I'm not robot
reCAPTCHA
Continue

How to turn on hotspot and wifi at the same time in android

How do i turn on wifi and hotspot at the same time. Can i turn on wifi and hotspot at the same time.

I don't know of any simple tools other than Android apps (that use Wi-Fi Direct) that use Wi-Fi and hotspot at the same time. None of the standard features currently found in custom ROMs (AFAIK). If you're familiar with the command line, you can do it manually as well. But you can't do it without a root.
What is required is that the handheld device supports the NL80211 driver, for example all new devices using the Qualcomm MSM chipset support this AFAIK. Redmi Note 4 (Mido) Xiaomi built with MSM8953. The kernel should also support this driver. Confirm: ~# zcat /proc/config.gz | grep config_cfg80211 config_cfg80211 = y ~#lshw | GREP Wireless = Configuration: Broadcast = YES Driver = WCNSS_WLAN multicast = YES Wireless = QCOM: 802.11n device must support AP mode and AP mode simultaneously. ~#Iw Phy | Grep -i2 'Permitted combinations of interfaces' Permitted combinations of interfaces: * #{Managed} <= 3, #{iBSS, ap} <= 1, #{p2p-callient, p2p-by} <= 1, total <= 3, total <= 3, fees #Channels <= 1 This means that your WiFi chip supports creating up to 3 interfaces, one of which can be in AP mode, while the others are in controlled or P2P mode, and on the maximum channel are supported. If the channel value is 2, both interfaces can be used on different channels. However, it is recommended to use the same channel to avoid interference. The default Wi-Fi interface on Android devices (typically WLAN0) always runs in managed mode. We will create a virtual wireless interface running in AP mode. Linux: IW, IP, IPTABLES, HOSTAPD, DNSMASQ, etc. You can also use WPA_SUPPLICANT instead of hotapd with a slightly different configuration. These binaries are usually related to Android. However, there are changes from Google and vendors to make these tools suitable for Android's needs. Therefore, they may not behave like standard Linux tools and in rare cases may need to be built from source. Wireless instruments should be updated enough to support the NL80211 driver. terminal emulator (the term is good)I have summarized all the steps in a shell script with a brief explanation for ease of use. You can put it in your $PATH for example. /system/bin/android_ap and run it directly: ~# android_ap start. You can also create a custom quick setting item for more convenient use. #!/system/bin/sh set -e #set -x ["$(id -u)" != 0] && echo 'Not running as root!' && exit SSID=MyAP # set your desired character string (avoid spaces and non-ASCII characters) PASSCODE=foobarfoobar # set your desired character string (8 to 63 characters) WIFI_INTERFACE=wlan0 # set to match your device (lshw | grep - A10 Wireless | grep 'logical name') SUBNET=192.168.42 # # must be different from WIFI_INTERFACE AP_INTERFACE=${WIFI_INTERFACE}-AP IP=${SUBNET}.1 DIR=/data/local/tmp/$AP_INTERFACE USING () { echo ' Usage:' printf '\t%s' "$(basename "$0") start|stop" exit } STOP() { # hope there are no other instances of the same daemon pkill -15 hostapd dnsmasq # delete iptables rules iptables -D INPT -i $AP_INTERFACE -p udp -m udp --dport 67 -j ACCEPT iptables -t nat -D POSTROUTING -s ${SUBNET}.0/24 ! -o $AP_INTERFACE -j MASQUERADE iptables -D FORWARD -i $AP_INTERFACE -s ${IP}/24 -j ACCEPT iptables -D FORWARD -i $WIFI_INTERFACE -d ${SUBNET}.0/24 -j ACCEPT # remove AP Show interface IP link | grep "${AP_INTERFACE}:" && iw $AP_INTERFACE del rm -rf $DIR } >/dev/null 2>&1 CHECK() { for binary in iw ip iptables hostapd dnsmasq; do what $binary >/dev/null && Continue exit done # This check is necessary if you need to use one channel when iw dev $WIFI_INTERFACE link | grep -q '^Not Connected' and then repeat "Connect to Wi-Fi first to share internet". stop when! i fi | grep -iqIs '{.*managed.*AP.*}' && ! i fi | grep -iqE '{.*AP.*managed.*}' and then repeat "AP mode not supported". exit fi } CREATE_AP() { if ! In interface iw dev $WIFI_INTERFACE add $AP_INTERFACE, type __ap and retry "Failed to create hotspot". # :(exit fi } FIND_CHANNEL() { # Find out which Wi-Fi channel CHANNEL="$(iw $WIFI_INTERFACE scan | grep -C5 "$(iw $WIFI_INTERFACE|. GREP SSID | Cut -D: -F2-) "| GREP -I -Canal | Tail -C3)" If [-z "$ channel"] to echo "I can't find information on the sewers. Are you connected to the Wi -Fi?" Stop fi output # If more than 1 channel is supported, use any frequency [! -Z "$ channel"] || Channel = 11} Add_ip_Route () { # activate the interface and insert an IP connection. For Android # because in the root table Route $ wi -fi_ interface priority (IP route) # and all the data traffic $ ap_ interface IP -Route $ {subset .conf # Name Network SSID = $ ssid # network interface on which to listen to interface = $ ap_ interface # driver wi-fi = nl80211 # Wlan channel to use channel = channel to be used = channel =. $ Channel # ser operating mode whose frequency for hw_mode = g # wireless protected access (wpa) wpa = 2 # passphraza for safe access wpa_pasphrase = $ passcode # protocol wpa_key_mgmt = wpa-piof # can be used. Options such as MTU, Beacon_int, IEEE80211N, WOWLAN_TRIGGERS (if supported) # For a better design and options such as *_Pairwise for greater safety} Internet_Share () { # IP -forwarding Echo 1>/proc/sys/net/ip_forward # Instrada e Enable requests through the IPtables Firewall -T Nat -i Potrouting -S $ {subnet} .0/24! -o $ ap_ interface -J mask iptables -i vorwärts -i $ ap_ interface -S $ {ip}/24 -J accept iptable -i $ wifi_ interface -d $ {# cat cat << -oef> $ say/dnsma .conf# We do not need a DNS server, only dhcp port = 0 # reading ap interface interface = $ ap_ interface listen-addres = $ IP # bind-interfaces #are #are # # #