

Application Android

Rapport Sleepy Finder



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Johnny Haymoz et Fabien Perroud

Informatique, Classe I2A

Johnny.haymoz@edu.hefr.ch

Fabien.perroud@edu.hefr.ch

<h2>Table des matières</h2>

1	Introduction	3
1.1	Cahier des charges.....	3
2	Analyse	3
2.1	Accès à Google Place.....	3
2.2	Filtrage des Place	4
2.3	Sélection du périmètre	5
2.4	Web API	6
2.4.1	Sécurité.....	6
2.4.2	Parsing.....	7
2.5	Calcul de distance	7
3	Conception	9
3.1	Information d'un point d'intérêt.....	10
4	Implémentation	10
5	Conclusion	11
5.1	Problème.....	11
5.2	Inutilisabilité de l'API Google Place pour Android	11
5.3	Remarques	12
5.4	Organisation.....	12
6	Références.....	12
7	Liste des figures	13

1 Introduction

Le but de ce projet est de créer une application Android qui permette de récupérer des informations sur le lodging dans un périmètre sélectionnable en fonction de la position de l'appareil.

1.1 Cahier des charges

Une application Android qui:

- Collecte et liste les appartements/studios et hotels à louer dans un périmètre sélectionnable.
- L'utilisation de Google Place vous permettra d'obtenir des informations sur le "lodging" disponible dans un secteur variable autour de la position de l'appareil.

L'app devra est compatible avec Android 4.4 mais devra être développée en Android 6 avec les nouvelles fonctionnalités (comme le Drawer). Attention aux permissions Android 6!

2 Analyse

2.1 Accès à Google Place

Pour pouvoir faire des requêtes sur la base de donnée Google Place il faut avoir un compte sur <http://console.developers.google.com/>.

Dans le cadre de ce projet, la version d'essai gratuite 3 mois a été utilisée.

Dans tous les cas il est précisé que: "Usage of the Google Places API for Android is free and unlimited for all apps." Dans la limite des 1000 requêtes par jours.

Une fois notre compte créé nous devons générer des Credentials pour notre application.

API keys

<input type="checkbox"/> Name	Creation date ▾	Type	Key
<input type="checkbox"/> MapKey	Aug 23, 2016	Android	AIzaSyBmNGkWMxml_wQ2Jy8s

Figure 1: Credentials de l'application

A partir de ce point nous pouvons passer à la configuration dans l'IDE.

Premièrement il nous faut ajouter les services Google Play à Android Studio.

☑	+	Extras		
☑	+	Android Support Repository	11	Update available: rev. 36
	+	Android Auto Desktop Head Unit emula	1.1	Not installed
☑	+	Google Play services	32	Not installed
☑	+	Google Repository	15	Update available: rev. 32
	+	Google Play APK Expansion library	1	Not installed
	+	Google Play Licensing Library	1	Not installed
	+	Google Play Billing Library	5	Not installed
	+	Android Auto API Simulators	1	Not installed
☑	+	Google USB Driver	11	Not installed
	+	Google Web Driver	2	Not installed
☑	+	Intel x86 Emulator Accelerator (HAXM)	5.2	Update available: rev. 6...

Figure 2: Ajout des services Google Play

Il ne reste plus qu'à définir les permissions nécessaires pour que l'application fonctionne et à mettre les Credentials dans le AndroidManifest.xml

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyBmNGkWMxmI_wQ2Jy8s07-5rXAYVnwdbyY"/>
```

Figure 3: Permissions et Credentials dans le Manifest

2.2 Filtrage des Place

Il est possible de créer un PlaceFilter permettant de filtrer la recherche des Places. Dans notre cas il faut nous assurer que la place soit de type lodging.

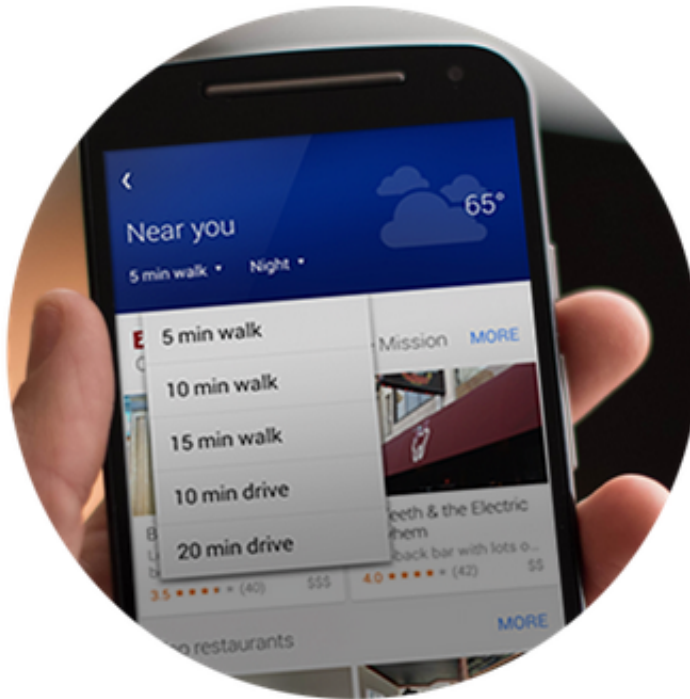
On peut le faire en utilisant la méthode: **public boolean matches (Place place)**

Il faut ensuite vérifier que **getPlaceTypes ()** contienne TYPE_LODGING.

Il n'est malheureusement pas possible de filtrer sur le PlacePicker ce qui simplifierait grandement le développement.

2.3 Sélection du périmètre

API Features



Location Awareness

Use the power of **mobile** to give your users contextual information about where they are, when they're there.

Figure 4: Publicité "mensongère"

Même si sur la page principale de Google Place il semble être possible de filtrer les recherches en fonction de la distance à laquelle l'utilisateur est d'accord de marcher, en pratique la fonction pour faire les requêtes ne le permet pas.

La seule fonction permettant de faire des requêtes est **getCurrentPlace** ([GoogleApiClient](#) client, [PlaceFilter](#) filter) `getCurrentPlace (GoogleApiClient client, PlaceFilter filter)`

Sauf que l'on doute que la distance des lieux retourner par cette fonction puisse aller jusqu'à 20 km (60km pendant 20 minutes comme montré sur la figure 4.).

2.4 Web API

L'API Web de Google Place contrairement à celle pour Android fait tout ce que l'on veut.

Les requêtes se font par GET en spécifiant l'emplacement central, et le radius de recherche en plus d'autre paramètre optionnel comme le type.

Voici un exemple de requête retournant les lieux à 500 mètres de l'école d'ingénieur étant de type lodging.

`https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=AlzaSyA9F4HNPUw_GGjeK-YDDz1IZUuEZA47Y38&location=46.792631,7.159997&radius=500&type=lodging`

Il y a cependant un problème majeur dans le fait d'utiliser la Web API dans notre projet.

2.4.1 Sécurité

Pour une application Android on utilise une Android Key sur laquelle on peut mettre des restrictions pour s'assurer qu'un utilisateur malveillant ne nous vole pas notre clé. Dans la version Web API on ne peut pas protéger notre clé de cette manière le seul moyen de protéger notre clé est de faire un filtrage par adresse IP.

Server API key

This key should be kept secret on your server

Every API request is generated by software running on a machine that you control. [Learn more](#)

API key	AlzaSyA9F4HNPUw_GGjeK-YDDz1IZUuEZA47Y38
Creation date	Aug 27, 2016, 5:06:19 PM
Created by	perroudfabien@gmail.com (you)

Name

Accept requests from these server IP addresses (Optional)

Examples: 192.168.0.1, 172.16.0.0/12, 2001:db8::1 or 2001:db8::/64

Note: It may take up to 5 minutes for settings to take effect

Figure 5: Configuration d'une clé serveur

Il serait donc possible pour un utilisateur malveillant d'analyser le réseau avec WireShark et de récupérer notre clé en analysant les requêtes html.

Dans le cas où l'on décide de faire les requêtes directement depuis le téléphone il n'est tout simplement pas possible de faire une restriction par adresse IP. On arrive donc rapidement à la conclusion qu'il faut un serveur entre notre application et le service google place. Ce qui soulève plusieurs problèmes:

- Il faut mettre en place un serveur ce qui n'est pas le thème du module
- Dans un contexte professionnel, si l'application fait un énorme boum et que le nombre de requête s'envole, le goulet d'engorgement sera le serveur qui traite les requêtes.
- On ajoute une étape et donc le temps de latence va doubler. Ce qui va inéluctablement réduire la fluidité de l'application.

Nous avons mis en place un serveur permettant de rediriger les requêtes mais celui-ci ne fait aucun check, on ne donne donc pas la clé directement à l'utilisateur mais n'importe qui peut utiliser la page pour obtenir des informations

D'autant plus que même en utilisant un serveur, il faut encore s'assurer que l'utilisateur ait le droit de faire des requêtes car il serait toujours possible d'utiliser notre clé en utilisant l'url de notre serveur et pour cela il faudrait encore ajouter un système d'authentification ce qui ajoute encore un niveau de complexité supplémentaire.

2.4.2 Parsing

Un autre inconvénient, bien moins grave cette fois, dans la version android pure on travaille, avec des objets directement et tout est simple d'utilisation, en utilisant l'api web on se retrouve à devoir parser du json et le convertir en objet, un moindre mal mais cela crée du travail supplémentaire pour au final pas grand-chose.

2.5 Calcul de distance

Il faut premièrement prendre en compte que les distances sont calculées à vol d'oiseau un facteur arbitraire de 0.66 sera appliqué sur les distance pour compenser les détours.

Ayant besoin de donner un rayon en mètre de recherche et trouvant que les champs du type "5 minutes à pied" ont plus d'impact que juste "300 mètres" nous utiliserons les variables suivantes:

"Many people tend to walk at about 1.4 m/s"

5 minutes de marche: 277 mètres

10 minutes de marche: 554 mètres

20 minutes de marche: 831 mètres

Pour les trajets à voiture on s'est basé sur la vitesse moyenne lors d'un tour à moto de 430 km.



Figure 6: Vitesse moyenne véhicule

On a donc une vitesse moyenne de 46km/h ce qui à première vue semble plutôt lent mais quand on prend en compte le trafic c'est une vitesse plutôt rapide. Sur le trajet en question il y a 150 km sur l'autoroute et une moto se déplace quand même plus facilement qu'une voiture. Faire les calculs en utilisant une vitesse de déplacement moyenne de 25km/h semble donc adapter. On arrive donc à une distance de 416 m/minute

5 minutes en voiture: 2 km.

10 minutes en voiture: 4 km.

20 minutes en voiture: 8 km.

3 Conception

Il y aura une vue principale avec la carte centrée sur la position actuelle de l'utilisateur.



Figure 7: Ebauche de vue

Les lieux à proximité seront affichés sur la carte, lors d'un clic une vue avec les informations sur le lieu sera affichée dans une DialogBox

Sur la vue principale un Spinner similaire à celui de la figure 4 sera ajouté pour sélectionner la distance de vue.

3.1 Information d'un point d'intérêt

Lors de l'affichage des résultats d'une recherche nous ajoutons des markers sur la carte de Google. En cliquant sur un marker il sera possible d'obtenir des informations supplémentaires qui s'afficheront dans une dialogBox comme le montre la maquette de la figure 8.

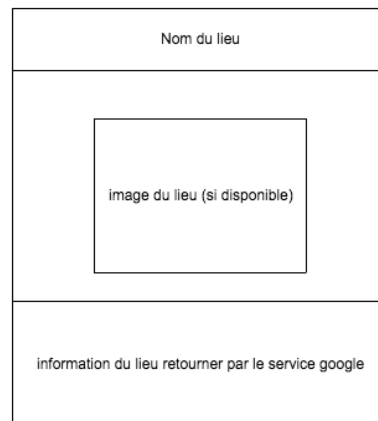


Figure 8: Maquette de la dialogBox d'information d'un lieu

4 Implémentation

L'application démarre sur la dernière position connue de l'appareil. De ce fait si la localisation GPS du téléphone est momentanément désactivée (ou non fonctionnelle pour diverses raisons), l'application pourra quand même fonctionner. La valeur par défaut du Spinner permettant de choisir le périmètre de recherche est à "5 minutes à pieds".

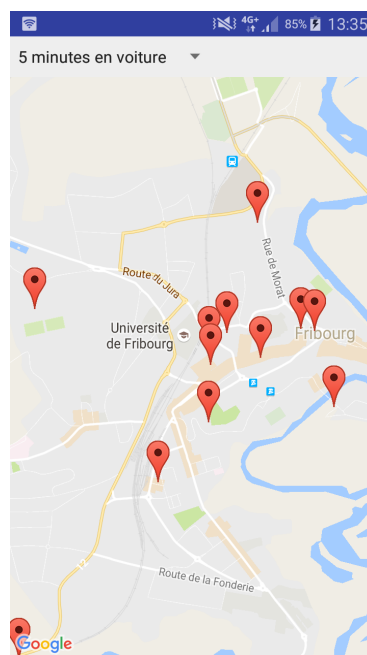


Figure 9: Capture d'écran de l'application finale

Lors du choix d'un périmètre différent de celui par défaut, il faut légèrement faire bouger la carte pour que les nouveaux marqueurs apparaissent. Lors d'un "clic" sur un marqueur, une DialogBox s'affiche à l'écran avec les informations concernant le marqueur en question comme le montre la figure 10.

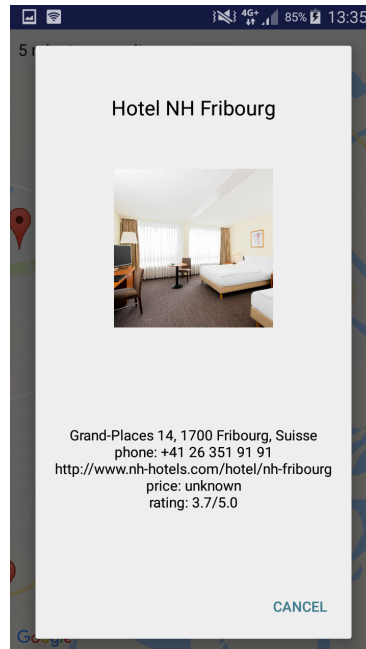


Figure 10: Capture d'écran des informations d'un lieu

5 Conclusion

5.1 Problème

Nous avons perdu beaucoup de temps sur la mise en place de l'environnement Android Studio. Au lieu de simplement réinstaller l'environnement nous avons décidé de mettre à jour celui-ci. Ce qui a créé pas mal de problème au niveau du gradle.build.

5.2 Inutilisabilité de l'API Google Place pour Android

L'API pour Android semble alléchante à première vue mais n'est vraiment pas adapté pour une utilisation spécifique comme celle que l'on veut.

Le PlacePicker qui pourtant fait exactement ce qu'on veut, l'application pourrait être faite en 5 minutes, on lance un PlacePicker qui n'affiche que les lieux en lodging, quand un lieu est sélectionné on affiche des informations supplémentaire et éventuellement l'itinéraire, rien de plus simple. Sauf qu'il n'est pas possible de filtrer les informations qu'affiche le PlacePicker (voir la référence sur l'issue 8484).

A côté de ça on pourrait espérer une fonction de recherche décente dans [GeoDataApi](#).

Malheureusement, on cherche toujours cette fonction. La seule fonction de recherche à disposition étant dans **PlaceDetectionApi** mais la méthode à disposition n'est pas satisfaisante non plus.

getCurrentPlace ([GoogleApiClient](#) client, [PlaceFilter](#) filter)

On peut cette fois définir le type de lieu que l'on souhaite voir, mais il n'est pas possible de chercher des lieux à un emplacement différent de l'actuel ce qui est très légèrement limitatif. Il n'est pas non plus possible de définir le radius de recherche.

5.3 Remarques

N'ayant pas fait de développement Android depuis longtemps et n'ayant pas forcément énormément de temps à disposition (l'un de nous ayant un job d'été à 100% durant toute la période mise à disposition pour la réalisation du projet) nous avons préféré mettre l'accent sur l'analyse des fonctionnalités proposées par Google Place plutôt que sur la partie Android en soi. De plus nous estimons que le manque de 2 crédits ETCS pour le module "Systèmes d'Information et Applications Mobiles" aurait pu être constaté largement plus tôt dans l'année par le service académique lors de l'augmentation des crédits de ce cours (En effet nous pensions que les crédits nous seraient automatiquement accordés étant donné que nous avions réussi le cours d'Android avec une note largement suffisante).

5.4 Organisation

M. Haymoz ayant un job d'été à 100% du 11 juillet au 9 septembre, il nous a été difficile de nous organiser pour des réunions de travail car nous n'étions pas forcément disponibles tous les deux chaque week-end de cette période. C'est pour cette raison que le projet est rendu avec une semaine de retard par rapport à la date de rendu initialement fixée et nous vous prions de nous en excuser.

6 Références

<https://developers.google.com/places/android-api/start>

<https://developers.google.com/android/reference/com/google/android/gms/location/places/PlaceFilter>

<https://code.google.com/p/gmaps-api-issues/issues/detail?id=8484>

<https://developers.google.com/places/web-service/search>

https://en.wikipedia.org/wiki/Preferred_walking_speed

7 Liste des figures

Figure 1: Credentials de l'application

Figure 2: Ajout des services Google Play

Figure 3: Permissions et Credentials dans le Manifest

Figure 4: Publicité "mensongère"

Figure 5: Configuration d'une clé serveur

Figure 6: Vitesse moyenne véhicule

Figure 7: Ebauche de vue

Figure 8: Maquette de la DialogBox d'information d'un lieu

Figure 9: Capture d'écran de l'application finale

Figure 10: Capture d'écran des informations d'un lieu