iOS SDK/API DOCUMENTATION

DOCUMENT VERSION:	11.1
iOS SDK VERSION:	3.9.2
iOS BUNDLE VERSION	3.9.2
DATE REVISED	April 2022

Installation

Manual

Drag the Framework folder into your project.

Cocoapods

CocoaPods is a dependency manager for Cocoa projects. For usage and installation instructions, visit their website.To integrate SpreoUIModule into your Xcode project using CocoaPods, specify it in your Podfile:

pod 'SpreoPod'

Usage

Read more about usage in “How to use” document

Table of Contents

1. Document Overview 9
2. System Overview 9
3. General Configurations and Settings 9
4. Methods 9
4.1. Get the SDK Version String 9
4.2. Registration and Association 9
4.3. Update client data and initialize. 9
4.3.1. Check for Data and updates 9
4.3.2. Delegate Protocol 10
4.3.3. Notify update and initialization state 10
4.4. Languages 10
4.4.1. Supported language 10
4.4.2. Get current language 10
4.4.3. Set current language 10
4.5. Monitor campus region 10
4.6. Set user identifier 10
4.7. Set logger mode 11
4.8. User reaction to local notification 11
4.8.1. User reaction to local notification text 11
4.9. Get Campuses IDs 11
4.10. Get info For Campus With ID 11
4.11. Receive all facilities IDs for a given campus ID 11
4.12. Get info for facility with ID at campus with ID 12
4.13. Get campus info 12
4.14. Get facility info 12
4.15. Get floor info 12
4.16. Get POI/POIs 12
4.17. Get POIs sorted alphabetically 12
4.18. Get POIs sorted by distance from location 13
4.19. Get POIs Categories List 13
4.20. Get POI/POIs with category 13
4.21. Get POIs sorted alphabetically by categories 13
4.22. Get POIs sorted by distance from location with categories 13
4.23. Get POIs distance API 14
4.24. Request Itinerary POI's Array 14
4.25. Request Ordered Itinerary POI's Array 14

4.26. Store Parking Location 14
4.25. Request Parking Location 14
4.26. Remove Parking Location 15
4.27. Convert Location 15
4.28. Get custom user location parking icon 15
4.29. Get default app settings 15
4.30. Get facility by facility name and campus id 15
4.31. Get facility titles by campus id 15
4.32. Get floors from route 15
4.33. Get icon by key 15
4.34. Get icons keys list 16
4.35. Get venue id for max floors count by campus id 16
4.36. Get nearby parking for poi 16
4.37. Get POIs with categories by path id 16
4.38. Get project id 16
4.39. Get project location type 16
4.40. Get server URL 16
4.41. Get show navigation markers status 16
4.42. Get theft protection state 17
4.43. Get web interface URL 17
4.44. Get handicapped routing status 17
4.45. Set handicapped routing status 17
4.46. Initialize data with delegate 17
4.47. Check is user in campus 17
4.48. Check location services enabled 18
4.49. Check navigation distance unit is feet 18
4.50. Reset floors in route 18
4.51. Set current user location 18
4.52. Set custom user location icon 18
4.53. Set dashed route 18
4.54. Set default user location 18
4.55. Set display user location icon 18
4.56. Set force fix to user location 19
4.57. Set location report limitation 19
4.58. Set monitoring mode 19
4.59. Set poi marker location 19
4.60. Set project location type 19

4.61. Set server URL 19
4.62. Set show navigation markers 19
4.63. Set staff routing 19
4.64. Set user location sharing mode 20
4.65. Set user location sharing time interval 20
4.66. Set zone status 20
4.67. Sort POIs alphabetically by categories and path id 20
4.68. Sort POIs alphabetically by path id 20
4.69. Sort POIs distantly by location 20
4.70. Sort POIs distantly by location, categories and path id 20
4.71. Sort POIs distantly by location and path id 21
4.72. Get staff routing status 21
4.73. Start initial zone detection 21
4.74. Start polling users location sharing data 21
4.75. Start user location track with CLLocationManager 21
4.76. Start zone detection 21
4.77. Stop polling users location sharing data 22
4.78. Stop zone detection 22
5. Map View Containers 22
5.1. IDMapViewProtocol 22
5.1.1. Map display adjustments 22
5.1.1.1. Set map type 22
5.1.1.2. Set map show layer 22
5.1.1.3. Show All POIs 23
5.1.1.4. Hide All POIs 23
5.1.1.5. Show All Labels 23
5.1.1.6. Hide All Label 23
5.1.1.7. Set map filter POIs 23
5.1.1.8. Set map rotation mode 23
5.1.1.9. Set time interval for auto follow 23
5.1.1.10. Set the in region radius for a POI. 23
5.1.2. Map content adjustments 24
5.1.2.1. Show location 24
5.1.2.2. Show Poi 24
5.1.2.3. Show Poi Bubble 24
5.1.2.4. Hide Poi Bubble 24
5.1.2.5. Centre Facility Map 24
5.1.2.6. Centre Campus Map 24
5.1.2.7. Show floor map at facility with Id 24
5.1.2.8. Show my position 25
5.1.2.9. Zoom in 25
5.1.2.10. Zoom out 25
5.1.2.11. Set Map Zoom Level 25
5.1.2.12. Map reload data 25
5.1.2.13. Current presented floor identifier 25
5.2. Container A Map - IDDualMapViewController 25

5.2.1. IDMapViewController Methods 26
5.2.1.1. Provide Google Maps API Key 26

5.2.2. IDDualMapViewController Map Settings 26
5.2.2.1. Indoor picker view 26
5.2.2.2. My Location button 26
5.2.2.2. Compass button 26

5.2.3. IDDualMapViewController Delegation Protocol IDDual- MapViewControllerDelegate 26
5.2.3.1. Custom the user annotation icon 26
5.2.3.2. Custom the pois icons 26
5.2.3.3. Custom the labels icons and visibility 27
5.2.3.4. Custom the Pois Bubble 27
5.2.3.5. Custom the Parking annotation 27
5.2.3.6. Custom the Navigation Route Colour 27
5.2.3.7. My location button did tapped 27
5.2.3.8. POI icon Tapped 27
5.2.3.9. POI callout was tapped 28
5.2.3.10. User icon did tapped 28
5.2.3.11. Parking icon icon did tapped 28
5.2.3.12. label icon did tapped 28
5.2.3.13. region focus did changed 28
5.2.3.14. floor id did changed 28
5.2.3.15. zoom level did changed 28
5.2.3.16. user follow me mode did changed 29

5.3. Container B Map - IDMapViewController 30
5.3.1.1. Get the view for a POI 30
5.3.1.2. Set the Max POI popups/ bubbles number 30
5.3.1.3. Close all POIs bubbles. 30
5.3.1.4. Add overlays to outdoor map 30
5.3.1.5. Draw trip overview 30
5.3.1.6. Show User Location With Bubble 31
5.3.1.7. Remove trip overview 31
5.3.1.8. Set map lock to region 31
5.3.1.9. Title for floor identifier 31

5.3.2. IDMapViewController Delegation Protocol IDMapView ControllerDelegate 31
5.3.2.1. View for user 31
5.3.2.2. View for User Location with Bubble 31
5.3.2.3. View for direction pointing 24
5.3.2.4. View for campus 24
5.3.2.5. View for campus graphics 24
5.3.2.6. View for facility 24
5.3.2.7. View for label 24
5.3.2.8. View for POI 24
5.3.2.9. View for Parking Annotation 24
5.3.2.10. POI selected 25
5.3.2.11. POI callout was tapped 25
5.3.2.12. POI callout action was tapped 25
5.3.2.13. User Annotation did tapped 25
5.3.2.14. Parking Annotation did tapped 25
5.3.2.15. Map will change floor 25

5.3.2.16. Map has changed floor 25
5.3.2.17. Map will be swapped 26
5.3.2.18. Map Swapped 26
5.3.2.19. Map arrow image for route 26
5.3.2.20. Map Color for trip overview route 26
5.3.2.21. Map Color for trip overview circle 26
5.3.2.22. Map Color for trip overview “Arrived Circle” 26

5.4. POI Custom View For Callout View Protocol IDCallouts Protocol 26
5.4.1. Bubbles Layer View 27
5.4.2. Allow The Protocol Feedbacks 27
5.4.3. Custom Bubble View for Poi 27
5.4.4. Present POI Bubble View 27
5.4.5. Dismiss POI Bubble View 27
5.5.1. Entered POI region 27
5.5.2. Exited the POI region 28

6. Positioning & Navigation: 28

6.1. Methods 28
6.1.1. Start user positioning updates 28
6.1.2. Set the navigation delegate. 28
6.1.3. Is user position tracked? 28
6.1.4. Stop user positioning tracking. 28
6.1.5. Simulate User Location. 29
6.1.6. Reset user positioning tracking. 29
6.1.7. Notify the SDK about background mode. 29
6.1.8. Notify SDK about foreground mode. 29
6.1.9. Load route using API call 29
6.1.10. Start navigation to location 29
6.1.11. Start navigation to multiple locations 30
6.1.12. Start navigation simulation to location 30
6.1.13. Start simulation navigation to multiple locations 30
6.1.14. Continue navigation to next destinations 30
6.1.15. Is the user currently navigating? 31
6.1.16. Is the user currently simulating location or navigation? 31
6.1.17. Stop navigation. 31
6.1.18. Get user location. 31
6.1.19. Set current location as parking location 31
6.1.20. Get parking location 31
6.1.21. Set parking location 31
6.1.22. Remove parking location 32

6.2. Location Listener protocol 32
6.2.1. Location detection changes 32
6.2.2. Update user position 32
6.2.3. Campus Region Detection Changes 32
6.2.4. Facility Region Detection Changes 32

6.3. Converted Location Listener protocol 33
6.3.1. Update user converted location 33

6.4. Navigation delegate protocol 33
6.4.1. Update navigation status 33
6.4.2. Update navigation Route 33
6.4.3. Update navigation status 33
6.4.4. play Instruction sound 33
6.4.5. Arrive to a location during multi location navigation 34

7. Analytics & Statistics: 34
7.1. Enable/Disable statistics mode. 34
7.2. Set analytics sample interval. 34
7.3. Set analytics time interval. 34

8. Instructions Controller 34

8.1. Get the instruction controller 34

8.2. Methods 35
8.2.1. Show instruction 35

8.2.2. Dismiss instruction 35
8.2.3. Update with instruction and status 35
8.2.4. Play instruction sound 35
8.2.5. Set instruction sound to mute 35
8.2.6. Get the route instruction list. 35
8.2.7. Get route overview 35

8.3. Delegate 36
8.3.1. Stop navigation 36
8.3.2. Show instruction list 36
8.3.3. Show route overview 36
8.3.3. Finished play instruction. 36

9. GeoFencing & Location Awareness: 36

9.1. Methods 37
9.1.1. Add geofence awareness 37
9.1.2. Remove geofence awareness 37
9.1.3. Register for geofence types with delegate 37
9.1.4. UnRegister for geofence types with delegate 37
9.1.5. UnRegister geofence delegate 37
9.1.6. Register for CLLocationManagerDelegate 37
9.1.7. Register for ConvertedLocationListener 38
9.1.8. Register for LocationListener 38
9.1.9. Register for ZoneDetectionListener 38
9.1.10. Unregister for CLLocationManagerDelegate 38
9.1.11. Unregister for ConvertedLocationListener 38
9.1.12. Unregister for LocationListener 38
9.1.13. Unregister for ZoneDetectionListener 38

9.2. Delegate Protocol: 38
9.2.1. Notify that a geofence event occurred 39

10. Fundamental Components and Types Definitions 39

10.1. IDError 39
10.2. IDLocation 39
10.3. IDUserLocation 39
10.4. IDConvertedLocation 39
10.5. IDGeofence 39
10.6. IDBeaconGeofence 39
10.7. IDPoi 39
10.8. IDQuery 39
10.9. IDDualMapViewController 39
10.10. IDMapViewController 39
10.11. IDDataUpdateStatus 39
10.12. IDMapType 39
10.13. IDMapRotationType 39
10.14. IDMapLayerType 40
10.15. IDNavigationStatus 40
10.16. IDNavigationOptions 40
10.17. IDGeofenceEventType 40

10.18. IDQueryType 41

11. Custom Markers 42

11.1. Methods 42
11.1.1. Add custom markers 42
11.1.2. Show custom markers 42

12. Analytics 42

12.1. Set analytics mode 42
12.2. Send analytics report 42
12.3. Set analytics timeInterval 43

13. Distance info 43

13.1. Getting distance to next navigation instruction 43
13.2. Setting units to meters/feet for distance info 43
13.3. Getting total distance of navigation route 43
13.4. Getting distance from location to poi 43
13.5. Getting distance from location to poi list 43

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.
Also logger mode YES will enable detailed console logging.
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. Get POIs distance API

The method returns an array of IDPoi objects with calculated distance object IDPoiDistance.

(NSArray<IDPoi *> *_Nonnull)getDistanceToPoiList:(NSArray *_Nonnull)poiList fromLocation:(IDLocation*)location sortedByDistance:(BOOL)isSortedByDistance;

The method will return IDPoi object with calculated distance from passed location.

(CLLocationDistance)getDistanceToPoi:(IDPoi*)poi fromLocation:(IDLocation *)location;

4.24. 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;

4.25. 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;

4.26. Store Parking Location

The method set the location as parking location.

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

+

4.27. Request Parking Location

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

+ (IDLocation *)getParkingLocation;

4.25. Remove Parking Location

The method removes the parking location.

+ (void)removeParkingLocation;

4.26. Convert Location

The method converts location from IDLocation Object to IDConvertedLocation Object.

+ (IDConvertedLocation *_Nonnull)convertLocation:(IDLocation *_Nonnull)aLocation;

4.27. Get custom user location parking icon

The method converts location from IDLocation Object to IDConvertedLocation Object.

+ (IDConvertedLocation *_Nonnull)convertLocation:(IDLocation *_Nonnull)aLocation;

4.28. Get default app settings

Call This method in order to get downloaded settings for application.

+ (nullable NSDictionary*)getDefaultAppSettings;

4.29. Get facility by facility name and campus id

Information dictionary contaning information on refered facility @code {@"title": facility.title, @"subtitle": facility.subtitle, @"location": location, @"perimeter" : @{@"top_left_coordinate_lat": @(facility.upperLeftCoordinate.latitude), @"top_left_coordinate_lon": @(facility.upperLeftCoordinate.longitude), @"bottom_right_coordinate_lat": @(facility.lowerRightCoordinate.latitude), @"bottom_right_coordinate_lon": @(facility.lowerRightCoordinate.longitude) @"rot_angle": @(facility.rotationAngle), @"entrance_floor": @(facility.entranceFloor), @"floors_count": @(facility.floors.count)}

+ (NSSet*)getFacilityForName:(NSString*)fName atCampusId:(NSString*)aCampusId;

4.30. Get facility titles by campus id

Returns array of facility titles

+ (NSArray <NSString *> *_Nonnull)getFacilityTitlesForCampusID:(NSString *_Nonnull)aCampusId;

4.31. Get floors from route

Returns array of all project floors ids of selected route

+ (NSArray <NSString *> *_Nonnull) getFloorsInRoute;

4.32. Get icon by key

Call This method in order to get icon image for key of downloaded if exist

+ (nullable UIImage *)getIconByKeyName:(NSString *_Nonnull)aKeyName;

4.33. Get icons keys list

Call This method in order to get all downloaded icons names

+ (NSArray <NSString *> *_Nonnull)getIconsKeysNamesList;

4.34. Get venue id for max floors count by campus id

Returns the venue id that has the max number of floors

+ (NSString* _Nonnull) getMaxVenueIdFloorsCountAtCampusId:(NSString *_Nullable)campusId;

4.35. Get nearby parking for poi

Returns closest parking Poi.

+ (IDPoi*_Nullable)getNearbyParkingForPoi:(IDPoi*_Nullable)aPoi;

4.36. Get POIs with categories by path id

Returns array of IDPoi objects that holds the category string specified at the specified path id.

+ (NSArray <IDPoi *> *_Nonnull)getPOIsWithCategories:(NSArray <NSString *> *_Nonnull)aCategories atPathID:(NSString *_Nonnull)aPathID;

4.37. Get project id

Returns the saved project id.

+ (NSString *_Nullable)getProjectId;

4.38. Get project location type

Returns the saved project id.

+ (NSString *_Nullable)getProjectId;

4.39. Get server URL

Returns the saved server url.

+ (NSString *_Nullable)getServerURL;

4.40. Get show navigation markers status

Returns YES, in case markers is enabled.

+ (BOOL) getShowNavigationMarkers;

4.41. Get theft protection state

Returns true in case the filtered devices count > the project exit facility param, otherwise false.

+ (BOOL)getTheftProtectionState;

4.42. Get web interface URL

Returns an webinterface url for sharing.

+ (NSString *_Nonnull)getWebInterfaceUrl;

4.43. Get handicapped routing status

Returns the Indoor Kit handicapped status

+ (BOOL)handicappedRouting;

4.44. Set handicapped routing status

The method is to be called to set The Indoor Kit set handicapped status, accordingly the server route path will be change

+ (void)setHandicappedRouting:(BOOL)aHandicapped;

4.45. Initialize data with delegate

The method initializes the check of remote data update on 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)initializeDataWithDelegate:(nullable id<IDDataUpdateDelegate>)aDelegate;

IDDataUpdateDelegate **returns** data download status

- (void)dataUpdateStatus:(IDDataUpdateStatus)aStatus;

IDDataUpdateDelegate **returns error**

- (void)dataUpdateFailedWithError:(IDError*)anError;

4.46. Check is user in campus

The method checks if user locating in Campu by max distance between current user location and campus location. If distance less then maxDistanceFromCampus, then user in the campus

+ (BOOL)isUserInCampus:(NSInteger)maxDistanceFromCampus;

4.47. Check location services enabled

The method checks the availability of the location services

+ (BOOL)locationServicesEnabled;

4.48. Check navigation distance unit is feet

The method returns a is distance info unit feet

+ (BOOL)navigationDistanceUnitIsFeet;

4.49. Reset floors in route

The method reset All floors id's.

+ (void) resetFloorsInRoute;

4.50. Set current user location

The method sets the current user location.

+ (void)setCurrentUserLocation;

4.51. Set custom user location icon

The method sets the custom user location icon

+ (void)setCustomUserLocationIcon:(UIImage *_Nullable)userLocationIcon;

4.52. Set dashed route

When it's turned on, Navigation route line will show dashed/dotted

+ (void)setDashedRoute:(BOOL)aMode;

4.53. Set default user location

The method sets the user default location.

+ (void)setDefaultUserLocation:

4.54. Set display user location icon

The method sets the display user location icon

+ (void)setDisplayUserLocationIcon:(BOOL)dispayUserLocationIcon;

4.55. Set force fix to user location

The method force a fix to the current user location object.

+ (void)setForceFixToUserLocation:(IDUserLocation *_Nonnull)aUserLocation;

4.56. Set location report limitation

With this method you can limit analytics location update with type = 0 - Always type = 1 - Indoor type = 2 - In Campus

+ (void)setLocationReportLimitation: (ReportLimitation)reportLimit;

4.57. Set monitoring mode

The method enables / disables the monitoring mode,when its enabled beacons data will be uploaded to the server , in case that that method is not called or a NO value is received, the statistics will not be upload any data to the server.

+ (void)setMonitoringMode:(BOOL)aMode;

4.58. Set poi marker location

The method set the poi marker location.

+ (void)setPoiMarkerLocation:(IDLocation *_Nullable)aLocation;

4.59. Set project location type

The method sets the project location type.

+ (void)setProjectLocationType:(IDProjectLocationType)aProjectLocationType;

4.60. Set server URL

The method save the server url.

+ (void)setServerURL:(NSString *_Nullable)serverURL;

4.61. Set show navigation markers

The method sets or removes the navigation markers

+ (void) setShowNavigationMarkers:(BOOL)show;

4.62. Set staff routing

The method is to be called to set The Indoor Kit set staff route status, accordingly the server route path will be change

+ (void)setStaffRouting:(NSInteger)aStaffRoute;

4.63. Set user location sharing mode

The method enables / disables the user location sharing mode. interval in seconds between updates is 300 seconds (5 min).

+ (void)setUserLocationSharingMode:(BOOL)aMode;

4.64. Set user location sharing time interval

The method sets the user location sharing time interval updates, 300 seconds (default)

+ (void)setUserLocationSharingTimeInterval:(NSTimeInterval)interval;

4.65. Set zone status

The method sets current zone status to the IDKit Zone Detector Manger. This change will be active until the IDKit Zone Detector Manger detects a Zone change:

+ (void)setZoneStatus:(IDZoneStatus)aZoneStatus;

4.66. Sort POIs alphabetically by categories and path id

The method returns sorted array alphabetically of IDPoi objects that holds the categories strings specified at the specified path id.

+ (NSArray <IDPoi *> *_Nonnull)sortPOIsAlphabeticallyWithCategories:(NSArray <NSString *> *_Nonnull)aCategories atPathID:(NSString *_Nonnull)aPathID;

4.67. Sort POIs alphabetically by path id

The method returns sorted array alphabetically of IDPoi objects for each suitable founded poi.

+ (NSArray <IDPoi *> *_Nonnull)sortPOIsAlphabeticallyWithPathID:(NSString *_Nonnull)aPathID;

4.68. Sort POIs distantly by location

The method returns sorted array of IDPoi objects for each suitable founded poi by distance of location

+ (NSArray <IDPoi *> *_Nonnull)sortPOIsDistantly:(NSArray *_Nonnull)aPOIs fromLocation:(IDLocation *_Nonnull)aLocation;

4.69. Sort POIs distantly by location, categories and path id

The method returns sorted array of IDPoi objects for each suitable founded poi by distance of location. of IDPoi objects that holds the categories strings specified at the specified path id.

+ (NSArray <IDPoi *> *_Nonnull)sortPOIsDistantlyWithCategories:(NSArray <NSString *> *_Nonnull)aCategories atPathID:(NSString *_Nonnull)aPathID fromLocation:(IDLocation *_Nonnull)aLocation;

4.70. Sort POIs distantly by location and path id

The method returns sorted array of IDPoi objects for each suitable founded poi by distance of location.

+ (NSArray <IDPoi *> *_Nonnull)sortPOIsDistantlyWithPathID:(NSString *_Nonnull)aPathID fromLocation:(IDLocation *_Nonnull)aLocation;

4.71. Get staff routing status

The method returns The Indoor Kit staff route 1/0 default will be 0

+ (NSInteger)staffRouting;

4.72. Start initial zone detection

The method starts the IDKit Zone Detector Manger initial zone updates.

+ (NSString *_Nonnull)getWebInterfaceUrl;

4.73. Start polling users location sharing data

The method starts polling users location at certain intervals.

+ (void)startPollingtUsersLocationSharingDataWithInterval:(NSTimeInterval)interval forUserIds:(NSArray<NSString *> *_Nullable)userIds update:(nullable void (^) (NSArray<IDUser *> *_Nonnull users))update failure:(nullable void (^) (IDError *_Nonnull error))failure;

Interval
Interval in seconds between updates.

userIds
filter users with specific Ids.

update
A block object to be executed when the request operation finishes successfully with result. This block has no return value and takes one argument : the users data array of IDUser objects.

failure
A block object to be executed when the request operation finishes unsuccessfully. This block has no return value and takes one argument : the error IDError.

4.74. Start user location track with CLLocationManager

The method starts user location track with CLLocation Manager

+ (BOOL)startUserLocationTrackWithCLLocationManager:(CLLocationManager *_Nullable)aLocationManager;

4.75. Start zone detection

The method starts the IDKit Zone Detector Manger zone updates.

+ (BOOL)startZoneDetection;

4.76. Stop polling users location sharing data

The method stop users location sharing polling data at certain intervals.

+ (void)stopPollingUsersLocationSharingData;

4.77. Stop zone detection

The method stop the IDKit Zone Detector Manger initial zone updates.

+ (void)stopZoneDetection;

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 zoom 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 byidentifier:

-(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. Load route using API call

The method starts loading route from api call

+ (void)callServerRouteAPIFrom:(IDLocation *_Nonnull)anOrigionLocation toLocation:(IDLocation *_Nonnull)anDestinationLocation

Route loading finishing received from the IDNavigationDelegate delegation method:

- (void)isServerAPICallSuccess:(BOOL)aStatus typeError:(NSInteger)type;

6.1.10. 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

6.1.11. Start navigation to multiple locations

The method starts a navigation session with multiple destinations. The method starts navigating 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

6.1.12. 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

6.1.13. 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

6.1.14. 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

6.1.15. Is the user currently navigating?

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

+ (BOOL)isDuringNavigation

6.1.16. Is the user currently simulating location or navigation?

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

+ (BOOL)isDuringSimulation

6.1.17. Stop navigation.

The method stops the current navigation session.

+ (void)stopNavigation

6.1.18. Get user location.

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

+ (IDUserLocation *)getUserLocation

6.1.19. Set current location as parking location

The method sets the current location as parking location

+ (void)setCurrentLocationAsParking

6.1.20. Get parking location

The method returns the saved parking location.

+ (IDLocation *)getParkingLocation

6.1.21. Set parking location

The method removes the parking location

+ (void)setParkingLocation

6.1.22. 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 *)anInstruction andStatus:(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

9.1.6. Register for CLLocationManagerDelegate

The method register to receive location and heading updates from a CLLocationManager object. Upon receiving a successful location or heading update, you can use the result to update your user interface or perform other actions. If the location or heading could not be determined, you might want to stop updates for a short period of time and try again later.

+ (BOOL)registerToCLLocationManagerDelegateWithDelegate:(nullable
id<CLLocationManagerDelegate>)aDelegate;

9.1.7. Register for ConvertedLocationListener

The method add the object that implements the IDConvertedLocationListener protocol

+ (void)registerToConvertedLocationListenerWithDelegate:(nullable id<IDConvertedLocationListener>)aDelegate;

9.1.8. Register for LocationListener

The method add the object that implements the IDLocationListener protocol

+ (void)registerToLocationListenerWithDelegate:(nullable id<IDLocationListener>)aDelegate;

9.1.9. Register for ZoneDetectionListener

The method add the object that implements the IDZoneDetectionListener protocol

+ (BOOL)registerToZoneDetectionListenerWithDelegate:(id<IDZoneDetectionListener> _Nullable)aDelegate;

9.1.10. Unregister for CLLocationManagerDelegate

The method unregister to receive location and heading updates from a CLLocationManager object. Upon receiving a successful location or heading update, you can use the result to update your user interface or perform other actions. If the location or heading could not be determined, you might want to stop updates for a short period of time and try again later.

+ (BOOL)**un**registerToCLLocationManagerDelegateWithDelegate:(nullable id<CLLocationManagerDelegate>)aDelegate;

9.1.11. Unregister for ConvertedLocationListener

The method remove the object that implements the IDConvertedLocationListener protocol

+ (void)**un**registerToConvertedLocationListenerWithDelegate:(nullable id<IDConvertedLocationListener>)aDelegate;

9.1.12. Unregister for LocationListener

The method remove the object that implements the IDLocationListener protocol

+ (void)registerToLocationListenerWithDelegate:(nullable id<IDLocationListener>)aDelegate;

9.1.13. Unregister for ZoneDetectionListener

The method remove the object that implements the IDZoneDetectionListener protocol

+ (BOOL)registerToZoneDetectionListenerWithDelegate:(id<IDZoneDetectionListener> _Nullable)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.

11. Custom Markers

11.1. Methods

11.1.1. Add custom markers

The method provides ability to add custom markers.

+ (NSArray*_Nonnull)addCustomMarkersWithPOIs: (NSArray*_Nonnull)aPOIs;

11.1.2. Show custom markers

The method provides ability to show custom markers, which was added early.

If markersIds = nil will show all exist custom markers

+ (void)showCustomMarkersWithIds:(NSArray*_Nullable)markersIds;

11.1.2. Remove custom markers

The method provides ability to remove custom markers, which was added early.

If markersIds = nil will remove all exist custom markers

+ (void)removeCustomMarkersWithIds:(NSArray*_Nullable)markersIds;

12. Analytics

12.1. Set analytics mode

The method enables / disables the analytics mode,when its enabled statistics data will be uploaded to the server , in case that that method is not called or a NO value is received, the statistics will not be upload any data to the server

+ (void)setAnalyticsMode:(BOOL)aMode;

12.2. Send analytics report

The method sends analytics data to server, use this API when user did search for poi or did navigate for poi, using this API updates the server analyitcs data

+ (void)sendAnalyticsReportWithAction:(NSString*_Nullable)anAction
andTitle:(NSString*_Nullable)aTtile floorId:(NSInteger)floorID
campusId:(NSString*_Nullable)campusID facilityId:

12.3. Set analytics timeInterval

The method set a time interval in minutes for the system, the time interval will be used for data upload timer, in case nil value is received, 300 sec 5 minutes is default.

+ (void)setAnalyticsTimeInterval:(CGFloat)aTimeInterval;

13. Distance info

13.1. Getting distance to next navigation instruction

In delegate method for update instruction (see item 8.2.3) you will receiving the (NSDictionary *)anInstruction object, it contains distance value by key ”distance”

Example:

```
- (void)updateWithInstruction:(NSDictionary *)anInstruction andStatus:(IDNavigationStatus)aStatus {
        NSNumber *distanceToNextInstriuction = anInstruction[@"distance"];
}
```

13.2. Setting units to meters/feet for distance info

The IDKit method is to set the Navigation distance info units to meters/feet

+ (void)setNavigationDistanceUnitToFeet:(BOOL)isFeet;

13.3. Getting total distance of navigation route

The IDDualMapViewController method returns a total disatace of current navigation route

- (NSNumber*)getTotalDistanceOfNavigationRoute;

13.4. Getting distance from location to poi

The IDKit method returns a disatace from location to poi

+ (CLLocationDistance)getDistanceToPoi:(IDPoi *)poi fromLocation:(IDLocation *)location;

13.5. Getting distance from location to poi list

The IDKit method returns a list of IDPoiDistance instances, containing the poi and the distance to it. Method returns list with distances from location with ability to sort it by distance.

+ (NSArray <IDPoiDistance *> *)getDistanceToPoiList:(NSArray <IDPoi *> *)poiList fromLocation:(IDLocation *)location sortedByDistance:(BOOL)isSortedByDistance;