Mi-C3 Integration API

0. Confidentiality Notice

This document contains confidential information that should not be distributed without permission from MI-C3.

You shall disclose and use Confidential Information contained in this document solely pursuant to the purpose of the Umbrella Management System Project at IHS Nigeria hereinafter “the Project”. Except as otherwise expressly requested so to do, you shall maintain the Confidential Information, including but not limited to the existence and content of this document, in confidence and shall not disclose Confidential Information (which shall include any notes, extracts, analyses, and materials which would disclose Confidential Information) to any third parties, except a parent or subsidiary of Mi-C3 International Limited, hereinafter “Mi-C3”. You shall protect Confidential Information by using at least the same degree of care, but no less than a reasonable degree of care, to prevent the unauthorized use, dissemination or publication of Confidential Information that you use to protect your own confidential information of a like nature. At a minimum, you shall limit disclosure of Confidential Information to only those of your employees who: (a) have a need to know such information for the purpose of the Project; and (b) have agreed to be bound by nondisclosure terms at least as comprehensive as those set forth herein. You understand and acknowledge that no license whatsoever under any of Mi-C3’s patent, copyright, mask work right, or other intellectual property right is granted to or conferred upon you or by the disclosure of any Confidential Information to you as contemplated hereunder, either expressly, by implication, inducement, estoppel or otherwise, and that any license under such intellectual property rights must be express and in writing. Except for the use and disclosure for the purpose of the Project, Confidential Information shall remain confidential until it becomes: (a) rightfully in the public domain other than by breach of your duty; or (b) rightfully received from a third party without any limitation on disclosure; or (c) rightfully known to you without any limitation on disclosure prior to the 22nd November 2015; or (d) independently developed by employees who have not had access to the Confidential Information, or guidance from those who have had access; or (e) generally made available to third parties without restriction on disclosure. Confidential Information is provided “as is” and Mi-C3 specifically disclaims all warranties, express or implied, including, without limitation, the warranties of merchantability and fitness for a particular purpose and any warranty against infringement of any intellectual property right of a third party regarding the Confidential Information. The Confidential Information may be controlled by law or national regulation. The export, re-export or disclosure to foreign employees, subsidiaries, branches (and employees of those companies) may require government approval. You shall insure that all activities involving distribution and export/re-export of Confidential Information are in compliance with all applicable laws, regulations, orders or other restrictions. You agree that neither you nor your parent nor any of your subsidiaries will use, export, transfer, make available or otherwise disclose any Confidential Information in violation of any applicable law. In no event shall Mi-C3 be liable for any loss of profits, loss of use, indirect, incidental, consequential or special damages, irrespective of whether they have advance notice of the possibility of such damages. You are not obligated to provide comments or suggestions regarding the Confidential Information to Mi-C3. However, should you provide comments or suggestions for the modification, correction, improvement or enhancement of: (a) the Confidential Information; or (b) the Mi-C3’s products which may embody the Confidential Information, then you grant Mi-C3 a nonexclusive, irrevocable, worldwide, royalty-free license, including the right to sublicense the other party’s licensees and customers, under your intellectual property rights, the rights to use and disclose such comments and suggestions in any manner Mi-C3 chooses and to display, perform, copy, have copied, make, have made, use, sell, offer to sell, and otherwise dispose of your and your sublicensees’ products embodying such comments or suggestions in any manner and via any media Mi-C3 chooses, but without reference to the source of such

comments and/or suggestions. By continuing to read this document beyond this point you confirm ipso facto your agreement to the terms of this notice.

1. Introduction

1.1. Who is this for?

A typical installation of the Mi-C3 platform requires integration with many systems, from signal generators to job systems and more. If you have a system or equipment that needs to integrate in some way with the Mi-C3 platform, this document will help you achieve your goal.

2. Getting started

2.1. API information

The API is a REST service that is hosted on the Mi-C3 platform.
Each of our customers have the option to host it on their own infrastructure so the exact location of the API varies between customers.
To integrate with the platform, your application can make requests to the API to send information like alarms that you would like to report.
Communication is done on the HTTPS protocol.

2.2. Getting access to the API

To accessing the API you need to get an API key. The API key is registered against the IP or domain from where the API requests will originate. You need to authenticate with the server using the key.
The API key must be sent in the HTTP header of each request. The header name is **AUTHTOKEN**.

2.3. Recommendations for testing and test server sandbox

2.3.1. TBA

3. The API

3.1. General API principles

3.1.1. Methods and return codes

It’s a best practice to always set the Accept and Content-Type (in case of posting/putting JSON) headers to application/json on the HTTP requests described below.

Methods

Method	Operations
GET	Get a single resource or get a collection of resources.
POST	Create a new resource. Also used for executing resource-queries which have a too complex request-structure to fit in the query-URL of a GET-request.
PUT	Update properties of an existing resource. Also used for invoking actions on an existing resource.
DELETE	Delete an existing resource.

- Depending on the vendor's security authorization, some methods like DELETE and PUT may not be available.

Response codes

Response	Description
200 - Ok	The operation was successful and a response has been returned (GET and PUT requests).
201 - Created	The operation was successful and the entity has been created and is returned in the response-body (POST request).
204 - No content	The operation was successful and entity has been deleted and therefore there is no response-body returned (DELETE request).
401 - Unauthorized	The operation failed. The operation requires an Authentication header to be set. If this was present in the request, the supplied credentials are not valid or the user is not authorized to perform this operation.
403 - Forbidden	The operation is forbidden and should not be re-attempted. This does not imply an issue with authentication not authorization, it’s an operation that is not allowed.

404 - Not found	The operation failed. The requested resource was not found.
405 - Method not allowed	The operation failed. The used method is not allowed for this resource.
409 - Conflict	The operation failed. The operation causes an update of a resource that has been updated by another operation, which makes the update no longer valid. Can also indicate a resource that is being created in a collection where a resource with that identifier already exists.
415 - Unsupported Media Type	The operation failed. The request body contains an unsupported media type. Also occurs when the request-body JSON contains an unknown attribute or value that doesn't have the right format/type to be accepted.
500 - Internal server error	The operation failed. An unexpected exception occurred while executing the operation. The response-body contains details about the error.

The media-type of the HTTP-responses is always `application/json` unless binary content is requested

3.1.2. Error Response Codes

Errors of code 4xx and 5xx has the following response body format:

```
{
  "statusCode" : 405,
  "errorMessage" : "This method is not allowed for
                    resource 'alarm'.
}
```

3.1.3. POST body format

The format for the body of a post request is JSON. For further clarification, see each specific resource's POST documentation.

3.1.4. Response

The response, in general, reflects the state of the resource in the database. If you do a post or put the server may adjust the parameters you sent. For example changing the case or rounding a number. The response will contain these modified fields so you can inspect them if needed.

3.1.5. Versioning

In the future there may be different versions of the API. In order to maintain backward compatibility, each request must be accompanied by the API

version number. This number must be included in the HTTP header of the request. Name of the field is **VERSION**. The following table lays out the existing versions:

Version	Date released
1.0	TBA

3.2. API Resources

3.2.1. Sites

3.2.1.1. To get a list of sites

GET /site
Response

Identifier	Data Type	Description
total	uint32	How many results on all pages
start	uint32	The index of the first result on this page
size	uint32	How many Results on this page
siteId	uint32	Unique ID of the site
name	string	Name of the site
location	JSON Object	Location details for the site
description	string	Description of the site

location

Identifier	Data Type	Description
latitude	float	latitude
longitude	float	longitude
name	string	Location Name(For IHS this is the **region**)
address	JSON	Address information

	Object	

address

Identifier	Data Type	Description
type	string	"Physical" or "Postal"
line1	string	Address line 1
line2	string	Address line 2
country	string	Country
province	string	Province or State
city	string	City
code	string	Postal code or zip

Response Body:


```
{
   "data":[{
      "siteId":75,
      "name":"IHS_ABE_003",
      "location":{
            "latitude":-15.367728,
            "longitude":28.23107,
            "name":"Lagos",
            "address":{
                  "type":"Physical",
                  "line1":"4/6 AKINMATEOLA STREET OFF RAJI
            RAZAKI ROAD",
                  "line2":"FESTAC TOWN AMUWO ODOFIN",
                  "country":"Nigeria",
                  "province":"Lagos",
                  "city":"Lagos",
                  "code":""
            }
      },
      "description":"Hillside school"
   },
   {
      "siteId":76,
      "name":"IHS_ABE_004",
      "location":{
```

```
            "latitude":-15.367728,
            "longitude":28.23107,
            "name":"Lagos",
            "address":{
                  "type":"Physical",
                  "line1":"4/6 AKINMATEOLA STREET OFF RAJI
            RAZAKI ROAD",
                  "line2":"FESTAC TOWN AMUWO ODOFIN",
                  "country":"Nigeria",
                  "province":"Lagos",
                  "city":"Lagos",
                  "code":""
            }
      },
      "description":"Hillside school2"
 }],
   "total":2,
   "start":0,
   "size":2
}
```


3.2.1.2. To get the information for one site

GET /site/{siteId}
Response

Identifier	Data Type	Description
siteId	uint32	Unique ID of the site (IHS Site ID)
name	string	Name of the site
location	JSON Object	Location details for the site
description	string	Description of the site
equipment	JSON array	A list of all the equipment on the site.
tenants		TODO
cascading		TODO
priority		
project		

towerHeight		
company		

location

Identifier	Data Type	Description
latitude	float	latitude
longitude	float	longitude
name	string	Location Name(For IHS this is the **region**)
address	JSON Object	Address information

address

Identifier	Data Type	Description
type	string	"Physical" or "Postal"
line1	string	Address line 1
line2	string	Address line 2
country	string	Country
province	string	Province or State
city	string	City
code	string	Postal code or zip

Response Body:


```
{
   "siteId":75,
   "name":"IHS_ABE_003",
   "location":{
            "latitude":-15.367728,
            "longitude":28.23107,
            "name":"Lagos",
            "address":{
                  "type":"Physical",
                  "line1":"4/6 AKINMATEOLA STREET OFF RAJI
RAZAKI ROAD",
                  "line2":"FESTAC TOWN AMUWO ODOFIN",
                  "country":"Nigeria",
```

```
                "province":"Lagos",
                "city":"Lagos",
                "code":""
            }
        },
    "description":"Hillside school"
    "equipment": [{
                "equipmentId": 3,
                "parentequipmentId": 75
                "name":"IHS_LSK_089M",
                "equipmentTypeId": 3,
                "siteId":75,
                "siteName":"IHS_ABE_003",
                "SIMs":[{
                    "telNo":"+2348033070224",
                    "IPAndPort":"127.0.0.1:3254",
                    "ICCID":"89310410106543789301"
                    }]
                }]
}
```


3.2.1.3. To create or update a site

Please note that, in most cases, POST and PUT will not be available. These are reserved for special cases.
POST or **PUT** /site/{siteId}

Identifier	Required	Data Type	Description
name	yes - POST no - PUT	string	Name of the site
location	no	string	Description of the site location
latitude	no	float	latitude
longitude	no	float	longitude
description	no	string	Description of the site

Response:
HTTP response codes

Response Code	Description

201	Indicates that the site was created
200	Indicates that the site was updated

Example:
POST /site/75
Request Body:

```
{
   "name":"IHS_ABE_003",
   "location":{
            "latitude":-15.367728,
            "longitude":28.23107,
            "name":"Lagos",
            "address":{
                  "type":"Physical",
                  "line1":"4/6 AKINMATEOLA STREET OFF RAJI
            RAZAKI ROAD",
                  "line2":"FESTAC TOWN AMUWO ODOFIN",
                  "country":"Nigeria",
                  "province":"Lagos",
                  "city":"Lagos",
                  "code":""
            }
      },
   "description":"Hillside school"
}
```

Successful Response Body:

```
{
   "siteId":75,
   "name":"IHS_ABE_003",
   "location":{
            "latitude":-15.367728,
            "longitude":28.23107,
            "name":"Lagos",
            "address":{
                  "type":"Physical",
                  "line1":"4/6 AKINMATEOLA STREET OFF RAJI
            RAZAKI ROAD",
                  "line2":"FESTAC TOWN AMUWO ODOFIN",
                  "country":"Nigeria",
                  "province":"Lagos",
                  "city":"Lagos",
```

```
                "code":""
            }
        },
    "description":"Hillside school"
}
```


PUT /site/75
(You only need to send the attributes that are to be updated)
Request Body:


```
{
    "name":"IHS_XYZ_003"
}
```


Successful Response Body:


```
{
    "siteId":75,
    "name":"IHS_XYZ_003",
    "location":{
            "latitude":-15.367728,
            "longitude":28.23107,
            "name":"Lagos",
            "address":{
                "type":"Physical",
                "line1":"4/6 AKINMATEOLA STREET OFF RAJI
            RAZAKI ROAD",
                "line2":"FESTAC TOWN AMUWO ODOFIN",
                "country":"Nigeria",
                "province":"Lagos",
                "city":"Lagos",
                "code":""
            }
        },
    "description":"Hillside school"
}
```


3.2.1.4. To delete a site

Please note that, in most cases, DELETE will not be available. These are reserved for special cases.
DELETE {url}

Identifier	Required	Data Type	Description
siteId	yes	uint32	Unique ID of the site

Response:
HTTP response codes

Response Code	Description
200	Indicates that the site was deleted

Example:
DELETE /site/75

3.2.2. Tenants

3.2.2.1. To retrieve a list of tenants

GET /tenants
Response:

Identifier	Data Type	Description
tenantId	uint32	How many results on all pages
name	string	Name of the tenant
		TODO: possibly a tenant code
total	uint32	How many results on all pages
start	uint32	The index of the first result on this page
size	uint32	How many Results on this page

TODO: the example in- and output

3.2.3. Equipment Types

Equipment types are configured in the MI-C3 platform. Mi-C3 works with vendors of equipment to define the equipment types that the vendors monitor. Vendors may retrieve the configuration details, using the API, but would likely not do so as part of normal API usage.

3.2.3.1. To retrieve a list of available equipment types

GET /equipmentType
Response:

Identifier	Data Type	Description
equipmentTypeId	uint32	Id of the equipment type
name	string	Name of the equipment type
readingTypes	JSON	How many Results on this page
total	uint32	How many results on all pages
start	uint32	The index of the first result on this page
size	uint32	How many Results on this page

readingTypes

Identifier	Data Type	Description
name	string	Name of the reading object.
dataType	string	One of ["string", "integer", "float","boolean"]
unitOfMeasure	string	The unit that will be displayed.
unitPosition	string	One of ["before","after",""]. Where to display the unit of measure in relation to the value of the reading. Default is "after".
displayValues	string	In the case where the datatype is boolean, the display value must be placed here with the 2 options separated by a "," comma. See the example.

HTTP response codes

Response Code	Description
200	Indicates that the data was retrieved successfully

Response body:


```
{
	"data": [{
		"equipmentTypeId": 1,
		"name": "Enclosure Door",
		"readingTypes": [{
			"name":"Main entrance",
			"dataType": "boolean",
			"unitOfMeasure": "",
			"unitPosition": "",
			"displayValues": "open,closed"
		}]
	},
	{
		"equipmentTypeId": 2,
		"name": "Generator type 1",
		"readingTypes": [{
			"name":"Oil Temperature",
			"dataType": "float",
			"unitOfMeasure": "&#8451;",
```

```
                "unitPosition": "after",
                "displayValues": ""
            },
            {
                "name":"Oil Pressure",
                "dataType": "float",
                "unitOfMeasure": "kpa",
                "unitPosition": "after",
                "displayValues": ""
            }]
        }],
        "total": 2,
        "start": 0,
        "size": 2
}
```


Example note: "℃" is unicode for ℃ (degrees Celcius)

3.2.4. Equipment

3.2.4.1. To get a list of equipment

GET /site/{siteId}/equipment
Response

Identifier	Data Type	Description
total	uint32	How many results on all pages
start	uint32	The index of the first result on this page
size	uint32	How many Results on this page
equipmentId	uint32	Unique ID of the equipment
name	string	Name of the equipment
equipmentTypeId	uint32	Type of equipment. Refer to Equipment Types for more information.
siteId	uint32	ID of the site where the equipment is installed
siteName	string	Name of the site where the equipment is

		installed
parentEquipmentId	uint32	Id of the parent equipment
SIMs	JSON Array	Sim cards installed

SIMs

Identifier	Data Type	Description
telNo	string	Telephone number of the equipment
IPAndPort	string	IP address and port of the equipment. (aaa.bbb.ccc.ddd:pppp)
ICCID	string	Sim ICCID

Response Body:


```
{
   "data":[{
            "equipmentId": 75,
            "name":"IHS_LSK_089M",
            "equipmentTypeId": 1,
            "siteId":75,
            "siteName":"IHS_ABE_003",
            "parentEquipmentId":2,
            "SIMs":[{
                  "telNo":"+2348033070224",
                  "IPAndPort":"127.0.0.1:3254",
                  "ICCID":"89310410106543789301"
            }]
            },
            {
            "equipmentId": 76,
            "name":"IHS_LSK_090M",
            "equipmentTypeId": 2,
            "siteId":75,
            "siteName":"IHS_ABE_003",
            "parentEquipmentId":2,
            "SIMs":,[{
                  "telNo":"+2348033070224",
                  "IPAndPort":"127.0.0.1:3254",
```

```
                "ICCID":"89310410106543789301"
            }]
            }],
    "total":2,
    "start":0,
    "size":2
}
```


3.2.4.2. To get a single equipment

GET /equipment/{equipmentId}

Response

Identifier	Required	Data Type	Description
dequipmentId	yes	uint32	Unique ID of the equipment
name	yes	string	Name of the equipment
equipmentTypeId	yes	uint32	Type of equipment. Refer to Equipment Types for more information.
siteId	yes	uint32	ID of the site where the equipment is installed
siteName	no	string	Name of the site where the equipment is installed
parentEquipmentId	no	uint32	Id of the parent equipment
SIMs	no	JSON Array	Sim cards installed

SIMs

Identifier	Data Type	Description
telNo	string	Telephone number of the equipment
IPAndPort	string	IP address and port of the equipment. (aaa.bbb.ccc.ddd:pppp)
ICCID	string	Sim ICCID

Request Body:

```
{
   "equipmentId": 75,
   "name":"IHS_LSK_089M",
   "equipmentTypeId": 1,
   "siteId":75,
   "siteName":"IHS_ABE_003",
   "parentEquipmentId":2,
   "SIMs":[{
                  "telNo":"+2348033070224",
                  "IPAndPort":"127.0.0.1:3254",
                  "ICCID":"89310410106543789301"

}]
}
```


3.2.4.3. To create new equipment

POST /equipment/{equipmentId}

Identifier	Required	Data Type	Description
name	yes	string	Name of the equipment
equipmentTypeId	yes	uint32	Type of equipment. Refer to Equipment Types for more information.
siteId	yes	uint32	ID of the site where the equipment is installed
siteName	no	string	Name of the site where the equipment is installed
parentEquipmentId	no	uint32	Id of the parent equipment
SIMs	no	JSON Array	Sim cards installed

SIMs

Identifier	Data Type	Description
telNo	string	Telephone number of the equipment
IPAndPort	string	IP address and port of the equipment. (aaa.bbb.ccc.ddd:pppp)

ICCID	string	Sim ICCID

Response:
HTTP response codes

Response Code	Description
201	Indicates that the equipment was created

Example:
POST /equipment/725
Request Body:

```
{
   "name":"IHS_LSK_089M",
   "equipmentTypeId": 2,
   "siteId":75,
   "siteName":"IHS_ABE_003",
   "parentEquipmentId":2,
   "SIMs":[{
                  "telNo":"+2348033070224",
                  "IPAndPort":"127.0.0.1:3254",
                  "ICCID":"89310410106543789301"
}]
}
```

3.2.4.4. To update update equipment

PUT /equipment/{equipmentId}

Identifier	Required	Data Type	Description
name	no	string	Name of the equipment
siteId	no	uint32	ID of the site where the equipment is installed
siteName	no	string	Name of the site where the equipment is installed
SIMs	no	JSON Array	Sim cards installed

SIMs

Identifier	Data	Description

	Type	
telNo	string	Telephone number of the equipment
IPAndPort	string	IP address and port of the equipment. (aaa.bbb.ccc.ddd:pppp)
ICCID	string	Sim ICCID

Response:
HTTP response codes

Response Code	Description
200	Indicates that the equipment was updated

Example:
PUT /equipment/725
Request Body:

```
{
   "SIMs":[{
                  "telNo":"+2348033070224",
                  "IPAndPort":"127.0.0.1:3254",
                  "ICCID":"89310410106543789301"
}]
}
```

3.2.4.5. To delete equipment

DELETE /equipment/{equipmentId}

Response:
HTTP response codes

Response Code	Description
200	Indicates that the equipment was deleted

Example:
DELETE /equipment/725

3.2.5. Alarms

Alarms can also be thought of as events or incidents.
The system monitoring equipment may have rules that define exceptions or behaviours that are out of the norm. These events need to be raised and dealt with by a person who will follow a defined process to resolve the issue.

3.2.5.1. To create one or more new Alarms

POST /alarm/
Body parameters:

Identifier	Required	Data Type	Description
alarmId	Yes	string	Id of the alarm in the vendor system.
siteId	Yes	uint32	The ID of the site where the alarm was generated.
siteName	No	string	The name of the site where the alarm was generated.
severity	Yes	byte	Severity ranging from 0 to 4. 0 = Cleared, 1 = Critical, 2 = Major, 3 = Minor, 4 = Event. 'Critical' events map internally to 'Site Down' 'Major' events map internally to 'Warning'.
timeStamp	Yes	string	Timestamp in the following string format (ODBC/JDBC): YYYY-MM-DD hh:mm:ss[.fff...] The timezone is UTC (GMT+0).
description	Yes	string	Description of the alarm. Used to establish what Standard Operating Procedure to invoke. TODO: talk about configuring the link to the description
equipmentId	No	uint32	ID of the equipment that generated the Alarm

Response:
HTTP response codes

Response Code	Description
201	Indicates that the alarm was created

Example:
POST
Request Body:

```
{[
   "siteId":3,
   "siteName":"IHS_ABE_003",
   "severity":2,
   "timeStamp":"2012-12-24 05:01:22.341",
   "description":"High Temperature over 70 degrees",
   "equipmentId":154
]}
```

Successful Response Body:

```
{
   "alarmReference":6653,
   "alarmId":48743,
   "siteId":3,
   "siteName":"IHS_ABE_003",
   "severity":2,
   "timeStamp":"2012-12-24 05:01:22.341",
   "description":"High Temperature over 70 degrees",
   "equipmentId":154
}
```

*Note on the response body: The alarmReference field is a number generated when the alarm was saved in the affectli system. It may potentially be used at a later stage to request the processing status of a particular alarm message.

3.2.6. Live site data

3.2.6.1. Initiating the data stream

The platform will send a POST request to your server.
It will contain the siteId for which live data is required.
Once your server receives this message, it should start sending data to the Mi-C3 platform as described below in the 'Pushing the data' section.
The response to each push will tell you if another push should be made.

Identifier	Data Type	Description
requestInterval	uint32	Number of seconds the Mi-C3 platform expects between requests.
expiryTime	uint32	The number of seconds after which push requests can be stopped, even if the Mi-C3 platform is still acceptiing

In the push request the

(expiry time for pushing)
Example:
POST {configured URL}
{
...
}

3.2.6.2. Pushing the data

After the message to start pushing live site data is received from the Mi-C3 platform, you can post data to the API as follows:
POST /live/site/{siteId}

Identifier	Required	Data Type	Description
messageTimestamp	yes	string	Timestamp in the following string format (ODBC/JDBC): YYYY-MM-DD hh:mm:ss.ffff The timezone is UTC (GMT+0).
status	yes	string	One of ["online","offline","unknown"]. Shows if the site is operational.
equipment	yes	JSON array	Each element in the format as below

equipment

Identifier	Required	Data Type	Description
equipmentId	yes	uint32	Unique ID of the equipment
parentequipmentId	no	uint32	The unique equipment ID of the parent equipment, if applicable.
status	yes	string	online/offline/problem/unknown online assumes no problem
readings	no	JSON array	Each element in the format shown below
problemText	no	JSON array	Each element in the format shown below

readings

Identifier	Required	Data Type	Description
[Reading name]	yes	string	The attribute is the name of the reading object. The value is a JSON object. This format was chosen to make it impossible to have duplicate readings.

dataType	yes	string	one of ["string", "integer", "float","boolean"]
name	yes	string	The name that will be displayed.
unitOfMeasure	yes	string	The unit that will be displayed. In the case where the datatype is boolean, the display value must be placed here with the 2 options separated by a "-" dash. See the example.
value	yes	string	Value of the reading in the defined data type.

problemText

Identifier	Required	Data Type	Description
text	yes	string	Error message
code	no	string	Error code if available

Response:
HTTP response codes

Response Code	Description
200	Indicates that the data was accepted.
201	Indicates that no more data is required and the push can be stopped.

TODO: add display values
Example:
POST /live/site/75
Request Body:

```
{
      "messageTimestamp": "2015-12-31 13:45:16.7623",
      "status": "online",
      "equipment": [{
            "equipmentId": 75,
            "status": "online",
            "readings": {
                  "Enclosure Door": {
                        "value": true,
                        "integrity": "valid"
                  }
            },
```

```
            "problemText": []
        },
        {
            "equipmentId": 76,
            "status": "problem",
            "readings": {
                "Oil Temperature": {
                    "value": 181.3,
                    "integrity": "valid"
                },
                "Oil Pressure": {
                    "value": -1,
                    "integrity": "invalid"
                }
            },
            "problemText": [{
                "text": "Temperature above threshold."
            },
            {
                "text": "Fuel sensor not reporting.",
                "code": "E13"
            }]
        }]
}
```

3.2.7. Aggregate Site Reports

Reports are still to be defined.

Example:

POST

Request Body:

```
{
   "messageId":752790,
   "reportType":"Some Report Type",
   "timeStamp":"2012-12-24 05:01:22.341",
   "reportData":{some: data},
}
```

Successful Response Body:

???

4. Appendix

4.1. Phased approach

In order to maximise the impact of employing the vendor RMS link to the system in the short term, several phases of implementation needs to occur.

4.1.1. Phase 1

In phase 1, RMS systems will only send alarm notifications. Site information does not have to be maintained.

4.1.1.1. Alarms

4.1.1.1.1. To create one or more new Alarms

POST /alarm/

Body parameters:

Identifier	Required	Data Type	Description
alarmId	Yes	string	Id of the alarm in the vendor system.
siteId	Yes	string	IHS site id or vendor site id
siteName	No	string	The name of the site where the alarm was generated.
severity	Yes	byte	Severity ranging from 0 to 4. 0 = Cleared, 1 = Critical, 2 = Major, 3 = Minor, 4 = Event. 'Critical' events map internally to 'Site Down' 'Major' events map internally to 'Warning'.
timeStamp	Yes	string	Timestamp in the following string format (ODBC/JDBC): YYYY-MM-DD hh:mm:ss[.fff...] The timezone is UTC (GMT+0).
description	Yes	string	Description of the alarm. Used to establish what Standard Operating Procedure to invoke. TODO: talk about configuring the link to the description
equipmentInfo	No	**EquipmentInfo**	ID of the equipment that generated the Alarm. For phase 1 it replaces the equimentID field.

EquipmentInfo

Identifier	Required	Data Type	Description

equipmentTypeId	yes	uint32	Type of equipment (see Equipment Types)
tenant	no	string	Tenant Code (see the tenant section)
tenantSeq	no	int32	Default is 1, use when Tenant has multiple equipment installed

Response:
HTTP response codes

Response Code	Description
201	Indicates that the request was accepted.

Example:
POST
Request Body:

```
{[
   "siteId":"IHS_ABE_003",
   "siteName":"IHS_ABE_003",
   "severity":2,
   "timeStamp":"2012-12-24 05:01:22.341",
   "description":"High Temperature over 70 degrees",
   "equipmentInfo":{
      "equipmentTypeId":4
      "tenant":3
      "tenantSeq":1
      }
]}
```

Successful Response Body:

```
{
   "alarmReference":6653,
   "alarmId":48743,
   "siteId":3,
   "siteName":"IHS_ABE_003",
   "severity":2,
   "timeStamp":"2012-12-24 05:01:22.341",
   "description":"High Temperature over 70 degrees",
   "equipmentInfo":{
      "equipmentTypeId":4
      "tenant":3
      "tenantSeq":1
```

```
            }
        }
```

*Note on the response body: The alarmReference field is a number generated when the alarm was saved in the affectli system. It may potentially be used at a later stage to request the processing status of a particular alarm message.

4.1.2. Phase 2

4.1.2.1. Site and Equipment Maintenance

4.1.3. Phase 3

4.1.3.1. Live site data