Becomap SDK Documentation

Overview

The Becomap SDK provides interactive indoor mapping capabilities for Android apps using a WebView. It supports map rendering, floor and location selection, route calculation, and two-way communication with JavaScript.

Initialization

Initialize Map:  
Becomap becomap = new Becomap(context);  
becomap.initializeMap(container, clientId, clientSecret, siteIdentifier);

Map Functions

searchLocation(String value): Searches for locations matching the given value.

selectFloor(String floorId): Selects a specific floor on the map.

getFloors(): Retrieves all floors in the current building.

selectLocationWithId(String locationid): Selects a location by its ID.

getlocation(): Gets all locations in the current map.

Route Management

getroute(String Startid, String toid, List<String> waypoints): Calculates a route between locations.

showroute(): Displays the calculated route on the map.

clearallroutes(): Clears all routes from the map.

showStep(int stepIndex): Highlights a specific step in the current route.

JavaScript Integration

injectGetSiteIdFunction(): Injects JS to retrieve the site ID.

injectGetSiteNameFunction(): Injects JS to retrieve the site name.

Information Retrieval

GetDefaultFloor(): Gets the building's default floor.

GetCurrentFloor(): Gets the currently selected floor.

GetLanguages(): Gets the available languages.

GetCategories(): Gets the available categories.

GetAllAmenities(): Gets all amenities in the map.

GetAmenities(): Gets specific amenities.

GetSessionId(): Gets the current session ID.

GetQuestions(): Gets survey questions.

GetHappenings(String type): Gets happenings or events by type.

Callback Registration

setCallback(BecomapCallback callback): Sets the callback listener to receive SDK events.

Callback Interface

Implement Becomap.BecomapCallback to handle SDK events:

Sample:  
  
becomap.setCallback(new Becomap.BecomapCallback() {  
 @Override  
 public void onMapRenderComplete() {  
 // Map is ready  
 }  
  
 @Override  
 public void onLocationsReceived(List<LocationModel> locations) {  
 // Locations received  
 }  
});

Available Callback Methods:  
- onMapRenderComplete()  
- onLocationsReceived(List<LocationModel>)  
- onSearchResultsReceived(List<SearchResult>)  
- onSiteIdAvailable(String siteId)  
- onSiteNameAvailable(String siteName)  
- onBuildingsReceived(List<BuildingModel>)  
- onDefaultFloorReceived(FloorModel)  
- onLanguagesReceived(List<LanguageModel>)  
- onCurrentFloorReceived(FloorModel)  
- onCategoriesReceived(List<Category>)  
- onAllAmenitiesReceived(List<LocationModel>)  
- onAmenityTypesReceived(List<String>)  
- onSurveyQuestionsReceived(List<BCQuestion>)  
- onSessionIdReceived(String sessionId)  
- onFloors\_Received(List<FloorModel>)  
- ongetroute(List<Route>)

WebView Lifecycle Handling

Override lifecycle methods and pass events to the SDK:  
  
@Override  
protected void onStart() {  
 super.onStart();  
 becomap.onStart();  
}  
  
@Override  
protected void onResume() {  
 super.onResume();  
 becomap.onResume();  
}  
  
@Override  
protected void onPause() {  
 super.onPause();  
 becomap.onPause();  
}  
  
@Override  
protected void onStop() {  
 super.onStop();  
 becomap.onStop();  
}  
  
@Override  
protected void onDestroy() {  
 super.onDestroy();  
 becomap.onDestroy();  
}

Conclusion

The Becomap SDK is a powerful tool for embedding indoor navigation, location search, and map interactivity into Android apps using WebView and JavaScript communication. With a comprehensive API and event-driven architecture, it enables full control over the user’s map experience.