06/06/2024 - Tutorial: Integrando o Google Maps com Rastreamento de Localização em Tempo Real em um Aplicativo Android Kotlin

Pré-requisitos

Android Studio instalado no seu computador.

Conhecimento básico da linguagem de programação Kotlin.

Uma conta na Google Cloud Platform para acessar a API do Google Maps.

Passo 1: Configurar um Novo Projeto no Android Studio

Abra o Android Studio e crie um novo projeto.

Selecione "Atividade Vazia" como o modelo do projeto e clique em "Avançar".

Nomeie seu projeto, escolha Kotlin como a linguagem de programação e defina a versão mínima do SDK como Android 4.1 (API 16) ou superior.

Clique em "Concluir" para criar seu projeto.

Passo 2: Adicionar a Chave da API do Google Maps

Visite a Google Cloud Platform e habilite a API "Maps SDK for Android".

Crie uma nova chave de API e restrinja-a para aplicativos Android.

Copie a chave de API gerada.

No seu projeto do Android Studio, abra o arquivo AndroidManifest.xml e adicione o seguinte código dentro da tag <application>:

xml

<meta-data

android:name="com.google.android.geo.API\_KEY"

android:value="SUA\_CHAVE\_DA\_API"/>

Substitua SUA\_CHAVE\_DA\_API pela chave de API que você gerou anteriormente.

Passo 3: Adicionar Dependências do Google Maps

Abra o arquivo build.gradle (Módulo: app) e adicione as seguintes dependências:

gradle

dependencies {

implementation 'com.google.android.gms:play-services-maps:17.0.1'

implementation 'com.google.android.gms:play-services-location:18.0.0'

}

Clique em "Sincronizar agora" para sincronizar o projeto com as novas dependências.

Passo 4: Criar a MainActivity para Rastreamento de Localização em Tempo Real

MainActivity.kt

import android.Manifest

import android.content.pm.PackageManager

import android.location.Location

import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import androidx.core.app.ActivityCompat

import androidx.core.content.ContextCompat

import com.google.android.gms.location.\*

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.LatLng

class MainActivity : AppCompatActivity(), OnMapReadyCallback {

private lateinit var mMap: GoogleMap

private lateinit var fusedLocationClient: FusedLocationProviderClient

private lateinit var locationCallback: LocationCallback

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

// Inicialize o cliente de localização

fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)

// Obtenha o SupportMapFragment e seja notificado quando o mapa estiver pronto para ser usado.

val mapFragment = supportFragmentManager.findFragmentById(R.id.map) as SupportMapFragment

mapFragment.getMapAsync(this)

}

override fun onMapReady(googleMap: GoogleMap) {

mMap = googleMap

// Verifique a permissão de localização

if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED) {

enableMyLocation()

} else {

// Solicite permissão de localização

ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.ACCESS\_FINE\_LOCATION), CODIGO\_DE\_PERMISSAO\_DE\_LOCALIZACAO)

}

}

private fun enableMyLocation() {

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return

}

mMap.isMyLocationEnabled = true

fusedLocationClient.lastLocation.addOnSuccessListener { location: Location? ->

// Obteve a última localização conhecida. Em algumas situações raras, isso pode ser nulo.

location?.let {

val currentLatLng = LatLng(it.latitude, it.longitude)

mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLatLng, 15f))

}

}

val locationRequest = LocationRequest.create().apply {

interval = 10000

fastestInterval = 5000

priority = LocationRequest.PRIORITY\_HIGH\_ACCURACY

um exemplo de xml que funcionaria com isto:

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<fragment

android:id="@+id/map"

android:name="com.google.android.gms.maps.SupportMapFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

</RelativeLayout>