

Appel d'autres applications

Titouan Roos (FA)

26/10/2017

Résumé

Lors de nos développements Android, il est parfois utile (ou indispensable dans certains cas) d'utiliser d'autres applications que celle que nous utilisons.

Dans ce TP, nous verrons comment appeler une autre application, et comment récupérer les données de celle-ci dans notre application à nous. Plus particulièrement, nous allons créer une application qui lis un QRCode, récupère les données qu'il contient et ensuite nous les affiche sur une activité.

Pré-requis

- *Savoir programmer une application Android avec plusieurs activités avec envoi de données.*
- *Savoir organiser son interface avec xml.*

Code source

Code source **initial** disponible à https://github.com/roosti/TP_Android_QRCode_initial

Code source **final** disponible à https://github.com/roosti/TP_Android_QRCode_final

Explications du TP

Etape 1: Création d'un QRCode personnel

Nous allons tout d'abord commencer par créer nous même notre propre QRCode.

Pour ce faire, cliquez sur cet URL : <http://goqr.me/>

Choisissez le type 'text'. Puis, pour que notre future application fonctionne correctement, je vous donne le squelette du texte à remplir dans la zone attendue pour générer notre QRCode :

```
{"nom":"<votre nom>","prenom":"<votre prenom>","age":"<votre age>","adresse":"<votre adresse>"}
```

1. Type

text

2. Contents

Text

```
{ "nom": "<votre nom>", "prenom": "<votre prenom>", "age": "<votre age>", "adresse": "<votre adresse>" }
```

95 characters

Hint: The shorter, the better. Some older scanner apps are having problems with texts longer than about 300 chars.

Your QR code data is encrypted during transmission (TLS/SSL) and not stored.

Like

+1

Tweet

i

3. Live preview

Add a logo!

Download

Embed

Cliquez ensuite sur Dowload et choisissez le format PNG.

Vous voilà donc avec votre propre QRCode. Maintenant, il ne reste plus qu'à créer l'application pour lire les informations.

Etape 2 : Récupération du projet initial sur GitHub

Pour bien démarrer ce TP, j'ai créé une version initiale où vous n'aurez qu'à décommenter certaines parties, pour ne pas perdre de temps à tout réécrire.

Vous la trouverez à cette URL : https://github.com/roosti/TP_Android_QRCode_initial

Etape 3 : Ajout de la librairie Zxing

Tout d'abord, afin de pouvoir utiliser une application qui lit les QRCode, il faut ajouter la librairie Zxing qui contient les outils nécessaires pour le faire.

Pour cela, il faut aller ajouter la ligne suivante dans les dépendances du projet, qui se trouvent dans le fichier build.gradle au niveau de notre 'app'.

```
/** ----- Ajout de la librairie ZXing ----- */
compile 'com.journeyapps:zxing-android-embedded:3.4.0'
```

N'oubliez pas de synchroniser votre projet à nouveau pour qu'elle soit prise en compte.

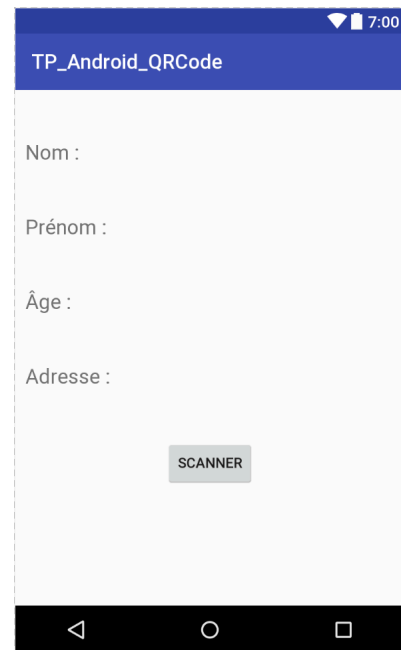
Etape 4 : Création de l'interface utilisateur

Pour l'interface utilisateur, nous allons faire quelque chose de simple.

Nous allons juste créer les champs nécessaires pour recueillir les informations que nous récupérerons dans le QRCode. C'est-à-dire un nom, prénom, âge et adresse.

Ensuite, ajouter un bouton qui appellera le scanner de QRCode.

Vous pouvez observer ci-contre un aperçu attendu.



Etape 5 : Gestion du QRCode scanner

Commencez par importer dans votre classe Java, les textView qui vont recevoir les informations, ainsi que le bouton qui va lancer le scanner.

Ensuite, il faut ajouter un événement sur votre bouton pour qu'il réagisse au click.

Créez ensuite un objet `IntentIntegrator`, qui va vous permettre de lancer le scanner. Puis, dans l'événement du bouton, lancer la fonction `initiateScan()` grâce au nouvel objet créé. Cette fonction a pour effet de lancer un `startActivityForResult`.

Étant donné que nous allons récupérer des données après le lancement du scanner de QRCode, il faut override la fonction `onActivityResult`. Dans cette fonction, vous allez récupérer l'intent renvoyé grâce à la fonction `parseActivityResult` présente dans la classe `IntentIntegrator`. Cette fonction va permettre de transformer la donnée en un `IntentResult`. Et c'est dans ce `IntentResult` que se trouve toutes nos informations.

Vérifiez que le intent n'est pas nul. Récupérez le contenu via un `JSONObject`. Puis fixer les valeurs récupérer aux textView associés.

Etape 6 : Testez l'application

Il ne vous reste plus qu'à tester l'application. Vous pouvez ouvrir votre QRCode sur votre ordinateur, et le scanner directement, cela fonctionne.