I'm not robot

reCAPTCHA

Continue

I'm not robot

reCAPTCHA

Continue

Android mapping sdk

Android sdk version mapping.

ATTENTION: Maps API V2.0 will be deprecated and replaced by Maps API V2.5, 26 October, 2020. For native job maps on the Tablet Fire Fire Fire OS 5+, you need to recompile your application using L 'API V2 Maps 0.5 and send the application again before October 26, 2020. Find out more here: Deprecation Notice for Amazon Maps SDK V1.0 and V2.0. The following steps show how to start with the Amazon Maps API. Tip: see the champion app with Amazon Maps API for an example of implementation. Step 1: Set the development environment Install and configure Android Studio on the computer. If you already have Android Studio installed, the upgrade to the latest version (4.x) going to Android Studio> Check updates. Android Studio is not necessary to use the Google Maps Amazon API, but if you are using another tool, make sure you have the Android SDK installed and other system requirements needed for Android development. Also note that the instructions contained in this documentation will take the use of Android Studio. Step 2: Set a fire tablet for the test Follow the procedure described in Tablet Fire Connection via ADB to connect the Tablet Focus to the computer. See Tablets Focus Support Google Maps V2.5 APIs For more details on which fire tablets that you can use for the test. Step 3: Download the SDK Library Mount Amazon Maps to use the Amazon SDK Maps in the project, you need to download the SDK and add it to the project. The Library is a gradle compatible project. To get the SDK Maps, go to the SDK Download, download the version of Maps Amazon SDK 2.5, and extract the files. For comfort, you can also download the SDK: Step 4: Add the AAR file to the project Adds the AAR file (Amazon-Maps-API-V2.Aar) to the Android project similar to how you should add more project libraries Of an Android Â ¢ By adding the file to an AAR bookcase folder. In Android Studio, move on to the project view. Create a directory called LIBS to the root-level level (unless you already have a library folder). To create this folder, right mouse button in the root directory and select New> Directory. Call it libs. Copy and paste the Amazon-Maps-API-V2.Aar file to this Libs directory. (You can do it directly in the Android studio user interface.) Click OK to confirm the Ready Copy dialog. In the Build.gradle file of your application, refer to the AAR file in the dependency object: dependencies {Implementation: 'Amazon-Maps-API-V2', Ext: 'AAR')} in the same file build.gradle, Add the following to the repository: repository {Google () JCenter () {Flatdir 'Libraries' Dirs}} Note that the SDK also includes a Maven POM file (Amazon-Maps-API-V2.POM) in case you want Upload the SDK from a local Maven repository. The SDK is not available in a public archive Maven or Jcenter. The library loading instructions from a local Maven repository are beyond the scope of this documentation. Step 5: Configure the Android manifest Use for Maps Amazon API functions, you need to update the app permissions listed in the AndroidManifest.xml file. At least, the following permissions are needed: Step 6: Check Amazon Maps Availability device to ensure a good user experience, your application should check Amazon's Maps APIs on the fire tablet and adjust accordingly, if not available. To check for Amazon maps, call amazonmapsruntimeutil.isamazonmapsruntimeeavailable (). This method returns a result code as an int. Use the constants provided in the ConnectionResult object to determine the result. If Maps Amazon is present, returns ConnectionResult.Success. Note: AmazonMapsruntimeutil and ConnectionResult classes belong to the package. For example, the following method returns a Boolean value indicating whether the Google Maps API is available Amazon: import com.amazon.geo.mapsv2.util.AmazonMapsRuntimeUtil; import com.amazon.geo.mapsv2.util.ConnectionResult; ... checkForAmazonMaps private boolean â â () {// Check for Amazon maps on the device back AmazonMapsRuntimeUtil .isAmazonMapsRuntimeAvailable (this) == ConnectionResult.SUCCESS; } The following example shows a onCreate method of activity that writes a message to the log if the Google Maps API Amazon is not available on the device: @ Override protected void onCreate (Bundle savedInstanceState) {super.onCreate (savedInstanceState) ; setcontentview (r.layout.Activity_main); if (checkForAmazonMaps ()) {// maps should be available. // code to get a reference to the map and proceed normally goes here // ...} else {// Maps is not available. Getting the exact result // code and write to a log message int cr = AmazonMapsRuntimeUtil.isAmazonMapsRuntimeAvailable (this); String msg = getString (R.string.no_maps) + "ConnectionResult =" + cr; Log.w (TAG, msg); }} For an example of an error window when viewing the Google Maps API Amazon is not available on the device, see the example app CapitolHillCoffee. Full MainActivity.java Example Here is an example for MainActivity com.mytest.maps app: com.mytest.maps package; Import androidx.appcompat.app.AppCompatatività; IMPORT Android.os.bundle; import com.amazon.geo.mapsv2.util.AmazonMapsRuntimeUtil; import com.amazon.geo.mapsv2.util.ConnectionResult; Maineattività of the public class extends AppCompatatività {@override protected void oncreate (SaviDinstancestate bundle) {super.oncreate (SaviDinstancestate); setcontentview (r.layout.Activity_main); if (checkForAmazonMaps ()) {// maps should be available. // code to get a reference to the map and proceed normally goes here // ...} else {// Maps is not available. Getting the exact result // code and write to a log message int cr = AmazonMapsRuntimeUtil.isAmazonMapsRuntimeAvailable (this); String msg = "ConnectionResult =" + cr; }} Private boolean checkForAmazonMaps â â () {// Check for Amazon maps on the device back AmazonMapsRuntimeUtil .isAmazonMapsRuntimeAvailable (this) == ConnectionResult.SUCCESS; }} Step 7: Recover MD5 Signature App from Android Studio Each application must have its own registration. Debug and release builds requires separate registration. To load map tiles during testing, you need to record the debug build of your application. Registration for the release build is a separate process and occurs automatically when you send your application if it has already created a debug recording. With your registration request, you must provide the MD5 fingerprint debugging. Note: Registration key remains the same for v2.0 and v2.5. Only users who are in v1.0 now need to migrate their registration. To get your MD5 signature: Open the project in Android Studio. Gradle Click the tab in the upper right corner. Expand the entry for your app, then expand the activities, then Android, and double-click signingReport. Android Studio generates a report on the status area at the bottom, where MD5, SHA1, SHA-256, and information. Copy the MD5 value and store this information for future reference. Note: The key store is configured in Android Studio must be different from the one used to sign the APK statement released in the Amazon Appstore. Using the above process only for testing purposes. If you do not use Android Studio, you can extract the MD5 using the utility keytool from the JDK. Step 8: Add your app in your Developer Console If you have not Your app in the console for developers, see Submit Apps for the Amazon AppStore. Step 9: Send your debug Debug registration recording request Allows access to map tiles and geocoding in the debug (s) build of your application. To send a debug registration request for applications that use the Amazon Maps API: on the developer portal, open the list of your applications. Click the application you want to register. (If your yours It's not listed, first add the app to the portal.) Click App Services in the left sidebar. Click Maps. Click Add a debug registration. In the Package Name box, enter the package name for your app. In the Signature Developer box, insert the MD5 fingerprint of the debug certificate you got previously. Click Save. See a reply that says: "The application with the name of the" Com.tomjoht.amazon.Capitolohillcoffee "package was recorded. Note that the portions of the Amazon Maps API are provided by North America, LLC or its affiliates ("Here"). By clicking "Save" to create a debug registration, you accept the terms and conditions of the materials here. It takes time to propagate registration to your device. After clicking Send, allow a ' Now before testing the app you registered (registration the process interacts with an external service such as IT processes). If a recently recorded app does not show the map tiles, you may need to allow more time for recording to have effect. Note: Although it is necessary to first create a debug registration, the version recording is automatic when you send your app. Register a debug build of your app to access the map cards. You must register yours Debug apps before loading the release app. Release registration is a separate process and a separate process occurs and a separate process occurs at the atically when sending your app to Amazon AppStore, if you have already created a debug recording recording 10: Upload your app on a Test device After registering your app, connects the device to a development computer for the test and debugging and debugging build your app. For information on using ADB, see Connect to Fire Tablet via ADB. The next steps see the example app with Amazon Maps API (V2.5). With SDK maps for Android, add maps to your Android app, including OS wear apps using Google Maps data, map views and map gesture answers. You can also provide additional information for map locations and support user interaction by adding markers, polygons and map overlaps. The SDK supports both Kotlin and Java programming languages and provides additional libraries and extensions for advanced functions and programming techniques. What is the next configuration and create your first app: QuickStart - Adding a map Create an Android base app that displays a map. Get an API key and enable billing details on billing enabled and get an API key, both required to add a map to your app. Setting an existing Android Studio project Configure an existing app to use SDK maps for Android. Use and billing Prices and use limits for the SDK. Terms of Service The Google Maps service platform of the service discusses legal requirements that your app must comply when using the SDK. Except that otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 license, and the code samples are licensed with the Apache 2.0 license. For details, consult Google's developer site policies. Java is a registered trademark of Oracle and / or its affiliates. Last update 2021-09-09 UTC. [{"TYPE": "thumb-down", "id": "MissingTheInformationINEED", "Label": "The information I need"} is missing, {"type": "thumb-down", "ID": "TOCOMPLICEDTOOMANYSTAILS", "Label": "Too complicated / too many passages"}, {"Type": "thumb-down", "ID": "OUTOFDATE", "Label": "Out of date"}, {"Type ":" Thumb - down "," id ": "Label": "Emission of samples / code"}, {"Type": "thumb-down", "ID": "OtherDown", "Label": "Other"}}][{"Type": "thumb- Up ", "ID ":" EasyToundStand "," label ":" Easy to understand "}, {" Type ":" thumb-up "," ID ":" SolvedmyProblem "," Label ": Solved my problem "}, {" Type ":" thumb-up "," ID ": otherp "," label ": " Other "}}}]' Other "}}}]