|  |
| --- |
| Titre du document |
| DATA2ODOO  Documentation technique |
| Référence du document |
| Data2ODOO\_Documentation technique.docx |



# Historique

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Version* | *Date* | *Description des modifications* | *Rédigé par* | *Diffusé* |
| 1.0 | 20/09/2020 | Version initiale | Chef Projet Interway |  |
|  |  |  |  |  |
|  |  |  |  |  |

Sommaire

[Historique 1](#_Toc51511787)

[Objectifs 3](#_Toc51511788)

[principe de la solution Data2ODOO 3](#_Toc51511789)

[Environnement de developpement et de tests 3](#_Toc51511790)

[Périmètre fonctionnel de la solution 4](#_Toc51511791)

[donnees - source / DESTINATION / correspondance 5](#_Toc51511792)

[Source de données 5](#_Toc51511793)

[Destination des données 5](#_Toc51511794)

[correspondance source ↔ destination 6](#_Toc51511795)

[PROCESSUS DE TRAITEMENT 7](#_Toc51511796)

[Etape 1 - lecture du fichier source 7](#_Toc51511797)

[Etape 2 - formatage des données 7](#_Toc51511798)

[Etape 3 - nettoyage des données 7](#_Toc51511799)

[Etape 4 - vérification des données et identification des anomalies 7](#_Toc51511800)

[Etape 5 - Insertion des données et affichage du rapport 8](#_Toc51511801)

[Etape 6 - affichage du rapport de reprise des données 8](#_Toc51511802)

[architecture applicative 9](#_Toc51511803)

[Organisation de la solution 9](#_Toc51511804)

[Projet GUI\_DATA2ODOO - Interface graphique 10](#_Toc51511805)

[Projet DAO\_DATA2ODOO - Acces aux données 11](#_Toc51511806)

[Projet MODEL\_DATA2ODOO - Classe métier 12](#_Toc51511807)

[Projet PROCESS\_DATA2ODOO - Processus traitement données client 12](#_Toc51511808)

# Objectifs

Développement d’un utilitaire de reprise des données clients dans l’ERP ODOO.

## principe de la solution Data2ODOO



**Script C#**

**Interface graphique** (WinForms)



PostgreSQL

**Fichier Client**

* Nettoyage et transformation des données (élimination des doublons, formatage,...)
* Détection des anomalies (adresses mail invalides par exemple)
* Insertion dans la base PostgreSQL de l’ERP ODOO
* Envoi d’un rapport d’importation par email (le rapport indique les éléments d’informations suivants *: nombre de lignes traitées, nombre de lignes insérées, nombre de doublons éliminés, nombre d’adresses mail invalides...*)

**ERP ODOO**

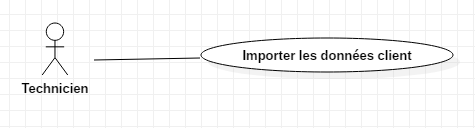
L’interface graphique doit permettre de définir   
les paramètres de la reprise des données (*source  
et destination - adresse du serveur, nom de la base -*)  
et de déclencher l’exécution du script

## Environnement de developpement et de tests

|  |  |
| --- | --- |
| Environnement de développement | **Visual Studio** |
| Gestion de versions | **Git** (bash, client GUI, extension EDI) / **GitHub** |
| ERP (Entreprise Resource Planning) | Serveur **ODOO** (version 9.6)  *Modules applicatifs  : Gestion des ventes* |
| Base de données | SGBD **PostgreSQL**  Interface d’administration **PgAdmin** |

# Périmètre fonctionnel de la solution

## Diagramme des cas d’utilisation



## Description textuelle du cas d’utilisation

|  |
| --- |
| **Cas d’utilisation : Importer les données client** |
| Acteur déclencheur : Technicien |
| Pré-conditions : Le fichier Excel a été converti au format CSV |
| Post-conditions : Les données client valides sont insérées dans ODOO |
| Scénario nominal :   1. Le système présente le formulaire 2. Le technicien sélectionne le fichier CSV contenant les données client, saisit les paramètres (adresse du serveur, nom de la base de données cible) et valide 3. Le système insère les données valides en BDD et affiche le rapport de la reprise des données |
| Scénario alternatif (extensions / exceptions) :  3a. Les paramètres saisis sont insuffisants (champs non renseignés)  Le système informe l’utilisateur. Retour à 2.  3b. Les paramètres saisis sont erronés  Le système informe l’utilisateur. Retour à 2. |

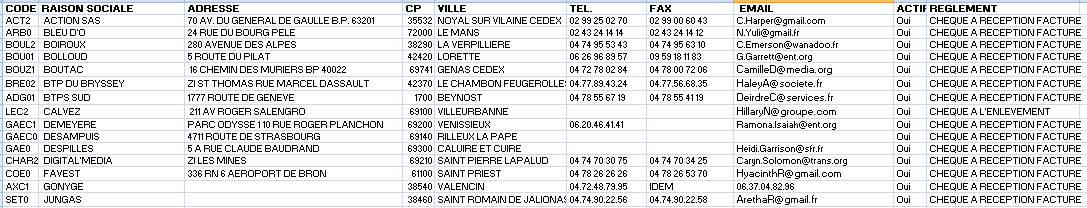
# donnees - source / DESTINATION / correspondance

## Source de données

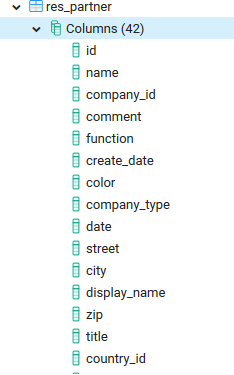
Les données des clients sont stockées dans un fichier EXCEL. Chaque client possède un certain nombre de caractéristiques :

* **CODE** : il s’agit du code utilisé par la société NEGOMAT pour identifier ses clients
* **RAISON** **SOCIALE** : nom de l’entreprise cliente
* **ADRESSE** : nom de la rue
* **CP** : code postal
* **VILLE** : ville
* **TEL** : numéro de téléphone
* **FAX** : numéro de fax
* **MAIL** : adresse mail du client
* **ACTIF** : indique si OUI ou NON un client est référencé
* **REGLEMENT** : indique le mode de règlement appliqué au client

Extrait du fichier **Negomat\_client.xls**



## Destination des données

Les données des clients seront insérées dans la table **Res\_Partner** de la base PostgreSQL de l’ERP ODOO.

*Extrait du schéma de la table Res\_Partner* ⏵

## correspondance source ↔ destination

|  |  |  |  |
| --- | --- | --- | --- |
| **Source** | **Destination** | | |
| *Colonne fichier  EXCEL* | *Colonne table Res\_Partner* | *Type colonne* | *Remarques* |
| CODE |  |  |  |
| RAISON SOCIALE | name | character varying |  |
| display\_name | character varying |  |
| ADRESSE | street | character varying |  |
| CP | zip | character varying |  |
| VILLE | city | character varying |  |
| TEL | phone / mobile | character varying | *phone renseigné si TEL commence par 01, 02, 03, 04 ou 05* |
| FAX |  |  |  |
| MAIL | email | character varying |  |
| ACTIF | active | boolean |  |
| REGLEMENT |  |  |  |
|  | country\_id | integer | 75 pour France |
|  | customer | boolean |  |
|  | is\_company | boolean |  |
|  | create\_date | timestamp |  |

# PROCESSUS DE TRAITEMENT

Le processus de reprise des données Clients de EXCEL vers ODOO défini est le suivant :

## Etape 1 - lecture du fichier source

*Pré-requis* : Le fichier Negomat\_client.xls est enregistré au format CSV

La lecture séquentielle du fichier CSV permet de générer une liste d’objets Client.

## Etape 2 - formatage des données

Le formatage des données consiste à éliminer certains caractères et à opérer différentes transformations sur les champs.

**Règles de formatage :**

|  |  |
| --- | --- |
| ***Champ*** | ***Formatage appliqué*** |
| *Tous les champs* | * transformation en majuscules * élimination des espaces à gauche et à droite |
| *Code* | * élimination des caractères ‘ |
| *RaisonSociale* | * élimination des caractères  ! |
| *Rue, Ville* | * élimination des caractères ‘ - , |
| *Telephone* | // non défini |
| *Fax* | // non défini |
| *Mail* | // non défini |

## Etape 3 - nettoyage des données

Le nettoyage consiste à éliminer les clients qui n’ont plus lieu d’être référencés dans l’ERP ODOO.

**Critères d’identification des clients à éliminer :**

* le champ ACTIF contient la valeur "NON"
* les champs CODE et/ou RAISON SOCIALE et/ou RUE et/ou VILLE contiennent une répétition de " XXXX "

## Etape 4 - vérification des données et identification des anomalies

Cette étape consiste à vérifier la validité des données et déterminer le type d’anomalie.

***Types d’anomalies*** : "RaisonSocialeNR", "RueNR", "CpNR", "VilleNR", "TelNR", "TelNV", "MailNR", " MailNV", "Doublon"

*NR = non renseigné, NV = Non valide*

**Règles de traitement :**

|  |  |  |  |
| --- | --- | --- | --- |
| ***CAS A VERIFIER*** | | ***TRAITEMENT*** | |
| *Insertion du client en BDD (OUI/NON) / valeur de la colonne* | *Anomalie (OUI/NON) / type* |
| RAISON SOCIALE | si champ vide | NON | OUI / "RaisonSocialeNR" |
| RUE | si champ vide | OUI / valeur "NR" | OUI / "RueNR" |
| CP | si champ vide | OUI / valeur "NR" | OUI / "CpNR" |
| VILLE | si champ vide | OUI / valeur "NR" | OUI / "VilleNR" |
| TEL | si champ vide  si invalide | OUI / valeur "NR"  OUI / valeur du TEL | OUI / "TelNR"  OUI / "TelNV" |
| MAIL | si champ vide  si invalide | OUI / valeur "NR"  OUI / valeur du MAIL | OUI / "MailNR"  OUI / "MailNV" |
| Doublons (sont considérés comme doublons des clients possédant la même RAISON SOCIALE et la même ADRESSE - rue, CP, ville) | | NON | OUI / "Doublon" |

## Etape 5 - Insertion des données

Cette étape consiste à insérer les clients valides dans la base PostgreSQL de l’ERP ODOO.

## Etape 6 - affichage du rapport de reprise des données

Le rapport de reprise des données indique :

* le nombre de clients traités (présents dans le fichier source)
* le nombre de clients insérés dans ODOO
* le nombre de clients

*[à compléter]*

# architecture applicative

## Organisation de la solution

La solution **DATA2ODOO.sln** comporte les composantes suivantes :

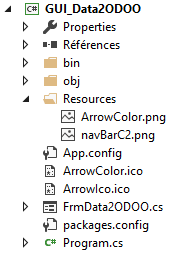
|  |  |  |
| --- | --- | --- |
| *Projet (type / rôle)* | *Classes* |  |
| **GUI\_Data2ODOO**  Windows Forms  *Couche Présentation* | FrmData2ODOO | Formulaire principal de l’utilitaire |
| **DAO\_Data2ODOO**  Bibliothèque de classes (.NET Framework)  *Couche Données* | Connexion.cs | Classe de connexion à la BDD PostgreSQL |
| DAO\_ResPartner.cs | Méthodes CRUD |
| **Model\_Data2ODOO**  Bibliothèque de classes (.NET Framework)  *Couche Métier* | Client.cs | Classe métier |
| **Process\_Data2ODOO**  Bibliothèque de classes (.NET Framework)  *Couche Utilitaire* | LectureSource.cs | Méthode de lecture du fichier source (format CSV) |
| Nettoyage.cs | Méthodes de nettoyage des données clients |
| Formatage.cs | Méthodes de formatage des données clients |
| Verification.cs | Méthodes de vérification des données et d’identification des anomalies |

La solution contient également un ensemble de tests unitaires :

|  |  |  |
| --- | --- | --- |
| **DAO\_Data2ODOOTests**  Bibliothèque de classes  *Tests unitaires* | ConnexionTests.cs | Méthodes de tests de la classe Connexion |
| **Process\_Data2ODOOTests**  Bibliothèque de classes  *Tests unitaires* | NettoyageTests.cs  VerificationTests.cs | Méthodes de tests des classes Nettoyage et Verification |

## Projet GUI\_DATA2ODOO - Interface graphique

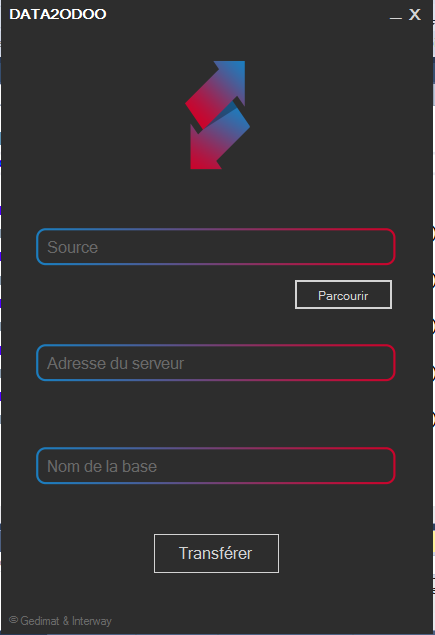
Arborescence du projet



*Ressources images référencées dans le formulaire*

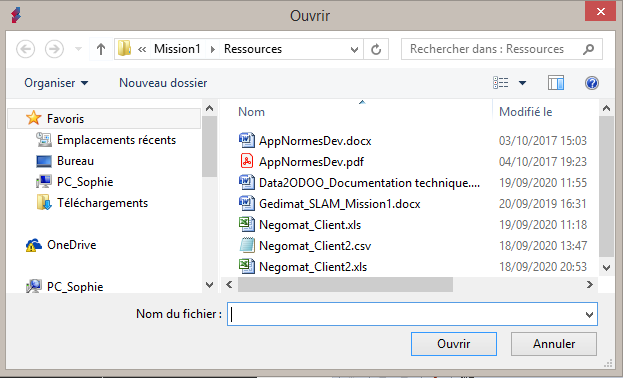
*Références :*  Model\_Data2ODOO, DAO\_Data2ODOO, Process\_Data2ODOO, Npgsql

Formulaire de l’utilitaire **FrmData2ODOO**



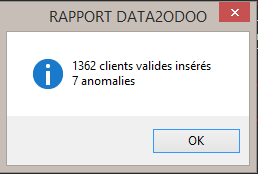
**Boite de dialogue « Ouvrir »**

*s’affiche lors du click sur le bouton « Parcourir »*



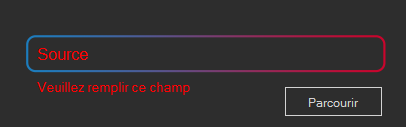
**Boite de dialogue « RAPPORT DATA2ODOO »**

*s’affiche lors du click sur le bouton « Transférer »*



Gestion des erreurs dans le formulaire **FrmData2ODOO**

***Principe*** : Les messages d’erreurs sont affichées en rouge sous les zones concernées, les intitulés des zones passent en rouge également

*Exemple* :

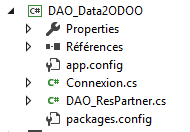
***Erreurs d’interface à gérer*** :

* Champ "Source" non renseigné
* Champ "Adresse du serveur" non renseigné
* Champ "Nom de la base" non renseigné

*[à compléter]*

## Projet DAO\_DATA2ODOO - Acces aux données

Arborescence du projet



*Références :*  Model\_Data2ODOO, Npgsql

Classe **Connexion.cs**

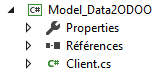
|  |  |
| --- | --- |
| Méthode SeConnecter()  Test unitaire associé :  SeConnecterTest() | Se connecte à la base de données PostgreSQL de ODOO  *Server : { saisi via IHM }*  *Port : 5432*  *User Id : openpg*  *Password : openpgpwd*  *Database : { saisi via IHM }* |

Classe **DAO\_ResPartner.cs**

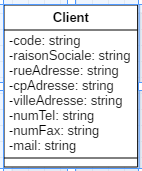
|  |  |
| --- | --- |
| Méthode InsertClient() | Insère un client dans la table Res\_Partner  *Objet Client passé en paramètre* |

## Projet MODEL\_DATA2ODOO - Classe métier

Arborescence du projet



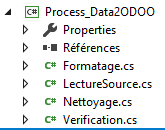
Classe **Client.cs**



Attributs / Constructeur / Accesseurs

## Projet PROCESS\_DATA2ODOO - Processus traitement données client

Arborescence du projet



*Références :*  Model\_Data2ODOO

Etape 1 - Lecture du fichier CSV - Classe **LectureSource.cs**

|  |  |  |
| --- | --- | --- |
| Méthode ChargerListClient() | Lit le fichier CSV et génère une liste d’objets Client  *Chemin du fichier CSV passé en paramètre* | |
|  | |  |

Etape 2 - Formatage des données - Classe **Formatage.cs**

|  |  |
| --- | --- |
| Méthode Formater() | Formate la liste des objets Client en entrée  [***Règles de formatage***](#Formatage_Regles) |

Etape 3 - Nettoyage des données - Classe **Nettoyage.cs**

|  |  |
| --- | --- |
| Méthode Nettoyer() | Elimine de la liste des objets Client en entrée les clients qui n’ont plus lieu d’être références dans l’ERP  ODOO  [***Critères d’identification***](#Nettoyage_Criteres) |
| Méthode RepetitiondeX()  Test unitaire associé :  RepetitiondeXTest() | Indique si la chaine en entrée contient une répétition de X ou non |

Etape 4 - Vérification des données - Classe **Verification.cs**

|  |  |
| --- | --- |
| Méthode Verifier() | Vérifie la liste des objets Client en entrée ; une liste de clients valides et un dictionnaire des anomalies sont produits en sortie  [***Règles de traitement***](#Verification_Regles) |
| Méthode ChampVide()  Test unitaire associé :  ChampVideTest() | Indique si le champ en entrée est vide ou non |