



C . E . S . A . R



THE DEVELOPER'S
CONFERENCE

WifiManager - O impacto das mudanças no Android Q

The Developer's Conference - São Paulo

17.07.2019



Haldny Santos

Engenheiro de Software no CESAR, trabalha com Android embarcado desde 2014, 9+ anos de experiência.



Gabriel Silva

Engenheiro de Software no CESAR, trabalha com Android embarcado desde 2017, 3+ anos de experiência.

Antes de tudo, O que é o WifiManager?



—
**API responsável por
gerenciar todos os
aspectos da
conectividade**

Wi-Fi.

WifiManager

Lista de Redes

Podemos visualizar e atualizar a lista de redes configuradas por aplicativos e operadoras.

Scan de AP's

Contém informações sobre todos os access points disponíveis e pode decidir qual o melhor para conectar-se.

Conexão com Redes

Estabelece uma conexão ou desconecta de redes. Além de consultar os estados das redes.

Enviar eventos de WiFi

Define e envia os broadcasts de WiFi quando há uma mudança no estado das redes.

E quem usa essa API?





Esse outros

Apps também!

Criar e Listar Redes - 41M+



Analise de Rede - 22M+



VPN - 110M+



Ufa, tudo isso?

**E se eu te disser que
tem mais?**



Free WiFi Password
Degoo Backup AB - Cloud



Free WiFi Connectio
48 Dev Studio



Free WiFi Connectio
MoDev Studio



Free WiFi Connect
Mentisco



Free WiFi Internet -
AppWorld Infotech



WiFi Password
Global Wifi Technology



WiFi Map — Free Pa
WiFi Map LLC



Free WiFi Spot
Rsp974 Apps



Swift WiFi - Free Wi
DotC United Swift Team



Free Open Wifi Con
Easy way inc.





Free WiFi
Wiman



Free Wifi Connect
PBreak Studio



Free Wifi Password
Gamesoftgames



WiFi Master - by wi
LINKSURE NETWORK H



Free WiFi Connect
Music Store Studio



Free WiFi App: WiFi
RIWW



Free WiFi Connect |
MobileToolingStation



OPEN FREE WiFi P
DENISE ALBESCU APPS



Free WiFi Access
Cyber Gatekeeper 2000



Free Wifi Password
kornel_studio





Google Wifi
Google LLC



Free Wifi Hotspot -
vinhash



Free WiFi Connect |
meewansa123



Free Wifi Password
Gamesoftgames



Find Wifi - Free wifi
WeFi



Free Wifi Hotspot P
Seer Tools



WiFi Finder - Free W
Internet Speed, Etrality



Free Wifi Hotspot P
48 Dev Studio



Share Wifi Mobile H
Free Wifi password proj



Free WiFi Password
Wifi Finder





WiFi Doctor Free - S
cxzh.ltd

★★★★★



Free Wifi Password
Droid-Developer

★★★★★



Wifi Hotspot Portab
vinhash

★★★★★



Oi WiFi
Oi Aplicativos

★★★★★



Free WiFi Internet F
kapron-ap

★★★★★



WiFi Free to Conne
Pinkbird Studio

★★★★★



Phone Free Call - G
Hotspot VPN(Proxy & S

★★★★★



WPSApp
TheMauSoft

★★★★★



Free Wifi connectio
NTSM

★★★★★



Free Wifi Password
InfoWork Tech

★★★★★



Mobile WiFi Router
TenTech

★★★★★



Free Wifi
unk.nown

★★★★★



Wifi Auto
Cygnus Software

★★★★★



Free Wifi Pass
fondev

★★★★★



FREE WIFI PASSWC
ANGERLAGERBICAPPS

★★★★★



osmino: Share WiFi
RIWW

★★★★★



Wifi hotspot router
Connexion

★★★★★



WiFi Key Connecto
Smart Tech My WiFi

★★★★★



Portable Wifi Hotsp
GOMIN MOBILE

★★★★★



WIFI Connection
Develop Design Stone

★★★★★



Wifi Calling
ppgirl



Free WiFi Trials
Cyber Gatekeeper 2000



WiFi Auto Unlock 2
App Plus 2018



WiFi Hacker : WiFi
Entertainment App Worl



WiFi WPS WPA TES
Sangiorgi Srl



WPS Wifi Checker F
RenderSoftware



Free WiFi Connect
NJ Application



WiFi Connection M
roamingsoft



WiFi Warden
EliyanPro



Best WiFi
Pintac





WiFi WPS Connect
WiFi Password Team Pr

★★★★★



Free Wifi Password
InfoWork Tech

★★★★★



WiFi Analyzer
Abdelrahman M. Sid

★★★★★



WiFi Connect
JH Jeong

★★★★★



Free WiFi
Versatile Mobitech Pvt L

★★★★★



WiFi Password Rec
WiFi Password Team Pr

★★★★★



WiFi Direct +
Dev Netcomps Android

★★★★★



Japan Connected-fi
NTT Broadband Platfor

★★★★★



Portable Wi-Fi hotspots
CORE TECHNOLOGY

★★★★★



WiFi Password Rec
WiFi Password Recover

★★★★★



Free Wifi Finder
SoulApps

★★★★★



Wifi Password Hack
Droid-Developer

★★★★★



FoxFi (WiFi Tether)
FoxFi Software

★★★★★



Wifi Connect
UMRSAMA

★★★★★



WiFi Monitor: analy
Alexander Kozyukov

★★★★★



Wifi Hotspot
Orange VN Team

★★★★★



Free WiFi
FreeWiFi

★★★★★



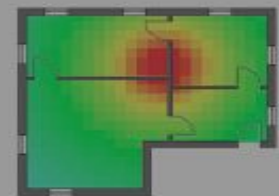
WiFiKill PRO - WiFi
devdev00

★★★★★



Show Wifi password
JunStudio

★★★★★



WiFi Heatmap
Wi-Fi Solutions

★★★★★

Como utilizamos o WifiManager

- Criar e Excluir redes
- Consultar se o WiFi está ligado
- Criar um hotspot local
- Consultar se o app pode fazer scan das redes
- Conectar em uma rede
- Consultar se o device suporta Wi-Fi Direct

E quais tipos de
redes **WiFi** eu
posso criar?

- Hotspot 1.0
- Hotspot 2.0 (Passpoint)

E o que mudou nas redes WiFi no Android Q?

Redes **Passpoint** sofreram poucos impactos no **Android Q!**

O grande **Impacto** aconteceu nas redes **Hotspot 1.0!**

Métodos foram Descontinuados

- addNetwork(WifiConfiguration config)*
- createWifiLock(String tag)
- disableNetwork(int netId)*
- disconnect()*
- enableNetwork(int netId, boolean attemptConnect)*
- getConfiguredNetworks()*
- getPasspointConfigurations()*
- isDeviceToApRttSupported()

Métodos foram Descontinuados

- `isScanAlwaysAvailable()`
- `reassociate()*`
- `reconnect()*`
- `removeNetwork(int netId)*`
- `setWifiEnabled(boolean enabled)*`
- `updateNetwork(WifiConfiguration config)*`

Métodos foram Criados

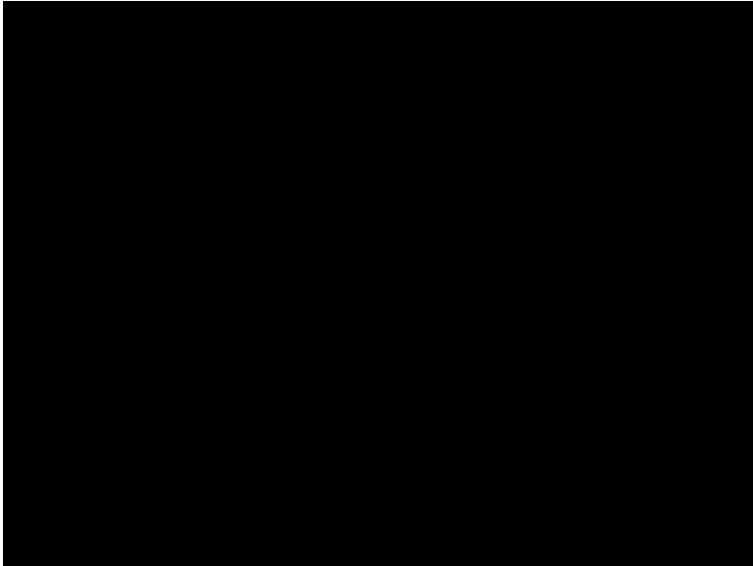
- `addNetworkSuggestions(List<WifiNetworkSuggestion> networkSuggestions)*`
- `getMaxNumberOfNetworkSuggestionsPerApp()*`
- `isEasyConnectSupported()`
- `isEnhancedOpenSupported()`

Métodos foram Criados

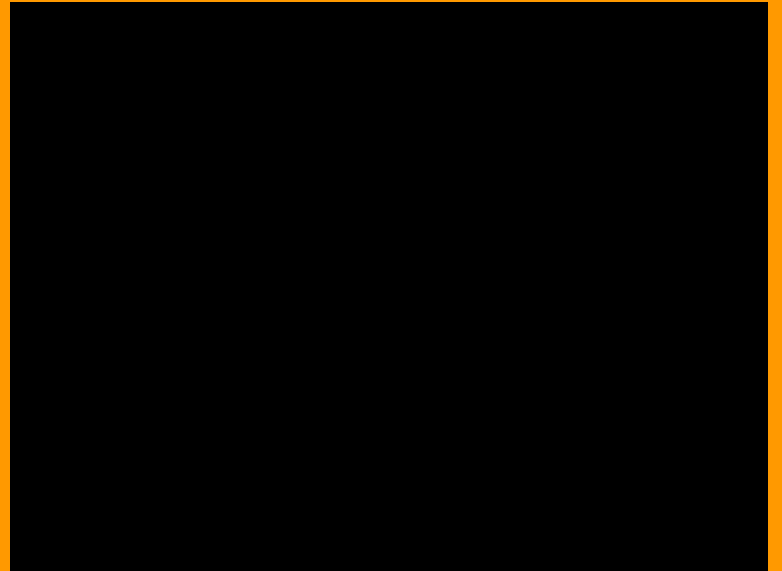
- `isWpa3SaeSupported()`
- `isWpa3SuiteBSupported()`
- `removeNetworkSuggestions`
(`List<WifiNetworkSuggestion`
`networkSuggestions`)*

Exemplo

Before



After



Talk is cheap show me the code!



WifiNetworkSuggestion

```
import android.net.wifi.WifiManager
import android.net.wifi.WifiNetworkSuggestion

val wifiManager = getSystemService(Context.WIFI_SERVICE) as WifiManager

val networkSuggestion = WifiNetworkSuggestion.Builder()
    .setSsid(quoteString(ssidText))
    .setWpa2Passphrase(quoteString(passwordText))
    .build()
```

Criar uma Rede - Antes do Q

```
private fun addNetwork(ssid: EditText, password: EditText) {  
    val ssidText = ssid.text.toString()  
    val passwordText = password.text.toString()  
  
    val wifiManager = getSystemService(Context.WIFI_SERVICE) as WifiManager  
  
    val wifiConfig = WifiConfiguration()  
    wifiConfig.SSID = quoteString(ssidText)  
    wifiConfig.preSharedKey = quoteString(passwordText)  
  
    Log.d("HSS", "Add network: ${wifiConfig.SSID}")  
    val result = wifiManager.addNetwork(wifiConfig)  
    Log.d("HSS", "Network was added: ${result != -1}")  
  
    viewAdapter.updateNetworks(getNetworks())  
    viewAdapter.notifyDataSetChanged()  
}
```

Criar uma Rede - Depois do Q

```
private fun addNetwork(ssid: EditText, password: EditText) {  
    val ssidText = ssid.text.toString()  
    val passwordText = password.text.toString()  
  
    val wifiManager = getSystemService(Context.WIFI_SERVICE) as WifiManager  
  
    Log.d("HSS", "How many network can I create? ${wifiManager.maxNumberOfNetworkSuggestionsPerApp}")  
  
    val networkSuggestion = WifiNetworkSuggestion.Builder()  
        .setSsid(quoteString(ssidText))  
        .setWpa2Passphrase(quoteString(passwordText))  
        .build()  
  
    Log.d("HSS", "Add network: $ssidText")  
    val result = wifiManager.addNetworkSuggestions(mutableListOf(networkSuggestion))  
    Log.d("HSS", "Network was added: ${result}")  
  
    viewAdapter.updateNetworks(getNetworks())  
    viewAdapter.notifyDataSetChanged()  
}
```

Deletar Redes - Antes do Q

```
remove.setOnClickListener {  
    Log.d("HSS", "Remove Network: $network")  
  
    val wifiManager = context.getSystemService(Context.WIFI_SERVICE) as WifiManager  
    wifiManager.removeNetwork(network.id)  
    networks.remove(network)  
    notifyItemRemoved(position)  
  
    Snackbar.make(it, "Removed Network: $network", Snackbar.LENGTH_LONG).show()  
}
```

Deletar Redes - Depois do Q

```
remove.setOnClickListener {
    Log.d("HSS", "Remove Network: $network")

    val wifiManager = context.getSystemService(Context.WIFI_SERVICE) as WifiManager

    val networkSuggestion = WifiNetworkSuggestion.Builder()
        .setSsid(quoteString(network.ssid.toString()))
        .build()

    val result = wifiManager.removeNetworkSuggestions(mutableListOf(networkSuggestion))

    Log.d("HSS", "Removed network status: $result")

    networks.remove(network)
    notifyItemRemoved(position)

    Snackbar.make(it, "Removed Network: $network", Snackbar.LENGTH_LONG).show()
}
```

Listar Redes - Antes do Q

```
private fun getNetworks(): MutableList<Network> {  
    val wifiManager = getSystemService(Context.WIFI_SERVICE) as WifiManager  
    val listWifiConfigs = wifiManager.configuredNetworks  
  
    return listWifiConfigs.stream()  
        .map { wifiConfig -> Network(wifiConfig.networkId, wifiConfig.SSID) }  
        .collect(Collectors.toList())  
}
```

Listar Redes - Depois do Q

```
private fun getNetworks(): MutableList<Network> {  
  
    if (checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION)  
        == PackageManager.PERMISSION_GRANTED) {  
        val wifiManager = getSystemService(Context.WIFI_SERVICE) as WifiManager  
        val listWifiConfigs = wifiManager.configuredNetworks  
  
        Log.d("HSS", "List of wifi $listWifiConfigs")  
  
        return listWifiConfigs.stream()  
            .map { wifiConfig -> Network(wifiConfig.networkId, wifiConfig.SSID) }  
            .collect(Collectors.toList())  
    } else {  
        if (shouldShowRequestPermissionRationale(Manifest.permission.ACCESS_FINE_LOCATION)) {  
  
            } else {  
                requestPermissions(arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), 302)  
            }  
        }  
    }  
  
    return mutableListOf()  
}
```

Outros - Antes do Q

```
connect.setOnClickListener {  
    Log.d("HSS", "Connect to network: $network")  
  
    val wifiManager = context.getSystemService(Context.WIFI_SERVICE) as WifiManager  
    val result = wifiManager.enableNetwork(network.id, true)  
  
    Log.d("HSS", "Network was enabled: $result")  
}
```


Outros - Antes do Q

```
disconnect.setOnClickListener {  
    val wifiManager = getSystemService(Context.WIFI_SERVICE) as WifiManager  
    wifiManager.disconnect()  
}
```

Outros - Antes do Q

```
private fun setWifiStatus(wifiWasClicked: Boolean) {  
    val wifiManager = getSystemService(Context.WIFI_SERVICE) as WifiManager  
  
    if (wifiWasClicked) {  
        wifiManager.isWifiEnabled = !wifiManager.isWifiEnabled  
    }  
  
    if (wifiManager.isWifiEnabled) {  
        wifi_on_off.setImageDrawable(ContextCompat.getDrawable(this, R.drawable.ic_wifi_black_24dp))  
    } else {  
        wifi_on_off.setImageDrawable(ContextCompat.getDrawable(this,  
            R.drawable.ic_portable_wifi_off_black_24dp))  
    }  
}
```

**Não existe mais a
possibilidade de
utilizar esses
métodos no
Android Q**

Palavras Chaves da Google

- Segurança
- Restrição
- Poder nas mãos dos usuários

Dúvidas?



THE DEVELOPER'S
CONFERENCE

Obrigado!!!

The Developer's Conference - São Paulo

17.07.2019

