Get latitude and longitude android studio example

Using the API of Google Play services, your app can request the last known location of the user's device. In most cases, you're interested in the user's current location, which is usually equivalent to the last known location of the device. Specifically, use a fused location provider to obtain the last known location of the device. The fused location provider is one of the API sites on Google Play. It manages location technology and provides a simple API so that you can specify high-level requirements, such as high accuracy or low power. It also optimizes the use of the battery device. Note: When your app is running in the background, access to the location should be critical to the core functionality of the app and be accompanied by proper disclosure to users. This lesson shows how to make a single request for the location of the device using the getLastLocation method in the fused location provider. Setting up Google Play services to access a fused location provider, and your app's development project should include Google Play services. Download and install the Google Play service component through SDK Manager and add a library to your project. See the Google Play customization guide for details. Provide permissions for Apps apps whose features are used by location services should request location permissions, depending on when you use these features. Create a location service customer in the onCreate method of your activity, create a customer instance of a fused location vendor, as shown in the following piece of code. private lateinit var fusedLocationClient: FusedLocationProviderClient override pleasure onCreate (savedInast State: Bundle?) { // ... FusionLocationClient - LocationServices.getFusedLocationProviderClient (it) - private FusedLocationProviderClient FusionLocationClent; // .. @Override protected void onCreate (Bundle saved Byvodstate) - / ... FusionLocationClent - LocationServices.getFusedLocationProviderClient Get the last known location After creating a Location Services client you can get the last known location of the user's device. When you connect your application to them, you can use the fused data provider's getLastLocation method to get the device's location. The accuracy of the location returned by this call is determined by the resolution parameter you place in the application manifest, as described in the guide on how to request location permission. To request the last known location, call getLastLocation(). The following piece of code illustrates the query and simple response: fusedLocationClient.lastLocation .addOnSuccessListener : Location? - / Got the last known location. In some rare situations, this may be invalid. fusionLocationClient.getLastLocation () .addOnSuccessListener (this is the new OnSuccessListener)- @Override public emptiness of the public emptiness of the lt. location) - / Got the last known location. In some rare situations, this may be invalid. If (location! null) / Logic for processing the location object The getLastLocation method returns a task that you can use to get a location object with latitude and longitude coordinates of a geographic location. Location can be zero in the following situations: Location is off in device settings. The result can be zero, even if the last location was previously recovered, because the location of the outage also clears the cache. The device has never recorded its location, which may be the case with a new device or device that has been restored in factory settings. Google Play services on the device have restarted, and there is no active fused location provider customer who requested the location after the services rebooted. To avoid this situation, you can create a new client and request a location update yourself. For more information, see Support the current best estimate, you can expect the Location object contained in the last call to be the most accurate. However, because the accuracy of the location varies, the latter value is not necessarily the best. You should include the logic of choosing a location based on several criteria. The set of criteria also varies depending on the use of the application and the results of field tests. To check the accuracy of the location that returns from getLastLocation, follow the following steps, which include: Check to see if the location you received is much newer than the previously received location. Check whether the accuracy claimed by the location is better or worse than the previous estimate. Check the vendor associated with the new location. Decide if you trust that provider more than the one used in your app's cache. Type: thumb down, id: missingTheInformationINeed, label:No information that I need, type: thumb down, id: tooComplicatedTooManySteps, tag:Too complicated /too many steps . down, id: outOfDate, tag:From date, type: thumb down, id: sampleCodeIssue, tag:Samples/code question, type: thumb down, id: otherDown, tag:Other Type: thumb up, id: easyToUnderstand, label:Easy to understand - type: thumb up, id: solvedMyProblem, label:Solved my problem - type: thumb-up, id: otherUp, label:Other tutorial shows you how to find the current location of the Android device and display the details of the device Follow this tutorial to build an Android app using Maps SDK for SDK locations for Android, and a fused location provider in the Google Play service API. Play. Этот учебник требует Google Play услуг версии 11.2.0 или позже. Получить код клона или скачать Google Maps Android API v2 Образцы репозитория из GitHub. < > Show/Hide the Java code for the map activity. // Copyright 2020 Google LLC // // Licensed under the Apache License, Version 2.0 (the License); // you may not use this file except in compliance with the License. // You may obtain a copy of the License at // // // // Unless required by applicable law or agreed to in writing, software // distributed under the License is distributed on an AS IS BASIS, // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. // See the License for the specific language governing permissions and // limitations under the License. package com.example.currentplacedetailsonmap; import android.content.DialogInterface; import android.content.pm.PackageManager; import android.location.Location; import android.os.Bundle; import android.util.Log; import android.view.Menu; import android.view.MenuItem; import android.view.View; import android.widget.FrameLayout; import android.widget.TextView; import androidx.annotation.NonNull; import androidx.appcompat.app.AlertDialog; import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat; import androidx.core.content.ContextCompat; import com.google.android.gms.location.FusedLocationProviderClient; import com.google.android.gms.location.LocationServices; import com.google.android.gms.maps.CameraUpdateFactory; import com.google.android.gms.maps.GoogleMap; import com.google.android.gms.maps.OnMapReadyCallback; import com.google.android.gms.maps.SupportMapFragment; import com.google.android.gms.maps.model.CameraPosition; импорт com.google.android.gms.maps.model.LatLng; импорт com.google.android.gms.maps.model.Marker; импорт com.google.android.gms.maps.model.MarkerOptions; импортировать com.google.android.libraries.places.api.Places; импорт com.google.android.libraries.places.api.model.Place; импорт com.google.android.libraries.places.api.model.PlaceLikelihood; импортировать com.google.android.libraries.places.api.net.FindCurrentPlaceRequest; импортировать com.google.android.libraries.places.api.net.FindCurrentPlaceResponse; импорт com.google.android.libraries.places.api.net.PlacesClient; импортировать java.util.Arrays; импортировать java.util.List; Действия, отображаемые на карте, показывающие место в текущем местоположении устройства. B/ Публичный класс MapsActivityCurrentPlace расширяет реализацию AppCompatActivity OnMapReadyCallback - частную статичную финальную строку TAG - MapsActivityCurrentPlace.class.getSimpleName(); частную карту GoogleMap; частную камеру CameraPosition; // Точка входа в API Места. частные местаClient placesClient; // Location provider. private FusedLocationProviderClient FusionProviderClient; The default location (Sydney, Australia) and the default zoom for use in location resolution are not available. Private final LatLng defaultLocation - new LatLng (-33.8523341, 151.2106085); Private static final int DEFAULT_ZOOM No 15; Private static final int PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION No 1; private boolean locationPermissionGranted; Geographical location where the device is currently located. That is the last known / location obtained by the supplier of the rafted location. private location lastKnownLocation; Keys to store activity status. private static KEY_CAMERA_POSITION - camera_position; Private static final KEY_LOCATION and location; Used to select your current location. Private static final int M_MAX_ENTRIES No 5; Private string, probably PlaceNames; Private string, probablyPlaceAddresses; A private list is likely Private LatLng, probably PlaceLatLngs; @Override protected void onCreate (Bundle savedInstanceState) - super.onCreate (savedInstanceState); Remove the location and position of the camera from the saved state of the instance. If (savedInstanceState! - null) - the lastMoan - savedInstanceState.getParcelable (KEY_LOCATION); cameraPosition - savedInstanceState.getParcelable (KEY_CAMERA_POSITION); Get an idea of the content R.layout.activity_maps that displays the map. Create PlacesClient Places.initialize (getApplicationContext(, getString (R.string.maps_api_key);; PlacesClient - Places.createClient (it); Build a fused LocationProviderClient. fusedLocationProviderClient - LocationServices.getFusedProviderClient Create a map. SupportMapFragment mapFragment (SupportMapFragment) getSupportFragmentManager () .findFragmentById (R.id.map); mapFragment.getMapAsync Saves the card status when suspended. Issue/@Override protected void onAveInstanceState - if (map! - null) - outState.putParcelable (KEY_CAMERA_POSITION, map.getCamepositionPosition ()); outState.putParcelable (KEY_LOCATION, lastKnownLocation); - super.onSaveInstanceState (outState); - @param menu options. @return Boulan. @Override boolean onCreateOptionsMenu (Menu menu) - getMenuInflater (R.menu.current_place_menu, menu); The return is true. - @param menu item to handle. @return Boulan. Issue/@Override public boolean onOptionsItemSelected (menuItem) - if (item.getItemId)- R.id.option_get_place) - showCurrentPlace (); This callback is triggered when the card is ready to use. Issue/@Override public void onMapReady (map - this.map - map; Use a custom window information information adapter process a few lines of text in the contents of the window / info. this.map.WindowAdapter (new GoogleMap.InfoWindowAdapter) - @Override/ Return zero here, so getInfoContents is called the following @Override. View infoWindow - getLayoutInflater (R.layout.custom_info_contents, (FrameLayout) findViewById (R.id.map), false); TextView title - infoWindow.findViewById (R.id.title); title.setText (marker.getTitle ()); TextView fragment - infoWindow.findViewById (R.id.snippet); snippet.setText (marker.) Prepare the user for a resolution. getLocationPermission(); Turn on the My Location layer and the associated control on the map. RenewalLocation Get the device's current location and set the map position. getDeviceLocation(); Gets the current location of the device and positions the map camera. The issue/private void of getDeviceLocation is to get the best and most recent location of the device, which can be invalid in rare cases when the location is not available. C/ Try if (locationPermissionGranted) locationResult.addOnCompleteListener (this is the new OnCompleteListener) - @Override public void on Complete (@NonNull Task) - if the location is a (task. lastKnownLocation - task.getResult (); if (lastKnownLocation ! - null) - map.moveCamera (CameraUpdateFactory.newLatLng'oom (new LatLng (lastKnownLocation.getLatitude.), lastKnownLocation.getLongitude (DEFAULT_ZOOM); use by default); Log.e (TAG, Exception: %s, task.getException(); map.moveCamera (CameraUpdateFactory .newLatLng'oom (default, DEFAULT_ZOOM))); map.getUiSettings ().setMyLocationButtonEnabled (false); }); - Catch (SecurityException e) - Log.e (Exception: %s, e.getMessage,), e); Pushes the user to permission to use the location of the device. Private Void getLocationPermission () - Request a location permit so we can get the location of the device. The result of the resolution request is handled by a callback, according to The RequestPermissionsResult. The question/ if (ContextCompat.checkSelfPermission (this.getApplicationContext), android. Manifest.permission.ACCESS_FINE_LOCATION) - PackageManager.PERMISSION_GRANTED) - locationPermissionGrant - true; - another - ActivityCompat.requestPermissions (this is a new string. Manifest.permission.ACCESS_FINE_LOCATION, PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION); Processes the result of a location permit request. The question/@Override of the public void on RequestPermissionsResult (int requestCode, qtot;Line permits @NonNull int grantResultans) - locationPermmissionGrant - false; (requestCode) - PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION case: / If the request is canceled, the results arrays are empty PackageManager.PERMISSION_GRANTED qgt. In/private void showsCurentPlace - if (map - invalid) - return; If (locationPermissionGranted) -/ Use fields to determine the types of data for return. PlaceFields and Arrays.asList (Place.Field.NAME, Place.Field.ADDRESS, Place.Field.LAT_LNG); Use a builder to create FindCurrentPlaceRequest. FindCurrentPlaceRequest - FindCurrentPlaceRequest.newInstance Get probable locations - that is, businesses and other points of interest that are the best for the current location of the device. @SuppressWarnings (MissingPermission) Final Place placeResult.addOnCompleteListener (new OnCompleteListener) - @Override public void onComplete (task @NonNull) - if (task.isSuccessful.) - task.getResult () if (likelyPlaces.getPlaceLikelihoods (m_max_entries). = else= {= count=M_MAX_ENTRIES; } = int= i=0; likelyplacenames'new string'count';likelyplaceaddresses'new string'count';'likelyplaceattributions'new list'count'; likelyplacelatlngs'new latlng'count;' forz (placelikelihood) placelilikehood' :'likelyplaces.getplacelikelilili Build a' list' of likely places to show the ' user.' likelyplaceattributions.placeLikelihood.getPlace () .getattributions(); i'if (i'gt; (account - 1)) - break; q / Show a dialogue that offers the user a list of likely locations, and add a marker at the chosen location. MapsActivityCurrentPlace.this.openPlacesDialog(); - still - Log.e (TAG, Exception: %s, task.getException); }); - still - / The user did not give permission. Log.i (TAG, User did not provide permission for location.); Add the default marker because the user hasn't chosen the location. map.addMarker (new MarkerOptions () .title (R.string.default_info_title)) .position (default) .snippet (R.string.default_info_snippet); </FindCurrentPlaceResponse></FindCurrentPlaceResponse></FindCurrentPlaceResponse></Place.Field>to obtain permission. getLocationPermission(); Displays a form that allows the user to choose a location from a list of likely locations. Issue/private void openPlacesDialog // Ask the user to choose where they are now. DialogInterface.OnClickListener listener - the new DialogInterface.OnClickListener () - @Override public void onClick (DialogInterface dialog, int which) / Argument which contains the position of the chosen element. LatLng markerLatLng - probably PlaceLatLngs,which); String markerSnippet - probably PlaceAddresses (which); If (probably PlaceAttributions, which) ! Add a marker for the chosen location with an information window ... with information about the location. map.addMarker .title .position (markerLatLng) .snippet (markerSnippet); Undo the card camera in the location of the marker. map.moveCamera (CameraUpdateFactory.newLatLng'oom (markerLatLng, DEFAULT_ZOOM)); } }; Displaying dialogue. AlertDialog Dialogue - new AlertDialog.Builder (it) .setTitle (R.string.pick_place) .setItems (probablyPlaceNames, listener) .show (); Updates the map's user interface settings depending on whether the user has given permission for the location. - private invalid updateLocationUI () - if (card - invalid) - return; - try if (locationPermissionGranted) - map.setMyLocationEnabled (admittedly); map.getUiSettings (.setMyLocationButtonEnabled (admittedly); - map.setMyLocationEnabled (false); map.getUiSettings (.setMyLocationButtonEnabled (false); lastKnownLocation - null; getLocationPermission (); < > Show/Hide kotlin Code for the map activity. Copyright 2020 Google LLC / / Apache License, Version 2.0 (License); You cannot use this file except under license. You can obtain a copy of the license // // // // Unless applicable law or agreed in writing, the software // is covered under the license on AS IS BASIS, // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. Cm. A license for a specific language that regulates permits and / restrictions under the license package. Import manifesto android.annotation.SuppressLint import android.content.DialogInterface import android.content.pm.pm.PackageManager import android.location.Location import android.os.Bundle import android.util.Log import android.view.menu import android.view.MenuIte import android.view.view import android.widget.FrameLayout import android.widget.TextView import androidx.appcompat.app.app.AlertDialog import androidx.appcompat.app.app.AppCompatActivity import androidx.core.app.app.ActivityCompat import импорт com.google.android.gms.location.FusedLocationProviderClient импорт com.google.android.gms.location.LocationServices импорт com.google.android.gms.maps.CameraUpdateFactory импорт com.google.android.gms.maps.GoogleMap импорт com.google.android.gms.maps.GoogleMap.InfoWindowAdapter импорт com.google.android.gms.maps.OnMapReadyCallback импорт com.google.android.gms.maps.SupportMapFragment импорт com.google.android.gms.maps.model.CameraPosition импорт com.. google.android.gms.maps.model.LatLng импорт com.google.android.gms.maps.model.Marker import com.google.android.gms.maps.model.MarkerOptions импортируют com.google.android.libraries.places.api.Places import com.google.android.libraries. places.api.model.Place import com.google.android.libraries.places.api.net.FindCurrentPlaceRequest импорт com.google.android.libraries.places.api.net.PlacesClient /- Деятельность, которая отображает карту, показывающую место в текущем местоположении устройства. B/ класс MapsActivityCurrentPlace : AppCompatActivity (), OnMapReadyCallback - частная var карта: GoogleMap? нулевой частный var cameraPosition: CameraPosition? Нулевой // Точка входа в Места API. частный lateinit var местаClient: PlacesClient // Точка входа в Fused Location Provider. частный lateinit var fusedLocationProviderClient: FusedLocationProviderClient // Местоположение по умолчанию (Сидней, Австралия) и увеличение по умолчанию для использования, когда разрешение местоположения // не предоставляется. частная val defaultLocation - LatLng (-33.8523341, 151.2106085) частное var местоположениеPermissionGranted - ложное // Географическое местоположение, где в настоящее время находится устройство. То есть последнее известное // местоположение, полученное поставщиком сплавленного местоположения. частный var lastKnownLocation: Местоположение? = null private var likelyPlaceNames: Array<String?> = arrayOfNulls(0) private var likelyPlaceAddresses: Array<String?> = arrayOfNulls(0) private var likelyPlaceAttributions: Array <> <*>?> = arrayOfNulls(0) private var likelyPlaceLatLngs: Array<LatLng?> = arrayOfNulls(0) override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) // Retrieve location and camera position from saved instance state. if (savedInstanceState != null) { lastKnownLocation = savedInstanceState.getParcelable(KEY_LOCATION) cameraPosition = savedInstanceState.getParcelable(KEY_CAMERA_POSITION) } // Retrieve the content view that renders the map. setContentView(R.layout.activity_maps) // Construct a PlacesClient Places.initialize(applicationContext, getString(R.string.maps_api_key)) placesClient = Places.createClient(this) // Construct a FusedLocationProviderClient. fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this) // Build the map. val mapFragment = supportFragmentManager .findFragmentById(R.id.map) as mapFragment?. getMapAsync (это) - сохраняет состояние карты при приостановлении действия. */ </LatLng?></*></String?></String?> </LatLng?></*></String?></String?/String?> fun onSaveInstanceState (outState: Bundle) - map?. Let the map - outState.putParcelable (KEY_CAMERA_POSITION, map.cameraPosition) outState.putParcelable (KEY_LOCATION, lastKnownLocation) - super.onSaveInstanceState (outstate) - / . - @param menu options. @return Boulan. Redefine fun onCreateOptionsMenu (menu: Menu): Boolean and menuInflater.inflate (R.menu.current_place_menu, menu) bring back the true menu. - @param menu item to handle. @return Boulan. Redefine fun onOptionsItemSelected (item: MenuItem): Boolean - if (item.itemId - R.id.option_get_place) - showCurrentPlace () - return true . This callback is triggered when the card is ready to use. Redefine the fun onMapReady (map: GoogleMap) - this.map - map/ Use a custom information window adapter to process multiple lines of text in the info window contents. this.map?. setInfoWindowAdapter (object : InfoWindowAdapter) / Return zero here, so getInfoContents is called next. override fun getInfoWindow (arg0: Marker): View? Return zero - override fun getInfoContents (marker: Marker): Browse / Inflate layouts for the information box, name and snippet. val infoWindow - layoutInflater.inflate (R.layout.custom_info_contents, findViewById'lt;FrameLayout'gt; (R.id.map), false) val name - infoWindow.findViewById'lt;textView'gt; (R.id.title) title.text-the marker.title val fragment - infoWindow.findViewById (R.id.snippet) snippet.text getDeviceLocation () . . . . . <TextView>. . . . . In/ private fun getDeviceLocation () - get the best and latest location of the device, which can be zero in rare cases when the location is not available. B/ try if (locationPermissionGranted) - val locationResult - fused PositionLocationProviderClient.lastLocationsResult.addOnCompleteListener (it) - the task - if (task.isSuccessful) moveCamera (CameraUpdateFactory.newLatLng'oom (LatLing (last note!!. latitude, last knowledge!!. longitude), DEFAULT_ZOOM.toFloat ()))- still - Log.d (TAG, Current location is zero. moveCamera (CameraUpdateFactory .newLatLng'oom (defaultLocation, DEFAULT_ZOOM.toFloat?). uiSettings?. isMyLocationButtonEnabled - False - Catch (e: SecurityException) - Log.e (Exception: %s, e.message, e)use the location of the device. The getLocationPermission is a location permission request so we can get the location of the device. The result of the resolution request is handled by a callback, according to The RequestPermissionsResult. The question/ if (ContextCompat.checkSelfPermission (this.applicationContext, Manifest.permission.ACCESS_FINE_LOCATION) - PackageManager.PERMISSION_GRANTED) - locationPermissionGrant - true - ActivityCompat.requestPermissions (this, (Manifest.permission.ACCESS_FINE_LOCATION PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION 1st) He said, he said, he said, he said. B/ Redefine pleasure onRequestPermissionsResult (requestCode: Int, resolution: Array'lt;string'gt;, grantResults: IntArray) - locationPermissionGranted - false when (requestCode) - PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION - / / If the request is canceled, the arrays of results are empty

PackageManager.PERMISSION_GRANTED. and shows the current location on the map -- provided @SuppressLint that the user has granted permission for the location. val placeFields - listOf (Place.Field.NAME, Place.Field.ADDRESS, Place.Field.LAT_LNG) / Use a builder to create FindCurrentPlaceRequest. val query - FindCurrentPlaceRequest.newInstance (placeFields) // Get probable locations - that is, businesses and other points of interest that // / are the best match for the current location of the device. val placeResult - placesClient.findCurrentPlace (request) placeResult.addOnCompleteListener - if (task.isSuccessful) task.result ! handle cases where fewer than 5 entries are returned. val count if (probably Places ! zero q likelyPlaces.placeLikelihoods.size qlt; m_max_entries m_max_entries) (count) likelyPlaceLatLngs - arrayOfNulls (account) for (place Likelihood in likelyPlaces?. placeLikelihoods ?: emptyList)/ / / Create a list of likely places to show the user. likelyPlaceNames-i' - placeLikelihood.place.name likelyPlaceAddresses.i - placeLikelihood.place.address likelyPlaceAttributions (i) - placeLikelihood.place.attributions likelyPlaceLatLn qgt'gs and add a marker !) in your chosen location. openPlacesDialog () - still - Log.e (TAG, Exception:task.exception) - still q / User did not issue permissions. Log.i (TAG, User did not provide permission for location.) / Add the default marker because the user did not choose the location. Map?. addMarker (MarkerOptions () .title (R.string.default_info_title)) .position (default) .snippet (R.string.default_info_snippet))))) / Encourage the user to resolve. getLocationPermission ()) - displays a form that allows the user to choose a place from a list of likely locations. OpenPlacesDialog (/ Ask the user to choose where they are now. val listener - DialogInterface.OnClickListener - dialogue, which - / / Argument what contains the position of the chosen product. Val markerLatLng - probably PlaceLatLngs, which var markerSnippet - probablyPlaceAddresses, if (probably PlaceAttributions, which) - markerSnippet - $markerSnippet $likelyPlaceAttributions,trimIndent () Map?. addMarker (MarkerOptions () .title (probably PlaceNames) .position (markerLatLng!!) .snippet (markerSnippet)) / The camera position of the map at the location of the marker. Map?. moveCamera (CameraUpdateFactory.newLatLng'oom (markerLatLng, DEFAULT_ZOOM.toFloat)) // Displaying dialogue. AlertDialog.Builder .setTitle (R.string.pick_place) .setItems (probablyPlaceNames, Listener) .show () . . . . . .
. . . . . .
. . . In/ private fun updateLocationUI () - if (map - zero) - return - try if (locationPermissionGranted) - map?. isMyLocationEnabled - the true map?. uiSettings?. isMyLocationButtonEnabled - a true map?. isMyLocationEnabled - false card?. uiSettings?. isMyLocationButtonEnabled - false lastKnownLocation - zero getLocationPermission (e: SecurityException) - Log.e (Exception: %s, e.message, e) - companion object - private val TAG PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION - MapsActivityCurrentPlace:class.java.simpleName private const val DEFAULT_ZOOM private const val KEY_CAMERA_POSITION - camera_position private const val KEY_LOCATION - location / Used to select the current location M_MAX_ENTRIES. Download and install Android Studio. Add a Google Play package to Android Studio. Clone or download Google Maps Android API v2 Sample repository if you didn't do so when you started reading this tutorial. Import tutorial project: In Android Studio, select the file of the new import project. Go to the place where you saved the Google Maps Android API v2 Sample Repository after downloading it. Find CurrentPlaceDetailsOnMap at this point: PATH-TO-SAVED-REPO/android-samples/lessons/CurrentPlaceDetailsOnMap Select project directory and then click OK. Android Studio is now building your project using the Gradle build tool. Get the API key and enable the necessary API to complete this tutorial, you need a Google API key that is authorized to use SDK maps for Android and SDK locations for Android. Click the button below to get the key and activate the API. For more information, see a complete guide to getting the API key. Add an API key to your project's gradle.properties file. Insert the API key into the cost of the GOOGLE_MAPS_API_KEY property: GOOGLE_MAPS_API_KEY-PASTE-YOUR-API-KEY-HERE When you create the Gradle app, copy the API key in the Android app's manifest. The build.gradle app file contains the following line, which displays a string of google_maps_key in the manifest to the GOOGLE_MAPS_API_KEY: resValue line, google_maps_key, (project.findProperty (GOOGLE_MAPS_API_KEY)?:) Create and run the Android Device Connect app to your computer. Follow the instructions to enable the developer settings on your Android device and set up a system to detect the device. (In addition, you can use the Android Virtual Device (AVD) Manager to customize the virtual device. Android Studio calls Gradle to create the app and then launches the app on the device or on the emulator. You should see a map with a number of markers around the current location similar to the image on this page. Troubleshooting: If you don't see the map, check that you received the API key and added it to the app as described above. Check out the log in Android Monitor Android Studio for reports of API key errors. If the map shows only one marker located on the Harbour Bridge (the default location in the app), check to see if you've granted permission to locate the app. The app offers location permission during the time to work, following the template described in the Android Resolution Guide. Note that you can also set permissions directly on your device by choosing the settings of the apps' name of the app's zgt; Resolution of the Location. For more information on how to handle permissions in the code, see the guide below to request permission for a location in your app. Use Android Studio debugging tools to review magazines and debug the app. This part of the tutorial explains the most significant parts of the CurrentPlaceDetailsOnMap app to help you understand how to create a similar app. Instant Customer API Locations The following interfaces provide basic entry points to SDK locations for Android: Android: GeoDataClient provides access to Google's local location database and business information. PlaceDetectionClient provides quick access to the current location of the device and provides an opportunity to report the location of the device in a specific location. The LocationServices interface is the main entry point for Android location services. To use the API, instantiate GeoDataClient, PlaceDetectionClient and FusedLocationProviderClient in the method of your fragment or activity onCreate () as shown in the following example of the code: protected void onCreate (Bundle savedInstanceState) - super.onstanceState;; setContentView (R.layout.activity_main); Build GeoDataClient. mGeoDataClient - Places.getGeoDataClient Build PlaceDetectionClient. mPlaceDetectionClient - Places.getPlaceDetectionClient (it's zero); Build FusedLocationProviderClient. mFusedLocationProvider Requesting a location permit Your app must request permission for a location to determine the location of the device and allow the user to click My Location on the map. This tutorial contains code that you need to request to obtain permission for the exact location. For more information, see the Android Resolution Guide. Add permission as a child to the element of the zlt'manifest in qlt;manifest xmlns:android/ android.permission.ACCESS_FINE_LOCATION map The following code checks whether the user has given permission for the exact location. If not, then he requests permission: private void getLocationPermission () . . . . . .
. . . . . The result of the resolution request is handled by a callback, according to The RequestPermissionsResult. The question/ if (ContextCompat.checkSelfPermission (this.getApplicationContext), android. Manifest.permission.ACCESS_FINE_LOCATION) - PackageManager.PERMISSION_GRANTED) - locationPermissionGrant - true; - another - ActivityCompat.requestPermissions (this is a new string. Manifest.permission.ACCESS_FINE_LOCATION, PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION); - private fun getLocationPermission (. . . . . .
. . . . The result of the permission request is handled by a callback, B/ если (ContextCompat.checkSelfPermission (this.applicationContext, Manifest.permission.ACCESS_FINE_LOCATION) - PackageManager.PERMISSION_GRANTED) - местоположениеПермиссияГрант - правда - иначе - ActivityCompat.requestPermissions (это, arrayOf (Manifest.permission.ACCESS_FINE_LOCATION), </manifest> </manifest> Override onRequestPermissionsResult () callback to process the outcome of a permit request: @Override public space on RequestPermissionsResult (int requestCode, @NonNull Resolution String, @NonNull int' grantResults) - locationPermissionGranted - false; (requestCode) - case PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION: -/ If the request is cancelled, the results arrays PackageManager.PERMISSION_GRANTED empty. LocationUI update redefinition of pleasure onRequestPermissionsResult (requestCode: Int, permissions: Array'lt;String'gt;, grantResults: IntArray) - locationPermissionGranted - false when (requestCode) - PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION PackageManager.PERMISSION_GRANTED - Add map display cards using SDK Maps for Android. Add an element to the layout file of your activity, activity_maps.xml. This element defines SupportMapFragment as a container for the map and for access to the GoogleMap object. The tutorial uses an android library-supported version of the map snippet to provide backward compatibility with earlier versions of the Android platform. qlt;fragment xmlns:android/ xmlns:tools/ android:id'id/map android:name'com.google.android.gms.maps.supportMapFragment android:layout_width match_parent layout_height match_parent tools:context.com.example.currentplacedetailsonmap.MapsActivityCurrent @Override Place Extract the content view that displays the map. setContentView (R.layout.activity_maps); override onCreate (savedInstanceState: Bundle?) - super.onCreate (savedInstanceState) / Get a view of the content that displays the map. setContentView (R.layout.activity_maps) - Implement the OnMapReadyCallback interface and override the onMapReady method to customize the map when a GoogleMap object is available: @Override // // Include the my Location layer and related control on the map. updateLocationUI (); Get the device's current location and set the map position. getDeviceLocation (::GoogleMap) - this.map - map / / Include my location layer and related control on the map. updateLocationUI () / Get the current location of the device and set the map position. getDeviceLocation ()In the onCreate method of your activity, get a pen to a snippet of the card by calling FragmentManager.findFragmentById. Then use getMapAsync to register on callback: SupportMapFragment mapFragment (SupportMapFragment) getSupportFragmentManager () .findFragmentById (R.id.map); mapFragment.getMapAsync val mapFragment - supportFragmentManager .findFragmentById (R.id.map) as SupportMapFragment? The phrase?. getMapAsync (this) Write updateLocationUI to customize your location controls on the map. If a user has given permission for a location, turn on my location layer and related control on the map, otherwise disable the layer and control, and set the current location to void: private invalid updateLocationUI () - if (map - null) - return; map.getUiSettings (.setMyLocationButtonEnabled (false); lastKnownLocation - null; getLocationPermission (); - catch (SecurityException e) - Log.e (Exception: %s, e.getMessage); isMyLocationEnabled - the true map?. uiSettings?. isMyLocationButtonEnabled - a true map?. isMyLocationEnabled - false card?. uiSettings?. isMyLocationButtonEnabled - false lastKnownLocation - zero getLocationPermission (e: SecurityException) - Log.e (Exception: %s, e.message, e) - Get the location of the Android device and position the map Use the fused location provider to find the last known location of the device, and then use this location to position the map. The textbook contains the code you need. For more information on the location of the device, visit the Google Play provider's guide. private void getDeviceLocation () . . . . . .
. . . . . C/ Try if (locationPermissionGranted) locationResult.addOnCompleteListener (this is the new OnCompleteListener() - @Override public void on The Complete (@NonNull Target) - if the site is a task. lastKnownLocation - task.getResult (); DEFAULT_ZOOM if (lastNowation ! - null) - map.moveCamera (CameraUpdateFactory.newLatLng'oom (new LatLng (lastKnownLocation.getLatitude.), lastKnownLocation.getLongitude ()) Log.e (TAG, Exception: %s, task.getException(); map.moveCamera (CameraUpdateFactory .newLatLng'oom (default, DEFAULT_ZOOM))); map.getUiSettings ().setMyLocationButtonEnabled (false); }); - To catch the location(e) - Log.e (Exception: %s, e.getMessage), e); Private getDeviceLocation is to get the best and latest location of the device, which can be zero in rare cases when the location is not available. In/ try if (locationPermissionGranted) - val locationResult - fused PositionProviderClient.lastLocationresult.addOnCompleteListener (it) - task - qgt if (task.isSuccessful) // Set the position of the camera map to the current location of the device. moveCamera (CameraUpdateFactory.newLatLng'oom (LatLng (last news!!. latitude, last knowledge!!. longitude), DEFAULT_ZOOM.toFloat ()))- still - Log.d (TAG, Current Location is zero. moveCamera (CameraUpdateFactory .newLatLng'oom (defaultLocation, DEFAULT_ZOOM.toFloat?). uiSettings?. isMyLocationButtonEnabled - false - catch (e: SecurityException) - Log.e (Exception: %s, e.message, e) - Get your current location Use SDK seats for Android to get a list of likely locations in the current location of the device. In this context, the place is a business or other point of interest. This tutorial gets the current location when the user presses the Get Place button. It offers the user a list of likely locations to choose from, and then adds a marker on the map at the location of the chosen location. The tutorial provides code that you need to interact with SDK places for Android. For more information, see a guide to get your current seat. Create a layout file (current_place_menu.xml) for the options menu and override the onCreateOptionsMenu method to customize the options menu. See the accompanying sample of the code application. Redefine onOptionsItemSelected () method to get the current location when a user clicks on the Get Place option: @Override public boolean onOptionsItemSelected (item MenuItem) - if (item.getItemId)- R.id.option_get_place) - showCurrentPlace(); The return is true; - Override fun onOptionsItemSelected (item: MenuItem): Boolean - if (item.itemId - R.id.option_get_place) - showCurrentPlace () - return true - Create showCurrentPlace method() to get a list of likely locations in the current location of the device: private invalid showrCurentPlace (locationPerGranted) PlaceFields and Arrays.asList (Place.Field.NAME, Place.Field.ADDRESS, Place.Field.LAT_LNG); Use a builder to create FindCurrentPlaceRequest. FindCurrentPlaceRequest - FindCurrentPlaceRequest.newInstance Get probable locations - that is, businesses and other points of interest that are the best for the current location of the device. (MissingPermission) окончательное место<FindCurrentPlaceResponse> задачиРезульта</FindCurrentPlaceResponse> </Place.Field> </Place.Field> placeResult.addOnCompleteListener (new OnCompleteListener) - @Override public void onComplete (task @NonNull) - if (task.isSuccessful.) / Set a count, processing cases where fewer than 5 entries are returned m_max_entries. = else= {= count=M_MAX_ENTRIES; } = int= i=0; likelyplacenames'new string'count';likelyplaceaddresses'new string'count';'likelyplaceattributions'new list'count'; likelyplacelatlngs'new latlng'count;' forz (placelikelihood) placelilikehood' :'likelyplaces.getplacelikelilili Build a' list' of likely places to show the ' user.' likelyplaceattributions.placeLikelihood.getPlace () .getattributions(); i'if (i'gt; (account - 1)) - break; / Show a dialogue offering the user a list of likely locations, and add a / marker at the chosen location. MapsActivityCurrentPlace.this.openPlacesDialog(); - still - Log.e (TAG, Exception: %s, task.getException); }); - still - / The user did not give permission. Log.i (TAG, User did not provide permission for location.); Add the default marker because the user hasn't chosen the location. map.addMarker (new MarkerOptions () .title (R.string.default_info_title)) .position (default) .snippet (R.string.default_info_snippet); Prepare the user for a resolution. getLocationPermission(); - @SuppressLint (MissingPermission) private fun showCurrentPlace () - if (map - null) - return if (locationPermissionGranted) / Use the fields to determine the types of data for a return. val placeFields - listOf (Place.Field.NAME, Place.Field.ADDRESS, Place.Field.LAT_LNG) / Use a builder to create FindCurrentPlaceRequest. val query - FindCurrentPlaceRequest.newInstance (placeFields) / Get probable locations - that is, businesses and other points of interest that are best for the current location of the device. val placeResult - placesClient.findCurrentPlace (request) placeResult.addOnCompleteListener - task - if (task.isSuccessful) task.result ! task - task.result / Set a count, processing cases when fewer than 5 entries are returned. val count if (probably Places!) null - likelyPlaces.placeLikelihoods.size qlt; m_max_entries m_max_entries) (counting) likelyPlaceLatLngs - arrayOfNulls (account) for (the place of Likelihood in the </FindCurrentPlaceResponse> </FindCurrentPlaceResponse> ?: EmptyList ()) / / Create a list of likely places to show the user. likelyPlaceNames-i' - placeLikelihood.place.name likelyPlaceAddresses.i - placeLikelihood.place likelyPlaceAttributions (i) - placeLikelihood.place.attributions likelyPlaceLatLn zgt.gs and add a marker!) at your chosen location. openPlacesDialog () - still - Log.e (TAG, Exception: %s, task.exception) - -3 / User did not give permission. Log.i (TAG, User did not provide permission for location.) / Add the default marker because the user did not choose the location. Map?. addMarker (MarkerOptions () .title (R.string.default_info_title)) .position (default) .snippet (R.string.default_info_snippet))))) / Encourage the user to resolve. getLocationPermission - Create an openPlacesDialog method to display a form that allows the user to select a spot from a list of likely locations. Add a marker to the map for your chosen location. The contents of the marker include the name and address of the location, as well as any attributes that the API delivers: the private void openPlacesDialog () / Ask the user to choose where they are now. DialogInterface.OnClickListener listener - the new DialogInterface.OnClickListener () - @Override public void onClick (DialogInterface dialog, int which) / Argument which contains the position of the chosen element. LatLng markerLatLng - probably PlaceLatLngs,which); String markerSnippet - probably PlaceAddresses (which); If (probably PlaceAttributions, which) ! Add a marker for the chosen location with an information window ... with information about the location. map.addMarker .title .position (markerLatLng) .snippet (markerSnippet); Undo the card camera in the location of the marker. map.moveCamera (CameraUpdateFactory.newLatLng'oom (markerLatLng, DEFAULT_ZOOM)); } }; Displaying dialogue. AlertDialog Dialogue - new AlertDialog.Builder (it) .setTitle (R.string.pick_place) .setItems (probablyPlaceNames, listener) .show (); - private fun openPlacesDialog () / Ask the user to choose where they are now. val listener - DialogInterface.OnClickListener - dialogue, which - / / Argument which contains the position of the chosen element. Val markerLatLngs, probably PlaceLatLngs, which var markerSnippet - probably PlaceAddresses , which if (probably PlaceAttributions, that) - markerSnippet $markerSnippet $likelyPlaceAttributions,trimIndent () Map?. addMarker (MarkerOptions () .title (probablyPlaceNames) .position (markerLatLng!!) .snippet (markerSnippet)) / Position Camera in the place of the marker. Map?. moveCamera (CameraUpdateFactory.newLatLng'oom (markerLatLng, DEFAULT_ZOOM.toFloat)) // Displaying dialogue. AlertDialog.Builder (it) .setTitle (R.string.pick_place) .setItems (probablyPlaceNames, listener) .show () - Create a custom layout for the contents of the information box. This allows you to display multiple lines of content in the information window. First, add the XML layout file, custom_info_contents.xml, which contains a text view for the information window header, and another text view for the piece (i.e. the text content of the information window): qlt?xml version?0 encoding'utf-8?'gt; implementation of the InfoWindowAdapter interface to inflate the layout and download the contents of the information window: / Use a custom information box adapter to process multiple lines of text in the contents of the window. this.map.setInfoWindowAdapter (новый GoogleMap.InfoWindowAdapter() - @Override // Возвращение нулевой здесь, так что getInfoContents () называется следующим @Override. <LinearLayout xmlns:android= android:layout_width=wrap_content android:layout_height=wrap_content android:layoutdirection=locale android:orientation=vertical> <TextView android:id=@+id/title android:layout_width=wrap_content android:layout_height=wrap_content android:layout_gravity=center_horizontal android:textcolor=#ff000000 android:textstyle=bold></TextView> <TextView android:id=@+id/snippet android:layout_width=wrap_content android:layout_height=wrap_content android:textcolor=#ff7f7f7f></TextView> </LinearLayout> Просмотр infoWindow - getLayoutInflater ().надуть (R.layout.custom_info_contents, (FrameLayout) найтиViewById (R.id.map), ложный); Название TextView - infoWindow.findViewById(R.id.title); title.setText (marker.getTitle()); Фрагмент TextView - infoWindow.findViewById(R.id.snippet); snippet.setText (marker.getSnippet.);; возврат infoWindow; } }); Use a custom information window adapter to process multiple lines of text in the contents of the window/ info. this.map?. setInfoWindowAdapter (object : InfoWindowAdapter) / Return zero here, so getInfoContents is called next. override fun getInfoWindow (arg0: Marker): View? Return zero - override fun getInfoContents (marker: Marker): Browse and / Inflate layouts for the information box, name and snippet. val infoWindow - layoutInflater.inflate (R.layout.custom_info_contents, findViewById'lt;FrameLayout(R.id.map), false) val name - infoWindow.findViewById'lt;R.id.title) title.text.text.title val fragment - infoWindow.findViewById'lt;Text SDK cards for Android displays content No need </TextView> </FrameLayout> </FrameLayout> listener for the marker if you are happy to use the default behavior. Save the state of the map Save the location of the camera map and the location of the device. When a user rotates an Android device or makes changes to the configuration, the Android framework destroys and rearranges the card. To keep users running smoothly, it's good to store the app's status and restore it when you need it. This tutorial contains all the code you need to save the map status. For more information, see the map's saved split guide to setting up key values to store activity status: the private static final line KEY_CAMERA_POSITION and camera_position; Private static final KEY_LOCATION and location; Private const val KEY_CAMERA_POSITION - camera_position private const val KEY_LOCATION - location Implementation onSaveInstanceState () callback to save a state when activity is suspended: @Override protected void onAveInstanceState (Bundle outState) - if (map!) - outState.putParcelable (KEY_CAMERA_POSITION, map.getCamePosition); outState.putParcelable (KEY_LOCATION, lastKnownLocation); - super.onSaveInstanceState (outState); - override the fun onAveInstanceState (outState: Bundle) - map?. Let the map - outState.putParcelable (KEY_CAMERA_POSITION, map.cameraPosition) outState.putParcelable (KEY_LOCATION, lastKnownLocation) - super.onSaveInstanceState (outState) - In the onCreate method of your activity, extracting the location of the device and the position of the camera map, if previously saved: If (savedVstanceState! - zero) - lastNow - savedInstanceState.getParcelable (KEY_LOCATION); cameraPosition - savedInstanceState.getParcelable (KEY_CAMERA_POSITION); If (savedInstanceState ! - null) - lastMoan - savedInstanceState.getParcelable (KEY_LOCATION) cameraPosition - savedInstanceState.getParcelable (KEY_CAMERA_POSITION) - Congratulations! You've created an android app that shows your current location on Google Maps. You've also learned how to use SDK Maps for Android, SDK Places for Android, and fused location providers. Provider.