EDI NEGOCE MANUEL FRANÇAIS / ENGLISH

VERSION FRANCAISE

- **Lecture de la table client dans l'application Négoce**
- **Création d'une fiche client dans l'application Négoce**
- **Modification de la fiche client dans l'application Négoce**
- **Création d'une adresse client Négoce livraison**
- **Modification d'adresse livraison client dans Négoce**
- **Lecture de la table panier**
- **Création d'un panier**
- **Lecture de Pièce**
- **Transformer un panier en pièce**
- **Passer une Commande Client en Commande confirmée**
- **Passer une Commande Client confirmée en non confirmée**
- **Solder une Commande Client**
- **Passer une Commande Client Soldée en Commande confirmée**
- **Passer une Commande Client Soldée en non confirmée**
- **Modifier une Commande Client**
- **Recevoir le pdf d'une piece**

ENGLISH VERSION

- **Description of the Negoce application's EDI interface**
- **Reading the client table in the Negoce application**
- **Creating a client file in the Negoce application**
- **Modifying a client file in the Negoce application**
- **Creating a Negoce client delivery address**
- **Modification of delivery address client in Negoce**
- **Reading the basket table**
- **Creating a basket**
- **Reading documents**
- **Transforming a basket into a document**
- **Turn a Client Order into a Confirmed Order**
- **Turn a confirmed Client Order into an unconfirmed one**
- **Complete a Client Order**
- **Turn a completed Client Order into a confirmed Order**
- **Turn a completed Client Order into an unconfirmed one**
- **Modify a Client Order**
- **Receive the pdf of a document**

VERSION FRANCAISE

Descriptif de l'interface EDI Négoce

L'interface EDI Négoce permet d'avoir des actions dans une fiche client dans Négoce.

Elle permet également des actions dans des paniers et des pièces commerciales.

FLUX DATA SOUS FORME JSON

Envoi d'information

Les données doivent être envoyées en JSon (langage d'échange de données) dans Fastmag et en appelant la page "EDINegoce.ips" depuis l'adresse du serveur hébergeant l'enseigne du client avec les paramètres POST suivants :

- enseigne
- magasin
- compte
- motpasse
- action
- data

Retour d'information

En retour, vous recevez un JSon:

ou

OK

ou

OK|JSon

ou

KO | message erreur

ORDRE DES TRANSACTIONS A ENVOYER

L'ordre des transactions est important et doit être le suivant :

- Transaction concernant la création de la fiche client
- Transaction concernant la création de pièce

Lecture de la table client dans l'application Négoce

Description de la fonction:

Cette transaction permet de lire les informations de la fiche client wholesale. Les clés permettant l'interrogation de la fiche sont, le code, le compte client, ou le nom intégral du client.

En retour l'ensemble des données de la fiche client seront restituées

Le nom de l'action pour cette commande est :

- LIRECLIENT

Le data comprend un objet JSon qui sera ensuite traduit et intégrer à la base de données.

Le JSon peut contenir un ou plusieurs champs de la table "NegoceClients" permettant d'identifier le client dont on veut récupérer les informations.

Par exemple :

data = [{nom:"ACE INFORMATIQUE"}] ou data = [{client:"248"}] ou data= [{compteclient:"ACE"}]

RETOUR :

Json

ou

KO|message erreur

Création d'une fiche client dans l'application Négoce

Description de la fonction:

Cette transaction permet de créer une fiche client wholesale

Le nom de l'action pour cette commande est :

- CREERCLIENT

Le data comprend un objet JSon qui sera ensuite traduit et intégrer à la base de données.

Par exemple :

data=[{nom:"ACE INFORMATIQUE", Adresse1:"9 rue de l'aqueduc", CodePostal:"75010", Ville:"Paris", Pays:"France", telephone:"0155260800", Portable:"06060606", EMail:"contacts@aceinformatique.fr", Fax:"0155260801"}]

On insère le client dans la table NegoceClients avec tous les champs contenu dans le JSON.

RETOUR :

OK|JSon avec le le champ client du client créé : ex OK|[{Client : "5063"}]

ou

KO|JSon la liste des erreurs possibles ex KO|[{"Nom":"ACE INFORMATIQUE","CompteClient":"","ErreurSurCompteClient":"","CodeExterne":"","ErreurSurCodeExterne":"","ModReg":"","ErreurSurModReg":"","Devise":"","ErreurSurDevise":"","Secteur":"","ErreurSurSecteur":"","Representant":"XSE","ErreurSurRepresentant":"REPRESENTANT INCONNU","Transporteur":"","ErreurSurTransporteur":"","FamilleClient":"","ErreurSurFamilleClient":""}]

ou

KO|message erreur

Modification de la fiche client dans l'application Négoce

Description de la fonction:

Cette transaction permet de modifier une fiche client wholesale. La clé permettant d'indiquer le client à modifier est le code client.

Le nom de l'action pour cette commande est :

- MODIFCLIENT

Le data comprend un objet JSon comprenant le champs client et l'ensemble des champs à mettre à jour pour ce dernier.

Par exemple :

data = [{client:"1", nom:"ACE INFORMATIQUE", telephone:"0155260800"}, {client:"2",nom:"ACE", telephone:"0155260802"}, {client:"5062",nom:"TITI", telephone:"01010101"}]

Ici on fait un Update sur la table NegoceClients de l'ensemble des champs pour le numéro de client indiqué.

Note: il est possible d'effectuer plusieurs MAJ à la fois (cf exemple)

RETOUR :

OK

ou

KO|message erreur

Création d'une adresse client Négoce livraison

Description de la fonction:

Cette transaction permet de créer une adresse de livraison pour un client wholesale. Le client peut avoir plusieurs adresse de livraison. Le champ “Adrlivraison” sera utiliser comme étant le code de l’adresse de livraison créée.

Le champ “client” permet d’identifier le client pour lequel l’adresse de livraison sera créée.

Le nom de l'action pour cette commande est :

- CREERADRCLIENT

Le data comprend un objet JSon qui sera ensuite traduit et intégrer à la base de données.

Par exemple :

data=[{client:"248", AdrLivraison:"SITE2", Nom:"LE SITE 2"}, {client:"248", AdrLivraison:"SITE3", Nom:"LE SITE 3"}]

On insère l'adresse dans la table NegoceClientsAdresses avec tous les champs contenu dans le JSON.

RETOUR :

OK

ou

KO|JSon la liste des adresses en erreur ex KO|[{client :"248", AdrLivraison:"SITE7"},{client :"248", AdrLivraison:"SITE7"}]

ou

KO|message erreur

Modification d'adresse livraison client dans Négoce

Description de la fonction:

Cette transaction permet de modifier une adresse de livraison pour un client wholesale. Le client peut avoir plusieurs adresse de livraison. Le champ “Adrlivraison” sera utiliser comme étant le code de l’adresse de livraison créée.

Le champ “client” permet d’identifier le client pour lequel l’adresse de livraison sera modifiée.

Le nom de l'action pour cette commande est :

- MODIFADRCLIENT

Le data comprend un objet JSon comprenant le champs client et l'ensemble des champs à mettre à jour pour ce dernier.

Par exemple :

data = [{client:"248", AdrLivraison:"SITE2", Nom:"LE SITE 22"}, {client:"248", AdrLivraison:"SITE3", Nom:"LE SITE 33"}]

Ici on fait un Update sur la table NegoceClientsAdresses de l'ensemble des champs pour le numéro de client et l'adrlivraison indiqués.

On peut effectuer plusieurs MAJ à la fois (cf exemple)

RETOUR :

OK

ou

KO|message erreur

Lecture de la table panier

Description de la fonction:

Cette transaction permet de lire les informations d'un panier article. Les clés permettant l'interrogation du panier est "IdentifiantPanier"

En retour l'ensemble des données constituant le panier seront restituées

Le nom de l'action pour cette commande est :

- LIREPANIER

Le data comprend un objet JSon qui nous permet d'identifier le panier.

Par exemple :

data = [{IdentifiantPanier:"TEST TARIF"}]

En retour vous recevrez un flux JSON contenant l'ensemble des éléments du panier

RETOUR :

Json

ou

KO|message erreur

Création d'un panier

Description de la fonction:

Cette transaction permet de créer un panier. Le panier est une étape nécessaire avant la création d'une pièce commerciale. En effet, on utilisera la fonction transformer un panier en pièce pour créer une pièce commerciale (exemple: COMMANDE, LIVRAISON, FACTURE, ...). Le contenu du panier est les éléments nécessaires à la création d'une pièce commerciale concernant les articles, quantités, prix, ...

Le nom de l'action pour cette commande est :

- CREERPANIER

Le data comprend un objet JSon qui comprend tous les élements nécessaires à la création d'un panier.

Par exemple :

data = [{IdentifiantPanier:"PANIER", Utilisateur:"LINDA", DatePanier:"2011-03-01", BarCode:"ANGELA", Couleur:"NOIR", taille:"T1", Qte:"3", Prix:"25.6", Remise:"0", Motif:""}, {IdentifiantPanier:"PANIER", Utilisateur:"LINDA", DatePanier:"2011-03-01", BarCode:"ANGELA", Couleur:"VERT", taille:"T2", Qte:"3", Prix:"25.6", Remise:"0", Motif:""}]

NB : Le champ barcode peut également contenir, si la couleur et la taille sont vides, l'id produit (sous la forme #ID), le gencod ou la référence fournisseur de l'article.

En retour vous recevrez la liste du ou des paniers créés.

RETOUR :

Json ex : [{IDPanier:"348"}]

ou

KO|message erreur

Lecture de Pièce

Description de la fonction:

Cette transaction permet de lire les informations d'une pièce commerciale wholesale. La clé permettant l'interrogation de la pièce est le code de l'entête de pièce Négoce.

En retour l'ensemble des données de la pièce commerciale seront restituées.

Le nom de l'action pour cette commande est :

- LIREPIECE

Le data comprend un objet JSon qui contient L'identifiant de la pièce (le champ entete de la table NegoceEntetes).

Par exemple :

data = [{Entete:"4"}]

En retour vous recevrez un flux JSON contenant l'ensemble des éléments de la pièce

RETOUR :

Json

ou

KO|message erreur

Transformer un panier en pièce

Description de la fonction:

Cette transaction permet de transformer un panier en pièce commerciale. La création du panier est préalablement nécessaire. Une fois la pièce commerciale créée il est possible de supprimer le panier ayant servi à la constitution de la pièce en indiquant GarderPanier:"0" . Si vous souhaitez conserver le panier indiquer GarderPanier:"1" .

Pour appliquer la remise du client plutôt que celle du panier il faut préciser le paramètre AppliquerRemiseClient avec une valeur à 1.

Pour les pièces utilisant le stock il est possible de préciser JSonErreurStock avec une valeur de 1 pour récupérer dans le retour un json avec les produits en erreur de stock.

Pour les BL, il est possible de préciser le type de BL grâce au champ reservation pouvant avoir les valeurs suivantes :

0 : BL Normal

1 : BL de réservation

2 : BL Non facturable

3 : Réservation Non livrable

Le nom de l'action pour cette commande est :

- PANIERVERSPIECE

Le data comprend un objet JSon comprenant les éléments permettant de faire la transformation du panier en pièce.

Par exemple :

data = [{IDPanier:"28", Client: "248", NaturePiece:"COMMANDE", DatePiece:"2009-10-15", FraisPort:"0", FraisPostaux:"0", TauxEscompte:"", NbColis:"", Representant:"", Secteur:"75", Transporteur:"", ModReg:"600", DateEcheance:"", DelaiReg:"", AdrLivraison:"OPERA", CriterePiece:"", Magasin:"CENTRAL", DateLivSouhaite:"2009-10-15", DateLivPrevue:"2009-10-15", RefCde:"TEST LINDA", observations:"", Devise:"", TauxDevise:"", Escompte:"0", GarderPanier:"0", BLDeductionCde:"0", CommandeNonConfirmee:"0"}]

RETOUR :

Json avec le numéro de piece ex [{Piece:"CD14100002"}]

ou - si problème de stock et JSonErreurStock à 1

KO|JSon Ex : [message : "Il y a des articles manquants ou inexistants." , StockInexistant : [{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "BLEU", Taille : "36", Quantite : "10"},{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "BLEU", Taille : "38", Quantite : "1"}] , StockManquant : [{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "ROUGE", Taille : "36", Quantite : "10", Stock : "8"},{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "ROUGE", Taille : "38", Quantite : "1", Stock : "0"}]]

ou

KO|message erreur

Passer une Commande Client en Commande confirmée

Description de la fonction:

Cette transaction permet de changer le statut d'une commande afin de la passer en commande confirmée.

Le nom de l'action pour cette commande est :

- CONFIRMERCOMMANDE

Le data comprend un objet JSon avec le numéro d'entête de la commande.

Par exemple :

data = [{Entete:"4"}]

RETOUR :

OK

ou

KO|message erreur

Passer une Commande Client confirmée en non confirmée

Description de la fonction:

Cette transaction permet de changer le statut d'une commande afin de la passer en commande non confirmée.

Le data comprend un objet JSon avec le numéro d'entête de la commande.

Par exemple :

data = [{Entete:"4"}]

RETOUR :

OK

ou

KO|message erreur

Solder une Commande Client

Description de la fonction:

Cette transaction permet de changer le statut d'une commande afin de la passer en commande soldée.

Une commande soldée ne figurera plus dans les commandes à livrer.

Le nom de l'action pour cette commande est :

- SOLDERCOMMANDE

Le data comprend un objet JSon avec le numéro d'entête de la commande.

Par exemple :

data = [{Entete:"4"}]

RETOUR :

OK

ou

KO|message erreur

Passer une Commande Client Soldée en Commande confirmée

Description de la fonction:

Cette transaction permet de changer le statut d'une commande afin de la passer en commande confirmée.

Une commande confirmée figurera plus dans les commandes à livrer.

Le nom de l'action pour cette commande est :

- DESOLDERCOMMANDE2CONF

Le data comprend un objet JSon avec le numéro d'entête de la commande.

Par exemple :

data = [{Entete:"4"}]

RETOUR :

OK

ou

KO|message erreur

Passer une Commande Client Soldée en non confirmée

Description de la fonction:

Cette transaction permet de changer le statut d'une commande afin de la passer en commande non confirmée.

Une commande non confirmée figurera plus dans les commandes à livrer.

Le nom de l'action pour cette commande est :

- DESOLDERCOMMANDE2NONCONF

Le data comprend un objet JSon avec le numéro d'entête de la commande.

Par exemple :

data = [{Entete:"4"}]

RETOUR :

OK

ou

KO|message erreur

Modifier une Commande Client

Description de la fonction:

Cette transaction permet de modifier les quantités d'une commande.

Les champs clés permettant d'identifier la commande et la ligne de commande à modifier sont les champs Entete et "Ligne" .

Le nom de l'action pour cette commande est :

- MODIFIERCOMMANDE

Le data comprend un objet JSon avec l'entete de la commande + Numero de la ligne + Nouvelle quantite.

Si la qté est < 0 on supprime la ligne.

On ne met que les lignes modifiées.

Par exemple :

data =
[{Entete:"4","Ligne":"36","newqte":"2"},{Entete:"4","Ligne":"37","newqte":"0"},{Entete:"4","Ligne":"38","newqte":"3"}]

RETOUR :

OK

ou

KO|message erreur

Recevoir le pdf d'une piece

Description de la fonction:

Cette transaction permet d'expédition par Email d'une pièce commerciale en pdf

La clé permettant l'envoi de la pièce est le code de l'entête de pièce Négoce.

Le nom de l'action pour cette commande est :

- ENVOIPDF

Le data comprend un objet JSon avec l'entete ou le numéro de la pièce, le client et la nature de la pièce.

Par exemple :

data = [{"Entete":"4","Client":"10","Nature":"BL"}] ou date = [{"Piece":"BL120600004","Client":"10","Nature":"BL"}]

RETOUR :

Le fichier pdf

ou

KO|message erreur

ENGLISH VERSION

Description of the Negoce application's EDI interface

The EDINegoce interface enables you to carry out actions on a client file in Negoce.

It also enables you to perform actions on baskets and business documents.

FLUX DATA IN THE FORM OF JSON

Sending information

The data must be sent in JSON (language for data exchange) to Fastmag and by opening the page "EDINEgoce.ips" from the address of the server hosting the client's brand, with the following POST parameters :

- brand
- store
- account
- password
- action
- data

Returning information

In return, you will receive a JSON:

or

OK

or

OK|JSon

or

KO|message error

ORDER OF TRANSACTIONS TO SEND

The order of transactions is important and must be as follows :

- Transaction involving the creation of the client file
- Transaction involving the creation of document

Reading the client table in the Negoce application

Feature Description:

This transaction allows you to read information from the wholesale client file. The keys which enable consultation of the file are the code, the client file, or the full name of the client.

In return, the client file data set will be restored

The action name for this command is :

Name	Explanation / Translation
LIRECLIENT	• Lire = Read • Client = Client

The data includes a JSon object which will be translated and integrated into the database after.

JSON may contain one or several fields from the table “NegoceClients”, allowing the identification of the client from whom information is to be collected.

For example:

data = [{nom:"ACE INFORMATIQUE"}]

ou

data = [{client:"248"}]

data = [{name:"ACE INFORMATIQUE"}]

or

data = [{client:"248"}]

BACK :

JSon

or

KO|message error

Creating a client file in the Negoce application

Feature Description:

This transaction allows you to create a wholesale client file

The action name for this command is :

Name	Explanation / Translation
CREERCLIENT	• Créer = Create • Client = Client

The data includes a JSON object which will be translated and integrated into the database after.

For example:

data=[{nom:"ACE INFORMATIQUE", Adresse1:"9 rue de l'aqueduc", CodePostal:"75010", Ville:"Paris", Pays:"France", telephone:"0155260800", Portable:"06060606", EMail:"contacts@aceinformatique.fr", Fax:"0155260801"}]

data=[{name:"ACE INFORMATIQUE", Address1:"9 rue de l'aqueduc", Postcode:"75010", City:"Paris", Country:"France", telephone:"0155260800", mobile:"06060606", EMail:"contacts@aceinformatique.fr", Fax:"0155260801"}]

Insert the client into the table “NegoceClients” with all the fields contained in the JSON.

Name	Explanation / Translation
CodePostal	• Code Postal = Postcode

BACK :

OK|JSON with the client field from the created client : ex OK|[{Client : "5063"}]

or

KO|JSON list of possible errors ex KO|[{"Name":"ACE INFORMATIQUE","ClientAccount":"","ErrorInClientAccount":"","ExternalCode":"","ErrorInExternalCode":"","MOP":"","ErrorInMOP":"","Currency":"","ErrorInCurrency":"","Sector":"","ErrorInSector":"","Representative":"XSE","ErrorInRepresentative":"REPRESENTANT INCONNU","Transporter":"","ErrorInTransporter":"","FamilyClient":"","ErrorInFamilyClient":""}]

or

KO|message error

Name	Explanation / Translation
CompteClient	• Compte = Account • Client = Client
ErreurSurCompteClient	• Erreur = Error • Sur = In • Compte = Account • Client = Client
ErreurSurModReg	• Erreur = Error • Sur = In • Mode = Method • Réglement = Payment

Modifying a client file in the Negoce application

Feature Description:

This transaction allows you to modify a wholesale client file. The key which enables you to specify the client to be modified is the client code.

The action name for this command is :

Name	Explanation / Translation
MODIFCLIENT	• Modifier = Modify • Client = Client

The data includes a JSON object comprising the client field and all the fields to be updated for the latter.

For example:

data = [{client:"1", nom:"ACE INFORMATIQUE", telephone:"0155260800"}, {client:"2",nom:"ACE", telephone:"0155260802"}, {client:"5062",nom:"TITI", telephone:"01010101"}]

Ici on fait un Update dans la table “NegoceClients” de l'ensemble des champs pour le numéro de client indiqué.

data = [{client:"1", name:"ACE INFORMATIQUE", telephone:"0155260800"}, {client:"2",nom:"ACE", telephone:"0155260802"}, {client:"5062",nom:"TITI", telephone:"01010101"}]

Here, update all fields in the table “NegoceClients” with the number of the specified client.

Note: it is possible to carry out several updates at the same time (see example)

BACK :

OK

or

KO|message error

Creating a Negoce client delivery address

Feature Description:

This transaction allows you to create a delivery address for a wholesale client. The client can have several delivery addresses. The field “DeliveryAddr” will be used as the code for the delivery address created.

The field “client” allows you to identify the client for whom the delivery address will be created.

The action name for this command is :

Name	Explanation / Translation
CREERADRCLIENT	• Créer = Create • Adresse = Address • Client = Client

The data includes a JSON object which will be translated and integrated into the database after.

For example:

data=[{client:"248", AdrLivraison:"SITE2", Nom:"LE SITE 2"}, {client:"248", AdrLivraison:"SITE3", Nom:"LE SITE 3"}]

data=[{client:"248", DeliveryAddr:"SITE2", Name:"LE SITE 2"}, {client:"248", DeliveryAddr:"SITE3", Name:"LE SITE 3"}]

Insert the client into the table “NegoceClients” with all the fields contained in the JSON.

Name	Explanation / Translation
NegoceClientsAdresses	• Négoce = Negoce • Clients = Clients • Adresses = Addresses

BACK :

OK

or

KO|JSon list of addresses with errors ex KO|[{client :"248", DeliveryAddr:"SITE7"},{client :"248", DeliveryAddr:"SITE7"}]

or

KO|message error

Modification of delivery address client in Negoce

Feature Description:

This transaction allows you to modify a delivery address for a wholesale client. The client can have several delivery addresses. The field “DeliveryAddr” will be used as the code for the delivery address created.

Name	Explanation / Translation
Adrlivraison	• Adresse = Address • Livraison = Delivery

The field “client” enables you to identify the client for whom the delivery address will be modified.

The action name for this command is :

Name	Explanation / Translation
MODIFADRCLIENT	• Modifier = Modify • Adresse = Address • Client = Client

The data includes a JSON object comprising the client field and all the fields to be updated for the latter.

For example:

data = [{client:"248", AdrLivraison:"SITE2", Nom:"LE SITE 22"}, {client:"248", AdrLivraison:"SITE3", Nom:"LE SITE 33"}]

data=[{client:"248", DeliveryAddr:"SITE2", Name:"LE SITE 22"}, {client:"248", DeliveryAddr:"SITE3", Name:"LE SITE 33"}]

Here you update all fields in the table “NegoceClientsAddresses” with the number of the client and the delivery address indicated.

It is possible to carry out several updates at the same time (see example)

BACK :

OK

or

KO|message error

Reading the basket table

Feature Description:

This transaction allows you to read information on a basket item. The key which allows you to consult the basket is “IDBasket”

Name	Explanation / Translation
IdentifiantPanier	• Identifiant = ID • Panier = Basket

In return the data set constituting the basket will be restored

The action name for this command is :

Name	Explanation / Translation
LIRECLIENT	• Lire = Read • Panier = Basket

The data includes a JSON object which enables us to identify the basket.

For example:

data = [{IdentifiantPanier:"TEST TARIF"}]

En retour vous recevrez un flux JSON contenant l'ensemble des éléments du panier

data = [{IDBasket:"TEST TARIF"}]

In return you will receive a JSON flux containing all the elements of the basket

BACK :

JSON

or

KO|message error

Creating a basket

Feature Description:

This transaction allows you to create a basket. The basket is a necessary step before the creation of a commercial document. In fact, you will use the function of transforming a basket into a document to create a commercial document (example: ORDER, DELIVERY, BILL, ... The contents of the basket are the elements needed to create a commercial document regarding articles, quantities, price, ...

The action name for this command is :

Name	Explanation / Translation
CREERPANIER	• Créer = Create • Panier = Basket

The date includes a JSON object which comprises all the elements needed to create a basket.

For example:

data = [{IdentifiantPanier:"PANIER", Utilisateur:"LINDA", DatePanier:"2011-03-01", BarCode:"ANGELA", Couleur:"NOIR", taille:"T1", Qte:"3", Prix:"25.6", Remise:"0", Motif:""}, {IdentifiantPanier:"PANIER", Utilisateur:"LINDA", DatePanier:"2011-03-01", BarCode:"ANGELA", Couleur:"VERT", taille:"T2", Qte:"3", Prix:"25.6", Remise:"0", Motif:""}]

data = [{IDBasket:"PANIER", User:"LINDA", DateBasket:"2011-03-01", BarCode:"ANGELA", Colour:"NOIR", size:"T1", Qty:"3", Price:"25.6", Discount:"0", Pattern:""}, {IDBasket:"PANIER", User:"LINDA", DateBasket:"2011-03-01", BarCode:"ANGELA", Colour:"VERT", size:"T2", Qty:"3", Price:"25.6", Discount:"0", Pattern:""}]

Name	Explanation / Translation
IdentifiantPanier	• Identifiant = ID • Panier = Basket

NB: The barcode field may also contain, if the colour and size are empty, the product ID (in the form of #ID), the EAN bar code or the article's supplier reference.

In return you will receive the list of the baskets created.

BACK :

JSON ex : [{IDBasket:"348"}]

or

KO|message error

Name	Explanation / Translation
IDPanier	• ID = ID • Panier = Basket

Reading documents

Feature Description:

This transaction allows you to read information on a wholesale business document.

The key which allows you to consult the document is the value of the field ENTETE from the NegoceEntetes table.

In return, all the data from the commercial document will be restored.

Name	Explanation / Translation
ENTETE	• Entête = header

The action name for this command is :

Name	Explanation / Translation
LIREPIECE	• Lire = Read • Pièce = Document

The data includes a JSON object which contains the document ID (the field “entete” from the table “NegoceEntetes”).

For example:

data = [{Entete:"4"}]

data = [{Heading:"4"}]

In return you will receive a JSON flux containing all the elements of the document

BACK :

JSON

or

KO|message error

Transforming a basket into a document

Feature Description:

This transaction allows you to transform a basket into a commercial document. The creation of a basket is necessary beforehand. Once the commercial document has been created, it is possible to delete the basket which was used to form the document by indicating KeepBasket: “0” . If you wish to keep the basket, indicate KeepBasket: “1”.

Name	Explanation / Translation
GarderPanier	• Garder = Keep • Panier = Basket

To apply a discount to the client rather than to the basket you must specify the parameter ApplyDiscountClient with a value of 1.

Name	Explanation / Translation
AppliquerRemiseClient	• Appliquer = Apply • Remise = Discount • Client = Client

For documents which use stock, it is possible to specify JSonErrorStock with a value of 1 to retrieve a json with the incorrect stock products.

For BL, it is possible to specify the type of BL thanks to the reservation field which may have the following values :

0 : BL Normal

1 : BL reservation

2 : BL Non-billable

3 : Reservation Non-deliverable

The action name for this command is :

Name	Explanation / Translation
PANIERVERSPIECE	• Panier = Basket • Vers = To • Pièce = Document

The data includes a JSon object containing elements which allow it to transform a basket into a document.

For example:

data = [{IDPanier:"28", Client: "248", NaturePiece:"COMMANDE", DatePiece:"2009-10-15", FraisPort:"0", FraisPostaux:"0", TauxEscompte:"", NbColis:"", Representant:"", Secteur:"75", Transporteur:"", ModReg:"600", DateEcheance:"", DelaiReg:"", AdrLivraison:"OPERA", CriterePiece:"", Magasin:"CENTRAL", DateLivSouhaite:"2009-10-15", DateLivPrevue:"2009-10-15", RefCde:"TEST LINDA", observations:"", Devise:"", TauxDevise:"", Escompte:"0", GarderPanier:"0", BLDeductionCde:"0", CommandeNonConfirmee:"0"}]

data = [{IDPanier:"28", Client: "248", TypeDocument:"COMMANDE", DateDocument:"2009-10-15", ChargeIncurred:"0", PostalCharge:"0", DiscountRate:"", NbColis:"", Representative:"", Sector:"75", Transporter:"", MOP:"600", EndDate:"", PaymentDelay:"", AddrDelivery:"OPERA", CriterePiece:"", Store:"CENTRAL", DesiredDelDate"2009-10-15", ExpectedDelDate:"2009-10-15", RefCde:"TEST LINDA", observations:"", Currency:"", TransactionFee:"", Discount:"0", KeepBasket:"0", BLDeductionCde:"0", UnconfirmedOrder:"0"}]

BACK :

JSON with the number of document ex [{Document:"CD14100002"}]

or - if stock problem and JSonErrorStock at 1

OK|JSon

Ex : [message : "Il y a des articles manquants ou inexistants." , StockInexistant : [{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "BLEU", Taille : "36", Quantite : "10"},{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "BLEU", Taille : "38", Quantite : "1"}] , StockManquant : [{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "ROUGE", Taille : "36", Quantite : "10",

Stock : "8"},{Magasin : "OPERA", Barcode : "ANGELA", Designation : "CHEMISE ANGELA", Couleur : "ROUGE", Taille : "38", Quantite : "1", Stock : "0"}]]

Ex : [message : “There are missing or non-existent items.” , StockInexistant : [{ Shop : "OPERA", Barcode : "ANGELA", Designation : "SHIRT ANGELA", Colour : "BLUE", Size : "36", Quantity : "10"},{Shop : "OPERA", Barcode : "ANGELA", Designation : "SHIRT ANGELA", Colour : "BLUE", Size : "38", Quantity : "1"}] , MissingStock : [{Shop : "OPERA", Barcode : "ANGELA", Designation : "SHIRT ANGELA", Colour : "RED", Size : "36", Quantity : "10", Stock : "8"},{Shop : "OPERA", Barcode : "ANGELA", Designation : "SHIRT ANGELA", Colour : "RED", Size : "38", Quantity : "1", Stock : "0"}]]

or

KO|message error

Turn a Client Order into a Confirmed Order

Feature Description:

This transaction allows you to change the status of an order to turn it into a confirmed order.

The action name for this command is :

Name	Explanation / Translation
CONFIRMERCOMMANDE	• Confirmer = Confirm • Commande = Order

The data includes a JSON object with the number in front of the order.

For example:

data = [{Heading:"4"}]

BACK :

OK

or

KO|message error

Turn a confirmed Client Order into an unconfirmed one

Feature Description:

This transaction allows you to change the status of an order to turn it into an unconfirmed order.

The data includes a JSON object with the number in front of the order.

For example:

data = [{Entete:"4"}]

data = [{Heading:"4"}]

BACK :

OK

or

KO|message error

Complete a Client Order

Feature Description:

This transaction allows you to change the status of an order to turn it into a completed order.

A completed order will no longer appear on the orders due for delivery.

The action name for this command is :

Name	Explanation / Translation
SOLDERCOMMANDE	• Solder = Complete • Commande = Order

The data includes a JSON object with the number in front of the order.

For example:

data = [{Entete:"4"}]

data = [{Heading:"4"}]

BACK :

OK

or

KO|message error

Turn a completed Client Order into a confirmed Order

Feature Description:

This transaction allows you to change the status of an order to turn it into a confirmed order.

A confirmed order will no longer appear on the orders due for delivery.

The action name for this command is :

Name	Explanation / Translation
DESOLDERCOMMANDE2CONF	• Dé-solder = De-complete • Commande = Order • Confirmé = Confirmed

The data includes a JSON object with the number in front of the order.

For example:

data = [{Entete:"4"}]

data = [{Heading:"4"}]

BACK :

OK

or

KO|message error

Turn a completed Client Order into an unconfirmed one

Feature Description:

This transaction allows you to change the status of an order to turn it into an unconfirmed order.

A confirmed order will no longer appear on the orders due for delivery.

The action name for this command is :

Name	Explanation / Translation
DESOLDERCOMMANDE2NONCONF	• Dé-solder = De-complete • Commande = Order • Non-confirmé = Unconfirmed

The data includes a JSON object with the number in front of the order.

For example:

data = [{Entete:"4"}]

data = [{Heading:"4"}]

BACK :

OK

or

KO|message error

Modify a Client Order

Feature Description:

This transaction allows you to modify the quantities of an order.

The key fields which allow you to identify the order and the command line which need to be modified are the fields Heading and" Line".

The action name for this command is :

Name	Explanation / Translation
MODIFIERCOMMANDE	• Modifier = Modify • Commande = Order

The data includes a JSON object with the command heading + line number + new quantity.

If the quantity is <0, delete the line.

Only put modified lines.

For example:

data = [{Entete:"4","Ligne":"36","newqte":"2"},{Entete:"4","Ligne":"37","newqte":"0"},{Entete:"4","Ligne":"38","newqte":"3"}]

data = [{Heading:"4","Line":"36","newqty":"2"},{Heading:"4","Line":"37","newqty":"0"},{Heading:"4","Line":"38","newqty":"3"}]

BACK :

OK

or

KO|message error

Receive the pdf of a document

Feature Description:

This transaction allows you to send a commercial document via email in pdf form.

The key which allows you to send the document is the code at the top of the document Negoce.

The action name for this command is :

Name	Explanation / Translation
ENVOIPDF	• Envoie = Send • PDF = PDF

The data includes a JSON object with the heading or the number of the document, the client and the type of document.

For example:

data = [{"Entete":"4","Client":"10","Nature":"BL"}]

data = [{"Piece":"BL120600004","Client":"10","Nature":"BL"}]

data = [{"Heading":"4","Client":"10","Type":"BL"}]

data = [{"Document":"BL120600004","Client":"10","Type":"BL"}]

BACK :

PDF File

or

KO|message error