Outils pour l'intégration et/ou le développement

Cette page présente l'usage des outils de développement dans le cadre des projets ESUP-SGC / ESUP-NFC-TAG. Ces outils peuvent être utilisés au moment de l'intégration de la solution ainsi que pour le développement de nouvelles interfaces.

Lors de l'intégration du projet il est intéressant d'importer les sources dans un outils de développement pour bénéficier de ses fonctionnalités (autocomplétion, coloration syntaxique, indentation..). Par ailleurs cela peut permettre d'implémenter de nouveaux services ex :

- Esup-SGC : CardIdsServices (gestion d'identifiants), ExtUserInfoService (Interfaces d'import des données utilisateurs), Export2AccessControlService (Exports pour les contrôles d'accès)
- Esup-Nfc-Tag : TagldCheckApi (contrôle des identifiants de cartes), AppliExtApi (Interfaces)
- SpringToolSuite
 - ^o Pré-requis
 - Importer le projet
 - Editer la configuration
 - Versionning de la configuration
 - Compilation / lancement
- Android Studio
 - Pré-requis
 - Importer le projet
 - Editer la configuration
 - Compilation
 - Pré-requis
 - Importer le projet
 - Compilation

SpringToolSuite

Esup-SGC et Esup-Nfc-Tag-Server sont développés à l'aide de SpringToolSuite STS (basé sur Eclipse) : https://spring.io/tools

Pré-requis

- Java JDK 8
- Maven

Importer le projet

Après avoir cloné le projet (dans /opt/esup-sgc), démarrez STS et créez un nouveau workspace dans /opt/workpaces/esup-sgc par exemple

	Workspace Launcher X
Select a wo	rkspace
Spring Tool S Choose a wo	Suite stores your projects in a folder called a workspace. rkspace folder to use for this session.
Workspace:	/opt/workspaces/esup-sgc
Use this	as the default and do not ask again
	Cancel

Lorsque STS est démarré faites "File -> Import" puis choisissez "Maven -> Existing Maven Projects" et naviguez dans le dossier des sources où se trouve le pom.xml

Cliquez sur "Finish", le projet s'importe puis vous devez obtenir l'arborescence suivante :



Editer la configuration

Dans STS l'auto-complétion est déclenchée par la combinaison CTRL + Espace

L'auto-complétion va nous permettre de modifier facilement la configuration. On prend comme exemple le ficher src/main/ressources/META-INF/spring /applcationContext-services.xml. Si l'on souhaite ajouter un service CardIdsService

A la saisie dans le nouveau bean au niveau du '<bean class="org.esupportail.sgc.services.cardid.' l'auto-complétion va proposer CnousCardIdService, ComueNuAccessControlCardIdService, GenericCardIdService ...

De même, pour la saisie des propriétés l'auto-complétion fonctionne et propose les noms des attributs de la classe sélectionnée.

Cette méthode permet de sécurisé l'édition des fichiers de configuration.

Versionning de la configuration

D'une manière générale il est conseillé de faite des commits GIT locaux pour sauvegarder vos modifications de configuration.

Lorsque vous avez fini la configuration du fichier vous pouvez exécuter les commandes suivantes depuis la racine de vos sources

```
git add src/main/resources/META-INF/spring/applicationContext-service.xml
git commit -m "config prod univ-ville-fr"
```

La modification du fichier applicationContext-service.xml sera versionnée ce qui permet de conserver les traces de toutes les modifications (procédure à faire à chaque modification). De plus lors d'une mise à jour d'ESUP-SGC (git pull) la configuration ne sera pas écrasée.

Compilation / lancement

Pour compiler ou lancer le projet il faut créer une configuration de lancement de type "Maven Build" dans "Run -> Run Configurations" comme suit :

Run Configurations					
Create, manage, and run co	onfigurations	D			
Image: Specific state Image: Specific state Image: Specific state Image: Specific state Image: Specific state Image: Specific state	Name: tomcat Image: Main A JRE A Refresh & Source Environment Common Base directory: \${workspace_loc:/sgc}				
Aspect/java Application Eclipse Application Eclipse Data Tools Generic Server Generic Server(Externa HTTP Preview IZFE Preview	Workspace File System Variable Goals: klean tomcat7:run Profiles:	es			
 Java Applet Java Application Jetty Webapp Ju JUnit JUnit Plug-in Test Maven Build 	Workspace File System Variable Offline Update Snapshots Debug Output Skip Tests Non-recursive Resolve Workspace artifacts	es			
G name OSGi Framework Pivotal tc Server Spring Boot App Spring Boot Devtools C Jy Task Context Test	1 ▼ Threads Parameter Na Value Edi Rem	d t			
X XSL Filter matched 24 of 24 items	Maven Runtime: EMBEDDED (3.3.9/1.7.0.20160603-1931) Configu Revert Ap	ply			

Android Studio

Esup-Nfc-Tag-Droid est développé sous Android Studio https://developer.android.com/studio/index.html

Pré-requis

Java

- Smartphone Android
- SDK 5.1

Pour verifier l'installation de SDK, il faut ouvrir SDK Manager depuis le menu "Configure"

Default Settings X							
Q*	Appearance & Behavior > System Settings	Android SD	c				
Appearance & Behavior	Manager for the Android SDK and Tools used by Android Studio						
Menus and Toolbars	Android SDK Location: /opt/tools/Android/Sdk			Edit			
System Settings							
Passwords	SDK Platforms SDK Tools SDK Update Sites						
HTTP Proxy	Each Android SDK Platform package includes the	e Android platform	and sources p	pertaining to			
Undeter	an API level by default. Once installed, Android Studio will automatically check for updates.						
Opdates	check show package details to display individual		Lo.				
Usage Statistics	Name	APILEVEI	Revision	Status			
Android SDK		27	3	Partially installed			
Natifications	Android 8.1 (Oreo)	26	2	Partially installed			
Notifications	Android 7.1.1 (Nougat)	25	3	Partially installed			
Quick Lists	Android 7.0 (Nougat)	24	2	Partially installed			
Path Variables	Android 6.0 (Marshmallow)	23	3	Update available			
Kauman	🗹 Android 5.1 (Lollipop)	22	2	Installed			
кеутар	Android 5.0 (Lollipop)	21	2	Partially installed			
▶ Editor	Android 4.4W (KitKat Wear)	20	2	Not installed			
Plugins	Android 4 4 (KitKat)	19	4	Partially installed			
Build, Execution, Deployment				Show Package Details			
		C	Canc	el Apply Help			

Importer le projet

Après avoir cloné le projet (dans /opt/esup-nfc-tag-droid), choisir "Import project"



Ouvrir le fichier build.gradle et contrôler que ja synchro est bonne. Si cela n'est pas le cas cliquer sur "Sync Now" ou "Try Again" (le cas echéant fixer les problèmes Gradle)

Si la version de gradle n'es	t pas bonne l'ide propose de fixer le projet	
The versions of the Androic Please do one of the follow Open migration guide Fix Gradle wrapper and re-	d Gradle plugin and Gradle are not compatible. ring: Update your plugin to version 2.4. This will require changes to build :li>Downgrade Gradle to version 3.5. import project	.gradle due to API change
ou		
Upgrade build tools		
osque le project compile correcten	nent la barre d'outils présente le bouton d'execution	
🖌 📂 anum mfa tan du		
🔨 🔚 esup-nfc-tag-dr	oid 🚽 🕨 🧍 🎁 📫 🧖 🔓 🔳 📗	
A Carl esup-nfc-tag-dr application peut etre lancée soit s pit sur un téléphone Android avec	oid -) + * * 1 • • • [•] [ur un emulateur Android paramétré depuis l'écran "Select deployment Targe" (v degobage USB activé	voir dans "Create New Virt
A Carl Contraction peut etre lancée soit s oit sur un téléphone Android avec	oid A I A A A A A A A A A A A A A A A A A A	voir dans "Create New Virt
A Coppected Devices	oid A Select Deployment Target	voir dans "Create New Virt
Connected Devices	oid 	voir dans "Create New Virt
Connected Devices	oid A I A A A A A A A A A A A A A A A A A A	voir dans "Create New Virt
Connected Devices	oid A A A A A A A A A A A A A A A A A A A	voir dans "Create New Virt
Connected Devices LGE Nexus 5 API 23	oid A A	voir dans "Create New Virt
Connected Devices LGE Nexus 5X (And Available Virtual Device Nexus 5 API 23 testandroid5 (Missi	oid A A A A A A A A A A A A A A A A A A A	voir dans "Create New Virt

						-	
test	api	15	(minSdk(API	19)	>	deviceSdk(API	15))

<u>C</u> reate New Virtual Device		
Use same selection for future launches	ок	Cancel Help

Editer la configuration

Modifier l'adresse d'Esup-Nfc-Tag-Server dans esupnfctag.properties

esupNfcTagServerUrl=https://esup-nfc-tag.univ-ville.fr

Compilation

Cliquer sur "Build -> Generate Signed APK" puis renseigner le keystore

Visual Studio

Esup-CNOUS-Client est développé sous Visual Studio Community https://www.visualstudio.com/fr/

Pré-requis

- Windows
- Framework .net 2.0 minimum

Importer le projet

Après avoir cloné le projet (dans c:\opt\esup-nfc-tag-droid), faire "Fichier -> Ouvrir un projet" puis choisir le fichier sIn

Compilation

Choisir "Release" au niveau de la barre d'outils" puis cliquer sur "Générer -> Générer la solution" (raccourci Ctrl+Maj+B)

Dans le dossier c:\opt\esup-crous-client\bin\Release on trouve l'executable CreationCarteCrous.exe ainsi que le fichier de configuration CreationCarteCrous.exe.config