

TRABALHANDO COM API DE VOZ

Prof. Danilo Ruy Gomes



Introdução

O que veremos:

- Api Text to Speak;
- Api Speak to Text;
- Setar idioma e instalar pacote de vozes.

O que precisaremos

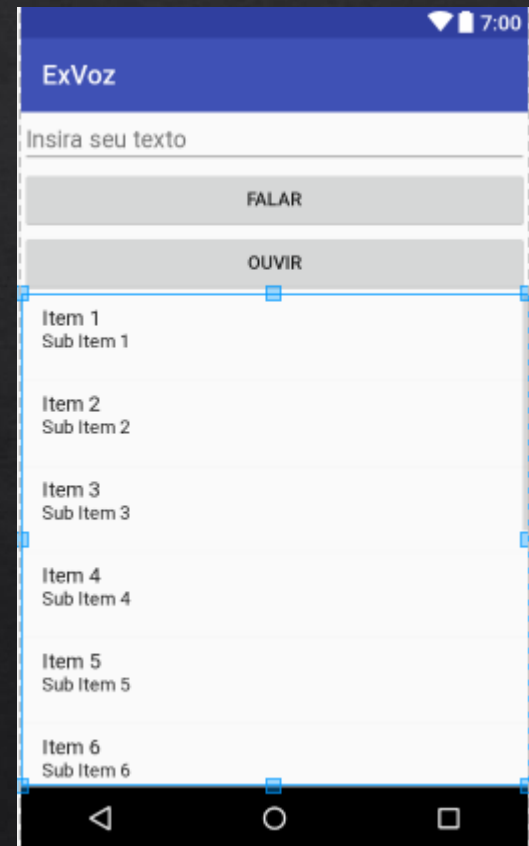
- Android Studio;
- Acesso a internet;
- Emulador ou Smartphone conectado no computador.

Api de voz

- Basicamente o Android utiliza duas bibliotecas para trabalhar com voz chamada TTS (Text to Speak) e STT (Text to Speak);
- A STT depende da internet para se ter uma maior precisão, porém as bibliotecas de vozes vem se tornando cada vez mais eficiente para o reconhecimento.

Api de voz

Crie um projeto em branco e insira respectivamente um “*edittext*”, dois “*buttons*” e um “*listview*”, como na imagem:



Api de Voz

Depois crie uma “*actionBar*” com dois itens, sendo respectivamente para setar o idioma português e setar o idioma inglês:

```
<?xml version="1.0" encoding="utf-8" ?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/action_portugues" android:title="Português"
          app:showAsAction="never"/>
    <item android:id="@+id/action_ingles" android:title="Inglês"
          android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

Api de Voz

- Na classe “*Main*”, faça a implementação do método “*OnInit*” de “*TextToSpeech*” que será utilizado para setará o idioma de voz inicial:

```
public class MainActivity extends AppCompatActivity implements TextToSpeech.OnInitListener{  
    private EditText tMsg;
```

- Depois clique na “*lâmpada vermelha*” para implementar o método automaticamente.

Api de Voz

- Na classe “Main” instancie os objetos como na imagem abaixo:

```
{  
    private EditText tMsg;  
    private TextToSpeech textFala;  
    private Button btFalar, btOuvirVoce;  
    private ListView lstV;  
    private Locale locale;  
}
```


Api de Voz

- Já no método “*create*” instancie o objeto “*TextToSpeech*” e declare a chamada do “*listener*” dos botões :

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textFala = new TextToSpeech(MainActivity.this, MainActivity.this);
    //no objeto acima que sera tratado a voz
    tMsg = (EditText) findViewById(R.id.edtTexto);
    botoes();
}
```

Api de Voz

- Insira o método para “*inflar o actionBar*”:

```
}  
@Override  
public boolean onCreateOptionsMenu(Menu menu)  
{  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.actionbar, menu);  
    return super.onCreateOptionsMenu(menu);  
}
```

Api de Voz

- Insira o método para escolha do item “*actionbar*” e sete o comando para colocar o idioma português brasileiro:

```
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    int i =item.getItemId();
    {
        if (i == R.id.action_portugues)
        {
            locale = new Locale("pt", "BR");
            textFala.setLanguage(locale);
        }
    }
    return super.onOptionsItemSelected(item);
}
```

Api de Voz

- No método “*onInit*” que iniciará o idioma coloque inglês:

```
@Override  
public void onInit(int i) {  
    textFala.setLanguage(Locale.ENGLISH);  
}
```

Api de Voz

- Depois insira as instruções do método “*botoes*”, começando pelo botão falar como abaixo:

```
private void botoes()
{
    btFalar = (Button)findViewById(R.id.btnFalar);
    btFalar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String texto = tMsg.getText().toString();
            textFala.speak(texto, TextToSpeech.QUEUE_FLUSH, Bundle.EMPTY, "1" );
        }
    });
}
```

Repare que há um método “speak” que é o responsável por falar o texto que foi digitado. Os parâmetros representam respectivamente a “string” falada, o segundo libera o recurso de áudio ao fim da fala, o terceiro é um parâmetro que pode ser armazenado entre a chamada de áudio e a liberação de recursos e o último um identificador do seu áudio.

Api de Voz

- Execute e faça o teste (mesmo sem internet);
- Depois acrescente mais idiomas pelo case.

Api de Voz

- Declare agora o método “*onActivityResult*”, acrescente as instruções para pegar o que foi captador pelo microfone, atribuído ao um “*arraylist de string*” que deverá ser atribuído para o “*listview*”.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if (resultCode== RESULT_OK)
    {
        ArrayList<String> palavras = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        lstV.setAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, palavras ));
    }
}
```

Api de Voz

- Crie agora o método que irá “preparar” o microfone.

```
protected Intent getRecognizerIntent()  
{  
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);  
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Fale aqui !");  
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, "pt-BR");  
    intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 10);  
    return intent;  
}
```

Instância
uma intent
de
microfone

Seta o
máximo de
possíveis
palavras
para retorno.

Seta o
idioma para
captação

Seta o texto
que vai
aparecer no
microfone

Api de Voz

- No método “*botoes*”, acrescente a instrução que irá chamar a “*intent do microfone*”, além disto, verifica se no seu dispositivo há suporte para o reconhecimento de voz.

```
btOuvirVoce = (Button)findViewById(R.id.btnOuvir);
btOuvirVoce.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        lstV = (ListView)findViewById(R.id.lstPalavras);
        //Verifica se ha suporte para reconhecimento de voz
        PackageManager pm = getPackageManager();
        List<ResolveInfo> activities = pm.queryIntentActivities(new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
        if (activities.size() != 0)
        {
            Intent intent = getRecognizerIntent();
            startActivityForResult(intent, 0);
        }
        else
        {
            Toast.makeText(getApplicationContext(), "Sem reconhecimento de voz", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Dúvidas

???

Referências

Google Android – 4º Edição – Ricardo R. Lecheta. Novatech São Paulo.
Capítulos 21.

Google Developer:

<https://developer.android.com/reference/android/speech/tts/TextToSpeech>
Acesso em 29/04/2017.