Source Documents - API - Location-based Services

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Mapping API v1.1

Mapping API Architecture

Cisco Prime Binding

Mapping API Methods

Mapping API Rich Example

Mapping API Response Handl

POI Types

Location API v1.2

Location API Architecture

Retrieve Location Data

Location API Response Handli

MARS API v1.1

MARS API Architecture

MARS API Methods

MARS API Response Handling

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Mapping API v1.1

ON THIS PAGE

This document describes the specification for the MaaS Mapping API ("API"). This API allows remote clients to manage the various resources associated with mapping, including venues, campuses, buildings, floors, points of interest, segments and routes.

Security

All calls made to the CME must adhere to the guidelines presented in the MaaS Security Protocol document.

Definitions

Below are the object types that need to be created, read, updated and destroyed (CRUD) for geofences, callbacks and associated zone elements. Other terms used with the Mapping service are defined as well.

Term	Definition
Venue	A location where events take place. Venues consist of one or more campuses.
Campus	A collection of one or more buildings grouped together by a common theme.
Building	A physical structure that contains one or more floors.

Floor	An object associated with a building containing a building ID, floor ID, zoom level and resource URL(s) (e.g. .svg, .pdf).
Resource	The image files associated with a floor. The .svg or .pdf asset URL and the associated metadata are often referred to as a map.
Point	A point of interest (POI), waypoint or portal location associated with a map.
Zoom Level	The zoom scale on the actual map. • Zoom Level 1 = 1.0 zoom scale on device • Zoom Level 2 = 2.0 zoom scale on device • Zoom Level 3 = 4.0 zoom scale on device • Zoom Level 4 = 8.0 zoom scale on device • Zoom Level 5 = 16.0 zoom scale on device • Undefined Zoom Level = -1

MSE	The Cisco Mobility Services Engine (MSE) is a network appliance that provides tools for wireless network monitoring and network asset location tracking.
GUID	A Globally Unique Identifier (GUID) is a unique identifier that typically conforms to the Universally Unique Identifier (UUID) standard as defined by the Open Software Foundation. For reference, see http://en.wikipedia.org/wiki/Globally_unique_identifier and http://en.wikipedia.org/wiki/Universally_unique_identifier.
JSON	Stands for JavaScript Object Notation and is used for the request and response formats due to its portability and simplicity.
RFC 3339	A date format that "provide[s] an unambiguous and well-defined method of representing dates and times." See http://www.ietf.org/rfc/rfc3339.txt for more details.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Mapping API Architecture

ON THIS PAGE

Production URL

The URL will have this basic structure:
http://map-api.phunware.com/v1.1/{object}/{object id}

The object ID is used for resource operations for all methods in this document except for GET. For the mapping service, examples of objects are maps, POIs or resource types.

Request / Response Format

JSON is used for the request and response format. JSON is a lightweight and portable format that maintains human-readability.

When making POST or PUT requests, the request parameters MUST be JSON-encoded and placed in the request body. When making GET requests, the request parameters MUST be JSON-encoded, then URI-encoded and placed directly in the query string.

Clients should expect JSON-encoded responses for every request, even those that result in an error.

HTTP Method

Each method defines the HTTP method required depending on what the usage is. For example, if the method is used to simply get data, the HTTP method would be GET. If the method is used to delete data it uses DELETE.

Date / Time Format

The date and time MUST be expressed in RFC 3339 format:
YYYY-MM-DDThh:mm:ssZ

- All times must be in the UTC time zone and indicated with a "Z". This is done to mitigate issues regarding Daylight Saving Time (DST).
- No fractional seconds are allowed.
- See section 5.6 of Date and Time on the Internet: Timestamps for more information.

Bracketed IDs ("[id]"s)

Whenever a bracketed ID ("[id]") appears in a method's name or URL (e.g. "GET /containers/[id]" or "http://cms-api.phunware.com/v1.0/containers/[id]"), treat it as a placeholder for the ID number of the item involved in the method (in this case, "GET /containers/12345" or "http://cms-api.phunware.com/v1.0/containers/12345").

Ellipses ("…"s)

Whenever an ellipsis ("…") appears in the JSON body, treat it as a placeholder for additional instances of the data that immediately precedes it.

PUT Methods and Partial Updates

With rare exceptions (e.g. segments), there are a few differences in how the request parameters get handled by PUT methods:

- There are no required parameters.
- If a parameter IS NOT specified, it will retain its current value.
- If a parameter IS specified, but with an empty value, then the value will be cleared.
- If you pass in an identifier field for a container, schema, structure or content ID, it will be ignored.

URL Definitions

The mapping web service will interact with the…

- Cisco Mobility Services Engine (MSE) (via two POST methods: venue and floor)
- Mapping SDK (via two GET methods: building and POI)
- MaaS portal (via GET / PUT / POST / DELETE methods on two types of objects: point and resource)

GET Methods and Query Strings

The GET methods outlined in this document will use query strings, not JSON bodies, when making a request to a URL. This requires the user to convert JSON into a URL-encoded query string. Here's an example:

```
{
 "tags":
 {
  "any":
  [
   "Zelda",
   "Link"
  ]
 }
}
```

Minified fragment (using http://bigaqua.org/minify_json.html).

```
{"tags":{"any":["Zelda","Link"
]}}
```

URL-encoded, minified fragment (using http://www.url-encode-decode.com/):

```
%7B%3Ftags%3F%3A%7B%3Fany%3F%3
A%5B%3FZelda%3F%2C%3FLink%3F%5
D%7D%7D
```

This URL-encoded, minified fragment is the query string. In order to compose a request, a "?" is appended to the end of the request URL, then the URL-encoded, minified fragment is added after that:

```
http://map-api.phunware.com/v1
.1/venues?%7B%3Ftags%3F%3A%7B%
3Fany%3F%3A%5B%3FZelda%3F%2C%3
FLink%3F%5D%7D%7D
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Cisco Prime Binding

ON THIS PAGE

We will construct venues, campuses, buildings, floors and resources from data provided by Cisco Prime. This will be accomplished by specifying a callback URL in Cisco Prime to which an HTTP POST request will be done. The callback URL is a property of the venue object (see below).

Step 1: data is POSTed

Once the registration URL has been submitted, the MSE first posts the venue hierarchy and dimensions of each floor. Here is a sample of the data POSTed by Cisco to the callback URL:

```
{
    "locationName": "Site 4",
    "latitude": 37.40714,
    "longitude": -121.92878,
    "mseUdi":
"AIR-MSE-3355-K9:V01:KQYGBRD",
    "appId":
"-4564396851560868792",
    "emailAddresses":
 [
        "somebody@gmail.com"
    ],
    "additionalInfo": "",
    "trackedElementsLimit":
100,
    "serverName": "alpha-mse",
    "streetAddress": "3625
Cisco Way, San Jose, CA
95134",
    "floors":
 [
        {
            "aesUid":
"-4564397058403860182",
            "name": "WNBU",
            "level": 4,
            "isOutdoor":
false,
            "length": 185,
            "height": 10,
            "offsetX": 5,
            "offsetY": 0,
            "textHierarchy":
   [
                "Cisco Site
4",
                "SJ-14",
                "WNBU"
            ],
            "idHierarchy":
   [
"-4564397058403860184",

"-4564397058403860183",

"-4564397058403860182"
            ],
            "imageName":
"domain_0_1349311055718.png",
            "imageType":
"image/png",
            "imageExists":
true,

"imageLastModified":
1349330637000
```

```
        }
    ]
}
```

Step 2: images that need to be uploaded are specified

Phunware then responds with a list of floor images that need to be uploaded:

```
{
    "id": 9999,
    "images_needed":
 [
        {
            "imageName":
"domain_0_1349311055718.png",
            "imageType":
"image/png",
            "uploadUrl":
"http://[S3 bucket]/[S3
subfolder]/filename"
        }
    ]
}
```

Step 3: Cisco uploads images

The MSE will then post the requested images to the URLs specified.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Below are the methods in use with our Mapping API.

ON THIS PAGE

VENUE

- Create a Venue
- Retrieve a Venue
- Retrieve a Collection of Venues
- Update a Venue

- Delete a Venue
- Venue Audit History
- Workflow - Publish and Revert

CAMPUS

- Create a Campus
- Retrieve a Campus
- Retrieve a Collection of Campuses
- Update a Campus
- Delete a Campus

BUILDING

- Create a Building
- Retrieve a Building
- Retrieve a Collection of Buildings
- Update a Building
- Delete a Building

FLOOR

- Create a Floor
- Retrieve a Floor
- Retrieve a Collection of Floors
- Update a Floor
- Delete a Floor

RESOURCE

- Create a Resource
- Retrieve a Resource
- Retrieve a Collection of Resources
- Update a Resource
- Delete a Resource

POINT

- Create a Point
- Retrieve a Point
- Retrieve a Collection of Points by Floor and/or Building
- Retrieve a Count of Points by Floor and/or Building
- Update a Point
- Delete a Point

POI TYPE

- Retrieve a POI Type
- Retrieve a POI Type by Value

ROUTE

- Create a Route
- Retrieve a Route
- Retrieve a Collection of Routes
- Retrieve a Location's Routes
- Retrieve a Count of Routes
- Update a Route
- Delete a Route
- Reset a Building's Routes
- Update a Building's Routes

SEGMENT

- Create a Segment
- Retrieve a Segment
- Retrieve a Collection of Segments
- Retrieve a Count of Segments
- Update a Segment
- Delete a Segment

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Mapping API Rich Example

1. Create a Venue
2. Create a Campus
3. Create a Building
4. Create a Floor
5. Create a Resource
6. Retrieve a Venue
7. View the Venue in the Map

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

1. Create a Venue

A venue is a location where events take place. Venues consist of one or more campuses.

ON THIS PAGE

Hyrule

The example request below creates a venue called Hyrule. The example response returns the newly created venue's GUID.

Example Request

```
POST /v1.1/venues HTTP/1.1
Host: map-api.phunware.com

X-Auth: see MaaS Security Protocol v1.0
Content-Type: application/json
```

```
{
    "name": "Hyrule",
    "appId": "123456",
    "clientIds": "98765,6543",
    "orgIds": "8,9",
    "mseUdi": "abcd",
    "supportsGeographicCoordinates": true
}
```

Example Response

```
HTTP/1.1 200 OK
Vary:
Accept-Encoding
Content-Type:
application/json;
charset=utf-8
Date: Tue, 20 May
2015 16:06:54 GMT
Connection:
keep-alive
```

```
{
    "data":

 {

"guid":
"f0685279-7
497-4d90-88
2e-12201e47
6b0e"
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

2. Create a Campus

A campus is a collection of one or more buildings grouped together by a common theme.

ON THIS PAGE

Hyrule Castle

The example request below creates a campus called Hyrule Castle. The example response returns the newly created campus' ID.

Hyrule Castle will live inside of the Hyrule venue, its parent container.

Example Request

```
POST /v1.1/campuses
HTTP/1.1
Host:
map-api.phunware.com

X-Auth: see MaaS
Security Protocol
v1.0
Content-Type:
application/json
```

```
{
 "name":
"Hyrule
Castle",

"venueGuid"
:
"f0685279-7
497-4d90-88
2e-12201e47
6b0e"
}
```

Example Response

```
HTTP/1.1 200 OK
Vary:
Accept-Encoding
Content-Type:
application/json;
charset=utf-8
Date: Tue, 20 May
2015 16:06:54 GMT
Connection:
keep-alive
```

```
{
 "data":
    {

"id": 1986
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

3. Create a Building

A building is a physical structure that contains one or more floors.

ON THIS PAGE

Hyrule Castle Tower

The example request below creates a building called Hyrule Castle Tower . The example response returns the newly created building's ID.

Hyrule Castle Tower will live inside of the Hyrule Castle campus, its parent container.

Example Request

```
POST /v1.1/buildings
HTTP/1.1
Host:
map-api.phunware.com

X-Auth: see MaaS
Security Protocol
v1.0
Content-Type:
application/json
```

```
{
 "campusId"
: 1986,
    "name":
"Hyrule
Castle
Tower",

"latitude":
30.25,

"longitude"
: 97.75,

"streetAddr
ess": "123
Ocarina",

"venueGuid"
:
"f0685279-7
497-4d90-88
2e-12201e47
6b0e",

"location":

    {

"latitude":
30.25,

"longitude"
: 97.75
    }
}
```

Example Response

```
HTTP/1.1 200 OK
Vary:
Accept-Encoding
Content-Type:
application/json;
charset=utf-8
Date: Tue, 20 May
2015 16:06:54 GMT
Connection:
keep-alive
```

```
{
 "data":
    {

"id": 6891
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

4. Create a Floor

A floor is an object associated with a building containing a building ID, floor ID, zoom level and resource URL(s) (e.g. .svg, .pdf).

ON THIS PAGE

Level One

The example request below creates a floor called Level One. The example response returns the newly created floor's ID.

Level One will live inside of the Hyrule Castle Tower building, its parent container.

Example Request

```
POST /v1.1/floors
HTTP/1.1
Host:
map-api.phunware.com

X-Auth: see MaaS
Security Protocol
v1.0
Content-Type:
application/json
```

```
{
 "venueGuid
":
"f0685279-7
497-4d90-88
2e-12201e47
6b0e",

"buildingId
": 6891,
    "name":
"Level
One",

"originalMa
pUrl":
"http://ima
ge.com/leve
l1.pdf",

"level": 1,

"isOutdoor"
: false,

"width":
100,

"height":
200,

"offsetX":
0,

"offsetY":
0,
"locationMapHierarchy": "Hyrule Castle>Hyrule Castle Tower>Level One",

"maxZoomLevel": 3,

"referencePoints":
    {

"rotation": 214,

"portal":
        {

"topRight":


{

"latitude": 33.77403309960294,

"longitude":
-84.32867120304763

},

"bottomLeft":

{

"latitude": 33.7740738782866,

"longitude":
-84.32673443277963

}
```

```
        },
"topLeft":
        {
"latitude":
33.773309,
"longitude":
-84.327331
        },
"topRight":
        {
"latitude":
33.774033,
"longitude":
-84.328671
        },
"bottomLeft":
        {
"latitude":
33.774073,
"longitude":
-84.326734
        },
"bottomRight":
        {
"latitude":
33.774797,
"longitude":
-84.328074
```

```
            }
        }
    }
```

Example Response

```
HTTP/1.1 200 OK
Vary:
Accept-Encoding
Content-Type:
application/json;
charset=utf-8
Date: Tue, 20 May
2015 16:06:54 GMT
Connection:
keep-alive
```

```
{
 "data":
     {

"id": 17
     }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

5. Create a Resource

Resources are the image files associated with a floor. The .svg or .pdf asset URL and the associated metadata are often referred to as a map.

ON THIS PAGE

Map for Level One

The example request below creates a map for the floor named Level One. The example response returns the newly created resource's ID.

The map resource for Level One will be associated with the Level One floor.

Example Request

```
POST /v1.1/resources HTTP/1.1
Host: map-api.phunware.com

X-Auth: see MaaS Security Protocol v1.0
Content-Type: application/json
```

```
{
 "floorId": 17,

"pdfUrl": "http://image.com/level1.pdf",

"svgUrl": "http://image.com/level1.svg",

"zoomLevel" : 0
}
```

Example Response

```
HTTP/1.1 200 OK
Vary:
Accept-Encoding
Content-Type:
application/json;
charset=utf-8
Date: Tue, 20 May
2015 16:06:54 GMT
Connection:
keep-alive
```

```
{
 "data":
    {

"id": 71
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

6. Retrieve a Venue

Hyrule

ON THIS PAGE

The example request below retrieves the Hyrule venue, which can now be represented by this hierarchy of resources:

- Hyrule (venue)
    - Hyrule Castle (campus)
        - Hyrule Castle Tower (building)
            - Level One

(flo
or)

- http://image.com/level1.p
  and
  http://image.com/level1.s
  (resource)

The example response will contain the venue object only.

Example Request

```
GET
/v1.1/venues/f068527
9-7497-4d90-882e-122
01e476b0e HTTP/1.1
Host:
map-api.phunware.com

X-Auth: see MaaS
Security Protocol
v1.0
Content-Type:
application/json
```

Example Response

```
HTTP/1.1 200 OK
Vary:
Accept-Encoding
Content-Type:
application/json;
charset=utf-8
Date: Tue, 20 May
2015 16:06:54 GMT
Connection:
keep-alive
```

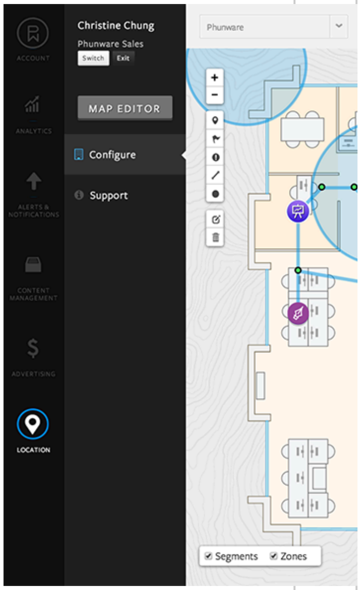
```
{
 "guid": "f0685279-7497-4d90-882e-12201e476b0e",
    "name": "Hyrule",

"appId": "123456",

"clientIds": "98765,6543",

"orgIds": "8,9",

"isActive": true,

"mseUdi": "abcd",

"callbackUrl": "http://www.Hyrule.com",

"createdAt": "2015-05-20T16:06:54Z",

"updatedAt": null,

"supportsGeographicCoordinates": true
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

7. View the Venue in the Map Editor

The steps below assume that your organization's application has already been created in the MaaS portal. If your organization was not provided with login credentials, contact your Phunware program manager.

Once the developer has successfully created floor resources for the venue, those floor resources will become available as maps in the MaaS portal's **Map Editor** (pictured right).

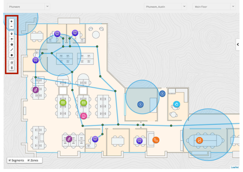


Using the Map Editor, credentialed users can add points of interest (POIs), waypoints, segments, routes and zones (geofences) using the toolbar.

There is no action required on the developer's part to make this map available for use in the Map Editor. This page will walk through the steps to take to access the new venue's map in the MaaS portal.

STEPS

1. Log into maas.phunware.com using your organization's credentials.



2. The **YOUR APPLICATIONS** page will display.
3. Click the **LOCATION** tab in the lefthand sidebar.
4. Click the **MAP EDITOR** button.

5. Use the three drop-down menus in the top navigation bar to select your:
    a. venue
    b. campus
    c. floor
6. Your newly created venue's resources (maps) will display.
7. From here, use the toolbar in the Map Editors upper righthand corner to add points of interest (POIs), waypoints, segments, routes and zones (see right).

> The Mapping API Rich Example assumes that your organization's application has already been created in the MaaS portal. If your organization was not provided with login credentials, contact your Phunware program manager.

This document provides examples that showcase how developers can build an example venue. Upon completion, the venue will appear in the Multiscreen as a Service (MaaS) portal's **Map Editor**, where credentialed users can then create points of interest (POIs), segments and routes for the newly created venue's map.

- 1. Create a Venue
- 2. Create a Campus
- 3. Create a Building
- 4. Create a Floor
- 5. Create a Resource
- 6. Retrieve a Venue
- 7. View the Venue in the Map Editor

For a venue to appear in the Map Editor, it must contain a building, campus, floor and (floor) resource, at a minimum. Here is a hierarchy of the items the developer will create:

- Hyrule (venue)
    - Hyrule Castle (campus)
        - Hyrule Castle Tower (building)
            - Level One (floor)
                - http://image.com/level1.pdf (resource)
                - http://image.com/level1.svg (resource)

All Mapping API objects—venues, campuses, buildings, floors, resources, points, POI types, POI type metadata, routes and segments—can be created at the API level. A subset of these will be created for this Mapping API Rich Example. See Mapping API Methods for the specific calls to make.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

HTTP Status Codes

ON THIS PAGE

Below is a detailed outline of the status codes and messages used in the API.

Status Code	Message	Description
200	Success	• Used for general success. • Used when a POST operation results in the creation of a new resource (such that response should contain URI of newly created resource).

202	Accepted	• Used to indicate that the message has been received, but processing may not yet be finished.
204	No Content	• Used for success when there is no response body to return.
400	Bad Request	• Used when one or more parameters are invalid. • Used when one or more required parameters are missing. • Used when the JSON payload is not valid JSON.
401	Unauthorized	• Used when the operation fails authorization checks.

403	Forbidden	• Used when the operation fails security checks (e.g. wrong app ID specified as input to retrieve specific data).
404	Not Found	• Used when the requested resource does not exist (e.g. app ID).
406	Not Acceptable	• Used when the content type passed in the request is not 'application/json'.

423	Locked	- Used to indicate that the resources is locked, such as during a process that takes longer than typical http response times, like route reset operations.
500	Internal Server Error	- Used when the (internal) server is down.

Error Payloads

Each failed response will include a JSON payload that includes the error code number and message:

400

The actual message for this "Bad Request" error will vary depending on whether the error was caused by bad data sent in the request, a validation error when validating the request data or any other generic request error.

```
{
    "error":
 {
        "code": 400,
        "message": <error
string>
    }
}
```

401

The actual message for this "Unauthorized" error will vary. The error string below will be one of the following:

- "Missing X-Auth header."
- "Auth header contains <num> parts; must be 3."
- "The request time is too far in the future (<num> seconds)."
- "The request time is too far in the past (<num> seconds)."
- "Invalid access key."
- "This access key has not been activated yet."
- "Could not decrypt the request."
- "Invalid signature."

```
{
    "error":
 {
        "code": 401,
        "message": <error
string>
    }
}
```

403

The actual message for this "Forbidden" error will vary.

```
{
    "error":
 {
        "code": 403,
        "message": <error
string>
    }
}
```

404

The actual message for this "Not Found" error will vary depending on what could not be found.

```
{
    "error":
 {
        "code": 404,
        "message": <error
string>
    }
}
```

406

The actual message for this "Not Acceptable" error will vary.

```
{
    "error":
 {
        "code": 406,
        "message": <error
string>
    }
}
```

423

```
{
    "data":
 {
        "code": 423,
        "message": "Resource
locked."
    }
}
```

500

```
{
    "data":
 {
        "code": 500,
        "message": An unknown
error occurred.
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

POI Types

Point of interest (POI) types are named types associated with an integer value. This is a list of the pre-defined types:

0	Unknown
1	Restroom
2	Drinking Fountain
3	Phone
4	Security Checkpoint
6	Information
9	Baggage Room
10	Valet
11	Luggage
12	Ticket Counter
13	Gate
14	Terminal
16	First Aid Station
17	Lost and Found
19	Luggage Carts
20	Lockers
21	Smoking Area
22	Airline Lounge
23	Business Area(s)
24	Play Area
25	Security Office
26	Management Office
27	Food Court
29	Service Counter
30	Cafeteria
36	Waiting Area
37	Meeting Area
43	Cafe
44	Conference Room
48	Education
49	Fitness
52	Mailroom
94	Picnic Area
95	Sports Field
96	Track
97	Water Feature
3578	ATM
4493	Marina
4581	Airport
5000	Business Facility
5400	Grocery Store
5511	Auto Dealerships
5540	Petrol/Gasoline Station
5571	Motorcycle Dealership
5800	Restaurant
5813	Nightlife
6000	Bank
6512	Shopping
7011	Hotel
7013	Other Accommodation
7389	Tourist Information
7510	Rental Car Agency
7522	Park & Ride
7538	Auto Service & Maintenance
7647	Park/Recreation Area
7832	Cinema
7897	Rest Area
7929	Performing Arts
7985	Casino
7990	Convention/Exhibition Centre
7992	Golf Course
7994	Civic/Community
9530	Post Office
9531	Banquet Hall
9532	Bar or Pub
9533	Cocktail Lounge
9534	Night Club
9535	Convenience Store
9536	Specialty Food Store
9537	Clothing Store
9538	Men's Apparel
9539	Shoe Store
9540	Specialty Clothing Store
9541	Women's Apparel
9542	Check Cashing Service
9543	Currency Exchange
9544	Money Transferring Service
9545	Department Store
9546	Discount Store
9547	Other General Merchandise
9548	Variety Store
9549	Garden Centre
9550	Glass & Window
9551	Hardware Store
9555	Paint
9556	Entertainment Electronics
9558	Furniture Store
9559	Major Appliance
9560	Home Specialty Store
9561	Computer & Software

53	Offices
54	Reception
55	Storage
57	Military Liaison
58	Customs
59	Fast Food
60	Charging Station
61	Shoe Shine
62	Massage
63	Spa
64	Vending Machine
65	Defibrillator - AED
66	Nursing Room
70	Pet Relief Area
72	Flight Monitor
73	Meditation Room
74	Seating
77	Wheelchair
79	Mail
80	Concourse
81	Aisle
82	Amusement Ride
83	Animal Enclosure
84	Aquarium
85	Booth
86	Campsite
87	Woodland
88	Drop Off Point
89	Gaming Area
90	Grass/Field
91	Hotel Room
92	Museum Exhibit
93	Pedestrian Path
	Centre
7996	Amusement Park
7997	Sports Centre
7998	Ice Skating Rink
7999	Tourist Attraction
8060	Hospital
8200	Higher Education
8211	School
8231	Library
8410	Museum
8699	Automobile Club
9121	City Hall
9221	Police Station
9500	Business Service
9501	Other Communication
9502	Telephone Service
9503	Cleaning & Laundry
9504	Hair & Beauty
9505	Health Care Service
9507	Photography
9508	Video & Game Rental
9509	Storage
9510	Tailor & Alteration
9511	Tax Service
9512	Repair Service
9513	Retirement/Nursing Home
9514	Social Service
9515	Utilities
9516	Waste & Sanitary
9518	Auto Parts
9519	Car Wash/Detailing
9520	Local Transit
9521	Travel Agent &
9562	Flowers & Jewelry
9563	Gift, Antique, & Art
9564	Optical
9565	Pharmacy
9566	Record, CD, & Video
9567	Specialty Store
9568	Sporting Goods Store
9569	Wine & Liquor
9570	Boating
9571	Theater
9574	Health Club
9575	Bowling Alley
9576	Sports Activities
9578	Attorney
9579	Dentist
9580	Physician
9581	Realtor
9583	Medical Service
9584	Police Service
9585	Veterinarian Service
9586	Sporting & Instructional Camp
9589	Public Restroom
9590	Residental Area/Building
9593	Transportation Service
9677	Recreation Centre
9715	Military Base
9719	Truck Dealership
9986	Home Improvement & Hardware Store
9987	Consumer Electronics Store
9988	Office Supply & Services Store

Ticketing		9989	Taxi Stand
9523	Church	9990	Premium Default
9525	Government Office	9992	Place of Worship
9527	Fire Department	9995	Bookstore
		9996	Coffee Shop
		20000	Elevator
		99999	Unknown

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Location API v1.2

This document describes the specification for the MaaS Location ("API"). This API allows remote clients to query for their devices' locations.

ON THIS PAGE

Security

All calls made must adhere to the guidelines presented in the MaaS REST API Security Protocol document.

Definitions

Term	Definition
JSON	Stands for JavaScript Object Notation and is used for the request and response formats due to its portability and simplicity.

RFC 3339	A date format that "provide[s] an unambiguous and well-defined method of representing dates and times." See http://www.ietf.org/rfc/rfc3339.txt for more details.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Location API Architecture

ON THIS PAGE

Request / Response Format

JSON is used for the request and response format. JSON is a lightweight and portable format that maintains human-readability.

When making POST or PUT requests, the request parameters MUST be JSON-encoded and placed in the request body. When making GET requests, the request parameters MUST be JSON-encoded, then URI-encoded and placed directly in the query string.

Clients should expect JSON-encoded responses for every request, even those that result in an error.

HTTP Method

Each method defines the HTTP method required depending on what the usage is. For example, if the method were used to simply get data, the HTTP method would be GET. If the method is used to delete data it uses DELETE.

Date / Time Format

The date and time MUST be expressed in RFC 3339 format: YYYY-MM-DDThh:mm:ssZ

- All times must be in the UTC time zone and indicated with a "Z". This is done to mitigate issues regarding Daylight Saving Time (DST).

- No fractional seconds are allowed.
- See section 5.6 of Date and Time on the Internet: Timestamps for more information.

Bracketed IDs ("[id]"s)

Whenever a bracketed ID ("[id]") appears in a method's name or URL (e.g. "GET /containers/[id]" or "http://cms-api.phunware.com/v1.0/containers/[id]"), treat it as a placeholder for the ID number of the item involved in the method (in this case, "GET /containers/12345" or "http://cms-api.phunware.com/v1.0/containers/12345").

Ellipses ("…"s)

Whenever an ellipsis ("…") appears in the JSON body, treat it as a placeholder for additional instances of the data that immediately precedes it.

PUT Methods & Partial Updates

There are a few differences in how the request parameters get handled by PUT methods:

- There are no required parameters.
- If a parameter IS NOT specified, it will retain its current value.
- If a parameter IS specified, but with an empty value, then the value will be cleared.
- If you pass in an identifier field for a container, schema, structure or content ID, it will be ignored.

GET Methods & Query Strings

The GET methods outlined in this document will use query strings, not JSON bodies, when making a request to a URL. This requires the user to convert JSON into a URL-encoded query string. Here's an example:

```
{
 "tags":
 {
  "any":
  [
   "Zelda",
   "Link"
  ]
 }
}
```

Minified fragment (using http://bigaqua.org/minify_json.html).

```
{"tags":{"any":["Zelda","Link"
]}}
```

URL-encoded, minified fragment (using http://www.url-encode-decode.com/):

```
%7B%3Ftags%3F%3A%7B%3Fany%3F%3
A%5B%3FZelda%3F%2C%3FLink%3F%5
D%7D%7D
```

This URL-encoded, minified fragment is the query string. In order to compose a request, a "?" is appended to the end of the request URL, then the URL-encoded, minified fragment is added after that:

```
http://location-api.phunware.c
om/v1.2/coordinates?%7B%3Ftags
%3F%3A%7B%3Fany%3F%3A%5B%3FZel
da%3F%2C%3FLink%3F%5D%7D%7D
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Retrieve Location Data

This method is used to return a device's location coordinates and venue map data.

ON THIS PAGE

Method

GET

URL

http://location-api.phunware.com/v1.2/coordinates

Headers

X-Auth (see Security)

Query Parameters

Required parameters are underlined.

Parameter	Value	Description
macAddress	string	The hashed MAC address of the device requesting coordinates, obfuscated using the HMAC SHA-256 algorithm. An example MAC address of "01:23:45:6", when hashed using http://www.movable-type.co.uk/scripts/sha256.html, results in: "8e6549939116b0ae0c4c6997355ba6086a81799f27c97711de5003026dbcc0d9".)
venueGuid	string	The identifier of the venue to retrieve map data from.

Example Query Fragment

```
{
 "macAddress":
"8e6549939116b0ae0c4c6997355ba6086a81799f27c97711de5003026dbcc0d9",
 "venueGuid":
"e2a53fd8-6dcc-48e1-8b33-71f12fc13966"
}
```

URL-encoded, minified fragment:

```
%7B%22macAddress%22%3A%228e6549939116b0ae0c4c6997355ba6086a81799f27c97711de5003026dbcc0d9%22%2C%22venueGuid%22%3A%22e2a53fd8-6dcc-48e1-8b33-71f12fc13966%22%7D
```

Example Request

GET
`http://location-api.phunware.com/v1.2/coordinates?%7B%22macAddress%22%3A%228e6549939116b0ae0c4c6997355ba6086a81799f27c97711de5003026dbcc0d9%22%2C%22venueGuid%22%3A%22e2a53fd8-6dcc-48e1-8b33-71f12fc13966%22%7D`

Response

A successful response will have a 200 status code and a body containing location data:

```
{
    "buildingId": <integer>,
    "floorId": <integer>,
    "level": <integer>,
    "macAddress": <string>,
    "x": <float>,
    "y": <float>,
    "location":
 {
        "latitude": <float>,
        "longitude": <float>,
        "accuracy": <integer>
    },
    "updatedAt": <string>
}
```

See Location API Response Handling for error payloads.

Example Response

Example body of a successful response:

```
{
    "buildingId": 32,
    "floorId": 45,
    "level": 2,
    "macAddress":
"8d969eef6ecad3c29a3a629280e68
6cf0c3f",
    "x": 40.59,
    "y": 39.49,
    "location":
 {
        "latitude":
30.35999462,
        "longitude":
-97.74246194,
        "accuracy": 123
    },
    "updatedAt":
"2002-10-02T10:00:00Z"
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

HTTP Status Codes

ON THIS PAGE

Below is a detailed outline of the status codes and messages used in the API.

Status Code	Message	Description
200	Success	• Used for general success. • Used when a POST operation results in the creation of a new resource (such that response should contain URI of newly created resource).

400	Bad Request	• Used when one or more parameters are invalid. • Used when one or more required parameters are missing. • Used when the JSON payload is not valid JSON.
404	Not Found	• Used when the requested resource does not exist (e.g. app ID).

Error Payloads

Each failed response will include a JSON payload that includes the error code number and message:

400

The actual message for this "Bad Request" error will vary depending on whether the error was caused by bad data sent in the request, a validation error when validating the request data or any other generic request error.

```
{
    "error":
 {
        "code": 400,
        "message": <error
string>
    }
}
```

404

The actual message for this "Not Found" error will vary depending on what could not be found.

```
{
    "error":
 {
        "code": 404,
        "message": <error
string>
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

MARS API v1.1

ON THIS PAGE

This document describes the specification for the MaaS MARS API ("API") and the contract between the API and the mobile SDKs. It describes the payload data delivered by the Cisco MSE for processing by the MARS server.

Definitions

Term	Definition

MSE	The Cisco Mobility Services Engine (MSE) is a network appliance that provides tools for wireless network monitoring and network asset location tracking.
MARS	MAC Address Resolution Service
Venue	A location where events take place, possibly consisting of many campuses.
Campus	A collection of buildings grouped together by a common theme.
Building	A physical structure that may contain one or more floors.
Floor	An object associated with a building containing building ID, floor ID, zoom level and resource URL(s) (e.g. .svg, .pdf).
Resource	Files associated with a floor. Contents contain an .svg or .pdf asset URL, along with associated metadata. These image files are often referred to as a map.
JSON	Stands for JavaScript Object Notation and is used for the request and response formats due to its portability and simplicity.

RFC 3339	A date format that "provide[s] an unambiguous and well-defined method of representing dates and times." See http://www.ietf.org/rfc/rfc3339.txt for more details.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

MARS API Architecture

ON THIS PAGE

Request / Response Format

JSON is used for the request and response format. JSON is a lightweight and portable format that maintains human-readability.

When making POST or PUT requests, the request parameters MUST be JSON-encoded and placed in the request body. When making GET requests, the request parameters MUST be JSON-encoded, then URI-encoded and placed directly in the query string.

Clients should expect JSON-encoded responses for every request, even those that result in an error.

HTTP Method

Each method defines the HTTP method required depending on what the usage is. For example, if the method is used to simply get data, the HTTP method would be GET. If the method is used to delete data it uses DELETE.

Date / Time Format

The date and time MUST be expressed in RFC 3339 format: YYYY-MM-DDThh:mm:ssZ

- All times must be in the UTC time zone and indicated with a "Z". This is done to mitigate issues regarding Daylight Saving Time (DST).

- No fractional seconds are allowed.
- See section 5.6 of Date and Time on the Internet: Timestamps for more information.

Ellipses ("…"s)

Whenever an ellipsis ("…") appears in the JSON body, treat it as a placeholder for additional instances of the data that immediately precedes it.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

MARS API Methods

This section describes the methods in use with our MARS API.

Register with MARS, Obtair

Deliver the MAC Address to th

AssociationEvent Request Boc

AssociationEvent Example Re

ContainmentEvent Request Bc

ContainmentEvent Example R

AbsenceEvent Request Body

AbsenceEvent Example Requ

MovementEvent Request Body

MovementEvent Example Req

Register or Update with Cisco

Upload Map Image Data for a

Delete Full Bind Record

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Register with MARS, Obtain a MAC Address

This method is used to register with MARS and attempt to retrieve the hashed MAC address.

ON THIS PAGE

Method

POST

URL

http://localhost:8889/mars-api.phunware.com/v1.1/client

Request Body

Required parameters are underlined.

Parameter	Value	Description
timestamp	string	The date and time in RFC 3339. See MARS API Architecture > Date / Time Format.
schemaVersion	string	Specifies the version of input schema to be used.
device	object	An object containing key-value pairs. Required keys are underlined: - deviceId (string): The unique identifier for a registered device.

		• apBSSID (string): The AP BSSID of the Wi-Fi access point. If neither the AP BSSID nor IP address is provided, then service will do a MAC address lookup for a given device ID. • ipAddress (string): The device's IP address. If neither the AP BSSID nor IP address is provided, then service will do a MAC address lookup for a given device ID. • platformId (string): The OS version. Accepted values: "iOS", "Android"

app	object	An object containing key-value pairs. Required keys are underlined: • applicationId (string): The identifier of the MaaS application. • venueId (string): The identifier of the venue. Obtained from the mapping API call.

```
{

"timestamp"
: <string>,

"schemaVers
ion":
<string>,

"device":
 {

"deviceId":
<string>,

"apBSSID":
<string>,

"ipAddress"
: <string>,

"platformId
": <string>
    },
    "app":
 {

"applicatio
nId":
<string>,

"venueId":
<string>
    }
}
```

Example Request

POST the following request body to: http://mars-api.phunware.com/v1.1/client/

```
{

"timestamp"
:
"2008-09-08
T22:47:31Z"
,
"schemaVers
ion":
"1.0",

"device":
 {

//This
example is
using the
Android-for
matted
device ID.
  //iOS
devices use
the UUID
format for
their
device IDs.

//(e.g."256
44720-8CED-
4E40-96F5-1
6102F90D2A8
")

"deviceId":
"9161ade538
b097cb",

"apBSSID":
"00:11:22:3
3:44:55:66"
,

"ipAddress"
:
"192.168.0.
10",

"platformId
":
"android"
    },
    "app":
 {

"clientId":
"244",

"venueGuid"
:
```

```
"7b99ab29-4
a74-44f6-b8
49-8da66220
```

```
5552"
    }
}
```

Response

A successful response will have a 200 status code and a body containing MAC address data:

```
{

"data":
 {

"hashedMACA
ddress":
<string>,

"deviceId":
<string>
    }
}
```

See MARS API Response Handling for error payloads.

Example Response

Example body of successful response:

```
{

"data":
 {

"hashedMACA
ddress":
"099a490bfd
494a071eea2
0dd66d5780d
d260c7",

"deviceId":
"9161ade538
b097cb"
    }
}
```

Hashing

The device MAC address returned to the client is obfuscated using the HMAC SHA-256 algorithm.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Deliver the MAC Address to the MARS Server

This method is issued by Cisco MSE to deliver MAC address to MARS server.

When a user requests a new venue to be added via the MaaS portal, the portal will generate two different endpoints: a registration endpoint and notifications endpoint.

ON THIS PAGE

Method

POST

URL

http://localhost:8889/mars-api.phunware.com/v1.1/api/southbound/[guid]

Request Bodies

Request body payloads will depend upon the Cisco MSE message type being created:

- AssociationEvent Request Body
- ContainmentEvent Request Body
- AbsenceEvent Request Body
- MovementEvent Request Body

Example Requests

POST the request body to: http://localhost:8889/mars-api.phunware.com/v1.1/api/southbound/6178e93f51604e349f6f8daa10d40dd4106d37bf

Example request bodies for the different Cisco MSE message types can be found here:

- AssociationEvent Example Requests
- ContainmentEvent Example Requests
- AbsenceEvent Example Requests
- MovementEvent Example Requests

Response

A success response will return a 200 status code with no data in the body of the response. See MARS API Response Handling for error payloads.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

AssociationEvent Request Body

This page is a continuation of the **Request Bodies** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

Request body payloads will depend upon the Cisco MSE message type being created:

- **AssociationEvent Request Body**
- ContainmentEvent Request Body
- AbsenceEvent Request Body
- MovementEvent Request Body

The following fields exist within a nested AssociationEvent object. Required fields are underlined.

v.7	v.8	Parameter	Value	Descriptio
X	X	subscriptionName	string	The name
X	X	entity	string	The source
X	X	deviceId	string	The device
	X	confidenceFactor	float	A numeric the percen coordinate
X	X	locationMapHierarchy	string	A concaten building na separated to identify t "Phunware

X	X	locationCoordinate	object	An object c of location • x (float locatio • y (float locatio • unit (st measu
	X	mseUdi	string	The unique
X	X	geoCoordinate	object	An object c of geograp MSE whil "latit • lat[t]itu coordir decima • longitu coordir decima • unit (st measu
	X	floorRefId	integer	The unique Cisco MSE
X	X	association	Boolean	Indicates w associated
X	X	ipAddress	array	An array of strings.
X	X	apMacAddress	string	The AP MA access poi address nc then servic for a given
	X	status	integer	A numeric associatior (for Associ
	X	username	string	Username username- identifiers (name.

	X	ssid	string	The name device is c Wi-Fi").
	X	band	string	The GHz b point recei usually 2.4
	X	dot11Status	string	A string inc associatior "ASSOCIA "DELETED
	X	guestUser	Boolean	Indicates w device is a
X	X	timestamp	string	The date a

MSE 7 Payload

For version 7 of MSE, construct the following payload:

```
{

"Associatio
nEvent":
 {

"subscripti
onName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"locationMa
pHierarchy"
: <string>,

"locationCo
ordinate":
  {

"x":
<float>,

"y":
```

```
<float>,

"unit":
<string>
        },

"geoCoordin
ate":
  {

"lattitude"
: <float>,

"longitude"
: <float>,

"unit":
<string>
        },

"timestamp"
: <string>,

"associatio
n":
<Boolean>,

"apMacAddre
ss":
<string>,

"ipAddress"
:
  [

<string>,

<string>
```

```
        ]
    }
}
```

MSE 8 Payload

For version 8 of MSE, construct the following payload:

```
{

"Associatio
nEvent":
 {

"subscripti
onName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"confidence
Factor":
<float>,

"locationMa
pHierarchy"
: <string>,

"locationCo
ordinate":
  {

"x":
<float>,

"y":
<float>,

"unit":
<string>
        },

"mseUdi":
<string>,
"geoCoordin
ate":
  {

"latitude":
<float>,

"longitude"
: <float>,

"unit":
<string>
        },

"floorRefId
":
<integer>,

"associatio
n":
<Boolean>,

"ipAddress"
:
  [

<string>,

<string>
        ],

"apMacAddre
ss":
<string>,

"status":
<integer>,

"username":
<string>,

"ssid":
<string>,

"band":
<string>,

"dot11Statu
s":
<string>,

"guestUser"
:
```

		`<Boolean>,`
		`"timestamp"`		

```
: <string>
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

AssociationEvent Example Requests

This page is a continuation of the **Example Requests** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

POST the request body to: http://localhost:8889/mars-api.phunware.com/v1.1/api/southbound/6178e93f51604e349f6f8daa10d40dd4106d37bf

Example request bodies for the other Cisco MSE message types can be found here:

- **AssociationEvent Example Requests**
- ContainmentEvent Example Requests
- AbsenceEvent Example Requests
- MovementEvent Example Requests

MSE 7

For version 7 of MSE, construct the following payload:

```
{
```

```
"AssociationEvent":
 {

"subscriptionName":
"CMX_Association_Event",

"entity":
"WIRELESS_CLIENTS",

"deviceId":
"18:af:61:51:6c:77",

"locationMapHierarchy":
"Phunware_Austin>South>1st_Floor",

"locationCoordinate":
  {

"x": 90.64,

"y": 80.26,

"unit":
"FEET"
        },

"geoCoordinate":
  {

"lattitude":
33.6600694426,

"longitude":
-117.8614988409,

"unit":
```

```
"DEGREES"
        },
"timestamp": "2014-02-18T20:02:41.218+0000",
"association": true,
"apMacAddress": "e8:ed:f3:1b:2e:60",
"ipAddress":
  [
"10.180.181.99",
"fe80:0000:0000:0000:10c9:596b:3448:1797"
```

```
        ]
    }
}
```

MSE 8

For version 8 of MSE, construct the following payload:

```
{

"Associatio
nEvent":
 {

"subscripti
onName":
"CMX_Associ
ation_Event
",

"entity":
"WIRELESS_C
LIENTS",

"deviceId":
"18:af:61:5
1:6c:77",

"confidence
Factor":
64.0,

"locationMa
pHierarchy"
:
"Phunware_A
ustin>South
>1st_Floor"
,

"locationCo
ordinate":
  {

"x": 90.64,

"y": 80.26,

"unit":
```

```
"FEET"
        },

"mseUdi":
"AIR-MSE-VA
-K9:V01:att
-corpmse01.
phunware.co
m_31b",

"geoCoordin
ate":
  {

"latitude":
33.66006944
26,

"longitude"
:
-117.861498
8409,

"unit":
"DEGREES"
        },

"floorRefId
":
74883379723
5015762,

"associatio
n": true,

"ipAddress"
:
  [

"10.180.181
.99",

"fe80:0000:
0000:0000:1
0c9:596b:34
48:1797"
        ],

"apMacAddre
ss":
"e8:ed:f3:1
b:2e:60",
"status": 3,

"username": "",

"ssid": Phunware,

"band": "5.0",

"dot11Status": "ASSOCIATED",

"guestUser": false,

"timestamp": "2014-02-18T20:02:41.2
```

```
18+0000"
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

ContainmentEvent Request Body

This page is a continuation of the **Request Bodies** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

Request body payloads will depend upon the Cisco MSE message type being created:

- AssociationEvent Request Body
- **ContainmentEvent Request Body**
- AbsenceEvent Request Body
- MovementEvent Request Body

The following fields exist within a nested ContainmentEvent object. Required fields are underlined.

v.7	v.8	Parameter	Value	Description
X	X	subscriptionName	string	The name of th
X	X	entity	string	The source age
X	X	deviceId	string	The device's M
	X	confidenceFactor	float	A numeric valu the percent acc coordinate retu

X	X	locationCoordinate	object	An object conta of location coor - x (float): Th location. - y (float): Th location. - unit (string) measurem
	X	mseUdi	string	The unique Cis
X	X	geoCoordinate	object	An object conta of geographica MSE 7 a while MS "latitude' - lat[t]itude (f coordinate decimal de - longitude (f coordinate decimal de - unit (string) measurem
	X	floorRefId	integer	The unique ID Cisco MSE dat
X	X	boundary	string	Defines if the u "OUTSIDE" of breach. The M floor breaches, "INSIDE".
X	X	areaType	string	The location ob seen on. Accep "CAMPUS", "B and "ZONE". T uses floor brea returns "FLOO
X	X	containerHierarchy	string	A concatenatio building name device was see '>' character. U uniquely (e.g. "Phunware_Au
X	X	timestamp	string	The date and ti

MSE 7 Payload

For version 7 of MSE, construct the following payload:

```
{

"Containmen
tEvent":
 {

"subscripti
onName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"locationCo
ordinate":
  {

"x":
<float>,

"y":
<float>,

"unit":
<string>
       },

"geoCoordin
ate":
  {

"lattitude"
: <float>,

"longitude"
: <float>,

"unit":
<string>
       },

"boundary":
<string>,
```

		"areaType": <string>, "containerH ierarchy": <string>, "timestamp"	

```
: <string>
    }
}
```

MSE 8 Payload

For version 8 of MSE, construct the following payload:

```
{

"Containmen
tEvent":
 {

"subscripti
onName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"confidence
Factor":
<float>,

"locationCo
ordinate":
  {

"x":
<float>,

"y":
<float>,

"unit":
<string>
        },

"mseUdi":
<string>,

"floorRefId
":
<integer>,
"boundary":
<string>,

"areaType":
<string>,

"geoCoordin
ate":
  {

"latitude":
<float>,

"longitude"
: <float>,

"unit":
<string>
        },

"containerH
ierarchy":
<string>,

"timestamp"
```

```
: <string>
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

ContainmentEvent Example Requests

This page is a continuation of the **Example Requests** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

POST the request body to: http://localhost:8889/mars-api.phunware.com/v1.1/api/southbound/6178e93f51604e349f6f8daa10d40dd4106d37bf

Example request bodies for the other Cisco MSE message types can be found here:

- AssociationEvent Example Requests
- **ContainmentEvent Example Requests**
- AbsenceEvent Example Requests
- MovementEvent Example Requests

MSE 7

For version 7 of MSE, construct the following payload:

```
{
```

```
"Containmen
tEvent":
 {

"subscripti
onName":
"CMX_Contai
nment_Event
",

"entity":
"WIRELESS_C
LIENTS",

"deviceId":
"fc:c2:de:9
d:58:62",

"locationCo
ordinate":
  {

"x":
254.4425,

"y":
70.65469,

"unit":
"FEET"
        },

"geoCoordin
ate":
  {

"lattitude"
:
37.40517472
7690955,

"longitude"
:
-121.976596
94880194,

"unit":
"DEGREES"
        },

"boundary":
"INSIDE",
"areaType":
"FLOOR",

"containerH
ierarchy":
"Phunware_A
ustin>South
>1st_Floor"
,

"timestamp"
:
"2014-10-10
T07:41:58.7
```

```
10-0700"
    }
}
```

MSE 8

For version 8 of MSE, construct the following payload:

```
{

"Containmen
tEvent":
 {

"subscripti
onName":
"CMX_Contai
nment_Event
",

"entity":
"WIRELESS_C
LIENTS",

"deviceId":
"fc:c2:de:9
d:58:62",

"confidence
Factor":
24,

"locationCo
ordinate":
  {

"x":
254.4425,

"y":
70.65469,

"unit":
"FEET"
        },

"mseUdi":
"AIR-MSE-VA
-K9:V01:MSE
```

```
-VA-77_32af
66dc-bb7b-1
1e3",

"floorRefId
":
46980412192
91283000,

"boundary":
"INSIDE",

"areaType":
"FLOOR",

"geoCoordin
ate":
  {

"latitude":
37.40517472
7690955,

"longitude"
:
-121.976596
94880194,

"unit":
"DEGREES"
        },

"containerH
ierarchy":
"Phunware_A
ustin>South
>1st_Floor"
,

"timestamp"
:
"2014-10-10
T07:41:58.7
```

```
10-0700"
     }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

AbsenceEvent Request Body

This page is a continuation of the **Request Bodies** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

Request body payloads will depend upon the Cisco MSE message type being created:

- AssociationEvent Request Body
- ContainmentEvent Request Body
- **AbsenceEvent Request Body**
- MovementEvent Request Body

The following fields exist within a nested AbsenceEvent object. Required fields are underlined.

v.7	v.8	Parameter	Value	Descri
X	X	subscriptionName	string	The na
X	X	entity	string	The so
X	X	deviceId	string	The de

	X	confidenceFactor	float	A nume the per coordin
X	X	locationMapHierarchy	string	A conc building separa to ident "Phunw
X	X	locationCoordinate	object	An obje of locat • x (f loc • y (f loc • uni me
	X	mseUdi	string	The un
X	X	geoCoordinate	object	An obje of geog I V ' • lat[coo dec • lon coo dec • uni me
	X	floorRefId	integer	The un Cisco M
X	X	absenceDurationInMinutes	integer	The nu user's (MSE.
X	X	lastSeen	string	The tim saw the
X	X	timestamp	string	The da

MSE 7 Payload

For version 7 of MSE, construct the following payload:

```
{

"AbsenceEvent":
 {

"subscriptionName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"locationMapHierarchy"
: <string>,

"locationCoordinate":
  {

"x":
<float>,

"y":
<float>,

"unit":
<string>
        },

"absenceDurationInMinutes":
<integer>,

"geoCoordinate":
  {

"lattitude"
: <float>,

"longitude"
: <float>,
"unit":
<string>
        },

"lastSeen":
<string>,

"timestamp"
```

```
: <string>
    }
}
```

MSE 8 Payload

For version 8 of MSE, construct the following payload:

```
{

"AbsenceEve
nt":
 {

"subscripti
onName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"confidence
Factor":
<float>,

"locationMa
pHierarchy"
: <string>,

"locationCo
ordinate":
  {

"x":
<float>,

"y":
<float>,

"unit":
<string>
        },

"mseUdi":
<string>,
"geoCoordin
ate":
  {

"latitude":
<float>,

"longitude"
: <float>,

"unit":
<string>
        },

"floorRefId
":
<integer>,

"absenceDur
ationInMinu
tes":
<integer>,

"lastSeen":
<string>,

"timestamp"
```

```
: <string>
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

AbsenceEvent Example Requests

This page is a continuation of the **Example Requests** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

POST the request body to: http://localhost:8889/mars-api.phunware.com/v1.1/api/southbound/6178e93f51604e349f6f8daa10d40dd4106d37bf

Example request bodies for the other Cisco MSE message types can be found here:

- AssociationEvent Example Requests
- ContainmentEvent Example Requests
- **AbsenceEvent Example Requests**
- MovementEvent Example Requests

MSE 7

For version 7 of MSE, construct the following payload:

```
{
```

```
"AbsenceEvent":
 {
"subscriptionName": "CMX_Absence_Event",
"entity": "WIRELESS_CLIENTS",
"deviceId": "9c:04:eb:a5:94:49",
"locationMapHierarchy": "Phunware_Austin>South>1st_Floor",
"locationCoordinate":
  {
"x": 223.45,
"y": 23.44,
"unit": "FEET"
        },
"absenceDurationInMinutes": 30,
"geoCoordinate":
  {
"lattitude": 30.3598096444,
"longitude":
```

```
-97.7419564
503,

"unit":
"DEGREES"
        },

"lastSeen":
"2014-10-10
T14:23:40.5
39+0000",

"timestamp"
:
"2014-10-10
T14:54:10.5
```

```
40+0000"
    }
}
```

MSE 8

For version 8 of MSE, construct the following payload:

```
{

"AbsenceEve
nt":
 {

"subscripti
onName":
"CMX_Absenc
e_Event",

"entity":
"WIRELESS_C
LIENTS",

"deviceId":
"9c:04:eb:a
5:94:49",

"confidence
Factor":
50,

"locationMa
pHierarchy"
:
"Phunware_A
ustin>South
>1st_Floor"
,

"locationCo
ordinate":
  {

"x":
223.45,

"y": 23.44,

"unit":
```

```
"FEET"
        },

"mseUdi":
"AIR-MSE-VA
-K9:V01:att
-corpmse01.
phunware.co
m_31b",

"geoCoordin
ate":
  {

"latitude":
30.35980964
44,

"longitude"
:
-97.7419564
503,

"unit":
"DEGREES"
        },

"floorRefId
":
74883379723
5015700,

"absenceDur
ationInMinu
tes": 30,

"lastSeen":
"2014-10-10
T14:23:40.5
39+0000",

"timestamp"
:
"2014-10-10
T14:54:10.5
```

```
40+0000"
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

MovementEvent Request Body

This page is a continuation of the **Request Bodies** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

Request body payloads will depend upon the Cisco MSE message type being created:

- AssociationEvent Request Body
- ContainmentEvent Request Body
- AbsenceEvent Request Body
- **MovementEvent Request Body**

The following fields exist within a nested MovementEvent object. Required fields are underlined.

v.7	v.8	Parameter	Value	Descriptio
X	X	subscriptionName	string	The name
X	X	entity	string	The source
X	X	deviceId	string	The device

	X	confidenceFactor	float	A numeric indicating t location co
X	X	locationMapHierarchy	string	A concate name, buil separated to identify t "Phunware
X	X	locationCoordinate	object	An object o of location • x (float locatio • y (float locatio • unit (st measu
	X	mseUdi	string	The unique
X	X	geoCoordinate	object	An object o of geograp MSE [sic] "latit • lat[t]itu coordi decima • longitu coordi decima • unit (st measu
	X	floorRefId	integer	The unique Cisco MSE
X	X	moveDistanceInFeet	integer	The displa from the la coordinate
	X	currentlyTracked	Boolean	Indicates w Movement client devic
	X	ipAddress	array	An array o strings.

	X	ssid	string	The name device is c Wi-Fi").
	X	band	string	The GHz b point recei usually 2.4
	X	dot11Status	string	A string inc association "ASSOCIA "DELETED
	X	apMacAddress	string	The AP MA access poi address no then servic lookup for
	X	guestUser	Boolean	Indicates w device is a
X	X	timestamp	string	The date a

MSE 7 Payload

For version 7 of MSE, construct the following payload:

```
{

"MovementEv
ent":
 {

"subscripti
onName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"locationMa
pHierarchy"
: <string>,

"locationCo
ordinate":
  {

"x": <float>,

"y": <float>,

"unit": <string>
        },

"geoCoordinate":
  {

"lattitude": <float>,

"longitude": <float>,

"unit": <string>
        },

"moveDistanceInFt": <float>,

"timestamp"
```

```
: <string>
    }
}
```

MSE 8 Payload

For version 8 of MSE, construct the following payload:

```
{

"MovementEv
ent":
 {

"subscripti
onName":
<string>,

"entity":
<string>,

"deviceId":
<string>,

"confidence
Factor":
<float>,

"locationMa
pHierarchy"
: <string>,

"locationCo
ordinate":
  {

"x":
<float>,

"y":
<float>,

"unit":
<string>
        },

"mseUdi":
<string>,
"floorRefId
":
<integer>,

"moveDistan
ceInFt":
<float>,

"currentlyT
racked":
<Boolean>,

"ipAddress"
:
  [

<string>,

<string>
        ],

"ssid":
<string>,

"band":
<string>,

"dot11Statu
s":
<string>,

"apMacAddre
ss":
<string>,

"guestUser"
:
<Boolean>,

"geoCoordin
ate":
  {

"latitude":
<float>,

"longitude"
: <float>,

"unit":
<string>
        },
```

		`"timestamp"`		

```
: <string>
    }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

MovementEvent Example Requests

This page is a continuation of the **Example Requests** section of the Deliver the MAC Address to the MARS Server method.

ON THIS PAGE

POST the request body to: http://localhost:8889/mars-api.phunware.com/v1.1/api/southbound/6178e93f51604e349f6f8daa10d40dd4106d37bf

Example request bodies for the other Cisco MSE message types can be found here:

- AssociationEvent Example Requests
- ContainmentEvent Example Requests
- AbsenceEvent Example Requests
- **MovementEvent Example Requests**

MSE 7

For version 7 of MSE, construct the following payload:

```
{
```

```
"MovementEvent":
 {
"subscriptionName": "CMX_Location_Event_PROD_NEW",
"entity": "WIRELESS_CLIENTS",
"deviceId": "cc:3a:61:27:1c:d2",
"locationMapHierarchy": "Phunware_Austin>South>1st_Floor",
"locationCoordinate":
  {
"x": 23.415796,
"y": 43.35957,
"unit": "FEET"
        },
"geoCoordinate":
  {
"lattitude": 30.359974591172836,
"longitude": -97.74253279388846,
"unit":
"DEGREES"
        },

"moveDistan
ceInFt":
32.533466,

"timestamp"
:
"2014-10-10
T14:53:25.4
```

```
82+0000"
    }
}
```

MSE 8

For version 8 of MSE, construct the following payload:

```
{

"MovementEv
ent":
 {

"subscripti
onName":
"CMX_Locati
on_Event_PR
OD_NEW",

"entity":
"WIRELESS_C
LIENTS",

"deviceId":
"cc:3a:61:2
7:1c:d2",

"confidence
Factor":
75.0,

"locationMa
pHierarchy"
:
"Phunware_A
ustin>South
>1st_Floor"
,

"locationCo
ordinate":
  {

"x":
23.415796,

"y":
43.35957,
"unit":
"FEET"
        },

"mseUdi":
"AIR-MSE-VA
-K9:V01:att
-corpmse01.
phunware.co
m_31b",

"floorRefId
":
74883379723
5015743,

"moveDistan
ceInFt":
32.533466,

"currentlyT
racked":
true,

"ipAddress"
:
  [

"10.180.181
.99",

"fe80:0000:
0000:0000:1
0c9:596b:34
48:1797"
        ],

"ssid":
Phunware,

"band":
"5.0",

"dot11Statu
s":
"ASSOCIATED
",

"apMacAddre
ss":
"e8:ed:f3:1
b:2e:60",
"guestUser"
: false,

"geoCoordin
ate":
  {

"latitude":
30.35997459
1172836,

"longitude"
:
-97.7425327
9388846,

"unit":
"DEGREES"
        },

"timestamp"
:
"2014-10-10
T14:53:25.4
```

```
82+0000"
     }
}
```

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Register or Update with Cisco MSE

This method is used to register or update a Cisco MSE. Given the list of floors, and associated information, this will populate the identified venue's campuses, buildings and floors. The response will then contain API URLs for each map resource that needs to be uploaded.

ON THIS PAGE

Method

POST

URL

http://mars-api.phunware.com/v1.1/mse/[guid]/register

Request Body

Required parameters are underlined.

Parameter	Value	Description
floors	array	

					An array of floor objects containing key-value pairs. Required keys are underlined: • name (string): The unique name of the floor. • level (integer): The floor level within the building. • isOutdoor (Boolean): Indicates whether the floor is outdoor. • length (integer): The length of this floor. • height (integer): The height of this floor. • offsetX (integer): The X offset of the floor within the map image. • offsetY (integer): The Y offset of the floor within the map image.		

- textHierarchy (array): An array containing the campus, building and floor name (in that order).
- imageExists (Boolean): Indicates whether an image exists for this floor. If true, the response will contain an item referencing an API URL where the actual resource data should be posted.
- imageName (string): The name of the image, if it exists. This is only used to send back in the response.
- imageType (string): The MIME type of the image, if it exists. This is only used to send back in the response.

```
{

"floors":
 [
        {

"name":
<string>,

"level":
<integer>,

"isOutdoor"
:
<Boolean>,

"length":
<integer>,

"height":
<integer>,

"offsetX":
<integer>,

"offsetY":
<integer>,

"textHierar
chy":
   [

<string>,

<string>,

<string>

],

"imageName"
: <string>,

"imageType"
: <string>,

"imageExist
s":
<Boolean>
        }
    ]
}
```

Example Request

POST the following request body to: http://mars-api.phunware.com/v1.1/mse/f0685279-7497-4d90-882e-12201e476b0e/register/

```
{

"floors":
 [
        {

"name":
"WNBU",

"level": 4,

"isOutdoor"
: false,

"length":
185,

"height":
10,

"offsetX":
5,

"offsetY":
0,

"textHierar
chy":
   [

"Cisco Site
4",

"SJ-14",

"WNBU"

],

"imageName"
:
"domain_0_1
34931105571
8.png",
"imageType"
:
"image/png"
,

"imageExist
s": true
```

```
        }
    ]
}
```

Response

A success response will return a 200 status code with no data in the body of the response. See MARS API Response Handling for error payloads.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Upload Map Image Data for a Floor

This method is used to upload the map image data for a particular floor. The request must use "*multipart/form-data"* as the value for the "Content-Type" header.

ON THIS PAGE

Method

POST

URL

http://mars-api.phunware.com/v1.1/mse/[guid]/[floorId]/upload-resource

Request Body

Required parameters are underlined.

Parameter	Value	Description

image	binary	The binary file data.

Example Request

POST
`http://mars-api.phunware.com/v1.1/mse/f0685279-7497-4d90-882e-12201e476b0e/17/upload-resource`

Response

A success response will return a 200 status code with no data in the body of the response. See MARS API Response Handling for error payloads.

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

Delete Full Bind Record

This method is used to Delete full bind record from Memcache. Whenever a device is not receiving location updates from MSE, you can start troubleshooting by deleting fullbind record associated with that device from Memcache and this is the endpoint you will use for deleting a record from Memcache. This endpoint is used mostly for troubleshooting purposes by SDK team.

Method :

DELETE

URL :

http://localhost:8889/mars-api.phunware.com/v1.1/bind/[DeviceMacAddr]

Ex : http://localhost:8889/mars-api.phunware.com/v1.1/bind/25:Ge:00:00:00:KP

Response :

A successful response will have a 200 status code and a body containing following message :

Successfully deleted fullbind record from Memcache

This documentation is no longer actively supported and may be out of date. Going forward, please visit and bookmark our new site (https://docs.phunware.com/) for up-to-date documentation.

HTTP Status Codes

ON THIS PAGE

Below is a detailed outline of the status codes and messages used in the API.

Status Code	Message	Description

200	Success	• Used for general success. • Used when a POST operation results in the creation of a new resource (such that response should contain URI of newly created resource).
404	Not Found	• Used when the requested resource does not exist (e.g. app ID). • Either the MAC address or device ID cannot be resolved.
500	Internal Server Error	• Used when the (internal) server is down.

Error Payloads

Each failed response will include a JSON payload that includes the error code number and message:

400

The actual message for this "Bad Request" error will vary depending on whether the error was caused by bad data sent in the request, a validation error when validating the request data or any other generic request error.

```
{
    "data":
 {
        "code": 400,
        "message": <error
string>
    }
}
```

404

The actual message for this "Not Found" error will vary depending on what could not be found.

```
{
    "data":
 {
        "code": 404,
        "message": <error
string>
    }
}
```

500

```
{
    "data":
 {
        "code": 500,
        "message": "An unknown
error occurred."
    }
}
```