SPREO iOS SDK/API DOCUMENTATION

DOCUMENT VERSION:	10.2
iOS SDK VERSION:	2.0.32
iOS BUNDLE VERSION	2.0.32
DATE REVISED	June 2016

Table of Contents

1. Document Overview

This document describes the Indoor Navigation System SDK including its API, delegation protocols, and fundamental components.

2. System Overview

The SDK uses the following iOS device components and frameworks to achieve the best indoor and outdoor location within a venue or campus:
- Core Bluetooth.
- Core Motion.
- Core Location.

The SDK effectively uses these components to ensure power conservation and resource management.
The system is multi-threaded using several threads for several tasks. All calls between the SDK and the applications are done within the main thread.

3. General Configurations and Settings

This section describes the initialization process of the API.

4. Methods

4.1. Get the SDK Version String

The method returns the SDK Version

+ (NSString *)getVersionString;

4.2. Registration and Association

The method registers to get call backs from services and enables the application to use the corresponding location, navigation, and geofence services.
+ (void)setApiKey:(NSString *)anApiKey value:(NSString *)aValue error:(IDError **)anError

4.3. Update client data and initialize.

The method checks for data updates on the corresponding web server and downloads the data. The data can be:
- Campus data updates.
- Facility data updates.

4.3.1. Check for Data and updates

The method initializes the data updates within the specified server and data module initialization. The data download is done on a back thread and the status is updated using the delegate protocol.
+ (void)checkForDataUpdatesAndInitialiseWithDelegate:(id <IDDataUpdateDelegate>)delegate

4.3.2. Delegate Protocol

The protocol provides feedback on the data update phase process.
@protocolIDDataUpdateDelegate<NSObject>

4.3.3. Notify update and initialization state

The method updates on any changes during the update and initialization processes of the data model.
- *(void)dataUpdateStatus:(IDDataUpdateStatus)status*

4.4. Languages

4.4.1. Supported language

The method returns an array of strings for supported languages @[@"en", @"es", @"ch", etc…]

+ (NSArray<NSString *> *)supportedLanguages;

4.4.2. Get current language

The method returns the SDK current language.

+ (NSString*)getCurrentLanguage;

4.4.3. Set current language

Call this method in order to set the SDK current language. The selected language is important and may affect (for example) the POI data and the map label, which are loaded according to the selected language.
The method Returns an answer whether the SDK supports a language.

+ (BOOL)setCurrentLanguage:(NSString)aLanguage;*

4.5. Monitor campus region

The method sets or removes the region monitoring for the project campuses.
When this method is called with a YES value, the system starts to monitor regions for all project campuses. As a result, the user will be notified upon entering the campus region.
Calling this method with a NO vale will stop the monitoring and will remove all the monitored campuses.

+ (void)monitorCampusesRegion:(BOOL)mode

4.6. Set user identifier

The method sets a user ID for the system. That user ID will be used for all statistics and server interactions. If the method is not called or a nil value is received, the system generates a random id that will represent the user.

*+ (void)setUserID:(NSString *)anId*

4.7. Set logger mode

The method sets the logger mode. A YES call will engage the logger; a NO call will turn it off. The logger enables further debugging abilities and analysis in case of issues.

+ (void)setLoggerWithMode:(BOOL)mode

4.8. User reaction to local notification

The user can choose to respond to the local notification. In such case, the application needs to update the IndoorSDK of that event.

Call the didReceiveLocalNotification method from the implementation of the

- (void)application:UIApplication *)app didReceiveLocalNotification(UILocalNotification *)notification

in the AppDelegate class.

4.8.1. User reaction to local notification text

*+ (void)setLocalNotificationText:(NSSting *)aText*

4.9. Get Campuses IDs

The method returns an array of all campuses IDs.

+ (NSArray <NSString *> *)getCampusIDs;

4.10. Get info For Campus With ID

The method returns a NSDictionary with details about the campus:

+ (NSDictionary *)getInfoForCampusWithID:(NSString *)campusId

4.11. Receive all facilities IDs for a given campus ID

The method returns an array of facility IDs associated with the campus ID

+ (NSArray <NSString *> *)getFacilityIDsForCampusID:(NSString *)aCampusId

4.12. Get info for facility with ID at campus with ID

The method returns an NSDirectory with details about each facility in the campus

+ (NSDictionary *)getInfoForFacilityWithID:(NSString *)facilityId

atCmpusWithID:(NSString*)campusId

4.13. Get campus info

The method returns an NSDictionary with details about the campus.

+ (NSDictionary *)getInfoForCampusWithID:(NSString *)campusId

4.14. Get facility info

The method returns an NSDictionary with detail about the facility in the campus

+ (NSDictionary *)getInfoForFacilityWithID:(NSString *)facilityId

atCmpusWithID:(NSString*)campusId

4.15. Get floor info

The method returns an NSDictionary with detais about the floor in facility in the campus

+ (NSDictionary *)getInfoForFloorID:(NSInteger)floorId

inFacilityWithID:(NSString *)facilityId

atCmpusWithID:(NSString *)campusId

4.16. Get POI/POIs

The method returns an array of IDPoi objects for each suitable found POI.

+ (NSArray<IDPoi *> *)getPOIsWithID:(NSString *)aPoiId;

4.17. Get POIs sorted alphabetically

The method returns an array of the IDPoi objects, which is sorted alphabetically for each POI found.

(NSArray<IDPoi *> *)sortedPOIsAlphabeticallyWithPathID:(NSString *)aPathID;

4.18. Get POIs sorted by distance from location

The method returns an array of the IDPoi object, which is sorted by distance for each suitable POI found.

(NSArray<IDPoi *> *)sortedPOIsDistantlyWithPathID:(NSString*)aPathID

fromLocation:(IDLocation*)aLocation;

4.19. Get POIs Categories List

The method associates the poi's categories in array with the object path set.

+ (NSArray <NSString *> *)getPOIsCategoriesListWithPathID:(NSString *)aPathID;

4.20. Get POI/POIs with category

The method returns an array of IDPoi objects, which holds the category string for the specified path ID.

-(NSArray<IDPoi *> *)getPOIsWithCategory:(NSString *)aCategory atPathID:(NSString *)aPathID;

4.21. Get POIs sorted alphabetically by categories

The method returns an array of the IDPoi objects, which is sorted alphabetically for each POI found, including the categories strings associated with the specified path ID.

(NSArray<IDPoi *> *)sortedPOIsAlphabeticallyWithCategories:(NSArray *)aCategories

atPathID:(NSString *)aPathID;

4.22. Get POIs sorted by distance from location with categories

The method returns an array of IDPoi objects sorted by distance for each POI found, including the categories strings associated with the specified path ID.

+ (NSArray<IDPoi *> *)sortedPOIsDistantlyWithCategories:(NSArray *)aCategories

atPathID:(NSString*)aPathID

fromLocation:(IDLocation*)aLocation;

4.23. Request Itinerary POI's Array

The method will add switch floors POIs if TRUE, and parking location as aprking POI if TRUE

```
+ (NSArray <IDPoi *> *)itineraryPoisList:(NSArray <IDPoi *> *)aPoisList
                        addSwitchFloorsPois:(BOOL)addSwitchFloors
                        addParkingPoi:(BOOL)addParking;
```

4.24. Request Ordered Itinerary POI's Array

The method requests a list of POIs according to the shortest path (route) algorithm.

```
+ (NSArray <IDPoi *> *)orderItineraryPoisList:(NSArray <IDPoi *> *)aPoisList
                        addSwitchFloorsPois:(BOOL)addSwitchFloors
                        addParkingPoi:(BOOL)addParking;
```

4.25. Store Parking Location

The method set the location as parking location.

```
+ (void)setParkingLocation:(IDLocation *)aLocation;
```

+

4.25. Request Parking Location

The method returns parking location. the returned location value if exists.

```
+ (IDLocation *)getParkingLocation;
```

+

4.26. Remove Parking Location

The method removes the parking location.

```
+ (void)removeParkingLocation;
```

5. Map View Containers

This section describes the map view containers, map shared delegation protocol.

The map view should be used as child view and child view controller, adding it to a container view and container controller. there are two Containers Maps Types:

- **Container A Map - IDDualMapViewController** - Based on Google Maps for indoor and outdoor locations.
- **Container B Map IDMapViewController** - container that contains two containers:

1) Container with View that Based on Apple Maps For outdoor location position.

2) Container with View that Based on Scroll View for indoor location position.
This Container has more APIs than **Container A Map.**
Both Containers have a shared protocol and Implements IDMapViewProtocol.

5.1. IDMapViewProtocol

The shared protocol IDMapViewProtocol defined as the shared API Methods for the Map Containers. The following methods enable adjustments on the display and content of the map. The methods are sent directly to the *IDMapViewProtocol* header.

5.1.1. Map display adjustments

5.1.1.1. Set map type

Sets the map type: standard map, satellite view or hybrid.

- (void)setMapType:(IDMapType)mapType

5.1.1.2. Set map show layer

Shows or hides different layers of the map.

- (void)setMapShowLayer:(IDMapLayerType)layerType mode:(BOOL)mode

5.1.1.3. Show All POIs

Call this method to in order to show all POIs

- (void)showAllPois

5.1.1.4. Hide All POIs

Call this method to in order to hide all POIs

- (void)hideAllPois

5.1.1.5. Show All Labels

Call this method to in order to show all POIs

- (void)showAllPois

5.1.1.6. Hide All Label

Call this method to in order to hide all POIs

- (void)hideAllPois

5.1.1.7. Set map filter POIs

Filters the displayed POIs by category. If the method is not called or is called with a nil array parameter, all POIs categories will be displayed.

- (void)setVisiblePOIsWithCategories:(NSArray<NSString *> *)categories

5.1.1.8. Set map rotation mode

Sets the map rotation mode: static, compass or orientation.

- (void)setMapRotationMode:(IDMapRotationType)rotationType

5.1.1.9. Set time interval for auto follow

Sets the time length from the last time the user moved the map until the map returns to the user's current location view.

- (void)setUserAutoFollowTimeInterval:(NSTimeInterval)duration

5.1.1.10. Set the in region radius for a POI.

Sets the radius for the displayed POIs

- (void)setPoiRegionRadius:(NSInteger)radius

5.1.2. Map content adjustments

5.1.2.1. Show location

Shows the given location

- (void)presentLocation:(IDLocation *)aLocation

5.1.2.2. Show Poi

Shows the given Poi on Map, and present poi bubble automatically

- (void)presentPoiOnMapWithPoi:(IDPoi *)aPoi;

5.1.2.3. Show Poi Bubble

Shows the bubble of the given Poi

- (void)showBubbleForPoi:(IDPoi *)aPoi;

5.1.2.4. Hide Poi Bubble

Hides the bubble of the given Poi

- (void)hideBubbleForPoi:(IDPoi *)aPoi;

5.1.2.5. Centre Facility Map

Centre facility map of given facility Id and campus Id

- (void)centerFacilityMapWithFacilityId:(NSString*)aFacilityId atCampusId:(NSString*)aCampusId;

5.1.2.6. Centre Campus Map

Centre campus map of given campus Id

- (void)centerCampusMapWithCampusId:(NSString*)aCampusId;

5.1.2.7. Show floor map at facility with Id

Call the following method in order to show the corresponding floor of the facility.

- (void)showFloorWithID:(NSInteger)aFloorID atFacilityWithId:(NSString*)aFacilityId;

5.1.2.8. Show my position

Call this method to center the map on the user's location and show the user's icon.

- (void)showMyPosition

5.1.2.9. Zoom in

Call on this method to increase the map's zoom level by one notch, until it reaches the maximum zoo m level.

- (BOOL)zoomIn

5.1.2.10. Zoom out

Call on this method to decrease the map's the zoom level by one notch, until it reaches the minimal zoom level.

- (BOOL)zoomOut

5.1.2.11. Set Map Zoom Level

Call on this method to custom the map's the zoom level

- (BOOL)setMapZoomLevel:(CGFloat)aMapZoomLevel;

5.1.2.12. Map reload data

Call on this method to reload the map data and to refresh the map view.

- (void)mapReload

5.1.2.13. Current presented floor identifier

The method returns the integer identifier for the current floor.

- (NSInteger)currentPresentedFloorId

5.2. Container A Map - IDDualMapViewController

The method returns the map view controller. The map view controller should be set on child view on any view controller in use.

+ (IDDualMapViewController *)getDualMapViewController

5.2.1. IDMapViewController Methods

5.2.1.1. Provide Google Maps API Key

Call This Method in order to set Google Maps App API Key String

- (BOOL)provideGoogleMapsAPIKey:(NSString*)apiKey;

5.2.2. IDDualMapViewController Map Settings

5.2.2.1. Indoor picker view

Enables (default) or disables the indoor floor picker. If enabled, it is only

visible when the view is focused on a building with indoor floor data.

5.2.2.2. My Location button

Enables or disables the My Location button. This is a button visible on the

map that, when tapped by users, will center the map on the current user location.

5.2.2.2. Compass button

Enables or disables the compass. The compass is an icon on the map that indicates the direction of nor th on the map. If enabled, it is only shown when the camera is rotated away from its default.

5.2.3. IDDualMapViewController Delegation Protocol IDDual-MapViewControllerDelegate

The map components can also be customized by responding to one or more of the map delegation met hods IDMapViewControllerDelegate.

The map view delegate protocol will receive notification of map changes.

- @protocol IDDualMapViewControllerDelegate<NSObject>

5.2.3.1. Custom the user annotation icon

Show custom icon image for user.

- (UIImage *)mapIconForUserAnnotaion;

5.2.3.2. Custom the pois icons

Show custom icon image for user.

- (UIImage *)mapIconForPoi;

5.2.3.3. Custom the labels icons and visibility

Show custom icon image for Labels annotations on map

- (UILabel*)mapLabelIconForLabel:(IDLabel *)aLabel;

Show/ hide all labels:

- (void)showAllLabels;
- (void) hideAllLabels;

Show/ hide label by identifier:

-(BOOL)setLabelVisibleByID:(NSString*)labelId showLabel(BOOL)visible

5.2.3.4. Custom the Pois Bubble

Called when a marker is about to become selected, and provides an optional, custom info window to use for that marker if this method returns a UIView.

```
- (UIView *)mapViewForCalloutOfPoi:(IDPoi *)aPoi;
```

5.2.3.5. Custom the Parking annotation

Show custom view for parking annotation on map.

```
- (UIImage *)mapIconForParkingAnnotaion;
```

5.2.3.6. Custom the Navigation Route Colour

Show custom colour for navigation route GIS on map.

```
- (UIColor *)mapColorForRoute;
```

5.2.3.7. My location button did tapped

This method is called when the my location button view did tapped.

```
- (void)mapDidTapMyLocationButton;
```

5.2.3.8. POI icon Tapped

This method is called when the user touches the POI icon.

```
- (void)mapDidTapPOI:(IDPoi *)aPoi;
```

5.2.3.9. POI callout was tapped

This method is called when the user taps the call out of the POI icon.

```
- (void)mapDidTapCalloutOfPOI:(IDPoi *)aPoi
```

5.2.3.10. User icon did tapped

This method is called when a map user annotation view did tapped.

```
- (void)mapDidTapUserAnnotaionIcon:(UIImage *)anAnnotationView;
```

5.2.3.11. Parking icon icon did tapped

This method is called when a map annotation view did tapped.

- (void)mapDidTapParkingAnnotaionIcon:(UIImage *)anAnnotationView;

5.2.3.12. label icon did tapped

This method is called when a map Label view did tapped

- (void)mapDidTapLabelAnnotaionIcon:(UIImage *)aLabelIcon forLabel:(IDLabel*)aLabel;

5.2.3.13. region focus did changed

This method is called when closest facility region contains the map centre point.

The location Id will be the 'facilityId'.

- (void)mapDidChangedFocusToLocationWithID:(NSString *)aLocationId;

5.2.3.14. floor id did changed

This method is called when the map did change floor map indication at facilityId

- (void)mapDidChangeFloorId:(NSInteger)aFloorId atFacilityId:(NSString*)aFacilityId;

5.2.3.15. zoom level did changed

This method is called when the mapView did zoomed to zoom level.

- (void)mapDidChangeZoomLevel:(CGFloat)aZoomLevel;

5.2.3.16. user follow me mode did changed

This method is called when the map follow user change.

- (void)mapFollowUser:(BOOL)mode;

5.3. Container B Map - IDMapViewController

5.3.1. IDMapViewController Methods

5.3.1.1. Get the view for a POI

The method returns the corresponding view for a specific POI.

- (UIView *)viewForPoi:(IDPoi *)aPoi

5.3.1.2. Set the Max POI popups/ bubbles number.

Call this method to set the max number of POS popup on map, this number limits the number of poi popups on map in vicinity to user location.

- (void)setMaxPoiPopupsNumber:(NSInteger)number;

5.3.1.3. Close all POIs bubbles.

The method closes all open POIs bubbles.

5.3.1.4. Add overlays to outdoor map

The method will overlay an image (map) on the outdoor map according to what will be positioned between the Top Left and Bottom Right coordinates.

(void)addCustomOverlayImage:atTopLeftCoordinates:andBottomRightCoordinates:

5.3.1.5. Draw trip overview

The method will add an overview of the routes with circles for each POI index onto the map. “Arrived” (visited) POIs will draw a circle to the POI location.(The circles and route colors can be customized by Implementing the

IDMapViewControllerDelegate methods:

1) mapColorForTripOverviewRoute

2) mapColorForTripOverviewCircle

3) mapColorForTripOverviewArrivedCircle)

- (void)setMapDrawTripOverviewWithPois:(NSArray*)aPois

arrivedPois:(NSArray*)anArraivedPois

drawSwitchFloorsCircles:(BOOL)addSwitchFloorsCircles

drawEntrancesCircles:(BOOL)addEntrancesCircles;

5.3.1.6. Show User Location With Bubble

Call the method to in order to customize the user location bubble when the user is outdoor

- (void)presentUserLocationWithBubble:(UIView*)aView;

in order to remove the Bubble View call presentUserLocationWithBubble: method with nil parameter.

5.3.1.7. Remove trip overview

The method will remove the trip overview.

- (void)setMapRemoveTripOverview

5.3.1.8. Set map lock to region

Sets the map to lock with in a region. Use that method together with the map's present Location meth od.

- (void)setMapLockToRegionPath:(NSString *)aPath;

5.3.1.9. Title for floor identifier

The method returns the title string of the floor ID

- (NSString *)getFloorTitleForFloorID:(NSInteger)aFloorID

5.3.2. IDMapViewController Delegation Protocol

IDMapView ControllerDelegate

The following method sets the delegation IDMapViewControllerDelegate for the IDMapViewController map view controller.

5.3.2.1. View for user

Call this method to show the user's location on the map. Respond to that delegate method for customizing the user icon.

- (UIView *)mapViewForUser

5.3.2.2. View for User Location with Bubble

Call this method in order to customize user location bubble

- (void)presentUserLocationWithBubble

5.3.2.3. View for direction pointing

Call this method to show user's direction (outdoor pointing navigation). Respond to that delegate metho d for customizing the way-pointing arrow.

- (UIView *)mapViewForPointingArrow

5.3.2.4. View for campus

Call this method for campus view on the map. The delegate can return a custom view of the campus.

- (UIView *)mapViewForCampusWithCampusID:(NSString *)aCampusId

5.3.2.5. View for campus graphics

Call this method to show a campus region on the map. The delegate can return a custom view for the campus region.

- (UIImage *)mapImageForCampusGraphicsWithCampusID:(NSString *)aCampusId

5.3.2.6. View for facility

Call this map to show the facility on the map. The delegate can return a facility view for the facility.

- (UIView *)mapViewForFacilityWithId:(NSString *)aFacilityId

atCampusWithId:(NSString*)aCampusId

5.3.2.7. View for label

Call this method to show a label on the map. The delegate can return a label view

for the label.

- (UIView *)mapViewForLabel:(NSString *)aLabel

5.3.2.8. View for POI

Call this method to show a POI on the map. The delegate can return a custom view for the POI.

- (UIView *)mapViewForPoi:(IDPoi *)aPoi

5.3.2.9. View for Parking Annotation

Call this method to customize the parking annotation view

- (UIView*) mapViewForParkingAnnotaion

5.3.2.10. POI selected

This method is called when the user touches the POI icon.

- (void)mapDidSelectPOI:(IDPoi *)aPoi

5.3.2.11. POI callout was tapped

This method is called when the user taps the call out of the POI icon.

- (void)mapDidTapCalloutOfPOI:(IDPoi *)aPoi

5.3.2.12. POI callout action was tapped

This method is called when the user taps the callout action button of the POI icon.

- (void)mapDidTapCalloutActionOfPOI:(IDPoi *)aPoi

5.3.2.13. User Annotation did tapped

implement this method to use your User annotation view

(void)mapDidSelectedUserAnnotaionView:(UIView *)anAnnotationView.

only when outdoor.

5.3.2.14. Parking Annotation did tapped

Implement this method to use your parking annotation view

- (void)mapDidSelectedParkingAnnotaionView:(UIView *)anAnnotationView

5.3.2.15. Map will change floor

This method is called before the map view is changing floor

- (void)mapWillChangeFloor:(NSInteger)aFloorID

5.3.2.16. Map has changed floor

This method is called after map has changed the floor

- (void)mapDidChangeFloor:(NSInteger)aFloorID

5.3.2.17. Map will be swapped

This method is called when the map is about to switch from indoor to outdoor map views

- (void)mapWillSwapMapTo:(IDMapViewId)mapId

5.3.2.18. Map Swapped

This method is called when the map switches from indoor to outdoor map views

- (void)mapDidSwapMapTo:(IDMapViewId)mapId

5.3.2.19. Map arrow image for route

The method returns an arrow image to indicate the navigation route view.

- (UIImage*)mapArrowImageForRoute

5.3.2.20. Map Color for trip overview route

The method returns UIColor for trip overview route.

- (UIColor*)mapColorForTripOverviewRoute

5.3.2.21. Map Color for trip overview circle

The method returns UIColor for trip circle Index titled view.

- (UIColor*) mapColorForTripOverviewCircle

5.3.2.22. Map Color for trip overview “Arrived Circle”

The method returns UIColor for arrived (visited) trip overview circle.

- (UIColor*) mapColorForTripOverviewArrivedCircle

5.4. POI Custom View For Callout View Protocol IDCallouts Protocol

The IDPoiViewProtocol protocol provides feedback on the poi, with user location and region from poi on map. With bubble view (callout) presentation, A custom view that adopts the protocol will receive the events when users enter or exit the region of the POI.

5.4.1. Bubbles Layer View

in order to custom UIView for Poi Bubble View, you must implement bubblesLayerView property in you custom view, the Indoor Kit SDK uses this property in order to draw your

custom view in to the map at the right position.

5.4.2. Allow The Protocol Feedbacks

Return YES in order to allow The Indoor Kit to call IDCalloutsProtocol feedbacks methods.

- (BOOL)shouldResponseToLocationRegioning;

5.4.3. Custom Bubble View for Poi

Custom bubble view in order to allow The Indoor Kit to change bubble scale or bubble rotation if need ed.

- (UIView*)bubbleView;

5.4.4. Present POI Bubble View

This method is called when user position enters region of Poi

- (void)presentBubbleViewWithMode:(IDCalloutMode)aMode;

5.4.5. Dismiss POI Bubble View

This method is called when user position exits region of Poi

- (void)dismissBubbleView;

5.5. POI in region protocol - *IDPoiViewRegioningProtocol*

It is possible to customize the POI view that will be presented. In order to enhance the user experience, a POI view can adopt the IDPoiViewRegioningProtocol. A custom view that adopts the protocol will receive the events when users enter or exit the region of the POI.
That region radius can be globally set by the setPoiRegionRadius method.

@protocolIDPoiViewRegioning<NSObject>

5.5.1. Entered POI region

The method is called with the provided view when the user enters the region of POI

- (void)didEnterRegionOfPOI:(IDPoi *)aPoi

5.5.2. Exited the POI region

The method is called with the provided view when the user exits the region of the POI

- (void)didExitRegionOfPOI:(IDPoi *)aPoi

6. Positioning & Navigation:

This section describes the positioning and navigation. The SDK provides full support for positioning and navigation within an outdoor campus, region and facility.

6.1. Methods

6.1.1. Start user positioning updates

The method starts location updates and geofence awareness.

The method returns YES if starting the location tracking was successful. Otherwise it returns

NO.

+ (BOOL)startUserLocationTrack

or

+ (BOOL)startUserLocationTrackingWithDelegate:(id<IDNavigationDelegate>)aDelegate

6.1.2. Set the navigation delegate.

The method sets the IDNavigationDelegate protocol delegate.

+ (void)setNavigationDelegate:(id<IDNavigationDelegate>)aDelegate

6.1.3. Is user position tracked?

The method returns YES when the user position is tracked. Otherwise it returns NO.

+ (BOOL)isUserLocationTracked

6.1.4. Stop user positioning tracking.

The method stops the user position tracking and updates.

+ (void)stopUserLocationTrack

6.1.5. Simulate User Location.

The method simulates user location either indoor or outdoor.

+ (void)setUserLocation:(IDUserLocation*)aUserLocation

When “aUserLocation.facilityId” is not nil the method will simulate user location indoor. In order to simulate user location outdoor, “aUserLocation.facilityId” property MUST BE NIL.

In order to stop the user simulated location call on this method with nil (restore to default location).

6.1.6. Reset user positioning tracking.

The method resets the user position tracking and positioning updates.

+ (void)resetUserLocationTrack

~~6.1.7. Notify the SDK about background mode.~~

The method will notify the SDK that the app is moving to background, and if you want the RF to turn off or not.

+ (void)moveToBackgroundAndContinueScanning:(BOOL)keepScan;

~~6.1.8. Notify SDK about foreground mode.~~

The method resets the user position tracking and positioning updates.

+ (void) moveToForeground

6.1.9. Start navigation to location

The method starts a navigation session to the destination.

Navigation options can be set to instruct the navigation path to suite users with disabilities.

+ (BOOL)startNavigateToLocation:(IDLocation *)aLocation
withOptions:(IDNavigationOptions)navigationOption
andDelegate:(id<IDNavigationDelegate>)aDelegate

6.1.10. Start navigation to multiple locations

The method starts a navigation session with multiple destinations. The method starts navigat-

ing to the first location, then the next location on the list and so on. Navigation options can be set to instruct the navigation path to enable handicap accessibility routing.

```
+ (BOOL)startNavigateToLocations:(NSArray<IDLocation *> *)locations
                     withOptions:(IDNavigationOptions)navigationOption
                     andDelegate:(id<IDNavigationDelegate>)aDelegate
```

6.1.11. Start navigation simulation to location

The method starts a navigation simulation session to a destination from a location. Navigation options can be set to instruct the navigation path to suit users with disabilities.

```
+ (BOOL)startSimulationNavigationToLocation:(IDLocation *)aLocation
                               fromLocation:(IDLocation *)origionLocation
                                withOptions:(IDNavigationOptions)navigationOption
                                andDelegate:(id<IDNavigationDelegate>)aDelegate
```

6.1.12. Start simulation navigation to multiple locations

The method starts a navigation simulation session to multiple destinations from a location.

Navigation options can be set to instruct the navigation path to suit users with disabilities.

```
+ (BOOL)startSimulationNavigationToLocations:(IDLocation *)aLocation
                               fromLocation:(IDLocation *)origionLocation
                                withOptions:(IDNavigationOptions)navigationOption
                                andDelegate:(id<IDNavigationDelegate>)aDelegate
```

6.1.13. Continue navigation to next destinations

Call this method to instruct the navigation manager to continue to the next location in the

lineup, the next location is available as received from the delegation method:

```
- (void)navigationArriveToLocation:(IDLocation *)aLocation
```

nextLocations:(NSArray<IDLocation *> *)nextLocations.

the continueWithLocations method will be called:

+ (BOOL)continueWithLocations:(NSArray<IDLocation *> *)locations

6.1.14. Is the user currently navigating?

The method returns YES if the user is currently navigating, otherwise NO is returned.

+ (BOOL)isDuringNavigation

6.1.15. Is the user currently simulating location or navigation?

The method returns YES if the user is currently navigating, otherwise NO is returned.

+ (BOOL)isDuringSimulation

6.1.16. Stop navigation.

The method stops the current navigation session.

+ (void)stopNavigation

6.1.17. Get user location.

The method returns the user location; if the user tracking was not started nil is returned.

+ (IDUserLocation *)getUserLocation

6.1.18. Set current location as parking location

The method sets the current location as parking location

+ (void)setCurrentLocationAsParking

6.1.19. Get parking location

The method returns the saved parking location.

+ (IDLocation *)getParkingLocation

6.1.20. Set parking location

The method removes the parking location

+ (void)setParkingLocation

6.1.21. Remove parking location

The method removes the parking location

+ (void)removeParkingLocation

6.2. Location Listener protocol

The IDLocationListener protocol provides feedback on the location updates, status and events.

@protocol IDLocationListener <NSObject>

6.2.1. Location detection changes

The method is called on any situation that the detection state has changed during user location tracking, including navigation.

- (void)locationDetectionStateChanged:(NSError *)error

6.2.2. Update user position

The method is called each time the system has a new location for the user.

- (void)updateUserLocationWithLocation:(IDUserLocation *)aLocation

6.2.3. Campus Region Detection Changes

This method is called when the user location changed for campusId

- (void)regionEventChangedForCampusId:(NSString*)aCampusId
withEvent:(IDRegionEventType)anEventType;

6.2.4. Facility Region Detection Changes

This method is called when the user location changed for facilityId at campusId

- (void)regionEventChangedForFacilityWithID:(NSString*)aFacilityId
campusId:(NSString*)aCampusId
withEvent:(IDRegionEventType)anEventType;

6.3. Converted Location Listener protocol

The IDLocationListener protocol provides feedback on the converted location updates.

@protocol IDConvertedLocationListener <NSObject>

6.3.1. Update user converted location

This method is called when the user location be updated.

- (void)updateUserConvertedLocation:(IDConvertedLocation *)aLocation;

6.4. Navigation delegate protocol

The navigation delegation protocol supplies call back to the application's events.

@protocol IDNavigationDelegate <NSObject>

6.4.1. Update navigation status

The method is called when the system has and update on the current navigation status (during navigation session). i.e. reroute , ended and more.

- (void)navigationStatusUpdateWithStatus:(IDNavigationStatus)aStatus

6.4.2. Update navigation Route

This method is called when navigation or simulation navigation started

- (void)updateWithRoute:(IDRoute *)aRoute;

6.4.3. Update navigation status

This method is called when navigation engine will update instruction with navigation status

- (void)updateWithInstruction:(NSDictionary *)anInstruction andStatus:(IDNavigationStatus)aStatus;

6.4.4. play Instruction sound

This method is called when navigation engine will play instruction sound

- (void)playInstructionSound;

6.4.5. Arrive to a location during multi location navigation

The method is called when the user arrives to one of the destinations. The next locations arrangement holds the next locations on the multi locations route in the navigation order.

```
- (void)navigationArriveToLocation:(IDLocation *)aLocation
                     nextLocations:(NSArray<IDLocation *> *)nextLocations
```

7. Analytics & Statistics:

The SDK can provide full support of analytic and statistics by the beacons management

7.1. Enable/Disable statistics mode.

The method enables or disables the analytics mode. When it is enabled, statistics data will be uploaded to the server.

```
+ (void)enableAnalyticsMode:(BOOL)aMode;
```

7.2. Set analytics sample interval.

The method sets a sample interval for the system, which will be used for statistics and server interactions frequency.

```
+ (void)setAnalyticsSampleInterval:(NSInteger)aSampleInterval;
```

7.3. Set analytics time interval.

The method sets a time interval for the system, which will be used for server interactions data upload frequency.

```
+ (void)setAnalyticsTimeInterval:(NSInteger)aTimeInterval;
```

8. Instructions Controller

The instruction controller is responsible for displaying and activating the instructions.

8.1. Get the instruction controller

The method returns the instruction controller. The instruction controller needs to be set on

base view controller that holds the map view controller as a child controller.

+ (IDInstructionsController *)getInstructionsController.

8.2. Methods

8.2.1. Show instruction

The method will animate the instruction to a position from an origin.

- (void)presentInstructionFromOriginY:(CGFloat)yOrigintoPositionY:(CGFloat)yPosition

8.2.2. Dismiss instruction

The method will animate the instruction back to origin.

- (void)dismissInstruction

8.2.3. Update with instruction and status

The method updates the next instruction to show and to sound. E.g. “Turn Right”

- (void)updateWithInstruction:(NSDictionary *)anInstructionandStatus:(IDNavigationStatus)aStatus

8.2.4. Play instruction sound

The method will immediately play the current instruction sound.

- (void)playInstructionSound

8.2.5. Set instruction sound to mute

The method sets the instruction sound to mute or un-mute.

- (void)setSoundMute:(BOOL)mode

8.2.6. Get the route instruction list.

The method returns an array of dictionaries which contain information on each instruction in the route.

- (NSArray<NSDictionary *> *)getInstructionsList

8.2.7. Get route overview

The method returns an array of dictionaries that contain information and images of the route.

- (NSArray<NSDictionary *> *)getRouteOverviews

8.3. Delegate

The Instruction Controller delegates protocol informs and notifies the

@protocol IDInstructionControllerDelegate<NSObject>

8.3.1. Stop navigation

The method is called when the user touches the stop icon on the instruction bar.

- (void)stopNavigationTapped

8.3.2. Show instruction list

The method is called when the user swipes right on the instruction bar

- (void)showInstructionsList

8.3.3. Show route overview

The method is called when the user swipes left on the instruction bar

- (void)showRouteOverview

8.3.3. Finished play instruction.

The method is called when the vocal instruction has finished playing.
- *(void)didFinishPlayInstruction:(NSDictionary *)anInstruction*

9. GeoFencing & Location Awareness:

The geofence and location awareness enable the application to receive the events of "enter", "exit" or "in" for a specific region/s.

9.1. Methods

9.1.1. Add geofence awareness

The method provides ability to add custom geofence awareness, each geofence object is of type IDGeofence.

+ (void)addGeofenceFromArray:(NSArray<IDGeofence *> *)aGeofenceArray

9.1.2. Remove geofence awareness

The method removes all geofence objects provided in the array.

+ (void)removeGeofenceInArray:(NSArray<IDGeofence *> *)geofenceArray

9.1.3. Register for geofence types with delegate

The method registers the IDGeofenceDelegate delegate for custom geofences types.

+ (void)registerForGeofenceTypes:(NSArray<NSString *> *)aTypes
withDelegate:(id<IDGeofenceDelegate>)aDelegate;

9.1.4. UnRegister for geofence types with delegate

The method unregisters the IDGeofenceDelegate delegate for custom geofences types.

+ (void)unRegisterForGeofenceTypes:(NSArray<NSString *> *)aTypes
withDelegate:(id<IDGeofenceDelegate>)aDelegate;

9.1.5. UnRegister geofence delegate

The method unregisters IDGeofenceDelegate delegate.

+ (void)unregisterGeofenceDelegate:(id <IDGeofenceDelegate>)aDelegate

9.2. Delegate Protocol:

This section describes the geofence delegation protocol.

@protocolIDGeofenceDelegate<NSObject>

9.2.1. Notify that a geofence event occurred

Call this method when you have an Enterprise Permission of API KEY to get the raw events when a user enters a region (or not). The method is called each time an enabled geofence event occurs: user enters a geofence region, user exits a geofence region, or receive events while in the geofence region.

- (void)geofenceRegion:(IDGeofence *)aGeofence withEvent:(IDGeofenceEventType)type

10. Fundamental Components and Types Definitions

That section describes the fundamental components and definitions of SDK and API language:

10.1. IDError

Indoor Kit error type derived from NSError ; It describes that an error has occurred.

IDError.h file.

10.2. IDLocation

Object type that holds data of a location; full description is available in the IDLocation.h file.

10.3. IDUserLocation

Object type that holds data of a location; full description is available in the IDLocation.h file.

10.4. IDConvertedLocation

Object type that holds data of a location; full description is available in the IDConvertedLocation.h file.

10.5. IDGeofence

Object type that holds data of a geofence; full description is available in the IDGeofence.h file.

10.6. IDBeaconGeofence

Object type that holds data of a beacon geofence; full description is available in the IDGeofence.h file.

10.7. IDPoi

Object type that holds the POI data; full description is available in the IDPoi.h file.

10.8. IDQuery

IDQuery Class is to be used in order to filter and search IDKit data models Objects.

10.9. IDDualMapViewController

Object type that represents the navigation map view controller; full description is available on theIDDualMapViewController.h

10.10. IDMapViewController

Object type that represents the navigation map view controller; full description is available on theIDMapViewController.h

10.11. IDDataUpdateStatus

Enumerator type that indicates the phase of update:

- Check for update.
- Copy files.
- Download data.
- Initialize data objects.
- Done.

10.12. IDMapType

Enumerator type that indicates possible outdoor map view options:

- Standard map view.
- Satellite map view.
- Hybrid map view.

10.13. IDMapRotationType

Enumerator type that indicates possible options of the map view rotation options:

- Static for both Indoor and outdoor.
- Compass for both Indoor and outdoor.

- Orientation for both Indoor and outdoor.

10.14. IDMapLayerType

Enumerator type that indicates map layers type:

- Map campus.
- Map facility.
- Map labels.
- Map POIs.
- Map walking path.
- Map route path.
- Map user position.

10.15. IDNavigationStatus

Enumerator type that indicates current navigation status:

- Idle.
- Start.
- Navigate.
- Turn back.
- Reroute.
- Stopped.
- Ended.

10.16. IDNavigationOptions

Enumerator type that indicates the required path type:

- Regular
- Disabled.
- Staff.

10.17. IDGeofenceEventType

Enumerator type that indicates the event when geofence event fires:

- In.

- Out.

- InRegion.

10.18. IDQueryType

Enumerator type that indicates the query conditions type

- BEGINSWITH.

- CONTAINS.

- ENDSWITH.

- LIKE.

- MATCHES.