Android toast duration milliseconds

I'm not robot
reCAPTCHA

Continue

I'm not robot
reCAPTCHA

Continue

How many times have I seen this code: Of course I wrote it myself. Toast.makeText (context, My text, 5000).) Note!! Here you can see the document: This is the source: LENGTH_LONG and LENGTH_SHORT THIS FLAGS! Then the duration is not expressed in milliseconds! If you fix this code you can see: And this is the central point, NotificationManagerService: This is the main series: long delay and immediately? 0 : (r.duration - Toast.LENGTH_LONG? LONG_DELAY : SHORT_DELAY); Where: private static final LONG_DELAY No. 3500; 3.5 seconds private static final int SHORT_DELAY 2000; 2 seconds Then, the code written at the beginning of WRONG Toast.makeText (context, My text, 5000). Toast.makeText expects the duration of Toast.LENGTH_SHORT or Toast.LENGTH_LONG, the user value of the duration is not supported. However, I have a slight doubt: in the source we can find: This time can be user defined. Honestly, I can't find a way to do that. Any suggestion is welcome. In Android SDK, android.widget.Toast is a small message that pops up at the bottom of the screen to display information. Toast will disappear on its own after a certain duration. Here's an example of what toast looks like and how to display it: Context context - getApplicationContext Toast.makeText (context, Hello, world, I toast., Toast.LENGTH_SHORT).. The length for which toast is displayed on the screen is unfortunately determined by the flag: you can show it within a short duration, which is 2 seconds or a duration of 3.5 seconds. But what if you have a long error message that should be shown longer than that? Or if you need to show a countdown that is updated every second? There is no way to directly change the duration during which toast is displayed using the show method without re-iming the entire Toast class in the app, but there is a workaround. You can use android.os.CountDownTimer to count the time to display the toast. The CountDownTimer class plans a countdown in milliseconds with notifications at certain intervals before the countdown ends. In this example, the countdown is used to display a toast message over a period of time at the touch of a button: private Toast mToastToShow; Public emptiness showToast (View view) / Set a toast and duration in toastDurationInMilliSeconds - 10,000; mToastToShow - Toast.makeText (this, Hello, world, I toast., Toast.LENGTH_LONG); Set a countdown to countDownTimer toastCountDown toast. toastCountDown 1000 / Tick duration/) - public void onTick (long - mToastToShow.show (); - public void on Finish - mToastToShow.cancel (); Here's how it works: Has a notification time shorter than the duration for which the toast is displayed according to the flag, so the toast can be shown again if the countdown is not over. If the toast is displayed again while it is still on the screen, it will stay there for the duration without blinking. When the countdown is over, the toast is cancelled to hide it, even if its display duration is not over. This works even if the toast should be shown for a duration shorter than the default duration: the first pop-up toast will simply be cancelled when the countdown is over. Toasts are used in android to display notifications. When the .show () method is called they disappear and stay for a while and disappear depending on how long in milliseconds we set for toast using the setDuration method. We can only specify two values at the time of Toast those you know that even if we can set any integer value for setDuration (), the toast message will only appear for 1.5secs (i.e. Shot Period) or 3secs (long period). Both of these variables are flags with 0 and 1 respectively. Reason if you see: NotificationManagerService.java code, which is called to display Toast MESSAGE_TIMEOUT, private invalid graphTimeoutLocked (ToastRecord d, boolean immediately) Toast.LENGTH_LONG - Message m . LONG_DELAY : SHORT_DELAY); mHandler.removeCallbacksAndMessages (r); mHandler.sendMessageDelayed Where, private static final int LONG_DELAY 3500; 3.5 seconds private static final int SHORT_DELAY 2000; 2 seconds From here there is no way you can change the value of LENGTH_LONG or LENGTH_SHORT. The only way to display Toast for a longer period is through some Java code settings! More messages related to Android, more messages: Toast is a view containing a quick little message to the user. A toast class helps you create and show them. When the view is displayed to the user, it appears as a floating view above the app. He'll never get the focus. The user will probably be in the middle of typing something different. The idea is to be as unobtrusive as possible, while at the same time showing the user the information you want them to see. Two examples are volume control and a brief message that your settings are saved. The easiest way to use this class is to name one of the static methods that builds everything you need and returns a new Toast object. Note that Snackbars is preferable to short messages while the app is in the foreground. For information about Toast notifications, see the Toast Notifications developer's guide. Toast.Callback Callback is an object that will be called when toast is displayed or hidden. LENGTH_LONG show a view or text notification over a long period of time. LENGTH_SHORT show a view or text notification within a short period of time. Time. Context) Build an empty Toast object. invalid addCallback (Toast.Callback callback) adds a callback that will be notified when the toast is displayed or hidden. cancel () Close the view if it's showing, or don't show it if it's not showing yet. int getDuration () Return duration. Get a place where the notification should appear on the screen. getHorizontalMargin float () Return horizontal margin. the float getVerticalMargin () Return of vertical margin. View getView () This method has been unified in API level 30. Custom toast views are decreced. Apps can create a standard text toast using the makeText (android.content.Context, java.lang.CharSequence, int) method, or use Snackbar when in the foreground. Starting with Android Build.VERSION_CODES-R, apps that focus on the API level Build.VERSION_CODES or above that are in the background will not display custom toast views. int getXOffset () Return X bias in pixels to apply to gravity location. int getYOffset () Bring back the Y offset in the pixels to apply to the gravity location. Static Toast makeText (Context Context, int resld, Int Duration) Make a standard toast that simply contains text from the resource. Static Toast makeText (Context Context, CharSequence text, int duration) Make a standard toast that simply contains text. invalid removeCallback (Toast.Callback callback) removes a callback previously added with addCallback (android.widget.Toast.Callback). Invalid Set Duration (int duration) Set how long to show the view for. Invalid setGravity (nongravity, int xOffset, int yOffset) Set the place where the notification should be displayed on the screen. Void setMargin (float horizontalMargin, float verticalMargin) Set the edges of the species. invalid setText (int resld) Text update in toast, which was previously created using one of the makeText methods. invalid setText (CharSequence s) Text update in toast, which was previously created using one of the makeText methods. This method was deprecated in API level 30. Custom toast views are decreced. Apps can create a standard text toast using the makeText (android.content.Context, java.lang.CharSequence, int) method, or use Snackbar when in the foreground. Starting with Android Build.VERSION_CODES-R, apps that focus on the API level Build.VERSION_CODES or above that are in the background will not display custom toast views. invalid show () Show presentation for Duration. From the java.lang.Object Object class, the clone creates and returns a copy of this object. boolean (Object obj) indicates whether any other object is equal to this. invalid completion () Is called by the garbage collector at the facility when the garbage collection determines that there are no more references to the object. the final class of the getClass returns the time class of the performance of this object. int hashCode () Returns the hash code value to the object. Object. invalid notify () Wakes up one thread that waits on the monitor of this object. The final void notifies All () All the threads that are waiting on the monitor of this object. ToString returns the view of the object line. The final expectation of emptiness (long time out, int nanos)) triggers anticipation of the current thread until another thread triggers the notification method () or the notifyAll method for that object, or some other thread interrupts the current thread, or a certain amount of real time has passed. The final expectation of emptiness (long time) triggers the wait for the current thread until another notification method () or notifyAll method has been triggered for that object, or a certain amount of time has passed. the final expectation of emptiness () causes the current thread to wait until another thread triggers the notification method () or the notifyAll method for that object. Public static final LENGTH_LONG show a presentation or text notification over a long period of time. This time can be defined by the user. See also: Permanent value: 1 (0x00000001) public static finale int LENGTH_SHORT Show submission or text notification in a short period of time. This time can be defined by the user. It's the default. See also: Permanent Value: 0 (0x00000000) Public Void addCallback (Toast.Callback callback) Adds a callback to be notified when toast is displayed or hidden. Please note that if the toast is blocked for some reason, you will not be re-elected. Toast.Callback Callback Call: This value can't be zero. See also: Public Cancellation () Close view if it shows, or don't show it if it doesn't show it yet. Normally you don't have to call it. Usually the species disappears on its own after the corresponding duration. Public int getGravity () Get a place in which the notification should appear on the screen. Warning: Starting with Android Build.VERSION_CODES'R, for apps that are Build.VERSION_CODES'R or higher, this method should not be called text toasts because its return value may not reflect the actual value because text notifications are no longer displayed by the app. See also: Public Swimming getHorizontalMargin () Return of horizontal margin. Warning: Starting with Android Build.VERSION_CODES'R, for apps that are Build.VERSION_CODES'R or higher, this method should not be called text toasts because its return value may not reflect the actual value because text notifications are no longer displayed by the app. getVerticalMargin vertical margin. Warning: Starting with Android Build.VERSION_CODES'R, for apps that are Build.VERSION_CODES'R or higher, this method should not be called text toasts because its return value may not reflect the actual value because text notifications are no longer displayed by the app. Added to API level 1 Deprecated in API level 30 public view getView () This method has been deprecated in API level 30. Custom toast views are decreced. Apps can create text toast using the makeText method (android.content.Context, java.lang.CharSequence, int) or use snackbar when in the foreground. Starting with Android Build.VERSION_CODES-R, apps that focus on the API level Build.VERSION_CODES or above that are in the background will not display custom toast views. Bring back the view. Toasts built with Toast (android.content.Context), which are not called setView (android.view.View) with a non-zero view, will not be invalidated here. Starting with Android Build.VERSION_CODES.R, in apps focused on the API level Build.VERSION_CODES'R or higher, toast built with makeText (android.content.Context, java.lang.CharSequence, int) or its variants will also return invalid here, unless they named setView (android.view.View) with a non-zero view. If you want to be notified when toast is displayed or hidden, use addCallback (android.widget.Toast.Callback). Public int getXOffset () Return X bias in pixels to apply to the location of gravity. Warning: Starting with Android Build.VERSION_CODES'R, for apps that are Build.VERSION_CODES'R or higher, this method should not be called text toasts because its return value may not reflect the actual value because text notifications are no longer displayed by the app. Public int getYOffset () Return of Y-offset in pixels for application to gravity location. Warning: Starting with Android Build.VERSION_CODES'R, for apps that are Build.VERSION_CODES'R or higher, this method should not be called text toasts because its return value may not reflect the actual value because text notifications are no longer displayed by the app. Public set of voids (int gravity, int xOffset, int yOffset) Set the place in which the notification should be displayed on the screen. Warning: Starting with Android Build.VERSION_CODES-R, for apps focused on the API level Build.VERSION_CODES'R or higher, this method is a non-op when calling for text toast. Gravity options int xOffset int yOffset int See also: public set of voidMargin (float horizontalMargin, float verticalMargin) Set the field of view. Warning: Starting with Android Build.VERSION_CODES-R, for apps focused on the API level Build.VERSION_CODES'R or higher, this method is a non-op when calling for text toast. Margin horizontal float options: Horizontal limit, as a percentage of container width, between the edges of the container and the notification of the vertical floatMargin: Vertical limit, as a percentage of the height of the container, between the edges of the container and the notice of public void setText resld) Text update in Toast, which was previously created using one of the makeText () methods. Resld int options: New text for toast. setText (CharSequence s) text update on toast, which was previously created using one of the makeText () methods. Options with CharSequence: New text for toast. public invalid show () Show presentation presentation duration. Duration.