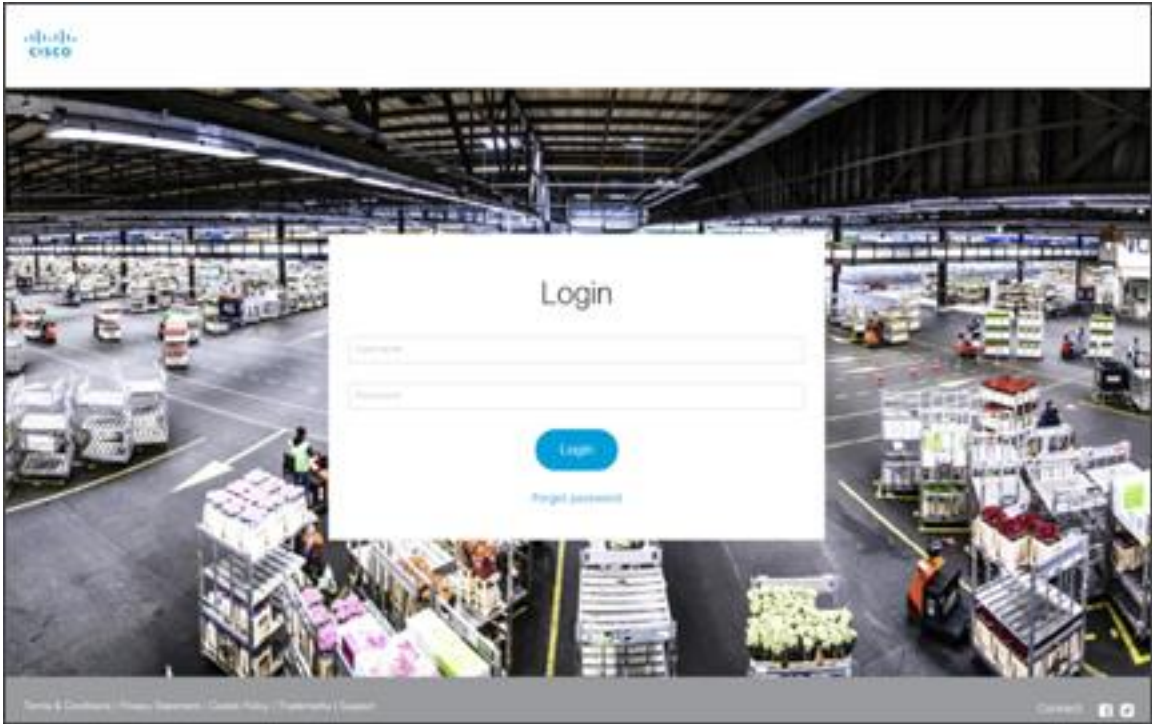


Code Samples for Cisco CMX Location SDK

Last Modified: August 20, 2018



Contents

- Incorporating Mobile SDK Location Elements into Your App 3**
 - Testing in Simulation Mode 3
 - Setting the SDK Token 3
 - Configuring Continuous Location Update 4
 - Code Sample: Using Callbacks to Derive Current Location 4
 - Code Sample: Derive Location without Callbacks 4
 - Incorporating Map Images and Metadata 5
 - Incorporating Points of Interest 5
 - Calculating Routes to a Location 5
 - Code Sample: Starting a Route 6
 - Code Sample: Stopping a Route 6

Incorporating Mobile SDK Location Elements into Your App

When you have created the location elements you need in Cisco CMX Location Experience Manager (LXM), you are ready to incorporate the tokens, maps, points of interest, paths, and metadata into your app.

- Refer to *Getting Started with Cisco CMX Location Experience Manager* to open an LXM account and create your authentication tokens.
- Refer to *Creating SDK Location Elements with Cisco CMX* to configure your maps.

Follow these examples, and be careful to replace any generic information.

Testing in Simulation Mode

The SDK can run in simulation mode. This allows you to test it from a developer's system, without requiring any Cisco infrastructure.

Note: Simulation mode requires an SDK token, but it does not have to be valid.

1. To enable simulation mode, change the **setSimulateClient** method to **true**:

```
CmxClient.setSimulateClient(true)
```

2. Set an SDK token. See *Setting the SDK Token*, below.
3. Invoke the **startLocationUpdates** method. See *Configuring Continuous Location Update*.

Once client location update is enabled, the device simulates a user walking in a square pattern around a floor. The map displays points of interest (POI) at each corner of the map. A notification displays as the user walks past each POI.

Setting the SDK Token

Set the SDK token within the application by invoking this method:

```
import CmxMobileSdk
CmxClient.setTenantToken ("sdk_token_ID")
```

For example:

```
import CmxMobileSdk
CmxClient.setTenantToken
("eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJ0Zm5JZCI6IjE0Ny1smlhdC
I6MTUzMzY4ODgyNn0")
```

Configuring Continuous Location Update

We recommend using a callback to manage location updates. Invoke these methods to control continuous location updates:

- `CmxClient.startLocationUpdates` – Enables continuous location updates.
- `CmxClient.stopLocationUpdates` – Disables continuous location updates.

To enable location without a callback, use this command:

- `CmxClient.getCurrentLocation` – Derives the current location without a callback.

Code Sample: Using Callbacks to Derive Current Location

Here is an example of continuous location updates that include the callback code:

```
class LocationMeasurementsViewController: UITableViewController {
    override func viewDidLoad() {
        super.viewDidLoad()

        CmxClient.setTenantToken("eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0ZW5JRCI6InRlbnl")

        CmxClient.startLocationUpdates(onUpdate: self.clientLocationUpdate)
    }

    func clientLocationUpdate(_ cmxLocation: CmxDeviceLocation, _ error: Error?) {
        if (error != nil) {
            print("Error: \(error?.localizedDescription ?? "Unkown")")
        }
        print("Client location update X: \(cmxLocation.floorCoordinate.x)")
        print("Client location update Y: \(cmxLocation.floorCoordinate.y)")
    }
}
```

Code Sample: Derive Location without Callbacks

Here is an example with no callback code:

```
class LocationMeasurementsViewController: UITableViewController {
    override func viewDidLoad() {
        super.viewDidLoad()

        CmxClient.setTenantToken("eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0ZW5JRCI6InRlbnl")

        CmxClient.startLocationUpdates()
    }

    func getLocation() {
        let cmxLocation: CmxDeviceLocation = CmxClient.getCurrentLocation()
        print("Client location update X: \(cmxLocation.floorCoordinate.x)")
        print("Client location update Y: \(cmxLocation.floorCoordinate.y)")
    }
}
```

Note: You should still call `startLocationUpdates`, even if you want location information on demand, without callbacks.

Incorporating Map Images and Metadata

You can incorporate the map images and metadata you imported to LXM. Follow this example to include metadata and an image to a floor map. See also *Importing a Site Map* from *Creating SDK Location Elements with Cisco CMX Location Experience Manager*.

```
CmxClient.getMap(onCompletion: { (mapImage, mapInfo, error) in
  if (error != nil) {
    print("Error: \(${error?.localizedDescription ?? "Unkown"}")")
  } else {
    print("Map: \(${mapInfo.floorId}")
    displayImageMethod(mapImage)
  }
}
```

Incorporating Points of Interest

You can incorporate the POI metadata you created in LXM. Follow this example to include metadata for a POI. See also *Adding Points of Interest* from *Creating SDK Location Elements with Cisco CMX Location Experience Manager*.

```
CmxClient.getPointOfInterestsInfo(onCompletion: { (poiList, error) in
  if (error != nil) {
    print("Error: \(${error?.localizedDescription ?? "Unkown"}")")
  } else {
    for poi in poisInfo! {
      print("Point Of Interest ID: \(${poi.id}")
      print("Point Of Interest Name: \(${poi.name}")
    }
  }
}
```

Calculating Routes to a Location

You can calculate routes to a specified location. See also *Adding Paths* from *Creating SDK Location Elements with Cisco CMX Location Experience Manager*.

Code Sample: Starting a Route

Follow this example to create a route from the current device location to a destination location.

```
class MapViewModel: CmxRouteController {
    func getRoute(forLocation location: CmxPointOfInterest) {
        // Calculate new route
        CmxClient.startRoute(destination: location.floorCoordinate, routeController:
self)
    }

    func onRouteCalculated(_ waypoints: [CmxFloorCoordinate]?, _ error: Error?) {
        if let waypoints = waypoints {
            // Draw line
            drawAllLine(onPath: &linePath, forWayPoints: waypoints)
        }
    }

    func onShouldReroute() {
        print("Should reroute")
    }

    func onDestinationReached() {
        print("Destination Reached")
    }
}
```

Code Sample: Stopping a Route

A route can be stopped at any time. For example, you can stop the route when the device is off. Follow this example to stop a route.

```
CmxClient.stopRoute()
```