

STAGE DE PREMIERE ANNEE

GIE Convergence

TABLE DES MATIERES

Découverte de l'outil de développement :	3
Création des nouvelles fenêtres dédiées au module :	4
Analyse du projet :.....	6
Création des nouvelles structures	8
Code sous le clic des boutons suivant	9
Gestion des documents :.....	11
Ajout de document.....	11
Suppression d' un document:	16
Validation d'une inscription	17
Coté API :.....	21
Connexion à la base de données	21
Création de la procédure CreationInscription() :.....	22
Création de l'endpoint de l'API	26

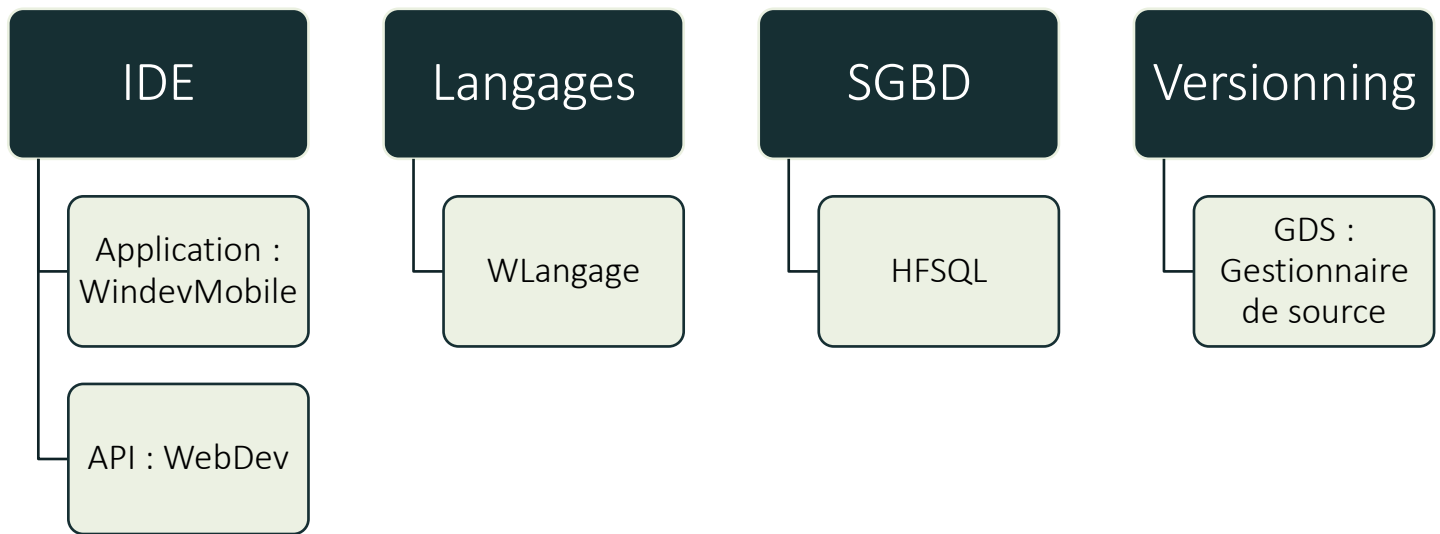
Présentation de l'existant

Une application de la ville de Laon était déjà en place et le GIE Convergence a été chargé d'en réaliser une refonte complète dans le but d'améliorer les fonctionnalités et répondre aux besoins des usagers.



Objectifs du stage

Après un point d'échange avec ma tutrice il m'a été confié le développement du module complémentaire afin de permettre aux usagers d'inscrire leurs enfants directement via l'application.



DECOUVERTE DE L'OUTIL DE DEVELOPPEMENT :

N'ayant jamais programmé avec WinDev auparavant, j'ai consacré quelques jours à une autoformation à l'aide de tutoriels. Cette démarche m'a permis de me familiariser avec l'environnement de développement ainsi qu'avec la logique propre à PC SOFT, qui se distingue par une syntaxe majoritairement en français.

PRESENTATION DU MODULE

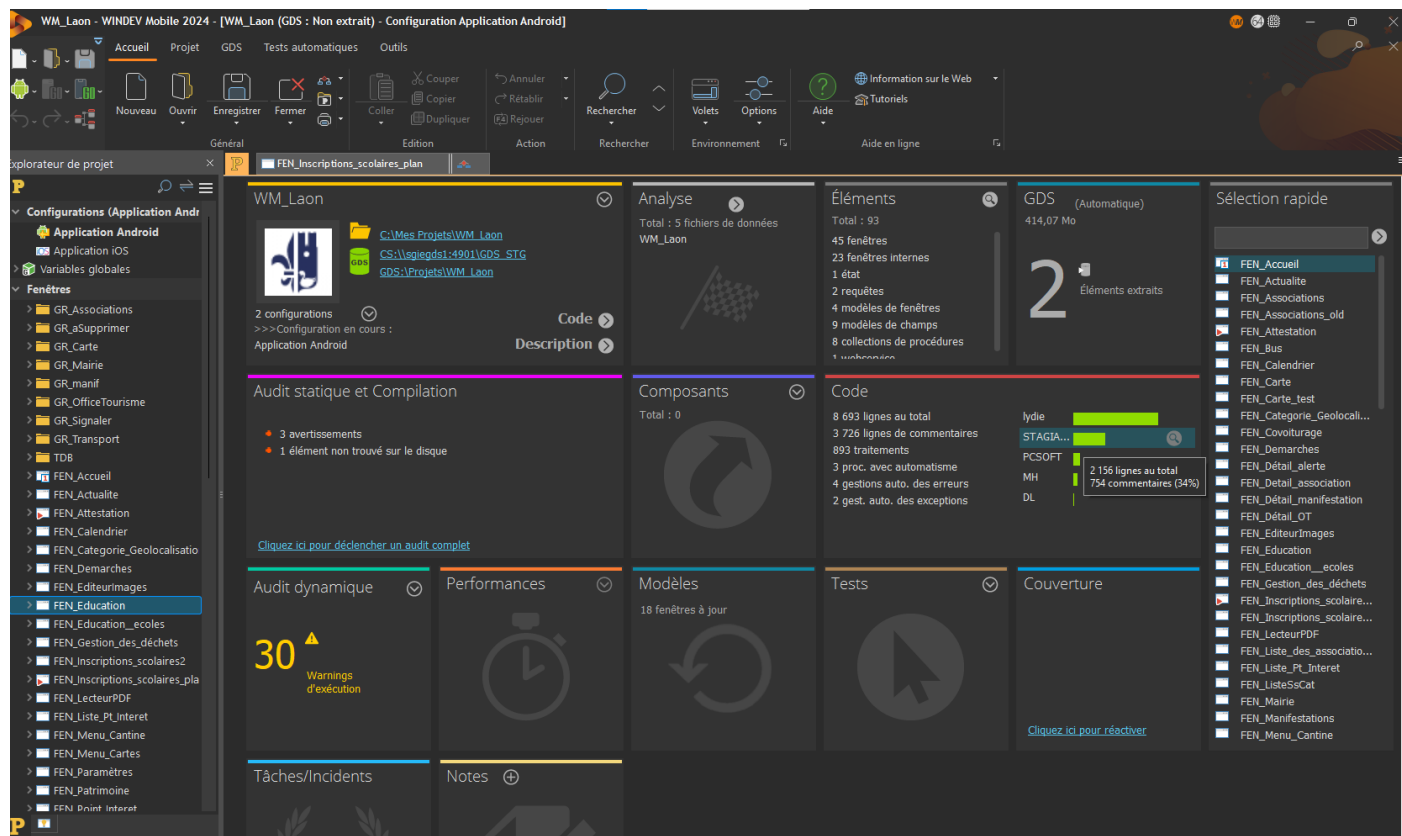
Ce module s'inscrira dans la section « Scolaire et Périscolaire » de l'application. Son objectif est de permettre aux familles d'inscrire directement leurs enfants via l'application, tout en leur offrant la possibilité de déposer les documents nécessaires en les sélectionnant depuis leur smartphone ou en les prenant en photo.

CREATION DES NOUVELLES FENETRES DEDIES AU MODULE :

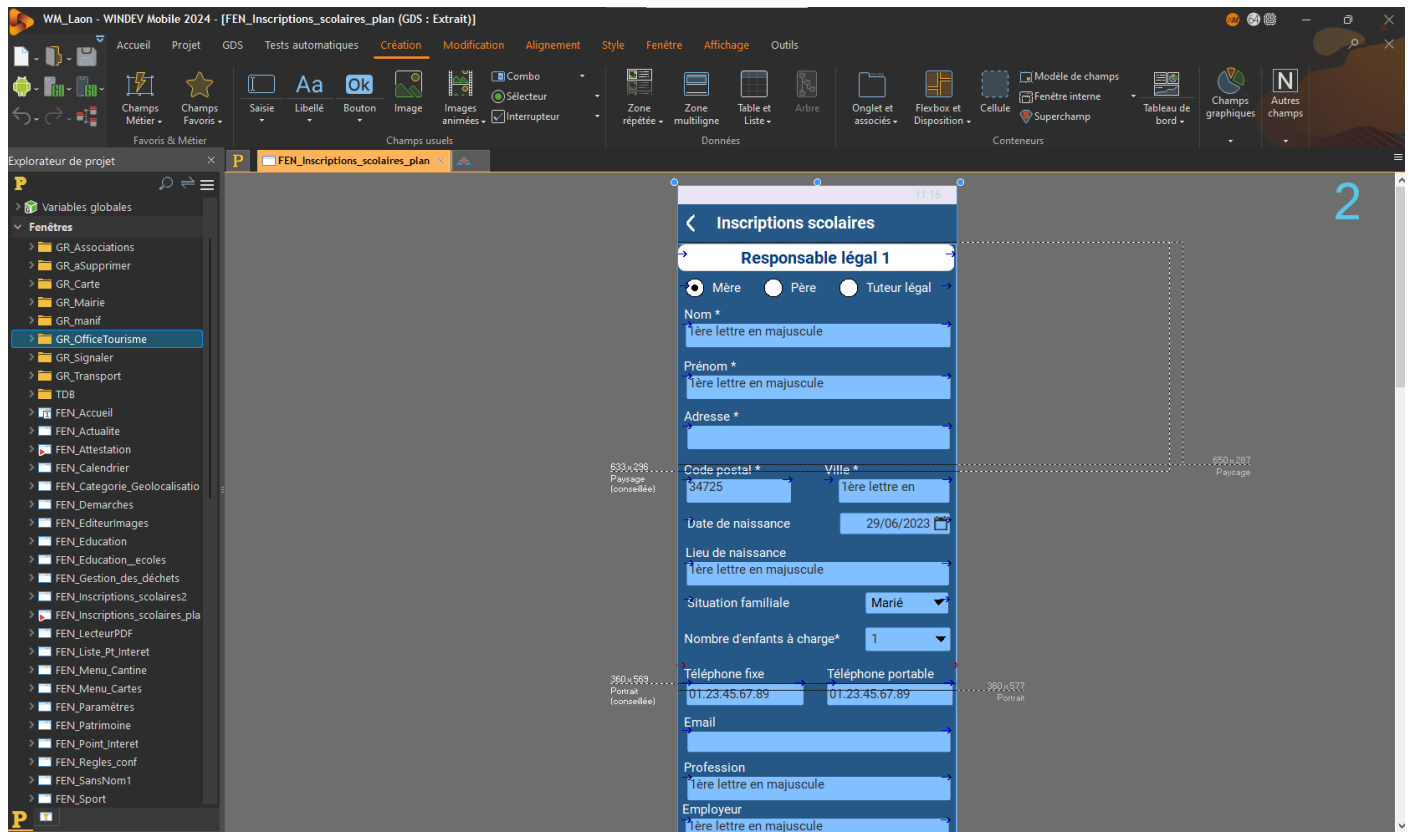
Voici l'interface déjà en place :



➡ Interface de l'outil de développement Windev mobile PCSOFT :



➡ Création des nouvelles fenêtres en intégrant les champs nécessaires et en respectant la chartre graphique déjà existante et en s’inspirant de la page web déjà mise en place :



ANALYSE DU PROJET

En WinDev (ou WebDev) l'**analyse** représente le modèle de données de l'application. C'est l'équivalent d'un schéma de base de données relationnelle, mais enrichi de fonctionnalités spécifiques à PCSOFT.

Elle permet :

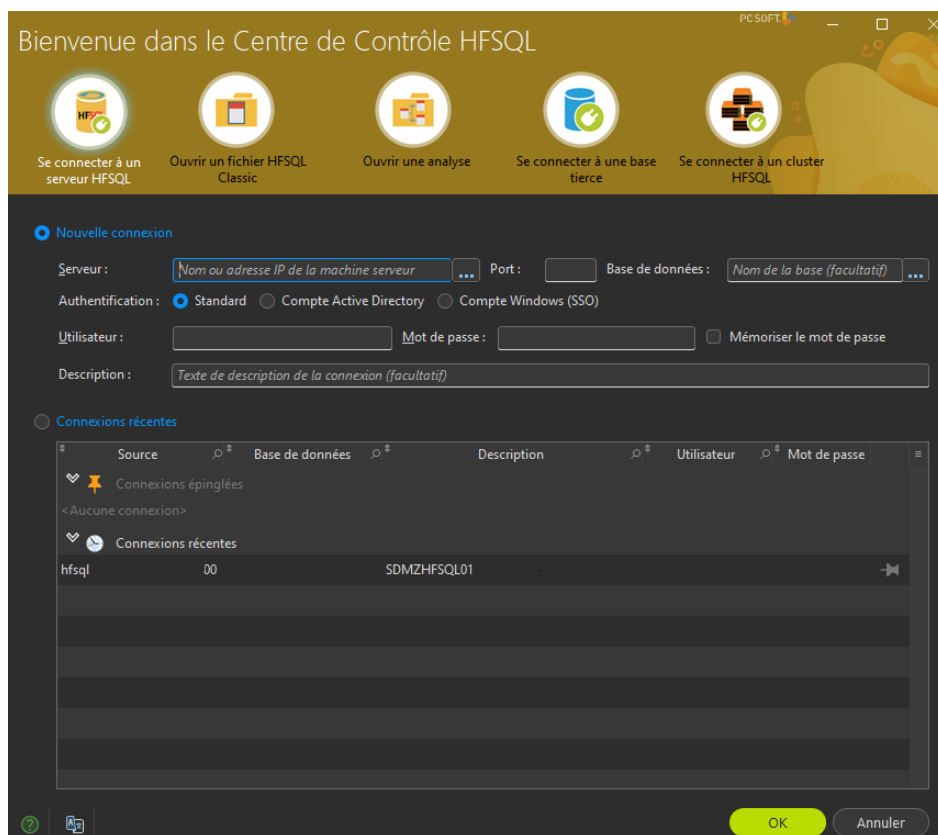
- De définir les fichiers de données (équivalent des tables en SQL)
- Définir les champs
- Générer automatiquement les requêtes



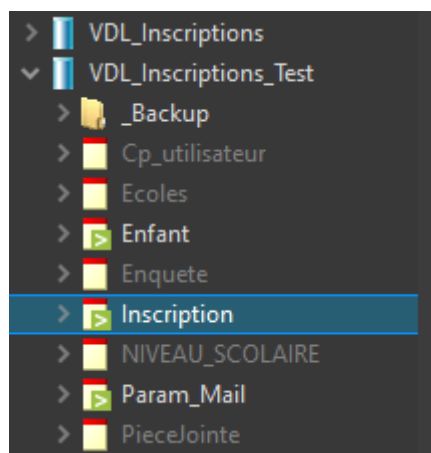
SYNCHRONISATION AVEC LA BASE DE DONNEES HFSQL :

Qu'est-ce que **HFSQL** ?

C'est une base de données développée par PCSOFT conçue pour s'intégrer avec les outils de développement WinDev et WebDev.



Après création de l'analyse les nouvelles tables ont bien été ajoutées à la base de données test:



CREATION DES NOUVELLES STRUCTURES

Création de deux structures, StEnfant et StInscriptions, au niveau de l'API VDL et de l'application mobile, permettant de collecter les informations saisies par les utilisateurs dans les champs de l'interface de l'application.

```
FEN_Inscriptions_scolaires_plan COL_Structures - Dé
stInscription est une Structure
  tabEnfant          est un tableau de stEnfant
  IDInscription      est un entier
  mptRespLegal1      est une chaîne
  nomRespLegal1      est une chaîne
  prenomRespLegal1   est une chaîne
  adresseRespLegal1  est une chaîne
  cpRespLegal1       est une chaîne
  villeRespLegal1    est une chaîne
  dateNaissRespLegal1 est une Date
  lieuNaissRespLegal1 est une chaîne
  sitFamilialeRespLegal1 est une chaîne
  nbEnfChargeRespLegal1 est un entier
  telFixeRespLegal1  est une chaîne
  telPortableRespLegal1 est une chaîne
  mailRespLegal1     est une chaîne
  professionRespLegal1 est une chaîne
  employeurRespLegal1 est une chaîne
  lieuTravailRespLegal1 est une chaîne
  telProRespLegal1   est une chaîne
  numCAFRespLegal1   est une chaîne
  numMSARespLegal1   est une chaîne
  assuranceScoRespLegal1 est une chaîne
  numAssScolRespLegal1 est une chaîne
  nomRespLegal2      est une chaîne
  prenomRespLegal2   est une chaîne
  adresseRespLegal2  est une chaîne
  cpRespLegal2       est une chaîne
  villeRespLegal2    est une chaîne
  dateNaissRespLegal2 est une Date
  lieuNaissRespLegal2 est une chaîne
  sitFamilialeRespLegal2 est une chaîne
  nbEnfChargeRespLegal2 est un entier
  telFixeRespLegal2  est une chaîne
  telPortableRespLegal2 est une chaîne
  mailRespLegal2     est une chaîne
  professionRespLegal2 est une chaîne
  employeurRespLegal2 est une chaîne
  lieuTravailRespLegal2 est une chaîne
  telProRespLegal2   est une chaîne
  numCAFRespLegal2   est une chaîne
  numMSARespLegal2   est une chaîne
  assuranceScoRespLegal2 est une chaîne
  numAssScolRespLegal2 est une chaîne
  autorisationParentMere est un booléen
```

```
stEnfant est une Structure
  IDEnfant          est un entier
  IDInscription      est un entier
  nom                est une chaîne
  prenom             est une chaîne
  bAdresseRL1        est un booléen
  adresse            est une chaîne
  codePostal         est une chaîne
  ville              est une chaîne
  dateNaissance      est une Date
  lieuNaissance      est une chaîne
  sexe               est une chaîne
  ecole              est une chaîne
  niveauScolaire     est une chaîne
  regAlimSansPorc    est un booléen
  allergieAlimentaire est un booléen
  allergieMedicaments est un booléen
  allergieAutres      est un booléen
  allergiePrécisions  est une chaîne
  portLunettes        est un booléen
  portLentilles       est un booléen
  appareilDentaire    est un booléen
  autreRecommandation est une chaîne
  nomMedecin          est une chaîne
  telMedecin          est une chaîne
  nomPrenomAutrePers1 est une chaîne
  telFixeAutrePers1   est une chaîne
  telPortableAutrePers1 est une chaîne
  lienAutrePers1      est une chaîne
  contactUrgence1     est un booléen
  priseEnCharge1      est un booléen
  nomPrenomAutrePers2 est une chaîne
  telFixeAutrePers2   est une chaîne
  telPortableAutrePers2 est une chaîne
  lienAutrePers2      est une chaîne
  contactUrgence2     est un booléen
  priseEnCharge2      est un booléen
  nomPrenomAutrePers3 est une chaîne
  telFixeAutrePers3   est une chaîne
  telPortableAutrePers3 est une chaîne
  lienAutrePers3      est une chaîne
  contactUrgence3     est un booléen
  priseEnCharge3      est un booléen
  autorisation1       est un booléen
```

CODE SOUS LE CLIC DES BOUTONS SUIVANT

Chaque bouton suivant contient un code qui, au fur et à mesure de la navigation dans les différentes fenêtres de l'application, permet de lier les structures et les champs correspondants.

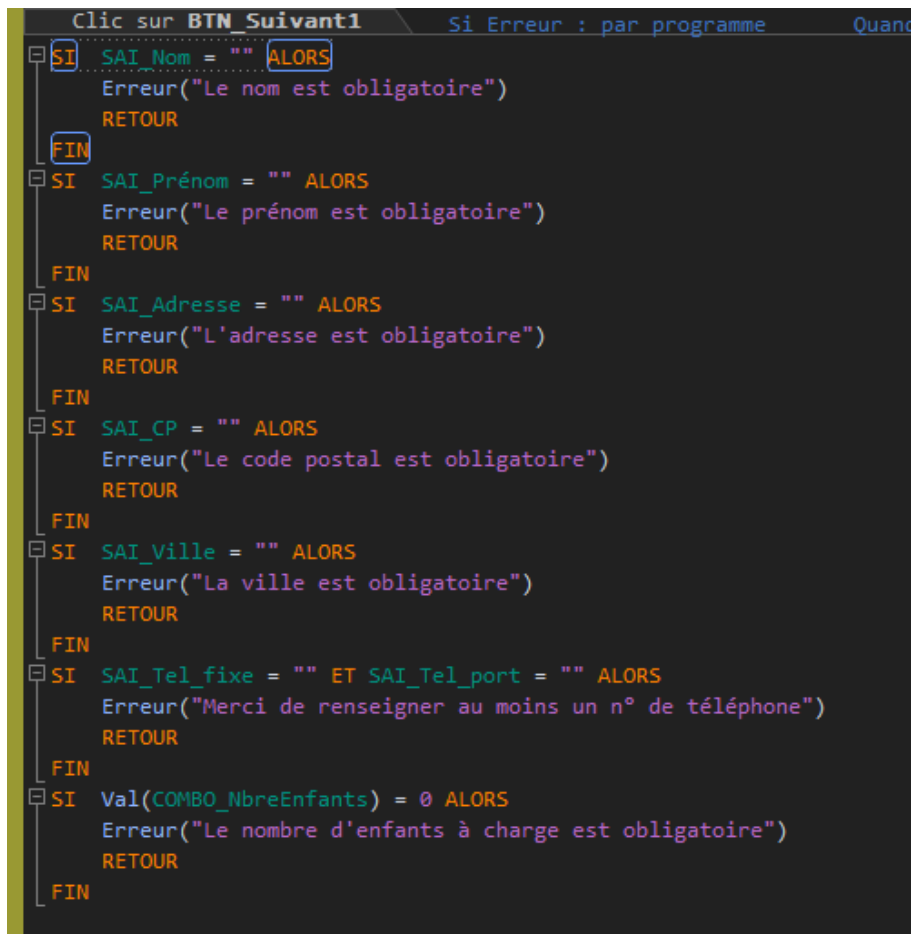
Ce code inclus également des vérifications pour garantir que les champs obligatoires sont bien remplis et que les règles logiques sont respectées. Par exemple, si un parent sélectionne "séparé" ou "divorcé" dans le menu déroulant de la situation familiale, l'option "autorité parentale" doit impérativement être cochée.

```
Clc sur BTN_Suivant2      Si Erreur : par programme      Quand Exception : par programme
// Sélection d'une catégorie de garde
// Si choix séparé ou divorcé
SI COMBO_Situation_familiale.ValeurAffichée = "Divorcé" OU COMBO_Situation_familiale.ValeurAffichée = "Séparé" ALORS
  SI INT_AutoriteMere = Faux ET INT_AutoritePere = Faux ALORS
    Info("L'autorité parentale est obligatoire")
    RETOUR
  FIN
  SI INT_exclusifMere = Faux ET INT_exclusifPere = Faux ET INT_Alterne = Faux ALORS
    Info("La garde d'enfant est obligatoire")
  SINON
    DonneFocus(LIB_Fiche_sanitaire)
  FIN
FIN

MaFenêtre..Plan = 4

// Contenu
stInscriptions.autoriteParentalePere      = INT_AutoritePere
stInscriptions.autoriteParentaleMere     = INT_AutoriteMere
stInscriptions.jugement                  = INT_jugement
stInscriptions.gardeClassique            = CELL_classique.INT_Classique
stInscriptions.gardeClassiqueMere        = CELL_classique.INT_classMere
stInscriptions.gardeClassiquePere        = INT_classPere
stInscriptions.gardeAlternee             = INT_Alterne
stInscriptions.semainePaireMere          = INT_SPmere
stInscriptions.semainePairePere          = INT_SPpere
stInscriptions.semaineImpMere            = INT_IMPmere
stInscriptions.semaineImpPere            = INT_IMPpere
stInscriptions.autreGarde                = SAI_Autre
stInscriptions.gardeExclusif             = INT_Exclusif
stInscriptions.gardeExclusifMere         = INT_exclusifMere
stInscriptions.gardeExclusifPere         = INT_exclusifPere
```

De même pour le remplissage des champs obligatoires pour l'utilisateur :



GESTION DES DOCUMENTS :

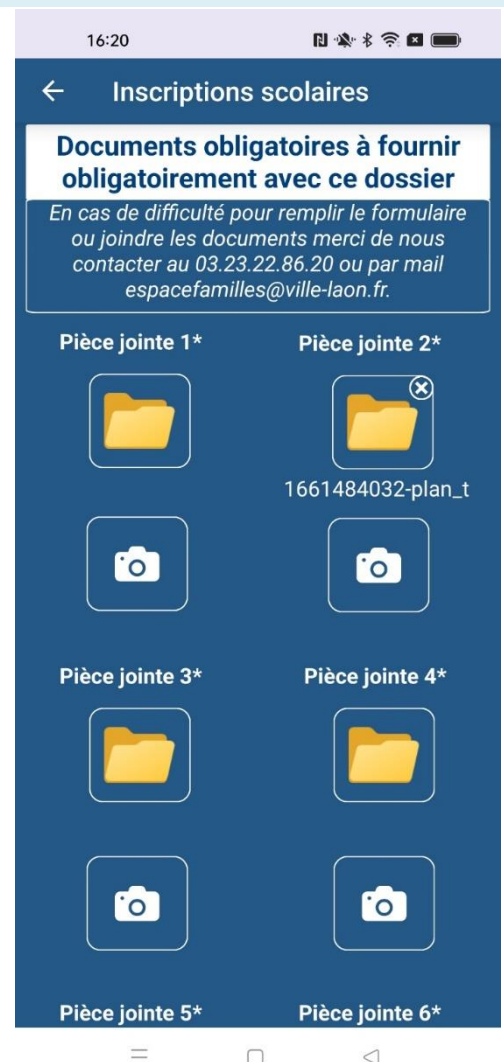
Déclaration des variables globales du projet :

```
Navigation Affichage Edition Points d'arrêt Marques Langues
P FEN_Inscriptions_scolaires_plan
1 2 Déclarations globales de FEN_Inscriptions_scolaires_plan Si Erreur : par programme Quand Exception : par programme //
2 3 PROCEDURE MaFenêtre()
3 4 stInscriptions est une stInscription
4 5 stEnfants est une stEnfant
5 6 autreEnfant est un stEnfant
6 7 // photos
7 8 gsNomPhoto1, gsNomPhoto2, gsNomPhoto3, gsNomPhoto4, gsNomPhoto5, gsNomPhoto6 sont des chaînes
8 9 // Docs
9 10 gsNomDoc1, gsNomDoc2, gsNomDoc3, gsNomDoc4, gsNomDoc5, gsNomDoc6 sont des chaînes
10 11 // buffer documents
11 12 bNomDoc1, bNomDoc2, bNomDoc3, bNomDoc4, bNomDoc5, bNomDoc6 sont des Buffer
12 13 // mémorise le choix de l'utilisateur
13 14 gnChoixutilisateur est un entier
14 15 // Variable affichage nom du document chargé
15 16 gsInformation est une chaîne
16 17
17 18 Fin d'initialisation de FEN_Inscriptions_scolaires_plan
18 19 SI EnModeTest() ALORS
19 20 MaFenêtre..Plan = 7
20 21 SINON
21 22 MaFenêtre..Plan = 1
22 23 FIN
23 24
24 25 Demande de mise à jour de l'affichage de FEN_Inscriptions_scolaires_plan
25 26
```

AJOUT DE DOCUMENT

➡ 1^{er} essai :

Au début, ne connaissant pas toutes les possibilités concernant les médias/ documents de Windev je suis partie sur une interface comme celle-ci :



Après avoir échangé avec ma tutrice, cette interface s'est révélée trop chargée et manquait d'optimisation. Elle m'obligeait à gérer les photos et les documents de manière distincte, avec plusieurs procédures différentes. J'ai donc poursuivi mes recherches sur le site de PCSOFT et ses forums pour trouver une solution plus efficace.

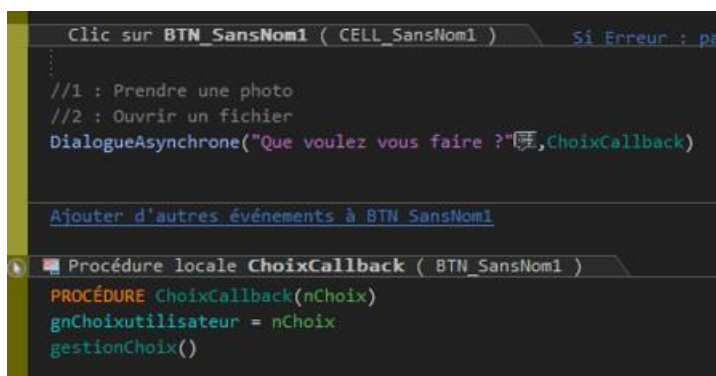
L'objectif est ici d'avoir pour chaque pièce demandée un seul et unique bouton laissant le choix à l'utilisateur de récupérer un document présent sur son téléphone ou de prendre une photo directement.

➡ 2^{ème} essai

Suite à mes recherches, j'ai opté pour une interface utilisant la fonction **DialogueAsynchrone()** avec un procédure **callback**.

<https://doc.pcsoft.fr/fr-FR/?1000025310&name=DialogueAsynchrone>

En WinDev, une procédure callback est une fonction que l'on définit et qui est automatiquement appelée lorsqu'un événement spécifique se produit, comme ici le choix entre sélectionner un document ou prendre une photo.

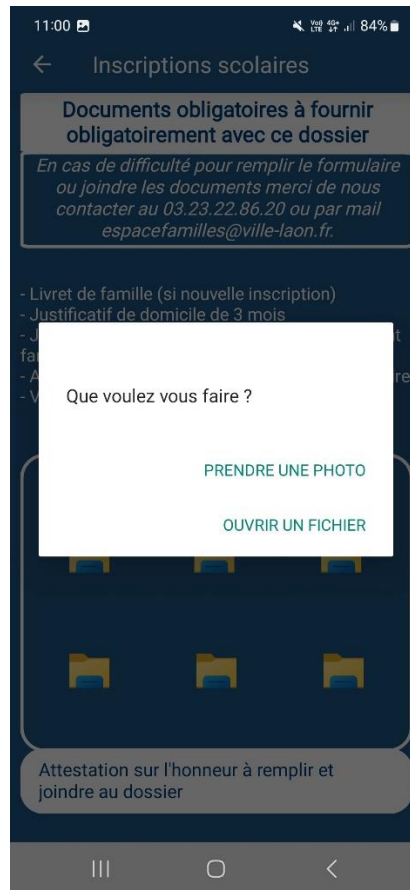
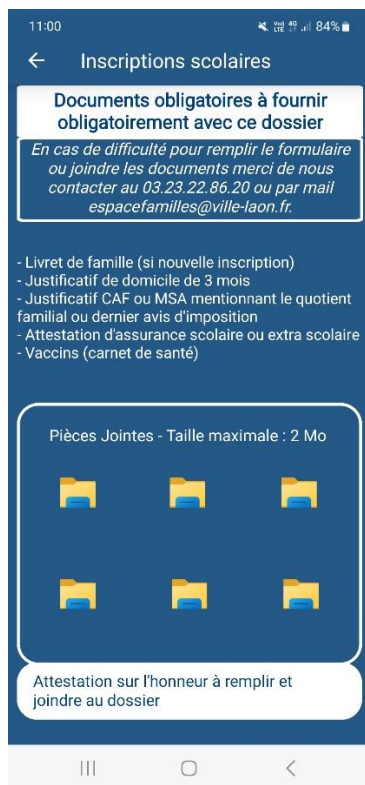


```
Clic sur BTN_SansNom1 ( CELL_SansNom1 )      Si Erreur : pas de gestion
...
//1 : Prendre une photo
//2 : Ouvrir un fichier
DialogueAsynchrone("Que voulez vous faire ?", CHOIX, ChoixCallback)

Ajouter d'autres événements à BTN_SansNom1

Procédure locale ChoixCallback ( BTN_SansNom1 )
PROCÉDURE ChoixCallback(nChoix)
gnChoixutilisateur = nChoix
gestionChoix()
```

Au clic sur le bouton de la pièce jointe une boîte de dialogue est ouverte et propose deux choix à l'utilisateur :



Le résultat du choix est ensuite géré dans la procédure `ChoixCallBack` : quand l'utilisateur fait un choix, cette procédure est appelée automatiquement avec la valeur en paramètre :

- `nChoix` contient la sélection (1 ou 2)
- Ce choix est stocké dans une variable globale `gnChoixutilisateur`
- La procédure `gestionChoix` est appelée pour exécuter le bon traitement selon le choix effectué

➡ Création de la procédure `gestionChoix()` :

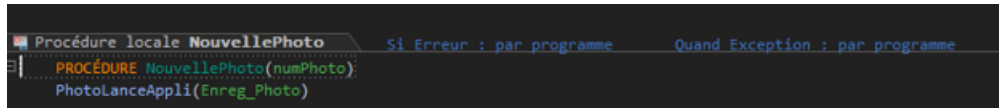
```

Procédure locale gestionChoix ( BTN_
PROCÉDURE gestionChoix()
SELON gnChoixutilisateur
CAS 1
    FichierPhoto(1)
CAS 2
    Plateforme(gsNomDoc1)
    LIB_doc1 = gsInformation
FIN
    
```

Selon le choix de l'utilisateur une procédure contenant le traitement à exécuté est appelée :

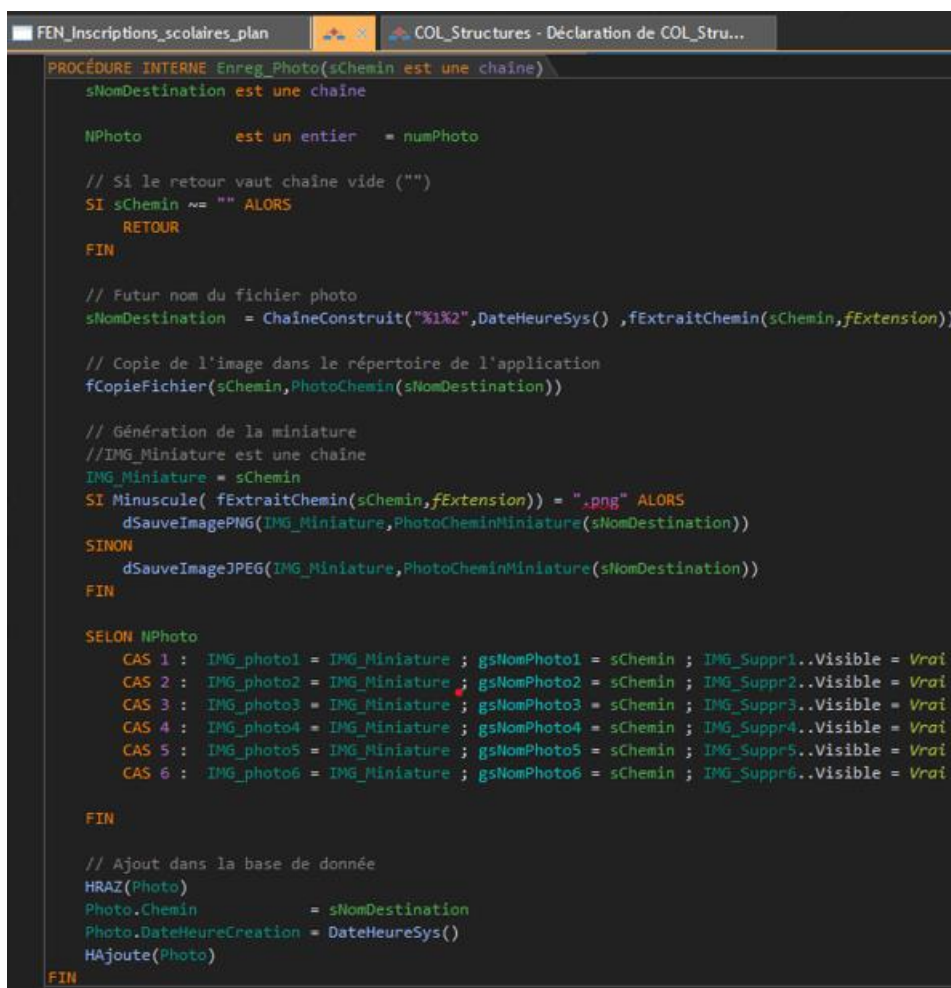
- Si le choix est de prendre une photo : procédure [NouvellePhoto\(\)](#)
- Si c'est un document : procédure [Plateforme\(\)](#) prenant en paramètre le document.

CREATION D'UNE PROCEDURE POUR GERER LE CAS DE LA PHOTO :



Utilisation de la fonction [PhotoLanceAppli\(\)](#) prenant en paramètre une procédure callback [Enreg_Photo\(\)](#).

➡ Création de la procédure [Enreg_Photo](#) :



- Prends en paramètre le chemin de la photo
- Construction d'un nom de fichier unique avec [ChaîneConstruit\(\)](#) afin d'éviter les conflits de noms.
- Copie du fichier dans le répertoire de l'application avec [fCopieFichier\(\)](#)
- Création d'une miniature en fonction du type de fichier (png ou jpeg), puis enregistrement de celle-ci dans la zone d'image pour qu'elle soit affichée dès son chargement, accompagnée du bouton de suppression.
- Enregistrement dans la base de donnée avec [HAJoute](#)

CAS D'UN DOCUMENT :

➡ Sur Android :

Ouverture d'un sélecteur avec `URISélecteur()` : <https://doc.pcsoft.fr/fr-FR/?1000024026>

Une fois que l'utilisateur a choisi un fichier et qu'aucune erreur n'a été détectée :

- On récupère le nom du fichier ainsi que son type MIME avec `URIRécupèreInfo()`
- Extraction du nom du fichier et de son extension avec `fExtraitChemin()` qui permet d'obtenir le nom réel du fichier sans le chemin complet.
- Le contenu est chargé dans un buffer avec `URIVersBuffer()`

```
<COMPILE SI TypeConfiguration=Android>
UriRessource    est une URI = URISélecteur(uriImage + TAB + "application/pdf")
Information     est une chaîne
SI PAS ErreurDétectée ALORS
    // Affiche le nom de la ressource sélectionnée
    Info(URIRécupèreInfo(UriRessource, uriInfoNom))
    // recupere uri
    gsNomDoc1 = fExtraitChemin(UriRessource..Valeur,fFichier+fExtension)
    sTypeMIME est une chaîne = URIRécupèreInfo(UriRessource, uriInfoTypeConversion)

    // Charge la ressource dans un buffer
    URIVersBuffer(UriRessource, ProcFinChargementRessources, sTypeMIME)
    PROCÉDURE INTERNE ProcFinChargementRessources(bResultat, buffer, sFormat)
        SI bResultat ALORS
            bNomDoc1 = buffer
            ToastAffiche("Fin du chargement")
        FIN
    FIN

    // Rempli le libellé avec le nom du fichier sélectionné
    Information = URIRécupèreInfo(UriRessource,uriInfoNom)
    LIB_doc1    = Information
    IMG_docSuppr1.Visible = Vrai
FIN
```

Une fois le chargement du fichier terminé, la procédure de callback `ProcFinChargementRessources()` est appelée automatiquement et reçoit :

- `Bresultat` : booléen indiquant si le chargement a été effectué avec succès.
- `Buffer` : contenu du fichier qui sera ensuite affecté à la variable globale correspondant au document.
- `sFormat` : format du contenu.

Après l'exécution de ces procédures le nom du fichier est affiché dans un champ dédié puis le bouton de suppression devient visible.

SUPPRESSION D' UN DOCUMENT:

```

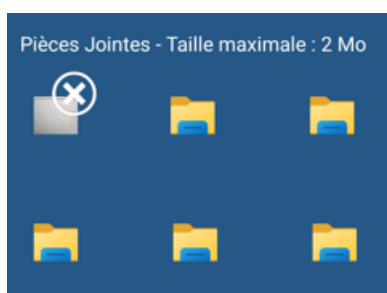
1 // Si photo
2
3 SI IMG_photo1 <> "" ALORS
4     // vide l'image enregistrée sans effacer l'image de base
5     IMG_photo1 = IMG_photo1.ValeurInitiale
6     // vide le chemin de la photo
7     gsNomPhoto1 = ""
8     Info (" La photo a bien été réinitialisée")
9     FIN
10 // Si fichier
11 SI gsNomDoc1 <> "" ALORS
12     // vide le chemin du document
13     gsNomDoc1 = ""
14     // vide le libellé du document
15     LIB_doc1 = ""
16     Info("Le document a bien été réinitialisé")
17
18 FIN
19
```

➡ Code sous l'image contenant une croix utilisée pour la suppression :

```

Initialisation de IMG_S1 ( CELL_SansNom3 )
// Positionnement de la croix de suppression:
IMG_S1..X = IMG_1..X + IMG_1..Largeur - IMG_S1..Largeur
IMG_S1..Y = IMG_1..Y
// Etat initial masquée
IMG_S1..Visible = Faux
```

- Permet de positionner l'image en haut à droite de l'image ajoutée afin de procéder à sa suppression si nécessaire.



INSCRIPTION DE PLUSIEURS ENFANTS

Si un utilisateur souhaite inscrire plusieurs enfants, il peut, au lieu de cliquer immédiatement sur le bouton envoyer la demande, cliquer sur « Inscrire un autre enfant » ce qui va le rediriger vers la fenêtre de renseignements de l'enfant.

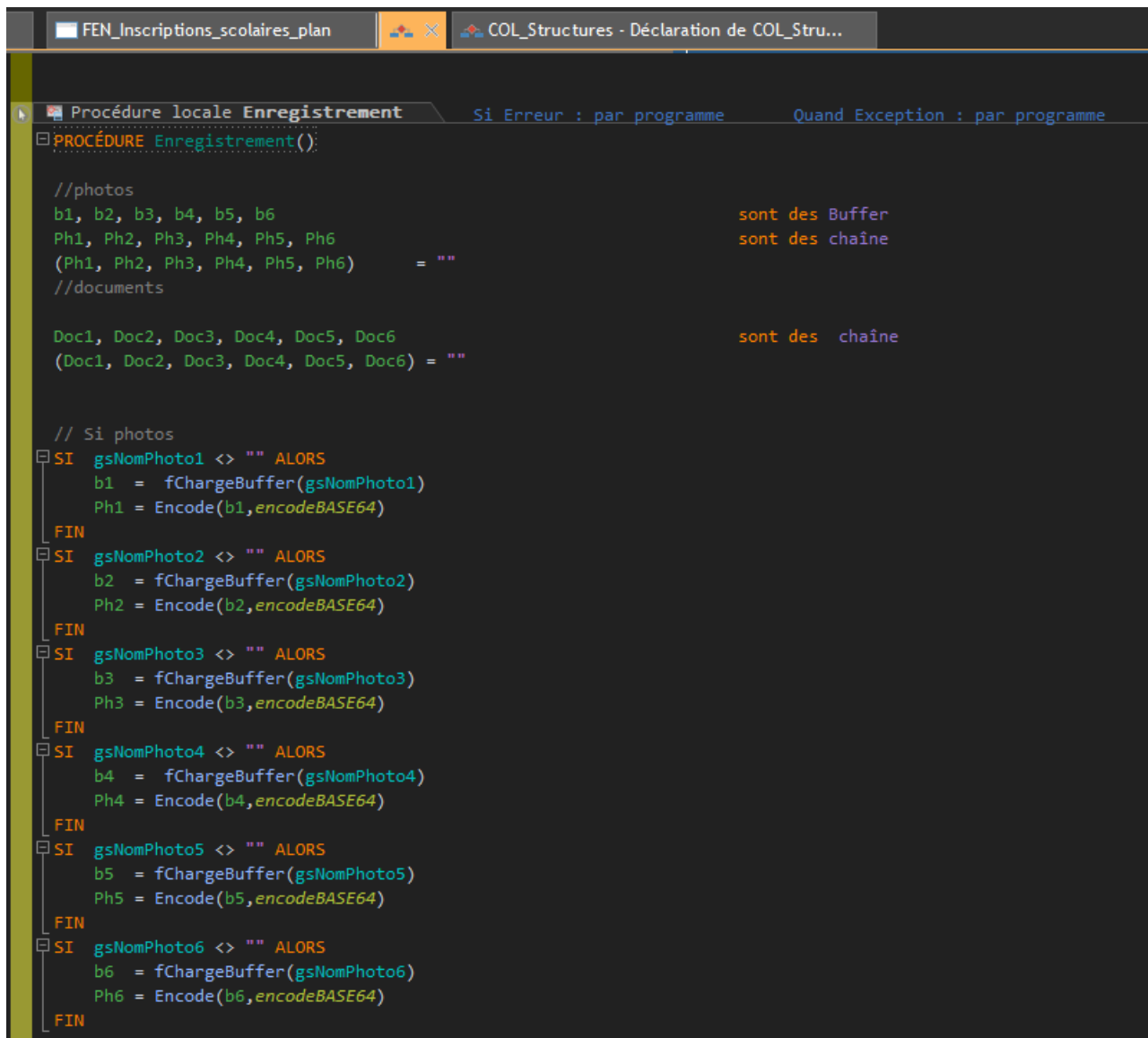
```
Initialisation de BTN_Inscrire_un_autre_enfant3
1 // Version 1
2 // Description
3 // Bouton simple
4
Clic sur BTN_Inscrire_un_autre_enfant3 Si Erreur : par programme
1 //RAZ(FEN Inscriptions scolaires_plan.Plan = 4);
2 MaFenêtre..Plan = 4
3 DonneFocus(LIB_Fiche_sanitaire)
4 // Valide collecte de données pour le 1er enfant
5 stEnfants.lu_approuve = INT_lu_approuve1
6 autreEnfant.lu_approuve = stEnfants.lu_approuve
7 // Ajout d'un autre enfant au tableau
8 stInscriptions.tabEnfant.Ajoute(autreEnfant)
9 // Ré initialise les champs présents au plan 4
10 RAZ(GR_Enfants)
```

VALIDATION D'UNE INSCRIPTION

➡ La validation d'une inscription se fait au clic sur le bouton « Envoyer la demande » qui appelle la procédure `Enregistrement()` :

```
FEN_Inscriptions_scolaires_plan
Initialisation de BTN_Envoyer_la_demande3
1 // Version 1
2 // Description
3 // Bouton simple
4
Clic sur BTN_Envoyer_la_demande3 Si Erreur : par programme Quand Exception : par
1 51 INT lu_approuve1 = Faux ALORS
2 Info("Vous devez avoir 'lu et approuvé' les informations saisies pour valider l'inscription")
3 RETOUR
4 FIN
5
6 // Valide collecte de données
7 stEnfants.lu_approuve = INT_lu_approuve1
8 autreEnfant.lu_approuve = stEnfants.lu_approuve
9 // Ajout du dernier enfant avant envoi demande
10 stInscriptions.tabEnfant.Ajoute(autreEnfant)
11
12 Enregistrement()
13
```

➡ Création d'une procédure Enregistrement :



```

FEN_Inscriptions_scolaires_plan  COL_Structures - Déclaration de COL_Stru...

Procédure locale Enregistrement  Si Erreur : par programme  Quand Exception : par programme
PROCÉDURE Enregistrement():

    //photos
    b1, b2, b3, b4, b5, b6          sont des Buffer
    Ph1, Ph2, Ph3, Ph4, Ph5, Ph6    sont des chaîne
    (Ph1, Ph2, Ph3, Ph4, Ph5, Ph6) = ""
    //documents

    Doc1, Doc2, Doc3, Doc4, Doc5, Doc6    sont des chaîne
    (Doc1, Doc2, Doc3, Doc4, Doc5, Doc6) = ""

    // Si photos
    SI gsNomPhoto1 <> "" ALORS
        b1 = fChargeBuffer(gsNomPhoto1)
        Ph1 = Encode(b1,encodeBASE64)
    FIN
    SI gsNomPhoto2 <> "" ALORS
        b2 = fChargeBuffer(gsNomPhoto2)
        Ph2 = Encode(b2,encodeBASE64)
    FIN
    SI gsNomPhoto3 <> "" ALORS
        b3 = fChargeBuffer(gsNomPhoto3)
        Ph3 = Encode(b3,encodeBASE64)
    FIN
    SI gsNomPhoto4 <> "" ALORS
        b4 = fChargeBuffer(gsNomPhoto4)
        Ph4 = Encode(b4,encodeBASE64)
    FIN
    SI gsNomPhoto5 <> "" ALORS
        b5 = fChargeBuffer(gsNomPhoto5)
        Ph5 = Encode(b5,encodeBASE64)
    FIN
    SI gsNomPhoto6 <> "" ALORS
        b6 = fChargeBuffer(gsNomPhoto6)
        Ph6 = Encode(b6,encodeBASE64)
    FIN

```

Les variables gsNomPhoto représentent le chemin des photos téléchargées par l'utilisateur.

Pour chaque photo, la procédure vérifie si une photo existe et si c'est le cas, charge l'image dans un buffer à l'aide de la fonction `fChargeBuffer` et l'encode en base64.

```

// Si document
SI gsNomDoc1 <> "" ALORS
  //charge le pdf en buffer
  // bNomDoc1 = fChargeBuffer(gsNomDoc1)
  //convertir
  Doc1 = Encode(bNomDoc1,encodeBASE64)
FIN
SI gsNomDoc2 <> "" ALORS
  //charge le pdf en buffer
  // bNomDoc2 = fChargeBuffer(gsNomDoc2)
  //convertir
  Doc2 = Encode(bNomDoc2,encodeBASE64)
FIN
SI gsNomDoc3 <> "" ALORS
  //charge le pdf en buffer
  // bNomDoc3 = fChargeBuffer(gsNomDoc3)
  //convertir
  Doc3 = Encode(bNomDoc3,encodeBASE64)
FIN
SI gsNomDoc4 <> "" ALORS
  //charge le pdf en buffer
  // bNomDoc4 = fChargeBuffer(gsNomDoc4)
  //convertir
  Doc4 = Encode(bNomDoc4,encodeBASE64)
FIN
SI gsNomDoc5 <> "" ALORS
  //charge le pdf en buffer
  // bNomDoc5 = fChargeBuffer(gsNomDoc5)
  //convertir
  Doc5 = Encode(bNomDoc5,encodeBASE64)
FIN
SI gsNomDoc6 <> "" ALORS
  //charge le pdf en buffer
  // bNomDoc6 = fChargeBuffer(gsNomDoc6)
  //convertir
  Doc6 = Encode(bNomDoc6,encodeBASE64)

```

Dans le cas des documents, ceux-ci étant déjà chargés en buffer lors de la procédure `Plateforme()`, il suffit donc de les encoder.

➡ Envoi des données :

Une fois toutes les photos et documents encodés en Base64, les variables correspondantes sont affectées aux champs de `stInscriptions` qui contiendra toutes les données à envoyer.

```

maRequete  est un restRequête
maReponse  est une restRéponse

// URL
maRequete.Méthode      = httpPost
maRequete.ContentType  = typeMimeJSON
maRequete.URL          = gswAdresseAPIVDL + "CreationInscription"

stInscriptions.Photo1  = Ph1
stInscriptions.Photo2  = Ph2
stInscriptions.Photo3  = Ph3
stInscriptions.Photo4  = Ph4
stInscriptions.Photo5  = Ph5
stInscriptions.Photo6  = Ph6
stInscriptions.Doc1    = Doc1
stInscriptions.Doc2    = Doc2
stInscriptions.Doc3    = Doc3
stInscriptions.Doc4    = Doc4
stInscriptions.Doc5    = Doc5
stInscriptions.Doc6    = Doc6

bufContenu est un Buffer
Séréalise(stInscriptions,bufContenu,psdJSON)
maRequete.Contenu      = bufContenu

// Requête
maReponse              = RESTEnvoie(maRequete)

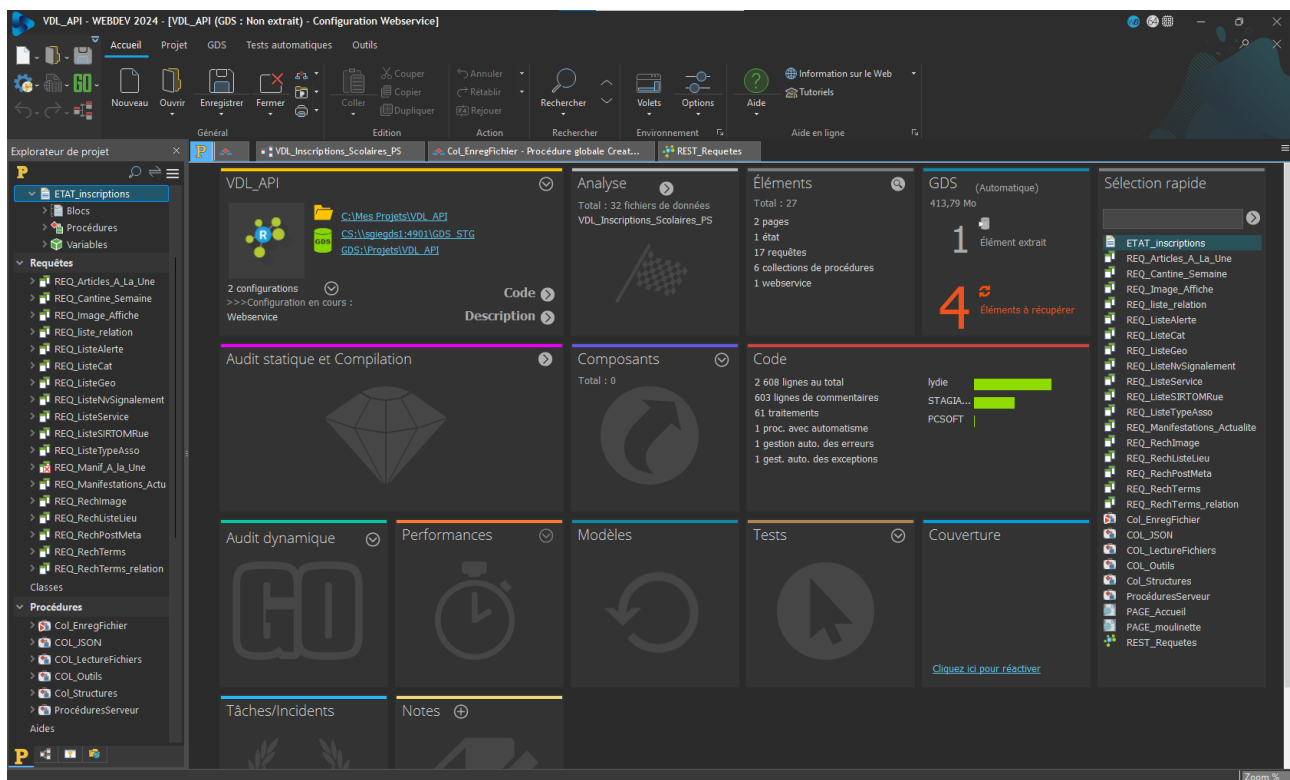
```

Une requête HTTP POST est utilisée pour envoyer ces informations à l'API REST:

- l'url de l'api et les autres paramètres de la requête sont configurés : méthode, type du contenu.
- les données sont sérialisées en JSON ce qui permet de les convertir en une chaîne structurée puis placée dans un buffer avant l'envoi.

COTE API :

Page d'accueil de l'API sous WebDev :



CONNEXION A LA BASE DE DONNEES

Création du code de connexion:

```
Cx_WS est une Connexion

SI EnModeTest() ALORS
    Cx_WS..BaseDeDonnées = "Ville_Laon"
    Cx_WS..InfosEtendues = ";Cryptage = aucun;Initial Catalog=Ville_Laon"
    Cx_WS..Provider = hAccèsHFClientServeur
    Cx_WS..Serveur = " "
    Cx_WS..Utilisateur = " "
    Cx_WS..MotDePasse = " "
SINON
    Cx_WS..BaseDeDonnées = "Ville_Laon"
    Cx_WS..InfosEtendues = ";Cryptage = aucun;Initial Catalog=Ville_Laon"
    Cx_WS..Provider = hAccèsHFClientServeur
    Cx_WS..Serveur = " "
    Cx_WS..Utilisateur = " "
    Cx_WS..MotDePasse = " "
FIN

HOuvreConnexion(Cx_WS)
HChangeConnexion("", Cx_WS)

HCréationSiInexistant("")

//recherche id service municipaux
gnIdServMun,gnIdStationnement,gnIdGeoloc_OT sont des entiers

HDésactiveFiltre(Parametre)
POUR TOUT Parametre
    SI Parametre.NomParametre = "ID_SERV_MUNICIPAUX" ALORS gnIdServMun = Parametre.ValeurParametre
    SI Parametre.NomParametre = "ID_STATIONNEMENT" ALORS gnIdStationnement = Parametre.ValeurParametre
    SI Parametre.NomParametre = "ID_GEOLOC_OT" ALORS gnIdGeoloc_OT = Parametre.ValeurParametre
FIN
```

CREATION DE LA PROCEDURE CREATIONINSCRIPTION() :

➡ Récupération des données de la structure stInscription pour remplir le fichier correspondant

```
PROCÉDURE CreationInscription() : chaîne

bRes est un booléen
// Prefixe chrono - enregistrement des fichiers
wChrono est une chaîne = DateHeureSys()

// Variables envoi mail
wSujet, swMessageHTML, swTabDest, swNomP3, swTabCCi, wExpéditeur, swNomP2 sont des chaînes
// Variable état
NomFichier, NomFichierImpression sont des chaînes

eIDInscription est un entier sur 8 octets
eNBEnreg est un entier
stInscription est une stInscription
bufContenu est un Buffer
bPhotos, bDocs sont des Buffers

bufContenu = WebserviceParamètre(paramBuffer)

// Désérialisation
Désérialise(stInscription, bufContenu, psdJSON)

eNBEnreg = TableauOccurrence(stInscription.tabEnfant)
swTabDest = stInscription.mailResplégall

SI HLitPremier(Param_Mail) ALORS
  xLog("HLitPremier(Param_Mail) ")
  wExpéditeur = Param_Mail.expéditeur
  swTabCCi = Param_Mail.copieA
  xLog("Param_Mail.expéditeur " + Param_Mail.expéditeur)
FIN

HRAZ(Inscription)
```

- Déclaration des variables
- Désérialisation des données avec Désérialise() qui va convertir les données JSON en structure stInscription.
- Utilisation de HLitPremier() afin de rechercher le premier enregistrement dans param mail qui contiendra l'adresse mail qui sera utilisée comme expéditeur des emails.

VDL_Inscriptions_Scolaires_PS	Col_EnregFichier - Procédure globale Cre...
AVEC Inscription	
.mpt1	= stInscription.mptRespLegal1
.mail1	= stInscription.mailRespLegal1
.nom1	= stInscription.nomRespLegal1
.pnom1	= stInscription.prenomRespLegal1
.datenaiss1	= stInscription.dateNaissRespLegal1
.lieunaiss1	= stInscription.lieuNaissRespLegal1
.adresse1	= stInscription.adresseRespLegal1
.cp1	= stInscription.cpRespLegal1
.ville1	= stInscription.villeRespLegal1
.telfixe1	= stInscription.telFixeRespLegal1
.telportable1	= stInscription.telPortableRespLegal1
.profession1	= stInscription.professionRespLegal1
.employeur1	= stInscription.employeurRespLegal1
.lieutravail1	= stInscription.lieuTravailRespLegal1
.telpro1	= stInscription.telProRespLegal1
.sitfam1	= stInscription.sitFamilialeRespLegal1
.nbenfcharge1	= stInscription.nbEnfChargeRespLegal1
.numallocCAF1	= stInscription.numCAFRespLegal1
.numallocMSA1	= stInscription.numMSARespLegal1
.AssScolaire1	= stInscription.assuranceScoRespLegal1
.NumAssScolaire1	= stInscription.numAssScolRespLegal1
.mpt2	= stInscription.mptRespLegal2
.mail2	= stInscription.mailRespLegal2
.nom2	= stInscription.nomRespLegal2
.pnom2	= stInscription.prenomRespLegal2
.sitfam2	= stInscription.sitFamilialeRespLegal2
.adresse2	= stInscription.adresseRespLegal2
.datenaiss2	= stInscription.dateNaissRespLegal2
.lieunaiss2	= stInscription.lieuNaissRespLegal2
.nbenfcharge2	= stInscription.nbEnfChargeRespLegal2
.cp2	= stInscription.cpRespLegal2
.ville2	= stInscription.villeRespLegal2
.telfixe2	= stInscription.telFixeRespLegal2
.telportable2	= stInscription.telPortableRespLegal2
.profession2	= stInscription.professionRespLegal2
.employeur2	= stInscription.employeurRespLegal2
.lieutravail2	= stInscription.lieuTravailRespLegal2
.telpro2	= stInscription.telProRespLegal2
.numallocCAF2	= stInscription.numCAFRespLegal2
.numallocMSA2	= stInscription.numMSARespLegal2
.AssScolaire2	= stInscription.assuranceScoRespLegal2
.NumAssScolaire2	= stInscription.numAssScolRespLegal2
.bJugement	= stInscription.jugement

Puis appel de **HAjoute** pour enregistrement en base de données

- Décode des photos

```
HAjoute(Inscription)
// Decode photos
SI Taille(stInscription.Photo1) > 0 ALORS
  bPhotos = Décode(stInscription.Photo1,encodeBASE64)
  xLog RM("1 - bphotos" + bPhotos)
  bRes = fSauveBuffer(stlocalisationDocument+ wChrono + "_photo1.jpg",bPhotos)
  xLog RM("1 - bRes" + bRes)
  stNomPJ += stlocalisationDocument+ wChrono + "_photo1.jpg"
// Inscription. = lien
FIN
SI Taille(stInscription.Photo2) > 0 ALORS
  bPhotos = Décode(stInscription.Photo2,encodeBASE64)
  xLog RM("2 - bphotos" + bPhotos)
  bRes = fSauveBuffer(stlocalisationDocument+ wChrono + "_photo2.jpg",bPhotos)
  xLog RM("2 - bRes" + bRes)
  stNomPJ +=[" "] + stlocalisationDocument+ wChrono + "_photo2.jpg"
FIN
SI Taille(stInscription.Photo3) > 0 ALORS
  bPhotos = Décode(stInscription.Photo3,encodeBASE64)
  xLog RM("3 - bphotos" + bPhotos)
  fSauveBuffer(stlocalisationDocument+ wChrono + "_photo3.jpg",bPhotos)
  stNomPJ +=[" "] + stlocalisationDocument+ wChrono + "_photo3.jpg"
FIN
SI Taille(stInscription.Photo4) > 0 ALORS
  bPhotos = Décode(stInscription.Photo4,encodeBASE64)
  fSauveBuffer(stlocalisationDocument+ wChrono + "_photo4.jpg",bPhotos)
  stNomPJ +=[" "] + stlocalisationDocument+ wChrono + "_photo4.jpg"
  *
FIN
SI Taille(stInscription.Photo5) > 0 ALORS
  bPhotos = Décode(stInscription.Photo5,encodeBASE64)
  fSauveBuffer(stlocalisationDocument+ wChrono + "_photo5.jpg",bPhotos)
  stNomPJ +=[" "] + stlocalisationDocument+ wChrono + "_photo5.jpg"
FIN
SI Taille(stInscription.Photo6) > 0 ALORS
  bPhotos = Décode(stInscription.Photo6,encodeBASE64)
  fSauveBuffer(stlocalisationDocument+wChrono + "_photo6.jpg",bPhotos)
  stNomPJ +=[" "] + stlocalisationDocument+ wChrono + "_photo6.jpg"
FIN
```


- Décode des documents

```

SI Taille(stInscription.Doc1) > 0 ALORS
  bDocs = Décode(stInscription.Doc1, encodeBASE64)
  xlog("1 - bDocs " + bDocs)
  // chemin pour la sauvegarde doc1
  bRes = fSauveBuffer(sLocalisationDocument+ wChrono + "_doc1.pdf", bDocs)
  xlog("1 doc - bRes " + bRes)
  stWomPJ +=[" "] + sLocalisationDocument+ wChrono + "_doc1.pdf"
FIN
SI Taille(stInscription.Doc2) > 0 ALORS
  bDocs = Décode(stInscription.Doc2, encodeBASE64)
  fSauveBuffer(sLocalisationDocument+ wChrono + "_doc2.pdf", bDocs)
  stWomPJ +=[" "] + sLocalisationDocument+ wChrono + "_doc2.pdf"
FIN
SI Taille(stInscription.Doc3) > 0 ALORS
  bDocs = Décode(stInscription.Doc3, encodeBASE64)
  fSauveBuffer(sLocalisationDocument+ wChrono + "_doc3.pdf", bDocs)
  stWomPJ +=[" "] + sLocalisationDocument+ wChrono + "_doc3.pdf"
FIN
SI Taille(stInscription.Doc4) > 0 ALORS
  bDocs = Décode(stInscription.Doc4, encodeBASE64)
  fSauveBuffer(sLocalisationDocument+ wChrono + "_doc4.pdf", bDocs)
  stWomPJ +=[" "] + sLocalisationDocument+ wChrono + "_doc4.pdf"
  // Inscription. = lien
FIN
SI Taille(stInscription.Doc5) > 0 ALORS
  bDocs = Décode(stInscription.Doc5, encodeBASE64)
  fSauveBuffer(sLocalisationDocument+ wChrono + "_doc5.pdf", bDocs)
  stWomPJ +=[" "] + sLocalisationDocument+ wChrono + "_doc5.pdf"
FIN
SI Taille(stInscription.Doc6) > 0 ALORS
  bDocs = Décode(stInscription.Doc6, encodeBASE64)
  fSauveBuffer(sLocalisationDocument+ wChrono + "_doc6.pdf", bDocs)
  stWomPJ +=[" "] + sLocalisationDocument+ wChrono + "_doc6.pdf"
FIN

//création identifiant inscription
eIDInscription = Inscription.IDInscription

```

- Pour la table enfant :

```

POUR i = 1 À eNBEnreg
  HRAZ(Enfant)
  Enfant.IDInscription = eIDInscription
  Enfant.nom = stInscription.tabEnfant[i].nom
  Enfant.pnom = stInscription.tabEnfant[i].pnom
  Enfant.adresse = stInscription.tabEnfant[i].adresse
  Enfant.cp = stInscription.tabEnfant[i].adresse
  Enfant.ville = stInscription.tabEnfant[i].ville
  Enfant.datenaiss = stInscription.tabEnfant[i].datenaiss
  Enfant.lieunaiss = stInscription.tabEnfant[i].lieunaiss
  Enfant.sexe = stInscription.tabEnfant[i].sexe
  Enfant.ecole = stInscription.tabEnfant[i].ecole
  Enfant.bSansporc = stInscription.tabEnfant[i].bSansporc
  Enfant.bAllAlimentaire = stInscription.tabEnfant[i].bAllAlimentaire
  Enfant.bAllMedicament = stInscription.tabEnfant[i].bAllMedicament
  Enfant.bAllAutres = stInscription.tabEnfant[i].bAllAutres
  Enfant.libAllergie = stInscription.tabEnfant[i].libAllergie
  Enfant.nomMedecin = stInscription.tabEnfant[i].nomMedecin
  Enfant.telMedecin = stInscription.tabEnfant[i].telMedecin
  Enfant.nom_soussigne = stInscription.tabEnfant[i].nom_soussigne
  Enfant.autorisation[1] = stInscription.tabEnfant[i].autorisation1
  Enfant.autorisation[2] = stInscription.tabEnfant[i].autorisation2
  Enfant.autorisation[3] = stInscription.tabEnfant[i].autorisation3
  Enfant.autorisation[4] = stInscription.tabEnfant[i].autorisation4
  Enfant.autorisation[5] = stInscription.tabEnfant[i].autorisation5
  Enfant.autorisation[6] = stInscription.tabEnfant[i].autorisation6
  Enfant.autorisation[7] = stInscription.tabEnfant[i].autorisation7
  Enfant.autorisation[8] = stInscription.tabEnfant[i].autorisation8
  Enfant.autorisation[9] = stInscription.tabEnfant[i].autorisation9
  Enfant.date_signe = stInscription.tabEnfant[i].date_signe
  Enfant.lu_approuve = stInscription.tabEnfant[i].lu_approuve
  Enfant.niveau_scolaire = stInscription.tabEnfant[i].niveau_scolaire
  Enfant.blunette = stInscription.tabEnfant[i].blunette
  Enfant.blentille = stInscription.tabEnfant[i].blentille
  Enfant.bAD = stInscription.tabEnfant[i].bAD
  Enfant.bAutre = stInscription.tabEnfant[i].bAutre

```

```

POUR j = 1 À 3
    Enfant.identite_autre[j] = stInscription.tabEnfant[i].Identite_autre[j]
    Enfant.tel1_autre[j] = stInscription.tabEnfant[i].tel1_autre[j]
    Enfant.tel2_autre[j] = stInscription.tabEnfant[i].tel2_autre[j]
    Enfant.lien[j] = stInscription.tabEnfant[i].lien[j]
    Enfant.bUrgence[j] = stInscription.tabEnfant[i].bUrgence[j]
    Enfant.bPriseCharge[j] = stInscription.tabEnfant[i].bPriseCharge[j]
FIN

SI HAJoute(Enfant) ALORS
    //renvoyer ""
FIN

FIN

// Création de l'état
NomFichier = "Formulaire_inscription_" + eIDInscription + "_" + DateSys() + ".pdf" // PDF
NomFichierImpression = sLocalisationDocument + NomFichier
iDestination(iPDFGénérique, NomFichierImpression)
iImprimeEtat(ETAT_inscriptions,eIDInscription,eIDInscription)
sWNomPJ2 = sWNomPJ + "," + NomFichierImpression
xLog("SWNomPJ " + sWNomPJ)

wSujet = "Ville de Laon, service éducation-jeunesse - inscription scolaire et périscolaire"

// mail
xLog("param envoi mail wSujet " + wSujet)
xLog("param envoi mail sWTabDest " + sWTabDest)
xLog("param envoi mail sWNomPJ " + sWNomPJ)
xLog("param envoi mail sWTabCCi " + sWTabCCi)
xLog("param envoi mail wExpediteur " + wExpediteur)

sWMessageHTML += "<p><div>Ville de Laon, service éducation-jeunesse : Votre formulaire dûment rempli a été transmis à la Ville de Laon. Merci de conserver ce document qui pourra vous être demandé le cas échéant.</div>"
sWMessageHTML += "<div><br></div><div>Une fois votre dossier complet et validé vous recevrez l'attestation d'inscription.</div>"
sWMessageHTML += "<div><br></div><div>Les réservations aux différents services (restauration scolaire, services périscolaires et autres) vous seront envoyées par mail.</div></p>"

// Envoi du mail
Envoyer_Mail(wSujet,"", sWMessageHTML, sWTabDest, sWNomPJ2, sWTabCCi,wExpediteur)

```

- Suite de la récupération des données relatives aux enfants
- Création d'un nom de fichier pdf , construit à partir de l'identifiant d'inscription et de la date du jour.
- Création d'un document PDF à l'aide d'un modèle d'état (ETAT_Inscription) en y injectant les données récupérées.
- Préparation des éléments nécessaires à l'envoi de l'e-mail : définition du sujet, du contenu HTML, et ajout du PDF en pièce jointe.

```

// Suppression état
fSupprime(NomFichierImpression)

// Suppression des photos après envoi du mail
SI stInscription.Photo1 <> "" ALORS
    fSupprime(sLocalisationDocument+ wChrono + "_photo1.jpg")
FIN
SI stInscription.Photo2 <> "" ALORS
    fSupprime(sLocalisationDocument+ wChrono + "_photo2.jpg")
FIN
SI stInscription.Photo3 <> "" ALORS
    fSupprime(sLocalisationDocument+ wChrono + "_photo3.jpg")
FIN
SI stInscription.Photo4 <> "" ALORS
    fSupprime(sLocalisationDocument+ wChrono + "_photo4.jpg")
FIN
SI stInscription.Photo5 <> "" ALORS
    fSupprime(sLocalisationDocument+ wChrono + "_photo5.jpg")
FIN
SI stInscription.Photo6 <> "" ALORS
    fSupprime(sLocalisationDocument+ wChrono + "_photo6.jpg")
FIN

```

```

// Suppression des documents après envoi du mail
SI stInscription.Doc1 <> "" ALORS
  fSupprime(sLocalisationDocument+ wChrono + "_doc1.jpg")
FIN
SI stInscription.Doc2 <> "" ALORS
  fSupprime(sLocalisationDocument+ wChrono + "_doc2.jpg")
FIN
SI stInscription.Doc3 <> "" ALORS
  fSupprime(sLocalisationDocument+ wChrono + "_doc3.jpg")
FIN
SI stInscription.Doc4 <> "" ALORS
  fSupprime(sLocalisationDocument+ wChrono + "_doc4.jpg")
FIN
SI stInscription.Doc5 <> "" ALORS
  fSupprime(sLocalisationDocument+ wChrono + "_doc5.jpg")
FIN
SI stInscription.Doc6 <> "" ALORS
  fSupprime(sLocalisationDocument+ wChrono + "_doc6.jpg")
FIN

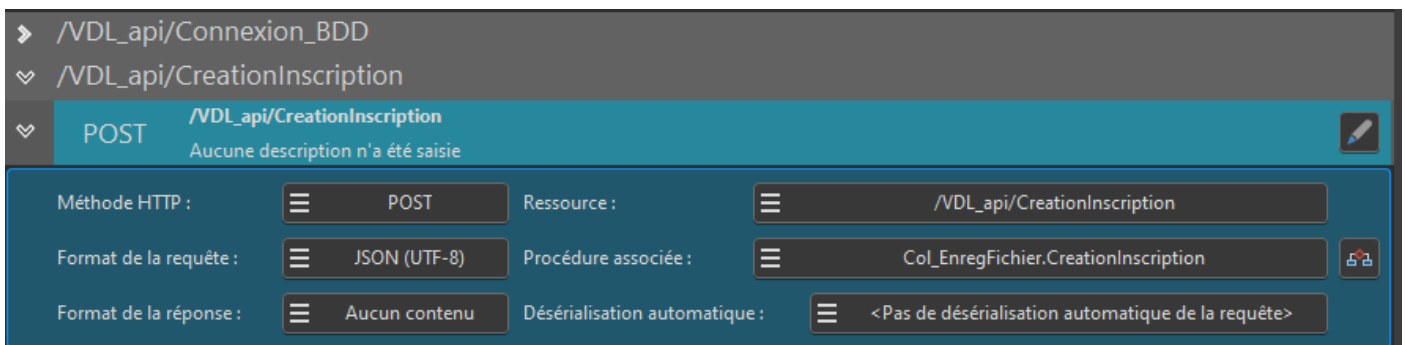
CAS ERREUR:
xLog("Nouvelle inscription:"+ErreurInfo(),Vrai)
REVOYER ""
CAS EXCEPTION:
xLog("CréationInscription :"+ExceptionInfo(),Vrai)
REVOYER ""

```

- Une fois l'email envoyé avec les documents en pièces jointes, ces derniers sont automatiquement supprimés du serveur afin d'éviter sa surcharge.

CREATION DE L'ENDPOINT DE L'API

Un endpoint a été mis en place pour permettre la création d'une inscription via un appel HTTP de type POST :



Ce point d'accès permet à l'application mobile de transmettre une demande d'inscription. Les données reçues sont ensuite traitées dans la procédure serveur [EnregFichier](#) présentée précédemment.

RECUPERATION DE LA LISTE DES ECOLES

➡ Création d'une nouvelle structure stEcole :

```
389
390  stEcole est une Structure
391      IDEcoles      est un entier sur 8 octets
392      Libellé       est une chaîne
393      Ordre         est un entier
394
395
396  FIN

Terminaison de COL_Structures
1
2
```

```
WL  Procédure globale __ListeEcole (serveur)

// Renvoie la liste des écoles
PROCÉDURE PRIVÉE __ListeEcole() : tableau de stEcole

    tabEcole est un tableau de stEcole
    stEcole est stEcole

    HLitPremier(Ecoles, IDEcoles)
    TANTQUE HTrouve(Ecoles)
        FichierVersTableau(stEcole, Ecoles)
        stEcole.IDEcoles = Ecoles.IDEcoles
        stEcole.Libellé = Ecoles.Libellé
        stEcole.Ordre = Ecoles.Ordre
        TableauAjoute(tabEcole, stEcole)
        HLitSuivant(Ecoles)
    FIN
    TableauTrie(tabEcole, ttMembre, "ordre")
    RENVOYER tabEcole

WL  Procédure globale ListeEcole (serveur)
PROCÉDURE ListeEcole() : tableau de stEcole
    RENVOYER __ListeEcole()
```

➡ Création de la procédure privée __ListeEcole() :

- Initialise `tabEcole` : un tableau de structure `stEcole`
- Utilisation de `HLitPremier` pour se positionner sur le 1^{er} enregistrement de la table `Ecoles`.
- Parcours de tous les enregistrements avec la boucle `TANTQUE HTrouve(Ecoles)` : pour chaque enregistrement, conversion des données de la table `Ecoles` vers la structure `stEcole`.
- Tri du tableau selon le champs `ordre`.
- Retourne un tableau `tabEcole` rempli et trié.

➡ Création de la procédure globale `ListeEcole()` :

- C'est la procédure publique qui peut être appelée depuis l'extérieur
- Retourne le résultat de `__ListeEcole`

➡ Création d'un endPoint pour permettre la récupération de la liste des écoles via un appel de type GET :

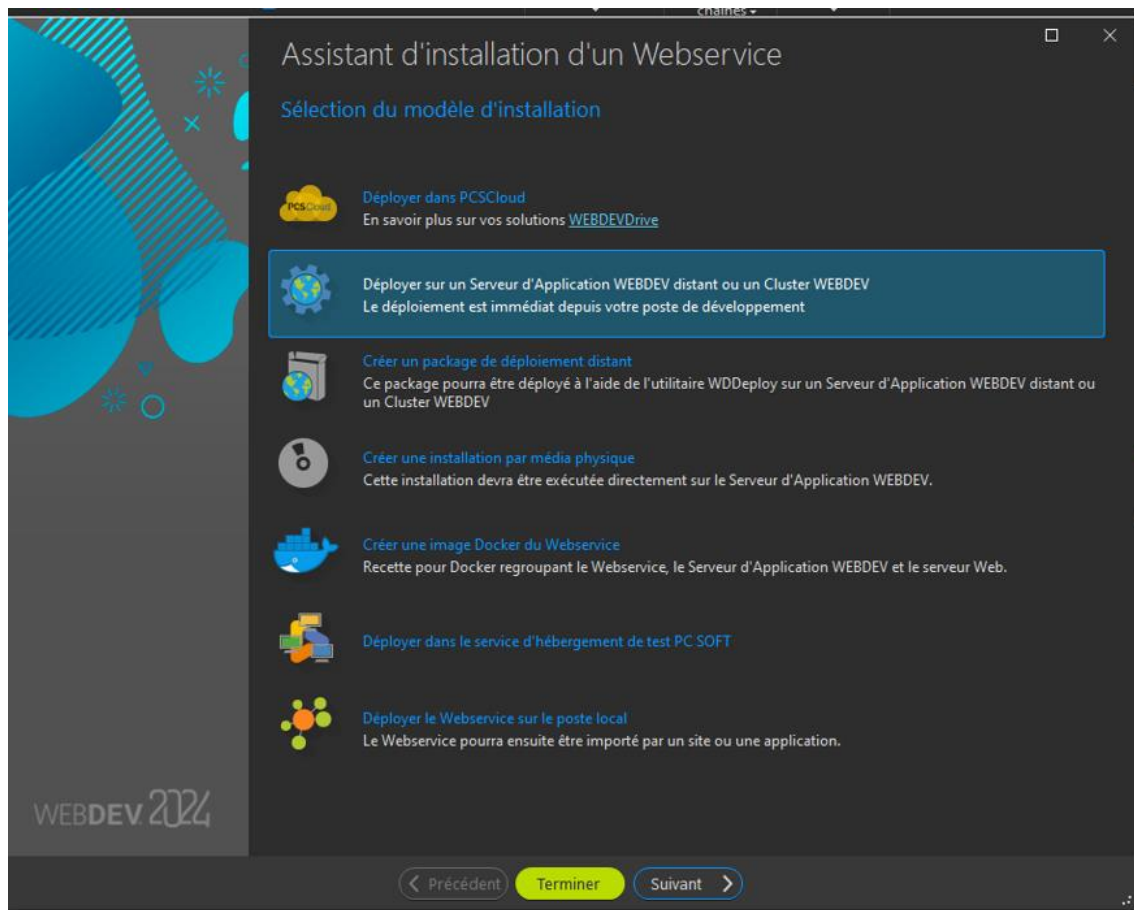
The screenshot shows the configuration for a new API endpoint. The URL is `/VDL_api/ListeEcole`. The HTTP method is set to `GET`. The request format is `JSON (UTF-8)`. The response format is `JSON (UTF-8)`. The associated procedure is `COL_LectureFichiers.ListeEcole`. A note indicates that no description has been entered.

Méthode HTTP :	GET	Ressource :	/VDL_api/ListeEcole
Format de la requête :	JSON (UTF-8)	Procédure associée :	COL_LectureFichiers.ListeEcole
Format de la réponse :	JSON (UTF-8)		

- Accessible via la méthode http GET à l'url `/VDL_api/ListeEcole`
- Associé à la procédure `Col_LectureFichiers.ListeEcole`

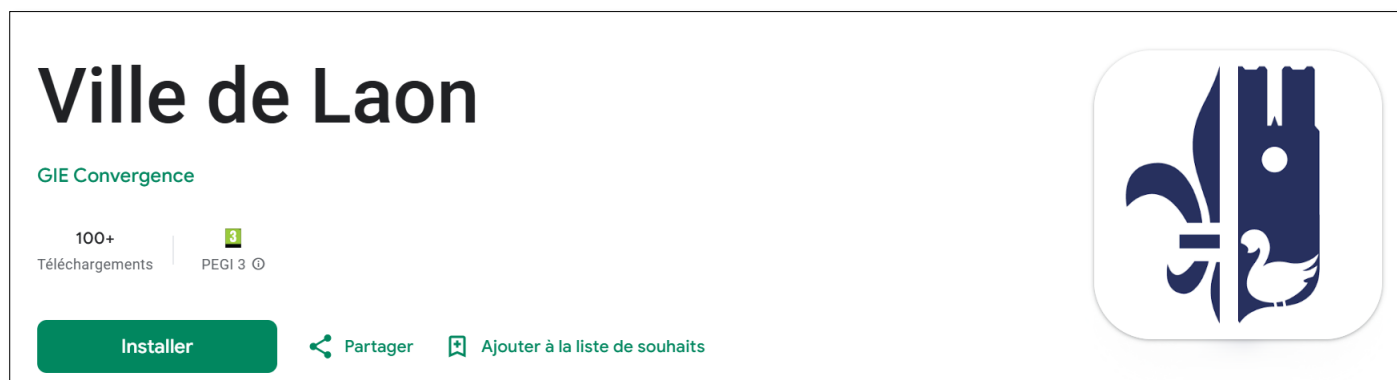
Ce point d'accès permet à l'application mobile de récupérer la liste complète des écoles au format JSON afin de remplir un menu déroulant (combo) permettant à l'utilisateur de sélectionner une école.

Après validation des modifications, déploiement de l'API pour enregistrer les changements :



LIENS

Après une période de tests, l'application à été définitivement publiée sur le Play Store :



Vous pouvez la télécharger sur ce lien :

<https://play.google.com/store/apps/details?id=fr.ville.laonV2&hl=fr>

La ville de Laon a également publié un article afin de mettre en avant les nouvelles fonctionnalités de l'application :

Actualités

Nouvelle application mobile Ville de Laon

L'application mobile Ville de Laon fait peau neuve. Si la présentation des widgets évolue, les fonctionnalités demeurent identiques pour vous informer des événements à venir, des jours et horaires des collectes de déchets, des menus de la cantine, des places de parkings disponibles, des offres d'emploi, des associations...

L'application est également très pratique pour accéder à l'espace famille (inscriptions scolaires et périscolaires...) mais aussi pour signaler une anomalie sur la voie publique et consulter son état d'avancement. Vous pouvez signaler par exemple des dégradations de mobilier urbain (aire de jeux, abris bus...), des dépôts sauvages, un nid de poule dans la chaussée, une fuite d'eau, un stationnement gênant...

L'application "Ville de Laon est disponible gratuitement sur app store et google store



Ce stage a été pour moi un véritable défi : je n'avais encore jamais travaillé avec ce langage, et j'ai été surprise de constater que la programmation se faisait en français. J'ai pris plaisir à apprendre de nouvelles notions et à me familiariser avec les particularités de ce langage.

J'ai également apprécié travailler de manière autonome, tout en pouvant compter sur l'aide et les conseils de ma tutrice lorsque j'en avais besoin. Cela a été particulièrement utile pour comprendre l'utilisation d'une API, une notion que je n'avais pas encore abordée en première année.

Etant restée en contact avec ma tutrice dans l'optique d'un futur stage, elle m'a informée du déploiement final de l'application, ce qui pour moi représente l'aboutissement du travail accompli durant ce stage.

Je suis contente du travail accompli et remercie encore une fois ma tutrice pour son accompagnement dans ce projet.