



*Cards shown are not guaranteed.
Duplication may occur.

Projet Android Magic : The Gathering

lionel.buathier@univ-lyon1.fr

Introduction

- ▶ On souhaite réaliser une application Android qui permet de collectionner des cartes Magic : L'assemblée (*The Gathering*), de les échanger entre "amis" et éventuellement de gérer une partie.
- ▶ Pour se faire, l'application mobile devra interroger l'API REST d'un serveur distant que vous réaliserez dans le cadre du module « Services Web ». Ce serveur gérera la liste de cartes de chaque utilisateur, la possibilité de les échanger, de gérer une partie, etc.
- ▶ L'application permettra à chaque utilisateur de :
 - de consulter sa liste de cartes,
 - de constituer un ou plusieurs decks,
 - d'échanger des cartes avec d'autres utilisateurs,



Organisation

- ▶ Les modules Android I/II/III et Services Web sont interdépendants et apparaîtront alternativement dans le calendrier
- ▶ Nous allons procéder par étapes :
 - 1. Authentification par Facebook (et/ou Google)
 - 2. Interrogation de l'API et mise à jour de la liste de cartes de l'utilisateur
 - 3. Affichage des cartes de l'utilisateur
 - 4. Création du ou des decks
 - 5. Gestion des échanges de cartes
 - 6. fonctionnalités supplémentaires





Institut Universitaire de Technologie

Département Informatique
Site de Bourg-en-Bresse



Authentification Facebook / Google

lionel.buathier@univ-lyon1.fr

Facebook Login

- ▶ Permet aux utilisateurs de se connecter à une app via le SDK Facebook.
- ▶ Lorsque les utilisateurs se connectent à l'app avec Facebook, ils peuvent autoriser celle-ci à récupérer des informations sur leur compte ou exécuter des actions sur Facebook en leur nom.
- ▶ Le SDK 4 Facebook, comporte une intégration automatique à 'Facebook Lite' pour afficher l'écran de connexion et obtenir des identifiants, sans avoir à installer l'app Facebook (qui était nécessaire auparavant) :
<https://developers.facebook.com/docs/facebook-login/android>
- ▶ **Rq:** peut nécessiter d'installer openSSL pour générer la clé SHA1:
<https://indy.fulgan.com/SSL/>
version d'appcompat à déclarer : '`com.android.support:appcompat-v7:25.3.1`'
- ▶ <https://www.studytutorial.in/android-facebook-integration-and-login-tutorial>
- ▶ <https://www.androidtutorialpoint.com/material-design/android-facebook-login-tutorial/>



Jeton d'accès (Access Token)

- ▶ Lorsqu'une personne se connecte avec une app à l'aide de Facebook Login, l'app peut obtenir un jeton d'accès qui fournit un accès provisoire et sécurisé aux API Facebook.
- ▶ Un jeton d'accès est une chaîne opaque qui identifie un utilisateur, une app ou une page, et que l'app peut utiliser pour passer des appels de l'API Graph. Plusieurs méthodes permettent d'obtenir des tokens d'accès.
- ▶ Le jeton contient notamment sa date d'expiration et le nom de l'app qui l'a généré. La plupart des appels d'API sur Facebook doivent contenir un jeton d'accès, en raison des contrôles de confidentialité. Il existe différents types de jetons d'accès pour prendre en charge les divers cas d'utilisation, mais le **jeton utilisateur** est le plus fréquemment utilisé.

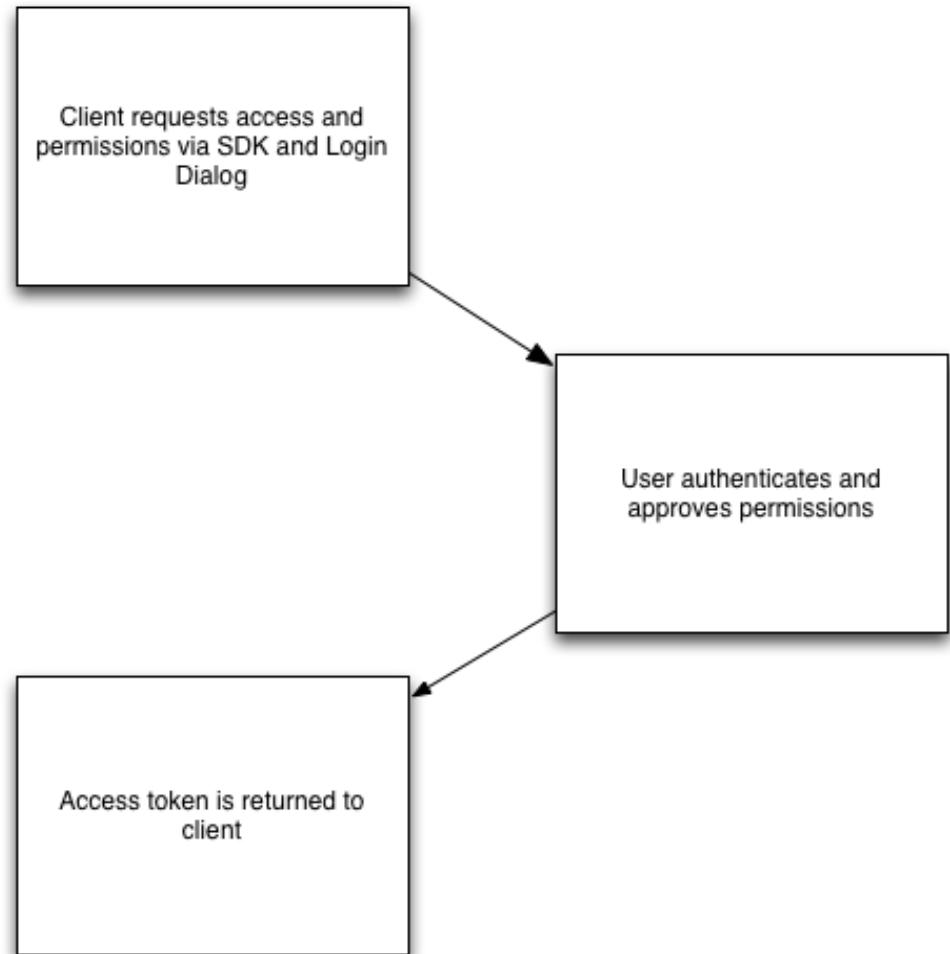
▶ <https://developers.facebook.com/docs/facebook-login/access-tokens>



Token d'accès

Le SDK Facebook pour Android gère automatiquement les jetons d'accès utilisateur par l'intermédiaire de la classe [com.facebook.AccessToken](#).

On peut récupérer le jeton d'accès utilisateur en inspectant
[Session.getCurrentAccessToken](#)



Google Sign In

- ▶ L'API Google Android permet également d'authentifier simplement un utilisateur avec les mêmes certificats utilisés sur Google.
- ▶ Elle permet de proposer des fonctionnalité "sociales" et de contrôler l'usage de l'application.
- ▶ Elle fait partie de la plateforme **Google Play services**. Il faut donc activer le SDK Google Play services dans le projet.
- ▶ Même principe de jeton d'accès que pour Facebook



Google Sign In

- ▶ Start Integrating Google Sign-In into Your Android App

<https://developers.google.com/identity/sign-in/android/start-integrating>

- ▶ Integrating Google Sign-In into Your Android App

<https://developers.google.com/identity/sign-in/android/sign-in>

Autre ressource :

- ▶ Adding Google Login to Android App

<https://www.codeproject.com/Articles/1113772/Adding-Google-Login-to-Android-App>



Google Sign In

Rq: Pour générer la clé SHA1 et obtenir le fichier google-services.json :

Sous windows, exécuter cmd.exe **en mode administrateur**

```
C:\Program Files\Android\Android Studio\jre\bin>keytool -exportcert -list -v -alias androiddebugkey -keystore %USERPROFILE%\.android\debug.keystore
```

Entrez le mot de passe du fichier de clés : android

Configuration du build.gradle 'top level' du projet:

Ajouter dans les dependencies du buildscript :

```
classpath 'com.google.gms:google-services:3.1.0'
```

