

DOCUMENTATION API

SPARKWEB – V24.16



SAS au capital de 30 000 € – RCS Mâcon 519 816 292 – NAF 6202A



SAS Greenspark
16 Rue Nationale 71420 G  nelard FRANCE

 +33 9 72 55 39 22  contact@greenspark.fr  www.greenspark.fr

SOMMAIRE

1	DESCRIPTIF DU DOCUMENT	3
2	GESTION DES TOKENS	4
3	INPUT / OUTPUT	5
3.1	Paramètre Entrée	5
3.2	Paramètre en sortie	5
4	FONCTIONS	6
4.1	Articles	6
4.1.1	Récupération d'un article (GetArticle) - Obsolète	6
4.1.2	Récupération de plusieurs articles (GetArticles)	7
4.1.3	Mise à jour d'un article (UpdateArticle)	9
4.1.4	Récupération des tarifs (GetTarifs)	10
4.2	Clients	11
4.2.1	Vérification qu'un client existe (DoesClientExist)	11
4.2.2	Création d'un client (CreateClient)	12
4.2.3	Création d'un client (UpdateClient)	12
4.2.4	Récupération d'un client (GetClient)	14
4.2.5	Récupération de plusieurs clients (GetClients)	15
4.2.6	Récupération de plusieurs Contacts (GetContacts)	18
4.2.7	Modification mail client (UpdateEmail)	19
4.3	Commandes	20
4.3.1	Création d'une commande (CreateOrder)	21
4.3.2	Création d'une ligne de commande (CreateOrderLine)	22
4.3.3	Validation de la commande (ValidateOrder)	23
4.3.4	Récupération (GetOrder)	24
4.3.5	DashBoard – Commande	25
5	DEBUTER AVEC L'API (PHP)	26
5.1	Régénérer le dossier API (si besoin)	26
5.2	Configuration du service	26
5.3	Exemples de codes	27
6	GESTION DES ERREURS	28

1 DESCRIPTIF DU DOCUMENT

Ce document représente la documentation concernant l'API de communication entre un site externe et le logiciel SparkWine.

Cette interface (API) va permettre l'échange de données entre le site WEB et l'ERP grâce à différents modules :

- Partie Article :
 - **GetArticle** : Récupération d'un article
 - **GetArticles** : Récupération de plusieurs articles
 - **UpdateArticle** : Mise à jour de l'id web d'un article
 - **GetTarifs** : Récupération de plusieurs tarifs associés aux articles
- Partie Commande :
 - Création de nouvelles commandes
 - En 3 étapes :
 - **CreateOrder** :
 - Création de l'entête
 - **CreateOrderLine** :
 - Création des lignes (On appelle autant de fois qu'il y a de lignes)
 - **ValidateOrder** :
 - Indique que l'on a terminé, on passe ici certains champs qui sont utilisés pour un contrôle d'intégrité + Infos de paiement
 - **GetOrder** : Récupération des informations d'une commande
- Partie Client :
 - **CreateClient** : Création d'un client
 - **DoesClientExist** : Retour le code client SparkWine en fonction d'un eMail
 - **UpdateClient** : Mise à jour d'un client

Nous allons dans la suite de ce document faire une description de chaque méthode d'api.

2 GESTION DES TOKENS

Dans chaque appel d'api il va falloir générer un système de clés afin de pouvoir communiquer en sécurité. Ainsi, deux paramètres seront toujours présents : la clé publique et la clé hash.

Pour chaque appel d'api :

- Création d'une clé publique
 - Type MD5
 - Doit être unique à chaque appel
- Création d'une clé hash
 - Concaténation de plusieurs données :
 - Token privé
 - Fourni au préalable par GreenSpark
 - Clé publique
 - Date au format « Ymd » : 20200908

Dans SparkWine il est possible d'avoir plusieurs Token, pour chaque Token on peut définir un comportement et des valeurs par défaut différentes dans SparkWine :

Infos générales		A Trier	
Token	39PKQE153MOGBSG2RMLPC302YOK	Caractéristiques d'entrep	CLIENT
Articles		Lancement Projet	INTERNET
Code dépôt	DAUMAS 1 DG	Zone Commerciale	EUROPE
Clients		Union Economique	FRANCE & PARTICULIER
Type n° séquence	SEQUENCE TIERS	Libellé Mode Com.	COURRIER ET MAIL
Tarif	PRIVE	Régime fiscal	DROITS ACQUITTÉS
Devise	EUR	Firme	
Conditions paiement	COMPTANT	Secteur	PRIVE
Conditions Expédition	ENLEVEMENT CHAI	Devise	EUR
Mode de paiement	CARTE INTERNET CA	Classe rem Client	
Mode d'expédition	NON LIVRE	% Remise Client	0,00
Langue	FRANÇAIS	Dernière récupération	
Catégorie Client	CLIENT FRANCE	PersSign	
TVA	FRANCE 20%	JSAM	
Commercial	NON DEFINI		
Exprimé en TTC ?	☒		
Reste pmt/règle inv. int.	PAS DE RELANCE		
Commande			
Type Commande	VENTE PAR INTERNET		
N° campagne	35		
Update OH			

3 INPUT / OUTPUT

3.1 Paramètre Entrée

Dans chacune des API il y a donc toujours 2 paramètres minimum en entrée :

- HashKey [string]
- PublicKey [string]

*NB : Dans la suite de l'API nous n'allons pas les préciser pour chaque fonction nous allons simplement écrire : **TokenManagement***

3.2 Paramètre en sortie

Les sorties de chaque fonction peuvent avoir 2 formes :

- Chaque méthode retourne un objet avec le nom de la méthode + Responses (Il ne retourne toujours qu'un seul élément Responses)
- Si la méthode gère un retour multiple elle finit par un « s » et elle contient donc un Tableau avec les éléments dans l'objet de réponse
- Exemple :
 - Méthode a retour unique : GetArticle
 - Structure de GetArticleResponse
 - IsError : STRING
 - ErrorCode : INTEGER
 - ErrorMessage : STRING
 - Code : STRING
 - WebCode : STRING
 - Méthode a retour multiple : GetArticles
 - GetArticleResponse :
 - IsError : STRING
 - ErrorCode : INTEGER
 - ErrorMessage : STRING
 - Articles : Articles [] (Tableau avec les articles)
 - Code : STRING
 - WebCode : STRING

Il y a aussi toujours au minimum 3 paramètres pour la gestion des erreurs dans les objets XXXResponses

- IsError [string] → 0 ou 1
 - Erreur lors de l'appel
- ErrorCode [int]
 - Code d'erreur
- ErrorMessage [string]
 - Message de l'erreur

*NB : Dans la suite de l'API nous n'allons pas les préciser pour chaque fonction nous allons simplement écrire : **ErrorManagement***

Attention : Il faut donc à chaque appel regarder si le champ « IsError » de la Response est à 1 (Si c'est le cas il faut raise une erreur)

4.1 Articles

4.1.1 Récupération d'un article (GetArticle) - Obsolète

Remplacée par l'utilisation de GetArticles en spécifiant le code article

Permet de récupérer l'articles en fonction du code Web / SparkWine et s'il est coché « Disponible en ligne »

- Paramètres :
 - **TokenManagement**
 - Code [string]
 - Code Sparkwine (Unique dans SparkWine)
 - WebCode [string]
 - Code Web (Identifiant dans le site Web)
- GetArticleResponse :
 - **ErrorManagement**
 - Code [string]
 - Code Sparkwine
 - WebCode [string]
 - Code Web
 - Title [string]
 - Nom de l'article
 - Millesime
 - Millesime de l'article
 - Brand [string]
 - Marque de l'article
 - Color [string]
 - Couleur de l'article
 - Conditioning [integer]
 - Conditionnement de l'article (nombre de bouteilles)
 - Packaging [string]
 - Format du carton
 - Price [float]
 - Prix tarifs de l'article (HT ou TTC en fonction du paramétrage du Tarifs dans SparkWine)
 - ConditioningPrice [float]
 - Prix de l'article en fonction du conditionnement (Conditioning * Price)
 - IsSalable [string] → 0 ou 1
 - L'article est-il vendable ?
 - GrossWeight [float]
 - Poids de l'article
 - Stock [float]
 - Stock de l'article

4.1.2 Récupération de plusieurs articles (GetArticles)

Permet de récupérer tous les articles qui sont cochés « Disponible en ligne »

Possibilité de filtrer par date dans un intervalle pour récupérer seulement les articles modifiés durant la période saisie.

Un filtre technique peut être paramétrer dans Sparkwine pour ne récupérer que des articles avec un certain paramétrage

- Paramètres :
 - **TokenManagement**
 - **WebArticleId [string]**
 - **Code Web ou Sparkwine ou Vide pour tout récupérer**
 - **StartDate [DateTime]**
 - **Date début pour l'intervalle de modification**
 - **EndDate [Datetime]**
 - **Date de fin pour l'intervalle de modification**
- GetArticleResponse :
 - **ErrorManagement**
 - Articles [] → Tableau des articles :
 - Code [string]
 - Code Sparkwine
 - WebCode [string]
 - Code Web
 - Title [string]
 - Nom de l'article
 - Millesime
 - Millesime de l'article
 - Brand [string]
 - Marque de l'article
 - Color [string]
 - Couleur de l'article
 - Conditioning [integer]
 - Conditionnement de l'article (nombre de bouteilles)
 - Packaging [string]
 - Format du carton
 - **Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)**
 - IsSalable [string] → 0 ou 1
 - L'article est-il vendable ?
 - GrossWeight [float]
 - Poids de l'article
 - RegionCode [string]
 - Code région
 - RegionTitle [string]
 - Libellé région
 - CuveeCode [integer]
 - Code cuvée (ou sous-région)
 - CuveeTitle [string]

- Libellé cuvée (ou sous-région)
- AppellationCode [string]
 - Code Appellation
- AppellationTitle [string]
 - Libellé appellation
- GammeCode [string]
 - Code domaine
- GammeTitle [string]
 - Libellé domaine
- NcCode [string]
 - Nomenclature douanière
- Degre [float]
 - Degré
- Custom1Code [string]
 - Code champs personnalisé 1
- Custom1Title [string]
 - Libellé champs personnalité 1
- Custom3Code [string]
 - Code champs personnalisé 3
- Custom3Title [string]
 - Libellé champs personnalité 3
- Stock [float]
 - Stock de l'article
 - En fonction du cumul de stock choisis dans les paramètres du Token
- CepageCode [string]
 - Code cépage
- CepageTitle [string]
 - Libellé cépage
- PaysCode [string]
 - Code pays
- PaysTitle [string]
 - Libellé pays
- EANCode [string]
 - Code EAN
- GestionRegieCode [string]
 - Code Gestion régie
- GestionRegieTitle [string]
 - Libellé gestion régie
- TVACode [integer]
 - Code TVA
- TVATaux [float]
 - Taux de TVA
- VATTitle [string]
 - Libellé TVA

4.1.3 Mise à jour d'un article (UpdateArticle)

Permet d'associer un article web avec un article SparkWine

- Paramètres :
 - **TokenManagement**
 - Code [string]
 - Code Sparkwine
 - WebCode [string]
 - Code Web
- UpdateArticleResponse
 - **ErrorManagement**
 - Code [string]
 - Code Sparkwine
 - WebCode [string]
 - Code Web

4.1.4 Récupération des tarifs (GetTarifs)

Permet de récupérer tous les prix d'un article pour un tarif donné

Dans le cas d'un paramétrage VINIVRP, on ne voit que les tarifs de l'agent

- Paramètres :
 - **TokenManagement**
 - CodeTarif [integer]
 - Code Tarif
 - StartDate [DateTime]
 - Date début pour l'intervalle de modification
 - EndDate [Datetime]
 - Date de fin pour l'intervalle de modification
- GetTarifsResponse
 - **ErrorManagement**
 - Code [string]
 - Code Sparkwine
 - WebCode [string]
 - Code web
 - Price [float]
 - Prix du tarif
 - LowerLimit [float] [nullable]
 - Quantité seuil du tarif
 - IsTtc [boolean]
 - Le tarif est-il géré en TTC ?
 - AlternativeUnit [string] [nullable]
 - Unité alternative
 - AlternativeUnitFactor [float]
 - Facteur de conversion de l'unité (1 par défaut)
 - BaremeCode
 - Code du Barème associé au tarif
 - TypeValeurCode
 - Code Type de valeur
 - TypeValeurLib
 - Nom Type de valeur
 - TypeSeuilCode
 - Code type de seuil
 - TypeSeuilLib
 - Nom Type de seuil
 - BaseCalcCode
 - Code base de calcul
 - BaseCalcLib
 - Nom Base de calcul
 - Devise
 - La devise

4.2 Clients

4.2.1 Vérification qu'un client existe (DoesClientExist)

Recherche si un client existe dans SparkWine à l'aide de l'adresse mail

- Paramètres :
 - **TokenManagement**
 - Email [string]
 - Email du client
- DoesClientExistResponse :
 - **ErrorManagement**
 - ClientId [string]
 - Id SparkWine du client
 - Name [string]
 - Nom du client

4.2.2 Création d'un client (CreateClient)

Création d'un client

- Paramètres :
 - **TokenManagement**
 - WebClientId [string]
 - Id web du client
 - Email [string]
 - Email du client
 - Civility [integer]
 - Civilité
 - Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)
 - FirstName [string]
 - Prénom du client
 - LastName [string]
 - Nom du client
 - BirthDate [date]
 - Date d'anniversaire
 - LastUpdate [date]
 - Date de mise à jour
 - WebCreationDate
 - Date de création sur le WEB
 - SiretNumber [string]
 - Numéro de Siret
 - IntraComVat [string]
 - Numéro tva intracom
 - Gestion Adresse :
 - AddressLine1 [string]
 - Ligne 1 de l'adresse
 - AddressLine2 [string]
 - Ligne 2 de l'adresse
 - Building [string]
 - Infos batiment
 - PostalCode [string]
 - Code Postal
 - City [string]
 - Ville
 - Country [string]
 - Pays
 - Valeur possible : Code Pays ISO
 - PhoneNumber [string]
 - Numéro de téléphone (De préférence Portable)
- CreateClientResponse :
 - **ErrorManagement**
 - ClientId [string]
 - Id SparkWine du client
 - WebClientId [string]
 - Id web du client

4.2.3 Création d'un client (UpdateClient)

Mise à jour d'un client

- Paramètres :
 - **TokenManagement**
 - WebClientId [string]
 - Id web du client
 - Email [string]
 - Email du client
 - Civility [integer]
 - Civilité
 - Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)
 - FirstName [string]
 - Prénom du client
 - LastName [string]
 - Nom du client
 - BirthDate [date]
 - Date d'anniversaire
 - LastUpdate [date]
 - Date de mise à jour
 - WebCreationDate
 - Date de création sur le WEB
 - SiretNumber [string]
 - Numéro de Siret
 - IntraComVat [string]
 - Numéro tva intracom
 - Gestion Adresse :
 - AddressLine1 [string]
 - Ligne 1 de l'adresse
 - AddressLine2 [string]
 - Ligne 2 de l'adresse
 - Building [string]
 - Infos bâtiment
 - PostalCode [string]
 - Code Postal
 - City [string]
 - Ville
 - Country [string]
 - Pays
 - Valeur possible : Code Pays ISO
 - PhoneNumber [string]
 - Numéro de téléphone (De préférence Portable)
- UpdateClientResponse :
 - **ErrorManagement**
 - ClientId [string]
 - Id SparkWine du client
 - WebClientId [string]
 - Id web du client

4.2.4 Récupération d'un client (GetClient)

Récupération d'un client :

- Paramètres :
 - **TokenManagement**
 - WebClientId [string]
 - Id web du client
 - Type : String
- GetClientResponse :
 - **ErrorManagement**
 - ClientId [string]
 - Id SparkWine du client
 - WebClientId [string]
 - Id web du client
 - Name [string]
 - Nom du client
 - Email [string]
 - Email du client
 - BirthDate [date]
 - Date d'anniversaire
 - SiretNumber [string]
 - Numéro de Siret
 - IntraComVat [string]
 - Numéro tva intracom
 - IsInactive [string] ➔ 0 ou 1
 - Le client est-il inactif

4.2.5 Récupération de plusieurs clients (GetClients)

Récupération de plusieurs clients :

- Paramètres :
 - **TokenManagement**
 - WebClientId [string]
 - Id web du client
 - Type : String
 - Ou null pour tout récupérer
- GetClientsResponse :
 - **ErrorManagement**
 - ClientId [string]
 - Id SparkWine du client
 - WebClientId [string]
 - Id web du client
 - Name [string]
 - Nom du client
 - Email [string]
 - Email du client
 - BirthDate [date]
 - Date d'anniversaire
 - SiretNumber [string]
 - Numéro de Siret
 - IntraComVat [string]
 - Numéro tva intracom
 - IsInactive [string] → 0 ou 1
 - Le client est-il inactif
 - WebGroupClient [integer]
 - Groupe Web
 - Adresse [string]
 - Adresse
 - CpltAdresse[string]
 - Complément d'adresse
 - InfosBat [string]
 - Information bâtiment
 - IdRelais [string]
 - ID point relais
 - CP [string]
 - Code postal
 - Commune [string]
 - Nom commune
 - Ville [string]
 - Nom ville
 - PaysCode [string]
 - Code pays
 - PaysTitle [string]
 - Nom pays
 - TVACode [integer]
 - Code TVA
 - TVATaux [float]

- Taux TVA
- TVATitle [string]
 - Nom TVA
- ModeDePaiementCode [string]
 - Code du mode de paiement
- ModeDePaiementTitle [string]
 - Nom du mode de paiement
- CondPaiementCode [string]
 - Code des conditions de paiement
- CondPaiementTitle [string]
 - Nom des conditions de paiement
- CategorieCode [integer]
 - Code catégorie client
- CategorieTitle [string]
 - Nom catégorie client
- EncoursCommande [float]
 - Montant Encours commande
- EncoursComptable [float]
 - Montant Encours comptable
- EncoursTotal [float]
 - Montant encours totaux
- SecteurComCode [string]
 - Code secteur com.
- SecteurComTitle [string]
 - Nom secteur com.
- MarcheComCode [string]
 - Code marché com.
- MarcheComTitle [string]
 - Nom marché com.
- RegionComLib [string]
 - Code région Com.
- RegionComTitle [string]
 - Nom région com
- ModeComCode [integer]
 - Code Mode Com.
- ModeComTitle [string]
 - Nom mode com.
- TelMobile [string]
 - Numéro de téléphone mobile
- TelFixe [string]
 - Numéro de téléphone fixe
- Email [string]
 - Adresse Mail
- CommercialCode [string]
 - Code commercial
- CommercialTitle [string]
 - Nom commercial
- PerRemise [float]
 - Pourcentage remise
- TarifCode [integer]
 - Code Tarif
- TarifTitle [string]

- Nom tarif
- LangueCode [integer]
 - Code langue
- LangueTitle [string]
 - Nom Langue

4.2.6 Récupération de plusieurs Contacts (GetContacts)

Récupération de plusieurs Contacts :

- Paramètres :
 - **TokenManagement**
 - WebClientId [string]
 - Id web du client
 - Type : String
 - Ou null pour tout récupérer
- GetContactsResponse :
 - **ErrorManagement**
- ContactCode [Integer]
 - Numéro de contact
- LastName [String]
 - Nom
- FirstName [String]
 - Prénom
- Name [String]
 - Nom complet
- CiviliteCode [Integer]
 - Code Civilité
- CiviliteTitle [String]
 - Nom civilité
- Email [String]
 - Adresse mail
- Mobile [String]
 - Numéro téléphone mobile
- Fixe [String]
 - Numéro téléphone Fixe
- ServiceCode [String]
 - Code service
- ServiceLib [String]
 - Nom service

4.2.7 Modification mail client (UpdateEmail)

- Paramètres :
 - **TokenManagement**
 - ClientId [string]
 - Id SparkWine du client
 - WebClientId [string]
 - Id web du client
 - Email [string]
 - Nouveau email du client
- UpdateEmail Response :
 - **ErrorManagement**
 - ClientId [string]
 - Id SparkWine du client
 - WebClientId [string]
 - Id web du client

4.3 Commandes

Important :

Le système de création de commande repose sur 3 fonctions à appeler dans l'ordre

1. CreateOrder : Création de l'entête de commande (Une fois par commande)
2. CreateOrderLine : Création ligne de commande
 - a. Autant de fois que nécessaire
 - b. Pareil pour générer un article de remise ou un article de transport
3. ValidateOrder : Quand on a terminé d'insérer des lignes (Une fois par commande)

Les commandes passent par un SAS et donc ne sont pas réellement générées dans l'ERP au moment du Validate Order, la validation peut soit être automatique soit manuelle, mais si elle bloque l'API n'est pas impacté.

4.3.1 Création d'une commande (CreateOrder)

Création d'une commande entête de commande

- Paramètres :
 - **TokenManagement**
 - WebOrderId [string]
 - Id web de la commande
 - Type : String
 - WebClientId [string]
 - Id web du client
 - Type : String
 - OrderDate [date]
 - Date de la commande
 - Gestion Adresse :
 - AddressAlias [string]
 - Nom de l'adresse
 - AddressLine1 [string]
 - Ligne 1 de l'adresse
 - AddressLine2 [string]
 - Ligne 2 de l'adresse
 - Building [string]
 - Infos batiment
 - PostalCode [string]
 - Code Postal
 - City [string]
 - Ville
 - Country [string]
 - Pays
 - Valeur possible : Code Pays ISO
 - PhoneNumber [string]
 - Numéro de téléphone (De préférence Portable)
 - DeliveryComment [string]
 - Commentaire de livraison
- CreateOrderResponse :
 - **ErrorManagement**
 - WebOrderId [string]
 - Id web de la commande
 - OrderNumber [string]
 - Id commande (N° temporaire de la commande, car les commandes

4.3.2 Création d'une ligne de commande (CreateOrderLine)

Création d'une ligne de commande

- Paramètres :
 - **TokenManagement**
 - WebOrderId [string]
 - Id web de la commande
 - WebArticleId [string]
 - Id de l'article SparkWine
 - UnitPriceET [float]
 - Prix Unitaire HT
 - UnitPriceIT [float]
 - Prix Unitaire TTC
 - DiscountAmount [float]
 - Montant des remises
 - DiscountPercent [float]
 - Pourcentage de remise
 - Quantity [float]
 - Quantité
 - AmountET [float]
 - Montant HT
 - AmountIT [float]
 - Montant TTC
 - PromotionCode [string]
 - Code du coupon
 - PromotionDescription [string]
 - Descriptif du coupon
 - PromotionValidityDate [date]
 - Date de validité
- CreateOrderLineResponse :
 - **ErrorManagement**
 - WebOrderId [string]
 - Id web de la commande
 - OrderNumber [string]
 - Id commande

4.3.3 Validation de la commande (ValidateOrder)

Validation de la commande

- Paramètres :
 - WebOrderId [string]
 - Id web de la commande
 - Type : String
 - AmountET [float]
 - Montant Total HT
 - AmountIT [float]
 - Montant Total TTC
 - PaymentMethod [string]
 - Méthode de paiement
 - **Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)**
 - PaymentId [string]
 - Id de la transaction
 - Carrier [string]
 - Transporteur
 - Valeur Possible :
 - WebAddressId [string]
 - Id de l'adresse de livraison à utiliser
 - ShippingMethod [integer]
 - Mode d'expédition
 - **Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)**
 - ShippingCondition [integer]
 - Condition d'expédition
 - **Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)**
 - Reference [string]
 - Référence de la commande
- ValidateOrderResponse :
 - WebOrderId [string]
 - Id web de la commande
 - OrderNumber [string]
 - Id commande

4.3.4 Récupération (GetOrder)

Récupération d'informations sur la commande

- Paramètres :
 - **TokenManagement**
 - WebOrderId [string]
 - Id web de la commande
- GetOrderResponse :
 - **ErrorManagement**
 - WebOrderId [string]
 - Id web de la commande
 - OrderNumber [string]
 - Id commande
 - OrderStatus [integer]
 - Statut « ADV » de la commande
 - **Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)**
 - Par Défaut :
 - 10 - ATTENTE VALIDATION
 - 13 - VALIDÉ
 - 20 - EXPEDITION PARTIELLE
 - 40 - EN COURS D'EXPÉDITION
 - 50 - PRET A FACTURER
 - 70 - FACTURÉ
 - 90 - ANNULÉ
 - PackageStatus [string]
 - Statut de préparation
 - **Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)**
 - PackageCount [integer]
 - Nombre de colis / cartons
 - InvoiceNumber [string]
 - Numéro de facture
 - InvoiceDate [date]
 - Date de facture
 - InvoiceDueDate [date]
 - Date d'échéance
 - DateExpedPrev [date]
 - Date d'expédition planifiée dans SparkWine
 - Transporteur [string]
 - Transporteur
 - **Valeur possible : Dépend de l'environnement (Voir avec le client dans SparkWine)**
 - TrackingNumber [string]
 - Tracking number de SparkWine (Pour l'instant le champ est vide)

4.3.5 DashBoard – Commande

Le DashBoard est un SAS par lequel les commandes passent avant d'entrer comme des commandes réelles.

Il se compose d'une liste avec les commandes en attente de validation :

Commande [5]	Code Web	N° Cde	Etat Commande	Code tiers	Lib Tiers	Con	Montant HT	Montant TTC	Lib Tiers	Client livré	Création clien	Web client id
▶ WEB-ORDER-7757			VALIDÉ	10083387		☐	75,83	910,00	DE SCHENKER		20/11/2021	WEB-CLIENT-5508
WEB-ORDER-7815			VALIDÉ	10083337		☐	75,83	80,00	DAUMAS PRODUCTEUR		25/11/2021	WEB-CLIENT-5529
WEB-ORDER-7820			VALIDÉ	10062563		☐	75,83	80,00			20/09/2018	WEB-CLIENT-5534
WEB-ORDER-7823			VALIDÉ	10036928		☐	151,66	160,00	DE SCHENKER		19/07/2018	WEB-CLIENT-904
WEB-ORDER-7825			VALIDÉ	10023385		☐	75,83	80,00	DE SCHENKER		19/07/2018	WEB-CLIENT-4785

Ensuite pour chaque ligne on va donc pouvoir consulter et valider les commandes : (La notion de validation va donc créer une commande SparkWine Réel)

Général | Synthèse | Tous les champs

Infos générales

Code web
WEB-ORDER-7815
N° pré-commande
25/11/2021 22:00:03:637206492
Commande validée ? ☐

Infos commande

N° Commande
Date Commande
25/11/2021
Type Commande
VENTE PAR INTERNET
Mode de paiement
CARTE INTERNET CA
Conditions paiement
COMPTANT
N° campagne
PRIVE 2021
% Remise Commande
0,00
Remise en Valeur
0,00
Montant HT
75,83
Montant TTC
80,00
Date expéd. prévue
25/11/2021

Infos client

Entreprise
10083337
Libellé Tiers
Création client
25/11/2021 [Changer client](#)
Web client id
WEB-CLIENT-5529
Langue
FRANÇAIS

Infos livraison

Nom destinataire
Adresse livraison 1
Adresse livraison 2
Informations bâtiment
CP/VILLE
Pays de livraison
Tél. mobile
Em@il

Transport

Produit Transporteur
MANQUANT : 01
Mode d'expédition
ROUTE
Conditions Expédition
PORT PAYE
Transporteur
DAUMAS PRODUCTEUR

Notes

Erreurs

Retraitement Fait

Actions

[Valider](#)

Ligne de commande [2]

Code article	Lib. Article	Quantité	Prix HT	Prix TTC	% Rem.	Monta
▶ PREST-009	SAUMON FUME BIO	1	75,83	80,00	0,00	
PORT	FRAIS DE PORT	1	16,67	20,00	0,00	

Mise à jour client

DteCréa	Type mise	Validé?	Code tiers	Lib Tiers
▶		☐		

5 DEBUTER AVEC L'API (PHP)

GreenSpark doit vous fournir un lien de ce genre (accès au WSDL) : <http://xxxxxx:9000/jsb-ws/SparkWeb?wsdl>

5.1 Régénérer le dossier API (si besoin)

Le lancement de index.php (avec le bon lien WSDL) permet de régénérer la structure des appels API en PHP (si GreenSpark livre une nouvelle un correctif ou mise à jour de version).

```
<?php
```

```
require 'autoload.php';
$generator = new \Wsd12PhpGenerator\Generator();
$generator->generate(
    new \Wsd12PhpGenerator\Config(array(
        'inputFile' => 'http://gspk-dev.greenspark.fr:9000/jsb-ws/SparkWeb?wsdl',
        'outputDir' => 'api'
    ))
);
```

5.2 Configuration du service

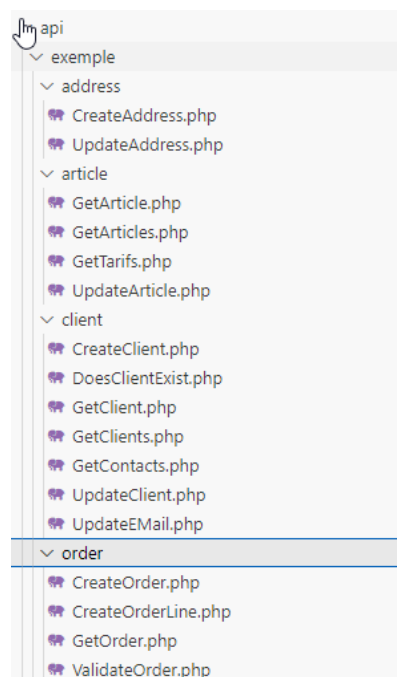
Pour appeler le bon service ou le bon serveur il faut configurer API/SparkWebService.php

(Dans le constructeur)

```
52 | */
53 | public function __construct(array $options = array(), $wsdl = null)
54 | {
55 |     foreach (self::$classmap as $key => $value) {
56 |         if (!isset($options['classmap'][$key])) {
57 |             $options['classmap'][$key] = $value;
58 |         }
59 |     }
60 |     $options = array_merge(array (
61 |         'features' => 1,
62 |     ), $options);
63 |     if (!$wsdl) {
64 |         $wsdl = 'http://gspk-dev.greenspark.fr:9000/jsb-ws/SparkWeb?wsdl';
65 |     }
```

5.3 Exemples de codes

Le dossier API contient les appels aux fonctions de l'API et le dossier api/exemple contient des exemples complet d'utilisations de l'API.



6 GESTION DES ERREURS

Code	Message
400	Token non reconnu
500	Erreur Interne
501	L'article n'existe pas
510	Le client existe déjà
511	Le client n'existe pas
512	L'adresse n'existe pas
513	L'adresse n'existe pas
520	La commande existe déjà
521	La commande n'existe pas