www.nout.fr
contact@nout.fr

Logiciels de gestion sur-mesure sans programmation

FORMAT XML-SIMAX

(dernière mise à jour le 11/02/11)

1. Qu'est-ce le format XML-SIMAX ?

C'est le format utilisé dans les requêtes et réponses **SOAP** à SIMAXService, c'est à dire pour l'échange de données entre SIMAXService et les clients SIMAXService.

En effet, pour interroger le service web SIMAXService, vous devez vous référer à la documentation SIMAXService qui décrit toutes les opérations disponibles mais aussi à ce document qui décrit le format XML de la représentation des données SIMAX.

Pour une meilleure compréhension de ce document, il est nécessaire de connaître les bases de la forme de Backus-Naur. Cette forme souvent abrégée en BNF, de l'anglais Backus-Naur Form, est une notation permettant de décrire les règles syntaxiques des langages de programmation. C'est donc un métalangage. La syntaxe d'un fichier XML-SIMAX sera décrite grâce à cette notation.

(voir la rfc 5234)

Voici les règles de base à maîtriser :

* BNF consiste à déclarer des règles syntaxiques :

```
règle = ...
```

* Les règles se concatènent en listant une séquence de nom de règles :

```
règle = regle1 regle2
```

* caractère '/' pour décrire une alternative entre deux règles :

```
regle3 = regle1 / règle
```

* caractère '*' pour préciser qu'une séquence peut apparaître n fois :

```
regle4 = *(regle3)
```

* les parenthèses permettent d'appliquer un nombre d'occurrence à toute une portion syntaxique :

```
règle = *(regle1 regle2)
```

* les portions de syntaxe qui sont optionnelles sont entre crochets :

```
regle5 = regle2 [regle4]
```

* les caractères sont représentés par leur valeur hexadécimale précédée du caractère '%' : %x30

* Des règles de bases sont prédéfinies, comme par exemple :

```
ALPHA   =  %x41-5A / %x61-7A   ; A-Z / a-z
DIGIT   =  %x30-39             ; 0-9
VCHAR   =  %x21-7E             ; caractères imprimables
```

En ce qui concerne les normes d'écriture, nous avons choisi de mettre :

- les attributs en minuscule et des majuscules au début de chaque mot
- le nom des balises commencent par des majuscules, le reste en minuscule sauf au début des mots
- les espaces de noms sont en minuscule avec des tirets pour séparer les mots

2. Document XML-SIMAX

Le fichier XML doit avoir une entête de la forme :

```
[1] S            =    *(%x20 / %x9 / %xD / %xA)                ; espaces
[2] Eq           =     [S] '=' [S]                              ; =
[3] SimaxDecl    =    '<?simax' VersionInfo *S '?>'             ; <?simax version="1.1"?>
[4] VersionNum       = '1.1'
[5] VersionNum       = '1'
[6] VersionInfo      = *S 'version' Eq ("'" VersionNum "'" / '"' VersionNum '"')        ; version="1.1"
[7] VersionSchemaInfo = *S 'VERSION' Eq ("'" VersionNum "'" / '"' VersionNum '"')     ; VERSION="1"
[8] VersionLectEcrit  = *S 'VERSION_LECTECRIT' Eq ("'" VersionNum "'" / '"' VersionNum '")
  ; VERSION_LECTECRIT="1"
[9] VersionLecture    = *S 'VERSION_LECTURE' Eq ("'" VersionNum "'" / '"' VersionNum '")
  ; VERSION_LECTURE="1"
[10] DateHeure       = *S 'DATEHEURE' Eq *(DIGIT)             ; DATEHEURE="2008052215213894"
[11] SchemaDecl      = '<schema' VersionSchemaInfo VersionLectEcrit VersionLecture DateHeure
*S '?>'
[12.1] entete = SimaxDecl CLRF SchemaDecl
[12.2] fin    = "</schema>"
```

Et un contenu de la forme

```
[13] contenu = *(element)
[14] element = (enreg / dataFichier / dataTexteLong) CLRF
[15] document = entete contenu fin
```

Exemple :

```
<?simax version="1.1"?>
<schema          VERSION="1"          VERSION_LECTECRIT="1"          VERSION_LECTURE="1"
DATEHEURE="2008052215213894">
<client id="456745" title="Dupont Pierre">
  ...
</client>
<adresse id = "434376" title = "rue des cerises, Montpellier">
  ...
</adresse>
...
```

3.Les balises XML-SIMAX

Le format XML-SIMAX est un format basé sur le standard XML du W3C (www.w3c.org). Il doit être bien formé, c'est à dire respecter les spécifications XML.
En plus de ces spécifications, le nom des balises doit être en minuscule.

1. Readable de OptionDialogue

Deux formats sont proposés pour représenter les colonnes:

- **Lisible** (libellé des formulaires et des colonnes du paramétrage SIMAX)
- **Illisible** (identifiant des formulaires et des colonnes de SIMAX dans la base de données)

1. DisplayValue de OptionDialogue

Deux formats sont proposés pour représenter les données:

- **Valeur affichée**
- **Valeur stockée** (comme stockée dans la base de données)

En valeur affichée, les éléments de tableau sont représentés par leur mini-description à la place de leur identifiant, même chose pour les listes déroulantes. En ce qui concerne les autres types (date, heure, ...) la valeur est mise en forme pour l'utilisateur.

Syntaxe

- balise :
  - le libellé en minuscule et respectant les normes XML (lisible)
  - ou l'identifiant de la colonne préfixé par "id_" (illisible)

```
[16] MINUS  =  %x61-7A                                  ; minuscule a-z
[17] DIGIT  =  %x30-39                                  ; chiffre 0-9
[18] VCHAR  =  %x21-7E                                  ; caractères visibles
[19] balise =  MINUS *(MINUS / DIGIT / "_") / "id_" *(DIGIT)   ; libelle ou identifiant
[20] baliseFin = '</' balise '>'
[21] valeur    =  *(S / VCHAR)                          ; chaîne de caractères
[22] bool      = '0'/'1'                                ; vrai/faux
```

Exemple:

- Lisible

<commande>12456</commande>

- Illisible

<id_3645536>12456</id_3645536>

remarque : 3645536 est alors l'identifiant du formulaire commande

4. Représentation d'un enregistrement

Un enregistrement se compose comme ceci :

- balise = nom du tableau ou id_(identifiant du tableau)
    - L'attribut **id** précise l'identifiant de l'enregistrement (clé primaire)
    - L'attribut **title** donne une description de l'enregistrement à partir des colonnes « repris dans l'intitulé »
    - Les autres attributs disponibles sont des attributs de mises en forme et de contrôle d'état de champ (voir les chapitres correspondants)
- valeur = liste des colonnes du tableau

Syntaxe

```
[23] id = 'id' eq "" *(DIGIT) ""
[24] title = 'title' eq "" valeur ""
[25] attributEnreg =  id title [attributMiseEnForme] [attributControleEtatChamp]
[26] enreg = '<' balise attributEnreg ( '>' listeColonne baliseFin / '/>' )
[27] listeColonne = *( colClassique / colTableau / liste / colFichier / colTexteLong / colCombo)
```

Exemple

```
<commande id="777" title="description">
  <ref>Valeur</ref>
  ...
</commande>
```

5.Représentation des colonnes (par type)

1. Classique

Cette partie concerne les champs de type texte, date, entier, etc ...

Syntaxe

- balise = colonne
    - attributs de la balise
- valeur de la balise = valeur de la colonne

[28] colClassique = '<' balise listeAttribut '/>' valeur baliseFin

Exemple : Un client avec des champs texte nom et prénom

```
<client id="777" title="Dupont Pierre">
  <nom>Dupont </nom>
  <prenom>Pierre</prenom>
</client>
```

2. Elément d'un tableau

Syntaxe

- balise = colonne
- valeur de la balise = ID de l'enregistrement représenté par cette colonne.

La description de l'enregistrement est placée en fin du document XML :

- balise = nom du tableau/formulaire
    - attribut id : identifiant de l'enregistrement
    - attribut title : mini description de l'enregistrement

[29] colTableau = '<' balise '>' *(DIGIT) baliseFin [30] descElementTableau = '<' balise attributBalise '/>'

Exemple :Une commande avec un champ 'client' de type 'Elément d'un tableau'

```
<commande id="777" title="description">
  <client_commande>123</client_commande>
</commande>
```

et à la fin du document XML

```
<client id="123"   title= "Mon client"/>
```

3. Liste

Syntaxe

- balise = colonne
    - sous-balise = (nom du tableau lié)__Liste
    - valeur sous-balise = ID de l'enregistrement (élément de la liste)

[31] liste = '<' balise listeAttribut '>' CLRF *(colTableau CLRF) baliseFin

Exemple

Une commande contient une liste de ligne de commande. Une ligne de commande contient un article qui est une colonne de type 'Elément d'un tableau', on retrouve donc aussi le cas précédent.

- balise = <liste_ligne_commande>
- sous-balise = <ligne_commande__liste>
- valeur sous-balise = 1234 par exemple

=> Description de la commande

```
<commande id="777" title="CMD_23424 M.Dupont">
  <ref>CMD_23424</ref>
  <client>123</client>
  <liste_ligne_commande>
    <ligne_commande__liste>1234</ligne_commande__liste>
    <ligne_commande__liste>5678</ligne_commande__liste>
    ...
  </liste_ligne_commande>
</commande>
```

=> description (à la fin du XML) des éléments de tableau présents dans la commande

```
<client id="123" title="Dupont Pierre"/>
<ligne_commande id="1234" title="ligne 1"/>
<ligne_commande id="5678" title="ligne 2"/>
```

4. Fichier et Texte long

Syntaxe

- balise = libellé de la colonne
    - attribut ref = ID pour retrouver la représentation binaire de la donnée
- valeur de la balise = chemin du fichier par exemple

[32] ref = 'ref' eq *(DIGIT) [33] colFichier = '<' balise ref '>' valeur baliseFin Les champs texte long sont gérés comme des fichiers si ils dépassent la taille max autorisée :

[34] colTexteLong = (colClassique / colFichier)

La représentation binaire de la donnée est à la fin du document XML.

- balise = data
    - attribut ref = ID pour retrouver la représentation binaire de la donnée
    - attribut title = description de la donnée
    - attribut encoding = type d'encodage
    - attribut size = taille de la donnée non encodée
    - attribut filename = nom du fichier contenant la donnée
    - attribut typemime = type de fichier
- valeur de la balise = représentation binaire de la donnée

```
[35] filename = 'filename 'eq valeur
[36] typemime = 'typemime' eq valeur
[37] size     = 'size' eq *(DIGIT)
[38] encoding = 'encoding' eq valeur
[39] attributFichier   =  ref filename title typemime size encoding
[40] attributTexteLong = ref title typemime size encoding
[41] data      = 'data'

[45] dataFichier   = '<' data attributFichier '>'  valeur baliseFin
[46] dataTexteLong = '<' data attributTexteLong '>'  valeur baliseFin
```

Exemple

- Document WORD

```
<cv ref="1">curriculum vitae de M.Dupont</cv>

<data ref = "1"
    title = "curriculum vitae de M.Dupont"
    encoding = "base64"
    size = "3087"
    filename = "dupont_cv.doc"
    typemime = "application/msword">
  ... binaire encodé ...
</data>
```

ou si le document n'a pas été trouvé

```
<cv ref="1">curriculum vitae de M.Dupont</cv>

<data ref = "1"
    title = "curriculum vitae de M.Dupont"
```

encoding = "base64"
size = "3087"
filename = "dupont_cv.doc"
typemime = "application/msword"/>

ou si le champ est vide
<cv/>

- Texte Long dépassant la taille Max autorisée

<description ref ="2"/>

<data ref = "2"
encoding = "QuotedPrintable"
size = "2108"
typemime = "text/plain">
... binaire encodé ...
</data>

- Texte Long de taille inférieure à la taille Max autorisée

<description/>
ou
<description>contenu du texte long</description>

5. Liste déroulante

Syntaxe

- balise : nom de la colonne
- valeur de la balise = id du choix dans la liste déroulante (valeur stockée)
- ou libellé du choix (valeur affichée)

[47] colCombo = '<' balise listeAttribut '>' *(DIGIT) baliseFin

Le libellé du choix correspondant à l'identifiant est à recherché dans le XSD.

Exemple
Un rendez-vous possède un état de disponibilité.

<rendez-vous id="756" title="reunion à 17h">
<date>12/05/08</date>
<disponibilite>1</disponibilite>
</rendez-vous >

<disponibilite id="1" title="Absent"/>

6. Colonnes particulières

- Les colonnes de type **séparateur** sont présentes dans le XML sous forme d'une balise portant le nom du séparateur et contenant une balise par colonne contenues dans le séparateur.

- Les colonnes de type **invalide** sont gérées comme n'importe quelle autre colonne

- Les colonnes **sans libellé** sont présente dans le XML au format id_(identifiant de la colonne)

- Les colonnes **cachées** dans le XML d'un enregistrement sont gérées comme n'importe quelle autre colonne et l'information « hidden » est présente dans le XSD.
  Les colonnes cachées dans le XML d'une liste d'enregistrements ne sont présentes ni dans le XML, ni dans le XSD.

- Les colonnes de type **calcul**
  Les valeurs de ces colonnes sont envoyées par SIMAXService. Par contre, le client SIMAXService ne doit renvoyer les valeurs de ces colonnes car ce sont des colonnes mises à jour par le service, il est impossible de les modifier. Les calculs mis à jour seront d'ailleurs renvoyées par le service après un Update.

- Les colonnes de type **bouton** n'apparaissent pas dans le XML (uniquement dans le XSD)

- Les colonnes de type **date/heure et dateheure** sont demandées à l'échelle UTC.
  Les dates doivent être de la forme "AAAAMMJJ" avec :
  - AAAA indique l'année
  - MM indique le mois
  - JJ indique le jour

  Les heures doivent être de la forme "hhmmss" avec :
  - hh indique les heures
  - mm indique les minutes
  - ss indique les secondes

  Les dateheure sont la concaténation du format date et heure.

6. Représentation d'une liste d'enregistrement

1. Liste simple

Syntaxe

. balise = id_(identifiant du formulaire) [Readable = 0] ou
(nom du formulaire)__Liste [Readable = 1]

[48] liste = '<' nom du formulaire '> *(enreg) '</'(nom du formulaire)'>'

Contrairement aux listes en tant que colonne où seul l'identifiant et la mini-description de chaque enregistrement sont donnés dans le XML, ici chaque enregistrement de la liste est représenté comme un enregistrement isolé, c'est à dire avec ses colonnes. Les colonnes présentes dans le XML-Liste sont toutes les colonnes autres que « détails » et « bouton ».

Exemple : 11654 est l'identifiant du formulaire listé

```
<id_11654 simax:id="47572362849623" simax:title="Général">
<id_11656>Général</id_11656>
<id_11657>
<id_11579>1</id_11579>
<id_11579>2</id_11579>
<id_11579>3</id_11579>
</id_11657>
</id_11654>

<id_11654 simax:id="46732686449104" simax:title="Article">
<id_11656>Article</id_11656>
<id_11657>
<id_11579>1</id_11579>
<id_11579>2</id_11579>
<id_11579>3</id_11579>
<id_11579>4</id_11579>
<id_11579>5</id_11579>
</id_11657>
</id_11654>
```

Remarque importante :

Quand vous demandez la liste des collaborateurs à SIMAXService, celui-ci vous renverra bien une liste de collaborateur. Or le formulaire 'collaborateur' peut se dériver en 'commercial' et 'client particulier' par exemple. Donc votre liste contiendra peut être des commerciaux.
Cependant, la liste ne contiendra que des balises 'collaborateur' avec les champs du formulaire 'collaborateur' uniquement. Pour avoir les informations d'un commercial, il faudra faire la demande explicitement.

2. Liste avec rupture

Les listes d'enregistrements peuvent contenir des lignes qui ne sont pas des enregistrements mais des ruptures. Les ruptures sont des calculs intermédiaires sur les colonnes et qui dépendent d'une colonne de tri.

Pour recevoir les ruptures, vous devenez le demander au service en indiquant ceci dans la balise <SpecialParamList> :

<**WithBreakRow**>1<**WithBreakRow**>

Les lignes qui correspondent à des lignes de ruptures ont l'attribut **type="breakRow "**

Exemple :

Si vous avez une rupture sur la colonne "service" de la vue "poste collaborateur". Quand vous allez afficher les postes collaborateurs et que vous allez trier sur la colonne "service", des calculs sur les autres colonnes vont venir s'insérer à chaque valeur différente de la colonne "service". Vous pourrez alors savoir combien de collaborateurs appartiennent au service "administration" ou quel est le salaire moyen des personnes du service "technique", etc …

```
<poste_collaborateur>
  <collaborateur>1234</collaborateur>
  <collaborateur>1452</collaborateur>
  <collaborateur>5678</collaborateur>
    ...
</poste_collaborateur>

<collaborateur id="1234" title="ligne 1">
  <calculs></calculs>
  <quantite>2</quantite>
  <prix>10</prix>
</collaborateur>

<collaborateur id="1452" title="ligne 2">
  <calculs></calculs>
  <quantite>4</quantite>
  <prix>30</prix>
</collaborateur>

<collaborateur id="5678" simax:type="breakRow">
  <calculs>moyenne</calculs>
  <quantite>3</quantite>
  <prix>20</prix>
</collaborateur>
```

Remarque :
La colonne 'calculs' a été ajoutée par SIMAX pour contenir les intitulés des calculs. Si une colonne du type 'type de l'enregistrement' existe, c'est elle qu'on utilisera pour renseigner les intitulés des calculs.

7. Calculs automatiques

Ces calculs automatiques sont présents uniquement sur demande, c'est à dire si vous ajoutez ceci dans le header SpecialParamList de vos requêtes :

<**WithEndCalculation**>1<**WithEndCalculation**>

1. Calculs de fin de colonne

Syntaxe

- une balise <simax:**columnCalculus**>
  - une sous-balise pour chaque opération
    - Chaque opération contient une sous-balise par colonne calculable

Remarque :

les calculs de fin de colonne sont des calculs automatiques fait par SIMAX pour les listes, tableaux croisés ou vues. La balise <simax:columnCalculus> est donc présente après la liste des enregistrements.

Attention, rien dans le XSD ne précise qu'il y a des calculs de fin de colonne. Le client SIMAXService est chargé de regarder si il y en a en recherchant la balise <simax:columnCalculus>.

Les différents calculs automatiques :

- sum : somme
- max : maximum
- mini : minimum
- average : moyenne
- count : compteur

Exemple : Ici on reçoit le résultat des calculs somme, moyenne, min et max sur les colonnes quantité et total HT

```
<simax:columnCalculus>
  <somme>
    <quantite>...</quantite>
    <total_ht>...</total_ht>
  </somme>
  <moyenne>
    <quantite>...</quantite>
    <total_ht>...</total_ht>
  </moyenne>
  <maximum>
    <quantite>...</quantite>
```

```
      <total_ht>...</total_ht>
    </maximum>
    <minimum>
      <quantite>...</quantite>
      <total_ht>...</total_ht>
    </minimum>
 </simax:columnCalculus>
```

2. calculs de fin de ligne

Contrairement aux calculs de fin de colonne, les calculs de fin de ligne apparaissent dans le XSD car au lieu de rajouter des lignes de type calcul, ces calculs rajoutent des colonnes de type calcul.

8. Mise en forme des lignes et des colonnes

La mise en forme des enregistrements et des colonnes est décrite grâce à des attributs appartenant au namespace **simax-layout**.

- **color** : string (6 chiffres du code couleur Hexa) pour la couleur du texte
- **bgcolor** : string (6 chiffres du code couleur Hexa) pour la couleur du fond
- **bold** : boolean (par défaut 0)
- **italic** : boolean (par défaut 0)

Les couleurs sont stockées dans SIMAX de la manière suivante :

- rouge poids faible
- vert milieu
- bleu poids fort

=> BBGGRR dans l'entier

Exemple : Liste dysfonctionnement mise en forme sur enregistrement et colonne

```
<dysfonctionnement id="346189452458301" title="probleme 1 de M. Dupont"
                simax-layout:bold="1" simax-layout:color="FF0000">
  <libelle>problème 1</libelle>
  <cree_par> M.Dupont </cree_par>
  <corrige>0</corrige>
</dysfonctionnement>
<dysfonctionnement simax-layout:bgcolor="CCFF66">
  <libelle>probleme 2</libelle>
  <cree_par> M.Marie </cree_par>
  <corrige>0</corrige>
</dysfonctionnement>
<dysfonctionnement simax-layout:italic="1" simax-layout:color="0000FF">
  <libelle>probleme 3</libelle>
  <cree_par simax-layout:italic="1"
          simax-layout:color="00FFFF"> Me.bidule </cree_par>
  <corrige>0</corrige>
</dysfonctionnement>
<dysfonctionnement simax-layout:bold="1" simax-layout:color="FF0000">
  <libelle>problème 1</libelle>
  <cree_par> M.Dupont </cree_par>
  <corrige>0</corrige>
</dysfonctionnement>
```

9. Contrôles d'état de champ

1. Fonctionnement

Les informations concernant les colonnes/champs en lecture seule, invisible, etc … sont présentes dans le XML uniquement si des contrôles d'état de champs s'appliquent sur l'enregistrement. Dans le reste des cas, c'est bien le XSD qui contient ces informations.

En effet, les contrôles d'état de champs sont appliqués quand leurs conditions sont remplies, les informations peuvent donc être amenées à changer en fonction des données. Dans le cas où il n'y a pas de contrôle d'état de champs, ces informations restent les mêmes et font partie de la structure de l'enregistrement, d'où leur présence dans le XSD.

Pour demander explicitement ces contrôles d'état de champs à SIMAXService, vous devez ajouter dans le paramètre « OptionDialogue » la balise suivante :

<**WithFieldStateControl**>1<**WithFieldStateControl**>

En conclusion, ce sont les informations contenues dans le XML qui sont prioritaires.

2. Liste des contrôles d'état de champs

- Invisible :
  - **hidden** = « 1 »
  - le champ ne doit pas être visible par l'utilisateur
- Grisé :
  - **disabled** = « 1 »
  - le champ est visible, non modifiable et contient une information à ne pas prendre en compte
- Lecture seule :
  - **readOnly** = « 1 »
  - le champ est visible, non modifiable
- Modifiable
  - (par défaut)

10. Exemple : Collaborateur

=> Résultat de la requête « modifier collaborateur » avec Readable = 1 et DisplayValue=0

<collaborateur simax:id="330224175953766" simax:title="Paul Emile">

=> Séparateur contenant une liste de colonnes
<general>
=> Colonne de type entier
<immatriculation>45</immatriculation>

=> Colonne de type élément d'un tableau
<utilisateur>18</utilisateur>

=> Colonne de type liste déroulante
<civilite>2375</civilite>

=> Colonne de type texte
<nom>Emile</nom>
<prenom>Paul</prenom>
<numero_de_telephone>0467.....</numero_de_telephone>
<adresse_email>paul@hebergeur.fr</adresse_email>
<adresse>224 rue des cerises</adresse>
<code_postal>34000</code_postal>

=> Colonne de type élément d'un tableau
<ville>634365090800938</ville>

=> Colonne de type date
<date_d_embauche>26/12/2010</date_d_embauche>

=> Colonne calcul
<anciennete__ans>0</anciennete__ans>

</general>

=> Séparateur
<fonction>
=> Colonnes contenues dans le séparateur
<poste>872447</poste>
<service>2572402</service>

=> Colonne de type liste

```
<equipe>
    <equipe__liste>1</equipe__liste>
    <equipe__liste>2</equipe__liste>
    <equipe__liste>3</equipe__liste>
</equipe>
</fonction>
```

=> Colonne vide

```
<observations></observations>
```

```
</collaborateur>
```

=> Description des éléments référencés dans le XML

```
<utilisateur simax:id="18" simax:title="Paul Emile"/>

<ville simax:id="634365090800938" simax:title="Montpellier"/>

<equipe__liste simax:id="1" simax:title="Paramétrage"/>
<equipe__liste simax:id="1=2" simax:title="Support"/>
<equipe__liste simax:id="3" simax:title="Administratif"/>
```

11. Exemple : Liste facture

=> Résultat de la requête «liste facture » avec Readable = 1 et DisplayValue=0

=> 2 factures

```
<facture__liste simax:id="5272424" simax:title="FC85648548 Dossier1">
        <type>25</type>
        <reference>FC85648548</reference>
        <date>31/01/2010</date>
        <projet>3877547252</projet>
        <client>2574863653</client>
        <pays_d_adresse>139728866165270</pays_d_adresse>
        <total_ht>225,50 €</total_ht>
        <total_ttc>269,70 €</total_ttc>
        <acompte/>
        <reste_a_payer>269,70 €</reste_a_payer>
        <date_de_reglement/>
        <a_payer_le>01/03/2010</a_payer_le>
        <suivi>
        <imprimee>0</imprimee>
        <payee>0</payee>
        </suivi>
</facture__liste>

<facture__liste simax:id="238573543" simax:title="FC8743274 Dossier2">
        <type>75</type>
        <reference>FC8743274</reference>
        <date>23/01/2010</date>
        <projet>3687</projet>
        <client>38735454</client>
        <pays_d_adresse>139728866165270</pays_d_adresse>
        <total_ht>289,50 €</total_ht>
        <total_ttc> 310,00 €</total_ttc>
        <acompte/>
        <reste_a_payer>200,00 €</reste_a_payer>
        <date_de_reglement/>
        <a_payer_le>10/04/2010</a_payer_le>
        <suivi>
        <imprimee>0</imprimee>
        <payee>0</payee>
        </suivi>
</facture__liste>
```

=> **Pour colonne « Type »**

<formulaire simax:id="25" simax:title="Facture finale"/>
<formulaire simax:id="75" simax:title="Facture client"/>

=> **Pour Colonne « Projet »**

<projet_d_integration simax:id="3877547252" simax:title="Projet 1"/>
<projet_d_integration simax:id="3687" simax:title="Projet 2"/>

=> **Pour Colonne Client**

<client simax:id="2574863653" simax:title="Client 1"/>
<client simax:id="38735454" simax:title="Client 2"/>

=> **Pour colonne « Pays d'adresse »**

<pays simax:id="139728866165270" simax:title="France"/>