

1 Voice Recognition and Android Speech to Text, TTS, Tutorial

Speak with Android phone? No Problem, Android provides such function.

1.1 Source

On github and apk could only run on android device but not on **AVM**.

1.2 Practice

1. Voice recognition,: speak to your phone,internet-permission require;
2. TTS: input string and speak in normal mode, man's tone, or woman's tone.

1.3 Environment Info

0. Kotlin
1. buildToolsVersion: '~~26.0.2~~'
2. gradle: 3.2.0-alpha15
3. Android version: minSdkVersion 19, targetSdkVersion 25

1.4 Install (Offline) Language Pack

Swipe android cell device up-side-down, then

⚙️ → [Language & input] → [Google voice typing] → [Offline speech recognition] → [ALL]

and install the language.

1.5 Steps of project

Create a new project

Application Name: ttsdemo

Company Domain: com.cgu.tts

0. Using TTS requires **Minimum SDKAPI 21**(Android 5.0, Lollipop) at least and here, we use kotlin to develop the app;
1. active the internet permission (**AndroidManifest.xml**)
2. UI design: (**activity_main.xml**)

```
<textview>Input column</textview> <button>
Record</button>
<button>NormalSound</button> <button>ManSo
und</button><button>FemaleSound</button>
```

3. Java part: Voice recognition, then TextToSpeak

1.6 UI, activity_main.xml

Create a `LinearLayout` whichs contains one `EditText` and `Button`:

```
...
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">
    <EditText
        android:id="@+id/edittext1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        "
        android:ems="10" >
        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        "
        android:text="Speak" />
</LinearLayout>
...
```

2 TTS, Text To Speak

2.1 Outline of the coding

```
package tts.cgu.com.ttsdemo

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.speech.tts.TextToSpeech
import android.util.Log
import android.widget.Button
import android.widget.EditText
import kotlinx.android.synthetic.main.activity_
main.*
import java.util.*

// Extend activity with TextToSpeech.OnInitList
ener.
class MainActivity : AppCompatActivity(),TextTo
Speech.OnInitListener {
    private var tts: TextToSpeech? = null
    private var buttonSpeak: Button? = null
    private var editText: EditText? = null

    override fun onCreate(savedInstanceState: B
undle?) {
        //
    }

    override fun onInit(status: Int) {
        // Initialize TextToSpeech class variab
le.
        ...
    }

    public override fun onDestroy() {
        // shutdown the TTS Engine while Activ
ity is destroyed
        ...
    }

    private fun speakOut() {
        // start tts service
        ...
    }
}
```

The steps to TTS app,

1. Initialize TextToSpeech class variable.

```
tts = TextToSpeech(context: this, listener:
this)
```

2. A button onClickListener waits for services, speakOut() :

```
buttonSpeak!!.setOnClickListener { speakOut
() }
...
private fun speakOut() {
    val text = editText!!.text.toString()
    tts!!.speak(text, TextToSpeech.QUEUE_FLUSH,
null, "")
}
```

3. When the application is started, TextToSpeech Engine may take some duration of time for initialization. To avoid speaking out, we may initially disable the button to speak. When the TextToSpeech Engine is initialized, onInit() function is called, which should be overridden and we may enable the button here.
4. finally, stop and shutdown the TTS Engine while Activity is destroyed:

```
public override fun onDestroy() {
    // Shutdown TTS
    if (tts != null) {
        tts!!.stop()
        tts!!.shutdown()
    }
    super.onDestroy()
}
```

2.2 Complete Code

```
package tts.cgu.com.ttsdemo

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.speech.tts.TextToSpeech
import android.util.Log
import android.widget.Button
import android.widget.EditText
import kotlinx.android.synthetic.main.activity_
main.*
import java.util.*

// Extend activity with TextToSpeech.OnInitList
```

```

ener.
class MainActivity : AppCompatActivity(), TextTo
Speech.OnInitListener {

    private var tts: TextToSpeech? = null
    private var buttonSpeak: Button? = null
    private var editText: EditText? = null

    override fun onCreate(savedInstanceState: B
undle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        buttonSpeak = this.button1;
        editText = this.edittext1;

        buttonSpeak!!.isEnabled = false;
        tts = TextToSpeech(this, this)

        buttonSpeak!!.setOnClickListener { spea
kOut() }
    }

    override fun onInit(status: Int) {

        if (status == TextToSpeech.SUCCESS) {
            // set US English as language for t
ts
            val result = tts!!.setLanguage(Loca
le.US)

            if (result == TextToSpeech.LANG_MIS
SING_DATA || result == TextToSpeech.LANG_NOT_SU
PPORTED) {
                Log.e("TTS", "The Language speci
fied is not supported!")
            } else {
                buttonSpeak!!.isEnabled = true
            }

        } else {
            Log.e("TTS", "Initilization Failed!
")
        }
    }

    private fun speakOut() {

```

```

        val text = editText!!.text.toString()
        tts!!.speak(text, TextToSpeech.QUEUE_FLUSH, null, "")
    }

    public override fun onDestroy() {
        // Shutdown TTS
        if (tts != null) {
            tts!!.stop()
            tts!!.shutdown()
        }
        super.onDestroy()
    }
}

```

In []: