Tutorial para Uso do MapBox no Android

Este tutorial abordará a integração do MapBox no Android, incluindo a configuração do projeto, a adição de permissões para localização, a simulação de localização no emulador, e a implementação de uma funcionalidade de marcação no mapa baseada na localização do GPS. Vamos utilizar Kotlin para o desenvolvimento do projeto.

Passo 1: Configuração Inicial do Projeto

1. Criação do Projeto no Android Studio\*\*

- Abra o Android Studio e crie um novo projeto com uma Activity vazia.

2. Adicionando Dependências e Configurações

- Adicione as dependências do MapBox no arquivo `build.gradle` do módulo (geralmente `app/build.gradle`).

```groovy

dependencies {

implementation 'com.mapbox.maps:android:10.4.0'

implementation 'com.mapbox.plugin:locationcomponent:10.4.0'

}

```

- Adicione o repositório do MapBox no arquivo `build.gradle` do projeto (geralmente `build.gradle` na raiz do projeto).

```groovy

allprojects {

repositories {

google()

mavenCentral()

maven { url 'https://api.mapbox.com/downloads/v2/releases/maven' }

}

}

```

- Configure a chave de acesso ao MapBox no arquivo `AndroidManifest.xml`. Obtenha a chave de API registrando-se no [site oficial do MapBox](https://account.mapbox.com/).

```xml

<application>

<meta-data

android:name="com.mapbox.token"

android:value="YOUR\_MAPBOX\_ACCESS\_TOKEN"/>

</application>

```

Passo 2: Adicionando Permissões para Localização

Adicione as permissões necessárias no arquivo `AndroidManifest.xml`.

```xml

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

```

Passo 3: Configuração do Layout e Inicialização do Mapa

Crie um layout XML para a Activity principal com um `MapView`.

```xml

<!-- res/layout/activity\_main.xml -->

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

xmlns:mapbox="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

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

<com.mapbox.maps.MapView

android:id="@+id/mapView"

android:layout\_width="match\_parent"

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

</RelativeLayout>

```

No arquivo `MainActivity.kt`, inicialize o MapView.

```kotlin

// MainActivity.kt

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import com.mapbox.maps.MapView

import com.mapbox.maps.Style

class MainActivity : AppCompatActivity() {

private lateinit var mapView: MapView

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

mapView = findViewById(R.id.mapView)

mapView.getMapboxMap().loadStyleUri(Style.MAPBOX\_STREETS)

}

override fun onStart() {

super.onStart()

mapView.onStart()

}

override fun onStop() {

super.onStop()

mapView.onStop()

}

override fun onLowMemory() {

super.onLowMemory()

mapView.onLowMemory()

}

override fun onDestroy() {

super.onDestroy()

mapView.onDestroy()

}

}

```

Passo 4: Adicionando Localização do Usuário

Para obter a localização do usuário e adicionar uma marcação no mapa, é necessário implementar o componente de localização.

```kotlin

import android.annotation.SuppressLint

import android.location.Location

import com.mapbox.android.core.permissions.PermissionsListener

import com.mapbox.android.core.permissions.PermissionsManager

import com.mapbox.maps.plugin.locationcomponent.LocationComponentPlugin

import com.mapbox.maps.plugin.locationcomponent.LocationComponentActivationOptions

import com.mapbox.maps.plugin.locationcomponent.OnLocationClickListener

import com.mapbox.maps.plugin.locationcomponent.OnLocationLongClickListener

class MainActivity : AppCompatActivity(), PermissionsListener {

private lateinit var mapView: MapView

private lateinit var locationComponent: LocationComponentPlugin

private lateinit var permissionsManager: PermissionsManager

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

mapView = findViewById(R.id.mapView)

mapView.getMapboxMap().loadStyleUri(Style.MAPBOX\_STREETS) { style ->

enableLocationComponent()

}

}

@SuppressLint("MissingPermission")

private fun enableLocationComponent() {

if (PermissionsManager.areLocationPermissionsGranted(this)) {

val locationComponentActivationOptions = LocationComponentActivationOptions.builder(this, mapView.getMapboxMap().getStyle()!!)

.useDefaultLocationEngine(true)

.build()

locationComponent = mapView.getLocationComponentPlugin()

locationComponent.activateLocationComponent(locationComponentActivationOptions)

locationComponent.isLocationComponentEnabled = true

locationComponent.addOnLocationClickListener(OnLocationClickListener {

val location = locationComponent.lastKnownLocation

if (location != null) {

addMarker(location)

}

})

} else {

permissionsManager = PermissionsManager(this)

permissionsManager.requestLocationPermissions(this)

}

}

private fun addMarker(location: Location) {

// Adicione o código para adicionar a marcação no mapa aqui

}

override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {

permissionsManager.onRequestPermissionsResult(requestCode, permissions, grantResults)

}

override fun onExplanationNeeded(permissionsToExplain: MutableList<String>?) {

// Explique ao usuário por que é necessária a permissão de localização

}

override fun onPermissionResult(granted: Boolean) {

if (granted) {

enableLocationComponent()

} else {

// Permissão não concedida

}

}

}

```

Passo 5: Adicionando Marcação no Mapa

Implemente a função `addMarker` para adicionar uma marcação com base na localização do GPS.

```kotlin

import com.mapbox.maps.extension.style.layers.generated.symbolLayer

import com.mapbox.maps.extension.style.sources.generated.geoJsonSource

import com.mapbox.geojson.Point

private fun addMarker(location: Location) {

val mapboxMap = mapView.getMapboxMap()

val point = Point.fromLngLat(location.longitude, location.latitude)

val source = geoJsonSource("marker-source") {

geometry(point)

}

mapboxMap.getStyle {

it.addSource(source)

it.addLayer(symbolLayer("marker-layer", "marker-source") {

iconImage("marker-icon")

iconAllowOverlap(true)

})

}

}

```

Passo 6: Simulando a Localização no Emulador

1. \*\*Configuração do Emulador\*\*

- Abra o emulador Android.

- Vá para \*\*Extended Controls\*\* (três pontos verticais no lado direito).

- Selecione \*\*Location\*\*.

- Insira as coordenadas desejadas (latitude e longitude) e clique em Send.

2. Testando o Projeto

- Execute o projeto no emulador.

- A aplicação deve solicitar permissões de localização e, uma vez concedidas, exibir o mapa com a localização simulada.

Referências da Documentação Oficial

- [MapBox Maps SDK for Android](https://docs.mapbox.com/android/maps/guides/)

- [MapBox Location Component](https://docs.mapbox.com/android/plugins/guides/location/)

Este tutorial cobre a configuração básica e a funcionalidade essencial para utilizar mapas do MapBox em um aplicativo Android. Para funcionalidades mais avançadas, consulte a documentação oficial do MapBox.