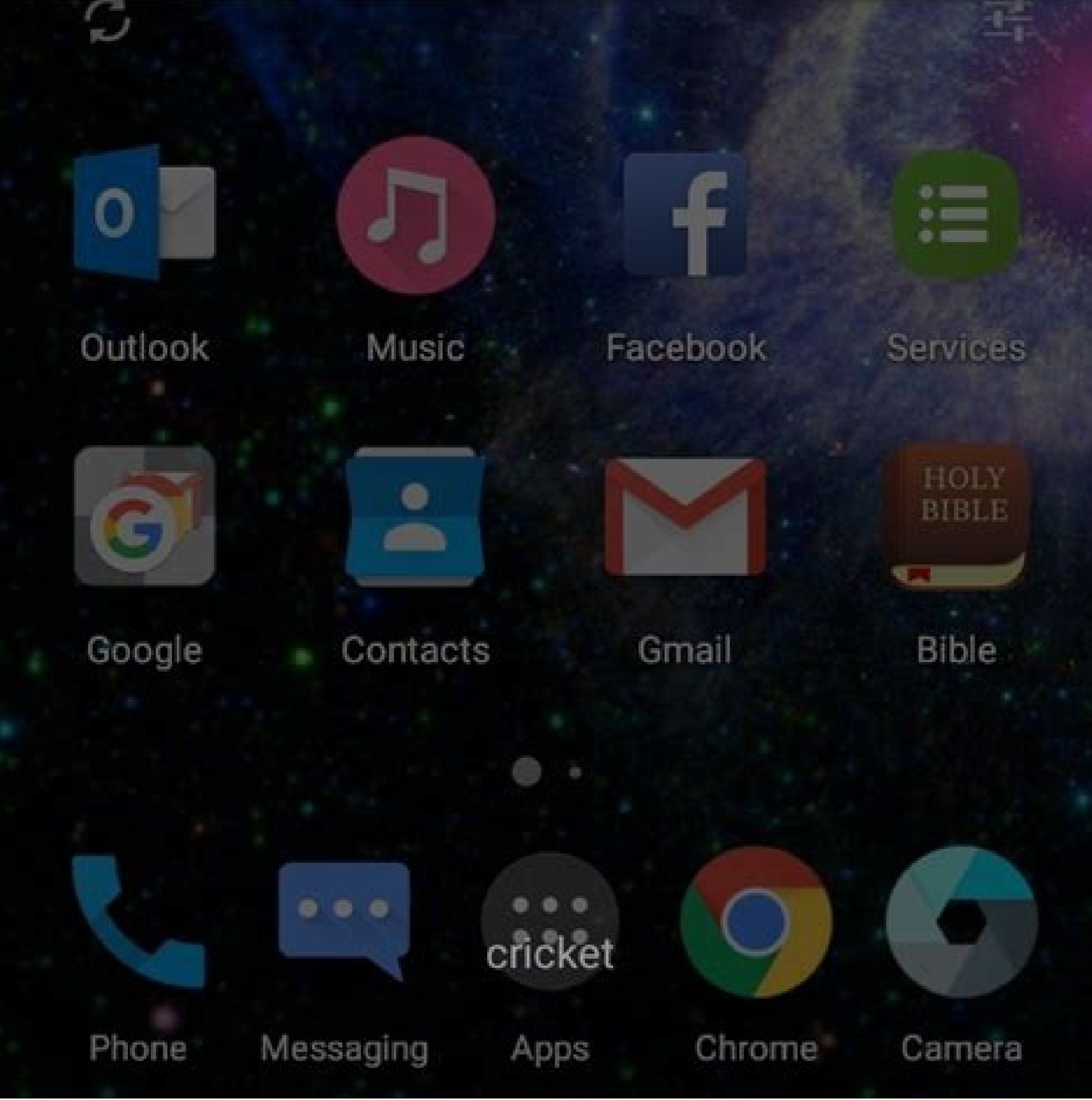
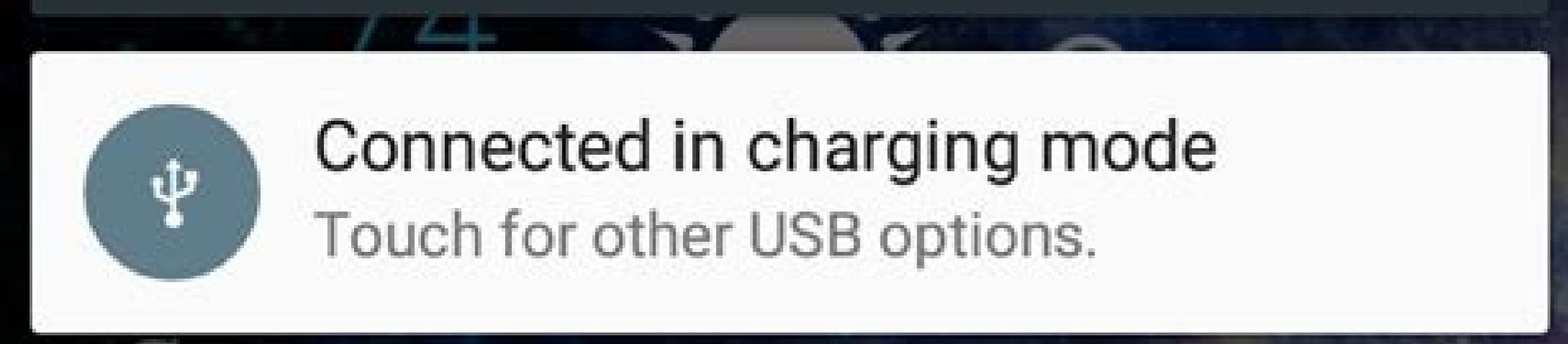
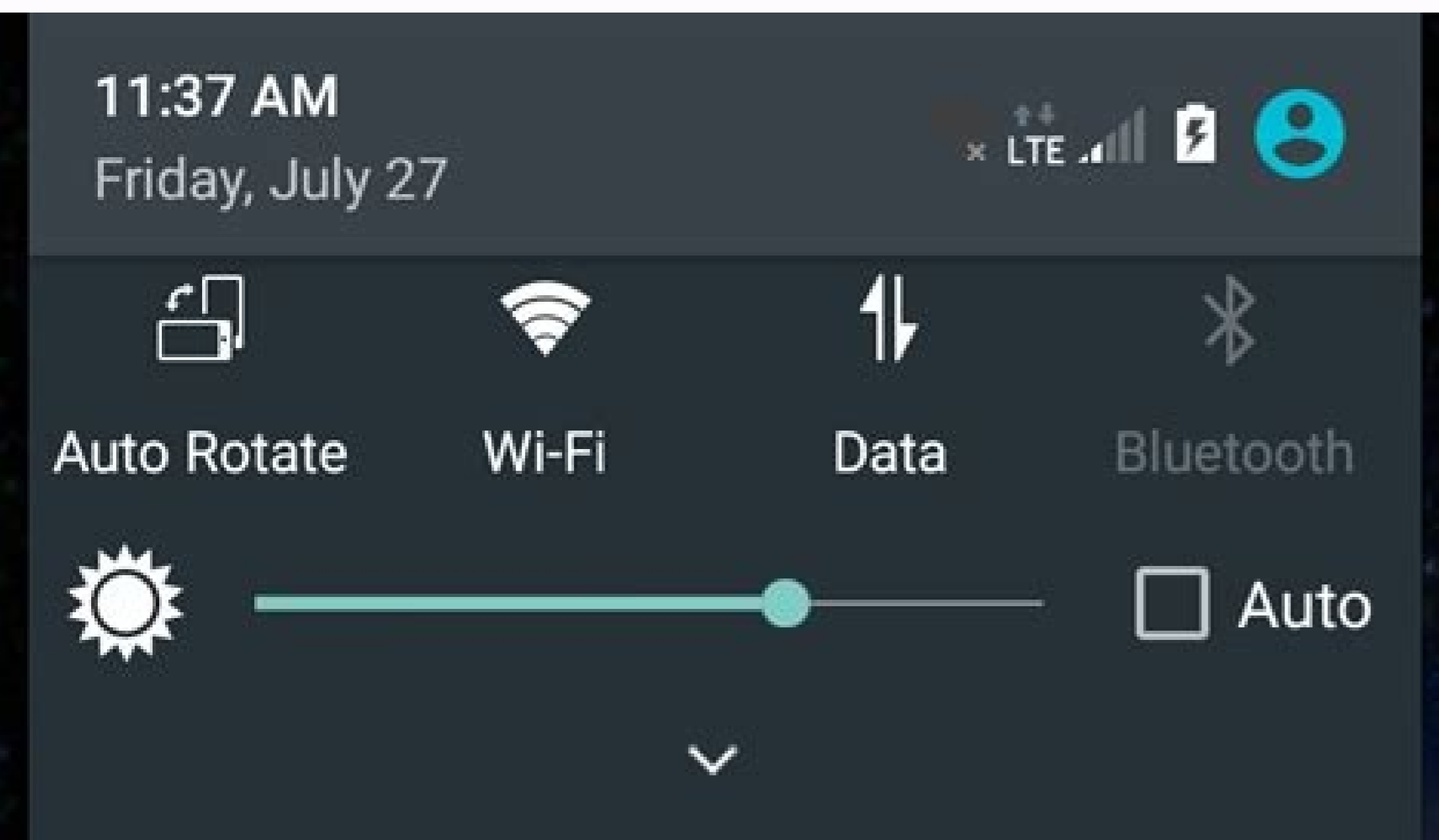
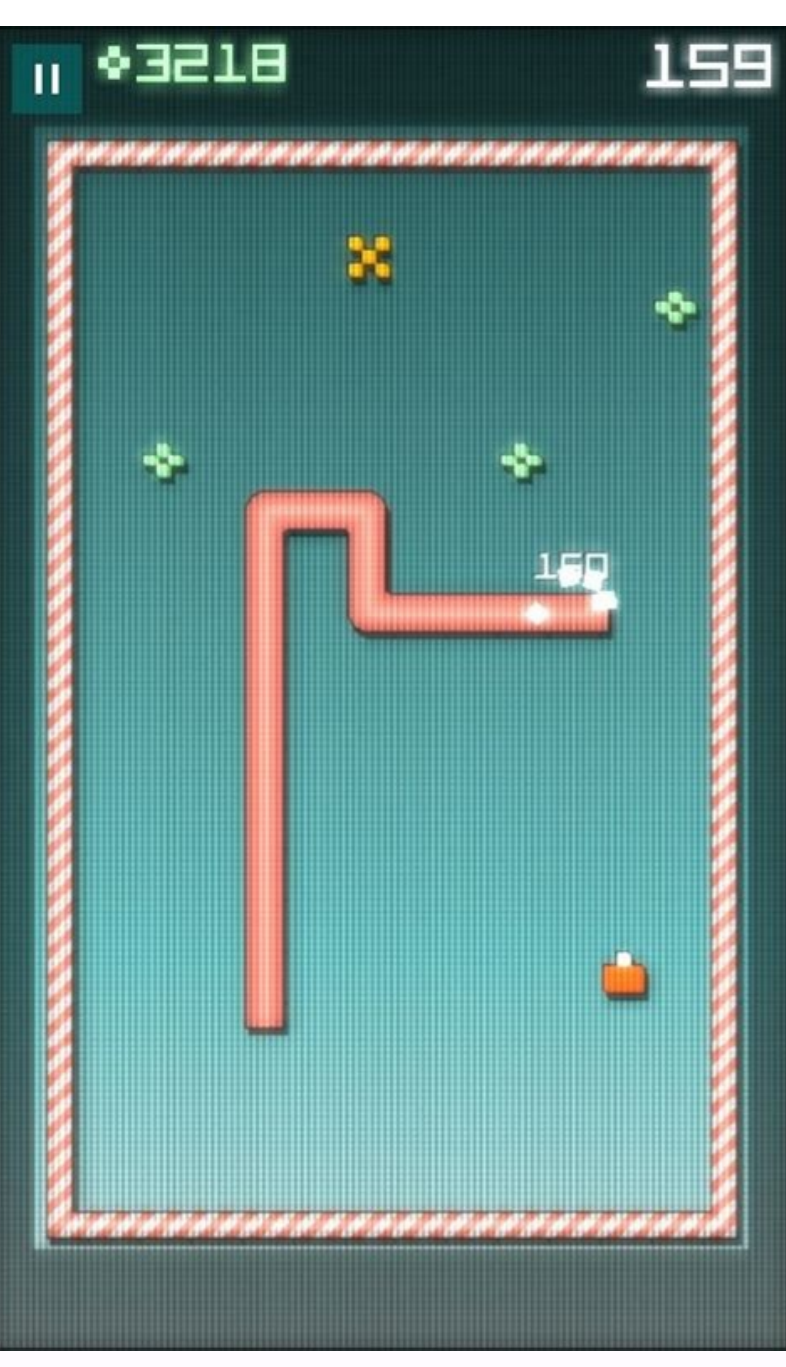
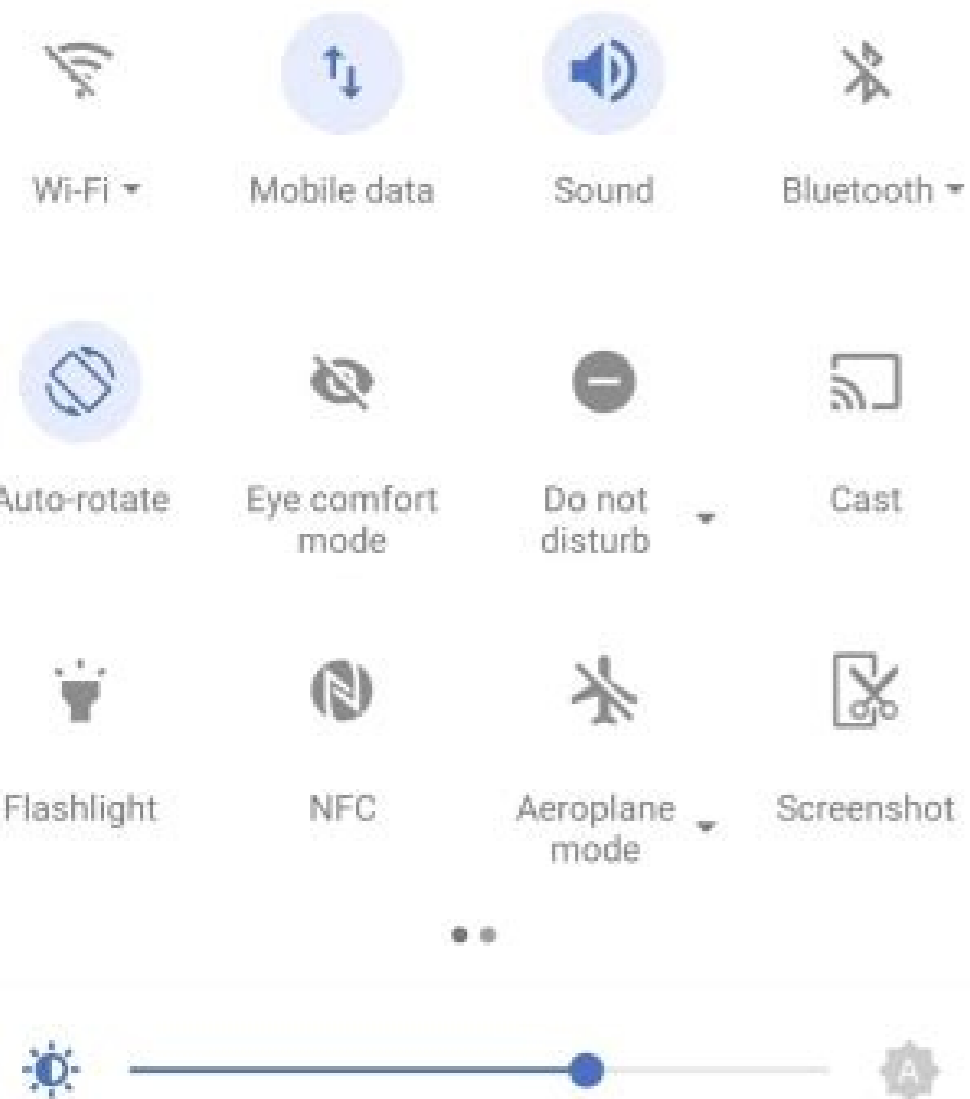


Continue



10:00

Wed, 26-08-2020



10:00

Settings



How to set internet on phone. How to set up internet on android phone. How to put internet settings on android phone.

Download Article Download Article Using Wi-Fi | Using a hotspot Connecting your Android phone to a network is very easy and you can do it in two ways: by connecting to your own Wi-Fi connection or by connecting to a device, access point. A hotspot is similar to Wi-Fi except that the network is provided by a phone and not a modem. 1 Unlock and open the device. If you have a password, enter it now. 2 Go to Settings. Find the gear icon and tap on it. 3 Tap Wi-Fi. The WLAN menu opens. 4 Turn on Wi-Fi. A list of nearby networks will appear. If you're already connected to one of these networks, your phone will connect automatically. If you haven't joined a network yet, you'll need to add a new one. Proceed to the next step to add unrecognized networks. 5 Enter the SSID/name of your network. 6 Enter the network security type. This is usually the WEP value. 7 Enter the network password. After that, your phone should connect to the network. 1 Go to Settings menu 2. Go to Wi-Fi. Open the Wi-Fi menu by tapping this option in the settings menu. 3 Turn on Wi-Fi. When the hotspot is detected, your phone will automatically connect to the hotspot. This is only possible if the provider has allowed it. If the access point is not detected, you must add it. Proceed to the next step to add an unknown network. 4 Enter the access point name. 5 Enter WEP as the security type for the access point. 6 Enter the password of the access point provider. If the hotspot is enabled, the phone should connect automatically. Advertisement Add a new question Question How do I know if my mobile device works with a prepaid SIM card? You can use the internet to check this by opening any application that works on the internet, e.g. B. WhatsApp, Spotify, YouTube, a browser for any online search, etc. If you don't see any results, you know the internet isn't working, not working or working on it To use the internet, you must have credit on your SIM card or have a paid data plan to use the internet. My phone is brand new and I want it to connect to the internet using mobile data. The network type icon does not appear. How can I solve the problem? You have to pay for an internet subscription - 4G, 5G or other depending on the capacity of the phone. You can call your SIM card/service provider customer service number to find out more. It will tell you if you need to top up your 4G data plan, if you have data or credit available, etc. Or you can search for data plans online and ask your dealer to activate your plan. There is a number you can call for a set of instructions. Each service provider has its own number. Start by finding yours. Q. How do I find my network settings? If you call your service provider's customer service number and ask for network settings, they usually receive them via SMS.

They also send you a password when prompted to use your device. View More Answers Ask a Question [Wikipedia-like "wiki,"](#) meaning many of our articles are co-authored by 9 people worked on simultaneously. You can change network settings like automatic connections, traffic metering, proxy settings and more. Control data usage with metered Wi-Fi. Connect your phone to a Wi-Fi network. Open the Settings app on your phone. Tap Network and Internet. If you still can't find it, ask your device manufacturer for help. Tap on the Wi-Fi network you are connected to. Tap Network usage. Consider as metered. If your network has a data limit, you can set itWi-Fi in the subway. When your network is measured, you can have better control over how much data your phone uses for downloads and other apps. Finding your phone's MAC address Open the Settings app on your phone. Tap About phone. Scroll down to "Wi-Fi MAC Address". Phones running Android 10 and above have a different MAC address. To find it: Turn on Wi-Fi. Open the Settings app on your phone. Click Network and Internet. Next to Network, click Settings. Scroll down to "Random MAC Address". Advice. If you are setting up parental controls for your network, please provide both MAC addresses. Private DNS Important! By default, the phone uses private DNS with all networks that can use private DNS. We recommend leaving private DNS enabled. Enable or disable private DNS or change its settings. Open the Settings app on your phone. Click Network and Private Internet DNS. Choose one of the following options: Off Automatic private DNS provider hostname Tip. Private DNS helps protect only DNS queries and responses. He cannot protect anything else. Changing other Wi-Fi settings Open the Settings app on your phone. Click Network and Internet. Click "Network Settings" at the bottom. Tap an option. Depends on phone and android version. Auto turn on Wi-Fi: Set Wi-Fi to turn on automatically near saved networks. Read about saved networks. Open Network Notifications: Get notified when automatic connections to high-quality open networks are not available. Other installation certificates. Digital certificates can identify your phone. Read about certificates. Wi-Fi Direct: Let your phone connect wirelessly to other devices that can use Wi-Fi Direct. If you're using an earlier version of Android, you can also select one of the following options: WPS button: Turn on Wi-Fi Protected Setup (WPS) for a network that supports WPS. WPS PIN entry: Enter the personal identification number (PIN) of Wi-Fi Protected Setup (WPS). Related Resources Connect to Wi-Fi Networks Tether to a Wi-Fi Hotspot Get answers from community experts if you haven't bought yethphone from SaskTel or cannot receive over-the-air programming, you may need to manually enter network information and settings. If you need assistance, visit a SaskTel store or authorized reseller. Fees may apply for manual programming assistance. NOTE. A SaskTel SIM card must be installed to use these settings. Pull down the notification bar and tap Settings (gear icon). Tap Connections > Mobile networks > Hotspot names. If there are no access point names in the list, click Add (top right). Enter your SaskTel browser and MMS settings (required for multimedia messaging to work). Check your internet connection: Open your phone's internet browser and go to the website. Test SMS: Open SMS on your phone and try to send a picture. Calling 1.800.SASKTEL (1.800.727.5835) (hours support, menu options) is still not working. Android is owned by Google, Inc. trade mark. Updated on January 18, 2022 at 15:21. In order to perform network operations in the application, the following permissions must be present in the manifest: Both internet permissions, both ACCESS_NETWORK_STATE permissions are normal permissions, meaning they are granted during installation and do not need to be requested at runtime. Best Practices for Secure Network Communications Before you add network functionality to your application, ensure that your application's data and information is secure while it is being sent over the network. To do this, follow these network security guidelines: Minimize the amount of sensitive or personal user data sent over the network. Send all traffic from your application over SSL. Consider creating a network security configuration that allows your application to trust custom certificate authorities (CAs) or restrict the set of system CAs it trusts for secure communication. For more information about using a secure networksee network security tips. Select an HTTP client Most web applications use the HTTP protocol to send and receive data. The Android platform includes an HttpURLConnection client that supports TLS, streaming upload and download, configurable timeouts, IPv6, and connection pooling. In this section, we use the Retrofit HTTP client library, which allows you to create an HTTP client declaratively. The update also supports automatic serialization of request structures and deserialization of response structures. Resolving DNS queries Android 10 devices (API level 29) and higher have built-in support for custom DNS lookups using both plain text and DNS-over-TLS lookups. The DnsResolver API provides a generic asynchronous name resolution mechanism that can search for SRV, NAPTR, and other types of records. Note that parsing the responses must be done by the application. On devices running Android 9 (API level 28) and below, the platform DNS resolver only supports A and AAAA records. It looks for IP addresses associated with a name, but does not support other record types. For NDK-based apps, see android.res.nsend. Encapsulating network operations with a repository To simplify the process of implementing network operations and reduce code duplication in different parts of an application, you can use the repository design pattern. A repository is a class that handles data operations and provides a pure API abstraction for specific data or resources. You can use Retrofit to declare an interface that specifies the HTTP method, URL, arguments, and response type for network operations, as in the following example: interface UserService { @GET("/users/{id}") suspend fun getUser(@ Path(" id") id: String): User } public interface UserService { @GET("/user/{id}") Call getUserById(@Path("id") String id); } In the Repository class, functions can encapsulate network operations and provide their results. This encapsulation ensures that the components calling the repository do not need it.how data is stored. All subsequent changes to the data store are also isolated by the Repository class. class constructor UserRepository (private userService: UserService) { fun expose getUserById (id: String): user { return userService.getUser (id) } } class UserRepository { private UserService userService; public UserRepository (UserService userService) { this.userService = userService; } public Call getUserById (string id) { return userService.getUser (id); } } To avoid creating an unresponsive user interface, do not perform network operations on the main thread. By default, Android requires a thread other than the main UI thread to perform network operations; if not, a NetworkOnMainThreadException is thrown. In the UserRepository class shown in the previous code example, the network operation is not actually started. The caller of UserRepository must implement the stream using coroutines or the enqueue() function. Saving configuration changes When a configuration change occurs, eg. screen rotation, your fragment or activity will be destroyed and recreated. Any data not stored in the state of your fragment or activity instance, which should contain only a small amount of data, will be lost and you may have to make network requests again. You can use ViewModel to have your data persist configuration changes. ViewModel is a component designed to store and manage data related to the user interface as part of its life cycle. Using the previously created UserRepository, the ViewModel can make the necessary network requests and provide the result for your fragment or activity using LiveData: MainViewModel class constructor(savedStateHandle: SavedStateHandle, userRepository: UserRepository) : ViewModel() { private val userId: String = saveStateHl["uid"]?.throw IllegalArgumentException("No user ID") private val user = MutableLiveData() val user = user as LiveData init { viewModelScope.launch { { // Calling the repository is safe because it defers execution // from the main thread val user = userRepository.getUserById(userId) user.value = user } catch (error: Exception) { // show error message to user } } } } class MainViewModel extends ViewModel { private final MutableLiveData user = new MutableLiveData(); LiveData user = (LiveData) user; public MainViewModel(SavedStateHandle savedStateHandle, UserRepository userRepository) { String userId = savedStateHandle.get("uid"); Call userCall = userRepository.getUserById(userId); userCall.enqueue(new Callback() { @Override public void onResponse(Call call, Response response) { if (response.isSuccessful()) { _user.setValue(response.body()) } } @Override public void onFailure(Call call, Throwable t) { // Show the user an error message } }); } } For more information on this topic, see these related guides: Guides: