Get Schedule Execution

URL	/scheduling/executions/{executionId}
Method	GET
Permission	mdpScheduleExecutions [View]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 500 Internal Server Error

```
{
  "execution": {
    "executionId" : 14,
    "scheduleId" : 12,
    "startDate" : "2013-01-21T10:00:00",
    "endDate" : "2013-01-21T12:00:00",
    "jobsWithSuccess" : 3,
    "jobsWithFailure" : 5,
    "status" : "Complete",
    "hasWarning" : "true"
  }
}
```

Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused

10.1.3. Execution Jobs

10.1.3.1. Search for Schedule Execution Jobs

URL	/scheduling/executionJobs /scheduling/executions/{executionId}/executionJobs (filtered by execution) /scheduling/schedules/{scheduleId}/executions/{executionID}/executionJobs (filtered by schedule id and execution) /scheduling/schedules/name/{scheduleName}/executions/{executionID}/executionJobs (filtered by schedule name and execution) ⚠️					
Method	GET					
Permission	mdpScheduleJobs [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	status	Filter by job status	string	Optional	Queued, Running, Complete, Failed	*
	plantNumber	Filter by plant number	string	Optional		*
	siteCode	Filter by site code	string	Optional		*
	limit ⚠️	Maximum number of devices to return status for in response	Integer	Optional	0 - 5000	5000
	offset ⚠️	Offset in devices to start returning statuses for in response	Integer	Optional	*	0
Returns	200 OK 400 Bad Request (one of the parameters are not valid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (Execution id not found or schedule not found) 500 Internal Server Error					

```

{
  "executionJobs" : [
    {
      "executionJobId" : 9875,
      "executionId" : 125,
      "deviceId" : 32,
      "plantNumber" : "Device01",
      "date" : "2013-01-21T10:00:00",
      "commsServer" : "Communications Pool",
      "status" : "Success",
      "hasWarning" : "true",
      "lastError" : "Communications Device Failure: System error:Connection
refused, Unable to connect to 192.168.3.20 on port 10002",
    },
    {
      "executionJobId" : 9876,
      "executionId" : 125,
      "deviceId" : 32,
      "plantNumber" : "Device01",
      "date" : "",
      "commsServer" : "",
      "status" : "Pending",
      "hasWarning" : "false",
      "lastError" : ""
    },...
  ]
}

```

10.1.3.2. Get Schedule Execution Job

URL	/scheduling/executionJobs/{executionJobId}
Method	GET
Permission	mdpScheduleJobs [View]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found 500 Internal Server Error

```
{
  "executionJob": {
    "executionJobId" : 9875,
    "executionId" : 125,
    "deviceId" : 32,
    "plantNumber" : "ADEVICE",
    "date" : "2013-01-21T10:00:00",
    "status" : "Success"
  }
}
```

Status can be:

- Unknown
 - Anything unhandled will report this status
- Queued
- Running
- Success
- Failed
 - Incorporates Failed, Remote Fail, Fail Use Resend, Fail Use 2nd Comm and Fail Use Mobile status from MultiDrive
- Error
- Expired
- Retry
- Cancelled
- Waiting
 - Incorporates Listen and Ready To Send from MultiDrive

10.1.3.3. Get Schedule Execution Job Log

URL	/scheduling/executionJobs/{executionJobId}/scheduleLog
Method	GET
Permission	mdpScheduleJobs [View]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (No schedule log was stored) 500 Internal Server Error

Returns

```
{
  "scheduleLog" : [
    {
      "entryDate" : "2013-01-25T09:56:09+0000",
      "logEntry" : "Connecting"
    },
    {
      "entryDate" : "2013-01-25T09:57:09+0000",
      "logEntry" : "Login"
    },
    {
      "entryDate" : "2013-01-25T09:58:09+0000",
      "logEntry" : "Timeout on meter "124000001""
    },
    {
      "entryDate" : "2013-01-25T09:59:09+0000",
      "logEntry" : "Rescheduled for 2013-01-24 18:20:41.432"
    },...
  ]
}
```

10.1.3.4. Get Schedule Execution Communications Log

URL	/scheduling/executionJobs/{executionJobId}/communicationsLog
Method	GET
Permission	mdpScheduleJobs [View]
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (No communications log was stored) 500 Internal Server Error

```
{
  "communicationsLog" : [
    {
      "entryDate" : "2013-01-25T09:56:09+0000",
      "data" : "8F4BA6945A1F74C5791C45444D49",
      "transmitted" : true,
    },
    {
      "entryDate" : "2013-01-25T09:57:09+0000",
      "data" : "8F4B2029EEC50C851CA6C06B",
      "transmitted" : false,
    },
    {
      "entryDate" : "2013-01-25T09:58:09+0000",
      "data" : "8F4BBE80E62347CF2CFB45444D49 ",
      "transmitted" : true,
    },
    {
      "entryDate" : "2013-01-25T09:59:09+0000",
      "data" : "8F066B60",
      "transmitted" : false,
    },...
  ]
}
```

10.1.3.5. Description

Property	Format	Usage	Description
entryDate	ISO8601	Mandatory	Date and time data was sent/received,
data	Text	Mandatory	Raw data sent/received.
transmitted	Boolean	Mandatory	<i>true</i> - data was sent to the device <i>false</i> - data was received from the device

11. Prepayment API

11.1.1.1. Top-up prepayment credit

URL	/prepayment/credit/{deviceId} /prepayment/credit/plantNumber/{plantNumber}
Method	POST
Permission	mdpPrepaymentTopup [View]
Payload	{ "prepaymentCredit" : { "dollarValue" : 1.00, }, }

Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } } </pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } } </pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter</p> <p>/scheduling/executions/{executionId}, This/scheduling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number or device group id does not exist)</p> <p>500 Internal Error</p>

11.1.1.2. Request prepayment information for a device

URL	<pre> /prepayment/status/{deviceId} /prepayment/status/plantNumber/{plantNumber} </pre>
Method	POST
Permission	mdpPrepaymentInformation [View]

Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } }</pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } }</pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter</p> <p>/scheduling/executions/{executionId}, This/schedoling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>400 Bad Request (function not supported by the device type)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number or device group id does not exist)</p> <p>500 Internal Error</p>

11.1.1.3. Get prepayment information for a device

URL	<pre> /prepayment/status/{deviceId} /prepayment/status/plantNumber/{plantNumber}</pre>
Method	GET
Permission	mdpPrepaymentInformation [View]

Returns	<p>200 OK</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number or device group id does not exist)</p> <p>500 Internal Server Error</p> <pre> { "prepaymentStatus" : { "readingDate" : "2013-01-21T10:00:00", "prepaymentMode" : "credit", "currency" : "dollars", "creditType" : "", "normalCredit" : 1.00, "emergencyCredit" : 5.00, "fuelDebt" : 1.00, "nonFuelDebt" : 1.33, "totalCreditAdded" : 10.00, "miscInfo" : "", }, }</pre>
---------	---

Property	Format	Valid Values	Description
readingDate	DateTime		Date & time the prepayment information was read out from meter
prepaymentMode	Text	credit, debit	Indicates if prepayment mode is activated or not. debit = prepayment active credit = prepayment disabled
currency	Text	dollars, euros	Type of currency in use
creditType	Text		
normalBalance	Double		Current normal credit available
emergencyBalance	Double		Current emergency credit available
fuelDebt	Double		Current fuel debt remaining
nonFuelDebt	Double		Current non-fuel debt remaining
totalCreditAdded	Double		Total credit added to the meter since started
miscInfo	Text		Misc information about prepayment

11.1.1.4. Write misc. prepayment info to meter

URL	<pre> /prepayment/misc/{deviceId} /prepayment/misc/plantNumber/{plantNumber} </pre>
Method	POST
Permission	mdpPrepaymentTopup [View]
Note	All items in payload are optional.

Payload

```
{
  "creditNotification" : //Only available for Atlas 1 series meter for
the moment
  {
    "creditBalance" : 5600, //$56.00, Integer with two implied
decimal places
    "remainingDays" : 10 //credit remaining days
  }
}

/* removed due to design change
  "prepaymentSetup" : //Only available for Atlas 2 series meter for the
moment
  {
    //Non Fuel Debt
    "nonFuelDebt" : 0.00,
    "nonFuelDebtRecoveryPeriodInSec": 0.00,
    "nonFuelDebtRecoveryRate": 0.00,
    "nonFuelDebtRemainingBalance" : 0.00,
    //Fuel Debt
    "fuelDebt" : 0.00,
    "fuelDebtRecoveryPeriodInSec" : 0.00,
    "fuelDebtRecoveryRate" : 0.00,
    "FuelDebtRemainingBalance" : 0.00,
    //Others
    "modeOfOperation" : 0,
    "emergencyCredit" : 0.00,
    "creditBalance" : 0.00,
    "paymentDebtPercentage" : 0.00
  }
}

*/
```

Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } } </pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } } </pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter</p> <p>/scheduling/executions/{executionId}, This/scheduling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>400 Bad Request (payload is empty, payload is invalid, or function not supported by the device type)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number does not exist)</p> <p>500 Internal Error</p>

11.1.1.5. Pass prepayment token to meter

URL	<pre> /prepayment/token/{deviceId} /prepayment/token/plantNumber/{plantNumber} </pre>
Method	POST
Permission	mdpPrepaymentTopup [View]

Payload	<pre>{ "prepaymentTokens" : [{ "token" : "azxjv0asjtoaj9ghaond", }, { "token" : "yxp4anywyxnqdg9hajln", }, { "token" : "exhwngfuexd5eg5xzgc5", }] }</pre>
Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre>{ "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } }</pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre>{ "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } }</pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter</p> <p>/scheduling/executions/{executionId}, This/scheduling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number or device group id does not exist)</p> <p>500 Internal Error</p>

12. Consumer API

12.1.1.1. Notify Consumer via Device's Local Display

URL	<code>/consumer/deviceMessage/{deviceId}</code> <code>/consumer/deviceMessage/plantNumber/{plantNumber}</code> <code>/consumer/deviceMessage/deviceGroup/{deviceId}</code>
Method	POST
Permission	mdpScheduleRunNow [View]
Payload	<pre>{ "deviceMessage" : { "message" : "Hello", }, }</pre>
Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre>{ "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } }</pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre>{ "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } }</pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>

Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter /scheduling/executions/{executionId}, This/scheduling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>400 Bad Request (message is too long, maximum length is 20 characters)</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id, plant number or device group id does not exist)</p> <p>500 Internal Error</p>
---------	--

13. Diagnostic API

13.1.1.1. Read diagnostic register	
URL	<p>/diagnostic/readRegister/{deviceId}</p> <p>/diagnostic/readRegister/plantNumber/{plantNumber}</p>
Method	POST
Permission	mdpDiagnosticRegister [View]
Payload	<pre>{ "diagnosticRegisters": [{ "register": "F002", "dataType": "String" },...] }</pre> <p>Valid Date Types -</p> <ul style="list-style-type: none"> • Byte • String • Int16 • UInt16 • Int32 • UInt32 • Float • Double • Date • Time • DateTime • Binary <p>Data Value Formats</p> <ul style="list-style-type: none"> • String always as a normal string. • Byte, Int16, UInt16, Int32, UInt32 as normal numbers • Float, Double as normal numbers with decimal places • Date, Time, DateTime always in ISO8601 • Binary always as a hex string

Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } } </pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } } </pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter</p> <p>/scheduling/executions/{executionId}, This/scheduling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id does not exist)</p> <p>500 Internal Server Error</p>

13.1.1.2. Get diagnostic register value read out

URL	/diagnostic/readRegister					
Method	GET					
Permission	mdpDiagnosticRegister [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	executionJobId	Execution job id to that register read ran in	string	Mandatory		
	register	Meter register to return data for in HEX (ie. F002)	string	Optional		*

Returns	200 OK 400 Bad Request (register is invalid) 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 404 Not Found (execution job id or register does not exist) 500 Internal Server Error
---------	---

AKA Diagnostic Registers. Note diagnostic registers are not supported by iE2 at this time.

MDX Export example

```

<diagnosticDataSets>
  <diagnosticDataSet>
    <recordDateTime>2012-06-10T12:30:00.000+0000</recordDateTime>
    <diagnosticValues>
      <diagnosticValue>
        <registerNumber>F061</registerNumber>
        <type>DateTime</type>
        <value>2012-06-10T10:30:00.000+0000</value>
      </diagnosticValue>
      <diagnosticValue>
        <registerNumber>2020</registerNumber>
        <type>Float</type>
        <value>123</value>
      </diagnosticValue>
    </diagnosticValues>
  </diagnosticDataSet>
</diagnosticDataSets>

```

Full diagnostic value data details

```

{
  "diagnosticDataSets": [{
    "recordDateTime": "2012-06-10T12:30:00.000+0000",
    "diagnosticValues": [{
      "registerNumber": "F061",
      "type": "DateTime",
      "value": "2012-06-10T10:30:00.000+0000"
    }, {
      "registerNumber": "2020",
      "type": "Float",
      "value": "123"
    }
  ]
}]
}

```

13.1.1.3. Description

Property			iE2 Import Usage	Format	Description
diagnosticDataSet			Unused	-	
	recordDateTime		Unused	ISO8601	The timestamp when the values were recorded
	diagnosticValues		Unused	-	Values for a single diagnostic register
		registerNumber	Unused	Text	The register number being recorded
		type	Unused	See data types below	The datatype from this register
		value	Unused	Text	The value from this register

13.1.2. Data types

Type	Format
Byte	Number
String	String
Int16	Number
UInt16	Number
Int32	Number
UInt32	Number
Float	Number (with decimal places)
Double	Number (with decimal places)
Date	ISO8601
Time	ISO8601
DateTime	ISO8601
Binary	Hex string

13.1.2.1. Write diagnostic register

URL	/diagnostic/writeRegister/{deviceId} /diagnostic/writeRegister/plantNumber/{plantNumber}
Method	POST
Permission	mdpDiagnosticRegister [Create]

Payload	<pre>{ "diagnosticRegisters" : [{ "register" : "F002", "dataType" : "String", "dataValue" : "Hello", },...] }</pre> <p>Valid Date Types -</p> <ul style="list-style-type: none">• Byte• String• Int16• UInt16• Int32• UInt32• Float• Double• Date• Time• DateTime• Binary <p>Data Value Formats</p> <ul style="list-style-type: none">• String always as a normal string.• Byte, Int16, UInt16, Int32, UInt32 as normal numbers• Float, Double as normal numbers with decimal places• Date, Time, DateTime always in ISO8601• Binary always as a hex string
---------	--

Optional Payload	<p>Call back notification is an optional element and when present will mean that a notification will be POSTed to the URI provided. If not requiring notification (using polling instead) then payload can be Empty.</p> <p>Payload</p> <p>Additional parameter to be added to existing payload if callback is required:</p> <pre> { "notification": { "uri": "http://someserver/notifyme/1" }, "notifyStatus": { "status": ["Complete", "No Jobs"] } } </pre> <p>Notification</p> <p>URI passed in will be called back as a POST request.</p> <p>Payload will be the same as Get Schedule Execution (/scheduling/executions/{executionId})</p> <pre> { "execution": { "executionId" : 14, "scheduleId" : 12, "startDate" : "2013-01-21T10:00:00", "endDate" : "2013-01-21T12:00:00", "jobsWithSuccess" : 3, "jobsWithFailure" : 5, "status" : "Complete", "hasWarning" : "true" } } </pre> <p>Status can be Unknown, Queued, Running, Complete, No Jobs, Cancelled, Paused</p>
Returns	<p>202 Accepted - includes location header to schedule execution history to determine if it has been sent to the meter</p> <p>/scheduling/executions/{executionId}, This/scheduling/executions/{executionId} endpoint will return a 200 OK with a representation of the commands status, or a when the device has successfully received the command.</p> <p>401 Unauthorized (Token invalid)</p> <p>403 Forbidden (Access not allowed)</p> <p>404 Not Found (Device Id does not exist)</p> <p>500 Internal Server Error</p>

14. System API

14.1.1.1. Request an authentication token for the system	
URL	/system/authToken
Method	POST
Payload	None
Permission	NA

Authentication	This call requires basic HTTP authentication credentials to be set
Returns	200 OK 400 Bad Request (Request is not properly formed) 401 Unauthorized (Username/password is invalid) 403 Forbidden (Username has been disabled) 500 Internal Server Error <pre>{ "authToken" : { "token" : "a0a8b777cd73739e99d000dfff", } }</pre>

The API is secured through the the use of SSL and proper certificates to secure the connection and basic HTTP authentication to authorize users.

Basic HTTP authentication is done using the username/password of the user as current stored in MultiDrive. This also provides the access capabilities for the user.

All users must first request an access token from the System Management API. That token then must be passed in with all API calls.

The token is entered as part of the HTTP Authorization header:

Authorization: AuthToken "Token = <token>"

Example: Authorization: AuthToken "Token = c1a0d63e-18b6-46df-bbda-5cb0282cecd8"

All API calls must validate the token in the system before allowing any request.

- Any request for which the token does not exist (or has been deleted) will respond with a "HTTP 401 Unauthorised" response.
- Any request for which the token indicates that the user is not allowed to access will respond with a "HTTP 403 Forbidden" response.

14.1.1.2. Delete Authentication Token

URL	/system/authToken/{token}					
Method	DELETE					
Authentication	This call requires basic HTTP authentication credentials to be set					
Returns	200 OK 400 Bad Request (Request is incorrectly formed) 401 Unauthorized (username/password is invalid) 403 Forbidden (Username is disabled) 404 Not Found (Token not found) 500 Internal Server Error					

14.1.1.3. Decrypt Device Password

URL	/system/decryption					
Method	POST					
Permission	mdpDevice [View]					
Parameters	Name	Description	Type	Optional/Mandatory	Valid values	Default
	password	Password to be decrypted	String	Mandatory	*	
	serialNumber	Serial number of device	String	Mandatory	*	

Returns	200 OK 401 Unauthorized (Token invalid) 500 Internal Server Error
	<pre>{ "decryption" : { "output" : "209111304", } }</pre>

14.1.1.4. Get MultiDrive License Information

URL	/system/mdLicense
Method	GET
Returns	200 OK 401 Unauthorized (Token invalid) 403 Forbidden (Access not allowed) 500 Internal Server Error
	<pre>{ "mdLicense" : { "numDevices" : "100", "maxDevices" : "1000", "numClients" : "5", "maxClients" : "10", "numCommsServers" : "1", "maxCommsServers" : "10" } }</pre>