

Édition en 2 passes

Fonctionnalités

1. L'application lit le QR code imprimé sur la carte à encoder qui correspond à l'identifiant de la carte.
2. L'application récupère et envoie au lecteur NFC les commandes à exécuter (APDU) sur la carte via esup-nfc-tag-server
3. Validation de l'encodage et activation de la carte sur esup-sgc
4. Éventuellement encodage de l'application CROUS (voir <https://www.esup-portail.org/wiki/display/SGC/FAQ#FAQ-Peut-onencoderl'applicationCROUSquandonutiliselescartervierges?>)

Environnement

Logiciel

- L'application fonctionne sous java avec JavaFX.
- L'application fonctionne sur Linux ou Windows 10 64bits (l'encodage CROUS nécessite Windows 10 64bits)
- pour l'encodage de l'application CROUS le pilote OMNIKEY CardMan 6121 ([pilote](#)) (voir [ESUP-CNOUS-CLIENT](#))

Matériel

L'application nécessite :

- une webcam gérant la résolution VGA (640x480)
- un lecteur de carte compatible PC/SC
- pour l'encodage de l'application CROUS il faut connecter la clé SAM OMNIKEY CardMan 6121 avec sa carte sim

La webcam est placée pour filmer le lecteur de carte (procéder à la mise au point si besoin). Lorsqu'une carte est posée sur le lecteur de carte, la webcam détecte le QR code et la procédure d'encodage démarre

[Documentation de mise en œuvre ESUP-SGC / ESUP-NFC-TAG#SGC/ESUP-NFC-TAG-Installationmaterielle](#)

Installation

Le plus rapide est de récupérer le jar esup-sgc-client et/ou l'installateur windows depuis <https://esup-sgc-client-web-installer.univ-rouen.fr/>

Ce site permettant de générer l'installateur et ces jar s'appuie sur esup-sgc-client-installer dont le code est également librement disponible depuis le github ESUP : <https://github.com/EsupPortail/esup-sgc-client-installer>

Vous pouvez packager/builder esup-sgc-client vous même également cependant.

Sources :

<https://github.com/EsupPortail/esup-sgc-client.git>

Merci de vous référer au fichier README pour toute information complémentaire concernant la compilation et mise en œuvre.

```
git clone https://github.com/EsupPortail/esup-sgc-client.git
```

Lancement

Si vous utilisez l'installateur windows, vous n'avez qu'à lancer le client depuis le raccourci créé par celui-ci, l'installateur embarque directement [une distribution d'un openjdk et openjfx proposés par la communauté Zulu](#) !

Vous pouvez aussi utiliser openJDK fourni depuis <https://jdk.java.net> et openJFX fourni depuis <https://gluonhq.com/products/javafx/> (voir également <https://openjfx.io/openjfx-docs/>) qui sont des produits open source et gratuits également.

esup-sgc-client est compatible avec les JDK 8 et supérieurs. Depuis février 2023, l'installateur windows embarque les versions JDK/JFX 17 packagées par Zulu (les dernières en date).

Suivant l'intégration de JAVA FX, la commande à lancer est la suivante :

```
java.exe -module-path %PATH_TO_FX% --add-modules javafx.controls,javafx.fxml,javafx.base,javafx.media,javafx.graphics,javafx.swing,javafx.web -Dcom.sun.webkit.useHTTP2Loader=false -jar esup-sgc-client-final.jar
```

L'option "-Dcom.sun.webkit.useHTTP2Loader=false" donnée ici à titre indicatif permet de forcer l'usage de HTTP 1 et peut permettre d'éviter d'éventuels problèmes avec des serveurs web (apache) un peu anciens.

Précisions

Les installations d'openJDK et OpenJFX correspondent à dézipper leurs archives dans un répertoire donné.

Sur un poste windows, on peut par exemple les dézipper dans un répertoire C:\esup-sgc-client, on aura ainsi

- C:\esup-sgc-client\jdk-12.0.1 pour le JDK
- C:\esup-sgc-client\javafx-sdk-12.0.1 pour le JavaFX

En plaçant le JAR d'esup-sg-client (esupsgcclient-shib.jar) dans ce même répertoire C:\esup-sgc-client on peut ainsi finalement exécuter le JAR ainsi :

```
C:\esup-sgc-client\jdk-12.0.1\bin\java.exe --module-path C:\esup-sgc-client\javafx-sdk-12.0.1\lib --add-modules javafx.controls,javafx.fxml,javafx.base,javafx.media,javafx.graphics,javafx.swing,javafx.web -jar C:\esup-sgc-client\esup-sg-client-final.jar
```

A l'exécution esup-sgc-client écrit 2 fichiers dans le répertoire utilisateur (ou sous-répertoire) :

1. un fichier de logs pour éventuellement tracer les actions/infos/erreurs
2. un fichier contenant un jeton d'authentification permettant à l'utilisateur d'éviter de devoir resssisir ses identifiants à chaque lancement de l'application.

Vidéos de démonstration

Une vidéo montrant cette édition de cartes en 2 passes est disponible depuis le Pod d'ESUP-Portail :

