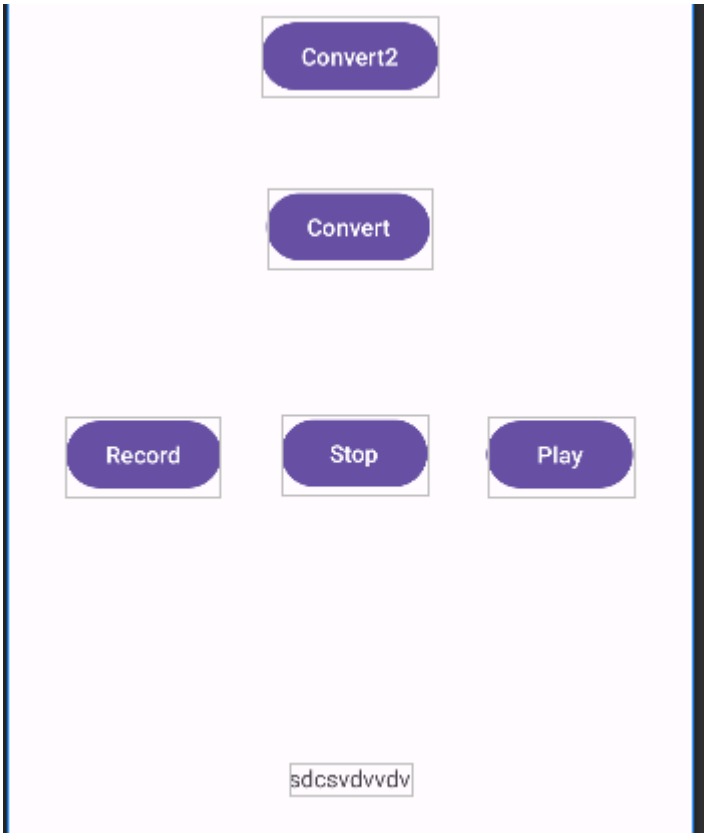
Documentation - VoiceRecorder



Working

- Press Record button to record audio.
- Press the Stop button to stop recording. (recording will be stored in local audio file)
- Press the Play button to Play the recording.
- Press the Convert button to display - the converted text + “ “ + its language code.
- Press the COnvert2 button to convert the text into English language.

To convert speech to text , Google API has been used and to translate the text into english language Google Translate API has been used.

Structure

- Add these lines to the *‘AndroidManifest.xml’* file.

```
4       <uses-permission android:name="android.permission.RECORD_AUDIO"/>
5       <uses-permission android:name="android.permission.INTERNET"/>
6       <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

- Add necessary dependencies in the *‘build.gradle(:app)’* files.
- Add the api credential file into the *‘res/raw/credential.json’* file.

- These functions check if the user has given the permission to use the device recorder and if not then ask the user to give the premission.

```
1 usage
259    private boolean isMicrophonePresent(){...}
1 usage
266    private void getMicrophonePermission(){...}
```

- The following function is called(triggered) when the user presses the Record button. Here we generate a local file in the music directory of the app and initialize the '*mediaRecorder'.*
- With '*file.getPath()*' we can assess the path of the local file.

```
75  public void btnRecord(View v) {
76      ContextWrapper contextWrapper = new ContextWrapper(getApplicationContext());
77      File musicDirectory = contextWrapper.getExternalFilesDir(Environment.DIRECTORY_MUSIC);
78      file = new File(musicDirectory , child: "test"+ ".wav");
79      try {
80          mediaRecorder = new MediaRecorder();
81          mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
82          //mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
83          mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
84          mediaRecorder.setOutputFile(file.getPath());
85          mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
86          mediaRecorder.setAudioChannels(2);
87          mediaRecorder.setAudioSamplingRate(44100);
88          mediaRecorder.prepare();
89          mediaRecorder.start();
90
91          Toast.makeText( context: this , text: "Recoding has started" , Toast.LENGTH_LONG).show();
92      }catch (Exception e){
93          e.printStackTrace();
94      }
95  }
```

- The following function is called(triggered) when the user presses the Stop button and Play Button.

```
96  public void btnStop(View v){...}
1 usage
102 public void btnPlay(View v){...}
```

- The following function is called when the user presses the *'Convert'* button. It uses google api to convert the short speech in the local audio file to text and the result will be shown in a text box on the user's screen. I have commented on the code in case you need a deeper understanding of this function.

```java
public void btnConvert(View v) throws IOException, InterruptedException {
    // Here we try to get relevant credentials from 'R.raw.credential' file.
    GoogleCredentials credentials;
    try {
        InputStream stream = this.getResources().openRawResource(R.raw.credential);
        credentials = GoogleCredentials.fromStream(stream);
    } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText( context: this,  text: "Error loading credentials", Toast.LENGTH_LONG).show();
        return;
    }

    // Create Speech-to-Text client with credentials generated above.
    SpeechClient speechClient;
    try {
        speechClient = SpeechClient.create(SpeechSettings.newBuilder().setCredentialsProvider(Fi
    } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText( context: this,  text: "Error creating SpeechClient", Toast.LENGTH_LONG).show(
        return;
    }
    /*
    All preferred languages(lang. code supported by google speech to text api.) of the user are
    We can add at most 3 languages here.

     */
    ArrayList<String> languageList = new ArrayList<>();
    languageList.add("en-IN");
    languageList.add("hi-IN");
```

- The following function is called when the user presses the '*Convert2*' button. It uses the '*getTranslateService()*' method to get the valid credentials and the '*translate(String textToTranslate, String sourceLanguage, String targetLanguage)*' method to get the final translated text. Again, I have commented on this code for a deeper understanding of the function.

```java
1 usage
public void btnConvert2(View v) throws IOException, InterruptedException, ExecutionException {
    textView = findViewById(R.id.textView);
    String Ss = textView.getText().toString();

    getTranslateService();
    String S = pares;
    if (lang.equals("en-in")) {
        S = translate(S, Language.ENGLISH, Language.HINDI);
    }
    S = translate(S, Language.HINDI, Language.ENGLISH);
    textView.setText(S);

}
```

Note : The codes defined in *'Laungauge.java'* file are valid only for google translate api. Google speech to text api slightly different language codes.