ESUP-NFC-TAG-KEYBOARD

L'application esup-ngc-tag-keyboard permet d'émuler des frappes clavier (permettant ainsi de saisir l'identifiant du porteur de carte par exemple) ou de faire une redirection en fonction des données récupérées via esup-nfc-tag-server

L'application lit le csn ou l'application Desfire d'une carte et le transmet a esup-nfc-tag-server qui répond en fonction de l'identifiant de périphérique choisi.

Environnement

Pré-requis

- Java 11 ou supérieur
- Maven

Logiciel

- L'application est prévue pour tourner avec openjdk 11 ou supérieur

Materiel

- un lecteur de carte USB compatible PC/SC (ex: Indentive Cloud 4700f, OMNIKEY CardMan 5x21-CL...)

Installateur/jar esup-nfc-tag-keyboard

Cf ci-dessous, vous pouvez récupérer les sources d'esup-nfc-tag-keyboard depuis le github EsupPortail et packager l'application vous même.

Vous pouvez aussi passer par https://esup-sgc-client-web-installer.univ-rouen.fr pour récupérer un installateur windows (ainsi que les jar intermédiaires qui peuvent être utilisés sur linux par exemple).

L'installateur vous créera un raccourci sur esup-nfc-tag-keyboard demandant à un openjdk/openjfx embarqué par l'installateur de lancer votre esup-nfc-tag-keyboard.

Sources : https://github.com/EsupPortail/esup-nfc-tag-keyboard

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

Paramétrage

L'application propose plusieurs options :

- Switcher entre plusieurs applications (~ champs à récupérer depuis esup-nfc-tag après badgegage pour saisi par émulation clavier) : **numeroIds** peut contenir plusieurs valeurs séparées par des virgules.
  - **numeroIds** permet une émulation clavier pour que soit saisi un identifiant de carte, d'utilisateur ou un email, ces **numeroIds** sont à définir dans le paramétrage à faire dans esup-nfc-tag-server (cf ci-dessous le paramétrage côté esup-nfc-tag-server) ; ces identifiants étant en effet récupérés par esup-nfc-tag
- **forceCsn** à true permet de proposer à l'utilisateur une saisie émulée directe du csn (ou csn renversé ; swap-pair) sans passer par esup-nfc-tag
- Avec **emulateKeyboard** à true, l'émulation clavier est effective, dans le cadre de l'usage de redirect à true (cf ci-dessous), ce paramètre peut être mis à false.
- Redirection dans un navigateur web vers une url construite avec l'identifiant retourné : mettre **redirect** = true et renseigner **redirectUrlTemplate**

- Possibilité de saisie d'un retour chariot (touche Entrée) après la saisie de l'identifiant (cas d'usage : 'valider' un formulaire ou passer à la ligne dans un fichier texte) : **lineFeed**
- Possibilté d'ajouter un préfixe et/ou un suffixe à ce qui est saisi : **prefix** et **suffix**
- **port** définit le numéro de port local que l'application va utiliser pour 'écouter' : c'est ce système qui permet d'éviter que l'application puisse se lancer 2 fois sur une même machine.
- **esupNfcTagServerUrl** : url du serveur esup-nfc-tag-server
- noResponseMessage : message à afficher si la carte n'est pas trouvée/valide pour le serveur esup-nfc-tag
- timeBetweenSameCard, cardReadSleepTime, onErrorSleepTime, beforeNextCardSleepTime : paramètres permettant d'indiquer le nombre de millisecondes pendant lesquels l'application se "met en pause"
  plus on baisse ces paramètres et plus l'application pourra être réactive lors du badgeage
  ces "pauses" permettent cependant d'éviter une consommation excessive de cpu, permettent de laisser du temps d'exécution au contrôle de la partie graphique dans la barre de tâche, d'éviter de trop solliciter le lecteur NFC ...

Il est possible de modifier le fichier src/main/resources/esupnfctagkeyboard.properties pour spécifier les paramètres par défaut :

```
port = 33333
esupNfcTagServerUrl = https://esup-nfc-tag.univ-ville.fr
noResponseMessage = -
numeroIds = keyboard-secondary-id, keyboard-csn, keyboard-eppn, keyboard-email
emulateKeyboard = true
forceCsn = false
lineFeed = true
redirect = false
redirectUrlTemplate = https://esup-sgc.univ-ville.fr/manager/{0}
prefix =
suffix  =
timeBetweenSameCard = 3000
cardReadSleepTime = 1000
onErrorSleepTime = 3000
beforeNextCardSleepTime = 1500
```

Compilation esup-nfc-tag-keyboard

Modifier le fichier esupnfctagkeyboard.properties pour y insérer l'adresse de l'instance esup-nfc-tag-server

Dans de répertoire des sources lancer la commande :

```
mvn clean package
```

le jar esupsgckeybemu-1.0-SNAPSHOT-jar-with-dependencies.jar va être générer dans le dossier target

Paramétrage esup-nfc-tag

Voici un exemple de configuration pour une utilisation en Bibliothèque universitaire

Il faut tout d'abord ajouter une configuration dans applicationContext-custom.xml. (exemple se basant sur ESUP-SGC)

```
    <bean id="esupSecondaryIdExtApi" class="org.esupportail.nfctag.service.api.impl.AppliExtRestWs">
        <property name="isTagableUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/isTagable"/>
        <property name="validateTagUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/validateTag"/>
        <property name="getLocationsUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/locationsSecondaryId"
/>
        <property name="displayUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/secondaryId?
idName=secondaryId"/>
        <property name="description" value="Web Service SecondaryId"/>
    </bean>
```

Créer une application dans esup-nfc-tag-server avec les paramètres suivants :

- Nom : Biblio
- Configuration NFC : Authentification CSN
- Application externe : Web Service SecondaryId
- Controle du tagId : via Esup SGC

Créer un périphérique dans esup-nfc-tag-server avec les paramètres suivants:

- Numero Id : keyboard-secondary-id
- Eppn init : un eppn ex : esup@univ-ville.fr
- Adresse MAC : 0
- IMEI : esup-nfc-tag-keyboard
- User Agent : esup-nfc-tag-keyboard
- Application : Biblio
- Salle : retourné par le ws https://esup-sgc.univ-ville.fr/wsrest/nfc/locationsSecondaryId
- Validation sans confirmation : true

Lancement de l'application

Il est possible de déclarer les propriétés systèmes suivantes, au lancement du jar, pour surcharger les propriétés définies dans esupnfctagkeyboard.properties (afin d'évité de recompiler l'application):

```
-DesupNfcTagKeyboard.esupNfcTagServerUrl=https://esup-nfc-tag.univ-ville.fr
-DesupNfcTagKeyboard.noResponseMessage="Carte non active ou non valide"
-DesupNfcTagKeyboard.numeroIds=<liste des numememeroIds séparés par des virgules>
-DesupNfcTagKeyboard.emulateKeyboard=true
-DesupNfcTagKeyboard.forceCsn=false
-DesupNfcTagKeyboard.lineFeed=true
-DesupNfcTagKeyboard.redirect=false
-DesupNfcTagKeyboard.redirectUrlTemplate = https://esup-sgc.univ-rouen.fr/manager/{0}
-DesupNfcTagKeyboard.prefix=
-DesupNfcTagKeyboard.suffix=
-DesupNfcTagKeyboard.timeBetweenSameCard = 500
-DesupNfcTagKeyboard.cardReadSleepTime = 200
-DesupNfcTagKeyboard.onErrorSleepTime = 500
-DesupNfcTagKeyboard.beforeNextCardSleepTime = 200
```

pour lancer l'application:

```
java -jar esupnfctagkeyboard-1.0-SNAPSHOT-jar-with-dependencies.jar
```

Sous windows, le plus simple est de créer un fichier bat comme suit (ici un exemple avec l'ajouter un prefix) :

```
@ECHO OFF
start javaw -DesupNfcTagKeyboard.prefix=TEST -jar c:\<path_to_jar>\esupsgckeybemu-1.0-SNAPSHOT-jar-with-dependencies.jar
exit
```

et de lancer le fichier bat, l'application va alors tourner en tache de fond

Il est possible de créer plusieurs applications en ajoutant d'autres AppliExtRestWs dans le fichier applicationContext-custom.xml puis en créant les nouvelles applications et les périphériques qui correspondent dans esup-nfc-tag-server.

L'application permet de switcher d'une application à une autre via le "tray icon"

Par exemple pour une émulation de l'eppn :

```
<bean id="esupEppnExtApi" class="org.esupportail.nfctag.service.api.impl.AppliExtRestWs">
        <property name="isTagableUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/isTagable"/>
    <property name="validateTagUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/validateTag"/>
    <property name="getLocationsUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/locationsSecondaryId"/>
    <property name="displayUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/secondaryId?
idName=eppn"/>
    <property name="description" value="Web Service SecondaryId"/>
</bean>
```

Usage

Pour contôler que l'application est bien lancée, un "tray icon" doit apparaitre dans la barre des taches. Celui-ci représente le logo esupnfctag et permet :

- de contôler l'état de l'application (le logo barré d'une croix rouge représente une erreur à consulter dans les log)
- switcher d'une application à une autre (entre l'eppn et le secondaryId par exemple)
- quitter l'application

Une fois l'application lancée, lorsque qu'une carte est posée, le retour du web service est émulé sur le clavier. Si l'application a été configurée avec "redirect = true", le navigateur ouvre le lien définit par redirectUrlTempalte
Dans le but d'évité une mauvaise manipulations, une même carte ne peut pas être passée une deuxieme fois pendant un délai de 3 secondes.

Démarrage automatique pour tous les utilisateurs du poste

copier le fichier esupnfctagkeyboard-1.0-SNAPSHOT-jar-with-dependencies.jar dans le dossier c:\<path_to_jar>

copier le .bat dans C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp

Lorsque qu'un utilisateur ouvre une session le .bat est executé et l'application se lance en tache de fond