



Android SDK / API Documentation

DOCUMENT VERSION	4.7
Android SDK Version	4.7
DATE REVISED	December 2018

Table of Contents

1. Application Setup	6
Step 1: Add Spreo SDK to you project	6
Step 2: Add Google dependencies	6
Step 3: Set the required SDK level	6
Step 4: Set the required permissions	6
Step 5: Handle permissions	6
Step 6: Set your SPREO API key	6
Step 7: Set your Google API key	7
2. Update and Load the Required Data	7
Step 1: Implement ConfigsUpdaterListener interface	7
Step 2: Subscribe to update events and start the update	7
Step 3: Subscribe to sdk messages events	7
Step 4: When the download ends you can start the location service	7
3. Implement the SpreoDualMapView	8
Step 1: Add a SpreoDualMapView to your XML layout	8
Step 2: Add a SpreoDualMapView member in your activity	8
Step 3: Initiate the SpreoDualMapView	8
Step 4: Add methods required for Google map functionality	8
4. Enable Application to Receive and React to SpreoDualMapView Events	8
Step 1: Implement SpreoDualMapViewListener interface	8
Step 2: Register to map events	9
5. Interact with the SpreoDualMapView	9
Example 1: Change the map zoom level	9
Example 2: Present a specific campus	9
Example 3: Present a specific facility.	10
Example 3: Center map to present a specific POI	10
Example 4: Center map to present the user location	10
Example 5: Center map to present a location	10
Example 6: Get the ID of the presented floor	10
Example 7: Get the title of a specific floor	10
Example 8: Change the default user icon	10
Example 9: Redraw pois	10
Example 10: Register/Unregister from mapView events	10
Example 11: Register/Unregister from navigation events	11
Example 12: Open poi bubble	11
Example 13: Close poi bubble	11
Example 14: Customizing my parking icon	11
Example 15: Open my parking bubble	11
Example 16: Close my parking bubble	11
Example 17: Show/hide all pois on map	11
Example 18: Change icon for a poi	11

Example 19: Change icon for poi list	11
Example 20: Set visible pois list	12
Example 21: Return to default zoom	12
Example 22: Present a specific floor	12
Example 23: Present multi point navigation	12
Example 24: Remove the multi point navigation	12
Example 25: Set the visible labels	12
Example 26: Hide all labels	12
Example 27: Show all labels	12
Example 28: Displaying friends locations on the map	12
6. Navigation	13
Example 1: Start navigation	13
Example 2: Stop navigation	13
Example 3: Simulate navigation	13
Example 4: Getting total route distance	13
Example 5: Using feet units for distance measurement	14
7. Enable Your Application to Receive and React to Navigation Events	14
Step 1: Implement the SpreoNavigationListener interface	14
Step 2: Register to the navigation events	14
Step 3: Use INavInstruction object from OnNavigationInstructionChanged callback to get the instruction details	14
Example 1: Display distance to next navigation instruction	14
8. Save and Navigate to Parking Location	15
Example 1: Save your current location as parking location	15
Example 2: Save a location as parking location	15
Example 3: Remove parking location	15
Example 4: Check if there is a saved parking location	15
Example 5: Get the saved parking location	15
Example 6: Navigate to the saved parking location	15
9. Enable your Application to Receive and React to Real Time Location	15
Step 1: Implement MyLocationListener interface	15
Step 2: Subscribe to location events	15
10. Working with the POI list	16
Example 1: Get the POI list for a specific campus	16
Example 2: Get the POI list for a specific facility	16
Example 3: Get the POI list for a specific floor	16
Example 4: Get the POI categories list	16
Example 5: Set the list of the visible categories on map	16
Example 6: Get POIs sorted alphabetically	17
Example 7: Get POIs sorted by distance from location	17
Example 8: Get POIs with category	17
Example 9: Get POIs sorted alphabetically with categories	17
Example 10: Get POIs sorted by distance from location with categories	17
Example 11: Getting distance to POI or to the list of POIs	17

11. Change Settings	17
Example 1: Set the type of Google map	17
Example 2: Set the navigation route color	17
Example 3: Get the navigation route color	18
Example 4: Set the visibility of the navigation instructions	18
Example 5: Get the visibility of the navigation instructions	18
Example 6: Set the status of the instructions sound	18
Example 7: Get the status of the instructions sound	18
Example 8: Set simplified instruction	18
Example 9: Set the interval for the map auto follow mode	18
Example 10: Set the map rotation type	18
Example 11: Get the map rotation type	19
Example 12: Set the map navigation type	19
Example 13: Set whether to include the parking in the multiPois or not	19
Example 14: Set whether to include entrances in the multiPois or not	19
Example 15: Set whether to include switch floors in the multiPois or not	19
Example 16: Set the ui options for multi point navigation	19
Example 17: Set the dual map default zoom	19
Example 18: Set the visibility of the floor picker	19
Example 19: Set feet as measured units for the distance to next navigation instruction	20
Example 20: Hide content of the top floor	20
Example 21: Set up location analytics	20
12. Data Provider	20
Example 1: Get the SDK version	20
Example 2: Get the selected campus ID	20
Example 3: Get the campuses list	20
Example 4: Get the facilities list for a specific campus	20
Example 5: Get the title of a specific floor	20
Example 6: Get info for a specific campus	21
Example 7: Get info for a specific facility	21
Example 8: Get info for a specific floor	21
Example 9: Get the facility of the floor picker (the facility with highest number of floors)	22
13. Interfaces	22
Interface 1: ILocation	22
Interface 2: IPoi	22
Interface 3: INavInstruction	23
Interface 4: ConfigsUpdaterListener	23
14. Enum Types	23
Enum 1: MapRotationType	23
Enum 2: NavigationState	23
Enum 3: ResUpdateStatus	23
Enum 4: LocationMode	23
Enum 5: NavigationType	23

15. Listening to Geofences	23
Step 1: Implement the ZoneDetection interface	23
Step 2: Subscribe to the geofence events and set the geofences names (types)	24
16. Change Log	24

1. Application Setup

Step 1: Add Spreo SDK to you project

Copy spreo-sdk-lib.aar file to the “<project>/<app>/libs/” folder on your project. If The libs folder doesn't exists create it.

Add the following code to the <project>/build.gradle:

```
allprojects {  
    repositories {  
        jcenter() flatDir {  
            dirs 'libs'  
        }  
    }  
}
```

Add the following code to the dependencies on the <project>/app/build.gradle:

```
compile(name:'spreo-sdk-lib', ext:'aar')
```

Step 2: Add Google dependencies

Add the following code to the dependencies on the <project>/app/build.gradle:

```
compile 'com.google.android.gms:play-services-maps:10.2.6'  
compile 'com.google.android.gms:play-services-location:10.2.6'
```

Step 3: Set the required SDK level

Add these lines in your AndroidManifest file:

```
<uses-sdk android:minSdkVersion="18" android:targetSdkVersion="25" />
```

Step 4: Set the required permissions

Add the following lines in your AndroidManifest file:

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Step 5: Handle permissions

Beginning in Android 6.0 permissions are not granted when users install the application. Some of the permissions are now considered as dangerous, so user must accept it at runtime. More information about the new permissions can be found on Google's <https://developer.android.com/training/permissions/requesting.html>.

A sample of handling the permissions request can be found on the attached sample Application.

Step 6: Set your SPREO API key

Add this line to the application element in your AndroidManifest file:

```
<meta-data android:name="Spreo_api_key" android:value="<PUT_YOUR_API_KEY_HERE>" />
```

Step 7: Set your Google API key

Add these lines to the application element in your AndroidManifest file.

```
<meta-data android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
<meta-data android:name="com.google.android.maps.v2.API_KEY"
android:value="<PUT_YOUR_API_KEY_HERE>" />
```

2. Update and Load the Required Data

Important: you have to update the data before you can start working with the SDK.

Step 1: Implement ConfigsUpdaterListener interface

```
public class MainActivity extends Activity implements ConfigsUpdaterListener {public void
onPreConfigsDownload() {}
```

This method will be called when the update starts.

```
public void onPostConfigsDownload(ResUpdateStatus status) {}
```

This method will be called when the update ends.

Step 2: Subscribe to update events and start the update

```
public class MainActivity extends Activity implements ConfigsUpdaterListener, ConfigsLoadListener {
SpreoResourceConfigsUtils.subscribeToResourceUpdateService(this); SpreoResourceConfigsUtils.update(this);
.
.
.
```

Step 3: Subscribe to sdk messages events

SDKMsgsManager manages the messages on the system level

```
public class MainActivity extends Activity implements SDKMsgsListener{
// subscribe to sdk messages events SDKMsgsManager.getInstance().subscribe(this);
.
.
@Override
public void onGPSproviderStatusChanged(boolean isEnabled) {
// check here if the GPS provider is enabled or not
// you need to turn it on.
String msg = (isEnabled == true )? "GPS PROVIDER IS ENABLED":"GPS PROVIDER IS DISABLED";

Toast.makeText(this, msg, Toast.LENGTH_LONG).show();
}
.
.
.
//unsubscribe from messages manager SDKMsgsManager.getInstance().unsubscribe(this);
```

Step 4: When the download ends you can start the location service

```
public void onPostConfigsDownload(ResUpdateStatus status){
    SpreoResourceConfigsUtils.unsubscribeFromResourceUpdateService(this);
    SpreoLocationProvider.getInstance().startLocationService(ctx);
```

Note: for stopping the location service use the following method:
SpreoLocationProvider.getInstance().stopLocationService();

3. Implement the SpreoDualMapView

Step 1: Add a SpreoDualMapView to your XML layout

```
<com.spreo.sdk.view.SpreoDualMapView android:id="@+id/SpreoMap" android:layout_width="wrap_content"
android:layout_height="wrap_content" />
```

Step 2: Add a SpreoDualMapView member in your activity

```
SpreoDualMapView mapView;
```

Step 3: Initiate the SpreoDualMapView

In your activity onCreate method add the following lines:

```
mapView = (SpreoDualMapView) findViewById(R.id.SpreoMap); mapView.onCreate(savedInstanceState);
```

Step 4: Add methods required for Google map functionality

In your Activity override the onResume, onPause, onDestroy, onLowMemory, onSaveInstanceState methods and call the following SpreoDualMapView methods:

```
protected void onResume() { mapView.onResume(); super.onResume();
}
```

```
protected void onDestroy() { mapView.onDestroy(); super.onDestroy();
}
```

```
protected void onPause() { mapView.onPause(); super.onPause();
}
```

```
public void onLowMemory() { mapView.onLowMemory(); super.onLowMemory();
}
```

```
protected void onSaveInstanceState(Bundle outState) { mapView.onSaveInstanceState(outState);
super.onSaveInstanceState(outState);
}
```

4. Enable Application to Receive and React to SpreoDualMapView Events

Step 1: Implement SpreoDualMapViewListener interface

```
void mapDidLoad();
```

This method will be called when the map load finished. **Important** - you should wait for this call before you start interacting with the map view.

```
public void onPoiClick(IPoi poi) {}
```


This method will be called when the map detect a click on a POI.

```
public void onBubbleClick(IPoi poi)
```

This method will be called when the map detect a click on a POI bubble

```
public void onMyParkingMarkerClick()
```

This method will be called when the map detects a click on my parking marker

```
public void onMyParkingBubbleClick()
```

This method will be called when the map detects a click on my parking marker bubble

```
void onUserlocationClick();
```

This method will be called when the map detects a click on the user location icon.

```
void onUserLocationBubbleClick();
```

This method will be called when the map detects a click on the user location bubble.

```
void onLabelClick(ILabel label);
```

This method will be called when the map detects a click on the label.

```
void onMapClick(LatLng latLng, String facilityId, int floor);
```

This method will be called when a click on the map detected.

```
void onMapLongClick(LatLng latLng, String facilityId, int floor);
```

This method will be called when a long click on the map detected.

```
void mapDidLoadFloor(String campusId, String facilityId, int floorId);
```

This method will be called when the map load floor in a specific facility.

```
public void OnFloorChange(int floor) {} This method will be called when the map change floor.
```

Set a custom view for the poi bubble:

```
public View aboutToOpenBubble(IPoi poi) { View view;  
.  
.  
.  
return view;  
}
```

Note: The bubble that is drawn is not a live view. The view is rendered as an image at the time it is returned. This means that any subsequent changes to the view will not be reflected by the bubble on the map. Furthermore, the view will not respect any of the interactivity typical for a normal view such as touch or gesture events.

Step 2: Register to map events

```
public class MapListener implements SpreoDualMapViewListener { mapView.registerMapListener(this);
```

5. Interact with the SpreoDualMapView

Important: you should wait for the "mapDidLoad()" event before you start interacting with the map.

Example 1: Change the map zoom level

```
mapView.mapZoomIn(); mapView.mapZoomOut();
```

Example 2: Present a specific campus

```
mapView.presentCampus(campusId);
```

Example 3: Present a specific facility.

```
mapView.presentFacility(campusId, facilityId);
```

Example 3: Center map to present a specific POI

```
IPoi poi;
```

```
.  
. .
```

```
mapView.showPoi(poi);
```

Example 4: Center map to present the user location

```
mapView.showMyLocation();
```

Example 5: Center map to present a location

```
ILocation location;
```

```
.  
. .
```

```
mapView.presentLocation(location);
```

Example 6: Get the ID of the presented floor

```
String facilityId;
```

```
.  
. .
```

```
int floorId = mapView.getPresentedFloorId(facilityId);
```

Example 7: Get the title of a specific floor

```
int floorId;
```

```
.  
. .
```

```
String title = mapView.getFloorTitleForFloorId(campusId, facilityId, floorId);
```

Example 8: Change the default user icon

```
Bitmap userBitmap;
```

```
.  
. .
```

```
mapView.setUserIcon(userBitmap);
```

Example 9: Redraw pois

```
mapView.reDrawPois();
```

Example 10: Register/Unregister from mapView events

```
mapView.registerListener(this); mapView.unregisterMapListener(this);
```

Example 11: Register/Unregister from navigation events

```
mapView.registerNavigationListener(this); mapView.unregisterNavigationListener(this);
```

Example 12: Open poi bubble

```
IPoi poi;  
.  
.  
.  
mapView.openPoiBubble(poi)
```

Example 13: Close poi bubble

```
IPoi poi;  
.  
.  
.  
mapView.closeBubble(poi)
```

Example 14: Customizing my parking icon

```
Bitmap icon;  
.  
.  
.  
mapView.setMyParkingMarkerIcon(icon);
```

Example 15: Open my parking bubble

```
mapView.openMyParkingMarkerBubble();
```

Example 16: Close my parking bubble

```
mapView.closeMyParkingMarkerBubble();
```

Example 17: Show/hide all pois on map

```
mapView.showAllPois(); mapView.hideAllPois();
```

Example 18: Change icon for a poi

```
IPoi poi = null; Bitmap icon = null;  
.  
.  
.  
mapView.setIconForPoi(poi, icon);
```

Example 19: Change icon for poi list

```
Bitmap poiicon; List<IPoi> poilist;  
.  
.  
.
```

```
mapView.setIconForPoiList(poilist, poiicon);
```

Example 20: Set visible pois list

```
List<String> poildsList;  
.  
.  
.  
mapView.setVisiblePoisWithIds(poildsList);
```

Example 21: Return to default zoom

```
mapView.returnToDefaultZoom();
```

Example 22: Present a specific floor

```
int floorId;  
.  
.  
.  
mapView.showFloorWithId(floorId);
```

Example 23: Present multi point navigation

```
List<IPoi> poilist;  
List<IPoi> visitedPoiList;  
.  
.  
.  
mapView.presentMultiPoiRoute(poilist, visitedPoiList);
```

Example 24: Remove the multi point navigation

```
mapView.removeMultiPoiRoute();
```

Example 25: Set the visible labels

```
List<String> labelIdsList;  
.  
.  
.  
mapView.setVisibleLabelsWithIds(labelIdsList);
```

Example 26: Hide all labels

```
mapView.hideAllLabels();
```

Example 27: Show all labels

```
mapView.showAllLabels();
```

Example 28: Displaying friends locations on the map

You can use methods of the `com.spreo.ui.utils.FriendsManager` class to display friends locations on the map.

To add (or update) friend location on the map you need to call `FriendsManager.updateFriend(String id, Bitmap icon, ILocation location, String name)` method. If you already displaying a `SpreoDualMapView` on the screen you have to call `FriendsManager.updateFriendsOnMap(SpreoDualMapView dualMapView)` after adding all friends location to the `FriendsManager` to update the map. Otherwise friends location will be displayed on the map as soon as `SpreoDualMapView` component would be added to the screen.

You also can use `FriendsManager.updateFriendsOnMap(SpreoDualMapView dualMapView)` `FriendsManager.removeAllFriends()` methods to remove friends from the map and `Friend FriendsManager.getFriend(String id)` methods to get the data associated with previously added friend.

6. Navigation

Example 1: Start navigation

Choose one of the following methods:

Navigate to a specific POI: `IPoi poi`;

.
.
.

`mapView.navigateTo(ipoi);`

Navigate to Location: `ILocation location`;

.
.
.

`mapView.navigateTo(location);`

Example 2: Stop navigation

`mapView.stopNavigation();`

Example 3: Simulate navigation

Choose one of the following methods:

Simulate navigation to specific POI:

`IPoi poi`;

.
.
.

`mapView.simulateNavigationTo(poi);`

Simulate navigation to a `IPoi` from a specific origin location:

`ILocation origin`; `IPoi poi`;

.
.
.

`mapView.simulateNavigationTo(origin, poi);` Simulate navigation to the saved parking location: `ILocation origin`;

.
.
.

`mapView.simulateNavigationToParking(origin);`

Stop simulation:

`mapView.stopSimulation();`

Example 4: Getting total route distance

`IPoi poi`;

.

```

.
.
mapView.navigateTo(ipoi);
double totalRouteDistance = mapView.getRouteDistance();

```

Example 5: Using feet units for distance measurement

If you want to retrieve distance value in feet units just enable this options with the SettingsProvider:
 SettingsProvider.getInstance().setUseFeetForDistance(true);

7. Enable Your Application to Receive and React to Navigation Events

Step 1: Implement the SpreoNavigationListener interface

```

public class NavigationListener implements SpreoNavigationListener {
public void onNavigationStateChanged(NavigationState state) {}

```

This method will be called when the map detects change in the navigation state.

```

public void OnNavigationInstructionChanged(INavInstruction instruction) {}

```

This method will be called when the map detects change in the navigation instructions.

```

public void onNavigationArriveToPoi(IPoi arrivedToPoi, List<IPoi> nextPois) {}

```

This method will be called when the user arrive to a poi.

```

public void onInstructionRangeEntered(INavInstruction instruction)

```

This method is deprecated.

Step 2: Register to the navigation events

```

mapView.registerNavigationListener(this);

```

Step 3: Use INavInstruction object from OnNavigationInstructionChanged callback to get the instruction details

```

public interface INavInstruction {

```

```

// instruction text
String getText();

```

```

// instruction icon
Bitmap getSignBitmap();

```

```

// distance to next navigation instruction
double getDistance();
}

```

Example 1: Display distance to next navigation instruction

```

public class NavigationListener implements SpreoNavigationListener {

```

```

public void OnNavigationInstructionChanged(INavInstruction instruction {

```

```

// when the instruction changed update the custom layout with the new instruction

```

```

if (instruction != null) {

```

```

String txt = instruction.getText();

```

```

if (txt != null) {

```

```

// output the distance to next navigation instruction with instruction text txt += String.format(", distance: %.2f
meter(s)", instruction.getDistance());insTextView.setText(txt);

```

```
}  
}  
}  
}
```

8. Save and Navigate to Parking Location

Example 1: Save your current location as parking location

```
mapView.setCurrentLocationAsParking();
```

Example 2: Save a location as parking location

```
ILocation location;  
.  
.  
.  
mapView.setLocationAsParking(location);
```

Example 3: Remove parking location

```
mapView.removeParkingLocation();
```

Example 4: Check if there is a saved parking location

```
boolean hasparkinglocation = mapView.hasParkingLocation();
```

Example 5: Get the saved parking location

```
ILocation location = mapView.getParkingLocation();
```

Example 6: Navigate to the saved parking location

```
mapView.navigateToParking();
```

9. Enable your Application to Receive and React to Real Time Location

Step 1: Implement MyLocationListener interface

```
public class LocationListener implements MyLocationListener {  
public void onLocationDelivered(ILocation location) {}
```

This method will be called each time device location changes.

```
public void onCampusRegionEntrance(String campusId) {}
```

This method will be called each time the user gets into the campus.

```
Public void onFacilityRegionEntrance(String campusId, String facilityId) {}
```

This method will be called each time the user gets into a facility.

```
Public void onFacilityRegionExit(String campusId, String facilityId) {}
```

This method will be called each time the user exits a facility.

```
Public void onFloorChange(String campusId, String facilityId, int floor) {}
```

This method will be called each time the user change floor.

Step 2: Subscribe to location events

public class LocationListener **implements** MyLocationListener {
SpreoLocationProvider.getInstance().subscribeForLocation(this); Notes:

- For unsubscribing from location events do the following:
SpreoLocationProvider.getInstance().unSubscribeFromLocationService(this);
- For checking if the location service is tracking user location use the following method:
- boolean istracking = SpreoLocationProvider.getInstance().isTrackingUserLocation() Returns true if is tracking user location. Otherwise, returns false.
- If you want to get the user location use the following method ILocation =
SpreoLocationProvider.getInstance().getUserLocation(); Returns ILocation object containing the location info of the user

10. Working with the POI list

Example 1: Get the POI list for a specific campus

```
String campusId;  
.  
.  
.  
List<IPoi> pois = PoisUtils.getAllCampusPoisList(campusId);
```

Example 2: Get the POI list for a specific facility

```
String campusId;  
String facilityId;  
.  
.  
.  
PoisUtils.getAllFacilityPoisList(campusId, facilityId);
```

Example 3: Get the POI list for a specific floor

```
String campusId;  
String facilityId;  
String floorId;  
.  
.  
.  
List<IPoi> pois = PoisUtils.getAllFloorPoisList(campusId, facilityId, floorId);
```

Example 4: Get the POI categories list

```
List<PoiCategory> cats = PoisUtils.getPoiCategories();
```

Example 5: Set the list of the visible categories on map

```
List<PoiCategory> categories;  
.  
.  
.  
PoisUtils.setPoiCategoriesVisible(categories);
```


Example 6: Get POIs sorted alphabetically

```
String campusID = SpreoDataProvider.getCampusId();
List<IPoi> pois = PoisUtils.getAllCampusPoisList(campusID);
List<IPoi> sortedList = SortingPoiUtil.getPoisSortedAlphabetical(pois);
```

Example 7: Get POIs sorted by distance from location

```
String campusID = SpreoDataProvider.getCampusId();
List<IPoi> pois = PoisUtils.getAllCampusPoisList(campusID);
List<IPoi> sortedList = SortingPoiUtil.getPoisSortedByLocation(pois, myLocation);
```

Example 8: Get POIs with category

```
String campusID = SpreoDataProvider.getCampusId();
List<IPoi> pois = PoisUtils.getAllCampusPoisList(campusID);
List<String> categoriesList = Arrays.asList("services", "workstations"); List<IPoi> sortedList =
PoisUtils.getPoisWithCategories(pois, categoriesList);
```

Example 9: Get POIs sorted alphabetically with categories

```
String campusID = SpreoDataProvider.getCampusId();
List<IPoi> pois = PoisUtils.getAllCampusPoisList(campusID);
List<IPoi> sortedList = SortingPoiUtil.getPoisSortedAlphabetical(pois);
List<String> categoriesList = Arrays.asList("services", "workstations");
List<IPoi> filteredList = PoisUtils.getPoisWithCategories(sortedList, categoriesList);
```

Example 10: Get POIs sorted by distance from location with categories

```
String campusID = SpreoDataProvider.getCampusId();
List<IPoi> pois = PoisUtils.getAllCampusPoisList(campusID);
List<IPoi> sortedList = SortingPoiUtil.getPoisSortedByLocation(pois, myLocation); List<String> categoriesList =
Arrays.asList("services", "workstations");
List<IPoi> filteredList = PoisUtils.getPoisWithCategories(sortedList, categoriesList)
```

Example 11: Getting distance to POI or to the list of POIs

```
//getting distance to single POI IPoi poi = ...
double distance = PoisUtils.getDistanceToPoi(outdoor, poi);
//getting distance to the list of POIs ILocation location = ...
List<IPoi> poisList = ...
List<PoiDistance> poiDistances
= PoisUtils.getDistanceToPoiList(location, poisList, true); PoiDistance distance = poiDistances.get(0);
System.out.println("distance to"+ distance.poi + ": " + distance.getDistance());
```

11. Change Settings

Example 1: Set the type of Google map

```
int mapType = GoogleMap.MAP_TYPE_NORMAL; (use the Google map types)
SettingsProvider.getInstance().setMapType(mapType);
```

Example 2: Set the navigation route color

```
String routeHexColor = "#701E84"; SettingsProvider.getInstance().setRouteColor(routeHexColor);
```

Example 3: Get the navigation route color

```
String routeHexColor = SettingsProvider.getInstance().getRouteColor();
```

Example 4: Set the visibility of the navigation instructions

```
boolean display;  
.  
.  
.  
SettingsProvider.getInstance().setDisplayNavigationInstructionsOnMap(display);
```

Example 5: Get the visibility of the navigation instructions

```
SettingsProvider.getInstance().isDisplayNavigationInstructionsOnMap();
```

Example 6: Set the status of the instructions sound

```
boolean mute;  
.  
.  
.  
SettingsProvider.getInstance().setNavigationInstructionsSoundMute(mute);
```

Example 7: Get the status of the instructions sound

```
boolean mute = SettingsProvider.getInstance().isNavigationInstructionsSoundMute();
```

Example 8: Set simplified instruction

```
Boolean isSimplified;  
.  
.  
SettingsProvider.getInstance().setSimplifiedInstruction(isSimplified);
```

Example 9: Set the interval for the map auto follow mode

Set the interval for the map auto follow mode:

Sets user auto follow time interval (in milliseconds). After the specified time interval the state of auto follow will be reset (interval from the last map interaction). Set this value to -1 if you want to disable follow me mode.

```
long interval; (interval in Milliseconds)  
.  
.  
.  
SettingsProvider.getInstance().setUserAutoFollowTimeInterval(interval);
```

Example 10: Set the map rotation type

```
MapRotationType rotationType;  
.  
.  
.  
SettingsProvider.getInstance().setMapRotation(rotationType);
```

Example 11: Get the map rotation type

```
MapRotationType rotationType = SettingsProvider.getInstance().getMapRotationType();
```

Example 12: Set the map navigation type

The NavigationType is related to three states:

- ESCALATORS in which the navigation algorithm prefer to choose escalator in navigation route path.
- DISABLED in which the navigation algorithm prefer to choose elevators in navigation route path.
- DEFAULT no preference to be chosen when navigation route path is built.

```
NavigationType navigationType= NavigationType .DEFAULT ;  
SettingsProvider.getInstance().setNavigationType(navigationType);
```

Example 13: Set whether to include the parking in the multiPois or not

```
boolean include;  
.  
SettingsProvider.getInstance().setIncludeParkingToMultiPois(include);
```

Example 14: Set whether to include entrances in the multiPois or not

```
boolean include;  
.  
.  
SettingsProvider.getInstance().setIncludeEntrancesToMultiPois(include);
```

Example 15: Set whether to include switch floors in the multiPois or not

```
boolean include;  
.  
.  
SettingsProvider.getInstance().setIncludeSwitchFloorsToMultiPois(include);
```

Example 16: Set the ui options for multi point navigation

```
MultiPoiUIOptions options;  
.  
.  
SettingsProvider.getInstance().setMultiPoiUIOptions(options);
```

Example 17: Set the dual map default zoom

```
float zoom;  
.  
SettingsProvider.getInstance().setDualMapDefaultZoom(zoom);
```

Example 18: Set the visibility of the floor picker

```
boolean visible;  
.  
.  
.
```

```
SettingsProvider.getInstance().setFloorPickerVisibility(visible);
```

Example 19: Set feet as measured units for the distance to next navigation instruction

```
SettingsProvider.getInstance().setUseFeetForDistance(true)
```

Example 20: Hide content of the top floor

You can choose which of the floors of your buildings would be shown on the map (see Example 22 of the Section 5). The buildings can have a different number of floors. When you choose a floor higher than the last floor of some buildings, we show the contents of the last floor for these buildings. You can configure your project to hide the contents of the last floor in this case:

```
SettingsProvider.getInstance().setDisplayTopFloorContent(false)
```

Example 21: Set up location analytics

```
//Configure SDK to report location every 20 seconds to server for location analytics
SettingsProvider.getInstance().setAnalyticsInterval(20000);
SettingsProvider.getInstance().setUserAnalyticsEnable(true);
```

You can limit location reports using `SettingsProvider.getInstance().setLocationReportLimitation(reportLimitation)` method. Call it with `ReportLimitation.INDOOR` to report only indoor locations, use `ReportLimitation.IN_CAMPUS` to report all location which are inside campus area or `ReportLimitation.ALWAYS` to send all locations. Default value is `ReportLimitation.IN_CAMPUS`.

12. Data Provider

Example 1: Get the SDK version

```
String version = SpreoDataProvider.getSdkVersion();
```

Example 2: Get the selected campus ID

```
String campusid = SpreoDataProvider.getCampusId();
```

Example 3: Get the campuses list

```
List<String> campuseslist = SpreoDataProvider.getCampusesList();
```

Example 4: Get the facilities list for a specific campus

```
String campusId;
.
.
.
List<String> facilitieslist = SpreoDataProvider.getCampusFacilities(campusId);
```

Example 5: Get the title of a specific floor

```
String campusId;
.
.
.
String facilityId;
.
.
```

.
int floorIndex;

.
.
.

String title = SpreoDataProvider.getFloorTitle(campusId, facilityId, floorIndex);

Example 6: Get info for a specific campus

String campusId;

.
.
.

HashMap<String, Object> campusInfo = SpreoDataProvider.getCampusInfo(campusId);

The method returns a map containing the following key/value pairs:

key	value / type
id	id of campus (String)
name	name of campus (String)
location	center location in lat/lon (LatLng)

Example 7: Get info for a specific facility

String campusId;

.
.
.

String facilityId;

.
.
.

HashMap<String, Object> facilityInfo = SpreoDataProvider.getFacilityInfo(campusId, facilityId);

The method returns a map containing the following key/value pairs:

key	value / type
id	id of facility (String)
name	name of facility (String)
location	center location center location in lat/lon (LatLng)
floors	list of floors numbers (List<Integer>)

Example 8: Get info for a specific floor

String campusId;

.
.
.

```
String facilityId;
.
.
.
int floorIndex;
.
.
.
HashMap<String, Object> floorInfo = SpreoDataProvider.getFloorInfo(campusId, facilityId, floorIndex);
```

The method returns a map containing the following key/value pairs:

key	value / type
title	title of floor (String)

Example 9: Get the facility of the floor picker (the facility with highest number of floors)

```
String facilityId = SpreoDataProvider.getFloorPickerFacilityId();
```

13. Interfaces

Interface 1: ILocation

```
JSONObject getAsJson(); JSONObject getAsJson();
void parse(JSONObject jsonobject); double getX();
double getY(); double getZ(); double getLat(); double getLon();
LocationMode getLocationType(); String getCampusId();
String getFacilityId(); void setLat(double lat); void setLon(double lon);
void setType(LocationMode locationMode); void setCampusId(String campusId);
void setFacilityId(String facilityId); void setX(double x);
void setY(double y); void setZ(double z);
```

Interface 2: IPoi

```
List<String> getPoiKeywords();
void setPoiKeywords(List<String> poiKeywords); String getPoiuri();
void setPoiuri(String poiuri); List<String> getPoitype();
void setPoitype(String poitype); String getpoiDescription();
void setpoiDescription(String description); PointF getPoint();
void setPoint(PointF point); double getZ();
void setZ(double z); String getUrl();
void setUrl(String url); String getDetails();
void setDetails(String details); boolean isShowPoiOnMap();
void setShowPoiOnMap(boolean showPoiOnMap); boolean isShowPoiOnSearches();
void setShowPoiOnSearches(boolean showPoiOnSearches); boolean isShowPoiBubble();
void setShowPoiBubble(boolean showPoiBubble); boolean isPoiPlayMultyMedia();
void setPoiPlayMultyMedia(boolean poiPlayMultyMedia); String getPoiID();
void setPoiID(String poiID); String getPoiNavigationType();
void setPoiNavigationType(String poiNavigationType); double getPoiLatitude();
void setPoiLatitude(double poiLatitude); double getPoiLongitude();
void setPoiLongitude(double poiLongitude); boolean isShowOnZoomLevel();
void setShowOnZoomLevel(boolean showOnZoomLevel); float getX();
```

```

float getY();
boolean isPoiClickable();
void setPoiClickable(boolean poiClickable); boolean isInstructionsParticipate();
void setInstructionsParticipate(boolean instructionsParticipate); String getPoiofficeinstructions();
void setPoiofficeinstructions(String poiofficeinstructions); boolean isPoishowincategory();
void setPoishowincategory(boolean poishowincategory);
String getMediaurl();
void setMediaurl(String mediaurl); String getKeyWordsAsString(); ILocation getLocation();
void setLocation(ILocation location); List<String> getActivehours();
void setActivehours(List<String> activehours); List<String> getPhone2hours();
void setPhone2hours(List<String> Phone2hours); List<String> getPhone1();
void setPhone1(List<String> Phone1); List<String> getPhone2();
void setPhone2(List<String> Phone2); String getEmailaddress();
void setEmailaddress(String Emailaddress); Bitmap getIcon();
void setIcon(Bitmap icon);

```

Interface 3: INavInstruction

```

String getText(); Bitmap getSignBitmap(); int getId();

```

Interface 4: ConfigsUpdaterListener

```

public void onPreConfigsDownload();
public void onPostConfigsDownload(ResUpdateStatus status)

```

14. Enum Types

Enum 1: MapRotationType

```

STATIC, COMPASS, ORIENTATION

```

Enum 2: NavigationState

```

IDLE, STARTED, NAVIGATE,
FLOOR_CHANGED, TURNUED_BACK, REROUTE, SILENT_REROUTE, STOPED,
PENDING;

```

Enum 3: ResUpdateStatus

```

API_KEY_VALIDATION_FAILED, API_KEY_ERROR_TYPE_CONNECTION , API_KEY_ERROR_TYPE_INVALID_APIKEY,
API_KEY_NO_RESPONSE,
OK, CAMPUSES_JSON_FAIL, CAMPUSES_JSON_OK, FAILED

```

Enum 4: LocationMode

```

INDOOR_MODE, OUTDOOR_MODE;

```

Enum 5: NavigationType

```

DEFAULT, ESCALATORS, DISABLED;

```

15. Listening to Geofences

Step 1: Implement the ZoneDetection interface

```

public class MapActivity extends Activity implements ZoneDetection {

```

void onZoneEnter(GeoFenceObject zone);
this method will be called when a zone enter detected.

void onZoneExit(GeoFenceObject zone);
this method will be called when a zone exit detected.

void setListeningTo(List<String> to); deprecated
List<String> getListeningTo();deprecated

Step 2: Subscribe to the geofence events and set the geofences names (types)

```
List<String> geoFenceingListenToList = new ArrayList<String>(); geoFenceingListenToList.add("trigger");  
GeoFencingUtils.subscribeToService(this, geoFenceingListenToList);
```

* add all the types of the geofences that you want to listen to

16. Change Log

Version 4.7

- New API to download data ZIP without maps. Should be used only if the project is using url tiles.
SettingsProvider.getInstance().setUseZipWithoutMaps(true);
- Support for drawing polygons on the map.
- Improved algorithm for parking Navigation.
- Improved support for bridges navigation.
- Internal improvements and bug fixes.

Version 4.5

- New labels mechanism.
- Support for dashed navigation route.

Version 4.4

- Map rotation logic for MapRotationType.NAVIGATION had been reworked.
- Fixes for POI icons sizes, now icons on all android devices have physical dimensions similar to icons dimensions for iOS version of the SDK.
- GPS usage optimizations for indoor mode, reducing battery consumption for this mode
- Some extra data has been removed from logcat output
- Minor fix for instructions text
- Internal bug fixes and improvements

Version 4.3

- Switching to new version of google play libraries (15.0.0). Starting from this version we strongly recommend to use this specific version of google play libraries on the app level. Check "sdk lib/adding .arr dependency.docx" file for details.
- Adding "onLocationModeChange(LocationMode mode)" callback to MyLocationListener. This can break client app code compilation. You can easily fix it by adding empty "onLocationModeChange" callback to all MyLocationListener implementations.
- Internal bug fixes and improvements

Version 4.2

- Internal bug fixes and improvements
- New api to get instructions list
- Navigation line thickness was increased to make look & feel similar to iOS version of SDK

Version 4.1

- Improvements in location analytics implementation
- Ability to limit location analytics to some location types

- Internal bug fixes

Version 4.0

Adding API for displaying friends on Map

Version 3.9

- Storage location for internal data had been changed. Now system will remove all sdk internal data when uninstalling client application from device
- Internal bug fixes and improvements

Version 3.8

Adding ability to hide the contents of the top floor

Version 3.7

Internal bug fixes and improvements

Version 3.6

Minor bug fixes and improvements

Version 3.5

- Added support for bridges between facilities
- Added ability to switch to another project in run-time

Version 3.4

- Added API for calculating distance to POI / list of POIs
- 3D buildings feature was disabled for underlying Google Map

Version 3.3

- Added API to retrieve full distance for the route
- Added API to retrieve the distance to next navigation instruction
- Added API to retrieve POIs which are belong to specific categories

Version 3.2

- Improvements in sorting POIs by distance approach
- Added ability to hide/show navigation markers on the map
- Added ability to get camera position from the map
- Added ability to animate map camera to specific position (including tilt)
- Minor fixes for map UI and behaviour
- Minor stability improvements

Version 3.1

- Now supporting floor/outdoor tiles
- Route building performance improvements
- Additional indoor location mechanisms based on proximity beacons
- Minor fixes related to location simulation functionality

Version 3.0

Fix labels issue