



Réussir l'intégration du paiement mobile via webview

Guide d'implémentation

Version du document 1.5

Sommaire

1. HISTORIQUE DU DOCUMENT.....	3
2. PRÉSENTATION.....	4
3. CINÉMATIQUE DU PAIEMENT.....	5
4. INTÉGRATION DU PAIEMENT.....	6
5. PHASE 1: LE SERVEUR MARCHAND.....	7
5.1. Création du formulaire de paiement.....	7
Transmettre les données de l'acheteur.....	10
Transmettre les données de la commande.....	11
Transmettre les données de livraison.....	13
5.2. Calculer la signature.....	14
5.3. Transfert de la demande de paiement.....	16
5.4. Réception de l'URL de paiement.....	16
5.5. Traitement de la notification de fin de paiement (IPN).....	16
5.6. Exemple de code.....	17
6. PHASE 2: L'APPLICATION MOBILE.....	18
6.1. Camera card scanning.....	18
6.2. NFC card scanning.....	19

1. HISTORIQUE DU DOCUMENT

Version	Auteur	Date	Commentaire
1.5	Groupe Crédit du Nord	12/05/2020	<ul style="list-style-type: none">• Intégration coté serveur : ajout des champs du formulaire pour transmettre les données de la commande, de l'acheteur et de livraison.• Ajout du calcul de signature.• Refonte de l'intégration coté application mobile.• Ajout du support du scan de la carte par camera et NFC.
1.4	Groupe Crédit du Nord	16/10/2019	Version initiale

Ce document et son contenu sont strictement confidentiels. Il n'est pas contractuel. Toute reproduction et/ou distribution de tout ou partie de ce document ou de son contenu à une entité tierce sont strictement interdites ou sujettes à une autorisation écrite préalable de Groupe Crédit du Nord. Tous droits réservés.

2. PRÉSENTATION

Clic&Pay by CDN vous propose une solution unique pour l'intégration du paiement mobile à vos applications.

Notre solution couvre les applications natives iOS et Android. Elle est basée sur l'utilisation du composant **webview**.

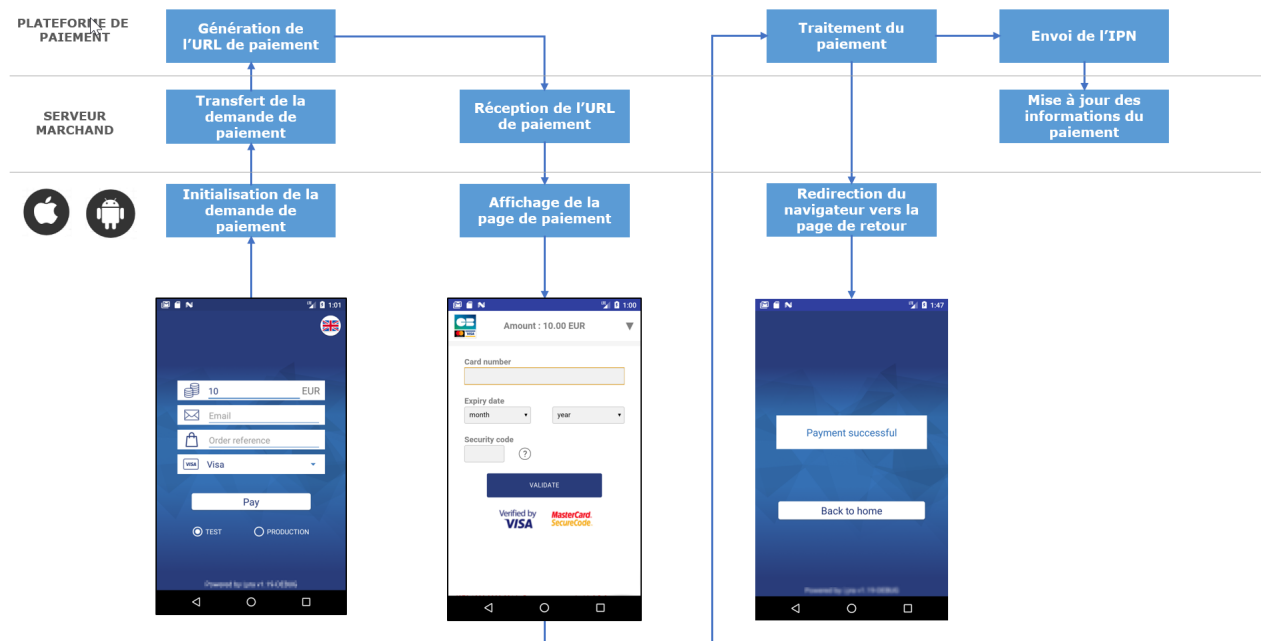
Une webview permet d'afficher du contenu déjà disponible sur le web au sein de l'application.

Ainsi, la solution Clic&Pay by CDN de paiement mobile via webview offre plusieurs avantages au marchand :

- Une configuration unique pour le web et le mobile.
Vous pouvez réutiliser à l'identique la configuration des paiements de votre site web.
Les moyens de paiement activés, les règles anti-fraudes, etc. sont repris dans l'application mobile.
- Une cohérence dans l'affichage des informations du parcours acheteur.
Nos pages de paiement sont responsives et donc, capables de s'adapter aux différents terminaux de vos clients (mobile, tablette ou ordinateur de bureau).
- Une haute sécurité grâce, d'une part, à notre certification PCI DSS et d'autre part à la gestion du 3DS intégrée dans le parcours de paiement.

PCI DSS (= Payment Card Industry Data Security Standard) est la norme de sécurité de l'industrie des cartes de paiement. C'est un standard de sécurité des données pour les principaux groupes de cartes de paiement tels que Visa, MasterCard, American Express, Discover et JCB.

3. CINÉMATIQUE DU PAIEMENT



L'acheteur valide son panier.

1. L'application mobile initialise une demande de paiement auprès du serveur marchand.
2. Le serveur marchand envoie une demande de paiement à la plateforme.
3. La plateforme génère une URL de paiement et la transmet en retour à l'application mobile.
4. Le serveur marchand envoie l'URL de paiement à l'application mobile.
5. L'application mobile ouvre la page de paiement dans une webview.
6. L'acheteur saisit les données de sa carte puis clique sur **Valider**.
7. La plateforme procède au paiement, puis transmet la notification de paiement au serveur marchand.
8. Le serveur marchand analyse le résultat du paiement.
9. L'acheteur est redirigé automatiquement vers l'application du marchand.

4. INTÉGRATION DU PAIEMENT

Des exemples de codes sont mis à disposition pour faciliter l'intégration :

Serveur marchand <https://github.com/lyra/webview-payment-sparkjava-integration-sample>

iOS <https://github.com/lyra/webview-payment-ios-integration-sample>

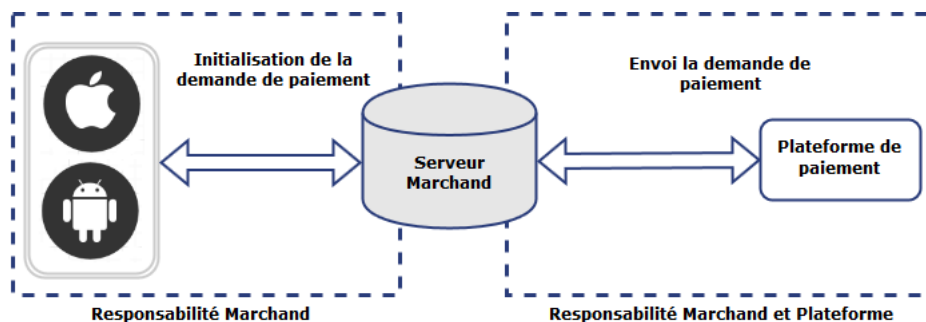
Android <https://github.com/lyra/webview-payment-android-integration-sample>

IMPORTANT

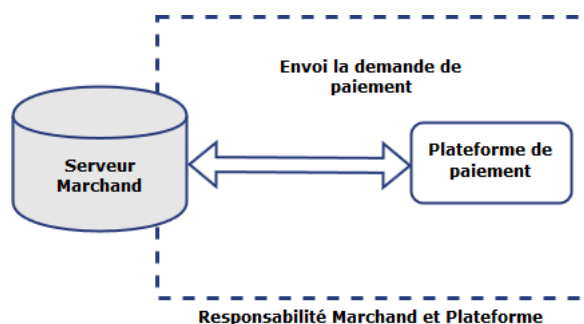
Assurez-vous d'avoir pris connaissance des commentaires présents dans les fichiers readme avant de lancer l'application. Les fichiers `MainActivity.kt` et `app-configuration.properties` doivent être modifiés suivants les instructions fournies dans les commentaires.

L'intégration se déroule en deux phases:

- intégration des échanges entre le serveur marchand et la plateforme de paiement
- intégration des échanges entre l'application mobile et le serveur marchand.



5. PHASE 1: LE SERVEUR MARCHAND



5.1. Création du formulaire de paiement

Le serveur marchand reçoit une demande de paiement de la part de l'application mobile et doit la transmettre à la plateforme de paiement.

Pour cela, le site marchand va générer un formulaire de paiement HTML qu'il postera à la plateforme de paiement.

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

Le serveur marchand transmettra la signature alphanumérique dans le formulaire de paiement (voir chapitre Calculer la signature à la page 14).

IMPORTANT

Toutes les données de votre formulaire doivent être encodées en UTF-8.

Les caractères spéciaux (accents, ponctuations, 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.

Pour créer le formulaire de paiement :

1. Utilisez l'ensemble des champs présents dans ce tableau pour construire la demande de paiement.

Nom du champ	Description	Format	Valeur
vads_site_id	Identifiant de la boutique	n8	Ex : 12345678
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_amount	Montant du paiement dans sa plus petite unité monétaire (le centime pour l'euro)	n..12	Ex : 3000 pour 30,00 EUR
vads_cust_email	Adresse e-mail de l'acheteur	ans..150	Ex: abc@example.com
vads_payment_cards	Type de carte.	String	Ex: VISA (Consultez le <i>Guide d'implémentation - API Formulaire</i> pour connaître la liste des valeurs possibles).
vads_order_id	Numéro de commande	ans..64	Ex : 2-XQ001
vads_version	Version du protocole d'échange avec la plateforme de paiement	enum	V2
vads_theme_config	Permet de gagner en performances en désactivant	map	SIMPLIFIED_DISPLAY=true

Nom du champ	Description	Format	Valeur
	des éléments de la page de paiement comme le sélecteur de langue, les logos du bas de page, etc.		
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	an6	Ex : xrT15p
vads_payment_config	Type de paiement	enum	SINGLE pour un paiement en 1 fois MULTI pour un paiement en plusieurs fois
vads_page_action	Action à réaliser	enum	PAYMENT
vads_ctx_mode	Mode de communication avec la plateforme de paiement	enum	TEST ou PRODUCTION
vads_action_mode	Mode d'acquisition des données du moyen de paiement	enum	INTERACTIVE
signature	Signature garantissant l'intégrité des requêtes échangées entre le site marchand et la plateforme de paiement. Le calcul de sa valeur est précisé ici : Calculer la signature à la page 14.	ans44	Ex: NrHSHyBBBc +TtcaudspNHQ5cYcy4tS4ljvdC0ztFe8=

2. Utilisez les champs ci-dessous pour gérer le retour vers l'application mobile à la fin du paiement.

Un paiement peut aboutir sur 4 états différents :

- Paiement accepté
- Paiement refusé
- Paiement en erreur
- Paiement abandonné par l'acheteur.

Pour chaque état vous devez associer une URL:

Nom du champ	Description	Format	Valeur
vads_url_success	URL où sera redirigé l'acheteur, en cas de succès.	ans..1024	Ex: http://webview.success
vads_url_refused	URL où sera redirigé l'acheteur, en cas de refus.	ans..1024	Ex: http://webview.refused
vads_url_cancel	URL où sera redirigé l'acheteur en cas d'abandon ou d'expiration (timeout).	ans..1024	Ex: http://webview.cancel
vads_url_error	URL où sera redirigé l'acheteur en cas d'erreur.	ans..1024	Ex: http://webview.error

3. Utilisez les champs ci-dessous pour configurer les temps de redirection vers l'application mobile à la fin du paiement:

Nom du champ	Description	Format
vads_redirect_success_timeout	Définit le délai d'attente avant redirection après un paiement réussi. Ce délai est exprimé en seconde et doit être compris entre 0 et 300 secondes. Valorisez ce champ à "0" pour ne pas afficher le ticket de paiement et rediriger automatiquement l'acheteur vers l'application mobile.	n..3

Nom du champ	Description	Format
vads_redirect_error_timeout	Définit le délai d'attente avant redirection après un paiement refusé. Ce délai est exprimé en seconde et doit être compris entre 0 et 300 secondes. Valorisez ce champ à "0" pour ne pas afficher la page de refus de paiement et rediriger automatiquement l'acheteur vers l'application mobile.	n..3

4. Ajoutez les autres champs optionnels en fonction de vos besoins (voir sous chapitres suivants).

Transmettre les données de l'acheteur

Le marchand peut transmettre des informations concernant l'acheteur (adresse e-mail, civilité, numéro de téléphone etc.). Ces données constitueront les informations de facturation.

Toutes les données qui seront transmises via le formulaire de paiement seront affichées dans le Back Office Marchand en consultant le détail de la transaction (onglet **Acheteur**).

Utilisez les champs facultatifs ci-dessous en fonction de vos besoins. *Ces champs seront renvoyés dans la réponse avec la valeur transmise dans le formulaire.*

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_title	Civilité de l'acheteur	an..63	Ex : Monsieur
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_cell_phone	Numéro de téléphone mobile	an..32	Ex: 06 12 34 56 78
vads_cust_address_number	Numéro de rue	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 États-Unis.

Remarque

Les champs **vads_cust_phone** et **vads_cust_cell_phone** acceptent tous les formats:

Exemples:

- 0123456789
- +33123456789
- 0033123456789
- (00.571) 638.14.00
- 40 41 42 42

Transmettre les données de la commande

Le marchand peut transmettre les informations de la commande (numéro de la commande, description, contenu du panier etc.).

Ces données seront affichées dans le Back Office Marchand en consultant le détail de la transaction.

1. Utilisez les champs facultatifs ci-dessous en fonction de vos besoins. Ces champs seront renvoyés dans la réponse avec la valeur transmise dans le formulaire.

Nom du champ	Description	Format	Valeur
vads_order_info	Informations supplémentaires sur la commande	an..255	Ex : Code interphone 3125
vads_order_info2	Informations supplémentaires sur la commande	an..255	Ex : Sans ascenseur
vads_order_info3	Informations supplémentaires sur la commande	an..255	Ex : Express
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 correspond à l'indice de l'article (0 pour le premier, 1 pour le second...)		Ex : vads_product_ext_id0 = "0123654789123654789" vads_product_ext_id1 = "0223654789123654789"
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 = "Séjour 3 jours datés" vads_product_label1 = "Concert privé"
vads_product_amountN	Montant de l'article exprimé dans la plus petite unité de la devise. N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).	n..12	Ex: vads_product_amount0 = "32150" vads_product_amount1 = "10700"
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 = "TRAVEL" vads_product_type1 = "ENTERTAINMENT"
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 = "1002127784" vads_product_ref1 = "1002127693"
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 = "1"

2. Valorisez le champ **vads_nb_products** avec le nombre d'articles contenu dans le panier.

Remarque :

Ce champ devient obligatoire pour que le panier soit pris en compte.

*En le renseignant, 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_**.*

3. Valorisez le champ **vads_product_amountN** avec le montant des différents articles contenus dans le panier dans l'unité la plus petite de la devise.

N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).

4. Valorisez le champ `vads_product_typeN` avec la valeur correspondant au type de l'article.

N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).

Valeur	Description
FOOD_AND_GROCERY	Produits alimentaires et d'épicerie
AUTOMOTIVE	Automobile / Moto
ENTERTAINMENT	Divertissement / Culture
HOME_AND_GARDEN	Maison et jardin
HOME_APPLIANCE	Equipement de la maison
AUCTION_AND_GROUP_BUYING	Ventes aux enchères et achats groupés
FLOWERS_AND_GIFTS	Fleurs et cadeaux
COMPUTER_AND_SOFTWARE	Ordinateurs et logiciels
HEALTH_AND_BEAUTY	Santé et beauté
SERVICE_FOR_INDIVIDUAL	Services à la personne
SERVICE_FOR_BUSINESS	Services aux entreprises
SPORTS	Sports
CLOTHING_AND_ACCESSORIES	Vêtements et accessoires
TRAVEL	Voyage
HOME_AUDIO_PHOTO_VIDEO	Son, image et vidéo
TELEPHONY	Téléphonie

5. Valorisez le champ `vads_product_labelN` avec le libellé de chacun des articles contenus dans le panier.

N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).

6. Valorisez le champ `vads_product_qtyN` avec la quantité de chacun des articles contenus dans le panier.

N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).

7. Valorisez le champ `vads_product_refN` avec la référence de chacun des articles contenus dans le panier.

N correspond à l'indice de l'article (0 pour le premier, 1 pour le second...).

8. Contrôlez la valeur du champ `vads_amount`. Elle doit correspondre au montant total de la commande.

Transmettre les données de livraison

Le marchand peut transmettre les données de livraison de l'acheteur (adresse, civilité, numéro de téléphone, etc.).

Ces données seront affichées dans le Back Office Marchand en consultant le détail de la transaction (onglet **Livraison**).

Utilisez les champs facultatifs ci-dessous en fonction de vos besoins. *Ces champs seront renvoyés dans la réponse avec la valeur transmise dans le formulaire.*

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 rue	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

5.2. Calculer la signature

Afin de pouvoir calculer la valeur du champ **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**.

Attention vous ne devez pas utiliser les clés d'API REST pour calculer la signature de votre formulaire de paiement.

Seule l'utilisation de l'algorithme HMAC-SHA-256 est implémentée dans notre exemple de code.

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. Calculez et encodez au format Base64 la signature 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://clilandpay.groupecdn.fr/vads-payment/entry.silentInit.a">
<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. 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=

5.3. Transfert de la demande de paiement

L'API de création de paiement est disponible en mode POST à cette adresse :

<https://clicandpay.groupecdn.fr/vads-payment/entry.silentInIt.a>

IMPORTANT

L'URL de l'API de création de paiement est différente de l'URL de la page de paiement, telle que décrite dans le *Guide d'implémentation - API Formulaire*.

5.4. Réception de l'URL de paiement

La plateforme de paiement retourne une réponse au format JSON contenant un code d'état HTTP de succès ou d'erreur.

Succès

En cas de succès, la plateforme de paiement renvoie un code d'état HTTP `200 OK`.

La réponse contient l'URL de paiement vers laquelle l'application mobile doit rediriger l'acheteur.

```
{
  "status": "INITIALIZED",
  "redirect_url": "https://clicandpay.groupecdn.fr:443/vads-payment/
exec.refresh.a;jsessionid=CE2Cb9daEDe7f6dBF31FE65e.vadpayment01bdx"
}
```

Erreur

En cas d'erreur, la plateforme de paiement renvoie un code d'état HTTP `400 Bad Request` OU `500 Internal Server Error`.

La réponse contiendra le détail de l'erreur.

```
{
  "status": "ERROR",
  "error": { "code": "09", "value": "Missing or invalid parameter value" }
}
```

Pour plus de détails, consultez la liste des codes d'erreur de l'API Formulaire:

<https://clicandpay.groupecdn.fr/doc/fr-FR/error-code/error-00.html>

5.5. Traitement de la notification de fin de paiement (IPN)

Une fois le paiement effectué, la plateforme de paiement notifie le serveur marchand du résultat de la transaction.

Les données seront envoyées sur l'URL de notification définie dans le Back Office Marchand.

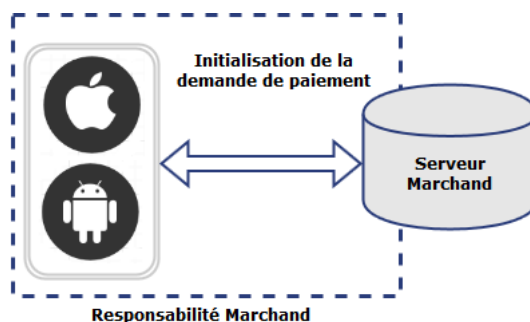
Consultez le *Guide d'implémentation - API Formulaire* pour plus d'informations sur le paramétrage des règles de notification et sur l'analyse des données transmises.

5.6. Exemple de code

Consultez l'exemple complet pour déployer le serveur marchand :

<https://github.com/lyra/webview-payment-sparkjava-integration-sample>

6. PHASE 2: L'APPLICATION MOBILE



L'intégration dans votre application mobile doit suivre les étapes suivantes :

1. Initialisation de la demande de paiement en transmettant les données nécessaires,

L'application génère une "payload" contenant les données du panier, les coordonnées de l'acheteur, les informations de livraison et transmet la demande de paiement au serveur marchand via une requête POST.

2. Affichage de la page de paiement dans une webview,

L'application initialise une webview et affiche la page de paiement en utilisant l'URL retournée par la plateforme de paiement.

3. Détection de la fin de paiement.

L'application mobile doit analyser les différentes URLs qui passent par la webview. Les URL de retour étant définies par le serveur marchand,, vous avez le contrôle sur la cinématique de paiement et pouvez décider à quel moment basculer sur votre application native.

Retrouvez tous les détails de l'intégration :

iOS <https://github.com/lyra/webview-payment-ios-integration-sample/>
Android <https://github.com/lyra/webview-payment-android-integration-sample>

6.1. Camera card scanning

Il est possible d'utiliser l'appareil photo numérique de l'équipement mobile pour pré-remplir les données de la carte au moment du paiement.

Un exemple complet d'intégration est fourni ici :

iOS https://github.com/lyra/webview-payment-iOS-integration-sample/tree/card_scanning
Android https://github.com/lyra/webview-payment-android-integration-sample/tree/card_scanning

Cet exemple utilise des bibliothèques externes développées par des tiers.

IMPORTANT

Groupe Crédit du Nord ne garantit pas et n'est pas responsable de la qualité des bibliothèques externes.
L'utilisation de ces bibliothèques n'est pas compatible PCI-DSS.

6.2. NFC card scanning

Il est possible d'utiliser le module NFC de l'équipement mobile pour pré-remplir les données de la carte au moment du paiement.

Un exemple complet d'intégration est fourni ici :

iOS Non disponible.

Android https://github.com/lyra/webview-payment-android-integration-sample/tree/card_scanning_by_nfc

Cet exemple utilise des bibliothèques externes développées par des tiers.

IMPORTANT

**Groupe Crédit du Nord ne garantit pas et n'est pas responsable de la qualité des bibliothèques externes.
L'utilisation de ces bibliothèques n'est pas compatible PCI-DSS.**