



MSE Location APIs

The location APIs contains the following four categories:

- [Maps API, page 3-1](#)
- [Real Time Location APIs, page 3-2](#)
- [Location History APIs, page 3-3](#)
- [Notification Event APIs, page 3-5](#)

Maps API

The Maps API provides details about the Map data. This provides summary of all your managed systems on campuses, buildings, outdoor areas, floors, zones, access points, dimensions, map images, GPS markers, obstacles, location filter regions, and excitors.

All Map API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

The following data formats are supported for Maps API. The response is determined based on the Accept HTTP header.

- XML—Use Accept: `application/xml`
- JSON—Use Accept: `application/jsonsupports`

This section contains [Resources, page 3-1](#).

Resources

- `/api/contextaware/v1/maps`
See the [maps](#) for more details.
- `/api/contextaware/v1/maps/count`
See the [mapscount](#) for more details.
- `/api/contextaware/v1/maps/info/{campusName}/{buildingName}/{floorName}`
See the [mapsinfo](#) for more details
- `/api/contextaware/v1/maps/image/{campusName}/{buildingName}/{floorName}`
See the [mapsimage](#) for more details
- `/api/contextaware/v1/maps/imagesource/{imageName}`

See the [mapsimagesource](#) for more details

Real Time Location APIs

The real time location APIs provides access to the location of devices that are currently being tracked by the MSE.

All Map API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

The following data formats are supported for real time location API. The response is determined based on the Accept HTTP header.

- XML—Use `Accept: application/xml`
- JSON—Use `Accept: application/jsonsupport`s

This section contains the following topics:

- [Paging, page 3-2](#)
- [Sorting, page 3-2](#)
- [Resources, page 3-2](#)

Paging

Pagination in the resources is done by specifying query Params “page” and “pageSize”. The default page size is 5000. When there is more than one page, the `nextResourceURI` will specify the URI to the next resource.

Sorting

Sorting requests in the API requests is achieved by specifying queryParams “sortBy”. By default, ascending sort order is assumed. To specify the sort order use format “sort parameter: order”. For example, to request location results by `lastLocatedTime` in descending order specify queryParams “sortBy=lastLocatedTime:desc”.

For multiple sort conditions, use multiple “sortBy” queryParams.

Resources

Real Time Location APIs for Client

- `/api/contextaware/v1/location/clients/{id}`
See the [locationclients](#) for more details.
- `/api/contextaware/v1/location/clients/count{?}`
See the [locationclientscount](#) for more details.

Real Time Location APIs for Tag

- `/api/contextaware/v1/location/tags{?}`
See the [locationtags](#) for more details.
- `/api/contextaware/v1/location/tags/count{?}`
See the [locationtagscount](#) for more details.

Real Time Location APIs for Rogue AP

- `/api/contextaware/v1/location/rogueaps{?}`
See the [locationrogueaps](#) for more details.
- `/api/contextaware/v1/location/rogueaps/count{?}`
See the [locationrogueapscount](#) for more details.

Real Time Location APIs for Rogue Client

- `/api/contextaware/v1/location/rogueclients/{?}`
See the [locationrogueclients](#) for more details.
- `/api/contextaware/v1/location/rogueclients/count{?}`
See the [locationrogueclientscount](#) for more details.

Real Time Location APIs for Interferer

- `/api/contextaware/v1/location/interferers/{id}`
See the [locationinterferers](#) for more details.
- `/api/contextaware/v1/location/interferers/count{?}`
See the [locationinterfererscount](#) for more details.

Location History APIs

The location history APIs provides access to the historical location of devices tracked by the MSE.

All location history API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

The following data formats are supported for location history API. The response is determined based on the Accept HTTP header.

- XML—Use `Accept: application/xml`
- JSON—Use `Accept: application/json`

This section contains the following topics:

- [Paging, page 3-4](#)
- [Sorting, page 3-4](#)
- [Resources, page 3-4](#)

Paging

Pagination in the resources is done by specifying queryParams “page” and “pageSize”. The default page size is 5000. When there is more than one page, the nextResourceURI will specify the URI to the next resource.

Sorting

Sorting requests in the API requests is achieved by specifying queryParams “sortBy”. By default, ascending sort order is assumed. To specify the sort order use format “sort parameter: order” For example, to request location results by lastLocatedTime in descending order specify queryParams “sortBy=lastLocatedTime:desc” For multiple sort conditions, use multiple “sortBy” queryParams.

Resources

Location History APIs for Client

- `/api/contextaware/v1/location/history/clients/{?}`
See the [locationhistoryclients](#) for more details.
- `/api/contextaware/v1/location/history/clients/{?}count{?}`
See the [locationhistoryclientscount](#) for more details.

Location History APIs for Tag

- `/api/contextaware/v1/location/history/tags/{?}`
See the [locationhistorytags](#) for more details.
- `/api/contextaware/v1/location/history/tags/count{?}`
See the [locationhistorytagscount](#) for more details.

Location History APIs for Rogue AP

- `/api/contextaware/v1/location/history/rogueaps{?}`
See the [locationhistoryrogueaps](#) for more details.
- `/api/contextaware/v1/location/history/rogueaps/{id}/count{?}`
See the [locationhistoryrogueapscount](#) for more details.

Location History APIs for Rogue Client

- `/api/contextaware/v1/location/history/rogueclients{?}`
See the [locationhistoryrogueclients](#) for more details.
- `/api/contextaware/v1/location/history/rogueclients/count{?}`
- See the [locationhistoryrogueclientscount](#) for more details

Location History APIs for Interferer

- `/api/contextaware/v1/location/history/interferers{?}`
See the [locationhistoryinterferers](#) for more details
- `/api/contextaware/v1/location/history/interferers/{id}/count{?}`
See the [locationhistoryinterfererscount](#) for more details.

Notification Event APIs

The Notification Event APIs allows an application to register for Push Notifications from the MSE.

The following Notification events are supported and these notifications can be configured for a specific MAC address or a list of MAC addresses with a wildcard for a specific device type.

1. **Absence Event**—Triggers a notification if the subscribed devices are absent for the configured absent interval.
2. **Battery Event**—Triggers a notification if the battery level of a RFID tag is at the configured level.
3. **Exciter Event**—Triggers a notification if the RFID tag is detected by a configured exciter.
4. **Containment Event**—Triggers a notification if a device is inside or outside of a configured zone.
5. **Emergency Event**—Triggers a notification if the emergency or panic button on the RFID tag is pressed.
6. **MapInfoChange Event**—Triggers a notification if the map configuration on the MSE changes.
7. **Movement Event**—Triggers a notification if a device moves more than a configured distance from its old location or a configured marker.
8. **Presence Event**—Triggers a notification when a device is first detected by the MSE.
9. **Streaming Notifications**—Triggers a stream of notification for every location calculation.
10. **Association Event**—Triggers a notification when a device associates to the Wi-Fi network. The subscription should have type “AssociationEventTrigger”, “association”: true/false.

All notification event API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

The following data formats are supported for notification APIs. The response is determined based on the Accept HTTP header. The transport can be either HTTPS or TCP.

- XML—Use `Accept: application/xml`
- JSON—Use `Accept: application/jsonsupports`
- Protocol Buffers

Resources

- `/api/contextaware/v1/notifications`
See the [notifications](#) for more details.

