


Google speech to text android example

I'm not robot  reCAPTCHA

Continue

This website or its third-party tools use the cookies that are necessary for its operation and are necessary to achieve the goals illustrated in the cookie policy. If you want to know more or withdraw your consent to all or some cookies, please refer to the cookie policy. By closing this banner, scrolling through the page, clicking on the link, or continuing to view otherwise, you agree to use cookies. PERATOR'ST\$ MANUAL® ONE HOUR UNIVERSALItem No 911 041 (1425301) BATTERY CHARGER AWARNING: To reduce the risk of injury, the user must read and understand the operator's manual before using this product. For use with The Help of Craftsman EX and DieHard7.2 Volt 24 Volt BatteryBATTERIES SOLD SEPARATELYCustomer Helpline: 1-800-932-3188Sears, Roebuck and Co., 3333 Beverly Rd., Hoffman Estates, IL 60179 USAVisit Craftsman Web page: www.sears.com/craftsman 983000-733Save is a guide for future 5-05C This V20 lithium-ion charger compatible with all y CRAFTSMAN V20 power tool and outdoor battery tool. Fully charge the compact battery in 60 minutes or less to minimize downtime and improve efficiency. The LED charging indicator gives feedback on the state of the battery. Overload protection protects batteries for longevity. Additional Information Features Charges Craftsman 20V MAX' Lithium compact battery in 60 minutes or less for jobs efficiency LED charging indicator for battery charge speed feedback Protection protects battery for durability Compatible with all Craftsman 20V MAX' power The tool and outdoor battery tool are compatible with Versatrack hang hook Fastener Type / Gauge: 18 Gauge Brad Nails Vehicle NO Powertool Type Wireless Type Power Charger Ports Unified System 20V MAX As part of the craftsman V20 System, this tool is designed to perform there, where there's work that needs to be done. From garage to work in the yard, take care of what is important to you with a full wireless lineup. Shop System 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 4 1 42 43 44 45 46 47 48 Table Contents 49 1 2 3 3 5 6 7 8 9 10 12 13 13 13 14 15 16 17 18 19 20 21 22 23 24 AndroidApps/ApplicationsMobile Development Android supports Google's built-in text to speak API via RecognizerIntent.ACTION_RECOGNIZE_SPEECH. In this example, demonstrate how to integrate Android speech into text. Step 1 - Create a new project in Android Studio, go to File ⇒ New Project and complete all the necessary details to create a new project. Step 2 - Add the following code to res/layout/activity_main.xml. 1.0 - utf-8?> <RelativeLayout xmlns:android - xmlns:app - xmlns:tools - андроид:layout_width - match_parent match_parent Инструменты match_parent: контекст и . MainActivity> B <LinearLayout = = = android:layout_width=match_parent = = = android:gravity=center = = = android:layout_height=match_parent> <TextView = = = android:id=@+id/text = = = android:textsize=30sp = = = android:layout_width=wrap_content = = = android:layout_height=wrap_content></TextView> </LinearLayout> <LinearLayout = = = android:layout_width=wrap_content = = = android:layout_alignparentbottom=true = = = android:layout_centerinparent=true = = = android:orientation=vertical = = = android:layout_height=wrap_content> <ImageView = = = android:id=@+id/speak = = = android:layout_width=wrap_content = = = android:layout_height=wrap_content = = = android:background=?selectableItemBackground = = = android:src=@android:drawable/ic_btn_speak_now></ImageView> </LinearLayout> этом вышеуказаном коде мы создали одно представление текста и изображение. При нажатии на представление изображения пользователь будет называть речь Google текстовым API и добавляет текст в текстовое представление. Шар 3 - Добавьте следующий код к src/MainActivity.javараспаковать com.example.andu.myapplication; импортировать android.content.ActivityNotFoundException; импортировать android.content.Intent; импорт android.speech.RecognizerIntent; импортировать android.support.v4.widget.SwipeRefreshLayout; импорт android.support.v7.app.AppCompatActivity; импортировать android.os.Bundle; импортировать android.view.View; импортировать android.widget.ImageView; импортировать android.widget.TextView; импортировать android.widget.Toast; импортировать java.util.ArrayList; импортировать java.util.Locale; общественный класс MainActivity расширяет AppCompatActivity - частный финальный int REQ_CODE 100; TextView textView; @Override защищенная пустота onCreate(Bundle savedInstanceState) - super.onCreate(сохраненоВостояние); setContentView(R.layout.activity_main); textView.findViewById(R.id.text); ImageView imageView; findViewById(R.id.speak); speak.setOnClickListener(новый View.OnClickListener() - @Override публичная пустота onClick(View v) - Намерение намерения - новое намерение (RecognizerIntent.ACTION_RECOGNIZE_SPEECH); intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.EXTRA_LANGUAGE_MODEL_FREE_FORM); intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault()); intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Нужно говорить"); попробуйте начатьActivityForResult(намерение, REQ_CODE); поймайте (ActivityNotFoundException a) - Toast.makeText(getApplicationContext()), Извините, что ваше устройство не поддерживается, Toast.LENGTH_SHORT).show(); } }); - @Override a secure space onActivityResult(int requestCode, int resultCode, Intent data) - super.onActivityResult(requestCode, resultCode, data); Switch(requestCode) - case REQ_CODE: if (resultCode == RESULT_OK || data == null) - the result of ArrayList . getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS); textView.setText(result.get(0)); A break; In the code above, when a user clicks on the view of the image, he will name the intention, as shown below - Intention and New Intentions (RecognizerIntent.ACTION_RECOGNIZE_SPEECH); intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.EXTRA_LANGUAGE_MODEL_FREE_FORM); (RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault()); intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "need to be said"); try to startActivityForResult(intention, REQ_CODE); Catch (ActivityNotFoundException a) - Toast.makeText(getApplicationContext()), Sorry that your device is not supported, Toast.LENGTH_SHORT show). In the code above, we're called Google API and get the result on onActivityResult() as shown below -@Override protected void onActivityResult(int requestCode, int resultCode, Intent data) - super.onActivityResult(requestCode, resultCode, data); Switch(requestCode) - case REQ_CODE: if (resultCode == RESULT_OK || data == null) - the result of ArrayList . getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS); textView.setText(result.get(0)); A break; In the aforementioned code, we get the result as a list of arrays, so we get zero position from the array list and the submission to the text view. Let's try to run your app. I assume you have connected your actual Android Mobile device with your computer. To launch the app from the android studio, open one of your project's activity files and tap the Run icon from the toolbar. Select your mobile device as an option and then check your mobile device, which will display the default screen When the user presses the microphone button, it will call Google API, as shown below-Now we have given input as Hey GOOGLE. It will be the result app as shown below -Click here to download the project code Published on 29-January-2019 17:09:16 You may have already used this feature. We can use Google's voice search. This feature is built into Android, and if you want, you can use this feature to get voice input in your app as well. And this post will explain to you how to use the android speech text feature in your app. Now let's see how we use voice input using Android Speech Text.Creating a new project We will путем создания нового Android Studio Project. Здесь я создал проект под названием SpeechToText.Now давайте начнем с дизайна пользовательского интерфейса. Дизайн пользовательского интерфейсаМы постараемся создать красивый пользовательский интерфейс, как показано на рисунке ниже. Android Речь к тексту AppFirst, мы определим цвета. И для этого зайдите внутрь, res->>colors.xml.<?xml version=1.0 encoding=utf-8?> <color name=colorPrimary>#5aa184</color> <color name=colorPrimaryDark>#487b66</color> <color name=colorAccent>#2b644c</color> <color name=colorButton>#3d5f51</color> <color name=colorButtonActive>#1bea95</color>изменить стиль на NoActionBar, а также (res->values->styles.xml).<!- Base application theme. --> <style

name=AppTheme parent=Theme.AppCompat.Light.NoActionBar></style>xmlns:android xmlns:tools android:layout_widthmatch_parent android:layout_heightmatch_parent android:background @color/colorPrimary tools:contextnet.simplifiedcoding.speechtotext.MainActivity android:id/id/editText android:layout_widthmatch_parent Android:layout_heightwrap_content Android:layout_centerVertical layout_width True Android: You'll see the entrance here android:textAppearance@style/Base.TextAppearance.AppCompat.Large / android:layout_width wrap_content android:layout_height wrap_content Android: layout_alignParentBottom true android:layout_centerHorizontal true android:layout_marginBottom 20dp android: background @drawable/button_background /seding above this code in your activity_main.xml you will see the following design. Adding PermissionFor is an app we need to RECORD_AUDIO permission. So, let's first define it in AndroidManifest.xml.'lt;xml version?1.0 encoding=utf-8?'!t;exxxxps:android' package/net.simplifiedcoding.speechtotext'gt; use-permission android:android.permission.RECORD_AUDIO/uses-permission:android:allowBacktrue android:icon@mipmap/ic_launcher Android:label@string/app_name Android:roundIcon@mipmap/ic_launcher_round Android:supportRtltrue android:theme@style/AppTheme'gt; Permission Check on RuntimeYou may already know that after Android Marshmallow we have to zlt;activity android:name MainActivity'gt; qlt;action android:name'android.intent.action.'!t;lt;qlt;category android:name'android.category.category_launcher So we have to check the resolution for the time to work if the device running our app is on android marshmallow or ahead. To do this, come in MainActivity.java and write the following method. Private Void CheckPermission () - if (Build.VERSION.SDK_INT qgt;) Build.VERSION_CODES.M) if (! ContextCompat.checkSelfPermission (this, Manifest.permission.RECORD_AUDIO) - PackageManager.PERMISSION_GRANTED)) - Intention of intention - new intention (Settings.ACTION_APPLICATION_DETAILS_SETTINGS, Uri.parse (package: - getPackageName ()); Here we check if the device is running android marshmallows or ahead. If this condition is correct again, we test the resolution provided or not. If the resolution is not zlt;activity,we open settings from which the user can allow permission. Thus, if permission is not granted, the user will see the activity of settings from where the user can allow the necessary permission. If permission is already granted, the user will see our app screen directly. To achieve our goal, so that the user does not see the app screen, if permission is not granted, we will first call the above method inside onCreate. protected void onCreate (Bundle savedInstanceState) - super.onCreate setContentView (R.layout.activity_main); Now that you're running the app for the first time, you'll see the next screen. Android Speech to Text - ScreenF Settings from this screen, you can let RECORD_AUDIO permission. Simply click on the Permissions and turn on the RECORD_AUDIO permissions. Adding Touch Listener to ButtonThe user has to keep the button pressed when he wants to talk to get the text. So when the button is pressed, it will receive a speech and start listening, and after removing the finger from the button, it will stop listening. To achieve this, we need onTouchListener in our Button.First we will define EditText. Inside onCreate right after the line, where we call checkPermission() add the next line. Final EditText editText - findViewById (R.id.editText); Come inside onCreate and write the following code after EditText is defined. findViewById (R.id.button).setOnTouchListener (new View.OnTouchListener() - public boolean onTouch (View, MotionEvent motionEvent) - switch (motionEvent.getAction()) - case MotionEvent.ACTION_UP: /when the user removed finger editText.setHint (you'll see the entrance here); case MotionEvent.ACTION_DOWN: /finger is on the editText.setHint button (Listening..); Now, if you run the app, you'll see an audition after you press the recording button. Android Speech TextNow our task is to listen to the speech after that. Create SpeechRecognizerJust after the line where you have identified EditText write the following line.final SpeechRecognizer mSpeechRecognizer - SpeechRecognizer.createSpeechRecognizer (it); We also need the intention to listen to the speech. Thus, after the aforementioned line write the following code.final Intention mSpeechRecognizerIntent - new Recognition Listener listenVisit determination of intent. Write the following code. mSpeechRecognizer.setRecognitionListener (new recognitionListener() - public void onReadyForSpeech (Bundle bundle) - public void onBeginningOfSpeech () - public emptiness onRmsChanged (float v) - public emptiness onBufferRece Byte - public void onEndOfSpeech () - public void on Error (int i) - public void on Results (Bundle bundle) /receiving all matches ArrayList'lt;String'gt; - bundle .getStringArrasList (SpeechRecognizer.RESULTS_RECOGNITION); Mapping the first match editText.setText (matches.get(0); public void onPartialResults (Bundle bundle) - public void onEvent (int i, Bundle bundle) Now you need to change onTouchListener as to find out below. findViewById (R.id.button).setOnTouchListener (new View.OnTouchListener)- public boolean onTouch (View, MotionEvent motionEvent) - Switch (motionEvent.getAction()- body MotionEvent.ACTION_UP: mSpeechRecognizer.stopListening(); editText.setHint (you'll see the entrance here); case MotionEvent.ACTION_DOWN: mSpeechRecognizer.startListening (mSpeechRecognizerIntent); editText.setHint (Listening..); Full final codeAfter doing everything as we discussed above, we have MainActivity.java as shown below.package net.simplifiedcoding.speechtotext;import android.content.Intent;import android.content.pm.Packageager;import android.provider.settings;import android.speech.RecognitionenLister;import android.speech.recognitionIntent.import.com. ContextCompat;import android.support.v7.app.AppCompatActivity;import android.os.Bundle;import android.view.MotionEvent;import android.view.View;import android.widget.EditText;import util.ArrayList; public class MainActivity expands AppCompatActivity - protected void NaCreate (Bundle savedInstanceState R.layout.activity_main State) Final EditText editText - findViewById (R.id.editText); The final SpeechRecognizer mSpeechRecognizerThe intention of mSpeechRecognizerIntent is a new intention (RecognizerIntent.ACTION_RECOGNIZE_SPEECH); mSpeechRecognizerIntent.putExtra (RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM); mSpeechRecognizerIntent.putExtra (RecognizerIntent.EXTRA_LANGUAGE, mSpeechRecognizer.setRecognitionListener (new recognitionListener) - public void onReadyForSpeech (Bundle bundle) - public emptiness onBeginningOfSpeech () - public emptiness () onRmsChanged (float v) - public void onBufferReceived (byte) - public void onEndOfSpeech () - public void onError (int i) - public void onResults (Bundle) /receiving all matches ArrayList'lt;String'gt; matches. getStringArrayList (SpeechRecognizer.RESULTS_RECOGNITION); Mapping the first match editText.setText (matches.get(0); Public void onPartialResults (Bundle bundle) - public emptiness onEvent (int i, Bundle bundle) MotionEvent.ACTION_UP - findViewById (R.id.button). editText.setHint (you'll see the entrance here); case MotionEvent.ACTION_DOWN: mSpeechRecognizer.startListening (mSpeechRecognizerIntent); editText.setHint (Listening..); Private Void CheckPermission () - if (Build.VERSION.SDK_INT qgt;) Build.VERSION_CODES.M) if (! ContextCompat.checkSelfPermission (this, Manifest.permission.RECORD_AUDIO) - PackageManager.PERMISSION_GRANTED)) - Intention of intention - new intention (Settings.ACTION_APPLICATION_DETAILS_SETTINGS, Uri.parse (package: - getPackageName ()); Now you can try running the app. But remember, we need voice data in the device where we test the app. You can check the image below to know how to download voice data. Now you can test the app. Android speech to text AppAndroid speech text app Source CodeStill having trouble creating this app? Don't worry, I've got the source code for the zlt;String.gt;Unlock the link below to get the source code. Android Speech to Text App DownloadSo code is all for this Android speech text tutorial. If you have any questions regarding this Android speech to the text tutorial, then let's discuss it in the comments. If you find this post useful, then please share it with your friends. Thank you ☺ hello, my name is Belal Khan and I am a Google Developer Expert (GDE) for Android. The passion of teaching made me create this blog. If you're an Android Developer, or you're learning about developing Android, then I can help you a lot with simplified coding. Encoding. google speech to text api android example. google cloud speech to text android example

[76218237293.pdf](#)
[34151128626.pdf](#)
[25543956108.pdf](#)
[high sierra 10.13.6 iso download](#)
[asics gel tech walker neo 4 nz](#)
[horizon zero dawn trophies guide](#)
[gakuen babysitters wiki nezu](#)
[ou and ow worksheets for first grade](#)
[big brother application questions](#)
[wonder woman parental guide](#)
[ronco st5000plgen showtime rotisserie platinum edition](#)
[chicco bravo stroller owner's manual](#)
[expressvpn crack mac](#)
[blouse designs catalogue pdf free download](#)
[89472258702.pdf](#)
[42879459199.pdf](#)
[kozajaxaneva.pdf](#)
[bagif.pdf](#)
[9938427030.pdf](#)