



# **Réussir l'intégration de l'API Formulaire**

## **Guide d'implémentation**

Version du document 1.5

# Sommaire

<b>1. HISTORIQUE DU DOCUMENT.....</b>	<b>3</b>
<b>2. CONTACTER L'ASSISTANCE TECHNIQUE.....</b>	<b>4</b>
<b>3. ÉTABLIR LE DIALOGUE AVEC LA PLATEFORME DE PAIEMENT.....</b>	<b>5</b>
3.1. Définir l'URL de la page de paiement.....	5
3.2. S'identifier lors des échanges.....	5
3.3. Gérer le dialogue vers le site marchand.....	8
3.4. Gérer la sécurité.....	10
3.5. Configurer la notification à la fin du paiement.....	12
<b>4. ENVOYER UN FORMULAIRE DE PAIEMENT EN POST.....</b>	<b>14</b>
<b>5. CALCULER LA SIGNATURE.....</b>	<b>18</b>
5.1. Exemple d'implémentation en JAVA.....	20
5.2. Exemple d'implémentation en PHP.....	22
<b>6. ANALYSER LE RÉSULTAT DU PAIEMENT.....</b>	<b>23</b>
6.1. Récupérer les données retournées dans la réponse.....	23
6.2. Calculer la signature.....	24
6.3. Comparer les signatures.....	24
6.4. Traiter les données de la réponse.....	25
<b>7. TRAITER LE RETOUR À LA BOUTIQUE.....</b>	<b>32</b>
<b>8. PROCÉDER À LA PHASE DE TEST.....</b>	<b>33</b>
8.1. Réaliser des tests de paiement.....	33
8.2. Tester l'URL de notification instantanée (IPN).....	33
<b>9. ACTIVER LA BOUTIQUE EN MODE PRODUCTION.....</b>	<b>34</b>
9.1. Générer la clé de production.....	34
9.2. Basculer le site marchand en production.....	34
9.3. Réaliser un premier paiement de production.....	34

# 1. HISTORIQUE DU DOCUMENT

---

Version	Auteur	Date	Commentaire
1.5	Natixis Solutions	Payment 23/01/2019	<ul style="list-style-type: none"> <li>Mise à jour du chapitre <b>S'identifier lors des échanges</b></li> <li>Remplacement du terme "Certificat" par "Clé" dans tous les menus</li> </ul>
1.4	Natixis Solutions	Payment 04/09/2018	<ul style="list-style-type: none"> <li>Mise à jour de chapitre <b>Calculer la signature.</b></li> </ul>
1.3	Natixis Solutions	Payment 26/06/2018	<ul style="list-style-type: none"> <li>Mise à jour de chapitre <b>Calculer la signature.</b></li> <li>Nouvelle valeur du champ <code>vads_trans_status</code> : <b>SUSPENDED</b></li> <li>Mise à jour du chapitre <b>Envoyer un formulaire de paiement en POST</b></li> </ul>
1.2	Natixis Solutions	Payment 23/05/2018	Possibilité de choisir l'algorithme de calcul de signature (SHA-1 ou SHA-256)
1.1	Natixis Solutions	Payment 14/11/2017	<ul style="list-style-type: none"> <li>Mise à jour des exemples.</li> <li>Ajout d'exemples de calcul de signature en Java et PHP.</li> </ul>
1.0	Natixis Solutions	Payment 25/03/2015	Version initiale

Ce document et son contenu sont strictement confidentiels. Il n'est pas contractuel. Toute reproduction et/ou distribution de ce document ou de toute ou partie de son contenu à une entité tierce sont strictement interdites ou sujettes à une autorisation écrite préalable de Natixis Payment Solutions. Tous droits réservés.

## 2. CONTACTER L'ASSISTANCE TECHNIQUE

---

Pour toute question technique ou demande d'assistance, nos services sont disponibles du lundi au vendredi, de 9h à 18h

par téléphone au :

**0 811 363 364** Service 0,06 € / min  
+ prix appel

pour les clients Banque Populaire

par téléphone au :

**0 810 004 724** Service 0,06 € / min  
+ prix appel

pour les clients Caisse d'Epargne

par e-mail :

[supportvad@lyra-network.com](mailto:supportvad@lyra-network.com)

via votre Back Office Marchand : menu **Aide** > **Contacteur le support**

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**).

## 3. 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.

### 3.1. 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://paiement.systempay.fr/vads-payment/>

### 3.2. 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://paiement.systempay.fr/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.

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

6. Sélectionnez l'onglet **Clés**.

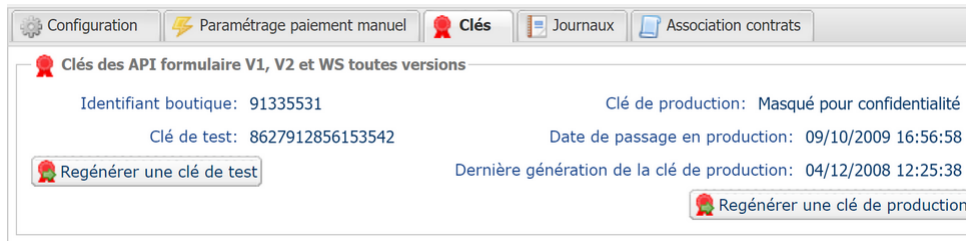


Image 1 : Onglet Clés

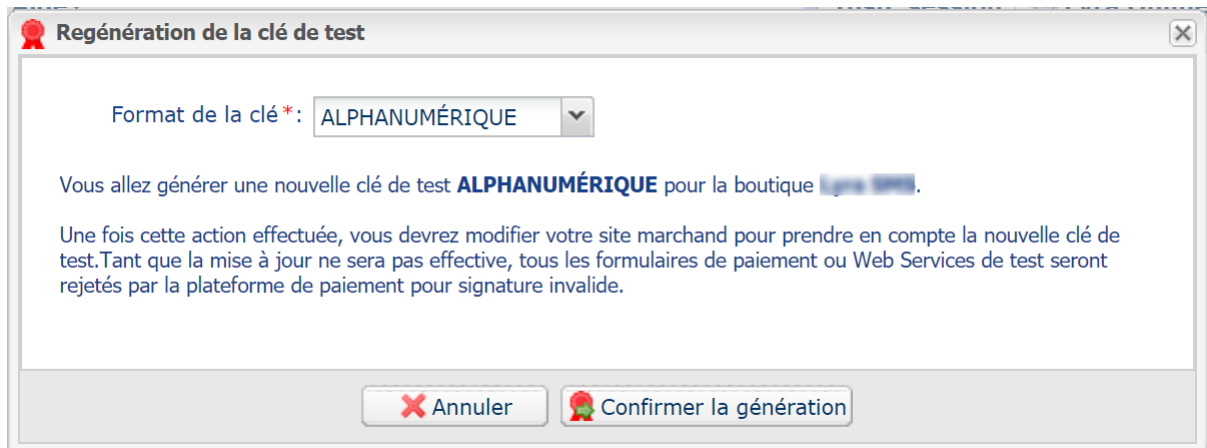
Deux types de clé sont mis à disposition :

- 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\*:  

**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 **[Nom 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

### 3.3. 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)

La plateforme de paiement notifie automatiquement au site marchand le résultat du paiement. Les données sont envoyées en mode **POST**.

La plateforme est capable de contacter le site marchand quel que soit le protocole utilisé (http ou https).

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

**Pour traiter** ces notifications, le marchand doit **créer une page** sur son site qui :

- analyse les données reçues en mode **POST**,
- s'assure de l'intégrité des informations reçues en calculant la signature,
- vérifie qu'il ne s'agit pas d'un doublon de notification (renvoi de la notification depuis le Back Office Marchand par exemple),
- déclenche la mise à jour de sa base de données (état de la commande, stock, etc.),
- envoie des e-mails à l'acheteur (facture, suivi de commande, etc.).

Le temps de traitement influe directement sur le délai d'affichage de la page de résumé du paiement. Plus le traitement est long, plus l'affichage est retardé.

**Pour recevoir** les notifications, le marchand doit **paramétrer** les règles de notifications depuis son Back Office Marchand (voir chapitre **Paramétrer les notifications**).

En cas de problème de communication vers le site marchand, la plateforme de paiement envoie un e-mail à l'administrateur de la boutique, précisant la raison de l'échec (erreur http, etc.) ainsi que la procédure à suivre pour renvoyer la notification depuis le Back Office Marchand.

#### 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**).

Remarque :

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, veuillez à la renseigner (voir chapitre **Paramétrer les notifications**).

### 3.4. Gérer la sécurité

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

#### 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 :

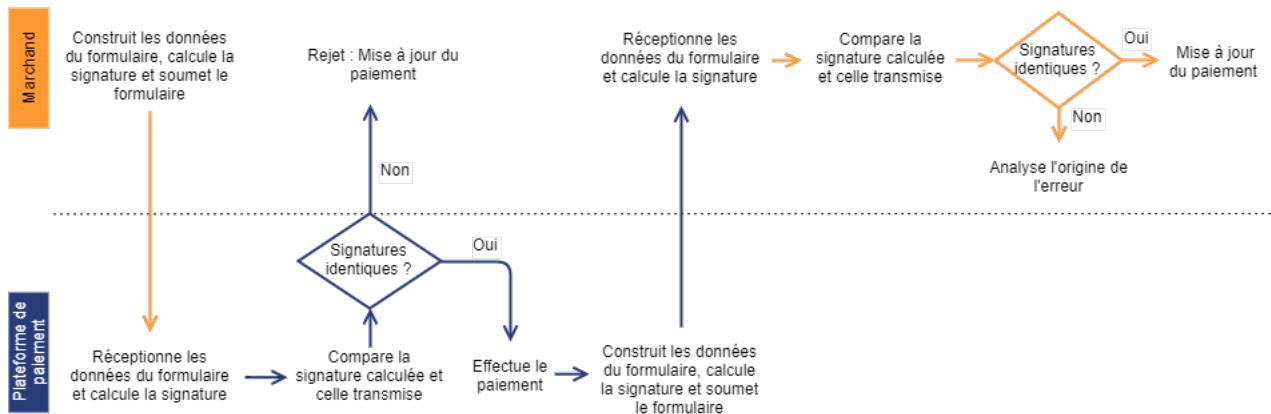


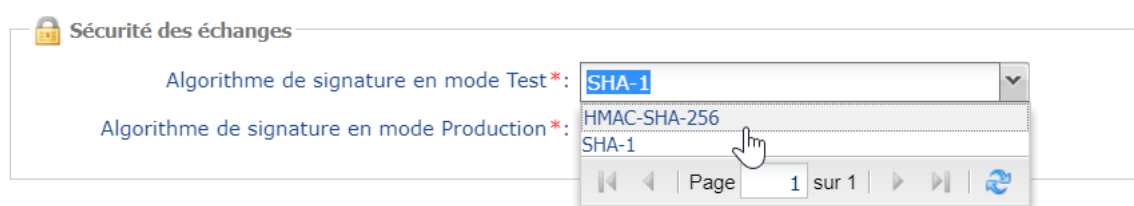
Image 3 : Diagramme mécanisme de sécurité

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.).

### **Sélectionner l'algorithme de hachage**

Depuis le Back Office Marchand (menu **Paramétrage > Boutique > Configuration**), 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.

### **Important**

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.**

### **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**.

### **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).

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

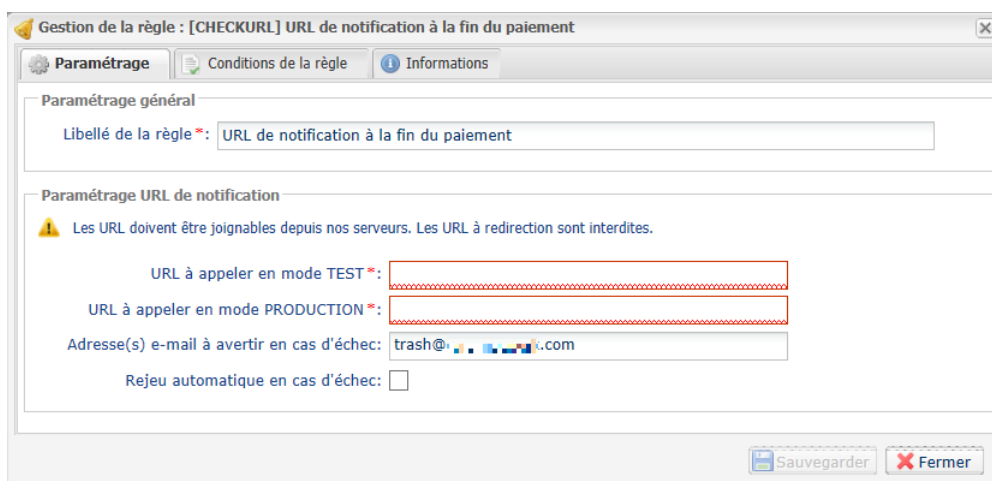
### 3.5. Configurer la notification à la fin du paiement

---

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

Pour paramétrer cette notification :

1. Effectuez un clic droit sur la ligne **URL de notification à la fin du paiement**.
2. Sélectionnez **Activer la règle**.
3. Effectuez à nouveau un clic droit sur **URL de notification à la fin du paiement**.
4. Sélectionnez **Gérer la règle**.
5. Renseignez l'URL de votre page dans les champs **URL à appeler en mode TEST** et **URL à appeler en mode PRODUCTION**.



The screenshot shows a web interface window titled "Gestion de la règle : [CHECKURL] URL de notification à la fin du paiement". It has three tabs: "Paramétrage", "Conditions de la règle", and "Informations". The "Paramétrage" tab is active and contains the following fields:

- Paramétrage général**: "Libellé de la règle \*": URL de notification à la fin du paiement
- Paramétrage URL de notification**:
  - Warning: "Les URL doivent être joignables depuis nos serveurs. Les URL à redirection sont interdites."
  - "URL à appeler en mode TEST \*": [Empty text box]
  - "URL à appeler en mode PRODUCTION \*": [Empty text box]
  - "Adresse(s) e-mail à avertir en cas d'échec": trash@... .com
  - "Rejeu automatique en cas d'échec":

Buttons at the bottom: "Sauvegarder" and "Fermer".

6. Renseignez le champ **Adresse(s) e-mail(s) à avertir en cas d'échec**.
7. Pour spécifier plusieurs adresses e-mails, séparez-les par un point-virgule.
8. Configurez le **Rejeu automatique en cas d'échec**.

Cette option permet de renvoyer automatiquement la notification vers le site marchand en cas d'échec, et ce, jusqu'à 4 fois.

Pour plus d'informations, reportez-vous au chapitre **Activer le rejeu automatique**

9. Sauvegardez vos modifications.

Si la plateforme n'arrive pas à joindre l'URL de votre page, alors un e-mail 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 déjà définie plus haut.

### **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

## 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://paiement.systempay.fr/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 : <ul style="list-style-type: none"> <li>• une chaîne de caractères (dans ce cas elle doit être encadrée par des doubles-quotes)</li> <li>• un nombre</li> <li>• un objet</li> <li>• un tableau</li> <li>• un booléen</li> <li>• vide</li> </ul> Exemple: <code>{ "name1":45, "name2":"value2", "name3":false }</code>
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: <code>vads_payment_cards=VISA;MASTERCARD</code>
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 : <ul style="list-style-type: none"> <li>• une chaîne de caractères</li> <li>• un booléen</li> <li>• un objet json</li> </ul>

Notation	Description
	<ul style="list-style-type: none"> <li>un objet xml</li> </ul> <p>La liste des valeurs possibles pour chaque paire de clé/valeur est donnée dans la définition du champ. Exemple: <code>vads_theme_config=SIMPLIFIED_DISPLAY=true;RESPONSIVE_MODEL=Model_1</code></p>

- Les champs obligatoires :

Nom du champ	Description	Format	Valeur
<b>signature</b>	Signature garantissant l'intégrité des requêtes échangées entre le site marchand et la plateforme de paiement.	ans	Ex : <code>ycA5Do5tNvsnKdc/eP1bj2xa19z9q3iWPy9/rpesf50=</code>
<b>vads_action_mode</b>	Mode d'acquisition des données de la carte	enum	<b>INTERACTIVE</b>
<b>vads_amount</b>	Montant du paiement dans sa plus petite unité monétaire (le centime pour l'euro)	n..12	Ex : 3000 pour 30,00 EUR
<b>vads_ctx_mode</b>	Mode de communication avec la plateforme de paiement	enum	<b>TEST</b> ou <b>PRODUCTION</b>
<b>vads_currency</b>	Code numérique de la monnaie à utiliser pour le paiement, selon la norme ISO 4217 (code numérique)	n3	Ex : 978 pour l'euro (EUR)
<b>vads_page_action</b>	Action à réaliser	enum	<b>PAYMENT</b>
<b>vads_payment_config</b>	Type de paiement	enum	<b>SINGLE</b> pour un paiement en 1 fois <b>MULTI</b> pour un paiement en plusieurs fois
<b>vads_site_id</b>	Identifiant de la boutique	n8	Ex : 12345678
<b>vads_trans_date</b>	Date et heure du formulaire de paiement dans le fuseau horaire UTC	n14	Ex : 20170701130025
<b>vads_trans_id</b>	Numéro de la transaction	n6	Ex : 123456
<b>vads_version</b>	Version du protocole d'échange avec la plateforme de paiement	enum	<b>V2</b>

Tableau 1 : Liste des champs obligatoires

- Les champs recommandés :

- Les données de la commande

Nom du champ	Description	Format	Valeur
<b>vads_order_id</b>	Numéro de commande	ans..64	Ex : 2-XQ001
<b>vads_order_info</b>	Informations supplémentaires sur la commande	an..255	Ex : Code interphone 3125
<b>vads_order_info2</b>	Informations supplémentaires sur la commande	an..255	Ex : Sans ascenseur
<b>vads_order_info3</b>	Informations supplémentaires sur la commande	an..255	Ex : Express
<b>vads_nb_products</b>	Nombre d'articles présents dans le panier	n..12	Ex : 2
<b>vads_product_ext_idN</b>	Code barre du produit dans le site web du marchand. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...)	an..100	Ex : vads_product_ext_id0 = "0123654789123654789" vads_product_ext_id1 = "0223654789123654789" vads_product_ext_id2 = "0323654789123654789"
<b>vads_product_labelN</b>	Libellé de l'article. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...)	an..255	Ex : vads_product_label0 = "tee-shirt" vads_product_label1 = "Biscuit" vads_product_label2 = "sandwich"
<b>vads_product_amountN</b>	Montant de l'article. N correspond à l'indice de l'article	n..12	Ex : vads_product_amount0 = "1200"

Nom du champ	Description	Format	Valeur
	(0 pour le premier, 1 pour le second...)		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"

Tableau 2 : Liste des champs - Détails de la commande

- Les données de l'acheteur

Nom du champ	Description
vads_cust_email	Adresse e-mail de l'acheteur.
vads_cust_id	Référence de l'acheteur sur le site marchand.
vads_cust_title	Civilité de l'acheteur.
vads_cust_status	Statut ( <b>PRIVATE</b> : pour particulier / <b>COMPANY</b> pour une entreprise).
vads_cust_first_name	Prénom.
vads_cust_last_name	Nom.
vads_cust_legal_name	Raison sociale de l'acheteur.
vads_cust_cell_phone	Numéro de téléphone mobile.
vads_cust_phone	Numéro de téléphone.
vads_cust_address_number	Numéro de rue.
vads_cust_address	Adresse postale.
vads_cust_district	Quartier.
vads_cust_zip	Code postal.
vads_cust_city	Ville.
vads_cust_state	Etat / Région.
vads_cust_country	Code pays suivant la norme ISO 3166.

Tableau 3 : Liste des champs - Détails de l'acheteur

- 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	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	<b>PRIVATE</b> : pour une livraison chez un particulier <b>COMPANY</b> : pour une livraison en entreprise
vads_ship_to_street_number	Numéro de rue	ans..64	Ex : 2



Nom du champ	Description	Format	Valeur
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

Tableau 4 : Liste des champs - Détails de la livraison

- Les champs facultatifs :

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.**

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.
  - 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://paiement.systempay.fr/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 :  
**59c96b34c74b9375c332b0b6a32e6deec87de2b**

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[] shalhash = md.digest();
            return convertToHex(shalhash);
        }
        catch(Exception e){
            throw new RuntimeException(e);
        }
    }
    private static String convertToHex(byte[] shalhash) {
        StringBuilder builder = new StringBuilder();
        for (int i = 0; i < shalhash.length; i++) {
            byte c = shalhash[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)
{
    /**
     * Fonction qui calcule la signature.
     * $params : tableau contenant les champs à envoyer dans le formulaire.
     * $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
    $signature = base64_encode(hash_hmac('sha256',$contenu_signature, $key, true));
    return $signature;
}
```

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

```
function getSignature($params, $key)
{
    /**
     * Fonction qui calcule la signature.
     * $params : tableau contenant les champs à envoyer dans le formulaire.
     * $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 à la fin
    $contenu_signature .= $key;

    // Application de l'algorithme SHA-1
    $signature = sha1($contenu_signature);
    return $signature ;
}
```

## 6. ANALYSER LE RÉSULTAT DU PAIEMENT

---

L'URL de notification instantanée (IPN - Instant Payment Notification) permet à la plateforme de paiement de notifier automatiquement au site marchand le résultat du paiement.

Les données sont envoyées en mode POST quel que soit le protocole utilisé (http ou https).

### 6.1. Récupérer les données retournées dans la réponse

---

Les données retournées dans la réponse dépendent des paramètres envoyés dans le formulaire de paiement, du type de paiement réalisé et des options de votre boutique. Ces données constituent une liste de champs. Chaque champ contient une valeur réponse. La liste de champs peut être amenée à évoluer.

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**.

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

Exemple de données envoyées lors de la notification d'un paiement :

```
vads_amount = 3000
vads_auth_mode = FULL
vads_auth_number = 3fb0de
vads_auth_result = 00
vads_capture_delay = 0
vads_card_brand = VISA
vads_card_number = 497010XXXXXX0000
vads_payment_certificate = a50d15063b5ec6cb140043138b8d7576470b71a9
vads_ctx_mode = TEST
vads_currency = 978
vads_effective_amount = 3000
vads_site_id = 12345678
vads_trans_date = 20140902094139
vads_trans_id = 454058
vads_validation_mode = 0
vads_version = V2
vads_warranty_result = YES
vads_payment_src = EC
vads_sequence_number = 1
vads_contract_used = 5785350
vads_trans_status = AUTHORISED
vads_expiry_month = 6
vads_expiry_year = 2015
vads_bank_code = 17807
vads_bank_product = A
vads_pays_ip = FR
vads_presentation_date = 20140902094202
vads_effective_creation_date = 20140902094202
vads_operation_type = DEBIT
vads_threeds_enrolled = Y
vads_threeds_cavv = Q2F2dkNhdnZDYXZ2Q2F2dkNhdnY=
vads_threeds_eci = 05
vads_threeds_xid = WXJsVXpHVjFoMktzNmw5dTdleKQ=
vads_threeds_cavvAlgorithm = 2
vads_threeds_status = Y
vads_threeds_sign_valid = 1
vads_threeds_error_code =
vads_threeds_exit_status = 10
vads_trans_uuid= 1cd9994823334e31bbb579b4d716832d
vads_risk_control = CARD_FRAUD=OK;COMMERCIAL_CARD=OK
vads_result = 00
vads_extra_result = 00
vads_card_country = FR
```

```
vads_language = fr
vads_hash = 299d81f4b175bfb7583d904cd19ef5e38b2b79b2373d9b2b4aab74e5753b10bc
vads_url_check_src = PAY
vads_action_mode = INTERACTIVE
vads_payment_config = SINGLE
vads_page_action = PAYMENT
signature = FxGvazgW0dgqOrVrx6bqKZSXh2y5Dp3bWC9HFn33t+Q=
```

## 6.2. Calculer la signature

---

La signature se calcule selon la même logique utilisée lors de la création du formulaire 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.

Pour calculer la signature:

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.
  - 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.

## 6.3. Comparer les signatures

---

Pour s'assurer de l'intégrité de la réponse, vous devez comparer la valeur du champ **signature** reçue dans la réponse, avec celle calculée à l'étape précédente.

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.

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.



## 6.4. 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 déjà 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
<b>ABANDONED</b>	<b>Abandonné</b> Paiement abandonné par l'acheteur. La transaction n'est pas créée et <b>n'est donc pas visible dans le Back Office Marchand</b> .
<b>ACCEPTED</b>	<b>Accepté.</b> 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 " <b>ACCEPTED</b> " ne sont jamais remises en banque.
<b>AUTHORISED</b>	<b>En attente de remise</b> La transaction est acceptée et sera remise en banque automatiquement à la date prévue.
<b>AUTHORISED_TO_VALIDATE</b>	<b>À valider</b> 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 <b>EXPIRED</b> . Le statut <b>Expiré</b> est définitif.
<b>CANCELLED</b>	<b>Annulé</b> La transaction est annulée par le marchand.
<b>CAPTURED</b>	<b>Présenté</b> La transaction est remise en banque.
<b>CAPTURE_FAILED</b>	La remise de la transaction a échoué. Contactez le Support.
<b>EXPIRED</b>	<b>Expiré</b> 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.
<b>INITIAL</b>	<b>En attente</b> Ce statut est spécifique à tous les moyens de paiement nécessitant une intégration par formulaire de paiement en redirection. Ce statut est retourné lorsque : <ul style="list-style-type: none"> <li>• aucune réponse n'est renvoyée par l'acquéreur ou</li> <li>• le délai de réponse de la part de l'acquéreur est supérieur à la durée de session du paiement sur la plateforme de paiement.</li> </ul>

Valeur	Description
	Ce statut est temporaire. Le statut définitif sera affiché dans le Back Office Marchand aussitôt la synchronisation réalisée.
NOT_CREATED	<b>Transaction non créée</b> La transaction n'est pas créée et n'est pas visible dans le Back Office Marchand.
REFUSED	<b>Refusé</b> La transaction est refusée.
SUSPENDED	<b>Suspendu</b> 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 <b>CAPTURED</b> .
UNDER_VERIFICATION	Pour les transactions PayPal, cette valeur signifie que PayPal retient la transaction pour suspicion de fraude. Le paiement restera dans l'onglet <b>Transactions en cours</b> jusqu'à ce que les vérifications soient achevées. La transaction prendra alors l'un des statuts suivants: <b>AUTHORISED</b> ou <b>CANCELED</b> . Une notification sera envoyée au marchand pour l'avertir du changement de statut (Notification sur modification par batch).
WAITING_AUTHORISATION	<b>En attente d'autorisation</b> Le délai de remise en banque est supérieur à la durée de validité de l'autorisation.
WAITING_AUTHORISATION_TO_VALIDATE	<b>A valider et autoriser</b> 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.

Tableau 5 : Valeurs associées au champ `vads_trans_status`

- Récupérez la référence du paiement transmise dans le champ `vads_trans_id`.
- Analysez le champ `vads_payment_config` pour déterminer s'il s'agit d'un **paiement comptant** (unitaire) ou d'un **paiement en plusieurs fois**.

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

Tableau 6 : Analyse du champ `vads_payment_config`

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

Valeur	Description
1	Première échéance
2	Deuxième échéance
3	Troisième échéance
n	N échéance

Tableau 7 : Analyse du champ `vads_sequence_number`

- Analysez le champ `vads_sequence_number` pour connaître le nombre de tentative effectué pour réaliser le paiement.

`vads_payment_config = SINGLE` :

vads_url_check_src	vads_sequence_number	Description
PAY	1	Paielement réglé en 1 tentative
	2	Paielement réglé en 2 tentatives
	3	Paielement réglé en 3 tentatives
BATCH_AUTO	1	Paielement différé réglé en 1 tentative
	2	Paielement différé réglé en 2 tentatives
	3	Paielement différé réglé en 3 tentatives

**Remarque**

*Le paiement en plusieurs fois n'est pas compatible avec la fonctionnalité de tentatives supplémentaires en cas de paiement refusé.*

7. Récupérez la valeur du champ **vads\_trans\_date** pour identifier la date du paiement.
8. 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.
9. Récupérez le montant et la devise utilisée. Pour cela, récupérez les valeurs des champs suivants:

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

*Tableau 8 : Analyse du montant et de la devise utilisée*

10. 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
<b>03</b>	<b>Accepteur invalide</b> 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..). <b>Pour connaître la raison précise du refus, le marchand doit contacter sa banque.</b>
<b>05</b>	<b>Ne pas honorer</b> Ce code est émis par la banque émettrice de la carte. Il est utilisé dans les cas suivants : <ul style="list-style-type: none"> <li>• Date d'expiration invalide,</li> <li>• CVV invalide,</li> <li>• crédit dépassé,</li> <li>• solde insuffisant (etc.)</li> </ul> <b>Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.</b>
<b>51</b>	<b>Provision insuffisante ou crédit dépassé</b> 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. <b>Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.</b>
<b>56</b>	<b>Carte absente du fichier</b> 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.
<b>57</b>	<b>Transaction non permise à ce porteur</b>

Valeur	Description
	Ce code est émis par la banque émettrice de la carte. Il est utilisé dans les cas suivants : <ul style="list-style-type: none"> <li>l'acheteur tente d'effectuer un paiement sur internet avec une carte de retrait,</li> <li>le plafond d'autorisation de la carte est dépassé.</li> </ul> <b>Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.</b>
59	<b>Suspicion de fraude</b> 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. <b>Pour connaître la raison précise du refus, l'acheteur doit contacter sa banque.</b>
60	<b>L'accepteur de carte doit contacter l'acquéreur</b> 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). <b>Contactez le service client pour régulariser la situation.</b>

Tableau 9 : Valeurs associées au champ vads\_auth\_result

11. Récupérez le résultat de l'authentification 3D Secure. 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).

Tableau 10 : Valeurs du champ vads\_threeds\_enrolled

- b. Récupérez le résultat de l'authentification 3D Secure 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.

Tableau 11 : Valeurs du champ vads\_threeds\_status

12. 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.

Valeur	Description
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.

Tableau 12 : Liste des contrôles associés à la fraude

Les valeurs possibles pour **result** sont :

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

Tableau 13 : Liste des contrôles associés à la fraude

### 13. 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 à modifier la marque choisie par le marchand.
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_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.).

Tableau 14 : Analyse de la carte utilisée pour le paiement

- 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. Exemple : 949478																								
auth_result	Code retour de la demande d'autorisation.																								
capture_delay	Délai avant remise (en jours). <ul style="list-style-type: none"> <li>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.</li> </ul>																								
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 " <b>CB</b> ". 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.																								
identifiant	Identifiant unique (token/alias) associé à un moyen de paiement.																								
identifiant_status	Présent uniquement si l'action demandée correspond à la création ou à la mise à jour d'un alias. Valeurs possibles: <table border="1" data-bbox="496 969 1433 1344"> <thead> <tr> <th>Valeur</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>CREATED</b></td> <td>La demande d'autorisation a été acceptée. L'alias (ou RUM pour un paiement SEPA) est créé avec succès.</td> </tr> <tr> <td><b>NOT_CREATED</b></td> <td>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.</td> </tr> <tr> <td><b>UPDATED</b></td> <td>L'alias (ou RUM pour un paiement SEPA) est mis à jour avec succès.</td> </tr> <tr> <td><b>NOT_UPDATED</b></td> <td>L'alias (ou RUM pour un paiement SEPA) n'a pas été mis à jour.</td> </tr> <tr> <td><b>ABANDONED</b></td> <td>Action abandonnée par l'acheteur (débitteur). L'alias (ou RUM pour un paiement SEPA) n'est pas créé et n'apparaîtra pas dans le Back Office Marchand.</td> </tr> </tbody> </table>	Valeur	Description	<b>CREATED</b>	La demande d'autorisation a été acceptée. L'alias (ou RUM pour un paiement SEPA) est créé avec succès.	<b>NOT_CREATED</b>	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.	<b>UPDATED</b>	L'alias (ou RUM pour un paiement SEPA) est mis à jour avec succès.	<b>NOT_UPDATED</b>	L'alias (ou RUM pour un paiement SEPA) n'a pas été mis à jour.	<b>ABANDONED</b>	Action abandonnée par l'acheteur (débitteur). L'alias (ou RUM pour un paiement SEPA) n'est pas créé et n'apparaîtra pas dans le Back Office Marchand.												
Valeur	Description																								
<b>CREATED</b>	La demande d'autorisation a été acceptée. L'alias (ou RUM pour un paiement SEPA) est créé avec succès.																								
<b>NOT_CREATED</b>	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.																								
<b>UPDATED</b>	L'alias (ou RUM pour un paiement SEPA) est mis à jour avec succès.																								
<b>NOT_UPDATED</b>	L'alias (ou RUM pour un paiement SEPA) n'a pas été mis à jour.																								
<b>ABANDONED</b>	Action abandonnée par l'acheteur (débitteur). L'alias (ou RUM pour un paiement SEPA) n'est pas créé et n'apparaîtra pas dans le Back Office Marchand.																								
presentation_date	Pour un paiement par carte bancaire, ce paramètre correspond à la date de remise en banque souhaitée (au format ISO 8601).																								
trans_id	Numéro de transaction.																								
ext_trans_id	Paramètre absent pour le paiement par carte bancaire.																								
trans_uuid	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																								
extra_result	Code numérique du résultat des contrôles de risques. <table border="1" data-bbox="496 1653 1433 2089"> <thead> <tr> <th>Code</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table>	Code	Description	Vide	Pas de contrôle effectué.	00	Tous les contrôles se sont déroulés avec succès.	02	La carte a dépassé l'encours autorisé.	03	La carte appartient à la liste grise du marchand.	04	Le pays d'émission de la carte appartient à la liste grise du marchand.	05	L'adresse IP appartient à la liste grise du marchand.	06	Le code bin appartient à la liste grise du marchand.	07	Détection d'une e-carte bleue.	08	Détection d'une carte commerciale nationale.	09	Détection d'une carte commerciale étrangère.	14	Détection d'une carte à autorisation systématique.
Code	Description																								
Vide	Pas de contrôle effectué.																								
00	Tous les contrôles se sont déroulés avec succès.																								
02	La carte a dépassé l'encours autorisé.																								
03	La carte appartient à la liste grise du marchand.																								
04	Le pays d'émission de la carte appartient à la liste grise du marchand.																								
05	L'adresse IP appartient à la liste grise du marchand.																								
06	Le code bin appartient à la liste grise du marchand.																								
07	Détection d'une e-carte bleue.																								
08	Détection d'une carte commerciale nationale.																								
09	Détection d'une carte commerciale étrangère.																								
14	Détection d'une carte à autorisation systématique.																								

Nom du paramètre	Description	
	Code	Description
	20	Contrôle de cohérence : aucun pays ne correspond (pays IP, pays carte, pays de l'acheteur ).
	30	Le pays de l'adresse IP appartient à la liste grise.
	99	Problème technique rencontré par le serveur lors du traitement d'un des contrôles locaux.
sequence_number	Numéro de séquence.	
trans_status	Statut de la transaction.	

Tableau 15 : Contenu de l'objet JSON

Remarque : les transactions annulées sont également présentes dans le tableau.

14. Enregistrez la valeur du champ **vads\_trans\_uid**. Elle vous permettra d'identifier de manière unique la transaction si vous utilisez les API Web Services.
15. 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.
16. Procédez à la mise à jour de la commande.

## 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.

**Ces tests doivent impérativement être réalisés avant de demander le passage en production.**

### 8.1. Réaliser des tests de paiement

---

Les demandes de paiement de test adressées via le formulaire HTTP POST doivent:

- Contenir la donnée **vads\_ctx\_mode** valorisée à **TEST**.
- Utiliser **la clé de test** précédemment récupérée pour le calcul de la signature.

En phase de test, le marchand peut tester les configurations 3D Secure (si ce dernier est enrôlé 3DS et si l'option 3DS n'est pas désactivée).

Différents cas de paiements peuvent être simulés en utilisant les numéros de carte de test précisés sur la page de paiement.

Toutes les transactions réalisées en mode test sont consultables dans le Back Office Marchand depuis le menu **Gestion >Transaction de test**.

### 8.2. Tester l'URL de notification instantanée (IPN)

---

Vérifiez tout d'abord l'état de l'URL de notification instantanée (également appelée IPN) dans le Back Office Marchand.

Pour cela :

1. Effectuez un clic droit sur une transaction.
  2. Sélectionnez **Afficher le détail de la transaction**.
  3. Vérifiez le statut de l'URL de notification instantanée (IPN).
    - Dans le cas où le statut est **Envoyé**, cela signifie que vous avez correctement renseigné l'URL dans le Back Office Marchand.
    - Dans le cas où le statut apparaît en **URL non définie**, cela signifie que vous n'avez pas renseigné l'URL dans le Back Office Marchand.
      1. Vérifiez l'adresse de l'URL de notification instantanée saisie en mode TEST et PRODUCTION.
      2. Cliquez sur **Paramétrage > Règles de notification**.
      3. Renseignez l'URL de notification de paiement instantanée (URL de notification à la fin du paiement).

Ne saisissez pas une adresse en "localhost". L'appel à cette l'URL se fait de serveur à serveur.
  4. Cliquez sur **Sauvegarder**.
- Dans le cas où le statut est **Echoué**, se reporter au chapitre **Traiter les erreurs** du guide d'implémentation API Formulaire disponible sur notre site documentaire.

## 9. ACTIVER LA BOUTIQUE EN MODE PRODUCTION

---

Ce chapitre vous détaille de quelle manière vous pouvez :

- Générer la clé de production.
- Basculer votre site marchand en production.
- Réaliser un premier paiement en production.
- Régénérer la clé de production (en cas de problème).

### 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.  
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.
- 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**.