

ESUP-NFC-TAG-DROID

Esup-nfc-tag-droid permet de lire (et potentiellement encoder) les cartes Mifare Desfire.

Le client s'appuie sur la plateforme <https://github.com/EsupPortail/esup-nfc-tag-server> qui calcule les commandes (APDU) à transmettre à la carte.

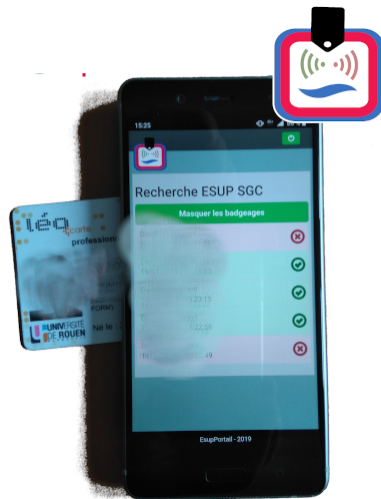
Esup-nfc-tag-droid permet d'utiliser un smartphone Android pour badger, en utilisant l'UID (CSN) ou en faisant une lecture d'un fichier Desfire (avec authentification AES)

Une APK ESUP officielle et générique d'esup-nfc-tag-droid est diffusée depuis le Google Play ESUP : <https://play.google.com/store/apps/details?id=org.esupportail.esupnfctagdroid>

esup-nfc-tagdroid est également disponible depuis F-Droid : <https://f-droid.org/packages/org.esupportail.esupnfctagdroid/>

L'application peut aussi être packagée sous la forme d'un apk que l'on peut alors installer sur un smartphone Android en autorisant les applications de sources inconnues.

L'application peut être installée et debuggée depuis Android-Studio ou compilée directement à l'aide de Gradle



ESUP-NFC-TAG-DROID

<https://esup-portail.org>

- [Fonctionnalités](#)
- [Environnement](#)
 - [Logiciel](#)
 - [Matériel](#)
- [APK officielle ESUP sur Google Play, F-Droid et Github](#)
- [Sources](#)
 - <https://github.com/EsupPortail/esup-nfc-tag-droid>
- [Génération de l'APK](#)
 - [Génération de l'APK en ligne de commandes](#)
 - [Pré-requis](#)
 - [Compilation esup-nfc-tag-droid](#)
 - [Integration dans esup-nfc-tag-server](#)

Fonctionnalités

- 1 - L'application esup-nfc-tag-droid se comporte de la même manière que l'application Java [esup-nfc-tag-desktop] (<https://github.com/EsupPortail/esup-nfc-tag-desktop> "esup-nfc-tag-desktop")
- 2 - L'application repose sur un composant webview qui se connecte et affiche la vue fournie par esup-nfc-tag-server
- 3 - Après l'authentification Shibboleth il faut choisir la salle de badgeage
- 4 - Pour badger il suffit de poser une carte sur le lecteur nfc (à l'arrière du smartphone)

Environnement

Logiciel

L'application est prévue pour tourner sous Android 5 minimum

Materiel

Un smartphone Android (version 5 ou supérieur) équipé d'un lecteur NFC et disposant d'un accès Internet

APK officielle ESUP sur Google Play, F-Droid et Github

Vous pouvez utiliser directement l'APK officielle et générique diffusée depuis Google Play par ESUP :

<https://play.google.com/store/apps/details?id=org.esupportail.esupnfctagdroid>

L'APK envoyée sur [Google Play](#) le 9 nov. 2023 est la version 1.2.6 disponible également depuis le [Tag github 1.2.6](#). Cette version est également disponible depuis [F-Droid](#).

Si vous êtes responsable d'un serveur esup-nfc-tag d'un établissement de l'ESR que vous souhaitez voir apparaître votre serveur esup-nfc-tag dans la liste des serveurs disponibles depuis cette version esup-nfc-tag-droid, vous pouvez en faire la demande au travers d'un Pull Request proposant la modification du listing des urls disponibles :

<https://github.com/EsupPortail/esup-nfc-tag-droid/blob/master/src/main/assets/urls>

Ce fichier est récupéré dynamiquement par esup-nfc-tag-droid à chaque démarrage, aussi dès que votre Pull Request est accepté, votre serveur esup-nfc-tag-server est disponible depuis un esup-nfc-tag-droid.

Sources

<https://github.com/EsupPortail/esup-nfc-tag-droid>

```
git clone https://github.com/EsupPortail/esup-nfc-tag-droid.git
```

Génération de l'APK

Si vous souhaitez proposer votre propre ESUP-NFC-TAG-DROID pointant directement sur votre serveur esup-nfc-tag, il vous faut construire votre propre APK.

Vous pouvez le faire de 3 manières :

1. en utilisant Android Studio (<https://developer.android.com/studio>), au 13/10/2023, Studio Giraffe a été testée avec succès
2. en ligne de commande depuis un linux

Génération de l'APK en ligne de commandes

Pré-requis

- OpenJDK 11 (par exemple une distribution zulu d'openjdk 11 : <https://www.azul.com/downloads/?version=java-11-lts&package=jdk#zulu>)

Compilation esup-nfc-tag-droid

- esup-nfc-tag-droid génère des logs à destination d'un fichier de logs local au téléphone, à destination d'esupNfcTagServer (envoi de logs par POST au serveur) et à destination d'une adresse mail système. Les éléments paramétrables (mail systeme, serveur esupNfcTagServer) sont à configurer dans ce fichier `src/main/assets/logback.xml`
- modifier `src/main/assets/esupnfctag.properties` pour spécifier l'adresse de votre esup-nfc-tag-server
- Vous pouvez spécifier les paramètres de signature de votre APK dans `build.gradle`, si vous ne souhaitez pas utiliser ceux donnés par défaut (connus de tous). Vous devrez alors créer un keystore :

```
keytool -genkey -v -keystore esup-android-apps.keystore -alias LeoDroidApp -keyalg RSA -keysize 2048 -validity 10000
```

- build de l'APK depuis le répertoire source :

```
chmod u+x gradlew
./gradlew clean assemble
```

Integration dans esup-nfc-tag-server

- copier l'APK dans EsupNfcTagServer pour le mettre à disposition des utilisateurs :

```
cp ./build/outputs/apk/release/esup-nfc-tag-droid-release.apk /<path to>/esup-nfc-tag-server/src/main/resources/apk/esupnfc-tag-droid.apk
```

- recompiler et redéployer esup-nfc-tag-server. Au redémarrage d'esup-nfc-tag-server la nouvelle version de l'apk sera prise en compte
- Vous pouvez aussi proposer le lien vers l'APK via esup-sgc (en ajoutant un "navbarapp" depuis l'onglet admin/navbarapp)