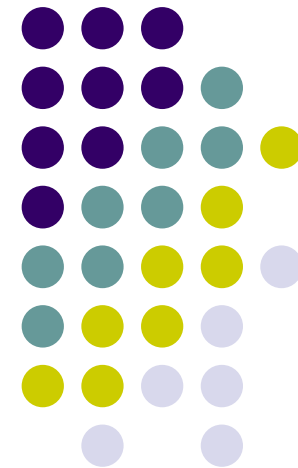


# Τεχνολογία και Προγραμματισμός Κινητών Συσκευών

9



**Text To Speech**  
Ιωάννης Έλληνας



# Text to Speech



- Είναι δυνατή η μετατροπή κειμένου σε ομιλία για διάφορες γλώσσες.
- Η χρησιμοποιούμενη κλάση είναι η **TextToSpeech** και ο ακροατής **OnInitListener**.

```
ttobj=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {  
    @Override  
    public void onInit(int status) {  
  
    }  
});
```

- Ο ορισμός της γλώσσας γίνεται με τη μέθοδο **setLanguage()**. Για παράδειγμα:

```
ttobj.setLanguage(Locale.UK);
```

- Η μετατροπή του κειμένου σε ομιλία γίνεται:

```
ttobj.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null, null);
```



- Άλλες χρήσιμοι μέθοδοι είναι:
  - ❑ **addSpeech(String text, String filename)** – Αντιστοίχιση κειμένου με αρχείο ήχου.
  - ❑ **getLanguage()** – Περιγράφει τη γλώσσα.
  - ❑ **isSpeaking()** – Ελέγχει τη διαθεσιμότητα της υπηρεσίας.
  - ❑ **setPitch(float pitch)** – Θέτει το speech pitch.
  - ❑ **setSpeechRate(float speechRate)** – Θέτει το speech rate.
  - ❑ **shutdown()** – Τερματίζει την υπηρεσία.
  - ❑ **stop()** – Τερματίζει την ομιλία.

# Speech to Text



- Η κλάση που υποστηρίζει το Speech to Text είναι η:

**`createSpeechRecognizer(Context)`**

- Οι μέθοδοι της κλάσης πρέπει να κληθούν από το κύριο νήμα. Η ομιλία αποστέλλεται σε έναν server στο διαδίκτυο ο οποίος μετατρέπει την ομιλία σε κείμενο το οποίο επιστρέφει στη συσκευή.
- Στο αρχείο μανιφέστου χρειάζονται οι άδειες:

```
<uses-permission android:name="android.permission.INTERNET" /> //online mode  
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

- Δημιουργούμε μια πρόθεση η οποία δείχνει το διάλογο με το μικρόφωνο για την αναγνώριση της ομιλίας.

```
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
```

- Με την πρόθεση μπορούμε να στείλουμε την γλώσσα σαν παράμετρο extra.

```
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "en-US");
```

- Χρησιμοποιούμε την **startActivityForResult()** καθώς περιμένουμε το κείμενο. Το κείμενο θα επιστρέψει στην **onActivityResult()** σαν ArrayList.



```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case RESULT_SPEECH: {
            if (resultCode == RESULT_OK && data != null) {
                ArrayList<String> text = data
                    .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            }break;
        }
    }
}
```



- Ένας δεύτερος τρόπος χωρίς να ενεργοποιήσουμε το διάλογο του μικροφώνου είναι να χρησιμοποιήσουμε τη διασύνδεση **RecognitionListener** και τις μεθόδους επανάκλησης που διαθέτει.
- Η αναγνώριση μπορεί να γίνει offline εφόσον έχουμε κατεβάσει στη συσκευή μας τα κατάλληλα πακέτα για τις γλώσσες που θέλουμε.
- Γλώσσα & Εισαγωγή → Αναζήτηση με φωνή → Αναγνώριση ομιλίας εκτός σύνδεσης.

```
public class DemoActivity extends Activity implements RecognitionListener {  
    SpeechRecognizer speech = SpeechRecognizer.createSpeechRecognizer(this);  
    speech.setRecognitionListener(this);  
    recognizerIntent = new Intent (RecognizerIntent.ACTION_RECOGNIZE_SPEECH);  
    recognizerIntent.putExtra (RecognizerIntent.EXTRA_LANGUAGE_PREFERENCE, "en");  
    recognizerIntent.putExtra (RecognizerIntent.EXTRA_LANGUAGE_MODEL,  
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);  
    recognizerIntent.putExtra (RecognizerIntent.EXTRA_MAX_RESULTS, 3);  
}
```

- Από τις πολλές μεθόδους επανάκλησης πρέπει να αναφέρουμε την **onError()** και την **onResults()**.

```

@Override
public void onError(int errorCode) {
    switch (errorCode) {
        case SpeechRecognizer.ERROR_AUDIO:
            message = R.string.error_audio_error;
            break;
        case SpeechRecognizer.ERROR_CLIENT:
            message = R.string.error_client;
            break;
        case SpeechRecognizer.ERROR_INSUFFICIENT_PERMISSIONS:
            message = R.string.error_permission;
            break;
        case SpeechRecognizer.ERROR_NETWORK:
            message = R.string.error_network;
            break;
        case SpeechRecognizer.ERROR_NETWORK_TIMEOUT:
            message = R.string.error_timeout;
            break;
        case SpeechRecognizer.ERROR_NO_MATCH:
            message = R.string.error_no_match;
            break;
        case SpeechRecognizer.ERROR_RECOGNIZER_BUSY:
            message = R.string.error_busy;
            break;
        case SpeechRecognizer.ERROR_SERVER:
            message = R.string.error_server;
            break;
        case SpeechRecognizer.ERROR_SPEECH_TIMEOUT:
            message = R.string.error_timeout;
            break;
        default:
            message = R.string.error_understand;
            break;
    }
}

```



```
@Override
public void onResults(Bundle bundle) {
    ArrayList<String> matches = bundle
        .getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
}
```



- Η έναρξη και η λήξη της ηχογράφησης γίνεται με τις:

```
SpeechRecognizer.startListening(recognizerIntent);
```

```
SpeechRecognizer.stopListening();
```

# Εργασία



Να γίνει εφαρμογή η οποία θα έχει σαν UI:

- Δύο LED (λευκό, κόκκινο και πράσινο)
- Ένα πλήκτρο SpeechToText.

Πατώντας το πλήκτρο και λέγοντας RED, το αντίστοιχο LED θα γίνεται κόκκινο ενώ λέγοντας GREEN το αντίστοιχο LED θα γίνεται πράσινο.

Αν πούμε CLEAR και τα δύο LED θα είναι λευκά.