

# TP1 : Traducteur « Français-Anglais »

Objectifs :

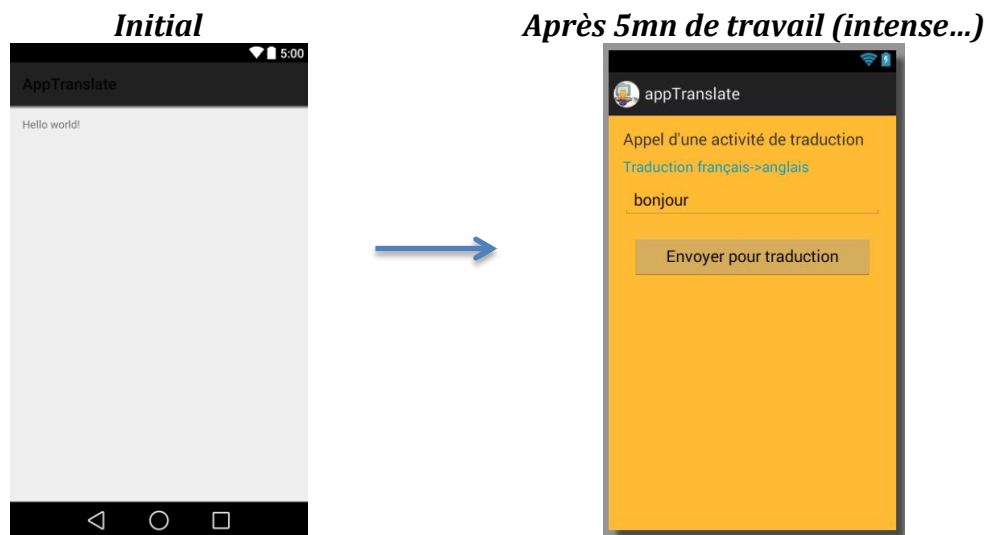
- créer deux activités, basculer entre activités, passer des paramètres entre activités,
- utiliser un service Internet pour faire réaliser une traduction,
- utiliser un composant intégré pour réaliser une synthèse vocale de la traduction.



## Mise en place de l'activité principale

1. Créer un nouveau projet **Android Studio Project** nommé **AppTranslate**, l'application portera le nom du projet et le paquetage sera nommé **net.votrenom.votreprenom.apptranslate** :
  - ✓ utiliser l'**API 18** comme sdk mini et choisir une « **Blank Activity** » comme activité de départ du projet,
  - ✓ conserver les noms par défaut de l'activité (**MainActivity.java**), du layout (**activity\_main.xml**) et du menu et donner comme titre de l'activité **AppTranslate**.

Voici le layout généré pour l'activité principale ainsi créée, et celui que vous devez obtenir... :



**Remarque :** vous pouvez utiliser l'icône fournie dans le dossier **Android\_Resources** pour « agrémenter » votre application. Pour modifier l'icône associée à l'application depuis l'explorateur de projet sélectionner l'onglet **res** puis clic droit **New->image asset**.

2. À vous de jouer, faites en sorte d'obtenir la vue présentée ci-dessus (à droite !).  
Vous allez à présent compléter le codage de la classe java associée à l'activité principale.
3. Compléter le codage de la classe **MainActivity** afin de lui ajouter deux attributs privés, un pour la zone d'édition « **EditText** » et l'autre pour le bouton « **Button** ».
4. Compléter la méthode surchargée **MainActivity.onCreate** afin :
  - ✓ d'initialiser les attributs précédents avec les références aux contrôles Widgets de l'activité,
  - ✓ d'associer un écouteur sur le bouton pour capturer l'événement **onClick** et déclencher la méthode associée.
5. Dans un premier temps, pour tester cette partie on peut se contenter d'afficher le texte du contrôle d'édition dans un message temporaire de type **Toast**.

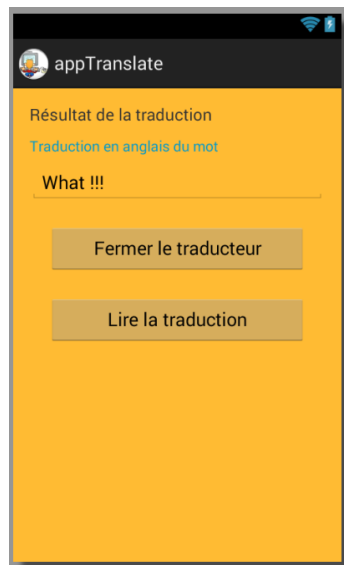
**Remarque :** vous aurez pu constater que le clavier est par défaut « ouvert » dès lors qu'il y a un contrôle d'édition « EditText » dans l'activité présente à l'écran. Pour annuler ce fonctionnement par défaut il faut demander à ce que le clavier soit caché pour l'activité. Cela se fait dans le manifeste, via l'attribut « **windowSoftInputMode** » de l'activité.

```
<activity
    android:name="net.marie.michel.apprtranslate.MainActivity"
    android:windowSoftInputMode="stateHidden">
</activity>
```

## Mise en place de l'activité enfant

Le clic sur le bouton « **Envoyer pour traduction** » devra ouvrir une « activité enfant » et lui passer le mot à traduire. Charge à cette activité de réaliser et d'afficher la traduction.

- Ajouter cette activité enfant, pour ceci :
  - ✓ ajouter au layout un nouveau fichier xml nommé **activity\_child** (répertoire **layout** puis **clic droit -> new -> layout resource file**),
  - ✓ ajouter au code source un nouveau fichier de code java nommé **ChildActivity** (répertoire **net.xxx.xxx.apprtranslate** puis **clic droit -> new->Java Calls**), la faire hériter de **Activity**. Ajouter la surcharge de la méthode **onCreate** (menu **Code -> Override Methods...** et choisir la méthode **onCreate(Bundle savedInstanceState):void**).
  - ✓ compléter le manifeste par la déclaration de cette activité.
- Faites en sorte de définir le layout de l'activité enfant afin d'obtenir la vue suivante :



*Comme c'est chouette...*

## Passer de la vue principale à la vue enfant

- Modifier le code java de la vue principale afin que le clic sur le bouton active l'activité enfant via une intention en lui passant en paramètre le mot à traduire. Supprimer le « Toast » devenu inutile.
- Compléter le code java de l'activité enfant afin de récupérer le paramètre reçu (le mot à traduire) et compléter la phrase affichée dans la TextView « **Traduction en anglais du mot** » avec le mot reçu.
- Vérifier le bon fonctionnement de l'ensemble.

## Traduction, utilisation d'un service Internet de traduction

Microsoft met à votre disposition un service de traduction gratuit que vous allez utiliser. Ce service fait partie des services Microsoft Windows Azure. Pour l'utiliser il faut créer un compte Microsoft puis à partir de celui-ci s'enregistrer sur Windows Azure, ce qui permet de demander des identifiants pour pouvoir utiliser le service « Translator »...Ouf !

Pour les besoins du TP je vous fournis le **Client\_ID** et le **Client\_SECRET** nécessaires à l'utilisation du service. Libre à vous pendant vos loisirs de passer une après-midi à vous créer votre propre compte et répondre aux multiples formulaires demandés...

Sauf oubli ils se trouvent dans un fichier **compteTranslator.txt** du dossier **Android\_Resources**.

Encore un point, qui n'est pas de détail, une bibliothèque java pour utiliser ce service est également fournie, « **microsoft-translator-java-api.jar** » que vous trouverez également dans le dossier **Android\_Resources**.

11. Ajouter la bibliothèque à votre projet, il suffit de coller celle-ci dans le dossier **libs** de votre projet. Il faut cependant avoir choisi le mode visualisation **Projet** (au lieu d'Android) dans la zone supérieure gauche, puis faire un clic droit sur le répertoire **libs** afin d'y copier le fichier **jar**. Il faut ensuite sélectionner le fichier **jar**, clic droit, puis dans le menu contextuel apparaissant choisir **Add As Library** pour enfin valider l'ajout du module dans **App**.

Pour vérifier que tout s'est correctement passé, choisir le fichier : « **build.gradle** » dans « **app** » et vérifiez dans la fenêtre de code s'ouvrant que vous avez bien une ligne : « **compile files(libs/microsoft-translator-java-api.jar)** ».

12. Compléter le code java de l'activité enfant afin de déclarer le **Client\_ID** et le **Client\_SECRET** comme attributs statiques finaux.
13. Compléter le code java de l'activité enfant afin, après avoir récupéré le paramètre reçu, de le faire traduire en utilisant la classe **Translate**<sup>1</sup>. Le résultat de la traduction sera affiché dans le contrôle « **EditText** ».

Mais qu'est-ce qu'il raconte ?? Ça ne marche pas son truc !!

En effet, il y a deux problèmes à résoudre pour espérer obtenir un résultat « correct » :

- ✓ pour utiliser le service Internet, il faut autoriser l'application à accéder à Internet,
- ✓ l'accès aux réseaux (Wifi ou 3G) est interdit depuis le « Thread UI », i.e. le thread de l'interface utilisateur qui gère les activités...

C'est à ce moment que l'auteur de cette..., pardon de ce sujet, devrait intervenir pour vous présenter la solution à ces problèmes si cela n'a pas encore été fait.

La solution étant donnée, on peut reprendre.

14. Compléter le manifeste avec la permission d'accès Internet.
15. Ajouter à votre classe **ChildActivity** une méthode de prototype ci-dessous qui réalisera la requête de traduction et fera afficher le résultat dans la zone « **EditText** ».

**private void translateFrtoEn(final String phraseAtraduire)**

16. Vérifier le bon fonctionnement de l'ensemble.

## **Pour le « fun », synthèse vocale du résultat de la traduction**

Les bibliothèques Android mettent à votre disposition une classe **TextToSpeech** qui permet la lecture vocale d'un texte spécifié dans une langue également spécifiée. Nous allons utiliser celle-ci

---

<sup>1</sup> Vous devez exploiter les méthodes statiques de la classe **Translate** : **setClientId**, **setClientSecret**, **execute**. La classe **Language** dispose de constantes permettant de spécifier le langage source et le langage de destination pour la traduction à réaliser à l'aide de la méthode **execute**.

pour, suite à un clic sur le bouton « Lire la traduction », réaliser la synthèse vocale de la phrase : « **La traduction du mot «mot en français » est « mot traduit en anglais »** ».

17. Ajouter à votre classe **ChildActivity** un attribut privé du type **TextToSpeech**.
18. Compléter la méthode **onCreate** afin d'instancier l'attribut de type **TextToSpeech**. L'utilisation de la classe **TextToSpeech** impose la prise en charge de l'interface **OnInitListener**, celle-ci imposant la surcharge de la méthode « **void onInit(int status)** » appelée à la fin de l'initialisation de l'instance pour signaler si l'objet est utilisable via le paramètre **status**.
19. Surcharger et coder la méthode **onInit** de l'interface **OnInitListener**. On se contentera de vérifier que l'initialisation est correcte, et dans ce cas de sélectionner l'usage de la langue française via la méthode **setLanguage**. Dans le cas contraire on signalera le problème avec un « **Toast** ».
20. Déclarer un attribut pour le bouton « **Lire la traduction** » et compléter la méthode surchargée **onCreate** afin :
  - ✓ d'associer un écouteur sur le bouton pour capturer l'événement **onClick** et déclencher la méthode associée,
  - ✓ de coder la méthode associée au clic pour qu'elle lise la phrase précédemment définie (« **La traduction du mot «mot en français » est « mot traduit en anglais »** »).
21. Ajouter à votre classe **ChildActivity** la surcharge de la méthode **Activity.onDestroy()** afin qu'elle libère les ressources de l'objet **TextToSpeech**.

## Quitter l'activité enfant en retournant une information à l'activité principale

On désire que lorsque l'on clique sur le bouton « **Fermer le traducteur** » cela ferme l'activité enfant pour revenir à l'activité principale en lui retournant le mot traduit. L'activité principale devra alors afficher le mot traduit dans un « **Toast** ».

22. Faites en sorte que cela fonctionne comme demandé !  
Quelques indications tout de même...  
Côté activité principale, il faut :
  - ✓ remplacer la méthode **startActivity** ayant servi à activer l'activité enfant par la méthode **startActivityForResult**,
  - ✓ surcharger la méthode **Activity.onActivityResult** qui sera appelée lorsque la vue enfant retournera le paramètre résultat via une intention à destination de la vue principale.Côté activité enfant, il faut :
  - ✓ associer un écouteur sur le bouton pour capturer l'événement **onClick** et faire en sorte que cet événement déclenche le retour à l'activité principale en lui passant le mot traduit comme valeur de retour via une intention.

23. Vérifier le bon fonctionnement de l'ensemble.

## Extension possible : ajout de la « localisation »

S'il vous reste du temps et un peu de courage, vous pouvez « internationaliser » votre application afin qu'elle prenne en compte les langues anglais et français. Vous n'avez aucune excuse à ne pas le faire, compte tenu de l'efficacité de l'application que vous venez de créer, qui va vous aider à traduire les chaînes de ressources !!!

Enfin il serait bien que la phrase lue par la synthèse vocale tienne également compte de la langue.

Quelques pistes :

- ✓ Il est possible de placer une chaîne paramétrée dans les fichiers de ressources « **string.xml** ». Par exemple la déclaration suivante prend en compte deux paramètres : **<string name="phraseParams">L'âge de %1\$s est, %2\$d.</string>**  
Qu'il est possible d'exploiter dans le code java comme ci-dessous :  
**String prenom = "Michel";**  
**Int age = 40 ;**  
**String phraseResultat ;**  
**phraseResultat = getResources().getString(R.string.phraseParams, prenom, age);**
- ✓ La classe statique **Locale** permet de connaître la langue configurée sur le périphérique et de la comparer.  
Pour obtenir le code de la langue configurée : **Locale.getDefault().getLanguage();**  
Pour obtenir le code d'une langue spécifiée (anglais) : **Locale.ENGLISH.getLanguage();**

Pour la traduction des ressources il est conseillé d'utiliser le « **Translations Editor** ». Pour ceci sélectionner un fichier de ressources **string.xml** puis **clik droit ->Open Translations Editor**

Key	Default Value	Untranslatable	English (en)	French (fr)
action_settings	Settings	<input type="checkbox"/>	Settings	Settings
app_name	appTranslate	<input type="checkbox"/>	appTranslate	appTranslate
btnReconnaitre	Reconnaissance vocale	<input type="checkbox"/>	Vocal recognizer	Reconnaissance vocale
btnRetourChild	Fermer le traducteur	<input type="checkbox"/>	Close the translator	Fermer le traducteur
btnTraduire	Envoyer pour traduction	<input type="checkbox"/>	Send to translate	Envoyer pour traduction
btnTraduireChild	Lire la traduction	<input type="checkbox"/>	Read the translation	Lire la traduction
gsmON	La connexion GSM est à présent active	<input type="checkbox"/>	The GSM connection isn't enabled	La connexion GSM est à présent active
msgToast1	Ouvrir l'activité de traduction	<input type="checkbox"/>	Open the translate activity	Ouvrir l'activité de traduction
msgToast2	Reconnaissance vocale démarrée	<input type="checkbox"/>	Reconnaissance vocale démarrée	Reconnaissance vocale démarrée
msgToast3	Reconnaissance vocale non supportée, évitez d'ache	<input type="checkbox"/>	Speech recognizer not supported, avoid buying Chinese !!!	Reconnaissance vocale non supportée, évitez d'acheter chinois !
msgToast4	"Traduction annulée.."	<input type="checkbox"/>	"Translation canceled.."	"Traduction annulée.."
msgToast5	"Traduction du mot en anglais : %1\$s"	<input type="checkbox"/>	"The english translation of the word: %1\$s"	"Traduction du mot en anglais : %1\$s"
phraseFormatFR	La traduction du mot %1\$s est	<input type="checkbox"/>	The translation of the word %1\$s is	La traduction du mot %1\$s est
txtAccueil	Appel d'une activité de traduction	<input type="checkbox"/>	Request a translation activity	Appel d'une activité de traduction
txtAccueilChild	Résultat de la traduction	<input type="checkbox"/>	Translation result	Résultat de la traduction
txt_Titre1	Traduction français-> anglais	<input type="checkbox"/>	Translate french-> english	Traduction français-> anglais
txt_Titre1Child	Traduction en anglais du mot	<input type="checkbox"/>	English translation of the word	Traduction en anglais du mot
txt_enter_your_mot	Taper un mot à traduire	<input type="checkbox"/>	Enter a word to translate	Taper un mot à traduire
txt_enter_your_motChild	What !!	<input type="checkbox"/>	What !!	What !!
txt_motSource	fabuleux	<input type="checkbox"/>	fabulous	fabuleux
txt_motSourceChild	What !!!	<input type="checkbox"/>	What !!!	What !!!
wifiON	La connexion WIFI est à présent active	<input type="checkbox"/>	The Wifi connection isn't enabled	La connexion WIFI est à présent active
wifigsmOFF	Les connexions réseau sont à présent inactives	<input type="checkbox"/>	The network connections are disabled	Les connexions réseau sont à présent inactives

*Sujet, ressources et corrigé :*

