Réussir l'intégration de l'API Formulaire

Guide d'implémentation

Version du document 1.9

SOCIETE GENERALE

DEVELOPPONS ENSEMBLE
L'ESPRIT D'EQUIPE

Sommaire

1. HISTORIQUE DU DOCUMENT

Version	Auteur	Date	Commentaire
1.9	Société Générale	20/11/2020	• Mise à jour du chapitre *Envoyer un formulaire de paiement en POST*.
1.8	Société Générale	30/07/2020	• Correction du format du champ **vads_trans_date**. • Mise à jour du chapitre *Configurer la notification à la fin du paiement*.
1.7	Société Générale	09/12/2019	• Mise à jour du chapitre **Procéder à la phase de test**. • Mise à jour de la procédure de configuration de l'IPN. • Ajout du chapitre **Implémenter l'IPN**. • Correction du format du champ **vads_product_label**. • Modification du format du champ **vads_trans_id**
1.6	Société Générale	17/06/2019	L'algorithme de hachage est désormais disponible dans le menu ParamétrageBoutique, onglet Clés.
1.5	Société Générale	23/01/2019	• Mise à jour du chapitre **S'identifier lors des échanges** • Remplacement du terme "Certificat" par "Clé" dans tous les menus
1.4	Société Générale	04/09/2018	• Mise à jour de chapitre **Calculer la signature**.
1.3	Société Générale	26/06/2018	• Mise à jour de chapitre **Calculer la signature**. • Nouvelle valeur du champ *vads_trans_status* : **SUSPENDED** • Mise à jour du chapitre **Envoyer un formulaire de paiement en POST**
1.2	Société Générale	23/05/2018	Possibilité de choisir l'algorithme de calcul de signature (SHA-1 ou SHA-256)
1.1	Société Générale	14/11/2017	• Mise à jour des exemples. • Ajout d'exemples de calcul de signature en Java et PHP.
1.0	Société Générale	10/01/2017	Version initiale

2. ÉTABLIR LE DIALOGUE AVEC LA PLATEFORME DE PAIEMENT

Le dialogue entre le site marchand et la plateforme de paiement s'effectue par un échange de données.

Pour créer un paiement, ces données sont envoyées au moyen d'un formulaire HTML via le navigateur de l'acheteur.

A la fin du paiement, le résultat est transmis au site marchand de deux manières :

- par le navigateur lorsque l'acheteur clique sur le bouton pour revenir au site marchand.
- automatiquement au moyen de notifications appelées URL de notification instantanée (également appelée IPN pour Instant Payment Notification) voir chapitre **Configurer la notification à la fin du paiement**.

Pour assurer la sécurité des échanges, les données sont signées au moyen d'une clé connue uniquement du marchand et de la plateforme de paiement.

2.1. Définir l'URL de la page de paiement

Le site marchand communique avec la plateforme de paiement en redirigeant l'acheteur vers l'URL ci-dessous.

https://sogecommerce.societegenerale.eu/vads-payment/

2.2. S'identifier lors des échanges

Pour dialoguer avec la plateforme de paiement, le marchand a besoin de deux informations :

- **L'identifiant boutique** : permet d'identifier le site marchand durant les échanges. Sa valeur est transmise dans le champ **vads_site_id.**
- **La clé**: permet de calculer la signature alphanumérique transmise dans le champ **signature**.

Pour récupérer ces valeurs :

1. Connectez-vous à votre Back Office Marchand : *https://sogecommerce.societegenerale.eu/vads-merchant/*

2. Saisissez votre identifiant de connexion.

   Votre identifiant de connexion vous a été communiqué par e-mail ayant pour objet **Identifiants de connexion - [nom de votre boutique]**.

3. Saisissez votre mot de passe.

   Votre mot de passe vous a été communiqué par e-mail ayant pour objet **Identifiants de connexion - [nom de votre boutique]**.

4. Cliquez sur **Valider**.

   Au bout de 3 erreurs dans la saisie du mot de passe, le compte de l'utilisateur est bloqué. Cliquez alors sur **Mot de passe oublié ou compte bloqué** pour réinitialiser.

   Le mot de passe d'un utilisateur a une durée de validité de 90 jours. Au-delà de cette durée, un renouvellement sera demandé lors de la connexion.

5. Cliquez sur **Paramétrage > Boutique**.

6. Sélectionnez l'onglet **Clés**.

Image 1 : Onglet Clés

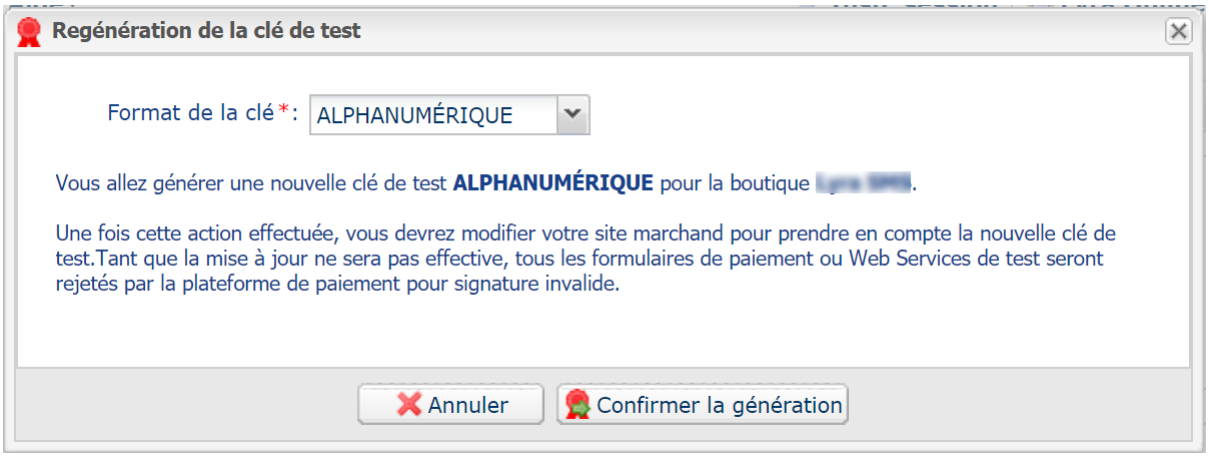
Deux types de clé sont mis à disposition :

- La **clé de test** qui permet de générer la signature d'un formulaire en mode test.
- La **clé de production** qui permet de générer la signature d'un formulaire en mode production.

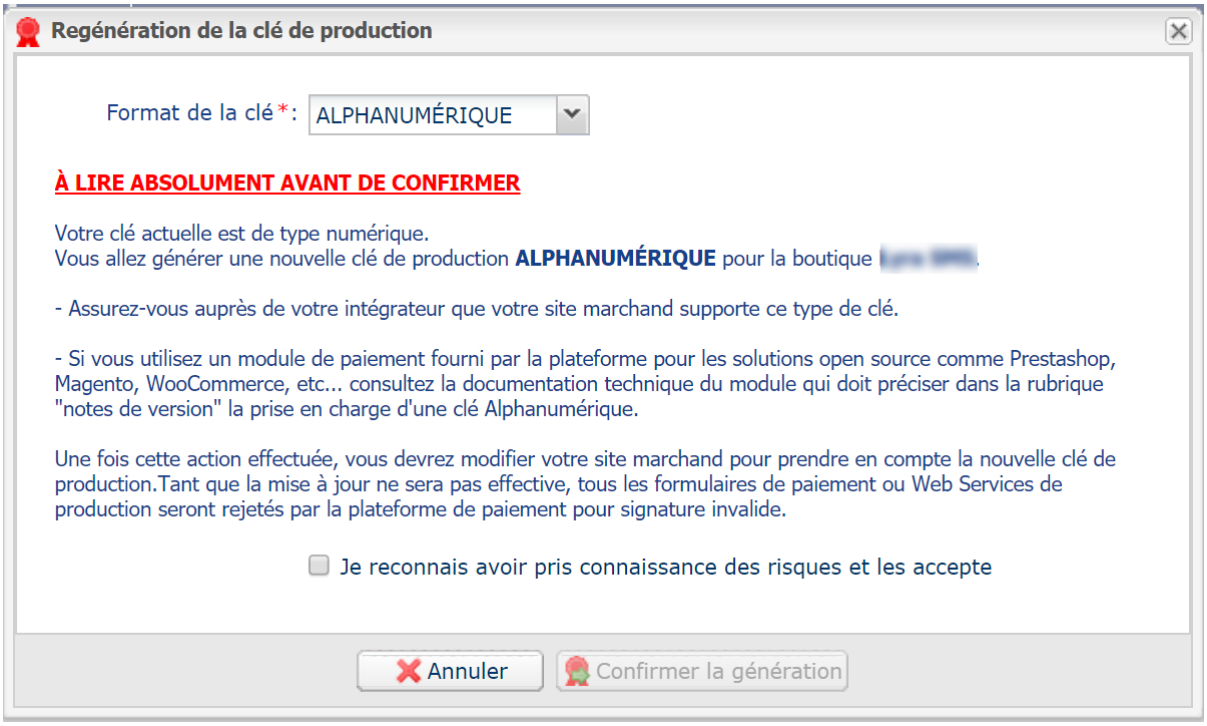
Ces clés peuvent être numériques ou alphanumériques.

Pour un maximum de sécurité, il est recommandé d'utiliser une clé alphanumérique.

Pour changer le format de votre clé de test, cliquez sur le bouton **Régénérer une clé de test**, puis sélectionnez le format ("ALPHANUMERIQUE" ou "NUMERIQUE").



Pour changer le format de votre clé de production, cliquez sur le bouton **Régénérer une clé de production**, puis sélectionnez "ALPHANUMERIQUE" ou "NUMERIQUE").


Regénération de la clé de production
Format de la clé *: ALPHANUMÉRIQUE
À LIRE ABSOLUMENT AVANT DE CONFIRMER
Votre clé actuelle est de type numérique.
Vous allez générer une nouvelle clé de production ALPHANUMÉRIQUE pour la boutique
- Assurez-vous auprès de votre intégrateur que votre site marchand supporte ce type de clé.
- Si vous utilisez un module de paiement fourni par la plateforme pour les solutions open source comme Prestashop, Magento, WooCommerce, etc... consultez la documentation technique du module qui doit préciser dans la rubrique "notes de version" la prise en charge d'une clé Alphanumérique.
Une fois cette action effectuée, vous devrez modifier votre site marchand pour prendre en compte la nouvelle clé de production.Tant que la mise à jour ne sera pas effective, tous les formulaires de paiement ou Web Services de production seront rejetés par la plateforme de paiement pour signature invalide.
Je reconnais avoir pris connaissance des risques et les accepte
Annuler
Confirmer la génération

2.3. Gérer le dialogue vers le site marchand

La gestion du dialogue vers le site marchand est réalisée grâce à deux types d'URL :

- **URL de notification instantanée**, également appelée IPN (Instant Payment Notification),
- **URL de retour** vers le site marchand.

URL de notification instantanée - IPN (Instant Payment Notification)

L'**URL de notification** est l'URL d'une page dédiée sur le site marchand appelée **automatiquement** par la plateforme de paiement lorsque des événements particuliers se produisent.

Par défaut des règles sont créées pour gérer les événements ci-dessous :

- fin d'un paiement (accepté ou refusé),
- abandon ou annulation durant le paiement,
- création ou mise à jour d'un alias,
- création d'un abonnement,
- nouvelle échéance d'un abonnement,
- autorisation réalisée dans le cas d'un paiement différé,
- modification du statut d'une transaction par l'acquéreur,
- opération réalisée depuis le Back Office Marchand (annulation, remboursement, duplication, paiement manuel, etc..).

Ces règles doivent être activées et convenablement configurées en fonction des besoins du marchand.

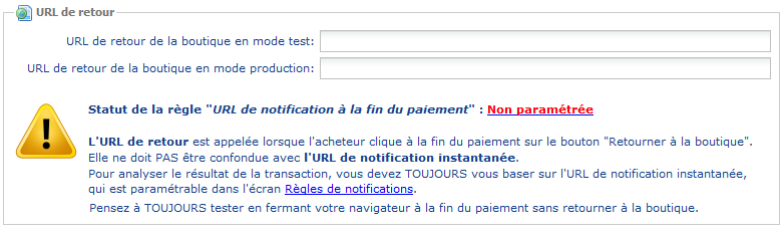
A chaque appel, la plateforme de paiement transmet au site marchand les données relatives à une transaction. C'est ce qu'on appelle une notification instantanée (ou **IPN** pour Instant Payment Notification).

Pour assurer la sécurité des échanges, les données sont signées au moyen d'une clé connue uniquement du marchand et de la plateforme de paiement.

URL de retour vers le site marchand

Le marchand peut paramétrer dans le Back Office Marchand les URL de retour "par défaut" depuis le menu **Paramétrage** > **Boutique** > onglet **Configuration** :

Image 2 : Spécification des URL de retour



Il peut configurer une URL de retour à la boutique différente en fonction du mode.

Par défaut, l'acheteur est redirigé vers l'URL de retour, et ce, quel que soit le résultat du paiement.

Si toutefois aucune URL n'est configurée à ce niveau, alors la redirection utilisera l'URL principale de la boutique (paramètre **URL** défini dans l'encadré **Détails** de la boutique).

Le marchand a la possibilité de surcharger cette configuration dans son formulaire de paiement (voir chapitre **Définir les URL de retour**).

Le statut de la règle "URL de notification à la fin du paiement" (IPN) est affiché dans cet écran. Si cette dernière est non paramétrée, veillez à la renseigner (voir chapitre **Paramétrer les notifications**).

2.4. Gérer la sécurité

Plusieurs moyens sont mis en place afin d'assurer la sécurité des transactions de paiement en ligne.

2.4.1. Garantir l'intégrité des échanges

L'intégrité des informations échangées est garantie par un échange de signatures alphanumériques entre la plateforme de paiement et le site marchand.

Le dialogue entre la plateforme de paiement et le site marchand s'effectue par soumission de formulaires HTML.

Un formulaire contient une liste de champs spécifiques (voir chapitre **Générer un formulaire de paiement**) utilisés pour générer une chaîne.

Cette chaîne est ensuite convertie en une chaîne d'une taille inférieure grâce à une fonction de hachage (SHA-1, HMAC-SHA-256).

*Le marchand pourra choisir l'algorithme de hachage dans son Back Office Marchand (voir chapitre **Sélectionner l'algorithme de hachage**).*

La chaîne résultante est appelée **empreinte** (*digest* en anglais) de la chaîne initiale.

L'empreinte doit être transmise dans le champ **signature** (voir chapitre **Calculer la signature**).

Modélisation des mécanismes de sécurité :

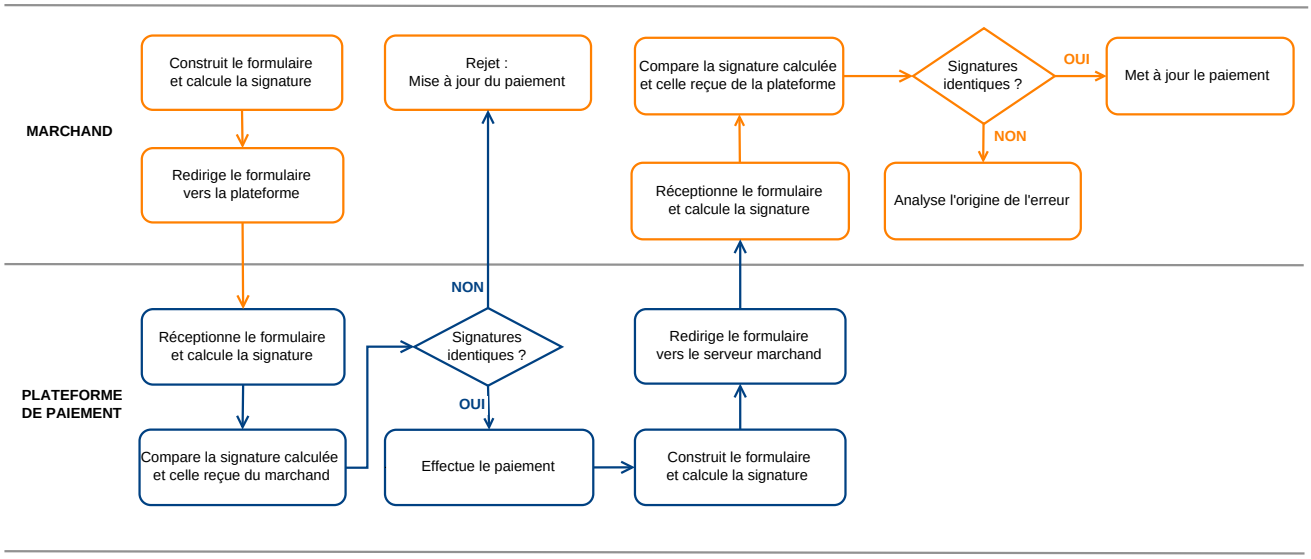


Image 3 : Diagramme mécanisme de sécurité

1. Le site marchand construit les données du formulaire et calcule la signature.
2. Le site marchand envoie le formulaire à la plateforme.
3. La plateforme réceptionne les données du formulaire et calcule la signature avec les données reçues.
4. La plateforme compare la signature calculée avec la signature transmise par le site marchand.
5. Si les signatures diffèrent, la demande de paiement est rejetée.

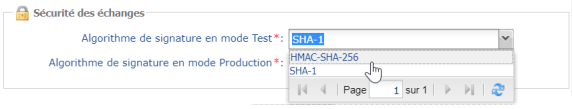
   Sinon, la plateforme procède au paiement.

6. La plateforme construit les données de la réponse et calcule la signature de la réponse.
7. En fonction du paramétrage de la boutique (voir chapitre **Paramétrer les notifications**), la plateforme transmet le résultat du paiement au site marchand.
8. Le site marchand réceptionne les données et calcule la signature. Il compare la signature calculée avec la signature transmise par la plateforme.
9. Si les signatures diffèrent, le marchand analyse l'origine de l'erreur (erreur dans le calcul, tentative de fraude etc.)

   Sinon, le site marchand procède à la mise à jour de sa base de données (état du stock, statut de la commande etc.).

2.4.2. Sélectionner l'algorithme de hachage

Depuis le Back Office Marchand (menu **Paramétrage** > **Boutique** > **Clés**), le marchand a la possibilité de choisir la fonction de hachage à utiliser pour générer les signatures.



Par défaut, c'est l'algorithme HMAC-SHA-256 qui sera appliqué.

Vous pouvez sélectionner un algorithme différent pour le mode Test et pour le mode Production.

Veillez cependant à utiliser la même méthode pour générer vos formulaires de paiement et pour analyser les données transmises par la plateforme de paiement lors des notifications.

Afin de faciliter le changement d'algorithme, les signatures en SHA-1 ou en HMAC-SHA-256 seront acceptées sans générer de rejet pour erreur de signature pendant 24h.

2.4.3. Conserver la clé de production

Dès le premier paiement réalisé avec une carte réelle, la clé de production est masquée pour des raisons de sécurité.

Nous vous conseillons fortement de conserver cette clé en lieu sûr (fichier chiffré, base de données etc.).

En cas de perte, le marchand aura la possibilité d'en générer une nouvelle depuis son Back Office Marchand.

Pour rappel, la clé de production est visible dans le Back Office Marchand depuis le menu **Paramétrage** > **Boutique** > onglet **Clés**.

2.4.4. Gérer les données sensibles

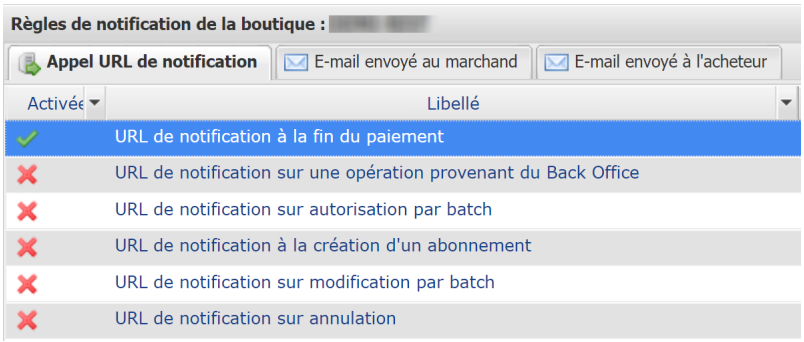
Des règles strictes régissent les transactions de paiement en ligne (Certification PCI-DSS).

En tant que marchand, vous devez vous assurer de ne jamais retranscrire en clair des données qui pourraient s'apparenter à un numéro de carte bancaire. Votre formulaire serait rejeté (code 999 - Sensitive data detected).

Evitez notamment les numéros de commandes de longueur comprise entre 13 et 16 caractères numériques et commençant par 3, 4 ou 5.

3. PARAMÉTRER LES NOTIFICATIONS

Pour accéder à la gestion des règles de notification, ouvrez le menu : **Paramétrage** > **Règles de notifications**.



L'onglet de configuration des règles de type "Appel URL de notification" s'affiche.

3.1. Configurer la notification à la fin du paiement

Cette règle permet de notifier le site marchand dans les cas suivants :

- Paiement accepté
- Paiement refusé
- Création ou mise à jour d'un alias
- Création d'un abonnement

L'événement **Paiement accepté** correspond à la création d'une transaction dans l'un des statuts (**vads_trans_status**) ci-dessous:

- **ACCEPTED**
- **AUTHORISED**
- **AUTHORISED_TO_VALIDATE**
- **CAPTURED**
- **INITIAL**
- **UNDER_VERIFICATION**
- **WAITING_AUTHORISATION**
- **WAITING_AUTHORISATION_TO_VALIDATE**
- **WAITING_FOR_PAYMENT**

Cette notification est indispensable pour communiquer le résultat d'une demande de paiement.

Elle informera le site marchand du résultat du paiement même si l'acheteur ne clique pas sur le bouton Retour à la boutique.

1. Effectuez un clic droit sur la ligne **URL de notification à la fin du paiement**.
2. Sélectionnez **Gérer la règle**.
3. Dans la section **Paramétrage général**, renseignez le champ **Adresse(s) e-mail(s) à avertir en cas d'échec**.

Pour spécifier plusieurs adresses e-mails, séparez-les par un point-virgule.

4. Cochez la case **Rejeu automatique en cas d'échec** si vous souhaitez autoriser la plateforme à renvoyer automatiquement la notification en cas d'échec, et ce, jusqu'à 4 fois.

   Pour plus d'informations, reportez-vous au chapitre *Rejeu automatique en cas d'échec* à la page 12.

5. Dans la section **URL de notification de l'API formulaire V1, V2**, renseignez l'URL de votre page dans les champs **URL à appeler en mode TEST** et **URL à appeler en mode PRODUCTION**.

6. Sauvegardez vos modifications.

3.2. Rejeu automatique en cas d'échec

Le rejeu automatique ne s'applique pas aux notifications déclenchées manuellement depuis le Back Office Marchand.

Le marchand peut activer un mécanisme qui permet à la plateforme de paiement de renvoyer automatiquement les notifications lorsque le site marchand est ponctuellement injoignable, et ce, **jusqu'à 4 fois**.

Une notification sera considérée en échec si le code retour HTTP retourné par le site marchand ne fait pas partie de la liste suivante: **200**, **201**, **202**, **203**, **204**, **205**, **206**, **301**, **302**, **303**, **307**, **308**.

Les tentatives d'appel sont programmées à heures fixes toutes les 15 minutes (00, 15, 30, 45).

Après chaque tentative infructueuse, un e-mail d'alerte est envoyé à l'adresse spécifiée dans la configuration de la règle de notification concernée.

L'objet de l'e-mail d'alerte contient le numéro de la tentative d'envoi de la notification. Il est présenté sous la forme `attempt #` suivi du numéro de tentative.

- Exemple d'objet d'un e-mail d'alerte reçu suite au premier échec de notification à la fin d'un paiement :

```
[MODE TEST] Ma Boutique - Tr. réf. 067925 / ECHEC lors de l'appel de votre URL de
 notification [unsuccessful attempt #1]
```

- Exemple d'objet d'e-mail reçu lors d'un deuxième échec :

```
[MODE TEST] Ma Boutique - Tr. réf. 067925 / ECHEC lors de l'appel de votre URL de
 notification [unsuccessful attempt #2]
```

- Exemple d'objet d'e-mail reçu lors d'un troisième échec :

```
[MODE TEST] Ma Boutique - Tr. réf. 067925 / ECHEC lors de l'appel de votre URL de
 notification [unsuccessful attempt #3]
```

- Exemple d'objet d'e-mail reçu lors de la dernière tentative :

```
[MODE TEST] Ma Boutique - Tr. réf. 067925 / ECHEC lors de l'appel de votre URL de
 notification [unsuccessful attempt #last]
```

  Pour notifier au site marchand l'échec de la dernière tentative de notification, l'objet de l'e-mail comportera la mention `attempt #last`.

Lors du rejeu automatique, certaines informations ne sont pas enregistrées en base de données ou sont modifiées.

Exemples de champs non disponibles / non enregistrés en base de données :

Nom du champ	Description
vads_page_action	Opération réalisée
vads_payment_config	Typologie de paiement (comptant ou en plusieurs échéances)
vads_action_mode	Mode d'acquisition des informations du moyen de paiement

Exemples de champs envoyés avec des valeurs différentes :

Nom du champ	Nouvelle valeur
vads_url_check_src	Toujours valorisé à **RETRY** lors d'un rejeu automatique.
vads_trans_status	Le statut de la transaction peut varier entre l'appel initial et le rejeu automatique (annulation du marchand, remise en banque de la transaction, etc.).
vads_hash	La valeur de ce champ est regénérée à chaque appel.
signature	La valeur de la signature dépend des différents statuts qui peuvent varier entre l'appel initial et le rejeu automatique.

Ces e-mails détaillent:

- le problème rencontré
- des éléments d'analyse en fonction de l'erreur
- ses conséquences
- la procédure à suivre depuis le Back Office Marchand pour déclencher manuellement la notification.

Après la quatrième tentative, il est toujours possible de rejouer l'URL de notification **manuellement** depuis votre Back Office Marchand.

Attention, pendant la période de rejeu automatique, tout appel manuel à l'URL de notification influera sur le nombre de tentatives automatiques :

- un appel manuel réussi provoquera l'arrêt du rejeu automatique
- un appel manuel en échec n'aura aucun impact sur le rejeu automatique en cours.

3.3. Autres cas de notification

En fonction des options commerciales souscrites, la plateforme de paiement pourra effectuer un appel vers l'url de notification dans les cas suivants :

- abandon ou annulation de la part de l'acheteur sur la page de paiement
- remboursement effectué depuis le Back Office Marchand
- annulation d'une transaction depuis le Back Office Marchand
- validation d'une transaction depuis le Back Office Marchand
- modification d'une transaction depuis le Back Office Marchand
- etc..

Pour plus d'informations sur le paramétrage des règles, consultez le manuel utilisateur *Centre de notifications*.

4. ENVOYER UN FORMULAIRE DE PAIEMENT EN POST

Le site marchand redirige l'acheteur vers la plateforme de paiement sous la forme d'un formulaire HTML POST en HTTPS.

Ce formulaire contient :

Les éléments techniques suivants :

- Les balises `<form>` et `</form>` qui permettent de créer un formulaire HTML.
- L'attribut `method="POST"` qui spécifie la méthode utilisée pour envoyer les données.
- L'attribut `action="`*https://sogecommerce.societegenerale.eu/vads-payment/*`"` qui spécifie où envoyer les données du formulaire.

Les données du formulaire :

Toutes les données du formulaire doivent être encodées en **UTF-8.**

Les caractères spéciaux (accents, ponctuation etc.) seront ainsi correctement interprétés par la plateforme de paiement. Dans le cas contraire, le calcul de signature sera erroné et le formulaire sera rejeté.

Nous vous invitons à consulter le tableau suivant pour mieux comprendre la codification des formats.

Notation	Description
a	Caractères alphabétiques (de 'A' à 'Z' et de 'a' à 'z')
n	Caractères numériques
s	Caractères spéciaux
an	Caractères alphanumériques
ans	Caractères alphanumériques et spéciaux (à l'exception de "<" et ">")
3	Longueur fixe de 3 caractères
..12	Longueur variable jusqu'à 12 caractères
json	JavaScript Object Notation. Objet contenant des paires de clé/valeur séparées par une virgule. Il commence par une accolade gauche " `{` et se termine par une accolade droite " `}` ". Chaque paire clé/valeur contient le nom de la clé entre double-quotes suivi par " `:` ", suivi par une valeur. Le nom de la clé doit être alphanumérique. La valeur peut être : • une chaine de caractères (dans ce cas elle doit être encadrée par des doubles-quotes) • un nombre • un objet • un tableau • un booléen • vide Exemple: `{"name1":45,"name2":"value2", "name3"=false}`
enum	Caractérise un champ possédant un nombre fini de valeurs. La liste des valeurs possibles est donnée dans la définition du champ.
liste d'enum	Liste de valeurs séparées par un " `;` ". La liste des valeurs possibles est donnée dans la définition du champ. Exemple: `vads_payment_cards=VISA;MASTERCARD`
map	Liste de paires clé/valeur séparées par un " `;` ". Chaque paire clé/valeur contient le nom de la clé suivi par " `=` ", suivi par une valeur. La valeur peut être : • une chaine de caractères • un booléen

Notation	Description
	• un objet json • un objet xml La liste des valeurs possibles pour chaque paire de clé/valeur est donnée dans la définition du champ. Exemple: vads_theme_config=SIMPLIFIED_DISPLAY=true;RESPONSIVE_MODEL=Model_1

- Les champs obligatoires :

Nom du champ	Description	Format	Valeur
signature	Signature garantissant l'intégrité des requêtes échangées entre le site marchand et la plateforme de paiement.	ans	Ex : **ycA5Do5tNvsnKdc/eP1bj2xa19z9q3iWPy9/rpesfS0=**
vads_action_mode	Mode d'acquisition des données du moyen de paiement	enum	**INTERACTIVE**
vads_amount	Montant du paiement dans sa plus petite unité monétaire (le centime pour l'euro)	n..12	Ex : 4525 pour 45,25 EUR
vads_ctx_mode	Mode de communication avec la plateforme de paiement	enum	**TEST** ou **PRODUCTION**
vads_currency	Code numérique de la monnaie à utiliser pour le paiement, selon la norme ISO 4217 (code numérique)	n3	Ex : 978 pour l'euro (EUR)
vads_page_action	Action à réaliser	enum	**PAYMENT**
vads_payment_config	Type de paiement	enum	**SINGLE** pour un paiement en 1 fois **MULTI** pour un paiement en plusieurs fois
vads_site_id	Identifiant de la boutique	n8	Ex : 12345678
vads_trans_date	Date et heure du formulaire de paiement dans le fuseau horaire UTC	n14	Respectez le format AAAAMMJJHHMMSS Ex : 20200101130025
vads_trans_id	Numéro de la transaction. Doit être unique sur une même journée (de 00:00:00 UTC à 23:59:59 UTC). **Attention : ce champ n'est pas sensible à la casse.**	an6	Ex : xrT15p
vads_version	Version du protocole d'échange avec la plateforme de paiement	enum	**V2**

- Les champs fortement recommandés :
  - Le moyen de paiement à utiliser

Nom du champ	Description	Format	Valeur
vads_payment_cards	Permet de forcer le type de carte à utiliser. Il est recommandé de proposer sur le site marchand un bouton de paiement différent pour chaque moyen de paiement. **Il est déconseillé de laisser le champ vide**. Consultez le chapitre *Gérer les moyens de paiement proposés à l'acheteur* du Guide d'implémentation - API Formulaire pour plus d'informations.	enum	Ex : • CB • CVCO • MASTERCARD • VISA • SDD

  - Les données de la commande

Nom du champ	Description	Format	Valeur
vads_order_id	Numéro de commande	ans..64	Ex : 2-XQ001

Nom du champ	Description	Format	Valeur
	Peut être composé de majuscules ou de minuscules, chiffres ou tiret ([A-Z] [a-z], 0-9, _, -).		
vads_order_info	Informations supplémentaires sur la commande	ans..255	Ex : Code interphone 3125
vads_order_info2	Informations supplémentaires sur la commande	ans..255	Ex : Sans ascenseur
vads_order_info3	Informations supplémentaires sur la commande	ans..255	Ex : Express
vads_ext_info_xxxx	Information complémentaire nécessaire au marchand qui apparaîtra dans l'e-mail de confirmation de paiement à destination du marchand et dans le Back Office Marchand (onglet **Extra** du détail de la transaction). **xxxx** correspond au nom de la donnée transmise. Par exemple : vads_ext_info_departure_city	ans..255	Ex : LHR

- Les données de l'acheteur

Nom du champ	Description	Format	Valeur
vads_cust_email	Adresse e-mail de l'acheteur	ans..150	Ex : abc@example.com
vads_cust_id	Référence de l'acheteur sur le site marchand	an..63	Ex : C2383333540
vads_cust_national_id	Identifiant national	ans..255	Ex : 940992310285
vads_cust_title	Civilité de l'acheteur	an..63	Ex : M
vads_cust_status	Statut	enum	**PRIVATE** : pour un particulier **COMPANY** : pour une entreprise
vads_cust_first_name	Prénom	ans..63	Ex : Laurent
vads_cust_last_name	Nom	ans..63	Ex : Durant
vads_cust_legal_name	Raison sociale de l'acheteur	an..100	Ex : D. & Cie
vads_cust_phone	Numéro de téléphone	an..32	Ex : 0467330222
vads_cust_cell_phone	Numéro de téléphone mobile	an..32	Ex : 06 12 34 56 78
vads_cust_address_number	Numéro de voie	ans..64	Ex : 109
vads_cust_address	Adresse postale	ans..255	Ex : Rue de l'innovation
vads_cust_address2	Deuxième ligne d'adresse	ans..255	Ex :
vads_cust_district	Quartier	ans..127	Ex : Centre ville
vads_cust_zip	Code postal	an..64	Ex : 31670
vads_cust_city	Ville	an..128	Ex : Labège
vads_cust_state	Etat / Région	ans..127	Ex : Occitanie
vads_cust_country	Code pays suivant la norme ISO 3166 alpha-2	a2	Ex : "FR" pour la France, "PF" pour la Polynésie Française, "NC" pour la Nouvelle Calédonie, "US" pour les Etats-Unis.

- Les champs recommandés :
  - Les données de livraison

Nom du champ	Description	Format	Valeur
vads_ship_to_city	Ville	an..128	Ex : Bordeaux
vads_ship_to_country	Code pays suivant la norme ISO 3166 (obligatoire pour déclencher une ou plusieurs actions si le profil **Contrôle du pays de la livraison** est activé).	a2	Ex : FR
vads_ship_to_district	Quartier	ans..127	Ex : La Bastide
vads_ship_to_first_name	Prénom	ans..63	Ex : Albert
vads_ship_to_last_name	Nom	ans..63	Ex : Durant
vads_ship_to_legal_name	Raison sociale	an..100	Ex : D. & Cie
vads_ship_to_phone_num	Numéro de téléphone	ans..32	Ex : 0460030288
vads_ship_to_state	Etat / Région	ans..127	Ex : Nouvelle aquitaine
vads_ship_to_status	Définit le type d'adresse de livraison	enum	**PRIVATE** : pour une livraison chez un particulier **COMPANY** : pour une livraison en entreprise
vads_ship_to_street_number	Numéro de voie	ans..64	Ex : 2
vads_ship_to_street	Adresse postale	ans..255	Ex : Rue Sainte Catherine
vads_ship_to_street2	Deuxième ligne d'adresse	ans..255	
vads_ship_to_zip	Code postal	an..64	Ex : 33000

  - Les données du panier

Nom du champ	Description	Format	Valeur
vads_nb_products	Nombre d'articles présents dans le panier	n..12	Ex : 2
vads_product_ext_idN	Code barre du produit dans le site web du marchand. N	an..100	Ex :

Nom du champ	Description	Format	Valeur
	correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).		vads_product_ext_id0 = "0123654789123654789" vads_product_ext_id1 = "0223654789123654789" vads_product_ext_id2 = "0323654789123654789"
vads_product_labelN	Libellé de l'article. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).	ans..255	Ex : vads_product_label0 = "tee-shirt" vads_product_label1 = "Biscuit" vads_product_label2 = "sandwich"
vads_product_amountN	Prix TTC de l'article. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).	n..12	Ex : vads_product_amount0 = "1200" vads_product_amount1 = "800" vads_product_amount2 = "950"
vads_product_typeN	Type de l'article. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).	enum	Ex : vads_product_type0 = "CLOTHING_AND_ACCESSORIES" vads_product_type1 = "FOOD_AND_GROCERY" vads_product_type2 = "FOOD_AND_GROCERY"
vads_product_refN	Référence de l'article. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).	an..64	Ex : vads_product_ref0 = "CAA-25-006" vads_product_ref1 = "FAG-B5-112" vads_product_ref2 = "FAG-S9-650"
vads_product_qtyN	Quantité d'article. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).	n..12	Ex : vads_product_qty0 = "1" vads_product_qty1 = "2" vads_product_qty2 = "2"

Remarque *:*

En renseignant le champ ***vads_nb_products****, l'onglet* ***Panier*** *dans le détail d'une transaction depuis le Back Office Marchand s'affichera.*

Cependant, si les autres champs commençant par ***vads_product_*** *ne sont pas renseignés, l'onglet ne comportera pas d'information. Pour cette raison, en valorisant le champ* ***vads_nb_products****, il devient obligatoire de valoriser les autres champs commençant par* ***vads_product_****.*

- Les champs facultatifs :

  Vous pouvez utiliser des paramètres facultatifs supplémentaires.

  Référez-vous au chapitre **Dictionnaire de données** du guide d'implémentation API Formulaire disponible sur notre site documentaire afin de visualiser la liste des champs disponibles.

Le bouton **Payer** qui va permettre l'envoi des données :

```
<input type="submit" name="payer" value="Payer"/>
```

5. CALCULER LA SIGNATURE

Afin de pouvoir calculer la signature vous devez être en possession :

- de la totalité des champs dont le nom commence par **vads_**
- du type d'algorithme choisi dans la configuration de la boutique
- de la **clé**

La valeur de la clé est disponible dans votre Back Office Marchand depuis le menu **Paramétrage** > **Boutique** > onglet **Clés**.

Le type d'algorithme est défini dans votre Back Office Marchand depuis le menu **Paramétrage** > **Boutique** > onglet **Configuration**.

Pour un maximum de sécurité, il est recommandé d'utiliser l'algorithme HMAC-SHA-256 ainsi qu'une clé alphanumérique.

L'utilisation de l'algorithme SHA-1 est dépréciée mais maintenue pour des raisons de compatibilité.

Pour calculer la signature :

1. Triez les champs dont le nom commence par **vads_** par ordre alphabétique.
2. Assurez-vous que tous les champs soient encodés en UTF-8.
3. Concaténez les valeurs de ces champs en les séparant avec le caractère "**+**".
4. Concaténez le résultat avec la clé de test ou de production en les séparant avec le caractère "**+**".
5. Selon l'algorithme de signature défini dans la configuration de votre boutique:
   a. si votre boutique est configurée pour utiliser "SHA-1", appliquez la fonction de hachage **SHA-1** sur la chaîne obtenue à l'étape précédente. **Déprécié.**
   b. si votre boutique est configurée pour utiliser "HMAC-SHA-256", calculez et encodez au format Base64 la signature du message en utilisant l'algorithme **HMAC-SHA-256** avec les paramètres suivants:
      - la fonction de hachage SHA-256,
      - la clé de test ou de production (en fonction de la valeur du champ **vads_ctx_mode**) comme clé partagée,
      - le résultat de l'étape précédente comme message à authentifier.
6. Sauvegardez le résultat de l'étape précédente dans le champ **signature**.

Exemple de paramètres envoyés à la plateforme de paiement:

```
<form method="POST" action="https://sogecommerce.societegenerale.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="5124" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20170129130025" />
<input type="hidden" name="vads_trans_id" value="123456" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="ycA5Do5tNvsnKdc/eP1bj2xa19z9q3iWPy9/rpesfS0="/>

<input type="submit" name="payer" value="Payer"/>
</form>
```

Cet exemple de formulaire s'analyse de la manière suivante:

1. On trie par ordre **alphabétique** les champs dont le nom commence par **vads_** :
   - vads_action_mode
   - vads_amount
   - vads_ctx_mode
   - vads_currency
   - vads_page_action
   - vads_payment_config
   - vads_site_id
   - vads_trans_date
   - vads_trans_id
   - vads_version
2. On concatène la valeur de ces champs avec le caractère "**+**" :

   ```
   INTERACTIVE+5124+TEST+978+PAYMENT+SINGLE+12345678+20170129130025+123456+V2
   ```

3. On ajoute la valeur de la clé de test à la fin de la chaîne en la séparant par le caractère "**+**". Dans cet exemple, la clé de test est **1122334455667788**

   ```
   INTERACTIVE+5124+TEST+978+PAYMENT+SINGLE+12345678+20170129130025+123456+V2+1122334455667788
   ```

4. Si vous utilisez l'algorithme SHA-1, appliquez-le à la chaîne obtenue.

   Le résultat à transmettre dans le champ signature est : **59c96b34c74b9375c332b0b6a32e6deeec87de2b**

5. Si votre boutique est configurée pour utiliser "HMAC-SHA-256", calculez et encodez au format Base64 la signature du message en utilisant l'algorithme **HMAC-SHA-256** avec les paramètres suivants:
   - la fonction de hachage SHA-256,
   - la clé de test ou de production (en fonction de la valeur du champ **vads_ctx_mode**) comme clé partagée,
   - le résultat de l'étape précédente comme message à authentifier.

   Le résultat à transmettre dans le champ signature est :

   ycA5Do5tNvsnKdc/eP1bj2xa19z9q3iWPy9/rpesfS0=

5.1. Exemple d'implémentation en JAVA

Définition d'une classe utilitaire Sha utilisant l'algorithme HMAC-SHA-256 pour calculer la signature:

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Base64;
import java.util.TreeMap;

public class VadsSignatureExample {
 /**
  * Build signature (HMAC SHA-256 version) from provided parameters and secret key.
  * Parameters are provided as a TreeMap (with sorted keys).
  */
 public static String buildSignature(TreeMap<String, String> formParameters, String
  secretKey) throws NoSuchAlgorithmException, InvalidKeyException, UnsupportedEncodingException
 {
   // Build message from parameters
   String message = String.join("+", formParameters.values());
   message += "+" + secretKey;
   // Sign
   return hmacSha256Base64(message, secretKey);
  }
    /**
    * Actual signing operation.
    */
 public static String hmacSha256Base64(String message, String secretKey) throws
  NoSuchAlgorithmException, InvalidKeyException, UnsupportedEncodingException {
   // Prepare hmac sha256 cipher algorithm with provided secretKey
   Mac hmacSha256;
   try {
    hmacSha256 = Mac.getInstance("HmacSHA256");
   } catch (NoSuchAlgorithmException nsae) {
    hmacSha256 = Mac.getInstance("HMAC-SHA-256");
   }
   SecretKeySpec secretKeySpec = new SecretKeySpec(secretKey.getBytes("UTF-8"), "HmacSHA256");
   hmacSha256.init(secretKeySpec);
   // Build and return signature
   return Base64.getEncoder().encodeToString(hmacSha256.doFinal(message.getBytes("UTF-8")));
  }
}
```

Définition d'une classe utilitaire Sha utilisant l'algorithme SHA-1 pour calculer la signature:

```
import java.security.MessageDigest;
import java.security.SecureRandom;

public class Sha {
    static public final String SEPARATOR = "+" ;
    public static String encode(String src) {
        try {
            MessageDigest md;
            md = MessageDigest.getInstance( "SHA-1" );
            byte bytes[] = src.getBytes( "UTF-8" );
            md.update(bytes, 0, bytes. length );
            byte[] sha1hash = md.digest();
            return convertToHex(sha1hash);
        }
        catch(Exception e){
            throw new RuntimeException(e);
        }
    }
    private static String convertToHex(byte[] sha1hash) {
        StringBuilder builder = new StringBuilder();
            for (int i = 0; i < sha1hash. length ; i++) {
               byte c = sha1hash[i];
               addHex(builder, (c >> 4) & 0xf);
               addHex(builder, c & 0xf);
        }
        return builder.toString();
    }
    private static void addHex(StringBuilder builder, int c) {
        if (c < 10)
            builder.append((char) (c + '0' ));
        else
            builder.append((char) (c + 'a' - 10));
    }
}
```

Fonction qui calcule la signature :

```
public ActionForward performCheck(ActionMapping actionMapping, BasicForm form,
    HttpServletRequest request, HttpServletResponse response){
        SortedSet<String> vadsFields = new TreeSet<String>();
        Enumeration<String> paramNames = request.getParameterNames();

        // Recupere et trie les noms des champs vads_* par ordre alphabetique
        while (paramNames.hasMoreElements()) {
            String paramName = paramNames.nextElement();
            if (paramName.startsWith( "vads_" )) {
               vadsFields.add(paramName);
            }
        }
        // Calcule la signature
        String sep = Sha.SEPARATOR;
        StringBuilder sb = new StringBuilder();
        for (String vadsParamName : vadsFields) {
            String vadsParamValue = request.getParameter(vadsParamName);
            if (vadsParamValue != null) {
               sb.append(vadsParamValue);
            }
            sb.append(sep);
        }
        sb.append( shaKey );
        String c_sign = Sha.encode(sb.toString());
        return c_sign;
}
```

5.2. Exemple d'implémentation en PHP

Exemple de calcul de signature utilisant l'algorithme HMAC-SHA-256:

```
function getSignature ($params,$key)
{
    /**
     *Function that computes the signature.
     * $params : table containing the fields to send in the payment form.
     * $key : TEST or PRODUCTION key
     */
    //Initialization of the variable that will contain the string to encrypt
    $signature_content = "";

    //sorting fields alphabetically
    ksort($params);
    foreach($params as $name=>$value){

        //Recovery of vads_ fields
        if (substr($name,0,5)=='vads_'){

            //Concatenation with "+"
            $signature_content .= $value."+";
         }
    }
    //Adding the key at the end
    $signature_content .= $key;

    //Encoding base64 encoded chain with SHA-256 algorithm
    $signature = base64_encode(hash_hmac('sha256',$signature_content, $key, true));
    return $signature;
 }
```

Exemple de calcul de signature utilisant l'algorithme SHA-1:

```
function getSignature($params, $key)
{
    /**
     * Function that computes the signature.
     * $params : table containing the fields to send in the payment form.
     * $key : TEST or  PRODUCTION key
     */
    //Initialization of the variable that will contain the string to encrypt
    $signature_content = "" ;

    // Sorting fields alphabetically
    ksort($params);
        foreach ($params as $name =>$value)
    {
    // Recovery of vads_ fields
        if (substr($name,0,5)=='vads_') {
            // Concatenation with "+"
            $signature_content .= $value."+";
        }
    }
    // Adding the key at the end
    $signature_content .= $key;

    // Applying SHA-1 algorithm
    $signature = sha1($signature_content);
    return $signature ;
}
```

6. IMPLÉMENTER L'IPN

Le script doit comporter au moins les étapes ci-dessous:

- Récupérer la liste des champs présents dans la réponse envoyée en POST
- Calculer la signature en prenant en compte les données reçues
- Comparer la signature calculée avec celle réceptionnée
- Analyser la nature de la notification
- Récupérer le résultat du paiement

Le script peut par exemple tester l'état de la commande (ou l'information de votre choix) pour vérifier qu'elle n'ait pas déja été mise à jour.

Une fois ces étapes réalisées, le script peut mettre à jour la base de données (nouvel état de la commande, mise à jour du stock, enregistrement des informations du paiement etc.).

Afin de faciliter le support et le diagnostic par le marchand en cas d'erreur lors d'une notification, il est recommandé d'écrire des messages qui permettront de connaître à quel stade du traitement l'erreur s'est produite.

La plateforme lit et stocke les 256 premiers octets du corps de la réponse HTTP.

Vous pouvez écrire des messages tout au long du traitement. Voici un exemple de messages que vous pouvez utiliser:

Message	Cas d'usage
Data received	Message à afficher lors de la récupération des données. Permet de confirmer que la notification a bien été reçue par le site marchand.
POST is empty	Message à afficher lors de la récupération des données. Permet de mettre en évidence une éventuelle redirection qui aurait fait perdre les paramètres postés par la plateforme de paiement.
An error occurred while computing the signature.	Message à afficher lorsque la vérification de la signature de la réponse a échouée.
Order successfully updated.	Message à afficher à la fin du fichier une fois vos traitements terminés avec succès.
An error occurred while updating the order.	Message à afficher à la fin du fichier si une erreur s'est produite pendant vos traitements.

6.1. Préparer son environnement

Les notifications de type Appel URL de notification sont les plus importantes car elles représentent l'unique moyen fiable pour le site marchand d'obtenir le résultat d'un paiement.

Il est donc primordial de s'assurer du bon fonctionnement des notifications.

Voici quelques recommandations à suivre:

- Pour que le dialogue entre la plateforme de paiement et votre site marchand fonctionne, vous devez vous assurer auprès de vos équipes techniques que la plage d'adresse IP **194.50.38.0/24** soit autorisée sur les différents équipements de votre architecture (firewalls, serveur apache, serveur proxy, etc.).

  Les notifications sont envoyées depuis une adresse IP comprise dans la plage 194.50.38.0/24 **en mode Test et en mode Production**.

- Les redirections entraînent la perte des données présentes dans le POST.

  C'est le cas s'il existe une configuration sur vos équipements ou chez votre hébergeur qui redirige les URL de type "http://**www.**example.com" vers "http://example.com" ou "**http**://example.com" vers "**https**://example.com".

- La page ne doit pas comporter d'affichage HTML. L'accès aux ressources telles que les images ou feuilles de styles ralentissent les échanges entre la plateforme de paiement et le site marchand.

- Evitez au maximum d'intégrer des tâches consommatrices de temps comme la génération de facture PDF ou l'envoi d'e-mail dans votre script.

  Le temps de traitement influe directement sur le délai d'affichage de la page de résumé du paiement.

  Plus le traitement de la notification est long, plus l'affichage est retardé. Au delà de 35s, la plateforme considère que l'appel a échoué (timeout).

- Si votre page n'est accessible qu'en https, testez votre URL sur le site de Qualys SSL Labs (*https://www.ssllabs.com/ssltest/* et modifiez votre configuration si nécessaire afin d'obtenir un grade A.

  Votre certificat SSL doit être signé par une autorité de certification connue et reconnue sur le marché.

- Assurez-vous d'utiliser les dernières versions du protocole TLS afin de maintenir un haut niveau de sécurité.

6.2. Récupérer les données retournées dans la réponse

Les données retournées dans la réponse dépendent des paramètres envoyés dans la demande de paiement, du type de paiement réalisé, des options de votre boutique et du format de la notification.

Les données sont toujours envoyées en **POST** par la plateforme de paiement.

La première étape consiste donc à récupérer le contenu reçu en mode POST.

Exemples :

- En PHP, les données seront stockées dans la superglobale **$_POST**.
- En ASP.NET (C#), vous devez utiliser la propriété **Form** de la classe **HttpRequest**.
- En java, vous devez utiliser la méthode **getParameter** de l'interface **HttpServletRequest**.

La réponse est constituée d'une liste de champs. Chaque champ contient une valeur réponse. La liste de champs peut être amenée à évoluer.

Le script devra effectuer une boucle pour récupérer la totalité des champs transmis.

Il est recommandé de tester la présence du champ **vads_hash**, présent uniquement lors d'une notification.

```
if (empty ($_POST)){
 echo 'POST is empty';

}else{
 echo 'Data Received ';
 if (isset($_POST['vads_hash'])){

  echo 'Form API notification detected';
  //Signature computation
  //Signature verification
  //Order Update
 }
}
```

6.3. Calculer la signature de l'IPN

La signature se calcule selon la même logique utilisée lors de la demande de paiement.

Les données transmises par la plateforme de paiement sont encodées en UTF-8. Toute altération des données reçues aboutira à un calcul de signature erroné.

Vous devez calculer la signature avec les champs reçus dans la notification et pas ceux que vous avez transmis dans la demande de paiement.

1. Prenez en considération la totalité des champs dont le nom commence par **vads_**.
2. Triez ces champs par ordre alphabétique.
3. Concaténez les valeurs de ces champs en les séparant avec le caractère "**+**".
4. Concaténez le résultat avec la clé de test ou de production en les séparant avec le caractère "**+**".
5. Selon l'algorithme de signature défini dans la configuration de votre boutique:
   a. si votre boutique est configurée pour utiliser "SHA-1", appliquez la fonction de hachage **SHA-1** sur la chaîne obtenue à l'étape précédente. **Déprécié.**
   b. si votre boutique est configurée pour utiliser "HMAC-SHA-256", calculez et encodez au format Base64 la signature du message en utilisant l'algorithme **HMAC-SHA-256** avec les paramètres suivants:
      - la fonction de hachage SHA-256,
      - la clé de test ou de production (en fonction de la valeur du champ **vads_ctx_mode**) comme clé partagée,
      - le résultat de l'étape précédente comme message à authentifier.

Exemples en PHP

```
function getSignature ($params,$key)
{
    /**
     * Fonction qui calcule la signature.
     * $params : tableau contenant les champs reçus dans l'IPN.
     * $key : clé de TEST ou de PRODUCTION
     */
    //Initialisation de la variable qui contiendra la chaine à chiffrer
    $contenu_signature = "";

    //Tri des champs par ordre alphabétique
    ksort($params);
    foreach($params as $nom=>$valeur){

        //Récupération des champs vads_
        if (substr($nom,0,5)=='vads_'){

            //Concaténation avec le séparateur "+"
            $contenu_signature .= $valeur."+";
         }
    }
    //Ajout de la clé en fin de chaine
    $contenu_signature .= $key;

    //Encodage base64 de la chaine chiffrée avec l'algorithme HMAC-SHA-256
    $sign = base64_encode(hash_hmac('sha256',$contenu_signature, $key, true));
    return $sign;
 }
```

6.4. Comparer les signatures

Pour s'assurer de l'intégrité de la réponse, vous devez comparer la signature contenue dans l'IPN avec la valeur calculée à l'étape précédente.

Il ne faut pas comparer la signature de l'IPN avec la signature que vous avez transmis dans votre demande de paiement.

Si les signatures correspondent,

- alors vous pouvez considérer la réponse comme sûre et procéder à la suite de l'analyse.
- sinon, le script devra lever une exception et avertir le marchand de l'anomalie.

Exemple PHP:

```
if ($_POST['signature'] == $sign){

    //Processing data

}else{
    throw new Exception('An error occurred while computing the signature');
}
```

Les signatures ne correspondent pas en cas :

- d'erreur d'implémentation (erreur dans votre calcul, problème d'encodage UTF-8, etc.),
- d'erreur dans la valeur de la clé utilisée ou dans celle du champ **vads_ctx_mode** (problème fréquent lors du passage en production),
- de tentative de corruption des données.

6.5. Analyser la nature de la notification

Lors d'une notification le champ **vads_url_check_src** permet de différencier les notifications en fonction de leur évènement déclencheur :

- création d'une transaction.
- renvoi de la notification depuis le Back Office Marchand par le marchand.

Il précise la règle de notification appliquée :

Valeur	Règle appliquée
PAY	La valeur PAY est envoyée dans les cas suivants : • paiement immédiat (paiement comptant ou première échéance d'un paiement en plusieurs fois) • paiement différé à moins de 7 jours uniquement si le marchand a configuré la règle **URL de notification à la fin du paiement**. • paiement abandonné ou annulé par l'acheteur uniquement si le marchand a configuré la règle **URL de notification sur annulation**.
BO	Exécution de la notification depuis le Back Office Marchand (clic droit sur une transaction > **Exécuter l'URL de notification**).
BATCH	La valeur BATCH est envoyée dans le cas de la mise à jour du statut d'une transaction après synchronisation auprès de l'acquéreur. C'est le cas des paiements à redirection vers l'acquéreur. Uniquement si le marchand a configuré la règle **URL de notification sur modification par batch**.
BATCH_AUTO	La valeur BATCH_AUTO est envoyée dans les cas suivants: • paiement différé à plus de 7 jours • échéances d'un paiement en plusieurs fois (hormis la première) uniquement si le marchand a configuré la règle **URL de notification sur autorisation par batch**. La notification est envoyée lors de la demande d'autorisation d'un paiement dont le statut est "En attente d'autorisation".
REC	La valeur REC est envoyée uniquement pour les paiements par abonnement si le marchand a configuré la règle **URL de notification à la création d'un paiement récurrent**.
MERCH_BO	La valeur MERCH_BO est envoyée : • lors d'une opération réalisée depuis le Back Office Marchand (annulation, remboursement, modification, validation, duplicata, création et/ou mise à jour d'alias), si le marchand a configuré la règle de notification : **URL de notification sur une opération provenant du Back Office**
RETRY	Rejeu automatique de l'URL de notification.

Tableau 1 : Valeurs associées au champ vads_url_check_src

En testant sa valeur, le script peut réaliser un traitement différent en fonction de la nature de la notification.

Par exemple :

Si **vads_url_check_src** est valorisé à **PAY** ou **BATCH_AUTO** alors le script met à jour le statut de la commande, ...

Si **vads_url_check_src** est valorisé à **REC** alors le script récupère la référence de l'abonnement et incrémente le nombre d'échéances échues en cas de paiement accepté, ...

6.6. Traiter les données de la réponse

Ci-dessous un exemple d'analyse pour vous guider pas à pas lors du traitement des données de la réponse.

1. Identifiez le mode (TEST ou PRODUCTION) dans lequel a été créé la transaction en analysant la valeur du champ **vads_ctx_mode.**

2. Identifiez la commande en récupérant la valeur du champ **vads_order_id** si vous l'avez transmis dans le formulaire de paiement.

   Vérifiez que le statut de la commande n'a pas déja été mis à jour.

3. Récupérez le résultat du paiement transmis dans le champ **vads_trans_status**.

   Sa valeur vous permet de définir le statut de la commande.

Valeur	Description
ABANDONED	**Abandonné** Paiement abandonné par l'acheteur. La transaction n'est pas créée et **n'est donc pas visible dans le Back Office Marchand.**
ACCEPTED	**Accepté**. Statut d'une transaction de type VERIFICATION dont l'autorisation ou la demande de renseignement a été acceptée. Ce statut ne peut évoluer. Les transactions dont le statut est "**ACCEPTED**" ne sont jamais remises en banque.
AUTHORISED	**En attente de remise** La transaction est acceptée et sera remise en banque automatiquement à la date prévue.
AUTHORISED_TO_VALIDATE	**À valider** La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la transaction afin qu'elle soit remise en banque. La transaction peut être validée tant que la date d'expiration de la demande d'autorisation n'est pas dépassée. Si cette date est dépassée alors le paiement prend le statut **EXPIRED**. Le statut **Expiré** est définitif.
CANCELLED	**Annulé** La transaction est annulée par le marchand.
CAPTURED	**Présenté** La transaction est remise en banque.
CAPTURE_FAILED	**La remise de la transaction a échoué.** Contactez le Support.
EXPIRED	**Expiré** Ce statut intervient dans le cycle de vie d'un paiement avec capture différée. La date d'expiration de la demande d'autorisation est atteinte et le marchand n'a pas validé la transaction. Le porteur ne sera donc pas débité.
REFUSED	**Refusé** La transaction est refusée.
SUSPENDED	**Suspendu** La remise de la transaction est temporairement bloquée par l'acquéreur (AMEX GLOBAL ou SECURE TRADING). Une fois la remise traitée correctement, le statut de la transaction deviendra **CAPTURED**.
UNDER_VERIFICATION	**Vérification en cours** En attente de la réponse de l'acquéreur. Ce statut est temporaire.

Valeur	Description
	Une notification sera envoyée au site marchand pour l'avertir du changement de statut. Nécessite l'activation de la règle de notification URL de notification sur modification par batch.
WAITING_AUTHORISATION	**En attente d'autorisation** Le délai de remise en banque est supérieur à la durée de validité de l'autorisation.
WAITING_AUTHORISATION_TO_VALIDATE	**A valider et autoriser** Le délai de remise en banque est supérieur à la durée de validité de l'autorisation. Une autorisation 1 EUR (ou demande de renseignement sur le réseau CB si l'acquéreur le supporte) a été acceptée. Le marchand doit valider manuellement la transaction afin que la demande d'autorisation et la remise aient lieu.

4. Analysez le champ **vads_occurrence_type** pour déterminer s'il s'agit d'un paiement unitaire ou d'un paiement faisant partie d'une série (abonnement ou paiement en N fois).

Valeur	Description
UNITAIRE	Paiement unitaire (paiement comptant).
RECURRENT_INITIAL	Premier paiement d'une série.
RECURRENT_INTERMEDIAIRE	Énième paiement d'une série.
RECURRENT_FINAL	Dernier paiement d'une série.

5. Analysez le champ **vads_payment_config** pour déterminer s'il s'agit d'un **paiement en N fois**.

Nom du champ	Valeur pour un paiement comptant	Valeur pour un paiement en plusieurs fois
vads_payment_config	SINGLE	MULTI (dont la syntaxe exacte est MULTI:first=X;count=Y;period=Z)

S'il s'agit d'un paiement en N fois, identifiez le numéro de l'échéance en récupérant la valeur du champ **vads_sequence_number**.

Attention : avec l'application du Soft Decline, le champ **vads_sequence_number** ne permet plus d'identifier facilement le premier paiement d'un paiement en N fois. Le premier paiement pouvant prendre un numéro de séquence différent de 1, le numéro de séquence du deuxième paiement ne sera pas forcément 2.

6. Récupérez la valeur du champ **vads_trans_date** pour identifier la date du paiement.

7. Récupérez la valeur du champ **vads_capture_delay** pour identifier le nombre de jours avant la remise en banque.

   Ceci vous permettra d'identifier s'il s'agit d'un paiement immédiat ou différé.

8. Récupérez le montant et la devise utilisée. Pour cela, récupérez les valeurs des champs suivants :

Nom du champ	Description
vads_amount	Montant du paiement dans sa plus petite unité monétaire.
vads_currency	Code de la devise utilisée pour le paiement.
vads_change_rate	Taux de change utilisé pour calculer le montant réél du paiement (voir vads_effective_amount).
vads_effective_amount	Montant du paiement dans la devise réellement utilisée pour effectuer la remise en banque.
vads_effective_currency	Devise dans laquelle la remise en banque va être effectuée.

9. Récupérez la valeur du champ **vads_auth_result** pour connaître le résultat de la demande d'autorisation.

La liste complète des codes renvoyés est consultable dans le dictionnaire de données.

Pour vous aider à comprendre le motif du refus, voici une liste des codes fréquemment retournés :

Valeur	Description
03	**Accepteur invalide** Ce code est émis par l'acquéreur. Il correspond à un problème de configuration sur les serveurs d'autorisation. (ex: contrat clos, mauvais code MCC déclaré, etc..). **Pour connaître la raison précise du refus, le marchand doit contacter sa banque.**
05	**Ne pas honorer** Ce code est émis par la banque émettrice de la carte. Il est utilisé dans les cas suivants : • Date d'expiration invalide, • CVV invalide, • crédit dépassé, • solde insuffisant (etc.) **Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.**
51	**Provision insuffisante ou crédit dépassé** Ce code est émis par la banque émettrice de la carte. Il peut être obtenu si l'acheteur ne dispose pas d'un solde suffisant pour réaliser son achat. **Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.**
56	**Carte absente du fichier** Ce code est émis par la banque émettrice de la carte. Le numéro de carte saisi est erroné ou le couple numéro de carte + date d'expiration n'existe pas.
57	**Transaction non permise à ce porteur** Ce code est émis par la banque émettrice de la carte. Il est utilisé dans les cas suivants : • l'acheteur tente d'effectuer un paiement sur internet avec une carte de retrait, • le plafond d'autorisation de la carte est dépassé. **Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.**
59	**Suspicion de fraude** Ce code est émis par la banque émettrice de la carte. Il peut être envoyé suite à une saisie répétée de CVV ou de date d'expiration erronée. **Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.**
60	**L'accepteur de carte doit contacter l'acquéreur** Ce code est émis par l'acquéreur. Il correspond à un problème de configuration sur les serveurs d'autorisation. Il est utilisé lorsque le contrat commerçant ne correspond pas au canal de vente utilisé. (ex : une transaction e-commerce avec un contrat VAD-saisie manuelle). **Contactez le service client pour régulariser la situation.**
81	**Le paiement non sécurisé n'est pas admis par l'émetteur** Ce code est émis par la banque émettrice de la carte. Sur réception de ce code, la plateforme de paiement réalise automatiquement une nouvelle tentative de paiement avec authentification 3D Secure quand cela est possible.

10. Récupérez le résultat de l'authentification du porteur. Pour cela:

a. Récupérez la valeur du champ **vads_threeds_enrolled** pour déterminer le statut de l'enrôlement de la carte.

Valeur	Description
Vide	Processus 3DS non réalisé (3DS désactivé dans la demande, marchand non enrôlé ou moyen de paiement non éligible au 3DS).
Y	Authentification disponible, porteur enrôlé.
N	Porteur non enrôlé.
U	Impossible d'identifier le porteur ou carte non éligible aux tentatives d'authentification (ex. Cartes commerciales ou prépayées).

b. Récupérez le résultat de l'authentification du porteur en récupérant la valeur du champ **vads_threeds_status**.

Valeur	Description
Vide	Authentification 3DS non réalisée (3DS désactivé dans la demande, porteur non enrôlé ou moyen de paiement non éligible au 3DS).
Y	Porteur authentifié avec succès.
N	Erreur d'authentification du porteur.
U	Authentification impossible.
A	Tentative d'authentification mais authentification non réalisée.

11. Récupérez le résultat des contrôles associés à la fraude en identifiant la valeur du champ **vads_risk_control**. Ce champ est envoyé uniquement si le marchand a:

- souscrit au service « **Aide à la décision** »
- activé au moins un contrôle depuis son Back Office Marchand (menu **Paramétrage** > **Contrôle des risques**).

Il prend comme valeur une liste de valeurs séparées par un « ; » dont la syntaxe est:
vads_risk_control = control1=result1;control2=result2

Les valeurs possibles pour **control** sont :

Valeur	Description
CARD_FRAUD	Contrôle la présence du numéro de carte de l'acheteur dans la liste grise de cartes.
SUSPECT_COUNTRY	Contrôle la présence du pays émetteur de la carte de l'acheteur dans la liste des pays interdits.
IP_FRAUD	Contrôle la présence de l'adresse IP de l'acheteur dans la liste grise d'IP.
CREDIT_LIMIT	Contrôle la fréquence et les montants d'achat d'un même numéro de carte, ou le montant maximum d'une commande.
BIN_FRAUD	Contrôle la présence du code BIN de la carte dans la liste grise des codes BIN.
ECB	Contrôle si la carte de l'acheteur est de type e-carte bleue.
COMMERCIAL_CARD	Contrôle si la carte de l'acheteur est une carte commerciale.
SYSTEMATIC_AUTO	Contrôle si la carte de l'acheteur est une carte à autorisation systématique.
INCONSISTENT_COUNTRIES	Contrôle si le pays de l'adresse IP, le pays émetteur de la carte de paiement, et le pays de l'adresse de l'acheteur sont cohérents entre eux.
NON_WARRANTY_PAYMENT	Transfert de responsabilité.
SUSPECT_IP_COUNTRY	Contrôle la présence du pays de l'acheteur, identifié par son adresse IP, dans la liste des pays interdits.

Les valeurs possibles pour **result** sont :

Valeur	Description
OK	OK.
WARNING	Contrôle informatif échoué.
ERROR	Contrôle bloquant échoué.

12. Récupérez le type de carte utilisé pour le paiement.

Deux cas de figures peuvent se présenter:

- Pour un paiement réalisé avec **une seule carte**. Les champs à traiter sont les suivants :

Nom du champ	Description
vads_card_brand	Marque de la carte utilisée pour le paiement. ex : CB, VISA, VISA_ELECTRON, MASTERCARD, MAESTRO, VPAY
vads_brand_management	Permet de connaître la marque utilisée lors du paiement, la liste des marques disponibles et de savoir si l'acheteur a modifié la marque choisie par le marchand.

Nom du champ	Description
vads_card_number	Numéro de la carte utilisée pour réaliser le paiement.
vads_expiry_month	Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).
vads_expiry_year	Année d'expiration sur 4 chiffres (ex : 2023).
vads_bank_code	Code de la banque émettrice
vads_bank_label	Nom de la banque émettrice
vads_bank_product	Code produit de la carte
vads_card_country	Code Pays du pays d'émission de la carte (Code alpha ISO 3166-2 ex : "FR" pour la France, "PF" pour la Polynésie Française, "NC" pour la Nouvelle Calédonie, "US" pour les Etats-Unis.).

- Pour un **paiement fractionné** (c'est-à-dire une transaction utilisant plusieurs moyens de paiement), les champs à traiter sont les suivants :

Nom du champ	Valeur	Description
vads_card_brand	MULTI	Plusieurs types de cartes sont utilisés pour le paiement.
vads_payment_seq	Au format json, voir détails ci-dessous.	Détails des transactions réalisées.

Le champ **vads_payment_seq** (format json) décrit la séquence de paiement fractionné. Il contient les éléments :

1. "trans_id" : identifiant de la transaction global à la séquence de paiement.
2. "transaction" : tableau des transactions de la séquence. Les éléments qui le composent sont les suivants :

Nom du paramètre	Description
amount	Montant de la séquence de paiement.
operation_type	Opération de débit.
auth_number	Numéro d'autorisation. Ne sera pas retourné si non applicable au moyen de paiement concerné. Exemple : 949478
auth_result	Code retour de la demande d'autorisation.
capture_delay	Délai avant remise (en jours). • Pour un paiement par carte bancaire, la valeur de ce paramètre tient compte du délai en nombre de jours avant la remise en banque. Si ce paramètre n'est pas transmis dans le formulaire de paiement, la valeur par défaut définie dans le Back Office Marchand sera utilisée.
card_brand	Moyen de paiement utilisé. Pour un paiement par carte bancaire (exemple CB ou cartes CB cobadgées Visa ou Mastercard), ce paramètre est valorisé à **"CB"**. Se référer au guide d'intégration du formulaire de paiement disponible sur notre site documentaire pour visualiser la liste complète des types de carte.
card_number	Numéro du moyen de paiement.
expiry_month	Mois d'expiration du moyen de paiement.
expiry_year	Année d'expiration du moyen de paiement.
payment_certificate	Certificat de paiement.
contract_used	Contrat utilisé pour le paiement.
identifier	Identifiant unique (token/alias) associé à un moyen de paiement.
identifier_status	Présent uniquement si l'action demandée correspond à la création ou à la mise à jour d'un alias. Valeurs possibles :

Valeur	Description
CREATED	La demande d'autorisation a été acceptée. L'alias (ou RUM pour un paiement SEPA) est créé avec succès.
NOT_CREATED	La demande d'autorisation a été refusée. L'alias (ou RUM pour un paiement SEPA) n'est pas créé et n'apparaîtra pas dans le Back Office Marchand.

<table>
<tr><th>Nom du paramètre</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Valeur</th><th>Description</th></tr>
<tr><td>UPDATED</td><td>L'alias (ou RUM pour un paiement SEPA) est mis à jour avec succès.</td></tr>
<tr><td>NOT_UPDATED</td><td>L'alias (ou RUM pour un paiement SEPA) n'a pas été mis à jour.</td></tr>
<tr><td>ABANDONED</td><td>Action abandonnée par l'acheteur (débiteur).
L'alias (ou RUM pour un paiement SEPA) n'est pas créé et n'apparaîtra pas dans le Back Office Marchand.</td></tr>
</table>
</td></tr>
<tr><td>presentation_date</td><td>Pour un paiement par carte bancaire, ce paramètre correspond à la date de remise en banque souhaitée (au format ISO 8601).</td></tr>
<tr><td>trans_id</td><td>Numéro de transaction.</td></tr>
<tr><td>ext_trans_id</td><td>Paramètre absent pour le paiement par carte bancaire.</td></tr>
<tr><td>trans_uuid</td><td>Référence unique générée par la plateforme de paiement suite à la création d'une transaction de paiement.
Offre une garantie d'unicité pour chaque transaction.</td></tr>
<tr><td>extra_result</td><td>Code numérique du résultat des contrôles de risques.
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>Vide</td><td>Pas de contrôle effectué.</td></tr>
<tr><td>00</td><td>Tous les contrôles se sont déroulés avec succès.</td></tr>
<tr><td>02</td><td>La carte a dépassé l’encours autorisé.</td></tr>
<tr><td>03</td><td>La carte appartient à la liste grise du marchand.</td></tr>
<tr><td>04</td><td>Le pays d’émission de la carte appartient à la liste grise du marchand.</td></tr>
<tr><td>05</td><td>L’adresse IP appartient à la liste grise du marchand.</td></tr>
<tr><td>06</td><td>Le code bin appartient à la liste grise du marchand.</td></tr>
<tr><td>07</td><td>Détection d’une e-carte bleue.</td></tr>
<tr><td>08</td><td>Détection d’une carte commerciale nationale.</td></tr>
<tr><td>09</td><td>Détection d’une carte commerciale étrangère.</td></tr>
<tr><td>14</td><td>Détection d’une carte à autorisation systématique.</td></tr>
<tr><td>20</td><td>Contrôle de cohérence : aucun pays ne correspond (pays IP, pays carte, pays de l’acheteur).</td></tr>
<tr><td>30</td><td>Le pays de l’adresse IP appartient à la liste grise.</td></tr>
<tr><td>99</td><td>Problème technique rencontré par le serveur lors du traitement d’un des contrôles locaux.</td></tr>
</table>
</td></tr>
<tr><td>sequence_number</td><td>Numéro de séquence.</td></tr>
<tr><td>trans_status</td><td>Statut de la transaction.</td></tr>
</table>

Les transactions annulées sont également présentes dans le tableau.

13. Enregistrez la valeur du champ **vads_trans_uuid**. Elle vous permettra d'identifier de manière unique la transaction si vous utilisez les API Web Services.

14. Récupérez toutes les informations concernant le détail de la commande, le détail de l'acheteur et le détail de la livraison.

    Ces données sont présentes dans la réponse que si elles ont été envoyées dans le formulaire de paiement.

    Leur valeur est identique à celle soumise dans le formulaire.

15. Procédez à la mise à jour de la commande.

6.7. Test et troubleshooting

Pour tester les notifications, suivez les étapes suivantes :

1. Réalisez un paiement (en mode TEST ou en mode PRODUCTION).
2. Une fois le paiement terminé, recherchez la transaction dans votre Back Office (Menu **Gestion** > **Transactions** ou **Transactions de TEST** si vous avez réalisé le paiement en mode TEST.
3. Double-cliquez sur la transaction pour afficher le **détail de la transaction**.
4. Dans le détail de la transaction, recherchez la section **Données techniques**.
5. Vérifiez le statut de l'URL de notification:

La liste des statuts possibles est donnée ci-dessous:

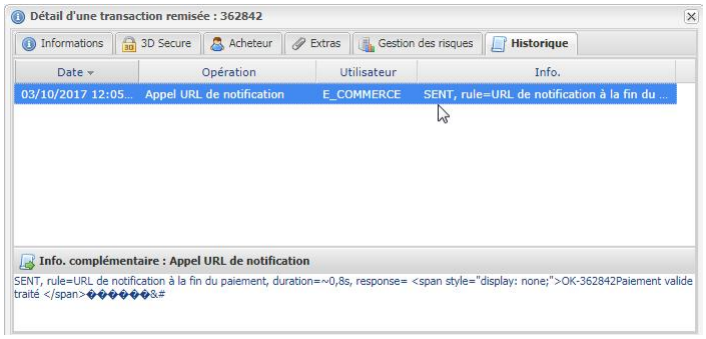
Statut	Description
N/A	La transaction n'a pas donné lieu a une notification ou aucune règle de notification n'est activée.
URL non définie	Un événement a déclenché la règle de notification de fin de paiement mais l'URL n'est pas configurée.
Appel en cours	La notification est en cours. Ce statut est temporaire.
Envoyé	La notification a bien été envoyée et un équipement distant a répondu avec un code HTTP 200, 201, 202, 203, 204, 205 ou 206.
Envoyé (redirection permanente)	Le site marchand a retourné un code HTTP 301 ou 308 avec une nouvelle URL à contacter. Un nouvel appel en mode POST est réalisé vers la nouvelle URL.
Envoyé (redirection temporaire)	Le site marchand a retourné un code HTTP 302 ou 307 avec une nouvelle URL à contacter. Un nouvel appel en mode POST est réalisé vers la nouvelle URL.
Envoyé (redirection vers une autre page)	Le site marchand a retourné un code HTTP 303 avec une nouvelle URL à contacter. Un nouvel appel en mode GET est réalisé vers la nouvelle URL.
Échoué	Erreur générique différente des codes décrits ci-après.
Serveur injoignable	La notification a duré plus de 35s.
Erreur handshake SSL	La configuration de votre serveur n'est pas correcte. Réalisez un diagnostic sur le site de Qualys (*https://www.ssllabs.com/ssltest/*) et corrigez les erreurs.
Connexion interrompue	Erreur de communication.
Connexion refusée	Erreur de communication.
Erreur serveur 300	Cas de redirection non supporté par la plateforme.
Erreur serveur 304	Cas de redirection non supporté par la plateforme.
Erreur serveur 305	Cas de redirection non supporté par la plateforme.
Erreur serveur 400	Le site marchand a retourné un code HTTP 400 Bad Request.
Erreur serveur 401	Le site marchand a retourné 'un code HTTP 401 Unauthorized. Assurez-vous que la ressource n'est pas protégée par un fichier .htaccess.
Erreur serveur 402	Le site marchand a retourné un code HTTP 402 Payment Required.
Erreur serveur 403	Le site marchand a retourné un code HTTP 403 Forbidden. Assurez-vous que la ressource n'est pas protégée par un fichier .htaccess.
Erreur serveur 404	Le site marchand a retourné un code HTTP 404 Not Found. Vérifiez que la saisie de l'URL est correcte dans le paramétrage de la règle. Vérifiez aussi que le fichier est bien présent sur votre serveur.
Erreur serveur 405	Le site marchand a retourné un code HTTP 405 Method Not allowed.
Erreur serveur 406	Le site marchand a retourné un code HTTP 406 Not Acceptable.
Erreur serveur 407	Le site marchand a retourné un code HTTP 407 Proxy Authentication Required.
Erreur serveur 408	Le site marchand a retourné un code HTTP 408 Request Time-out.
Erreur serveur 409	Le site marchand a retourné un code HTTP 409 Conflict.

Statut	Description
Erreur serveur 410	Le site marchand a retourné un code HTTP 410 Gone.
Erreur serveur 411	Le site marchand a retourné un code HTTP 411 Length Required.
Erreur serveur 412	Le site marchand a retourné un code HTTP 412 Precondition Failed.
Erreur serveur 413	Le site marchand a retourné un code HTTP 413 Request Entity Too Large.
Erreur serveur 414	Le site marchand a retourné un code HTTP 414 Request-URI Too long.
Erreur serveur 415	Le site marchand a retourné un code HTTP 415 Unsupported Media Type.
Erreur serveur 416	Le site marchand a retourné un code HTTP 416 Requested range unsatisfiable.
Erreur serveur 417	Le site marchand a retourné un code HTTP 417 Expectation failed.
Erreur serveur 419	Le site marchand a retourné un code HTTP 419 Authentication Timeout.
Erreur serveur 421	Le site marchand a retourné un code HTTP 421 Misdirected Request.
Erreur serveur 422	Le site marchand a retourné un code HTTP 422 Unprocessable Entity.
Erreur serveur 423	Le site marchand a retourné un code HTTP 423 Locked.
Erreur serveur 424	Le site marchand a retourné un code HTTP 424 Failed Dependency.
Erreur serveur 425	Le site marchand a retourné un code HTTP 425 Too Early.
Erreur serveur 426	Le site marchand a retourné un code HTTP 426 Upgrade Required.
Erreur serveur 429	Le site marchand a retourné un code HTTP 431 Request Header Fields Too Large.
Erreur serveur 431	Le site marchand a retourné un code HTTP 415 Unsupported Media Type.
Erreur serveur 451	Le site marchand a retourné un code HTTP 451 Unavailable For Legal Reasons.
Erreur serveur 500	Le site marchand a retourné un code HTTP 500 Internal Server Error. Une erreur applicative est survenue au niveau du serveur hébergeant votre boutique. Consultez les logs de votre serveur HTTP (le plus souvent apache). Le problème ne peut être corrigé qu'en intervenant sur votre serveur.
Erreur serveur 501	Le site marchand a retourné un code HTTP 501 Not Implemented.
Erreur serveur 502	Le site marchand a retourné un code HTTP 502 Bad Gateway / Proxy Error.
Erreur serveur 503	Le site marchand a retourné un code HTTP 503 Service Unavailable.
Erreur serveur 504	Le site marchand a retourné un code HTTP 504 Gateway Time-out. Le serveur marchand n'a pas accepté l'appel dans le délai imparti de 10s.
Erreur serveur 505	Le site marchand a retourné un code HTTP 505 HTTP Version not supported.

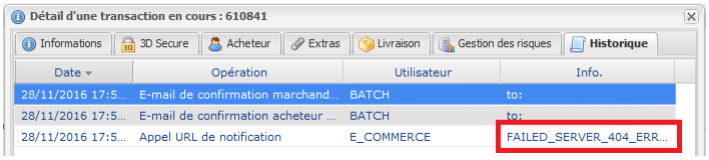
Pour obtenir plus d'informations sur une notification, cliquez sur le lien **Afficher les informations** ou cliquez sur l'onglet **Historique** et recherchez la ligne **Appel URL de notification**.

Afin d'aider le marchand à identifier l'origine de l'erreur, la plateforme analyse systématiquement les 512 premiers caractères retournés par le site marchand et les affiche dans la colonne **Info**.

- Exemple de notification traitée avec succès:



- Exemple de notification en erreur



Si la plateforme n'arrive pas à joindre l'URL de votre page, alors un e-mail d'alerte est envoyé à l'adresse e-mail spécifiée.

Il contient :

- Le code HTTP de l'erreur rencontrée
- Des éléments d'analyse en fonction de l'erreur
- Ses conséquences
- La procédure à suivre depuis le Back Office Marchand pour renvoyer la requête vers l'URL définie dans le paramétrage de la règle.

7. TRAITER LE RETOUR À LA BOUTIQUE

Par défaut, lorsque l'acheteur revient sur le site marchand, aucun paramètre n'est transmis par son navigateur.

Néanmoins si le champ **vads_return_mode** a été transmis dans le formulaire de paiement (voir chapitre **Gérer le retour vers le site marchand** du guide d'implémentation API Formulaire disponible sur notre site documentaire) il sera possible de récupérer les données :

- soit en GET : données présentes dans l'url sous la forme : ?param1=valeur1&param2=valeur2.
- soit en POST : données envoyées dans un formulaire POST.

Les données transmises au navigateur sont les mêmes que lors des notifications (IPN).

Seuls les champs **vads_url_check_src** et **vads_hash** ne seront envoyés que dans la notification instantanée.

Vous pouvez vous référer au chapitre **Analyser le résultat du paiement** pour analyser ces données.

Remarque : le retour à la boutique doit vous permettre uniquement d'afficher un contexte visuel à l'acheteur. N'utilisez pas les données reçues pour effectuer le traitement en base de données.

8. PROCÉDER À LA PHASE DE TEST

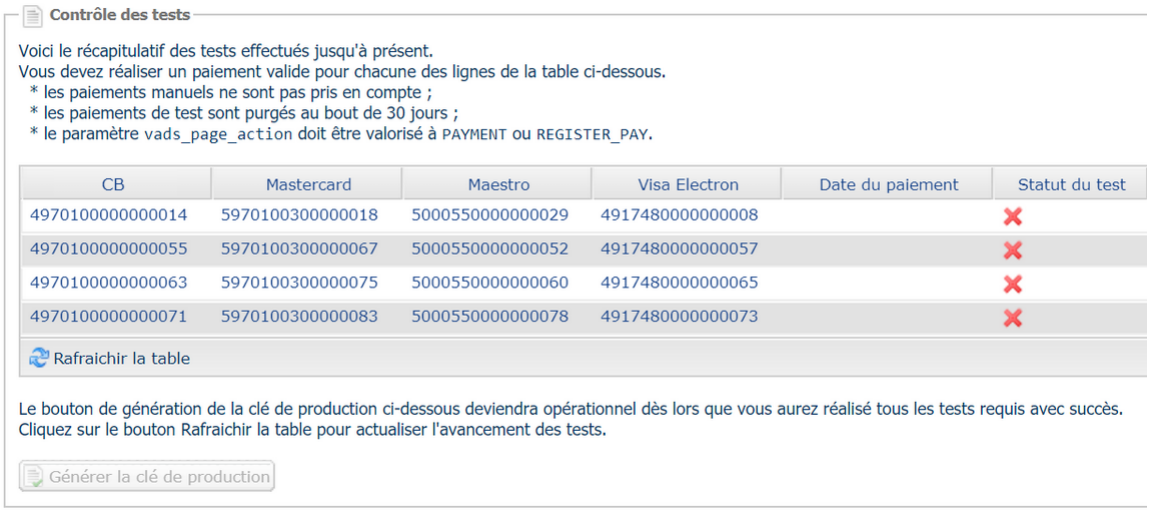
Préalablement au passage en production de la boutique, il est nécessaire de réaliser des tests pour s'assurer du bon fonctionnement entre le site marchand et la plateforme de paiement.

Les demandes de paiement de test doivent:

- contenir le champ **vads_ctx_mode** valorisé à **TEST**.
- utiliser **la clé de test** pour le calcul de la signature.

Plusieurs cas de paiements peuvent être simulés en utilisant les numéros de carte de test précisés sur la page de paiement. Le marchand pourra notamment tester les différents résultats d'authentification 3D Secure (si ce dernier est enrôlé 3DS et si l'option 3DS n'est pas désactivée).

La liste des tests à réaliser pour générer la clé de production est donnée dans le Back Office Marchand, menu **Paramétrage** > **Boutique** > **Clés**.

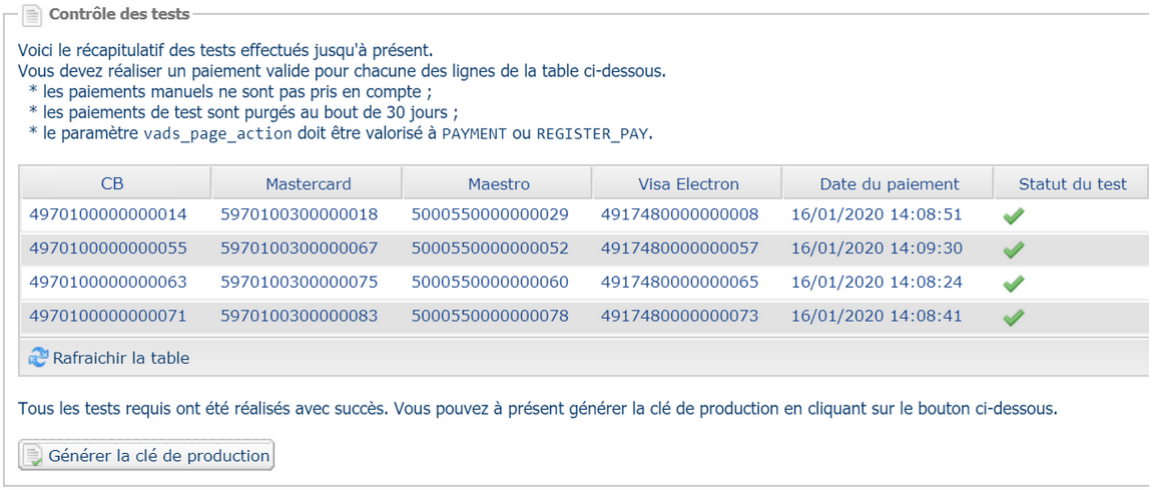


Chaque ligne de la liste regroupe les numéros de cartes associées au même scénario (soit 2 paiements acceptés et 2 paiement refusés).

Chaque colonne correspond à un type de carte différent : CB/VISA, MASTERCARD, MAESTRO, VISA ELECTRON).

Pour réaliser la phase de test :

1. Passez une commande sur votre site marchand comme si vous étiez un de vos acheteurs.
2. Une fois redirigé vers la page de paiement, sélectionnez le type de carte de votre choix
3. Reportez-vous à la liste des tests pour identifier le numéro de carte à utiliser.
4. Lorsque qu'un test est validé, son statut est mis à jour dans la liste. Utilisez le bouton **Rafraichir la table** si le statut ne s'est pas rafraichi automatiquement.
5. Une fois les 4 tests validés, le bouton **Générer la clé de production** devient accessible.



6. Cliquez sur le bouton **Générer la clé de production** et acceptez les différents messages d'avertissement.

La clé de production est maintenant disponible.

9. ACTIVER LA BOUTIQUE EN MODE PRODUCTION

9.1. Générer la clé de production

Vous pouvez générer la clé de production depuis le menu **Paramétrage** > **Boutique** > Onglet **Clés** > bouton **Générer la clé de production**.

Une fois la clé de production générée, sa valeur apparaît sous l'onglet **Clés**.

Un e-mail est envoyé à l'interlocuteur en charge du dossier (responsable administratif de la société) pour lui confirmer la génération de la clé de production.

9.2. Basculer le site marchand en production

1. Valorisez le champ **vads_ctx_mode** à **PRODUCTION**.
2. Modifiez la valeur de la clé de test avec la valeur de votre clé de production pour calculer la signature.
   Vous trouverez cette valeur depuis le menu **Paramétrage** > **Boutique** > Onglet **Clés**.
3. Renseignez correctement l'URL de notification à la fin du paiement en mode PRODUCTION depuis le menu **Paramétrage** > **Règles de notification**.

9.3. Réaliser un premier paiement de production

Nous vous conseillons de vérifier les deux points suivants :

- Le bon fonctionnement en environnement de production de bout-en-bout.

  Pour ce faire, effectuez une transaction réelle d'au moins 2€.

  Cette transaction pourra être annulée par la suite depuis le Back Office Marchand via le menu **Gestion** > **Transactions** > onglet **Transactions en cours**. Cette transaction ne sera donc pas remise en banque.

  Cependant il est recommandé de laisser la transaction être remise en banque pour valider que le crédit soit fait sur le compte du marchand. Il sera ensuite possible de procéder à un remboursement.

- Le bon fonctionnement de l'URL de notification de paiement (URL de notification à la fin du paiement) renseignée dans le Back Office Marchand.

  Pour ce faire, ne cliquez pas sur le bouton **Retour à la boutique** après un paiement.

  Affichez le détail de la transaction dans le Back Office Marchand et vérifiez que le statut de l'URL de notification (Statut URL de notification) est bien **Envoyé.**

10. OBTENIR DE L'AIDE

Vous cherchez de l'aide? Consultez notre FAQ sur notre site

https://sogecommerce.societegenerale.eu/doc/fr-FR/faq/sitemap.html

Pour toute question technique ou demande d'assistance, contactez *le support technique*.

Pour faciliter le traitement de vos demandes, il vous sera demandé de communiquer votre identifiant de boutique (numéro à 8 chiffres).

Cette information est disponible dans l'e-mail d'inscription de votre boutique ou dans le Back Office Marchand (menu **Paramétrage** > **Boutique** > **Configuration**).

