

Scanner l'ISBN d'un livre

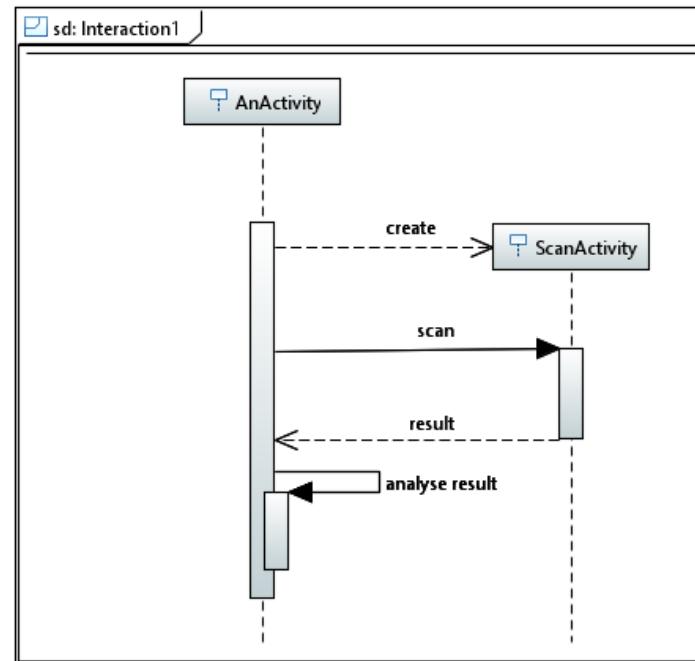
Cédric Dumoulin

2016

Scan d'un code barre

Scan : Principe général

- Appel à une application/Activity de scan
 - L'Activity prend en charge le scan
 - Elle retourne le résultat du scan



Library de scan

- Library API google Play
- Library ZXing

Scan avec API Google Play

Prerequisites

- Android Studio
- Android 4.2.2 or later
 - -or- A configured Android Emulator (this is available in Android Studio)
- Android SDK including the SDK tools component.
- **Google Play Services SDK.**

Exemples

- Barcodes – tirés du codelab
 - <https://codelabs.developers.google.com/codelabs/barcodes>
 - Scan du barcode dans une image
- Barcode-reader
 - Exemple officiel
 - <https://github.com/googlesamples/android-vision>
 - Scan de plusieurs barcode
 - Exemple reutilisable

Exemple Barcodes

- <https://codelabs.developers.google.com/codelabs/barcodes>
- Tutorial d'introduction de google
- Fonctionnement :
 - Lit une image en mémoire
 - Crée un BarcodeDetector
 - Scan l'image
 - Reçoit un tableau de résultats
 - Lit le résultat



```
// Get image from file
Bitmap myBitmap = BitmapFactory.decodeResource(
    getApplicationContext().getResources(), R.drawable.puppy);

// Get barcode detector
BarcodeDetector detector =
    new BarcodeDetector.Builder(getApplicationContext())
        .setBarcodeFormats(Barcode.DATA_MATRIX | Barcode.QR_CODE).build();

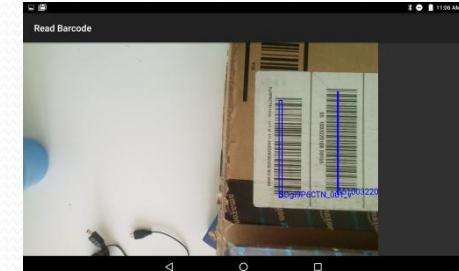
// Create a frame with the image, and call detector
Frame frame = new Frame.Builder().setBitmap(myBitmap).build();
// Read barcodes
SparseArray<Barcode> barcodes = detector.detect(frame);

// Show result
Barcode thisCode = barcodes.valueAt(0); // Takes first element
txtView.setText( thisCode.rawValue ); // Read result
```

- Avantages
 - Simple à mettre en oeuvre
- Inconvénients
 - Lit le barcode dans une image statique
 - Ne fait pas de scan ‘live’

Exemple Barcode-reader

- <https://github.com/googlesamples/android-vision>
 - Barcode-reader
- Une Activity principale – MainActivity
 - Appelle l'activité de scan
 - Analyse le résultat
- Une Activity de scan – BarcodeCaptureActivity
 - Permet de scanner des codes barres
 - Retourne un tableau de résultats



Appeler l'activité de scan

- Créer un Intent

```
// launch barcode activity.  
Intent intent = new Intent(this, BarcodeCaptureActivity.class);  
intent.putExtra(BarcodeCaptureActivity.AutoFocus, autoFocus.isChecked());  
intent.putExtra(BarcodeCaptureActivity.UseFlash, useFlash.isChecked());  
  
startActivityForResult(intent, RC_BARCODE_CAPTURE);
```

- Le résultat sera envoyé à la méthode
onActivityResult(...) de l'activity appelante

Réception du résultat onActivityResult(...)

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == RC_BARCODE_CAPTURE) {  
        if (resultCode == CommonStatusCodes.SUCCESS) {  
            if (data != null) {  
                Barcode barcode = data.getParcelableExtra(BarcodeCaptureActivity.BarcodeObject);  
                statusMessage.setText(R.string.barcode_success);  
                barcodeValue.setText(barcode.displayValue);  
                Log.d(TAG, "Barcode read: " + barcode.displayValue);  
            } else {  
                statusMessage.setText(R.string.barcode_failure);  
                Log.d(TAG, "No barcode captured, intent data is null");  
            }  
        } else {  
            statusMessage.setText(String.format(getString(R.string.barcode_error),  
                CommonStatusCodes.getStatusCodeString(resultCode)));  
        }  
    } else {  
        super.onActivityResult(requestCode, resultCode, data);  
    }  
}
```

Barcode-reader

- Avantages
 - Exemple réutilisable
- Inconvénients
 - Lit plusieurs barcodes
 - Nécessite un click supplémentaire de la part de l'utilisateur pour confirmer la saisie

Documents

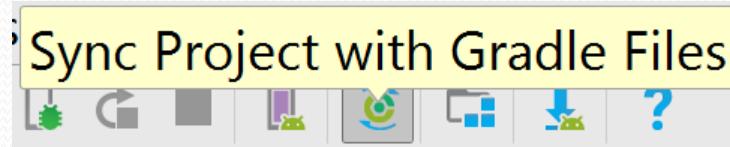
- <https://codelabs.developers.google.com/codelabs/barcodes>

Librairies

- Dans build.gradle (app)
- Ajoutez :
 - compile '**com.android.support:appcompat-v7:24.2.1**'
 - Si nécessaire
 - compile '**com.google.android.gms:play-services:7.8+**'

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
  
    compile 'com.android.support:appcompat-v7:24.2.1'  
    compile 'com.google.android.gms:play-services:7.8+'  
}
```

- Synchronisez graddle :



Scan avec ZXing

Zxing – Zebra Crossing

- Utiliser la bibliothèque zxing
 - <https://github.com/zxing/zxing/>
- Exemple d'utilisation
 - Voir exemple
 - **Android SDK: Create a Book Scanning App**

Zxing – Lancer le scan

- Crée un objet IntentIntegrator
 - Cet objet lancera l'activity avec les bons paramètres
- Lance le scan `scanIntegrator.initiateScan();`
 - Cela lance l'activity avec les bons paramètres

```
public void onClick(View v) {  
    //check for scan button  
    if(v.getId()==R.id.scan_button) {  
        //instantiate ZXing integration class  
        IntentIntegrator scanIntegrator = new IntentIntegrator(this);  
        //start scanning  
        scanIntegrator.initiateScan();  
    }  
}
```

- Le résultat est envoyé à la méthode `onActivityResult(...)` de l'activity appelante

Zxing – Récupérer le résultat onActivityResult(...)

- IntentResult : classe Zxing, récupère le résultat du scan.

```
public void onActivityResult(int requestCode, int resultCode, Intent intent) {  
    //retrieve result of scanning - instantiate ZXing object  
    IntentResult scanningResult  
        = IntentIntegrator.parseActivityResult(requestCode, resultCode, intent);  
    //check we have a valid result  
    if (scanningResult != null) {  
        //get content from Intent Result  
        String scanContent = scanningResult.getContents();  
        //get format name of data scanned  
        String scanFormat = scanningResult.getFormatName();  
        Log.v("SCAN", "content: "+scanContent+" - format: "+scanFormat);  
    }  
}
```

Déclarer Xzing dans son code

- 2 méthodes :
 - 1) Partager Xzing entre les applications
(== installer Xzing sur la tablette)
 - 2) Utiliser la librairie Xzing embarqué

Méthode 1

Partager Zxing

- Le scanneur est partagé entre toutes les applications
- Si le scanneur n'est pas installé sur la tablette, un popup demande si on veut l'installer
- Avantages
 - Le scanneur est partagé : gain de place
- Inconvénients
 - Ne marche pas avec l'émulateur (l'émulateur ne peut pas installer d'applications)

Méthode 1

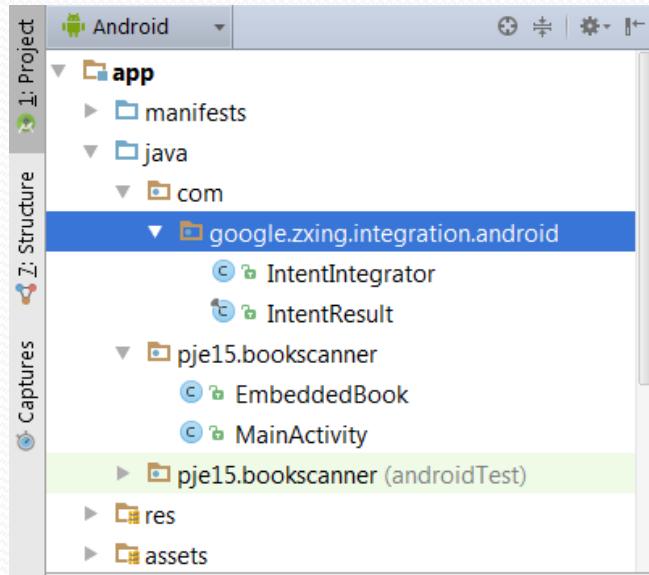
- Voir tutoriel book scanning
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app-interface-book-search--mobile-17790>

Méthode 1

- Télécharger les classes :
 - <http://code.google.com/p/zxing/source/browse/trunk/android-integration/src/com/google/zxing/integration/android/IntentIntegrator.java>
 - <http://code.google.com/p/zxing/source/browse/trunk/android-integration/src/com/google/zxing/integration/android/IntentResult.java>
- Les mettre dans votre projet
 - Attention au nom du package !

Méthode 1

- Dans votre Activity
 - Importer les classes
 - Lancer le scan



```
import com.google.zxing.integration.android.IntentIntegrator;
import com.google.zxing.integration.android.IntentResult;
```

```
public void onClick(View v) {
    //check for scan button
    if(v.getId()==R.id.scan_button){
        //instantiate ZXing integration class
        IntentIntegrator scanIntegrator = new IntentIntegrator(this);
        //start scanning
        scanIntegrator.initiateScan();
    }
}
```

Méthode 1

- Dans votre Activity
 - Récupérer le résultat du scan (String)

```
public void onActivityResult(int requestCode, int resultCode, Intent intent) {  
    //retrieve result of scanning - instantiate ZXing object  
    IntentResult scanningResult  
        = IntentIntegrator.parseActivityResult(requestCode, resultCode, intent);  
    //check we have a valid result  
    if (scanningResult != null) {  
        //get content from Intent Result  
        String scanContent = scanningResult.getContents();  
        //get format name of data scanned  
        String scanFormat = scanningResult.getFormatName();  
        Log.v("SCAN", "content: "+scanContent+" - format: "+scanFormat);  
        if(scanContent!=null && scanFormat!=null && scanFormat.equalsIgnoreCase("EAN_13")) {  
            previewBtn.setTag(scanContent);  
            String bookSearchString = "https://www.googleapis.com/books/v1/volumes?" +  
                "q=isbn:"+scanContent+"&key=YOUR_KEY";  
            //fetch search results  
            //new GetBookInfo().execute(bookSearchString);  
            doGoogleBookRequest(bookSearchString);  
        }  
    }  
}
```

Méthode 2

Embarquer zxing

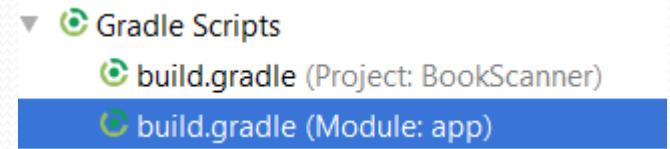
- <http://stackoverflow.com/a/23788163>
- Peut fonctionner avec l'émulateur
 - Si il supporte la caméra ...

Méthode 2

Embarquer zxing

- Modifier gradle (Module:app)
 - Pour lui demander de télécharger la lib zxing embedded
 - Ajouter les lignes suivantes :

```
repositories {  
    jcenter()  
}  
  
dependencies {  
    . . .  
    // ZXing  
    // Supports Android 4.0.3 and later (API level 15)  
    compile 'com.journeyapps:zxing-android-embedded:2.0.1@aar'  
  
    // Convenience library to launch the scanning and encoding Activities.  
    // It automatically picks the best scanning library from the above two, depending on the  
    // Android version and what is available.  
    compile 'com.journeyapps:zxing-android-integration:2.0.1@aar'  
  
    // Version 3.0.x of zxing core contains some code that is not compatible on Android 2.2 and earlier.  
    // This mostly affects encoding, but you should test if you plan to support these versions.  
    // Older versions e.g. 2.2 may also work if you need support for older Android versions.  
    compile 'com.google.zxing:core:3.2.1'  
}
```



Méthode 2

Embarquer zxing

- Spécifier les imports

```
import com.google.zxing.integration.android.IntentIntegrator;
import com.google.zxing.integration.android.IntentResult;
```

- Prendre un scan (idem méthode 1) :

```
public void onClick(View v) {
    //check for scan button
    if(v.getId()==R.id.scan_button) {
        //instantiate ZXing integration class
        IntentIntegrator scanIntegrator = new IntentIntegrator(this);
        //start scanning
        scanIntegrator.initiateScan();
    }
}
```

**CU : Ajouter un livre en
scannant son code barre**

Cas d'Utilisation

- Nom: Ajouter un livre en scannant son code barre
- L'utilisateur décide d'ajouter un livre. Il sélectionne « Ajouter un livre en scannant son code barre ». Le système ouvre l'application pour scanner le code barre. L'utilisateur scanne le code barre du livre. Le système recherche les métadonnées correspondant au code barre. Le système propose un formulaire pré-remplie (avec les métadonnées). L'utilisateur vérifie les données et les modifie si il le désire. L'utilisateur valide le nouveau livre. Le système ajoute le livre à la collection.

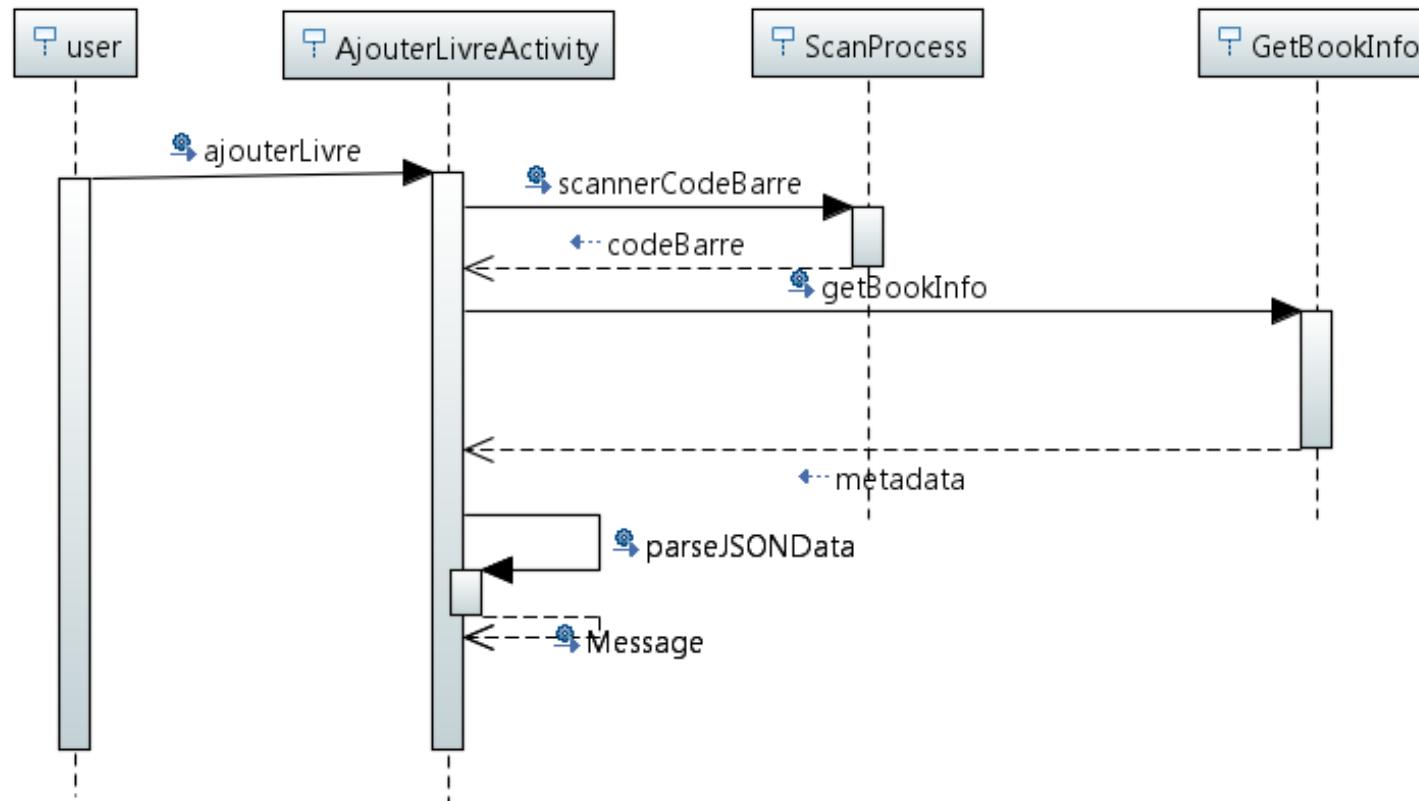
Variante

- L'utilisateur peut entrer le code barre manuellement, sans passer par l'application de scan.
- Le choix entre une saisie manuelle ou par scan peut se faire dans le même formulaire/la même activité.
- Utile pour les essais si l'application de scan ne peut pas fonctionner (problème avec l'émulateur)

Principe

- Appeler l'application de scan pour avoir un code barre
 - Ou saisir le code barre manuellement
- Rechercher les métadonnées sur le réseau
 - Google book, amazon ...
- Parser les métadonnées pour extraire l'info
- Afficher le formulaire d'ajout de livre
 - Le préremplir avec les données trouvées

Principe



Exemple

- **Android SDK: Create a Book Scanning App**
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app--mobile-17634>
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app-interface-book-search--mobile-17790>
 - <http://code.tutsplus.com/tutorials/android-sdk-create-a-book-scanning-app-part-3--mobile-17880>
- Permet :
 - De saisir un code barre (zxing)
 - De rechercher sur google book
 - De parser les données
 - D'afficher les données (y compris l'image)

Scan Book

- Problèmes
 - Version de apache.http plus supporté.
 - <http://stackoverflow.com/questions/30556605/org-apache-http-httppentity-is-deprecated-how-to-solve-this-error-in-android-s>
 - Use cz...
 - Demander une cle pour l'API book
 - Le scan et la connexion internet ne fonctionnent pas sur l'emulateur

Remplacer la lib http

Solution 1

- Utiliser une version de remplacement :
 - **cz.msebera.android:httpclient**
- Dans gradle.build (app:)

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:appcompat-v7:23.0.1'  
    compile "cz.msebera.android:httpclient:4.4.1.1"  
}
```

- Dans vos import
 - Remplacer org.apache.http... par
cz.msebera.android:http...
- Relancer la compilation

Remplacer la lib http

Solution 2 - Volley

- Utiliser Volley
 - Librairie de google
- **Voir : Transmitting Network Data Using Volley**
 - <http://developer.android.com/training/volley/index.html>
- La leçon **Sending a Simple Request**
 - Explique comment envoyer une requête http, et recevoir la réponse
- Vous pouvez utiliser cette solution dans l'exemple de Scan Book

Scan Book avec Volley

```
private void doGoogleBookRequest( String url) {
    // Instantiate the RequestQueue.
    RequestQueue queue = Volley.newRequestQueue(this);

    // Request a string response from the provided URL.
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                // Display the first 500 characters of the response string.
                parseJsonResult(response);
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                titleText.setText("NOT FOUND (response error)");
                authorText.setText("");
                descriptionText.setText("");
                dateText.setText("");
                starLayout.removeAllViews();
                ratingCountText.setText("");
                thumbView.setImageBitmap(null);
                previewBtn.setVisibility(View.GONE);
            }
        });
    // Add the request to the RequestQueue.
    queue.add(stringRequest);
}
```

Utiliser la tablette Archos Oxygen 70

Tablette Archos 70 Oxygen

- Elle arbore un écran de 7 pouces avec une résolution Full HD de 1920 x 1080 pixels, ce qui lui permet d'obtenir une densité de 314 points par pouce.
- Une puce de chez MediaTek
- À l'intérieur, nous avons un processeur de chez MediaTek le MT8163A composé de 4 coeurs (Cortex A53) cadencés à 1,3 GHz couplé à 2 gigaoctets de mémoire vive (RAM) et d'une puce graphique Mali 720 MP2.
- Pour ce qui est de l'espace de stockage, l'Archos 70 Oxygen dispose de 32 gigaoctets qu'il faudra se partager avec la dernière version d'Android 6.0 Marshmallow. Cependant, il est possible d'étendre cette mémoire interne grâce à une carte microSD puisque la tablette dispose d'un emplacement prévu à cet effet.
- Il existe aussi une partie photo sur cette 70 Oxygen, mais il ne faudra pas trop compter dessus. À l'arrière, nous avons un capteur de 5 mégapixels avec flash à LED et autofocus tandis qu'à l'avant un capteur de 2 mégapixels.
- La connectivité est assurée pour le Wi-Fi (802.11 b/g/n), le Bluetooth 4.0 et une puce GPS.
- La batterie de 3000 mAh devrait proposer une autonomie tout à fait correcte pour ce genre de produit surtout qu'il est destiné à une utilisation multimédia.

Lancer une appli sur tablette

- Connecter la tablette par le câble USB
- Sur la tablette :
 - Activer « Options pour les développeurs »
 - Paramètres -> A propos de la tablette
 - Taper 7 fois sur « Numéro de build »
 - Autoriser le débogage USB :
 - Paramètres -> Options pour les développeurs -> Débogage USB
- Dans Android Studio
 - Run app
 - Choisir la tablette
- <https://developer.android.com/training/basics/firstapp/runnning-app.html>