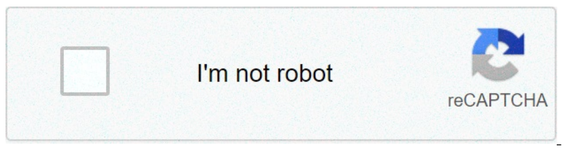

Continue

Android get current location using network provider

Public Class LocationManager extends Object Java.lang.Object Ã, Ã, Ã, ã, â³ Android.Location.LocationManager This class allows access to system location services. These services allow applications to obtain periodic updates geographical position of the device, or to be notified when the device enters the proximity of a specific geographical place. Unless otherwise specified, all Methods Location API require the permissions manifest.permission.access_coarse_location or manifest.permission.access_fine_location. If the application has only the maximum permission then the suppliers will still return the location results, but the exact location will be blurred at a coarse level of accuracy. Requires the function # feature_location packageManager which can be detected via packageManager.HassystemFeature (string). Boolean addgpsstatusListener (GPSStatus.Listener listener) This method is advised against API level 24. Application RegisterGnsSstatusCallback (Android.Location.Gnsstatus.Callback) instead. It is no longer supported in S and superior applications. Boolean AddNameAlistener (OnnMeMessagelistener Listener, Handler Manager) adds a NMEA listener. Boolean AddNameAlistener (OnnMEAMessagelistener listener) This method is not recommended at API level 30. Use AddNameAlistener (Android.Location.onNmeMessagelistener, Android.os.Handler) or AddNameAlistener (java.util.concurrent.executor, Android.Location.onnmaMessagelistener) instead. Boolean AddNameAlistener (GPSStatus.Nmealistener listener) This method is advised against API level 24. Use AddNameAlistener (GPSStatus.Nmealistener) instead. AddNameAlistener Boolean (executor performer, OnnMaMessagelistener listener) adds a NMEA listener. ADDPROXIMITITYALERT Empty (double latitude, double longitude, floating radio, long deadline, pendingIntent pendingInt) Sets a proximity notice for the position indicated by the position (latitude, longitude) and given ray. Empty AddTestProvider (String Supplier, Property ProviderPerties) creates a test position provider and adds it to the active supplier series. Empty AddTestProvider (String Supplier, ProviderProperties Property, Set ExtraAttributiDags) Create a test position provider and add to the active supplier series. Empty AddTestProvider (supplier String, RequiresNetwork Boolean, Boolean Requiredatellite, Boolean Requesh, Boolean HasmonetaryCost, Boolean SupportSalititude, Boolean SupportsSpeed, Boolean SupportSbearing, Int PowerUsage, Int Precision) Create a test position provider and add to the set of active suppliers. Empty ClearTestprovidenabled (supplier string) This method was not recommended at API level 29. Use SettestProviNeaBled (Java.lang.String, Boolean), on the other hand. ClearTestProviderlocation Vacuum (String Supplier) This method was not recommended at API level 29. This method has always been a no-op, and can be removed in the future. Empty Cleartestproviderstatus (String Supplier) This method is advised against API level 29. This method has no effect. List GetallProviders () Returns a list of names of all available position providers. String GetBestProvider (criteria criteria, Boolean enabledonly) Returns the name of the provider that best meets the criteria indicated. GetCurrentLocation Vacuum (String Supplier, LocationRequest LocationRequest, Cancertionsignal Cancellationsignal, Executor Executor, Consumers Consumer) Returns a single current correction position from the date provider based on the locationrequest date. Empty GetCurrentLocation (String Supplier, Cancellationsignal Cancellationsignal, Executor Executor, Consumers Consumer) Returns asynchronously a single current correction position Data Provider. List getgnsantennaInfos () Returns the current list of GNSS antenna information, or null if unknown or not supported. GNSSCAPABILITIES GETGNSSCAPABILITIES () Returns the supported capacity of the GNSS chipset. String GetGnSshardWaremodelName () Returns the model name (including supplier and hardware / software software GNSS hardware drivers, or null if this information is not available. getgnssyearfhardware int () returns the year model of GNSS hardware and software builds, or 0 if the model year is before 2016. GPSSTAtus GetGpsStatus (GPSSTATUS State) This method was deprecated in API level 24 . the GPSStatus APIs are deprecated, instead use the GNSSTATUS API. No longer supported in apps that turn me S and above. GetLastKnownLocation Position (String provider) Gets the last known position as the supplier, or NULL if this is not the last known position. LocationProvider GetProvider (String Provider) This method has been deprecated in API level 31. This method has no way of indicating that the properties of a provider are not unknown, and then can return incorrect results on rare occasions. Use instead GetProviderProperties (java.lang.String). ProviderProperties GetProviderProperties (String provider) returns the properties of the specified vendor or null if the properties are currently unknown. List GetProviders (Enabledonly Boolean) Returns a list of names of available location providers. List GetProviders (Criteria criteria, Boolean enabledonly) Returns a list of localization vendors names available that meet the specified criteria. HasProvider Boolean (String provider) Returns true if there is a localization vendor indicated on this device, regardless of whether it is currently enabled or not. IsLoonnedEnabled Boolean () returns the current enabled / disabled state. ISProviderEnabled Boolean (String provider) returns the current enable / disable state of the data provider. Boolean RegisterAnNennainFolistener (Executor Executor, GnsSantennainFo.Listener listener) registers a listener of information on the antenna GNSS antenna will receive all changes to information. Boolean RegisterGnssMeasurementsCallback (Executor Executor, GnssMeasurementsEvent.Callback Recall) Record a GNSS measurement callback. Boolean RegisterGnsssMeasurementsCallback (GnssmeasurementsEvent.Callback Recall) This method has been deprecated in API Level 30. Use RegisterGnSSementSuuremeNtsCallback (GNSSmeasurementEvent.Callback, Manager) or RegisterGnsSeasurementsCallback (EXECUOR, GNSSmiSulementsEvent.Callback) instead. Require Manifest.Permission.Access_Fine_Location Boolean RegisterGnssSeasurementsCallback (Request GNSSmiSeurementRequest, Executor Executor, GnssMeasurementsEvent.Callback Recall) Record a GNSS measurement callback. Boolean RegisterGnssMeasurementsCallback (GNSSmiSulementsEvent.Callback Callback, Handler Handler) registers a callback GNSS measurement. Boolean RegisterGnssSnavigationMessageCallback (GNSSNavigationMessage.Callback Callback, Handler Handler) records a GNSS navigation message callback. Boolean RegisterGnssSnavigationMessageCallback (Executor Executor, GNSSNavigationMessage.Callback Recall) Record a GNSS navigation message callback. Boolean RegisterGnssSnavigationMessageCallback (GNSSNavigationMessage.Callback callback) This method was deprecated in API level 30. Use the registroGnssNavigationMessageCallback (Android.Location.GNSSNavigationMessage.Callback, Android.os.Handler) or RegisterGnssNavigationMessageCallback (Java.util.concurrent.executor, Android. Location.GNSSNavingMessage. .Callback) instead. Boolean RegisterGnssStatusCackback (GNSSSTATUS.Callback Recall) This method was deprecated in API level 30. Use RegisterGnssStatusCallback (Android.Location.GnssStatus.Callback, Android.os.Handler) or RegisterGnssStatusCackLack (Java.util.concurrent.executor, Android.Location .GnssStatus .Callback) instead. Boolean RegisterGnssStatusCallback (GNSSTATUS.Callback Callback, Handler Handler) registers a callback been GNSS. Boolean (Executor executor, gnsstatus.callback callback) Record a GNSS status callback. Void RemovegputStatusListener (GPSStatus.Listener Listuder) This method has been deprecated in the API level 24. Use unregistergnstTatusCallback (Android.Location.GnsSstatus.Callback) instead. No more supported in targeting targeting apps And above. Void RemovenMealistener (Onnmeassagelistener Listup) removes a NMEA listener. Void RemovenMealistener (GPSStatus.Nmealistener listener) This method has been deprecated in the API level 24. Instead use RemovenMealistener (Android.Location.onNmemessagelistener). Void RemoverProximyalert (Intent Intent) removes the proximity notice with the indicated student. Void RemoveTestProvider (String Provider) Removes the test position provider with the specified name or there is nothing if there is no test position providers. Void RemovepDates (LocationListener Listener) removes all position updates for the specified locationListener. VoID RemovepDates (Pendingint PendingInT) removes position updates for passing. Void Requetflush (String Provider, LocationListener Listener, Int Requercode) requires that the specified supplier waves any discounted places to listeners. Void RequetFlush (String Provider, PendingIntent PendingIntent, Int Remerscode) requires the specified supplier invalidates the listings discounts to listeners. Void RequestloonationUpdates (String Provider, Long Mintimemms, Float Mindistancem, LocationListener listener) Register for position updates from the specified supplier with the specified topics and a callback on the caller thread looper. Void RequestlocationUpdates (Long Mintimems, Float Mrosofancem, Criteria Criteria, LocationListener Listener, Looper Looper) This method has been deprecated in the API level 31. Use RequestolCationUpdates (Java.lang.String, Long, Float, Android.Location.LocationListener, Android. OS. Looper) Instead, select a supplier explicitly. Void RequestloonationUpdates (String Provider, Long Mintimems, Float Mindistancem, LocationListener Listuper, Looper Looper) Register for position updates from the supplier specified with the indicated topics and a callback on the specified looper. Void RequestLoonationUpdates (String Provider, Long Mintimems, Float Mindistancem, Executor Executor, Listenter LocationListener) Register for position updates using the named supplier and call back to the specified executor. Void RequestloonationUpdates (String Provider, LocationRequest LocationRequest, PendingIntent PendingIntent) Register for position updates from the specified supplier, using a locationrequest and callbacks provided through the PendingInt supplied. Void RequestLoonationUpdates (String Provider, LocationRequest LocationRequest, Executor Executor, Listener LocationListener) Register for position updates from the specified supplier, using a locationrequest and a callback on the specified executor. Void RequestlocationUpdates (Long Mintimems, Float Momofancem, Criteria, Criteria, Pendants Waiting) This method has been deprecated in the API level 31. Use RequestLocationUpdates (Java.lang.String, Long, Float, Android.App.pendingintent) instead of selecting Explicitly a supplier. Void RequestlocationUpdates (Long Mintimems, Float Mindistancem, Policy criteria, execution performer, LocationListener listener) This method has been deprecated in the API level 31. Use RequestolocationUpdates (java.lang.String, Long, Float, Java.util.Concurrent.Executor , Android. Location.LocationListener) Instead to explicitly select a supplier. Void RequestLoonationUpdates (String Provider, Long Mintimems, Float Mindistancem, PendingIntent PendingIntent) Register for position updates using the named supplier and callbacks provided through the PendingIntent supplied. Void RemersingleUpdate (Laces Supplier, PendingIntent PendingInt) This method has been deprecated in the API level 30. Use GetCurrentLocation (Java.lang.String, Android.os.Cancellationsignal, java.util.concurrent.executor, java.util.function.consumer) Instead since it does not carry a risk of extreme drainage of the battery. Void requestingleupdate (criteria policies, pendingintent pendingintent) This method has been deprecated in the API level 30. Use GetCurrentLocation (Java.lang.String, Android.S.Cancellationsignal, java.util.cancellationnignal, java.util.concurrent.executor, java.util.function.consumer) instead as it does It brings a risk of extreme battery discharge. Void RemersSingleUpdate (String Provider, LocationListener Ascoltore, Looper Looper) This method was deprecated in API level 30. Use GetCurrentLocation (java.lang.String, Android.os.Cancellationsignal, Java.util.CancellationSignal, java.util.concurrent. executor, java.util.funzionamento. Consumer) instead because it does not carry a risk of extreme drain on the battery. Void RichiestaSlePappdate (criteria, LocationListener Ascoltore, Looper Looper criteria) This method was deprecated in API level 30. Use GetCurrentLocation (java.lang.String, Android.os.Cancellationsignal, Java.util.Cancellationnignal, java.util.concurrent. executor, java.util.function.executor, java.util.functionign. Consumer) instead because it does not carry a risk of extreme drain on the battery. Boolean SendExtracommand (Provider String, String Command, Bundle Extras) Send additional commands to a localization vendor. Void SettesProviderNesabled (Provider String, Boolean Enabled) Set the testing vendor specified to be enabled or disabled. Void SettestProviderlocation (String Provider, location Position) Sets a new location for the specified test provider. Void SettestProviderstatus (Provider String, Int Status, Bundle Extras, Long UpdateTime) This method was deprecated in API level 29. This method has no effect. Void not registrantentennainfolidităÂ (GNSSASTENNAINFO.Listener Listener) Unrugisters A listener Info GNSS Antenna. Void UnregisterGnSsMeasurementsCallback (GNSSmiSulementsEvent.Callback Recall) Unrugisters A GPS measuring callback. Void UnregisterGnssNSNavigationMessageCallback (GNSSNavigationMessage.Callback callback) Unrugisters A GNSS navigation message callback. Void UnregisterGnssStatusCallback (GNSSTATUS.Callback Recall) Removes a status callback GNSS. From the Java.lang.Object Object Clone () class creates and returns a copy of this object. Equal Boolean (object OBJ) indicates whether some other object is "equal to" this. Void finalizes () called by the garbage collector on an object when the waste collection determines that there are no more references to the object. FINAL CLASS GETCLASS () Returns the runtime class of this object. INT hashcode () Returns a hash code value for the object. The final flight notification () wakes up on a single thread that is waiting for the monitor of this object. Final Void Notifyall () Choose all threads waiting on the monitor of this object. String tostring () Returns a string representation of the object. Final vacuum Wait (long timeout, int Nanos) Does so that the current thread awaits until another thread invokes the notify () method or the notifyall method () for this object or some other threads interrupts the current wire or a certain Quantity of real time has elapsed. Final vacuum Waiting (long timeout) does so that the current thread awaits that the other thread calls the notify () method or the notifyALAL () method for this object or has spent a specified amount of time. Final vacuum Wait () Causes The current thread Wait until another thread invokes the Notify () method or the NotifyAll method () for this object. Cash static final String Public Fused_Provider standard fuse location provider name. If present, this provider can combine input from various other localization providers to provide the best possible correction of position. It implicitly used for all RichiestaLolocationUpdates API involving a policy. Constant value: static final String "merged" static String GPS_Provider standard GNSS location provider name. If present, this provider determines your location using GNSS satellites. The responsiveness and Position corrections may depend on the GNSS signal conditions. The extra package for the positions derived from this position provider can contain the following key / value pairs: satellites - the number of satellites used to derive the constant value of the correction: "GPS" String final static String Final String (Key_ProxityMity_Entering used for the Extra bundle holding a boolean indicate whether a notice of proximity is entering (true) or coming out (false) .. constant value: "Entering" added in API level 3 deprecated in API level 29 static static final public Key_status_changed This constant was deprecated in API level is 29. The state changes are deprecated and do not transmit more from Android Q onwards. This key is no longer in use. Key used for EXTRA package Keep a value of intact when transmitted a status change using a pendentente. Constant value: "Status" Final String String String Public Network_Provider Standard name of the network location provider. If present, this provider determines the position based on the proximity of access points of the Tower and WiFi phone. The operation of this provider may require a data connection. Constant value: static final String "Network" Passive_Provider A special localization provider for receiving positions without actively initiate a position correction. This location provider is always there. This provider can be used to passively receiving position updates when other applications or services they require without actually take positions for themselves. This will return the supplier only positions created by other suppliers. Constant value: public methods "passive" Public Void AddProximityAlert (double latitude, double longitude, float-range, long-term, pendententententententententente) set a proximity alert for the location provided by the position (latitude, longitude) and the given radius. When the device detects that has been entered or exited from the area that surrounds the position, will be fired the indicated message. The intent fired will have a boolean added added key key_proximity_enting. If set to true, the device is entering the proximity of the region; If false, it is output. Because of the approximate nature of the estimate of the position, if the device briefly passes the given area, it is possible that no intent will be fired. Similarly, an intent could be fired if the device passes very close to the date but it is not actually inserted. Prior to version 17 API, this method could be used with or Manifest.Permission.Access_Fine_Location Manifest.Permission.Access_coarse_location. From the API version 17 and onwards, this method requires Manifest.Permission.access_Fine_Location Autorix. Require Manifest.Permission.access_coarse_location or manifest.permission.Access_Fine_Location Parameters Double Latitude: The latitude of the central point of the view region Double Longitude: the longitude of the central point of the radius of floating alert region: the radius of the central point of the center point Allert Region in Meters long maturity: real-time deadline for this proximity alert in milliseconds, or -1 to indicate no expiration pending in size during expiration: a pendentente that is sent when the input or the output is detected the warning region. This value can not be null. Public Void AddTestProvider (String Provider, ProviderProproperties Properties, Set ExtraattributionTags) Create a test location providers and adds to the set of active suppliers. This provider will replace any supplier by the same name that exists prior to this call. String parameter provider: provider name This value can not be null. ProviderProperties Property: the property of the provider This value can not be null. of extraattributionTags Set: Additional tag allocation associated with this supplier, this value can not be null. AddTrestProvider Public Void (Provider String, Boolean RichiedeNetwork, Resomesessellatellite, ReceiCell Boolean, Boolean HasmonetaryCost, BOOLEAN SupportAltitude, BooleanSpeedsPeed, Boolean Supportbearing, Powerusage Int, Int Accuracy) Create a test location provider and adds to the set of active suppliers. This provider will replace Supplier with the same name that exists before this call. Added in API Level 3 deprecated in API Level 29 Public Void Cleartestproviderlocation (stringe supplier) This method has been deprecated in the API 29 level. This method has always been a no-op and can be removed in the future. Does nothing. Parameter provider string: This value cannot be null. Added in the API level 3 deprecated in API level 29 public public vacuum (Supplier String) This method was not recommended at API level 29. This method has no effect. This method has no effect the supplier status has been deprecated and is no longer supported. String supplier parameters: This value cannot be null. PUBLIC LIST GETALLPROVIDERS () Returns a list of names of all available position providers. All suppliers are returned, including those that are currently disabled. Return List List of provider names This value cannot be null. Public String GetBestProvider (criteria criteria, Boolean enabledonly) Returns the name of the provider that best meets the criteria indicated. Only suppliers who have permission to access by the caller will be returned. If several suppliers satisfy the criteria, the one with the best accuracy is returned. If no provider meets the criteria, the criteria are loosened in the following order: Power Request Bearing Speed Precision Altitude Notes that the monetary cost requirement is not removed in this process. Criteria parameter criteria: The criteria that must be combined This value cannot be null. enabledonly boolean: if true suppliers then only enabled are included returns string name of the provider that best suits the criteria, or null if no game generates illegalargumentexception if criteria is null public getcurrentlocation empty (supplier string, locationrequest locationrequest, cancellationsignal cancellationsignal, performer Testation executor, consumers Consumer) asynchronously returns a single current correction position from the date provider based on the locationrequest date. This can activate the sensors in order to calculate a new position, unlike GetlastknownLocation (Java.lang.String), which will only return a solution in the cache, if available. The callback date is called only once, both with a valid position or a null position if the provider is unable to generate a valid path. A client can provide an optional erasignal. If this is used to cancel the operation, not called up should be expected after cancellation. This method can return positions of the very recent past (of the order of a few seconds), but will never return old offices (for example, a few minutes or older). Customers can count on the guarantee that if this method returns a position, which will represent the best estimate of the position of the device in the present moment. Customers calling this method from the background can notice that the method fails to determine a valid fixation position most of the time, while in the foreground. Background applications can be choked in their positions position to some extent. The date required warehouse can be used to provide suggestions on how a cool place is calculated if necessary. In particular locationrequest # GetDurationMillis () can be used to provide maximum duration allowed before failing. The system will always be crowned the maximum amount of time required for the current position can work for a reasonable value (less than a minute for example) before the request failed. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Supplier String: A supplier listed by the GetallProviders () This value cannot be null. LocationRequest LocationRequest: The position request containing pink parameters This value cannot be null. Cancellationsignal Cancellationsignal: An optional signal that allows you to cancel this call This value can be null. Executor performer: callback willow place this executor this value cannot be null. Callback and listener Events are sent This performer, providing an easy way to check which wire is used. To send events through the main application thread, you can use Context.getMainExecutor (). For shipping events through a shared thread pool, you can use asynctask # thread_pool_executor. Consumer consumers: callback invoked both with a location or null this value cannot be null. Public void void (String Provider, Cancertionsignal Cancellationsignal, Executor Executor, Consumer Consumer) Return asynchronous asynchronously a single correction of the current position from the specified supplier. See getcurrentlocation (java.lang.string, android.location.locationsrequest, android.os.cancellationsignal, java.util.concurrent.executor, java.util.function.consumer) For more information. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Pronters String String: A supplier listed by GetallProviders () This value cannot be null. Cancellationsignal Cancellationsignal: An optional signal that allows you to cancel this call This value may be null. Executor Executor: callback will take place on this executor this value cannot be null. Callback and Listener events are sent via this performer, providing an easy way to check which thread is used. To send events through the main application thread, you can use Context.getMainExecutor (). To send events through a shared thread pool, you can use asynctsask # thread_pool_executor. Consumer Consumer: callback invoked with a position or null This value cannot be null. Public string GetGnSwarModelName () Returns the model name (including the supplier and hardware / software version) of the GNSS or NULL hardware driver if this information is not available. No serial number or device-specific ID is returned by this API. PUBLISH INT GETGNSYEAREOFHARDWADWADWARE () Return the model of the GNSS hardware and the software build, or 0 if the model year is before 2016. Added in the API level 3 deprecated in API Level 24 Public GPSStatus GetGPSstatus (GPSStatus status) This method has been deprecated in API Level 24. GPSStatus APIs are deprecated, use GNSSSTATUS API instead. No more supported in the apps that turn to S and above. Recover information on the current status of the GPS engine. This should only be called from within the callback GPSStatus.Listener # OngPstatusschanged Callback to ensure that the data is copied atomically. The caller can pass in an existing GPSStatus object to overwrite or pass null to create a new GPSStatus object. Requires manifest.permission.access_fine_location Parameters GPSStatus status: Object containing GPS or NULL status details. This value may be null. Returns the GPSStatus status object containing the updated GPS status. Public position GetlastknownLationLocation (String Provider) Gets the last position known by the supplier given, or null if the last known position is not present. The returned position could be quite old in some circumstances, so the age of the position should always be controlled. This will never activate the sensors to calculate a new position and never return a position stored in the cache. See also GetCurrentLocation (Java.lang.String, Android.os.Cancellationsignal, Java.util.Concurrent.Executor, Java.util.Function.consumer) that you will always try to return a current position, but potentially will use further power over the course of Attempt with respect to this method. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Pronters String String: A supplier listed by GetallProviders () This value cannot be null. Returns the position the last known position for the specified or null supplier if available provider providerproperties GetProviderProperties (String Provider) returns the owner of the specified or NULL supplier if the properties are currently unknown. The owners of the supplier can change over time, even if this is And it should be rare. The most common transition is when the properties of the provider go from being unknown known, which is more common near the startup time. Parameter provider string: A supplier listed by GetallProviders () This value cannot be null. Return ProviderProperties Provider provider provider or null if the properties are currently unknown to throw illegalargumentException if the provider is null or there is no public list GetProvviders Getprovviders enabledOnly) Returns a list of the names of the available positions vendors. If abilitalmente is false, this is functionally as GetAllproviders (). Parameters Enadiblely BOOLEAN: If it's true, so only qualified suppliers are included refunds List List of names of suppliers This value can not be null. public List GetProviders (Criteria criteria, Boolean enadibililmente) Returns a list of localization vendors names available that meet the specified criteria. PARAMETERS OF CRITERIA CRITERIA: The criteria that suppliers must match this value can not be null. ENABESABILY BOOLEAN: If it's true, then only the enabled providers are included list of return List of names of illegally Jets IndovengumentException providers If the criteria are null Public HasProvider Boolean (String provider) Returns true if the specified location provider exists on this device, regardless of whether it is currently enabled or not. String parameter provider: a positional potential supplier This value can not be null. Boolean TRUE If the position providers exists, false otherwise illegally throws illegalArgumentiException if the provider is null public boolean isLoonnedSenabled () returns the enable / disable current status. To listen to the changes, see mode_changed_action. Boolean True If the position is enabled and false if the position is disabled. Public Void Requetflush (String Provider, PendingIntent PendingIntent, Int RemersCode) Requires that the specified provider has not wash scontrette positions listeners. The figure pendentente (registered with the provider) will be sent with key_flush_complete present in additional and requestcode keys as corresponding value. String of parameters provider: a supplier listed by getallproviders () This value can not be null. PendingIntent PendingIntent: pendentente a supplier registered under this value can not be null. RichiestaTode Int: an arbitrary integer number that will be transmitted as added value for Key_Flush_Complete Public Void Reacqueationupdates (String Provider, Long Mintimems, Float Mindistancem, LocationListener Listener) recorded for the location updates from the provider specified with the specified arguments and a callback looper of the calling thread. See Requestlocationupdates (java.lang.String, Android.Location.LocationSrequest, Java.util.concurrent.executor, Android.Location.LocationListender) for more details on how this method works. Before Jellybean, the MINTIME parameter was only a suggestion, and some providers ignorÃ² position implementations. For Jellybean and onwards, however, it is mandatory for Android compatible devices to observe both the minimum and mindistance parameters. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Pronters String String: A supplier listed by GetallProviders () This value cannot be null. Mintimems Long: minimum time interval between location updates in milliseconds Mindistancem float: minimum distance between the location updates to an extent Listener LocationListener: the listener to receive location updates This value can not be null. Added in API Level 9 Obrecated in API Level 31 Public Void RequestlocationUpdates (Mintimems Long, Float Mindistancem, policies, criteria, LocationListener Ascoltore, Looper Looper) This method has been deprecated in API Level 31. Use RequestLocationUpdates (java.lang.String, Long , Float.lang.String, Android.Location.LocationListener, Android.os.Looper) instead explicitly select a vendor. Register for position updates using a supplier selected through criteria And a callback on the specified looper. Note: Because Android Kitkat, the criteria requests will always make the use of fused_provider. See RequestlocationUpdates (Java.lang.String, Android.Location.LocationsRequest, Java.util.Concurrent.Executor, Android.Location.LocationListender) For more details on how this method works. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Mintimems Long: minimum minimum time Among the position updates in Milliseconds Mindistancem Float: minimum distance between position updates in criteria criteria: it contains parameters to choose the appropriate provider for position updates this value cannot be null. Listener LocationListener: the listener to receive position updates this value cannot be null. Looper Looper: This value may be null. Public Void RequestlocationUpdates (String Provider, Long Mintimems, Float Mindistancem, LocationListener Listener, Looper Looper) Register for position updates from the specified supplier with specified topics and callback on the specified looper. See RequestlocationUpdates (Java.lang.String, Android.Location.LocationsRequest, Java.util.Concurrent.Executor, Android.Location.LocationListender) For more details on how this method works. Before Jellybean, the Mintime parameter was just a suggestion, and some implementations of the location provider ignored it. For Jellybean and onwards, however, it is mandatory for Android compatible devices to observe both the minimum and mindstand parameters. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Pronters String String: A supplier listed by GetallProviders () This value cannot be null. Mintimems Long: Minimum time interval between position updates in Milliseconds Mindistancem Float: minimum distance between position updates to Made Listener LocationListener: the listener to receive position updates This value cannot be null. Looper Looper: The Looper Handling Listener Callbacks, or NULL To use the call thread looper This value may be null. Public Void RequestlocationUpdates (String Provider, Long Mintimems, Float Mindistancem, Executor Executor, LocationListener Listener) Register for position updates using the named supplier and call back to the specified executor. See RequestlocationUpdates (Java.lang.String, Android.Location.LocationsRequest, Java.util.Concurrent.Executor, Android.Location.LocationListender) For more details on how this method works. Before Jellybean, the Mintime parameter was just a suggestion, and some implementations of the location provider ignored it. For Jellybean and onwards, however, it is mandatory for Android compatible devices to observe both the minimum and mindstand parameters. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Pronters String String: A supplier listed by GetallProviders () This value cannot be null. Mintimems Long: Minimum time interval between position updates in Milliseconds Mindistancem Float: minimum distance between position updates in Executor Executor meters: The Executor Handling Listener Callbacks This value cannot be null. Callback and Listener events are sent via this performer, providing an easy way to check which thread is used. To send events through the main application thread, you can use Context.getMainExecutor (). To send events through a shared thread pool, you can use asynctsask # thread_pool_executor. Listener LocationListener: the listener to receive position updates this value cannot be null. Public Void RequestloonationUpdates (String Provider, LocationRequest LocationRequest, Executor Executor, Listener LocationListener) Register for position updates from the specified supplier, using a locationrequest and a callback on the specified executor. Only one request can be recorded for each pair of listener / provider So any subsequent requests with the same supplier and listener will overwrite all the associated topics. The same listener can be used on multiple suppliers with different requests for each supplier. You may need time to receive the first update of the position depending on the conditions in which the device is located. To take advantage of cached positions, the application may consider the use of getlastknownlocation (java.lang.String) or GetCurrentLocation (Java.lang.String, Android.Location.Locationsrequest, Android.Location.Locationsrequest, java.util.concurrent.executor, java.util.function.consumer) instead. See the LocationRequest documentation For an explanation of the various request parameters and how they can affect the received positions. If the application wants to passively observe location updates from all suppliers, then use the passive_provider. This provider does not turn on or modify active localization providers, so there is no need to be the most attentive to minimum time and minimum distance parameters. However, if your running heavy work applications on a position update (such as network activity) you need to set an explicit fast interval corresponding to the position request if another application allows a position supplier with Extremely fast updates. In the event that the provider has been selected it is deactivated, position updates will cease, and an availability supplier update will be sent. As soon as the supplier is reactivated, another supplier availability update will be sent and updates to the position will resume. When callback location are recalled, the system will keep a wakelock on behalf of your application for a certain period of time, but not indefinitely. If the application requires a long wakelock running within the callback position, you should acquire yourself. Spamming positions requests is a drain for system resources, and the system has preventive measures in place to ensure that this behavior will never be able to translate more positions than what could be reached with a single request position with an equivalent interval that It is left at home all the time. As part of this improvement, Android s intended applications and, above all, can receive caches or historical places through their listener. These positions will never be older than the interval of the position request. To cancel position updates, use RemoveUpdates (Android.Location.LocationListener). Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Pronters String String: A supplier listed by GetallProviders () This value cannot be null. LocationRequest LocationRequest: The position request containing pink parameters This value cannot be null. Executor performer: Handling callback listener This executor value can not be null. Callback and Listener events are sent via this performer, providing an easy way to check which thread is used. To send events through the main application thread, you can use Context.getMainExecutor (). To send events through a shared thread pool, you can use asynctsask # thread_pool_executor. Listener LocationListener: the listener to receive position updates this value cannot be null. Added to API level 9 Not recommended API level 31 RequestlocationUpdates empty (long Mintimems, Mindistancem float, criteria criteria, Pentingintent PendingIntent) This method is advised against API level 31. Use RequestlocationUpdates (Java.lang.String, Long, Float, Android . App.pendingIntent) Instead of explicitly selecting a provider. Register for position updates that use a selected provider through the indicated criteria, and callbacks provided through the appropriate pendingintent. Note: Because Android Kitkat, the criteria requests will always make the use of fused_provider. See RequestlocationUpdates (Java.lang.String, Long, Float, Android.App.pendingintent) For more details on how this method works. Requires manifest.permission.access_coarse_location or Long Mintimems parameters: Minimum time interval between Millisecond position updates Mindistancem Float: minimum distance between position updates in criteria criteria: contains the parameters to choose the appropriate supplier for the post update this value can not be null. PendingIntent PendingIntent: The wait intent to send position Update This value cannot be null. Added at API level 30 not recommended API level 31 RequestLocationUpdates empty (long Mintimems, Mindistancem float, criteria criteria, performer performer, performer, LISTENER) This method is not recommended at API level 31. Use RequestlocationUpdates (Java.lang.String, Long, Float, Java.util.Concurrent.Executor, Android.Location.LocationListener) Instead to select a provider explicitly. Register for position updates that use a selected provider through the criteria indicated, and a callback to the specified performer. Note: Because Android Kitkat, the criteria requests will always make the use of fused_provider. See RequestlocationUpdates (Java.lang.String, Android.Location.LocationsRequest, Java.util.Concurrent.Executor, Android.Location.LocationListender) For more details on how this method works. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Long mintimems parameters: Minimum time interval between Millisecond position updates Mindistancem Float: Minimum distance between position updates in criteria Policy: Contains the parameters to choose the appropriate supplier for The place updates this value can not be null. Executor performer: Handling callback listener This executor value can not be null. Callback and Listener events are sent via this performer, providing an easy way to check which thread is used. To send events through the main application thread, you can use Context.getMainExecutor (). To send events through a shared thread pool, you can use asynctsask # thread_pool_executor. Listener LocationListener: the listener to receive position updates this value cannot be null. Added in API level 9 deprecated in API level 30 Post RequestsingleUpdate Empty (String supplier, PendingIntententententent) This method was not recommended at API level 30. USA GetCurrentLocation (Java.lang.String, Android.S.Cancellationsignal, Java .util.Current. performer, java.util.function.consumer) instead as they do not involve a risk of exhaustion of extreme battery. Register for a single position update using a name provider and waiting for intent. See RequestlocationUpdates (Long, Float, Android.Location.criteria, Android.App.pendingIntent) For more details on how to use this method. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Parameters Pronters String String: A supplier listed by GetallProviders () This value cannot be null. PendingIntent PendingIntent: The wait intent to send position Update This value cannot be null. Added in API level 9 deprecated in API level 30 Public Void RequestsingleUpdate (criteria criteria, PendingIntent Pentingintent) This method was not recommended at API level 30. USA GetCurrentLocation (Java.lang.String, Android.S.Cancellationsignal, Java .util.Current. performer, java.util.function.consumer) instead as they do not involve a risk of exhaustion of extreme battery. Register for a single position update using a criteria and waiting for intent. Note: Because Android Kitkat, the criteria requests will always make the use of fused_provider. See RequestlocationUpdates (Long, Float, Android.Location.criteria, Android.App.pendingIntent) For more details on how to use this method. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Policy criteria parameters: Contains the parameters to choose the appropriate supplier for the post update this value cannot be null. PendingIntent PendingIntent: The wait intent to send position Update This value cannot be null. Added in API level 9 deprecated in API level 30 RequestsingleUpdate Public void (provider string, listener locationListener, Looper) This method has been advised against API 30. USA GetCurrentLocation (Java.lang.String, Android.S.Cancellationsignal, Java.util .Current.Executor, Java.util.Function.consumer) Instead as they do not involve A risk of exhaustion of extreme battery. Register for a single position update using the provider name and callback. See RequestlocationUpdates (Java.lang.String, Long, Float, Android.Location.LocationListener, Android.os.looper) For more details on how to use this method. method. Manifest.permission.access_coarerse_location or manifest.permission.access_fine_location Parameters String Pronters: A supplier listed by GetallProviders () This value cannot be null. Listener LocationListener: the listener to receive position updates this value cannot be null. Looper Looper: The Looper Handling Listener Callbacks, or NULL To use the call thread looper This value may be null. Added in API Level 9 Obrecated in API Level 30 Public VoID RequestsPraupdate (criteria criteria, LocationListener listener, Looper Looper) This method has been deprecated in the API level 30. Use GetCurrentLocation (Java.lang.String, Android.os.Cancellationsignal, Java .Util .concurrent.ececutor, java.util.function.consumer) Instead how it does not carry a risk of extreme drainage of the battery. Register for a single position update using a criterion and callback. Note: Because Android Kitkat, the criteria requests will always make the use of fused_provider. See RequestlocationUpdates (Long, Float, Android.Location.criteria, Android.App.pendingIntent) For more details on how to use this method. Requires manifest.permission.access_coarse_location or manifest.permission.access_fine_location Policy criteria parameters: Contains parameters to choose the appropriate provider for position updates This value cannot be null. Listener LocationListener: the listener to receive position updates this value cannot be null. Looper Looper: The Looper Handling Listener Callbacks, or NULL To use the call thread looper This value may be null. Public Boolean Sendextracommand (String Provider, String Command, Bundle Extras) sends additional commands to a localization provider. It can be used to support the specific extensions of the supplier to the localization manager API. Parameter provider string: A supplier listed by GetallProviders () This value cannot be null. Command String: command name to be sent to the supplier This value cannot be null. Extra package: Optional topics for command or null This value may be null. Always returns the Bolean True, the return value can be ignored public device Void Settesproviderlocation (String Provider, Position position) Sets a new position for the specified test provider. This location will be identical as a nice location to all clients via position # ismock (). The position object must have a minimum number of fields set to be considered valid, as per documentation on the position class. Parameter provider string: the name of the provider This value cannot be null. Location Location: The Dich Position This value cannot be null. Added in API Level 3 Depliced â €