Android 10 Changes On P2P Connection And Its Further Action

1. Android 10 or higher, it must have the ACCESS_FINE_LOCATION permission in order to use several methods within the Wi-Fi, Wi-Fi Aware, or Bluetooth APIs

Some telephony, Bluetooth, Wi-Fi APIs require FINE location permission

If your app targets Android 10 or higher, it must have the `ACCESS_FINE_LOCATION` permission in order to use several methods within the Wi-Fi, Wi-Fi Aware, or Bluetooth APIs. The following sections list the affected classes and methods.

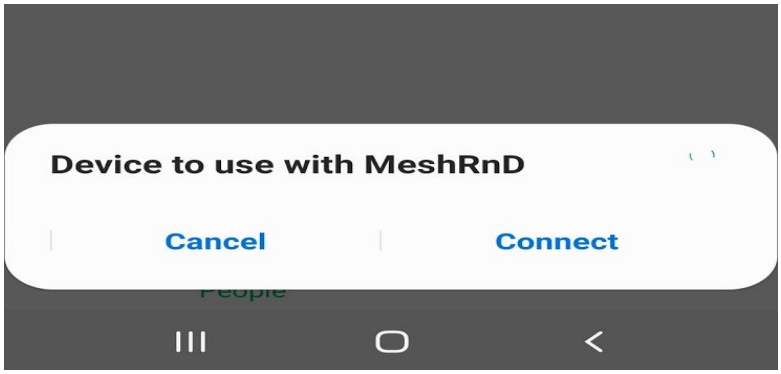
> ★ **Note:** If your app runs on Android 10 or higher but targets Android 9 (API level 28) or lower, you can use the affected APIs (except for **`WifiP2pManager`** APIs) as long as your app has declared either the **`ACCESS_COARSE_LOCATION`** or the **`ACCESS_FINE_LOCATION`** permission.

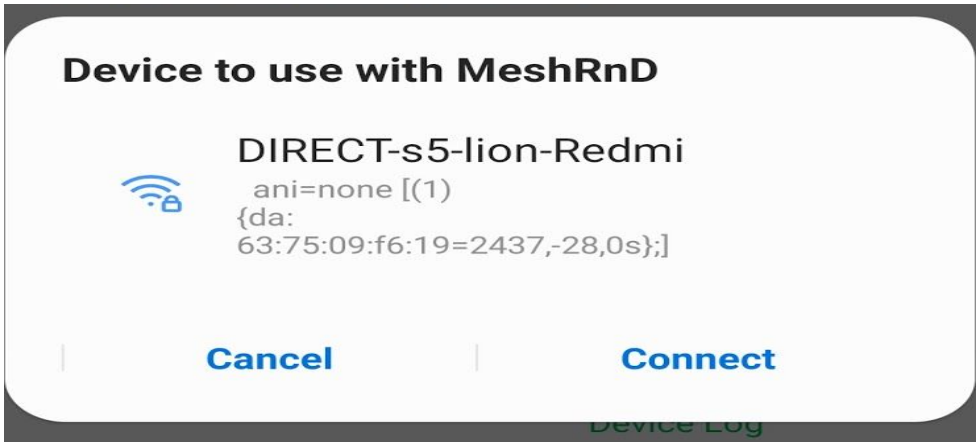
https://developer.android.com/about/versions/10/privacy/changes

2. Wi-Fi Network Request API for peer-to-peer connectivity

Initiating the request to connect to a peer device launches a dialog box on the same device, from which that device's user can accept the connection request.

> ★ **Note:** Creating a connection using this API does not provide an internet connection to the app or to the device. To provide an internet connection to the apps on a device, use the Wi-Fi Suggestion API instead.





https://developer.android.com/guide/topics/connectivity/wifi-bootstrap

3. Wi-Fi suggestion API for internet connectivity

Devices running Android 10 (API level 29) and higher allow your app to add network credentials for a device to auto-connect to a Wi-Fi access point. You can supply suggestions for which network to connect to using `WifiNetworkSuggestion`. The platform ultimately chooses which access point to accept based on the input from your app and others.

The suggestions from the app must be approved by the user before the platform initiates a connection to them. This approval is provided by the user in response to a notification the first time the platform finds a network matching one of the suggestions from the app in scan results. When the platform connects to one of the network suggestions, the settings show text that attributes the network connection to the corresponding suggester app.

Handling user disconnects

If the user uses the Wi-Fi picker to explicitly disconnect from one of the network suggestions when connected to it, then that network is blacklisted for 24 hours. During the blacklist period, that network will not be considered for auto-connection, even if the app removes and re-adds the network suggestion corresponding to the network.

Changing approval status for app

A user declining the network suggestion notification removes the `CHANGE_WIFI_STATE` permission from the app. The user can grant this approval later by going into the Wi-Fi control menu (**Settings** > **Apps & notifications** > **Special App access** > **Wi-Fi Control** > App name).

https://developer.android.com/guide/topics/connectivity/wifi-suggest#change-approval

4. Wi-Fi Direct connection API

The `WifiP2pConfig` and `WifiP2pManager` API classes have updates in Android 10 to support fast connection establishment capabilities to Wi-Fi Direct using predetermined information.

https://developer.android.com/about/versions/10/features

The following code sample shows how to create a group using predetermined information:

KOTLIN JAVA

```
WifiP2pManager manager = (WifiP2pManager) getSystemService(Context.WIFI_P2P_SERVICE);
Channel channel = manager.initialize(this, getMainLooper(), null);

// prefer 5G band for this group
WifiP2pConfig config = new WifiP2pConfig.Builder()
.setNetworkName("networkName")
.setPassphrase("passphrase")
.enablePersistentMode(false)
.setGroupOperatingBand(WifiP2pConfig.GROUP_OWNER_BAND_5GHZ)
.build();

// create a non-persistent group on 5GHz
manager.createGroup(channel, config, null);
```

To join a group using credentials, replace `manager.createGroup()` with the following:

KOTLIN JAVA

```
manager.connect(channel, config, null);
```