



API Webservices SOAP v5

Guide d'implémentation

Version du document 2.2

Sommaire

1. HISTORIQUE DU DOCUMENT.....	4
2. CONTACTER L'ASSISTANCE TECHNIQUE.....	5
3. PRÉSENTATION DES WEB SERVICES.....	6
3.1. Description des web services.....	6
L'en-tête SOAP HEADER.....	7
Le corps (Body).....	7
3.2. Gérer les codes d'erreur et les exceptions.....	9
Gérer les exceptions.....	9
Gérer les erreurs applicatives.....	10
3.3. Enchaîner des requêtes Web service (maintenir une même session HTTP).....	14
3.4. Gérer les délais d'attente (Timeout).....	16
3.5. Spécifier les types de données.....	17
4. S'IDENTIFIER LORS DES ÉCHANGES.....	18
4.1. Procéder à l'authentification.....	19
4.2. Construire l'en-tête SOAP HEADER de la requête.....	20
Exemple de code en PHP pour construire l'en-tête SOAP HEADER.....	22
4.3. Vérifier l'en-tête SOAP dans la réponse.....	23
Exemple de code PHP pour récupérer les en-têtes SOAP dans la réponse.....	23
5. GÉNÉRER UN UUID - RÉTROCOMPATIBILITÉ.....	24
5.1. Requête à envoyer.....	24
legacyTransactionKeyRequest.....	25
5.2. Réponse en retour.....	26
commonResponse.....	26
paymentResponse.....	26
6. RÉALISER DES OPÉRATIONS COURANTES SUR LES TRANSACTIONS.....	27
6.1. Créer une transaction de paiement 'createPayment'.....	27
Le partage d'identifiants.....	27
Requête à envoyer.....	29
Réponse en retour.....	41
Créer un paiement sans authentification 3D Secure.....	55
Créer un paiement avec authentification 3D Secure.....	58
Rejouer un paiement refusé.....	66
6.2. Modifier une transaction de paiement 'updatePayment'.....	67
Requête à envoyer.....	67
Réponse en retour.....	71
6.3. Modifier les informations (panier) d'une transaction 'updatePaymentDetails'.....	84
Requête à envoyer.....	84
Réponse en retour.....	86
6.4. Annuler une transaction de paiement 'cancelPayment'.....	101
Requête à envoyer.....	101
Réponse en retour.....	102
6.5. Rechercher des paiements 'findPayments'.....	104
Requête à envoyer.....	104
Réponse en retour.....	105
6.6. Rembourser un acheteur 'refundPayment'.....	108
Requête à envoyer.....	108
Réponse en retour.....	110
6.7. Dupliquer une transaction de paiement 'duplicatePayment'.....	124
Requête à envoyer.....	124
Réponse en retour.....	127
6.8. Valider une transaction de paiement 'validatePayment'.....	141
Requête à envoyer.....	141

Réponse en retour.....	142
6.9. Remiser une transaction de paiement 'capturePayment'.....	144
Requête à envoyer.....	144
Réponse en retour.....	145
6.10. Obtenir le détail d'une transaction de paiement 'getPaymentDetails'.....	146
Requête à envoyer.....	146
Réponse en retour.....	147
6.11. Vérifier l'authentification 3D Secure 'verifyThreeDSEnrollment'.....	160
Requête à envoyer.....	160
Réponse en retour.....	163
6.12. Vérifier le statut de l'authentification 3D Secure 'checkThreeDSAuthentication'.....	165
Requête à envoyer.....	165
Réponse en retour.....	166

7. RÉALISER DES OPÉRATIONS SPÉCIFIQUES AUX PAIEMENTS PAR IDENTIFIANT..... 168

7.1. Créer un alias (identifiant) 'createToken'.....	169
Le partage d'identifiants.....	169
Requête à envoyer.....	169
Réponse en retour.....	176
7.2. Créer un alias (identifiant) à partir d'une transaction 'createTokenFromTransaction'.....	178
Requête à envoyer.....	178
Réponse en retour.....	180
7.3. Modifier un alias (identifiant) 'updateToken'.....	182
Requête à envoyer.....	182
Réponse en retour.....	188
7.4. Récupérer le détail d'un alias (identifiant) 'getTokenDetails'.....	190
Requête à envoyer.....	190
Réponse en retour.....	191
7.5. Résilier un alias (identifiant) 'cancelToken'.....	196
Requête à envoyer.....	196
Réponse en retour.....	197
7.6. Réactiver un alias (identifiant) 'reactivateToken'.....	198
Requête à envoyer.....	198
Réponse en retour.....	199
7.7. Réaliser des paiements récurrents (abonnements) 'createSubscription'.....	200
Requête à envoyer.....	200
Réponse en retour.....	204
7.8. Modifier un abonnement 'updateSubscription'.....	206
Requête à envoyer.....	206
Réponse en retour.....	209
7.9. Récupérer le détail d'un abonnement 'getSubscriptionDetails'.....	210
Requête à envoyer.....	210
Réponse en retour.....	211
7.10. Annuler un abonnement 'cancelSubscription'.....	214
Requête à envoyer.....	214
Réponse en retour.....	215

8. EXEMPLES EN PHP..... 216

8.1. createPayment.....	228
8.2. findPayments.....	233
8.3. createToken.....	235
8.4. createSubscription.....	237
8.5. cancelSubscription.....	240

9. DICTIONNAIRE DE DONNÉES..... 242

9.1. Codes réponse d'une demande d'autorisation (authorizationResponse.result).....	243
9.2. Codes réponse de l'analyseur de risque externe (fraudManagementResponse.riskAnalysis.resultCode).....	244
9.3. Codes d'erreurs pouvant survenir durant le paiement (paymentResponse.paymentError)....	246
9.4. Codes produits VISA / MASTERCARD (cardResponse.productCode).....	249
9.5. Liste des devises supportées (paymentRequest.currency).....	251
9.6. Liste des types de carte supportées (cardRequest.scheme / cardResponse.scheme).....	251

1. HISTORIQUE DU DOCUMENT

Version	Auteur	Date	Commentaire
2.3	CSB (Calédonienne de Services Bancaires)	xx/xx/2018	<ul style="list-style-type: none">• Suppression de l'objet paymentRequest de la description des paramètres d'entrée de la méthode updateSubscription• Ajout du paramètre cardRequest.walletPayload• Ajout du Dictionnaire de données contenant la liste des valeurs possibles pour les champs: authorizationResponse.result, fraudManagementResponse.riskAnalysis.result, cardResponse.productCode, paymentRequest.currency, cardRequest.scheme.• Mise à jour de la liste des valeurs du champ paymentResponse.paymentError.• Mise à jour des définitions suivantes: paymentRequest.acquirerTransientData, extraResponse.paymentOptionCode, extraResponse.paymentOptionOccNumb, extraResponse.boletoPdfUrl.
2.2	CSB (Calédonienne de Services Bancaires)	31/05/2018	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 CSB (Calédonienne de Services Bancaires). Tous droits réservés.

2. CONTACTER L'ASSISTANCE TECHNIQUE

Pour toute question technique ou demande d'assistance, nos services sont disponibles de 07h30 à 17h30

par téléphone au : (687) 46 33 33
par e-mail : sav@csb.nc

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 (menu **Paramétrage** > **Boutique** > **Configuration**).

3. PRÉSENTATION DES WEB SERVICES

Ce document détaille une collection d'opérations pouvant être appelées à distance via Internet indépendamment des langages de programmation et des plateformes utilisés.

Les web services sont utilisés pour intégrer une ou plusieurs fonctionnalités de paiement à un progiciel de gestion intégré.

Deux types d'opérations sont mis à disposition :

Opérations courantes sur les transactions

- Créer des paiements (avec ou sans authentification 3D Secure).
- Automatiser les actions sur les transactions (remboursement, annulation...).

Pour utiliser cette solution, le marchand doit souscrire aux options suivantes:

- "Accès à l'acquisition sur site marchand via web-service" pour la création des paiements
- "Paiement par web services" pour toutes les autres opérations.

Opérations spécifiques aux paiements par identifiant

- Gérer des alias (utilisés pour le paiement en un clic).
- Gérer des abonnements.

Pour utiliser cette solution, le marchand doit souscrire à l'option "Paiement par identifiant".

Contactez le service clients pour plus de renseignements.

3.1. Description des web services

Les web services sont développés suivant le protocole SOAP version 1.2 (Simple Object Access Protocol) et sont décrits par le fichier wsdl suivant :

<https://epaync.nc/vads-ws/v5?wsdl>

Remarque : si vous passez par un proxy pour vous connecter à Internet depuis un serveur applicatif, veuillez-vous rapprocher de votre service informatique pour savoir s'il est nécessaire de configurer l'accès à cette URL.

Un message SOAP est contenu dans une enveloppe.

La structure d'une enveloppe SOAP est définie de la façon suivante :

- **un en-tête (Header)**
Il contient des informations sur le traitement du message.
- **un corps (Body)**
Il contient les informations de la requête ou de la réponse
- **une gestion d'erreurs contenue dans le corps**

L'en-tête SOAP HEADER

Le HEADER est un en-tête qui véhicule des informations permettant d'authentifier et sécuriser les données échangées entre le site marchand et la plateforme de paiement.

Il contient :

- **shopId**
Identifiant de la boutique du marchand.
- **requestId**
UUID (identifiant universel unique).
Sa valeur permet de calculer le jeton d'authentification.
- **timestamp**
Représentation numérique de la date et de l'heure de la requête au format ISO 8601 - W3C et UTC.
- **mode**
Type de transaction.
Il peut être valorisé à **TEST** (pour une transaction de test) ou **PRODUCTION** (pour une transaction réelle).
- **authToken**
Jeton d'authentification.
Il doit être transmis systématiquement et les valeurs qu'il contient doivent être recalculées à chaque appel.

Exemple de HEADER dans la requête et dans la réponse :

```
<soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">
  <soapHeader:shopId>12345678</soapHeader:shopId>
  <soapHeader:requestId>04967dae-af01-43ff-a7d8-f3f228b9b1c2</soapHeader:requestId>
  <soapHeader:timestamp>2014-10-31T16:38:19Z</soapHeader:timestamp>
  <soapHeader:mode>TEST</soapHeader:mode>
  <soapHeader:authToken>NxofUSStqmMjwaDzTXyCN4nNpMOVJKb5UxHdS9TBuTg=</soapHeader:authToken>
</soap:Header>
```

Le corps (Body)

Le corps du message SOAP est contenu dans une balise <Body> obligatoire.

Il contient les données échangées entre le client et le service sous la forme d'un fragment de document XML. Ces données correspondent à la requête ou à la réponse.

Les messages échangés entre le marchand et la plateforme de paiement sont élaborés en respectant une syntaxe précise (voir chapitre **Spécifier les types de données**).

Exemple de BODY :

Requête

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:v5="http://v5.ws.vads.lyra.com/">
  <soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">
    ...
  </soap:Header>
  <soap:Body>
    <myOperationRequest>
      ...
    </myOperationRequest>
  </soap:Body>
</soap:Envelope>
```

Réponse

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    ...
  </env:Header>
  <soap:Body>
    <myOperationResponse>
      <return>
        ...
        <commonResponse>
          <responseCode>0</responseCode>
          <responseCodeDetail>Action successfully completed</responseCodeDetail>
          ...
        </commonResponse>
        ...
      </return>
    </myOperationResponse>
  </soap:Body>
</soap:Envelope>
```

3.2. Gérer les codes d'erreur et les exceptions

Les opérations web services effectuent différents contrôles sur les paramètres de la requête.

Elles sont susceptibles à ce titre de produire deux types de retour en cas d'erreurs :

- les exceptions (SOAP Fault exceptions),
- les erreurs applicatives.

Gérer les exceptions

Gérer les exceptions

Les exceptions levées par une méthode web services sont renvoyées au site marchand sous la forme d'un élément XML **Fault**.

Une exception est renvoyée par exemple lorsque les attributs des objets nécessaires à l'exécution des opérations sont mal formatés.

Exemple :

Une adresse e-mail mal formatée (sans @) lors de la création d'un alias.

L'exception `<soap:Fault>` contiendra des détails tels que la chaîne d'exception et la source de l'exception.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope"/>
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value xmlns:ns1="http://www.w3.org/2003/05/soap-envelope">ns1:Sender</soap:Value>
      </soap:Code>
      <soap:Reason>
        <soap:Text xml:lang="en">CreateRequest.customerRequest.billingDetails.email: Adresse email mal formée</soap:Text>
      </soap:Reason>
      <soap:Detail>
        <requestId>43a61cf4-e467-490e-871e-d61604577cb0</requestId>
      </soap:Detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Ce mécanisme de gestion des exceptions permet d'identifier et corriger des données dont le format est incorrect avant d'exécuter une opération.

Gérer les erreurs applicatives

Gérer les erreurs applicatives

Des messages d'erreurs applicatifs normalisés peuvent être envoyés dans le message SOAP réponse lié à un appel.

Ils se présentent sous la forme d'un code d'erreur associé à une description du problème rencontré.

Le tableau ci-dessous récapitule les codes d'erreurs susceptibles d'être retournés dans l'attribut **responseCode** :

Code d'erreur	responseCodeDetail	Description
0	Action successfully completed	Action réalisée avec succès
1	Unauthorized request	Action non autorisée.
2	Bad Parameter	Attribut invalide.
3	Bad Request	La requête n'a pu être traitée.
10	Transaction was not found	Transaction non trouvée.
11	Bad transaction status	Statut de la transaction incorrect.
12	Transaction already exists	Transaction existe déjà.
13	Date is too far from current UTC date	Mauvaise date (la valeur de l'attribut 'submissionDate' est trop loin de la date actuelle).
14	Nothing has changed	Aucun changement.
15	Too much results	Trop de résultats.
20	Bad amount	Montant invalide dans l'attribut 'amount'.
21	Unknown currency	Devise invalide dans l'attribut 'currency'.
22	Unknown card type	Type de carte inconnu.
23	Invalid Expiration Date	Date invalide dans les attributs 'expiryMonth' et/ou 'expiryYear'.
24	CVV Mandatory	Le 'cvv' est obligatoire.
25	Contract not found	Numéro de contrat inconnu.
26	Invalid card number	Le numéro de carte est invalide.
30	Payment token not found	L'alias n'est pas trouvé.
31	Invalid payment token (cancelled, ...)	L'alias est invalide (Résilié, vide...).
32	SubscriptionID was not found	Attribut 'subscriptionId' non trouvé.
33	Invalid Subscription	Attribut 'rrule' invalide ou Abonnement déjà résilié
34	Payment token already exists	L'alias existe déjà.
35	Payment token creation declined	Création de l'alias refusé.
36	Payment token purged	Attribut 'paymentToken' purgé.
40	Amount not authorized	Attribut 'amount' non autorisé
41	Card range not found	Plage de carte non trouvée
42	Not enough credit	Le solde du moyen de paiement n'est pas suffisant.
43	No credit	Le remboursement n'est pas autorisé pour ce contrat.
50	Brand not found	Aucune brand localisée.
51	Merchant not enrolled	Marchand non enrôlé.
52	Invalid ACS Signature	Signature de l'ACS invalide.
53	Technical error 3DS	Erreur technique 3DS.
54	Wrong Parameter 3DS	Paramètre 3DS incorrect.
55	3DS Disabled	3DS désactivé.
56	PAN not found	PAN non trouvé.
97	OneyWsError	OneyWs Erreur.
98	Bad request Id	Attribut RequestId invalide.
99	Undefined Error	Erreur inconnue.

Précisions sur les codes d'erreurs

- **0 - Action réalisée avec succès**

Indique que l'action demandée a été réalisée avec succès, traduisant ainsi que le format de la requête est correct.

- **1 - Action non autorisée**

Indique que vous n'avez pas souscrit à une offre permettant d'utiliser les web services.

- **2 - Paramètre invalide**

Ce code d'erreur est retourné lorsqu'un attribut est invalide. Il est accompagné d'une erreur de type **param** qui fournit un complément d'informations sur l'attribut posant problème.

Code d'erreur	Description	Explication
33	Paramètre 'paymentSource' invalide dans le cas d'un abonnement	Origine de la transaction invalide pour un abonnement. Les valeurs possibles sont "EC", "MOTO", "CC" ou "OTHER".
34	Paramètre 'scheme' invalide dans le cas d'une création d'un alias	Type de carte invalide lors de la création d'un alias.
35	Paramètre 'phoneNumber' invalide	Numéro de téléphone de l'acheteur invalide.
36	Paramètre 'email' invalide	E-mail de l'acheteur invalide.
37	Paramètre 'zipCode' invalide	Code postal de l'acheteur invalide.
38	Paramètre 'cellPhoneNumber' invalide	Numéro de téléphone mobile de l'acheteur invalide.
50	Paramètre 'shopId' invalide	Identifiant boutique mal renseigné.
51	Paramètre 'submissionDate' invalide	Date et heure UTC de la transaction non renseignées.
66	Paramètre 'contractNumber' invalide	Numéro de contract commerçant invalide.
82	Paramètre 'initialAmount' invalide	Montant initial de l'abonnement invalide (inférieur à 0).
83	Paramètre 'initialAmountNumber' invalide	Nombre d'échéances auxquelles il faut appliquer le montant initialAmount . Cet attribut est obligatoire si initialAmount est valorisé.
84	Paramètre 'effectDate' invalide	Date d'effet de l'abonnement invalide. La date ne peut pas être dans le passé.
85	Paramètre 'commission' invalide	Paramètre non renseigné et obligatoire pour le Boletto au Brésil.
90	Paramètre 'enrolled' invalide	Statut de l'enrôlement du porteur est invalide
92	Paramètre 'eci' invalide	Indicateur de commerce électronique invalide
93	Paramètre 'xid' invalide	Numéro de transaction 3DS invalide
94	Paramètre 'cavv' invalide	Certificat de l'ACS invalide
95	Paramètre 'cavvAlgorithm' invalide	Algorithme de vérification de l'authentification du porteur (cavv) invalide
96	Paramètre 'brand' invalide	Réseau de la carte invalide
101	Paramètre 'paymentOptionCode' invalide	Code de l'option invalide.
102	Paramètre 'paymentOptionCode (invalid date)' invalide	Date de validité de l'option de paiement invalide.
103	Paramètre 'amount/optionCode (inconsistency)' invalide	Code de l'option de paiement incohérent vis-à-vis du montant.
104	Paramètre 'optionCode (not found)' invalide	Code de l'option de paiement inconnu.

Exemple : Adresse e-mail manquante pour une opération où elle doit être renseignée.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">222e970d-9c8b-466f-b672-7d830af18a8c</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T09:41:11Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">HW4+kJD1ErT3g2z5KnUEjFxBPsg9NTjR6Q0sXjfsKvk=</authToken>
```

```

</env:Header>
<soap:Body>
<ns2:createTokenResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
  <createTokenResult>
    <requestId>222e970d-9c8b-466f-b672-7d830af18a8c</requestId>
    <commonResponse>
      <responseCode>2</responseCode>
      <responseCodeDetail>Error param 36: email</responseCodeDetail>
    </commonResponse>
  </createTokenResult>
</ns2:createTokenResponse>
</soap:Body>
</soap:Envelope>

```

• 3 - Requête non traitée

Indique que la requête n'a pu être traitée.

Dans la réponse un complément d'informations est retourné pour plus de détails.

Exemple :

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope"/>
  <soap:Body>
    <ns2:updatePaymentDetailsResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <updatePaymentDetailsResult>
        <requestId>024a5593-8537-46cf-afc3-fabb7c76bc97</requestId>
        <commonResponse>
          <responseCode>3</responseCode>
          <responseCodeDetail>Bad Request[Détails du non traitement de la requête]</responseCodeDetail>
        </commonResponse>
        <paymentResponse/>
        <orderResponse/>
        <cardResponse/>
        <authorizationResponse/>
        <captureResponse/>
        <customerResponse/>
        <markResponse/>
        <threeDSResponse/>
        <extraResponse/>
        <fraudManagementResponse/>
        <shoppingCartResponse/>
      </updatePaymentDetailsResult>
    </ns2:updatePaymentDetailsResponse>
  </soap:Body>
</soap:Envelope>

```

• 25 - Numéro de contrat inconnu

Indique un défaut au niveau du contrat commerçant.

Plusieurs cas possibles :

- La valeur transmise dans la requête ne correspond à aucun contrat enregistré sur la boutique (**shopId**),
- Il n'y a pas de contrat associé à la boutique,
- Le contrat spécifié est clôturé,
- Aucun contrat ne correspond au type de contrat nécessaire pour effectuer le paiement. C'est le cas si vous ne possédez pas de contrat acceptant le paiement manuel et que **paymentSource** est valorisé à **MOTO**, **CC** ou **OTHER** dans votre requête.

• 35 - Alias non créé

Indique que l'alias n'a pas été créé.

Le motif du refus est donné dans l'attribut **result** de l'objet **authorizationResponse**.

Reportez-vous au chapitre **Gérer les codes de retour d'une demande d'autorisation** pour plus de détails.

Exemple :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">8fbf0b7a-c5bd-419d-b14d-23bf557c6139</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T09:47:58Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">DjY7Jr+a
+7jzqD4FtYj7MflmVc8o/8QDPZkjdFSNk/k=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:createTokenResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createTokenResult>
        <requestId>8fbf0b7a-c5bd-419d-b14d-23bf557c6139</requestId>
        <commonResponse>
          <responseCode>35</responseCode>
          <responseCodeDetail>PaymentToken creation declined</responseCodeDetail>
        </commonResponse>
        <authorizationResponse>
          <result>51</result>
        </authorizationResponse>
      </createTokenResult>
    </ns2:createTokenResponse>
  </soap:Body>
</soap:Envelope>
```

- **36 - Attribut 'paymentToken' purgé**

Au delà de 15 mois de non utilisation d'un alias, les données du titulaire du moyen de paiement sont purgées (référentiel de sécurité PCI DSS - sécurité et protection des données bancaires).

- **40 - Montant non autorisé**

Indique que le montant de la requête de création ou de remboursement de paiement n'est pas conforme aux montants minimum / maximum définis sur le contrat commerçant.

Vous pouvez vous rapprocher du service clients pour connaître les détails de votre contrat.

- **99 - Erreur technique**

Ce code d'erreur est retourné en cas d'une erreur technique interne.

Pour plus de renseignements, veuillez contacter le support technique.

Exemple : Adresse e-mail manquante pour une opération où elle doit être renseignée.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">ca3d38d5-0344-461c-9f52-192482e09bda</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T09:47:58Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">d3W/6dtIebGUpReqzrS40KHEImEway6ixrpn05pSGLY=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:createTokenResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createTokenResult>
        <requestId>ca3d38d5-0344-461c-9f52-192482e09bda</requestId>
        <commonResponse>
          <responseCode>99</responseCode>
        </commonResponse>
        <authorizationResponse>
          <result>96</result>
        </authorizationResponse>
      </createTokenResult>
    </ns2:createTokenResponse>
  </soap:Body>
</soap:Envelope>
```

3.3. Enchaîner des requêtes Web service (maintenir une même session HTTP)

L'architecture de la plateforme de paiement repose sur un ensemble de serveurs avec répartition de charge.

Pour assurer la continuité du processus, chaque requête associée à un même paiement dans un laps de temps très court doit être réalisée avec la même session HTTP.

Pour cela, à chaque requête, une session est créée coté serveur. L'ID de la session est renvoyé dans l'en-tête HTTP de la réponse. Il devra être retourné dans les requêtes suivantes afin que la requête soit traitée par le même serveur, évitant ainsi à votre requête d'être rejetée car la transaction ne serait pas encore disponible sur les autres serveurs.

Exemple d'application :

Vous souhaitez créer un paiement à remettre dans 30 jours en mode de validation manuelle.

Une fois le paiement accepté, vous décidez de changer la date de remise pour le lendemain et de valider la transaction.

Pour réaliser cette opération :

- Vous devez appeler le web service de création de paiement (**createPayment**).
- La plateforme vérifie la présence d'un ID de session dans l'en-tête HTTP de votre requête.

Comme rien n'a été précisé, une nouvelle session et un nouvel ID sont créés.

La plateforme de paiement procède ensuite au traitement de votre requête et envoie sa réponse en indiquant dans l'en-tête HTTP l'identifiant de session attribué ainsi que le nom du serveur ayant traité la requête :

Exemple d'en-tête de requête :

```
POST /vads-ws/v5 HTTP/1.1
Host: epaync.nc
Connection: Keep-Alive
User-Agent: PHP-SOAP/5.4.14
Content-Type: text/xml; charset=utf-8
SOAPAction: ""
Content-Length: 1922
```

Exemple d'en-tête de réponse :

```
HTTP/1.1 200 OK
Date: Tue, 24 Feb 2015 11:37:06 GMT
Server: Apache
Set-Cookie: JSESSIONID=6qeorHaVgOGr5avgh6lHnzEm.vadpayment01bdx;
Path=/vads-ws; Secure
Content-Length: 2711
Vary: Accept-Encoding,User-Agent Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/xml; charset=UTF-8
```

- Dans les en-têtes HTTP de la réponse, vous récupérez le cookie JSESSIONID.
- Vous initialisez le cookie JSESSIONID de votre en-tête HTTP.
- Vous appelez le web service de modification et de validation du paiement.

Exemple d'en-tête de requête pour maintenir la session :

```
POST /vads-ws/v5 HTTP/1.1
Host: epaync.nc
Connection: Keep-Alive
User-Agent: PHP-SOAP/5.4.14
Content-Type: text/xml; charset=utf-8 SOAPAction: ""
Content-Length: 5793
Cookie: JSESSIONID=6qeoRHaVgOGr5avgh6lHnzEm.vadpayment01bdx;
```

- La plateforme vérifie la présence d'un ID de session dans l'en-tête HTTP de votre requête.

La requête est ensuite envoyée au serveur ayant généré la session.

Si la session existe, elle est réutilisée, sinon une nouvelle session et un nouvel identifiant seront créés.

La plateforme de paiement procède au traitement de la requête et envoie sa réponse.

Remarque :

L'ID de session sera renvoyé dans l'en-tête HTTP de la réponse uniquement dans le cas où une nouvelle session a été créée.

Il est donc conseillé de tester systématiquement la présence du cookie dans l'en-tête HTTP de la réponse et d'utiliser l'ID de session présent avant d'enchaîner un autre appel (**getPaymentDetails** par exemple).

Exemple d'en-tête HTTP de réponse utilisant la même session :

```
HTTP/1.1 200 OK
Date: Thu, 26 Feb 2015 10:26:01 GMT
Access-Control-Allow-Origin: *
Content-Type: text/xml; charset=UTF-8
Content-Length: 2858
Vary: Accept-Encoding
Connection: close
```

Remarque : pas d'information sur l'identifiant de session.

Exemple d'en-tête HTTP de réponse avec une nouvelle session :

```
HTTP/1.1 200 OK
Date: Thu, 26 Feb 2015 10:31:39 GMT
Set-Cookie: JSESSIONID=W10zI16iiiDiZqGV309xDtLV.vadpayment01bdx; Path=/vads-ws; Secure
Access-Control-Allow-Origin: *
Content-Type: text/xml; charset=UTF-8
Content-Length: 2858
Vary: Accept-Encoding
Connection: close
```

Remarque : présence du Set-Cookie précisant l'identifiant de la nouvelle session.

3.4. Gérer les délais d'attente (Timeout)

Le traitement d'une requête web service s'articule autour d'un enchaînement d'évènements asynchrones comme :

- l'envoi de la requête via le réseau du site marchand,
- le transport des informations sur le réseau internet,
- le traitement du paiement par la plateforme,
- l'interrogation des serveurs bancaires, etc

Un incident peut survenir à chaque étape et augmenter le temps du traitement (et donc implicitement le temps d'attente pour l'acheteur).

La réponse à une requête peut être retardée pour de multiples raisons :

- Un temps de réponse long de la part de l'émetteur du porteur de la carte (cas des cartes étrangères, cas de période de forte charge comme les soldes, ...).
- Un temps de réponse long de la part de l'acquéreur lors de la transmission et de la réception de la demande d'autorisation.
- Un temps de réponse long du côté de votre application suite à une charge importante.
- Un temps de réponse long de la plateforme de paiement.
- Un problème de peering sur Internet pouvant entraîner des pertes de messages, etc...

Selon les délais d'attente paramétrés, vous pouvez ne pas recevoir de réponse alors que le traitement asynchrone continue à s'exécuter côté plateforme de paiement.

Un temps de traitement long ne doit pas être considéré comme un paiement refusé.

Pour cette raison, vous devez configurer votre code pour gérer les problèmes potentiels pouvant survenir avec la connexion à l'API SOAP.

Conseils

Le temps moyen de traitement d'une demande de paiement par la plateforme est inférieur à 5 secondes.

Vous devez définir un délai d'attente de 20 à 30 secondes côté acheteur.

Pendant ce temps, vous pouvez :

- Informer l'acheteur que son paiement est en cours de traitement. Pendant ce temps là, vous pouvez consulter le statut de la transaction sur la plateforme de paiement et revenir vers lui une fois le résultat final affiché.
- Informer l'acheteur que son paiement est refusé en vous assurant au préalable que vous ne devez pas valider manuellement le paiement sur la plateforme de paiement.

3.5. Spécifier les types de données

Les messages échangés entre le marchand et la plateforme de paiement sont élaborés en respectant une syntaxe précise.

Vous trouverez dans le tableau ci-dessous la description des annotations utilisées.

Annotation	Description
a	Caractères alphabétiques (de A à Z et de a à z)
n	Caractères numériques (de 0 à 9)
s	Caractères spéciaux
an	Caractères alphanumériques
ans	Caractères alphanumériques et spéciaux
3	Longueur fixe de 3 caractères
..12	Longueur variable jusqu'à 12 caractères

Tableau 1 : Description des annotations utilisées

Les données véhiculées dans les messages peuvent être de différents types:

Types de données	Description
boolean	Un booléen ne peut avoir que deux réponses: true ou false . Une réponse true peut aussi être yes ou 1 . Une réponse false peut aussi être no ou 0 .
dateTime	Représente un instant, généralement exprimé sous la forme d'une date ou d'une heure. Contient une année, un mois, un jour, une heure, des minutes, des secondes et des millisecondes. La valeur est exprimée en temps universel coordonné (UTC) et ISO 8601 - W3C. Contrairement à l'heure locale, une date et une heure donnée en UTC est la même partout en même temps. <i>Exemple</i> : 2016-07-16T19:20Z
int	Représente un nombre entier (integer) c'est-à-dire sans décimale.
long	Représente un nombre entier (integer) codé sur 64 bits. Ce type de données est utilisé lorsque le type de données int n'est pas assez grand (pour spécifier le montant d'une transaction par exemple).
string	Peut contenir des caractères, des sauts de lignes, des retours chariots et des tabulations.

Tableau 2 : Description des données véhiculées dans les messages

4. S'IDENTIFIER LORS DES ÉCHANGES

L'identification s'effectue au moyen de l'en-tête SOAP HEADER de la requête.

Pour identifier le site marchand lors des échanges avec la plateforme de paiement, les éléments suivants sont requis :

- l'identifiant de la boutique (**shopId**),
- le certificat,

Pour récupérer ces deux valeurs :

1. Connectez-vous à : <https://epaync.nc/vads-merchant/>
2. Cliquez sur **Paramétrage > Boutique**.
3. Sélectionnez l'onglet **Certificats**.

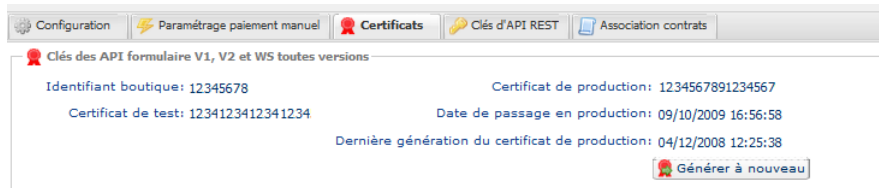


Image 1 : Visualiser l'identifiant de la boutique et le certificat

Remarque : nous vous conseillons de stocker ces informations dans un fichier de configuration.

- un jeton d'authentification à transmettre dans l'en-tête SOAP (SOAP HEADER).

Le calcul du jeton d'authentification est basé sur l'algorithme de hachage HMAC_SHA256.

Il est construit à partir de la fonction de hachage SHA-256 et utilisé comme du code HMAC (Hash-based Message Authentication Code).

Le hachage de sortie a une longueur de 256 bits.

Remarque : la fonction HMAC_SHA256 n'est disponible en PHP qu'à partir de la version PHP 5.1.2.

4.1. Procéder à l'authentification

Les étapes pour procéder à l'authentification sont les suivantes :

1. Le site marchand récolte les données lui permettant de construire l'en-tête SOAP HEADER.
Voir chapitre L'en-tête SOAP HEADER.
2. Le site marchand calcule la valeur du jeton d'authentification et l'ajoute dans l'en-tête SOAP HEADER.
Voir chapitre Construire l'en tête SOAP HEADER de la requête.
3. Le site marchand envoie la requête.
4. La plateforme de paiement reçoit la requête et analyse l'en-tête SOAP HEADER.
5. La plateforme de paiement calcule la valeur du jeton d'authentification **authToken**.
6. La plateforme de paiement compare la valeur du jeton d'authentification calculée avec celle transmise par le site marchand.
7. Si les valeurs diffèrent, la requête est rejetée et renvoie une exception SOAP Fault exception de type "**bad.authToken: Invalid authentication token**".
Sinon, la plateforme traite la requête.
8. La plateforme de paiement calcule la valeur du jeton d'authentification et l'ajoute dans le HEADER de la réponse.
9. La plateforme de paiement construit le message de réponse et l'envoie au site marchand.
10. Le site marchand réceptionne les données. Il calcule la valeur du jeton d'authentification en utilisant les valeurs contenues dans le HEADER de la réponse.
Il compare la valeur du jeton d'authentification calculée avec celle transmise dans le HEADER de la réponse.
*Remarque : le **requestId** transmis dans l'en-tête de la réponse sera identique à celui transmis dans la requête par le site marchand.*
11. Si les valeurs diffèrent, le marchand analyse l'origine de l'erreur (erreur, tentative de fraude...).
Sinon, le site marchand procède à l'analyse de la réponse.

4.2. Construire l'en-tête SOAP HEADER de la requête

Pour construire l'en-tête SOAP HEADER :

1. Ajoutez un nouvel en-tête nommé **shopId** dont la valeur sera l'identifiant de la boutique.

Sa valeur est disponible sur votre Back Office en sélectionnant le menu **Paramétrage > Boutique > onglet Certificats**.

2. Ajoutez un nouvel en-tête nommé **timestamp**.

Sa valeur spécifie la représentation numérique de la date et de l'heure de la requête, codée dans le format ISO 8601 - W3C et UTC.

Exemple de génération en PHP :

```
$timestamp = gmdate("Y-m-d\TH:i:s\Z");
```

Résultat : 2014-10-31T16:38:19Z

3. Ajoutez un nouvel en-tête nommé **mode**.

Sa valeur permet de définir le type de transaction. Elle peut être valorisée à **TEST** (pour une transaction de test) ou à **PRODUCTION** (pour une transaction réelle).

4. Ajoutez un nouvel en-tête nommé **requestId**.

L'attribut **requestId** est un UUID (identifiant universel unique). Sa valeur permet de calculer le jeton d'authentification.

L'attribut **requestId** doit être généré par le site marchand. Son format doit respecter la syntaxe xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxx (où M=1,2,3,4,5 et N=8,9,A,B).

- Exemple de génération en JAVA :

```
java.util.UUID.randomUUID().toString();
```

- Exemple de génération en PHP :

```
function gen_uuid() {  
    if (function_exists('random_bytes')) {  
        // PHP 7  
        $data = random_bytes(16);  
    } elseif (function_exists('openssl_random_pseudo_bytes')) {  
        // PHP 5.3, Open SSL required  
        $data = openssl_random_pseudo_bytes(16);  
    } else {  
        return sprintf(  
            '%04x%04x-%04x-%04x-%04x%04x%04x',  
            mt_rand(0, 0xffff),  
            mt_rand(0, 0xffff),  
            mt_rand(0, 0xffff),  
            mt_rand(0, 0xffff) | 0x4000,  
            mt_rand(0, 0x3fff) | 0x8000,  
            mt_rand(0, 0xffff),  
            mt_rand(0, 0xffff),  
            mt_rand(0, 0xffff)  
        );  
    }  
  
    $data[6] = chr(ord($data[6]) & 0x0f | 0x40); // set version to 100  
    $data[8] = chr(ord($data[8]) & 0x3f | 0x80); // set bits 6 & 7 to 10  
  
    return vsprintf('%s%s-%s-%s-%s%s%s', str_split(bin2hex($data), 4));  
}
```

- Exemple de génération en ASP.NET :

```
System.Guid.NewGuid().ToString();
```

5. Ajoutez un nouvel en-tête nommé **authToken**.

Pour obtenir sa valeur :

- a. Concaténez les attributs **requestId** et **timestamp** sans séparateur.

Exemple de résultat de concaténation avec :

- requestId = 04967dae-af01-43ff-a7d8-f3f228b9b1c2
- timestamp = 2014-10-31T16:38:19Z

```
"04967dae-af01-43ff-a7d8-f3f228b9b1c22014-10-31T16:38:19Z"
```

- b. Appliquez l'algorithme HMAC_SHA256 sur la chaîne obtenue en utilisant la valeur du certificat de test ou de production (en fonction de la valeur de **mode**) comme clé partagée.

- c. Encodez le résultat en Base64.

- Exemple d'implémentation en JAVA 7 :

```
public String hmacsha256(String stringToSign , String key ){
    try {
        byte[] bytes = encode256 ( key .getBytes( "UTF-8" ), stringToSign .getBytes( "UTF-8" ));
        return Base64.encodeBase64String( bytes );
    } catch (Exception e ){
        throw new RuntimeException( e );
    }
}
private static byte[] encode256(byte[]keyBytes, byte[] text ) throws
NoSuchAlgorithmException, InvalidKeyException {
    Mac hmacShal ;
    try {
        hmacShal = Mac.getInstance ( "HmacSHA256" );
    } catch (NoSuchAlgorithmException nsae ){
        hmacShal = Mac.getInstance ( "HMAC-SHA-256" );
    }
    SecretKeySpec macKey = new SecretKeySpec( keyBytes, "RAW" );
    hmacShal.init( macKey );
    return hmacShal .doFinal( text );
}
```

Remarque : l'implémentation est basée sur la classe Mac du package javax.crypto.

Exemple d'appel en JAVA:

```
hmacsha256("04967dae-af01-43ff-a7d8-f3f228b9b1c22014-10-31T16:38:19Z", "1234567887654321")
```

- Exemple d'implémentation en PHP :

```
// $data est la concaténation des attributs requestId et timestamp
// $shopKey est la valeur du certificat

<?php $authToken = base64_encode(hash_hmac('sha256',$data, $shopKey, true));?>
```

- Exemple d'implémentation en ASP.NET :

```
private static byte[] StringEncode (string text)
{
    var encoding = new ASCIIEncoding();
    return encoding.GetBytes(text);
}
private static string HashEncode(byte[] hash)
{
    return System.Convert.ToBase64String(hash)
}
private static byte[] HashHMAC (byte[] key, byte[] message)
{
    var hash = new HMACSHA256(key);
    return hash.ComputeHash(message);
}
public String HmacSha256(String stringToSign, String key)
{
    return HashEncode(HashHMAC(StringEncode(key), StringEncode(stringToSign)));
}
```

Exemple d'appel en ASP.NET :

```
HmacSha256("RF5GJlpZwcra2N7Ie/04Xn/SxFVnqy/6lYr6F6lFrHo=", "1234567887654321")
```

Résultat :

```
<soapHeader:authToken>NxofUSSTqmMjwaDzTXyCN4nNpMOVJKb5UxHdS9TBuTg=</soapHeader:authToken>
```

Exemple de code en PHP pour construire l'en-tête SOAP HEADER

Ci-dessous un exemple PHP pour vous aider à construire l'en-tête SOAP :

```
//Génération du header

$client = new soapClient("https://epaync.nc/vads-ws/v5?wsdl", $options = array(
    'trace'=>1,
    'exceptions'=> 0,
    'encoding' => 'UTF-8',
    'soapaction' => '')
);

//Calcul des valeurs à transmettre dans l'en-tête

$ns = 'http://v5.ws.vads.lyra.com/Header/';
$shopId = "12345678";
$requestId = gen_uuid();
$timestamp = gmdate("Y-m-d\TH:i:s\Z");
$mode = "TEST";
$authToken = base64_encode(hash_hmac('sha256',$requestId.$timestamp, $key, true));

//Création des en-têtes shopId, requestId, timestamp, mode et authToken

$headerShopId = new SOAPHeader($ns, 'shopId', $shopId);
$headerRequestId = new SOAPHeader($ns, 'requestId', $requestId);
$headerTimestamp = new SOAPHeader($ns, 'timestamp', $timestamp);
$headerMode = new SOAPHeader($ns, 'mode', $mode);
$headerAuthToken = new SOAPHeader($ns, 'authToken', $authToken);

//Ajout des en-têtes dans le SOAP Header

$headers = array(
    $headerShopId,
    $headerRequestId,
    $headerTimestamp,
    $headerMode,
    $headerAuthToken
);

$client->__setSoapHeaders($headers);
```

Des exemples dans des langages de programmation différents sont disponibles sur Internet.

- En JAVA

<http://www.mkyong.com/webservices/jax-ws/jax-ws-soap-handler-in-client-side/>

- En Visual Basic .NET

[https://msdn.microsoft.com/en-fr/library/vstudio/whew6x7f\(v=vs.100\).aspx](https://msdn.microsoft.com/en-fr/library/vstudio/whew6x7f(v=vs.100).aspx)

<http://forums.asp.net/t/1137408.aspx?Adding+information+to+the+SOAP+Header>

4.3. Vérifier l'en-tête SOAP dans la réponse

Pour vous assurer que la réponse provient de la plateforme de paiement vous devez vérifier la valeur du jeton d'authentification reçu.

Pour cela, recalculez le jeton d'authentification :

1. Récupérez les valeurs de **shopId**, **timestamp**, **requestId**, **mode** et **authToken** dans l'en-tête SOAP de la réponse.
2. Concaténez les attributs **timestamp** et **requestId** sans séparateur.
Attention, l'ordre est inversé par rapport à la requête.

Exemple de résultat de concaténation avec :

- timestamp = 2014-10-31T16:38:19Z
- requestId = 04967dae-af01-43ff-a7d8-f3f228b9b1c2

```
"2014-10-31T16:38:19Z04967dae-af01-43ff-a7d8-f3f228b9b1c2"
```

3. Appliquez l'algorithme HMAC_SHA256 sur la chaîne obtenue en utilisant la valeur du certificat de test ou de production (en fonction de la valeur de **mode**) comme clé partagée.
4. Encodez le résultat en Base64.
5. Comparez la valeur de **authToken** présent dans l'en-tête SOAP HEADER avec celle calculée.

Résultat :

Si les valeurs diffèrent, le marchand analyse l'origine de l'erreur (erreur dans le calcul, fraude...).

*Remarque : le **requestId** transmis dans l'en-tête de la réponse sera identique à celui transmis dans la requête par le site marchand.*

Exemple de code PHP pour récupérer les en-têtes SOAP dans la réponse

Ci-dessous un exemple pour vous aider à récupérer les en-têtes SOAP HEADER dans la réponse :

```
//Récupération du SOAP Header de la réponse afin de stocker
// les en-têtes dans un tableau (ici $responseHeader)

$dom = new DOMDocument;
$dom->loadXML($client->__getLastResponse(), LIBXML_NOWARNING);
$path = new DOMXPath($dom);
$headers = $path->query('//*[local-name()="Header"]/*');

$responseHeader = array();
foreach($headers as $headerItem) {

    $responseHeader[$headerItem->nodeName] = $headerItem->nodeValue;
}
```

5. GÉNÉRER UN UUID - RÉTROCOMPATIBILITÉ

Un **uuid** (Universally Unique IDentifier) est un identifiant unique qui permet, dans cette version des Web Services, d'identifier de façon absolument sûre une transaction.

Cependant, l'**uuid** n'était pas généré dans les versions précédentes.

Pour identifier et interroger une transaction, les attributs **transactionId**, **sequenceNumber** et **creationDate** étaient utilisés.

Ils permettaient de cibler une transaction spécifique à partir de :

- son identifiant unique sur une journée,
- la date et l'heure de la requête à laquelle la transaction était créée,
- le numéro de séquence.

Dans cette version, seul l'attribut **uuid** (référence unique de la transaction) est nécessaire.

Il est utilisé pour remplacer les anciens attributs et simplifier les requêtes.

Il est généré par la plateforme de paiement suite à la création d'une transaction de paiement. Cet identifiant unique offre une garantie d'unicité.

La valeur de cet attribut est retournée dans l'objet **paymentResponse** de l'opération **createPayment**.

Ainsi, pour toute requête impliquant une transaction spécifique, l'attribut **uuid** sera demandé au sein de l'objet **queryRequest**.

Cependant, pour des raisons de retrocompatibilité, cette version des Web Services permet de récupérer l'**uuid** d'une transaction (référence unique de la transaction) à partir de son ancienne identification.

Pour cela, l'opération **getPaymentUuid** est mise à disposition.

5.1. Requête à envoyer

La requête **getPaymentUuid** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **getPaymentUuid** prend en entrée un objet de type **getPaymentUuid**.

Le type **getPaymentUuid** est composé du paramètre suivant :

Objet	Format	Requis
legacyTransactionKeyRequest	legacyTransactionKeyRequest	✓

legacyTransactionKeyRequest

L'objet **legacyTransactionKeyRequest** permet de traduire les anciens attributs **transactionId**, **sequenceNumber** et **transmissionDate** en **transactionUuid** (référence unique de transaction).

Les attributs à valoriser pour obtenir la valeur de **transactionUuid** sont les suivants :

legacyTransactionKeyRequest		
Attribut	Requis	Format
transactionId Identifiant de la transaction à rechercher.	✓	string
sequenceNumber Numéro de séquence de la transaction à rechercher. Vaut "1" pour un paiement unitaire. Prend la valeur du numéro d'échéance dans le cas d'un paiement en plusieurs fois créé à partir du formulaire de paiement.	✓	n..3
creationDate Date de création de la transaction au format ISO 8601 définit par W3C. Si aucune date est renseignée, la date du jour sera valorisée par défaut. <i>Exemple : 2016-07-16T19:20:00Z</i>	✓	dateTime ans..40

Tableau 3 : Objet legacyTransactionKeyRequest

5.2. Réponse en retour

La réponse à l'opération **getPaymentUuid** est constituée d'un HEADER et d'un BODY de type **getPaymentUuidResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Il contient les informations relatives à l'authentification (voir chapitre L'en tête SOAP HEADER).

- **BODY**

La structure du message **getPaymentUuidResponse** est la suivante :

Nom	Type
legacyTransactionKeyResult	legacyTransactionKeyResult

La structure du message **legacyTransactionKeyResult** est la suivante :

Objet	Type
commonResponse	commonResponse
paymentResponse	paymentResponse

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 4 : Objet commonResponse

paymentResponse

L'objet **paymentResponse** renvoie l'attribut **transactionUuid**.

paymentResponse	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32

Tableau 5 : Objet paymentResponse

6. RÉALISER DES OPÉRATIONS COURANTES SUR LES TRANSACTIONS

L'option "Paiement par web services" permet de réaliser un certain nombre d'opérations courantes telles que :

Nom de l'opération web service	Description
updatePayment	Modifier une transaction de paiement
updatePaymentDetails	Modifier les informations d'une transaction (panier)
cancelPayment	Annuler une transaction de paiement
findPayments	Rechercher des transactions de paiements
refundPayment	Rembourser un acheteur
duplicatePayment	Dupliquer une transaction de paiement
validatePayment	Valider une transaction de paiement
capturePayment	Remiser une transaction de paiement
getPaymentDetails	Obtenir les détails d'une transaction de paiement
verifyThreeDSEnrollment	Vérifier l'authentification 3D Secure de la carte de l'acheteur
checkThreeDSAuthentication	Vérifier le statut de l'authentification 3D Secure

Tableau 6 : Opérations disponibles avec l'option "Paiement par web services"

L'option "Accès à l'acquisition sur site marchand via web-service" permet la création des transactions via l'opération :

Nom de l'opération web service	Description
createPayment	Initialiser une transaction de paiement

Tableau 7 : Opération disponible avec l'option "Accès à l'acquisition sur site marchand via web-service"

Des exemples de codage sont proposés en annexe de ce document.

6.1. Créer une transaction de paiement 'createPayment'

L'utilisation de cette fonction requiert la souscription à l'option "Accès à l'acquisition sur site marchand via web-service".

L'appel à l'opération **createPayment** permet d'initialiser une transaction de paiement.

Selon la valorisation des attributs dans la requête, il est possible de réaliser :

- des paiements comptants immédiats,
- des paiements comptants différés,
- des paiements avec ou sans authentification 3D Secure,
- des paiements en utilisant un alias déjà existant.

L'opération **createPayment** est structurée de la manière suivante :

Opération	Entrée	Sortie
createPayment	createPayment	createPaymentResponse

Le partage d'identifiants

Il est possible de partager des identifiants (alias) entre plusieurs entités juridiques.

Les identifiants partagés entre plusieurs entités juridiques doivent être uniques et doivent être impérativement générés par la plateforme de paiement (en d'autres termes l'attribut **paymentToken** de l'objet **cardRequest** ne doit pas être renseigné).

Cependant, cette fonctionnalité est soumise à des conditions particulières. Veuillez contacter l'interlocuteur de votre plateforme de paiement pour en prendre connaissance.

Requête à envoyer

La requête **createPayment** est constituée d'un HEADER et d'un BODY.

• HEADER

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre Construire l'en-tête SOAP HEADER).

• BODY

L'opération **createPayment** prend en entrée un objet de type **createPayment**.

Le type **createPayment** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	✓
threeDSRequest	threeDSRequest	
paymentRequest	paymentRequest	✓
orderRequest	orderRequest	
cardRequest	cardRequest	✓
customerRequest	customerRequest	
techRequest	techRequest	
shoppingCartRequest	shoppingCartRequest	

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Plusieurs attributs peuvent être spécifiés dans la requête. Cependant, un seul doit obligatoirement être envoyé.

commonRequest		
Attribut	Requis	Format
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• EC pour le commerce électronique. Paramètre utilisé par défaut si aucune valeur est renseignée ou si différente des valeurs possibles.• MOTO pour une commande par courrier ou téléphone.• CC pour un centre d'appel.• OTHER pour un autre canal de vente. Seule la valeur EC permet de créer une transaction avec 3D Secure. Les autres valeurs ne doivent être utilisées que pour de la vente à distance, pour laquelle le 3D Secure n'est pas applicable.		string
submissionDate Date et heure UTC de la transaction exprimée au format ISO 8601 défini par W3C. <i>Exemple : 2016-07-16T19:20:00Z</i> Si la valeur de cet attribut est trop éloignée de l'heure actuelle, la requête sera rejetée (code d'erreur 13).	✓	dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé. Si ce champ est renseigné, veuillez à utiliser le bon contrat en fonction du réseau de la carte. Par exemple, un contrat CB ne peut être utilisé pour une transaction AMEX.		string

Tableau 8 : Objet commonRequest

L'attribut **comment** n'est pas aujourd'hui pris en considération pour cette opération.

threeDSRequest

L'objet **threeDSRequest** permet de :

- déterminer si un paiement est réalisé avec ou sans authentification 3D Secure,
- transmettre des informations liées à 3D Secure.

Si aucune valeur n'est renseignée pour cet objet, un paiement sans authentification 3D Secure sera effectué par défaut.

Les objets et attributs de l'objet **threeDSRequest** diffèrent selon le type de paiement.

• Paiement sans 3D Secure

Un paiement sans authentification 3D Secure est une opération de débit dans laquelle l'authentification du porteur de la carte n'est pas effectuée.

Un seul attribut doit obligatoirement être envoyé :

threeDSRequest	
Attribut	Format
mode Sélection du mode d'authentification 3D Secure. DISABLED Paiement sans authentification 3D Secure. Paramètre utilisé par défaut si aucune valeur est renseignée.	string

Remarque : tous les autres attributs ne sont pas pris en considération pour créer un paiement sans 3D Secure.

• Paiement avec 3D Secure

Le paiement avec authentification 3D Secure nécessite deux appels à l'opération **createPayment**.

- Un premier appel** pour vérifier l'enrôlement de la carte et récupérer les informations nécessaires pour rediriger l'acheteur vers l'ACS.

Un seul attribut doit obligatoirement être envoyé :

threeDSRequest	
Attribut	Format
mode Sélection du mode d'authentification 3D Secure. ENABLED_CREATE Permet de vérifier l'enrôlement de la carte à 3D Secure avant d'effectuer le paiement.	string

Remarque : tous les autres attributs ne sont pas pris en considération pour créer un paiement avec 3D Secure.

- Un deuxième appel** pour analyser le retour du 3D Secure et finaliser la transaction.

Trois attributs doivent obligatoirement être envoyés :

threeDSRequest	
Attribut	Format
mode Sélection du mode d'authentification 3D Secure. ENABLED_FINALIZE Permet de finaliser un paiement 3D Secure.	string
requestId Avec le mode ENABLED_FINALIZE , ce champ doit contenir la valeur retournée dans l'attribut threeDSRequestId de l'objet threeDsResponse dans l'opération createPayment avec le mode ENABLED_CREATE .	string
pares	string

threeDSRequest	
Attribut	Format
Message PaRes (Payer Authentication Response) renvoyé par l'ACS.	

Remarque : tous les autres attributs ne sont pas pris en considération pour créer un paiement avec 3D Secure.

- **Paiement avec authentification 3D Secure réalisé par le MPI du marchand**

Plusieurs attributs doivent obligatoirement être envoyés pour ce type de paiement.

Attention : une valeur spécifique pour un attribut peut impliquer la valorisation d'un autre attribut !

threeDSRequest																					
Attribut					Requis	Format															
mode Sélection du mode d'authentification 3D Secure. MERCHANT_3DS Permet de faire un paiement 3DS avec le MPI du marchand.					✓	string															
brand Réseau de la carte.					✓	string															
enrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.					✓	string															
status Statut de l'authentification du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut authentifié 3 DS.• N pour une erreur d'authentification.• U pour une authentification impossible.• A pour un essai d'authentification.					✓ Si enrolled est valorisé à Y	string															
eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont : <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>05</td><td>06</td><td>07</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>						status = Y	status = A	status = U	status = N	VISA - AMEX	05	06	07	-	MasterCard	02	01	-	-	✓ Si status est valorisé à Y ou A	string
	status = Y	status = A	status = U	status = N																	
VISA - AMEX	05	06	07	-																	
MasterCard	02	01	-	-																	
xid Numéro de transaction 3DS.					✓ Si status est valorisé à Y	string															
cavv Certificat de l'ACS.					✓ Si status est valorisé à Y ou A	string															
algorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont : <ul style="list-style-type: none">• 0 pour HMAC.• 1 pour CVV.• 2 pour CVV_ATN.• 3 pour Mastercard SPA.					✓ Si status est valorisé à Y ou A	string															

Tableau 9 : Objet threeDSRequest

paymentRequest

L'objet **paymentRequest** permet de transmettre des informations liées au paiement.

Il possède les attributs suivants :

paymentRequest		
Attribut	Requis	Format
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none">• <u>Soit cet identifiant est généré par la plateforme.</u> Dans ce cas, ce paramètre ne doit pas être renseigné.• <u>Soit cet identifiant est généré par le site marchand.</u> Dans ce cas, ce paramètre doit être renseigné avec la valeur de l'identifiant souhaité. Attention, il incombe au site marchand de s'assurer de l'unicité des identifiants. Toute demande d'enregistrement contenant un identifiant déjà existant, sera rejetée, et retournera un code d'erreur 12. Remarque : cet attribut ne peut être envoyé à vide.		an..6
retryUuid Permet de spécifier l'identifiant unique de la transaction afin de réitérer la demande du paiement refusée. Pour effectuer ce rejeu, veuillez à récupérer la valeur de la référence unique de la transaction refusée véhiculée par l'attribut transactionUuid de l'objet paymentResponse .		string
amount Montant de la transaction dans sa plus petite unité monétaire . <u>Remarque</u> : <ul style="list-style-type: none">• Ne doit pas être envoyé à vide ou être à 0.• Ne doit pas être supérieur au montant initial (cas du remboursement).	✓	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	✓	n3
expectedCaptureDate Date de remise demandée exprimée au format ISO 8601 défini par W3C. <i>Exemple</i> : 2016-07-16T19:20:00Z. Ce paramètre est utilisé pour effectuer un paiement différé. Si le nombre de jours entre la date de remise demandée et la date actuelle est supérieur à la durée de validité de l'autorisation , une autorisation de 100 XPF sera réalisée le jour de la transaction. Ceci afin de vérifier la validité de la carte. L'autorisation pour le montant total sera effectuée : <ul style="list-style-type: none">• fonctionnement par défaut : le jour de la date de remise en banque souhaitée,• fonctionnement avec autorisation anticipée : selon le moyen de paiement sélectionné, à J- le nombre de jours correspondant à la durée de validité d'une autorisation avant la date de remise en banque souhaitée. Si vous souhaitez être notifié du résultat de cette demande d'autorisation, vous devez configurer la règle de notification URL de notification sur autorisation par Batch dans le Back Office (Paramétrage > Règles de notifications). <u>Remarque</u> : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.		dateTime ans..40
manualValidation Permet de valider manuellement une transaction tant que la date de remise en banque souhaitée n'est pas dépassée. Pour cela, cet attribut doit être valorisé à 1 (validation manuelle). Valorisé à 0 , la validation sera automatique.		n1
paymentOptionCode . Permet de définir le code de l'option de paiement à utiliser.		
acquirerTransientData Permet de transmettre des informations utiles à l'acquéreur pour réaliser des contrôles anti-fraude: URL de la boutique, nom de la boutique, numéro de commande, frais de port, méthode de livraison, etc..		json

paymentRequest		
Attribut	Requis	Format
Exemple : <code>acquirerTransientData = {VISANET={ "field91": "10XXXXXXX", "field92": "Lyra Peru", "field93": "specific data" }}</code>		

Tableau 10 : Objet paymentRequest

orderRequest

L'objet **orderRequest** permet de transmettre des informations liées à la commande.

Il est composé de l'attribut suivant :

orderRequest		
Attribut	Requis	Format
orderId Référence de la commande.		string an..64
extInfo Champs personnalisables permettant d'ajouter des données supplémentaires (champ supplémentaire qui sera persisté dans la transaction et sera retourné dans la réponse). L'attribut extInfo est composé de sous objets : <ul style="list-style-type: none">• key : nom de la donnée. Son format est "string".• value : valeur de la donnée. Son format est "string". <u>Exemple</u> : <extInfo><key>keyData</key><value>valuedata</value></extInfo>		extInfo

Tableau 11 : Objet orderRequest

cardRequest

L'objet **cardRequest** permet de transmettre les informations sur la carte de paiement.

Selon le type de paiement (paiement avec saisie des données bancaires ou paiement par alias), un ou plusieurs attributs sont requis.

- Paiement avec saisie des données bancaires :

cardRequest		
Attribut	Requis	Format
number Numéro de la carte.	✓	string
scheme Type de carte choisi par l'acheteur. Exemple de valeurs possibles: AMEX, CB, MASTERCARD, VISA, VISA_ELECTRON, VPAY, MAESTRO, E-CARTEBLEUE, JCB. Pour plus de détails, consultez la Liste des types de carte supportés.	✓	string
expiryMonth Mois d'expiration de la carte, entre 1 et 12.	✓	n..2
expiryYear Année d'expiration de la carte sur 4 digits. <i>Exemple : 2016</i>	✓	n4
cardSecurityCode Cryptogramme visuel à 3 chiffres (ou 4 pour Amex). <u>Ce champ est obligatoire</u> lorsque la carte dispose d'un code cryptogramme visuel. Si le CVV n'est pas transmis, la banque émettrice refusera le paiement. En revanche, ce champ est facultatif lorsque l'origine de la transaction est valorisée à MOTO.		string
cardHolderBirthday Date de naissance du porteur au format YYYY-MM-DD.	Requis selon le moyen de paiement	dateTime ans..64
cardHolderName Nom du porteur. Ce champ correspond au nom du porteur de carte. Obligatoire pour DECIDIR et VISANET, acquéreurs d'Amérique en Sud.	Requis selon le moyen de paiement	string
proofOfIdType Type de pièce d'identité. Valeurs possibles : <ul style="list-style-type: none">• DNI (= Documento Nacional de Identidad) Obligatoire pour DECIDIR.	Requis selon le moyen de paiement	string
proofOfIdNumber Numéro de pièce d'identité officielle. Correspond au numéro de pièce d'identité du type de pièce sélectionné. <u>Exemple :</u> 2AZ265480. Le format dépend du type de pièce d'identité et est de 7 à 13 caractères, chiffres, lettres et/ou points. Obligatoire pour DECIDIR.	Requis selon le moyen de paiement	an..13
walletPayload Permet de transmettre une payload chiffrée lors d'un paiement par wallet (ApplePay, GooglePay, etc...)		json

Tableau 12 : Objet cardRequest

- Paiement utilisant un alias existant :

cardRequest		
Attribut	Requis	Format
paymentToken Identifiant unique (alias) associé à un moyen de paiement. <ul style="list-style-type: none">• Soit cet identifiant a été généré par la plateforme.• Soit cet identifiant a été généré par le site marchand.	✓	string ans..64

cardRequest		
Attribut	Requis	Format
cardSecurityCode Cryptogramme visuel à 3 chiffres (ou 4 pour Amex). Pour que le paiement soit garanti, le marchand doit demander à l'acheteur : <ul style="list-style-type: none"> la saisie du cryptogramme visuel une authentification 3D Secure 		string ans..64

Tableau 13 : Objet cardRequest

Remarque :

Pour qu'un paiement utilisant un alias existant (paiement en un clic) soit garanti, le marchand doit demander à l'acheteur :

- la saisie du cryptogramme visuel
- une authentification 3D Secure

customerRequest

L'objet **customerRequest** permet de transmettre des informations liées à la livraison, à la facturation et des données techniques liées à l'acheteur.

Cet objet doit obligatoirement être envoyé dans la requête.

Cependant, les sous-objets qui le composent (voir tableau ci-après) sont facultatifs pour cette opération.

Format	Sous-objet	Requis
billingDetails Données de facturation de l'acheteur.	billingDetailsRequest	
shippingDetails Données de livraison de l'acheteur.	shippingDetailsRequest	
extraDetails Données techniques liées à l'acheteur.	extraDetailsRequest	

Tableau 14 : Sous-objets de **customerRequest**

6.1.2.6.1. Objet billingDetails

billingDetails possède les attributs suivants :

billingDetails		
Attribut	Requis	Format
reference Référence de l'acheteur.		string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..		string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 		string (enum)
firstName Nom de l'acheteur.		string ans..128
lastName Prénom de l'acheteur.		string ans..128
phoneNumber Numéro de téléphone de l'acheteur.		string ans..32
email E-mail de l'acheteur.		string ans..150

billingDetails		
Attribut	Requis	Format
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>		string ans..5
address Adresse de l'acheteur.		string ans..255
address2 Complément d'adresse de livraison.		string ans..255
district Quartier de l'acheteur.		string ans..127
zipCode Code postal de l'acheteur.		string ans..64
city Ville de l'acheteur.		string ans..128
state Etat/Région de l'acheteur.		string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none"> • DE pour l'Allemagne • IT pour l'Italie • PT pour le Portugal • GB pour le Royaume-Uni • JA pour le Japon • RU pour la Russie • ES pour l'Espagne • NL pour les Pays-Bas • SE pour la Suède • FR pour la France • PL pour la Pologne • CN pour la Chine 		string - a2
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> • de pour l'allemand • it pour l'italien • pt pour le portugais • en pour l'anglais • ja pour le japonais • ru pour le russe • es pour l'espagnol • nl pour le néerlandais • sv pour le suédois • fr pour le français • pl pour le polonais • zh pour le chinois • tr pour le turc 		string - a2
cellPhoneNumber Numéro de téléphone mobile		string ans..32
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).		string ans..255

Tableau 15 : Objet billingDetails

6.1.2.6.2. Objet shippingDetails

shippingDetails possède les attributs suivants :

shippingDetails		
Attribut	Requis	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> • PRIVATE pour un particulier. 		string (enum)

shippingDetails		
Attribut	Requis	Format
<ul style="list-style-type: none"> COMPANY pour une entreprise. 		
firstName Nom de l'acheteur.		string ans..128
lastName Prénom de l'acheteur.		string ans..128
phoneNumber Numéro de téléphone de l'acheteur.		string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>		string an..5
address Adresse de livraison.		string ans..255
address2 Complément d'adresse de livraison.		string ans..255
district Quartier de livraison.		string ans..127
zipCode Code postal de livraison.		string ans..64
city Ville de livraison.		string ans..128
state Etat/Région de livraison.		string ans..128
country Pays de livraison.		string - a2
deliveryCompanyName Informations sur le transporteur.		string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> STANDARD pour une livraison standard. EXPRESS pour une livraison express. 		string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont : <ul style="list-style-type: none"> RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). ETICKET pour l'émission d'un billet électronique, téléchargement. 		string (enum)
legalName Raison sociale de la société.		string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).		string ans..255

Tableau 16 : Objet shippingDetails

6.1.2.6.3. Objet extraDetails

extraDetails possède les attributs suivants :

extraDetails		
Attribut	Requis	Format
ipAddress L'adresse IP de l'acheteur.		string ans40
fingerPrintId Identifiant unique de session. Spécifique au Brésil et à l'analyseur de fraude ClearSale. Codé sur 128 octets, peut être composé de majuscules ou de minuscules, chiffres ou tiret ([A-Z] [a-z], 0-9, _, -).		string ans128

Tableau 17 : Objet extraDetails

techRequest

L'objet **techRequest** permet de transmettre des informations techniques à propos du navigateur de l'acheteur.

Cet objet doit obligatoirement être envoyé dans la requête.

Cependant, ses attributs sont facultatifs.

techRequest		
Attribut	Requis	Format
browserUserAgent Header « User-Agent » du navigateur de l'acheteur (HTTP/1.1 - RFC. 2616).		string
browserAccept Header « Accept » du navigateur de l'acheteur (HTTP/1.1 - RFC. 2616).		string
integrationType Nom et /ou version de la solution e-commerce utilisée.		string

Tableau 18 : Objet techRequest

shoppingCartRequest

L'objet **shoppingCartRequest** permet de transmettre le contenu du panier.

shoppingCartRequest																																				
Attribut	Requis	Format																																		
insuranceAmount Montant de l'assurance		n..3																																		
shippingAmount Frais d'expédition.		n..3																																		
taxAmount Montant des taxes.		n..3																																		
cartItemInfo Champs personnalisables permettant d'ajouter les éléments du panier. L'attribut cartItemInfo est composé de sous objets : <ul style="list-style-type: none">productLabel : nom du produit. Son format est "string".productType : type de produit. Son format est "string (enum)". <table><tr><th>Valeur</th><th>Description</th></tr><tr><td>FOOD_AND_GROCERY</td><td>Produits alimentaires et d'épicerie</td></tr><tr><td>AUTOMOTIVE</td><td>Automobile / Moto</td></tr><tr><td>ENTERTAINMENT</td><td>Divertissement / Culture</td></tr><tr><td>HOME_AND_GARDEN</td><td>Maison et jardin</td></tr><tr><td>HOME_APPLIANCE</td><td>Equipement de la maison</td></tr><tr><td>AUCTION_AND_GROUP_BUYING</td><td>Ventes aux enchères et achats groupés</td></tr><tr><td>FLOWERS_AND_GIFTS </td><td>Fleurs et cadeaux</td></tr><tr><td>COMPUTER_AND_SOFTWARE</td><td>Ordinateurs et logiciels</td></tr><tr><td>HEALTH_AND_BEAUTY</td><td>Santé et beauté</td></tr><tr><td>SERVICE_FOR_INDIVIDUAL</td><td>Services à la personne</td></tr><tr><td>SERVICE_FOR_BUSINESS</td><td>Services aux entreprises</td></tr><tr><td>SPORTS</td><td>Sports</td></tr><tr><td>CLOTHING_AND_ACCESSORIES</td><td>Vêtements et accessoires</td></tr><tr><td>TRAVEL</td><td>Voyage</td></tr><tr><td>HOME_AUDIO_PHOTO_VIDEO</td><td>Son, image et vidéo</td></tr><tr><td>TELEPHONY</td><td>Téléphonie</td></tr></table> <p>Tableau 20 : Valeurs associées à productType</p> <ul style="list-style-type: none">productRef : référence produit. Son format est "string".productQty : quantité de produit. Son format est "integer".productAmount : montant en centimes du produit. Son format est "string".productVat : montant de la taxe sur le produit. Son format est "string". <p>Exemple :</p> <pre><cartItemInfo> <productLabel>CHIPS</productLabel> <productType>FOOD_AND_GROCERY</productType> <productRef>188545</productRef> <productQty>10</productQty> <productAmount>10000</productAmount> </cartItemInfo></pre>	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		cartItemInfo
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																																			

Tableau 19 : Objet shoppingCartRequest

Réponse en retour

La réponse à l'opération **createPayment** est constituée d'un HEADER et d'un BODY de type **createPaymentResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **createPaymentResponse** est la suivante :

Nom	Type
createPaymentResult	createPaymentResult

La structure du message **createPaymentResult** est la suivante :

Objet	Type
commonResponse	commonResponse
paymentResponse	paymentResponse
orderResponse	orderResponse
cardResponse	cardResponse
authorizationResponse	authorizationResponse
captureResponse	captureResponse
customerResponse	customerResponse
markResponse	markResponse
threeDSResponse	threeDSResponse
extraResponse	extraResponse
fraudManagementResponse	fraudManagementResponse
shoppingCartResponse	shoppingCartResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque :

La valeur 0, indicateur de succès de l'opération, ne signifie pas pour autant que la transaction est validée. Pour vérifier le statut de la transaction, il est nécessaire d'analyser l'attribut **transactionStatusLabel**.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">INITIAL En traitement. Ce statut est temporaire. Le statut définitif sera retourné aussitôt la synchronisation réalisée.NOT_CREATED La transaction n'est pas créée et n'est pas visible dans le Back Office.AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue.AUTHORISED_TO_VALIDATE A valider. La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la remise en banque. La transaction peut être validée tant que la date de remise n'est pas dépassée. Si cette date est dépassée, le paiement prend le statut Expiré (statut définitif).WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement.WAITING_AUTHORISATION_TO_VALIDATE A valider et à autoriser. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation sera déclenchée automatiquement à J-1 avant la date de remise en banque. Le paiement pourra être accepté ou refusé. La remise en banque est automatique.REFUSED Refusée. La transaction est refusée.	string
shopId Identifiant de la boutique.	n8
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">EC pour le commerce électronique.MOTO pour une commande par e-mail ou téléphone.CC pour un centre d'appel.OTHER pour un autre canal de vente.	string (enum)
submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40

commonResponse	Format
contractNumber Numéro de contrat commerçant utilisé.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 21 : Objet commonResponse

paymentResponse

L'objet **paymentResponse** détaille les informations sur la transaction.

paymentResponse	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none"> • Soit cet identifiant est généré par la plateforme. • Soit cet identifiant est généré par le site marchand. 	an..6
amount Montant de la transaction dans sa plus petite unité monétaire .	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	n3
effectiveAmount Montant de la transaction dans la devise réellement utilisée pour effectuer la remise en banque.	n..12
effectiveCurrency Devise réellement utilisée pour effectuer la remise en banque.	n3
expectedCaptureDate Date de remise en banque souhaitée. Remarque : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.	dateTime ans..40
operationType Type d'opération. Les valeurs possibles sont : <ul style="list-style-type: none"> • 0 pour une opération de débit. • 1 pour une opération de remboursement. 	n1
creationDate Date et heure de l'enregistrement de la transaction exprimée au format W3C.	dateTime ans..40
externalTransactionId Référence fournie par un tiers : numéro de transaction pour PayPal, Boletto, RRN pour Prism, etc...	string
liabilityShift Transfert de responsabilité. Les valeurs possibles sont : <ul style="list-style-type: none"> • YES lorsque le paiement est garanti. • NO lorsque le paiement n'est pas garanti. 	string
paymentType Type de paiement. Les valeurs possibles sont : <ul style="list-style-type: none"> • SINGLE Paie ment en une seule fois. • INSTALLMENT Paie ment en plusieurs fois. • SPLIT Paie ment effectué avec plusieurs moyens de paiement. • SUBSCRIPTION 	string (enum)

paymentResponse	Format
<p>Païement par alias ou lié à un abonnement.</p> <ul style="list-style-type: none"> RETRY Lors d'un paiement refusé, il est possible de réitérer la demande de paiement. Pour toute(s) réitération(s), le paiement est valorisé avec cette valeur. <p><i>Remarque :</i> Les valeurs INSTALLMENT et SPLIT peuvent être retournées uniquement si des paiements ont été créés via le formulaire de paiement.</p> 	
<p>sequenceNumber Numéro de séquence de la transaction. Sa valeur est fonction du contexte du paiement :</p> <ul style="list-style-type: none"> Est valorisé à "1" pour un paiement unitaire. Est valorisé à "2" lors du rejeu d'un paiement initialement refusé. Prend la valeur du numéro d'échéance dans le cas d'un paiement en plusieurs fois créé à partir du formulaire de paiement. 	n..3
<p>paymentError Complément d'information en cas d'erreur technique. Retourne un code d'erreur associé à l'erreur technique (voir chapitre Codes d'erreurs pouvant survenir durant le paiement).</p>	n..3
<p>nsu Apparaît lorsque isNsuRequested est valorisé à 1 dans la requête.</p>	string

Tableau 22 : Objet paymentResponse

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
<p>orderId Référence de la commande.</p>	string an..64
<p>extInfo Données personnalisées retournées en fonction des besoins. <i>Exemple</i> : extInfo.key, extInfo.value</p> <ul style="list-style-type: none"> key : nom de la donnée (son format est "string"). value : valeur de la donnée (son format est "string"). 	extInfo

Tableau 23 : Objet orderResponse

cardResponse

L'objet **cardResponse** détaille les informations du moyen de paiement utilisé.

cardResponse	Format
<p>number</p> <ul style="list-style-type: none"> Numéro de carte masqué. Contient les 6 premiers chiffres du numéro, suivi par "XXXXXX" et enfin les 4 derniers chiffres. IBAN et BIC utilisés pour le paiement, séparés par un « _ » dans le cas d'un paiement par prélèvement. 	string
<p>scheme Type de la carte.</p>	string
<p>brand Marque de la carte utilisée pour la transaction</p>	string
<p>country Code pays du pays d'émission de la carte (Code numérique ISO 3166).</p>	ISO 3166
<p>productCode Code produit de la carte.</p>	an..3
<p>bankCode Code banque de la banque émettrice.</p>	n..5
<p>expiryMonth Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).</p>	n..2
<p>expiryYear</p>	n4

cardResponse	Format
Année d'expiration sur 4 chiffres (ex : 2023).	

Tableau 24 : Objet cardResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
mode Spécifie de quelle manière est réalisée la demande d'autorisation. Deux valeurs possibles : <ul style="list-style-type: none">• MARK Une autorisation de a été réalisée afin de vérifier la validité de la carte. Ce cas se présente lorsque la date de remise dépasse la période de validité d'une autorisation .• FULL Autorisation pour le montant total demandé dans la requête.	string
amount Montant de l'autorisation dans la plus petite unité monétaire dans le cas où mode vaut FULL .	n..12
currency Code de la monnaie utilisée lors de la demande d'autorisation (suivant la norme ISO 4217) dans le cas où mode vaut FULL .	n3
date Date et heure de la demande d'autorisation dans le cas où mode vaut FULL .	dateTime ans..40
number Numéro de la demande d'autorisation dans le cas où mode vaut FULL .	n..2
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 25 : Objet authorizationResponse

captureResponse

L'objet **captureResponse** permet d'obtenir des informations à propos de la remise dans le cas où la transaction est remise.

captureResponse	Format
date Date et heure de remise.	dateTime ans..40
number Numéro de remise.	n3
reconciliationStatus Statut du rapprochement bancaire de la transaction.	n1
refundAmount Montant ayant déjà fait l'objet d'un remboursement dans sa plus petite unité monétaire.	n..12
refundCurrency Devise du montant ayant déjà fait l'objet d'un remboursement (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF) .	n3
chargeback Litige, impayé. Les valeurs possibles sont : <ul style="list-style-type: none">• 0 : transaction ne faisant pas l'objet d'un litige.• 1 : transaction faisant l'objet d'un litige.	

Tableau 26 : Objet captureResponse

customerResponse

L'objet **customerResponse** détaille de nombreuses informations à propos de l'acheteur.

La réponse se décompose en l'analyse de :

- **billingDetails**

Données de facturation de l'acheteur.

- **shippingDetails**

Données de livraison de l'acheteur. Les valeurs présentes dans cet objet seront identiques à celles de l'objet **billingDetails**.

- **extraDetails**

Données techniques liées à l'acheteur.

6.1.3.7.1. Objet billingDetails

billingDetails	
Attribut	Format
reference Référence de l'acheteur.	string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..	string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none">• PRIVATE pour un particulier.• COMPANY pour une entreprise.	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de l'acheteur.	string ans..255
district Quartier de l'acheteur.	string ans..127
zipCode Code postal de l'acheteur.	string ans..64
city Ville de l'acheteur.	string ans..128
state Etat/Région de l'acheteur.	string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none">• DE pour l'Allemagne• IT pour l'Italie• PT pour le Portugal• GB pour le Royaume-Uni• JA pour le Japon• RU pour la Russie	string - a2

billingDetails	
Attribut	Format
<ul style="list-style-type: none"> ES pour l'Espagne FR pour la France NL pour les Pays-Bas PL pour la Pologne SE pour la Suède CN pour la Chine 	
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> de pour l'allemand en pour l'anglais es pour l'espagnol fr pour le français tr pour le turc it pour l'italien ja pour le japonais nl pour le néerlandais pl pour le polonais pt pour le portugais ru pour le russe sv pour le suédois zh pour le chinois 	string - a2
cellPhoneNumber Numéro de téléphone mobile	string ans..32
legalName Raison sociale de la société.	string ans..128

Tableau 27 : Objet billingDetails

6.1.3.7.2. Objet shippingDetails

shippingDetails	
Attribut	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de livraison.	string ans..255
address2 Complément d'adresse de livraison.	string ans..255
district Quartier de livraison.	string ans..127
zipCode Code postal de livraison.	string ans..64
city Ville de livraison.	string ans..128
state Etat/Région de livraison.	string ans..128
country	string - a2

shippingDetails	
Attribut	Format
Pays de livraison.	
deliveryCompanyName Informations sur le transporteur.	string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> • STANDARD pour une livraison standard. • EXPRESS pour une livraison express. 	string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont : <ul style="list-style-type: none"> • RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. • RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). • RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. • PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). • ETICKET pour l'émission d'un billet électronique, téléchargement. 	string (enum)
legalName Raison sociale de la société.	string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).	string ans..255

Tableau 28 : Objet shippingDetails

6.1.3.7.3. Objet extraDetails

extraDetails	Format
ipAddress L'adresse IP de l'acheteur.	string ans40

Tableau 29 : Objet extraDetails

markResponse

L'objet **markResponse** permet d'obtenir des informations sur la demande d'autorisation à 100 XPF .

markResponse	Format
amount Montant utilisé pour vérifier la validité de la carte, dans la plus petite unité monétaire .	n..12
currency Code de la monnaie utilisée pour vérifier la validité de la carte (suivant la norme ISO 4217) Ex : 953 pour le Franc CFP (XPF) .	n3
date Date et heure de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
number Numéro d'autorisation de la demande d'autorisation dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
result Résultat de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK . Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 30 : Objet markResponse

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

Cet objet se décompose :

- **authenticationRequestData**

Décrit le résultat de la demande d'enrôlement ainsi que le message encodé qui sera transmis par le navigateur de l'acheteur à l'ACS.

- **authenticationResultData**

Décrit les détails de l'authentification 3D Secure.

authenticationRequestData	Format
threeDSAcctId Certificat renvoyé par le Directory Server.	string
threeDSAcUrl Url de l'ACS à contacter.	string
threeDSBrand Réseau de la carte.	string
threeDSEncodedPareq Message PAREq encodé, prêt à envoyer à l'ACS.	string
threeDSEnrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	a1
threeDSRequestId Numéro de requête, à rappeler dans l'appel ENABLED_FINALIZE de l'attribut mode de l'objet threeDSRequest .	string

Tableau 31 : Objet authenticationRequestData

authenticationResultData	Format															
<div>transactionCondition Statut de l'authentification 3D Secure. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">COND_3D_SUCCESS Succès de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure et le porteur s'est authentifié correctement.COND_3D_FAILURE Echec de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas réussi à s'authentifier (mauvais mot de passe).COND_3D_ERROR Authentification en erreur. Le marchand participe au programme 3D Secure mais le serveur de la plateforme de paiement a rencontré un problème technique durant le processus d'authentification (lors de la vérification de l'inscription de la carte au programme 3D ou de l'authentification du porteur).COND_3D_NOTENROLLED Porteur non enrôlé. Le marchand participe au programme 3D Secure mais la carte du porteur n'est pas enrôlée.COND_3D_ATTEMPT Tentative d'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas eu à s'authentifier (le serveur de contrôle d'accès de la banque qui a émis la carte n'implémente que la génération d'une preuve de tentative d'authentification).COND_SSL 3D Secure non applicable. Le marchand n'est pas enrôlé à 3D Secure ou le canal de vente n'est pas couvert par cette garantie.</div>	string															
<div>enrolled Statut enrôlement du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut enrôlé.N pour un statut non enrôlé.U pour un statut inconnu.</div>	string															
<div>status Statut de l'authentification du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut authentifié 3 DS.N pour une erreur d'authentification.U pour une authentification impossible.A pour un essai d'authentification.</div>	a1															
<div>eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont :</div> <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>5</td><td>6</td><td>7</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>		status = Y	status = A	status = U	status = N	VISA - AMEX	5	6	7	-	MasterCard	02	01	-	-	string
	status = Y	status = A	status = U	status = N												
VISA - AMEX	5	6	7	-												
MasterCard	02	01	-	-												
<div>xid Numéro de transaction 3DS.</div>	string															
<div>cavvAlgorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">0 pour HMAC.1 pour CVV.</div>	n1															

authenticationResultData	Format
<ul style="list-style-type: none"> 2 pour CVV_ATN. 3 pour Mastercard SPA. 	
cavv Certificat de l'ACS.	string
signValid Signature de l'authentification 3DS.	string
brand Réseau de la carte.	string

Tableau 32 : Objet authenticationResultData

extraResponse

L'objet **extraResponse** permet d'obtenir des informations supplémentaires à propos du paiement.

extraResponse	Format
paymentOptionCode Code de l'option de paiement utilisée lors du paiement.	string
paymentOptionOccNumb Nombre d'échéances lors d'un paiement en plusieurs fois. Par exemple, pour un paiement en trois fois, cet attribut sera valorisé à 3.	n..2

Tableau 33 : Objet extraResponse

fraudManagementResponse

L'objet **fraudManagementResponse** permet d'obtenir les résultats des contrôles de gestion de risques.

La réponse se décompose en l'analyse des attributs :

- **riskControl**
Retourne le résultat du contrôle de gestion de risques effectué.
- **riskAnalysis**
Retourne le résultat de l'analyse de gestion de risques effectué par un système externe (Konduto, ClearSale, CyberSource,...).
- **riskAssessment**
Retourne le résultat de l'analyse de gestion des risques avancée effectuée par la plateforme de paiement.

L'attribut riskControl

Le format est le suivant : name1=result1;name2=result2

Nom	Format	Description
name	string	Nom de la règle de gestion de risque.
result	string	Résultat du contrôle.

Valeurs possibles pour 'name'	
CARD	Carte enregistrée en liste grise.
COUNTRY	Pays de l'acheteur enregistré en liste grise.
IPADDR	Adresse IP de l'acheteur enregistrée en liste grise.
AMOUNT	Montant maximum autorisé par commande est atteint.
BIN	Le code BIN de la carte est enregistré en liste grise.
ECB	Détection d'une e-carte bleue.
CARD_COMMERCIAL_NATIONAL	Détection d'une carte commerciale nationale.
CARD_COMMERCIAL_FOREIGN	Détection d'une carte commerciale étrangère.
CAS	Détection d'une carte à autorisation systématique.

Valeurs possibles pour 'name'	
COUNTRY_CONSISTENCY	Le pays d'origine de la carte, le pays de l'adresse IP de l'acheteur et le pays de l'acheteur ne correspondent pas.
NON_GUARANTEED_PAYMENT	Détection d'un paiement sans transfert de responsabilité.
IPADDR_COUNTRY	Le pays de l'adresse IP de l'acheteur est enregistré en liste grise.

Tableau 34 : Valeurs possibles pour 'name'

Valeurs possibles pour 'result'	
OK	Indique que le contrôle est correct.
KO	Indique que le contrôle est en erreur.

Tableau 35 : Valeurs possibles pour 'result'

L'attribut riskAnalysis

riskAnalysis	Format
score Score attribué à chaque transaction permettant d'évaluer le risque qui lui est associé.	string
resultCode Code renvoyé par un analyseur de risque externe. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour renvoyés par l'analyseur de risque externe .	string
status Statut de l'analyse de risque. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • P_SEND_OK, "Sent to clearsale and successfully processed" Succès • P_TO_SEND, "Transaction analysis is scheduled to be sent to risk analyzer" L'envoi est programmé • P_TO_SEND_KO, "Problem when tried to send to risk analyzer" Erreur de traitement • P_PENDING_AT_ANALYZER, "Analysis result is still being processed by the risk analyzer. We should keep checking/waiting for the analysis result" En cours de traitement par l'analyseur • P_MANUAL, "Analysis should be requested through user request (not automatically)" Attente d'envoi manuel • P_SKIPPED, "Analysis request discarded by current transaction status/problem" Ecarté • P_SEND_EXPIRED, "Analysis request expired" Expiré 	string
requestId Identifiant de l'analyse chez l'analyseur de risque.	string
extraInfo Pas de valorisation pour ClearSale. Pour CyberSource, cet attribut retourne tous les codes renvoyés par le DecisionManager. <ul style="list-style-type: none"> • key : nom de la donnée (son format est "string"). • value : valeur de la donnée (son format est "string"). Exemple : <ul style="list-style-type: none"> • COR-BA=Adresse de facturation corrigée ou corrigable • A=Changements d'adresse excessifs. L'acheteur a changé d'adresse de facturation au moins deux fois au cours des six derniers mois. • etc. 	extInfo

Tableau 36 : Attribut riskAnalysis

L'attribut riskAssessment

Valeurs	Description
ENABLE_3DS	3D Secure activé.
DISABLE_3DS	3D Secure désactivé.
MANUAL_VALIDATION	La transaction est créée en validation manuelle. La remise du paiement est bloquée temporairement pour permettre au marchand de procéder à toutes les vérifications souhaitées.
REFUSE	La transaction est refusée.
RUN_RISK_ANALYSIS	Appel à un analyseur de risques externes sous condition que le marchand possède un contrat. Se référer à la description du champ vads_risk_analysis_result pour identifier la liste des valeurs possibles et leur description.
INFORM	Une alerte est remontée. Le marchand est averti qu'un risque est identifié. Le marchand est informé via une ou plusieurs des règles du centre de notification (URL de notification, e-mail ou SMS).

shoppingCartResponse

L'objet **shoppingCartResponse** détaille le contenu du panier.

shoppingCartResponse																																			
Attribut	Format																																		
<p>cartItemInfo Champs personnalisables permettant d'ajouter les éléments du panier. L'attribut cartItemInfo est composé de sous objets :</p> <ul style="list-style-type: none"> productLabel : nom du produit. Son format est "string". productType : type de produit. Son format est "string (enum)". <table> <tr> <th>Valeur</th><th>Description</th></tr> <tr> <td>FOOD_AND_GROCERY</td><td>Produits alimentaires et d'épicerie</td></tr> <tr> <td>AUTOMOTIVE</td><td>Automobile / Moto</td></tr> <tr> <td>ENTERTAINMENT</td><td>Divertissement / Culture</td></tr> <tr> <td>HOME_AND_GARDEN</td><td>Maison et jardin</td></tr> <tr> <td>HOME_APPLIANCE</td><td>Equipement de la maison</td></tr> <tr> <td>AUCTION_AND_GROUP_BUYING</td><td>Ventes aux enchères et achats groupés</td></tr> <tr> <td>FLOWERS_AND_GIFTS </td><td>Fleurs et cadeaux</td></tr> <tr> <td>COMPUTER_AND_SOFTWARE</td><td>Ordinateurs et logiciels</td></tr> <tr> <td>HEALTH_AND_BEAUTY</td><td>Santé et beauté</td></tr> <tr> <td>SERVICE_FOR_INDIVIDUAL</td><td>Services à la personne</td></tr> <tr> <td>SERVICE_FOR_BUSINESS</td><td>Services aux entreprises</td></tr> <tr> <td>SPORTS</td><td>Sports</td></tr> <tr> <td>CLOTHING_AND_ACCESSORIES</td><td>Vêtements et accessoires</td></tr> <tr> <td>TRAVEL</td><td>Voyage</td></tr> <tr> <td>HOME_AUDIO_PHOTO_VIDEO</td><td>Son, image et vidéo</td></tr> <tr> <td>TELEPHONY</td><td>Téléphonie</td></tr> </table> <p><i>Tableau 37 : Valeurs associées à productType</i></p> <ul style="list-style-type: none"> productRef : référence produit. Son format est "string". productQty : quantité de produit. Son format est "integer". productAmount : montant en centimes du produit. Son format est "string". productVat : montant de la taxe sur le produit. Son format est "string". <p><u>Exemple :</u></p> <pre><cartItemInfo> <productLabel>CHIPS</productLabel> <productType>FOOD_AND_GROCERY</productType> <productRef>188545</productRef> <productQty>10</productQty> <productAmount>10000</productAmount> </cartItemInfo></pre>	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	cartItemInfo
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																																		

Créer un paiement sans authentification 3D Secure

Définir la cinématique du paiement sans authentification 3D Secure

La cinématique du paiement sans authentification 3D Secure est la suivante :

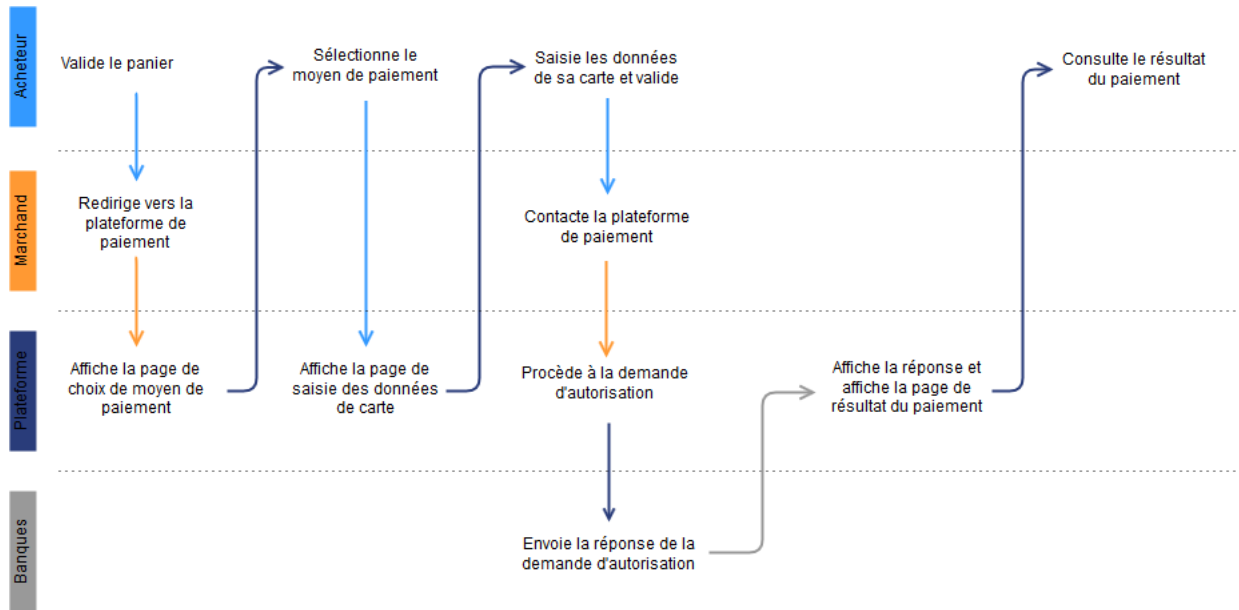


Image 2 : Cinématique du paiement sans authentification 3D Secure

1. L'acheteur valide sa commande et saisit les données de sa carte sur le site marchand pour procéder au paiement.
2. Le site marchand contacte la plateforme de paiement.

Il appelle l'opération **createPayment**.

Pour un paiement sans authentification 3D Secure il peut, au choix, :

- ne pas valoriser l'attribut **mode** de l'objet **threeDSRequest** (sans valorisation, la valeur **DISABLED** est affectée à l'attribut par défaut),
- valoriser l'attribut **mode** de l'objet **threeDSRequest** à **DISABLED**.

3. La plateforme de paiement procède à la demande d'autorisation et retourne le résultat du paiement au site marchand.

Exemple de code pour initier un paiement sans 3D Secure

Exemple de code pour initier un paiement sans 3D Secure

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:v5="http://v5.ws.vads.lyra.com/">
  <soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">
    <soapHeader:shopId>12345678</soapHeader:shopId>
    <soapHeader:requestId>9a4bf6ef-af95-4078-b791-4e9e87888eaa</soapHeader:requestId>
    <soapHeader:timestamp>2015-04-01T12:07:34Z</soapHeader:timestamp>
    <soapHeader:mode>TEST</soapHeader:mode>
    <soapHeader:authToken>OWXz7lNdyoaQInKJcDxaKA/djtx6lk7RbaMTgwPhFVo=</soapHeader:authToken>
  </soap:Header>
  <soap:Body>
    <v5:createPayment>
      <commonRequest>
        <paymentSource>EC</paymentSource>
        <submissionDate>2015-04-01T12:05:42Z</submissionDate>
      </commonRequest>
      <threeDSRequest>
        <mode>DISABLED</mode>
      </threeDSRequest>
      <paymentRequest>
        <amount>1</amount>
        <currency>953</currency>
      </paymentRequest>
      <orderRequest>
        <orderId>TEST-01</orderId>
      </orderRequest>
    </v5:createPayment>
  </soap:Body>
</soap:Envelope>
```

```

<cardRequest>
  <number>4970100000000000</number>
  <scheme>VISA</scheme>
  <expiryMonth>12</expiryMonth>
  <expiryYear>2015</expiryYear>
  <cardSecurityCode>123</cardSecurityCode>
  <cardHolderBirthDay>1976-04-18</cardHolderBirthDay>
</cardRequest>
<customerRequest>
  <billingDetails>
    <email>mail@example.com</email>
  </billingDetails>
  <extraDetails>
    <ipAddress>127.0.0.1</ipAddress>
  </extraDetails>
</customerRequest>
</v5:createPayment>
</soap:Body>
</soap:Envelope>

```

Exemple de réponse pour un paiement sans 3D Secure

Exemple de réponse pour un paiement sans 3D Secure réalisé avec succès

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">9a4bf6ef-af95-4078-b791-4e9e87888eaa</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T12:07:34Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">BmIP5CNeLyAQQDuzrFjvfoSWEZlCa5OidTV3WNqUXL4=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:createPaymentResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createPaymentResult>
        <requestId>9a4bf6ef-af95-4078-b791-4e9e87888eaa</requestId>
        <commonResponse>
          <responseCode>0</responseCode>
          <responseCodeDetail>Action successfully completed</responseCodeDetail>
          <transactionStatusLabel>AUTHORISED</transactionStatusLabel>
          <shopId>12345678</shopId>
          <paymentSource>EC</paymentSource>
          <submissionDate>2015-04-01T14:05:42+02:00</submissionDate>
          <contractNumber>5785350</contractNumber>
        </commonResponse>
        <paymentResponse>
          <transactionId>124478</transactionId>
          <amount>1</amount>
          <currency>953</currency>
          <expectedCaptureDate>2015-04-01T14:07:54.396+02:00</expectedCaptureDate>
          <operationType>0</operationType>
          <creationDate>2015-04-01T14:07:54.387+02:00</creationDate>
          <liabilityShift>NO</liabilityShift>
          <transactionUuid>a27d1907d7f74be1843318dce5875b99</transactionUuid>
        </paymentResponse>
        <orderResponse>
          <orderId>TEST-01</orderId>
        </orderResponse>
        <cardResponse>
          <number>497010XXXXXX0000</number>
          <scheme>CB</scheme>
          <brand>CB</brand>
          <country>NC</country>
          <productCode>A</productCode>
          <bankCode>17807</bankCode>
          <expiryMonth>12</expiryMonth>
          <expiryYear>2015</expiryYear>
        </cardResponse>
        <authorizationResponse>
          <mode>FULL</mode>
          <amount>1</amount>
          <currency>953</currency>
          <date>2015-04-01T14:07:54.387+02:00</date>
          <number>3fel9a</number>
          <result>0</result>
        </authorizationResponse>
        <captureResponse/>
        <customerResponse>
          <billingDetails>
            <email>mail@example.com</email>
            <language>fr_NC</language>
          </billingDetails>
          <shippingDetails/>
          <extraDetails>
            <ipAddress>127.0.0.1</ipAddress>
          </extraDetails>
        </customerResponse>
      </createPaymentResponse>
    </soap:Body>
  </soap:Envelope>

```

```

    </extraDetails>
  </customerResponse>
</markResponse/>
<threeDSResponse>
  <authenticationResultData>
    <transactionCondition>COND_SSL</transactionCondition>
  </authenticationResultData>
</threeDSResponse>
<extraResponse/>
<fraudManagementResponse>
  <riskControl>
    <name>CARD_FRAUD</name>
    <result>OK</result>
  </riskControl>
  <riskControl>
    <name>COMMERCIAL_CARD</name>
    <result>OK</result>
  </riskControl>
</fraudManagementResponse>
</createPaymentResult>
</ns2:createPaymentResponse>
</soap:Body>
</soap:Envelope>

```

Créer un paiement avec authentification 3D Secure

Définir la cinématique du paiement avec authentification 3D Secure

La cinématique du paiement avec authentification 3D Secure est la suivante :

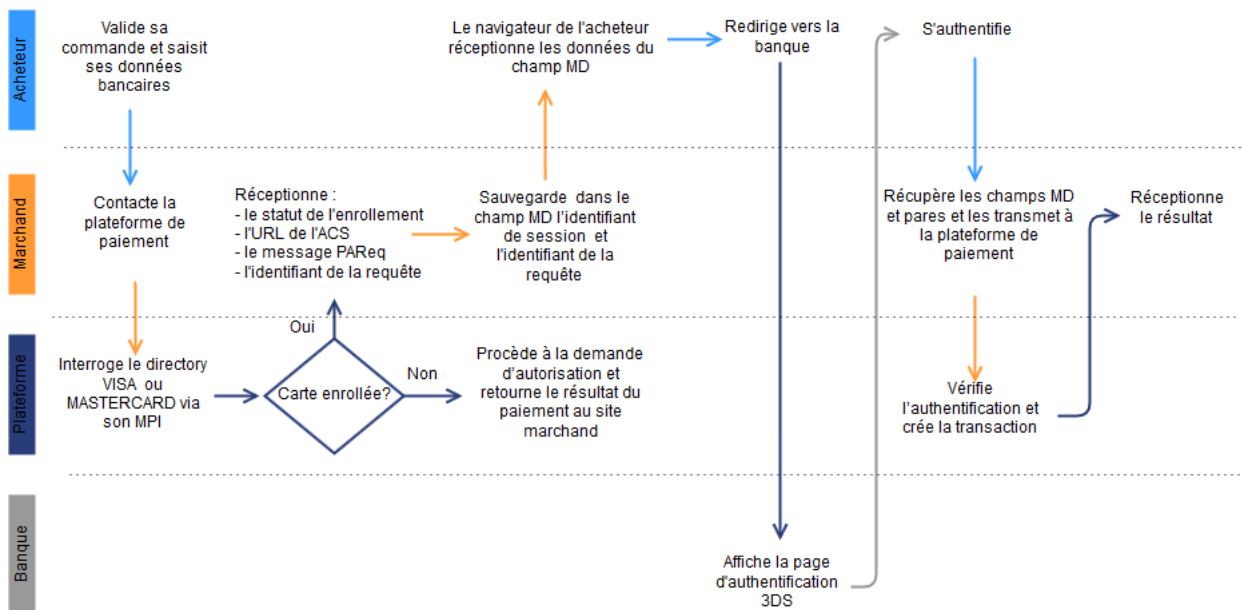


Image 3 : Cinématique du paiement avec authentification 3D Secure

1. L'acheteur valide sa commande et saisit les données de sa carte sur le site marchand pour procéder au paiement.

2. Le site marchand contacte la plateforme de paiement.

Il appelle l'opération **createPayment**. Il valorise l'attribut **mode** de l'objet **threeDSRequest** à **ENABLED_CREATE**.

3. La plateforme de paiement, via son MPI, interroge le directory serveur VISA, Mastercard ou AMEX (SafeKey).

- Si la carte n'est pas enrôlée, la plateforme de paiement procède à la demande d'autorisation et retourne le résultat du paiement au site marchand.

- l'attribut **threeDSEnrolled** de l'objet **authenticationRequestData** de **threeDSResponse** est valorisé à **N**.

- Si la carte est enrôlée :

la plateforme de paiement renvoie au marchand les informations suivantes :

- l'attribut **threeDSEnrolled** de l'objet **authenticationRequestData** de **threeDSResponse** est valorisé à **Y**,
- l'URL du site internet de la banque du porteur (ACS) vers laquelle le marchand devra rediriger l'acheteur,
- le message **PAREq** encodé (**threeDSEncodedPareq**),
- l'identifiant de la requête (**threeDSRequestId**).

4. Le site marchand sauvegarde dans le champ **MD** :

- l'identifiant de session contenu dans l'en-tête HTTP de la réponse (**JSESSIONID**),
- l'identifiant de la requête (**threeDSRequestId**) contenu dans la réponse **authenticationRequestData**.

MD est l'abréviation de MerchantData. Il s'agit d'un champ créé pour la requête.

5. Le site marchand envoie au navigateur de l'acheteur une requête HTTP POST avec les champs :
 - PaReq
 - TermUrl
 - MD
 6. L'acheteur est redirigé vers le site de la banque émettrice (ACS) et s'authentifie.
 7. A la fin de l'authentification, l'acheteur est redirigé vers le site marchand. Son navigateur effectue une requête POST à destination du site marchand contenant les champs **MD** et **PaRes**.
 8. Le site marchand récupère ces deux champs et les transmet à la plateforme de paiement pour vérifier l'authentification et créer la transaction.
- Pour créer la transaction, il doit :
- rappeler l'opération **createPayment** en renseignant l'attribut **mode** de l'objet **threeDSRequest** à **ENABLED_FINALIZE**,
 - renseigner l'attribut **requestId** de l'objet **threeDSRequest**,
 - renseigner l'attribut **pares** de l'objet **threeDSRequest**,
9. Le MPI de la plateforme de paiement vérifie les données contenues dans le **PaRes** :
 - l'acheteur ne s'est pas authentifié, le paiement est refusé.
 - l'acheteur s'est authentifié, la plateforme de paiement procède à la demande d'autorisation.
 10. La plateforme de paiement renvoie le résultat au site marchand (**authenticationResultData**).

Initialiser un paiement avec 3D Secure

Initialiser un paiement avec 3D Secure

L'exemple ci-dessous permet d'initialiser un paiement (**createPayment**) avec authentification 3D Secure.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:v5="http://v5.ws.vads.lyra.com/">
  <soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">
    <soapHeader:shopId>12345678</soapHeader:shopId>
    <soapHeader:requestId>09016abd-2869-40cc-b32f-a467bd541e42</soapHeader:requestId>
    <soapHeader:timestamp>2015-04-01T12:09:44Z</soapHeader:timestamp>
    <soapHeader:mode>TEST</soapHeader:mode>
    <soapHeader:authToken>ASQu/agMyf5UDEkd2HZbBZAKkIjrAMoJV98ZN2W9o/g=</soapHeader:authToken>
  </soap:Header>
  <soap:Body>
    <v5:createPayment>
      <commonRequest>
        <paymentSource>EC</paymentSource>
        <submissionDate>2015-04-01T12:09:44Z</submissionDate>
      </commonRequest>
      <threeDSRequest>
        <mode>ENABLED_CREATE</mode>
      </threeDSRequest>
      <paymentRequest>
        <amount>1</amount>
        <currency>953</currency>
        <manualValidation>1</manualValidation>
      </paymentRequest>
      <orderRequest>
        <orderId>TEST-01</orderId>
      </orderRequest>
      <cardRequest>
        <number>4970100000000009</number>
        <scheme>VISA</scheme>
        <expiryMonth>12</expiryMonth>
        <expiryYear>2015</expiryYear>
        <cardSecurityCode>123</cardSecurityCode>
        <cardHolderBirthDay>1976-04-18</cardHolderBirthDay>
      </cardRequest>
    </v5:createPayment>
  </soap:Body>
</soap:Envelope>
```

La création d'un paiement avec authentification 3D Secure retourne un objet **threeDSResponse**.

Exemples de réponses à un paiement avec authentification 3D Secure

Exemples de réponses : paiement avec authentification 3D Secure demandée

- Cas 1 : la carte est enrôlée

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">09016abd-2869-40cc-b32f-a467bd541e42</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T12:09:44Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">3fRs8IRBhkPvomIoILJmP/4sdfWg4V5AbppfGEN7PW0=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:createPaymentResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createPaymentResult>
        <requestId>09016abd-2869-40cc-b32f-a467bd541e42</requestId>
        <commonResponse>
          <responseCode>0</responseCode>
          <responseCodeDetail>Action successfully completed</responseCodeDetail>
        </commonResponse>
        <paymentResponse/>
        <orderResponse/>
        <cardResponse/>
        <authorizationResponse/>
        <captureResponse/>
        <customerResponse/>
        <markResponse/>
        <threeDSResponse>
          <authenticationRequestData>
            <threeDSAcctId>98b7b83c590d4aaabab3667b4ec2</threeDSAcctId>
            <threeDSAcUrl>https://[ACS-URL]</threeDSAcUrl>
            <threeDSBrand>VISA</threeDSBrand>
            <threeDSEncodedPareq>eJxVUstu2zAQ/BVBd5kPiaZkrBgkNYqka[...]</threeDSEncodedPareq>
            <threeDSEnrolled>Y</threeDSEnrolled>
            <threeDSRequestId>_a7c5c935-24d1-4d6c-873e-7e29f4e5dec1</threeDSRequestId>
          </authenticationRequestData>
        </threeDSResponse>
        <extraResponse/>
        <fraudManagementResponse/>
      </createPaymentResult>
    </ns2:createPaymentResponse>
  </soap:Body>
</soap:Envelope>
```

- Cas 2 : la carte n'est pas enrôlée

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">c63b5e8e-e6b9-47f1-8890-98ff62f4d018</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-09-17T09:19:44Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">BKlVnOafVsEJTtdQRvYgHvHTjlt68CFKefT0s8sikk=</authToken>
  </SOAP-ENV:Header>
  <soap:Body>
    <ns2:createPaymentResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createPaymentResult><requestId>c63b5e8e-e6b9-47f1-8890-98ff62f4d018</requestId>
      <commonResponse>
        <responseCode>0</responseCode>
        <responseCodeDetail>Action successfully completed</responseCodeDetail>
        <transactionStatusLabel>AUTHORISED</transactionStatusLabel>
        <shopId>12345678</shopId>
        <paymentSource>EC</paymentSource>
        <submissionDate>2015-09-17T11:19:44+02:00</submissionDate>
        <contractNumber>5555555</contractNumber>
      </commonResponse>
      <paymentResponse>
        <transactionId>962003</transactionId>
        <amount>2990</amount>
        <currency>953</currency>
        <expectedCaptureDate>2015-09-17T11:19:44.732+02:00</expectedCaptureDate>
        <operationType>0</operationType>
        <creationDate>2015-09-17T11:19:44.702+02:00</creationDate>
        <transactionUuid>04474adae6c24b688d4892a1a80833c3</transactionUuid>
      </paymentResponse>
    </ns2:createPaymentResponse>
  </soap:Body>
</soap:Envelope>
```

```

        <sequenceNumber>1</sequenceNumber>
        <paymentType>SINGLE</paymentType>
    </paymentResponse>
    <orderResponse>
        <orderId>myOrder</orderId>
    </orderResponse>
    <cardResponse>
        <number>497010XXXXXX0001</number>
        <scheme>CB</scheme>
        <brand>CB</brand>
        <country>NC</country>
        <productCode>F</productCode>
        <expiryMonth>12</expiryMonth>
        <expiryYear>2023</expiryYear>
    </cardResponse>
    <authorizationResponse>
        <mode>FULL</mode>
        <amount>2990</amount>
        <currency>953</currency>
        <date>2015-09-17T11:19:44.702+02:00</date>
        <number>3fc075</number>
        <result>0</result>
    </authorizationResponse>
    <captureResponse/>
    <customerResponse>
        <billingDetails>
            <email>test@exemple.com</email>
            <language>fr_NC</language>
        </billingDetails>
        <shippingDetails/>
        <extraDetails/>
    </customerResponse>
    <markResponse/>
    <threeDSResponse>
        <authenticationRequestData/>
        <authenticationResultData>
            <brand>VISA</brand>
            <enrolled>N</enrolled>
            <transactionCondition>COND_3D_NOTENROLLED</transactionCondition>
        </authenticationResultData>
    </threeDSResponse>
    <extraResponse/>
    <fraudManagementResponse>
        <riskControl>
            <name>COMMERCIAL_CARD</name>
            <result>OK</result>
        </riskControl>
        <riskControl>
            <name>SYSTEMATIC_AUTO</name>
            <result>OK</result>
        </riskControl>
        <riskAssesments/>
    </fraudManagementResponse>
    </createPaymentResult>
</ns2:createPaymentResponse>
</soap:Body>
</soap:Envelope>

```

Maintenir une même session HTTP pour un paiement avec authentification 3D Secure

Maintenir une même session HTTP pour un paiement avec authentification 3D Secure

L'architecture de la plateforme de paiement repose sur un ensemble de serveurs avec répartition de charge.

Pour assurer la continuité du processus, chaque requête associée à un même paiement doit être réalisée avec la même session HTTP.

Ainsi pour chaque requête **createPayment** avec authentification 3D Secure (**threeDSRequest**), une session côté serveur est créée.

L'ID de cette session doit être renvoyé dans l'en-tête HTTP de la réponse et devra être retourné dans la requête **createPayment** en mode **ENABLED_FINALIZE**.

Selon le langage utilisé, ci-dessous deux exemples pour réaliser cette opération :

- **En JAVA**

Utilisez la propriété **SESSION_MAINTAIN_PROPERTY** en vous assurant qu'elle soit définie à **True** pour récupérer automatiquement les informations de session associées à la requête HTTP et maintenir un cookie avec l'ID de session (Standard Java , JAX-WS).

```
((BindingProvider)port).getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY,true);
```

- **En PHP**

Voici un exemple de code pour récupérer l'id de session et le transmettre :

```
/* La méthode ci-dessous permet de récupérer l'entête HTTP de la réponse
*/$header= $client -> __getLastResponseHeaders ();
/* Dans la chaîne de caractère obtenue, nous recherchons la présence de l'ID de la session
HTTP, stockée dans l'élément "JSESSIONID" : */
if(!preg_match("#JSESSIONID=([A-Za-z0-9\.\.])#", $header, $matches)){
return "Aucun ID de Session Renvoyé. " ; //Cas d'erreur technique
}
$cookie= $matches[1] ) ;
/*La méthode ci-dessous permet de spécifier un cookie qui sera envoyé dans chaque entête http
*/
$client -> __setCookie ("JSESSIONID",$cookie );
```

Cette méthode permet au serveur de récupérer le contenu du header et le transmettre en tant que cookie dans la requête HTTP.

Rediriger le navigateur de l'acheteur vers son ACS

Rediriger le navigateur de l'acheteur vers son ACS

Après avoir récupéré le contenu de l'objet **authenticationRequestData**, il faut rediriger le navigateur de l'acheteur vers son ACS, en renvoyant une page HTML avec un formulaire POST auto soumis.

L'url de l'ACS est utilisée comme action du POST. Sa valeur est celle retournée dans le champ **threeDSAcsUrl**.

Il faut également disposer d'une URL de retour sur le serveur pour récupérer la réponse de l'ACS retournée par POST.

Ce formulaire doit obligatoirement contenir les attributs suivants :

- **PaReq**

Message PaReq encodé, prêt à envoyer à l'ACS.

- **TermUrl**

URL de retour pour traiter le retour de l'ACS.

- **MD**

Il contient l'identifiant de session contenu dans l'en-tête HTTP de la réponse (JSESSIONID) et l'identifiant de la requête (**threeDSRequestId**) contenu dans l'objet **authenticationRequestData** de la réponse, séparés par un délimiteur (par exemple le caractère « + »).

Ces données seront restituées lors de la réponse de l'ACS.

Note concernant le mode TEST :

Afin de conserver la continuité des transactions en mode test, il est nécessaire de transmettre l'identifiant de la session lors de la redirection vers l'ACS.

Ceci devra se faire en concaténant :

- L'URL de l'ACS obtenue dans la réponse **authenticationRequestData**
- L'identifiant de session renvoyé dans l'en-tête http, séparés par « ;jsessionId= »

La syntaxe à respecter est : \${URL};jsessionid=\${session}

Exemple :

```
<form name="Form" method="post" action=https://epaync.nc/vads-payment/
acs.silent_authenticate.a;jsessionId=B420BF68835F6563FB6E4B289ABB9080.bdxvad3" >
...
</form>
```

EN MODE PRODUCTION VOUS NE DEVEZ EN AUCUN CAS TRANSMETTRE UN IDENTIFIANT DE SESSION A L'ACS

Récupérer la réponse de l'ACS

Récupérer la réponse de l'ACS

Pour récupérer la réponse de l'ACS, il est nécessaire de mettre en place une URL de retour de l'ACS.

Celle-ci permettra de renvoyer les données du **PaRes** au site marchand.

Les paramètres renvoyés par l'ACS sont les suivants :

- **PaRes** : contient le message **PaRes** (Payer Authentication Response).
- **MD** : contient l'identifiant de session contenu dans l'en-tête de la réponse et l'identifiant de la requête (**threeDSRequestId**) contenu dans la réponse **authenticationRequestData**.

Ces deux valeurs doivent être extraites afin de les utiliser lors de l'appel à l'opération **createPayment** lorsque l'attribut **mode** de l'objet **threeDSRequest** sera valorisé à **ENABLED_FINALIZE**.

Exemple :

```
PaRes:eJzNWVmPo0i2frfk/1CqeXRXsZjFtJw5YjVgg81mlpcrdjCrzWr/+glnlpLdqjvqOz0PN6UUcIg4ceIs33cCb/85V+WnMb51eVO/fEa+wp8/xXXYRHmdvny2TOHL5vM/X7dmdotjzD4Ra/bpW46/w0/pRHL5//hyBQHESi/EuyJsMvGBmvv1BUspLCU1QUUFiAoQH5+XV7ovW4e5vR5WkdR1//6sRvtr0C076iW+j7IzDiFmZ+3b9u/[...]  
MD:pcpSryKqB0NynWVHj8LQj0uz+_66254f65-f37c-47e3-99b8-799db94b42b7
```

Dans cet exemple, le champ **MD** est composé de l'identifiant de la session et de l'identifiant de la requête, séparés par le caractère « + » :

```
JSESSIONID: pcpSryKqB0NynWVHj8LQj0uz  
requestId: _66254f65-f37c-47e3-99b8-799db94b42b7
```

Vérifier l'authentification 3DS et finaliser le paiement

Vérifier l'authentification 3DS et finaliser le paiement

Pour vérifier le résultat de l'authentification 3DS et finaliser le paiement, il faut soumettre à la plateforme de paiement :

- le **requestId**,
- le message **PaRes** reçu après l'authentification 3DS.

Pour cela, il est nécessaire d'appeler l'opération **createPayment** et valoriser l'attribut **mode** de l'objet **threeDSRequest** à **ENABLED_FINALIZE**.

Cette opération retournera une réponse **threeDSResponse** avec un objet **authenticationResultData**.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:v5="http://v5.ws.vads.lyra.com/">  
  <soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">  
    <soapHeader:shopId>12345678</soapHeader:shopId>  
    <soapHeader:requestId>1416ffba-66ca-4609-bac8-041764fa4cad</soapHeader:requestId>  
    <soapHeader:timestamp>2015-04-01T12:18:21Z</soapHeader:timestamp>  
    <soapHeader:mode>TEST</soapHeader:mode>  
    <soapHeader:authToken>J8wgBbvfbMGWZAMUbn+3HFUJMb1BG+//rkclkJOz2aA=</soapHeader:authToken>  
  </soap:Header>  
  <soap:Body>  
    <v5:createPayment>  
      <commonRequest>  
        <paymentSource>EC</paymentSource>  
        <submissionDate>2015-04-01T12:18:21Z</submissionDate>  
      </commonRequest>  
      <threeDSRequest>  
        <mode>ENABLED_FINALIZE</mode>  
        <pares>eJzNWVmPo0i2frfk/1CqeXRXsZjFtJw5YjVgg81mlpcrdjCrzWr/+glnlpLdqjvqOz0PN6UUcIg4ceIs33cCb/85V+WnMb51eVO/fEa+wp8/xXXYRHmdvny2TOHL5vM/X7dmdotjzD4Ra/bpW46/w0/pRHL5//hyBQHESi/EuyJsMvGBmvv1BUspLCU1QUUFiAoQH5+XV7ovW4e5vR5WkdR1//6sRvtr0C076iW+j7IzDiFmZ+3b9u/[...] </pares>  
        <requestId>_66254f65-f37c-47e3-99b8-799db94b42b7</requestId>  
      </threeDSRequest>  
    </v5:createPayment>  
  </soap:Body>  
</soap:Envelope>
```

```
</soap:Body>
</soap:Envelope>
```

Exemple de fichier réponse généré suite à l'appel **ENABLED FINALIZE** :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">1416ffba-66ca-4609-bac8-041764fa4cad</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T12:18:21Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">CwOM0syYYLzxcDyY0+7JyNi70uUbNEYGUAD81MttVnA=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:createPaymentResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createPaymentResult>
        <requestId>1416ffba-66ca-4609-bac8-041764fa4cad</requestId>
        <commonResponse>
          <responseCode>0</responseCode>
          <responseCodeDetail>Action successfully completed</responseCodeDetail>
          <transactionStatusLabel>AUTHORISED_TO_VALIDATE</transactionStatusLabel>
          <shopId>12345678</shopId>
          <paymentSource>EC</paymentSource>
          <submissionDate>2015-04-01T14:09:44+02:00</submissionDate>
          <contractNumber>5785350</contractNumber>
        </commonResponse>
        <paymentResponse>
          <transactionId>124472</transactionId>
          <amount>1</amount>
          <currency>953</currency>
          <expectedCaptureDate>2015-04-01T14:18:58.522+02:00</expectedCaptureDate>
          <operationType>0</operationType>
          <creationDate>2015-04-01T14:18:58.514+02:00</creationDate>
          <transactionUuid>170fd39a02c847feb5a5f750e8b320d2</transactionUuid>
          <sequenceNumber>1</sequenceNumber>
          <paymentType>SINGLE</paymentType>
        </paymentResponse>
        <orderResponse>
          <orderId>TEST-01</orderId>
        </orderResponse>
        <cardResponse>
          <number>497010XXXXXX0009</number>
          <scheme>CB</scheme>
          <brand>CB</brand>
          <country>NC</country>
          <productCode>G1</productCode>
          <bankCode>17807</bankCode>
          <expiryMonth>12</expiryMonth>
          <expiryYear>2015</expiryYear>
        </cardResponse>
        <authorizationResponse>
          <mode>FULL</mode>
          <amount>1</amount>
          <currency>953</currency>
          <date>2015-04-01T14:18:58.514+02:00</date>
          <number>3fea6c</number>
          <result>0</result>
        </authorizationResponse>
        <captureResponse/>
        <customerResponse>
          <billingDetails>
            <language>fr_NC</language>
          </billingDetails>
          <shippingDetails/>
          <extraDetails/>
        </customerResponse>
        <markResponse/>
        <threeDSResponse>
          <authenticationResultData>
            <brand>VISA</brand>
            <enrolled>Y</enrolled>
            <status>Y</status>
            <eci>05</eci>
            <xid>TUN3a0JZczB2QlRDeHZTT2lqakk=</xid>
            <cavv>Q2F2dkNhdnZDYXZ2Q2F2dkNhdnY=</cavv>
            <cavvAlgorithm>2</cavvAlgorithm>
            <transactionCondition>COND_3D_SUCCESS</transactionCondition>
          </authenticationResultData>
        </threeDSResponse>
        <extraResponse/>
        <fraudManagementResponse>
          <riskControl>
            <name>CARD_FRAUD</name>
            <result>OK</result>
          </riskControl>
        </fraudManagementResponse>
      </createPaymentResult>
    </ns2:createPaymentResponse>
  </soap:Body>
</soap:Envelope>
```

```

    <riskControl>
      <name>COMMERCIAL_CARD</name>
      <result>OK</result>
    </riskControl>
  </fraudManagementResponse>
</createPaymentResult>
</ns2:createPaymentResponse>
</soap:Body>
</soap:Envelope>

```

Rejouer un paiement refusé

1. Récupérez la valeur de l'attribut **transactionUuid** dans le paiement qui a échoué.

Exemple :

```

<paymentResponse>
  <transactionId>901542</transactionId>
  <amount>10101</amount>
  <currency>953</currency>
  <expectedCaptureDate>2017-01-10T17:32:09+01:00</expectedCaptureDate>
  <operationType>0</operationType>
  <creationDate>2017-02-28T14:45:07.915+01:00</creationDate>
  <transactionUuid>2c9b916d95b6464aa4a0848bd4a4fd0a</transactionUuid>
  <sequenceNumber>1</sequenceNumber>
  <paymentType>SINGLE</paymentType>
</paymentResponse>

```

2. Renseignez, dans la requête suivante, au niveau de l'objet **paymentRequest**, l'attribut **retryUuid** avec la valeur de l'attribut **transactionUuid** comme suit :

```

<paymentRequest>
  <amount>10101</amount>
  <currency>953</currency>
  <expectedCaptureDate>2017-01-10T16:32:09Z</expectedCaptureDate>
  <retryUuid>2c9b916d95b6464aa4a0848bd4a4fd0a</retryUuid>
</paymentRequest>

```

La réponse contiendra les éléments suivants :

```

<paymentResponse>
  <transactionId>901542</transactionId>
  <amount>10101</amount>
  <currency>953</currency>
  <expectedCaptureDate>2017-01-10T17:32:09+01:00</expectedCaptureDate>
  <operationType>0</operationType>
  <creationDate>2017-02-28T14:47:31.107+01:00</creationDate>
  <liabilityShift>NO</liabilityShift>
  <transactionUuid>45e1f7aebb4d409f82d05aac20dcb82f</transactionUuid>
  <sequenceNumber>2</sequenceNumber>
  <paymentType>RETRY</paymentType>
</paymentResponse>

```

6.2. Modifier une transaction de paiement 'updatePayment'

updatePayment permet de :

- Modifier le montant d'une transaction (à la baisse)
- Modifier la date de remise souhaitée

Les transactions pouvant faire l'objet d'une modification possèdent l'un des statuts suivants :

- A valider
- A valider et autoriser
- En attente d'autorisation
- En attente de remise

Remarque : si aucune information n'est modifiée, la requête sera rejetée avec un code erreur.

Requête à envoyer

La requête **updatePayment** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **updatePayment** prend en entrée un objet de type **updatePayment**.

Le type **updatePayment** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓
paymentRequest	paymentRequest	✓

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Un seul attribut peut être valorisé si besoin :

commonRequest		
Attribut	Requis	Format
comment Commentaire libre.		string

Tableau 38 : Objet commonRequest

Le commentaire sera affiché dans l'historique de la transaction visible depuis le Back Office.

paymentRequest

L'objet **paymentRequest** permet de transmettre des informations liées au paiement.

Il possède les attributs suivants :

paymentRequest		
Attribut	Requis	Format
amount Montant de la transaction dans sa plus petite unité monétaire . <u>Remarque :</u> <ul style="list-style-type: none">Ne doit pas être envoyé à vide ou être à 0.Ne doit pas être supérieur au montant initial (cas du remboursement).	✓	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	✓	n3
expectedCaptureDate Date de remise demandée exprimée au format ISO 8601 défini par W3C. <i>Exemple : 2016-07-16T19:20:00Z.</i> Ce paramètre est utilisé pour effectuer un paiement différé. Si le nombre de jours entre la date de remise demandée et la date actuelle est supérieur à la durée de validité de l'autorisation , une autorisation de 100 XPF sera réalisée le jour de la transaction. Ceci afin de vérifier la validité de la carte. L'autorisation pour le montant total sera effectuée : <ul style="list-style-type: none">fonctionnement par défaut : le jour de la date de remise en banque souhaitée,fonctionnement avec autorisation anticipée : selon le moyen de paiement sélectionné, à J- le nombre de jours correspondant à la durée de validité d'une autorisation avant la date de remise en banque souhaitée. Si vous souhaitez être notifié du résultat de cette demande d'autorisation, vous devez configurer la règle de notification URL de notification sur autorisation par Batch dans le Back Office (Paramétrage > Règles de notifications). <u>Remarque :</u> si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.		dateTime ans..40
manualValidation Permet de valider manuellement une transaction préalablement créée avec une validation manuelle tant que la date de remise en banque souhaitée n'est pas dépassée. Pour cela, cet attribut doit être valorisé à 1 (validation manuelle). <u>Remarques :</u> <ul style="list-style-type: none">Si le paiement a été créé en validation automatique, l'attribut manualValidation n'a aucune utilité.Si manualValidation est valorisé à 0, l'action demandée ne sera pas prise en considération.		n1
acquiereTransientData Permet de transmettre des informations utiles à l'acquéreur pour réaliser des contrôles anti-fraude: URL de la boutique, nom de la boutique, numéro de commande, frais de port, méthode de livraison, etc.. Exemple :		json

paymentRequest		
Attribut	Requis	Format
acquirerTransientData = {VISANET={ "field91": "10XXXXXXX", "field92": "Lyra Peru", "field93": "specific data" }}		

Tableau 39 : Objet paymentRequest

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

L'attribut à valoriser est le suivant :

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction. Remarque: cet attribut est utilisé pour remplacer les anciens attributs transactionId , sequenceNumber et creationDate . Cependant il est possible de continuer à utiliser ces anciens attributs pour des raisons de retrocompatibilité. Pour plus d'informations, référez-vous au chapitre Gérer le rétrocompatibilité .	✓	string

Tableau 40 : Objet queryRequest

Les attributs **orderId**, **subscriptionId** et **paymentToken** ne sont pas aujourd'hui pris en considération pour cette opération.

Réponse en retour

La réponse à l'opération **updatePayment** est constituée d'un HEADER et d'un BODY de type **updatePaymentResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **updatePaymentResponse** est la suivante :

Nom	Type
updatePaymentResult	updatePaymentResult

La structure du message **updatePaymentResult** est la suivante :

Objet	Type
commonResponse	commonResponse
paymentResponse	paymentResponse
orderResponse	orderResponse
cardResponse	cardResponse
authorizationResponse	authorizationResponse
captureResponse	captureResponse
customerResponse	customerResponse
markResponse	markResponse
threeDSResponse	threeDSResponse
extraResponse	extraResponse
fraudManagementResponse	fraudManagementResponse

Remarque : l'objet **subscriptionResponse** n'est pas valorisé dans la réponse.

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue.AUTHORISED_TO_VALIDATE A valider. La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la remise en banque. La transaction peut être validée tant que la date de remise n'est pas dépassée. Si cette date est dépassée, le paiement prend le statut Expiré (statut définitif).WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement.WAITING_AUTHORISATION_TO_VALIDATE A valider et à autoriser. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation sera déclenchée automatiquement à J-1 avant la date de remise en banque. Le paiement pourra être accepté ou refusé. La remise en banque est automatique.REFUSED Refusée. La transaction est refusée.	string
shopId Identifiant de la boutique.	n8
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">EC pour le commerce électronique.MOTO pour une commande par e-mail ou téléphone.CC pour un centre d'appel.OTHER pour un autre canal de vente.	string (enum)
submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé.	string

Tableau 41 : Objet commonResponse

paymentResponse

L'objet **paymentResponse** détaille les informations sur la transaction.

paymentResponse	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none">• Soit cet identifiant est généré par la plateforme.• Soit cet identifiant est généré par le site marchand.	an..6
amount Montant de la transaction dans sa plus petite unité monétaire .	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	n3
effectiveAmount Montant de la transaction dans la devise réellement utilisée pour effectuer la remise en banque.	n..12
effectiveCurrency Devise réellement utilisée pour effectuer la remise en banque.	n3
expectedCaptureDate Date de remise en banque souhaitée. <u>Remarque</u> : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.	dateTime ans..40
operationType Type d'opération. Les valeurs possibles sont : <ul style="list-style-type: none">• 0 pour une opération de débit.• 1 pour une opération de remboursement.	n1
creationDate Date et heure de l'enregistrement de la transaction exprimée au format W3C.	dateTime ans..40
externalTransactionId Référence fournie par un tiers : numéro de transaction pour PayPal, Boleto, RRN pour Prism, etc...	string
liabilityShift Transfert de responsabilité. Les valeurs possibles sont : <ul style="list-style-type: none">• YES lorsque le paiement est garanti.• NO lorsque le paiement n'est pas garanti.	string
paymentType Type de paiement. Les valeurs possibles sont : <ul style="list-style-type: none">• SINGLE Païement en une seule fois.• INSTALLMENT Païement en plusieurs fois.• SPLIT Païement effectué avec plusieurs moyens de paiement.• SUBSCRIPTION Païement par alias ou lié à un abonnement.• RETRY Lors d'un paiement refusé, il est possible de réitérer la demande de paiement. Pour toute(s) réitération(s), le paiement est valorisé avec cette valeur. <u>Remarque</u> : Les valeurs INSTALLMENT et SPLIT peuvent être retournées uniquement si des paiements ont été créés via le formulaire de paiement.	string (enum)

paymentResponse	Format
sequenceNumber Numéro de séquence de la transaction. Sa valeur est fonction du contexte du paiement : <ul style="list-style-type: none"> Est valorisé à "1" pour un paiement unitaire. Est valorisé à "2" lors du rejeu d'un paiement initialement refusé. Prend la valeur du numéro d'échéance dans le cas d'un paiement en plusieurs fois créé à partir du formulaire de paiement. 	n..3
paymentError Complément d'information en cas d'erreur technique. Retourne un code d'erreur associé à l'erreur technique (voir chapitre Codes d'erreurs pouvant survenir durant le paiement).	n..3
nsu Apparaît lorsque isNsuRequested est valorisé à 1 dans la requête.	string

Tableau 42 : Objet paymentResponse

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
orderId Référence de la commande.	string an..64
extInfo Données personnalisées retournées en fonction des besoins. Exemple : extInfo.key, extInfo.value <ul style="list-style-type: none"> key : nom de la donnée (son format est "string"). value : valeur de la donnée (son format est "string"). 	extInfo

Tableau 43 : Objet orderResponse

cardResponse

L'objet **cardResponse** détaille les informations du moyen de paiement utilisé.

cardResponse	Format
number <ul style="list-style-type: none"> Numéro de carte masqué. Contient les 6 premiers chiffres du numéro, suivi par "XXXXXX" et enfin les 4 derniers chiffres. IBAN et BIC utilisés pour le paiement, séparés par un « _ » dans le cas d'un paiement par prélèvement. 	string
scheme Type de la carte.	string
brand Marque de la carte utilisée pour la transaction	string
country Code pays du pays d'émission de la carte (Code numérique ISO 3166).	ISO 3166
productCode Code produit de la carte.	an..3
bankCode Code banque de la banque émettrice.	n..5
expiryMonth Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).	n..2
expiryYear Année d'expiration sur 4 chiffres (ex : 2023).	n4

Tableau 44 : Objet cardResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
mode Spécifie de quelle manière est réalisée la demande d'autorisation. Deux valeurs possibles : <ul style="list-style-type: none"> • MARK Une autorisation de a été réalisée afin de vérifier la validité de la carte. Ce cas se présente lorsque la date de remise dépasse la période de validité d'une autorisation . • FULL Autorisation pour le montant total demandé dans la requête. 	string
amount Montant de l'autorisation dans la plus petite unité monétaire dans le cas où mode vaut FULL .	n..12
currency Code de la monnaie utilisée lors de la demande d'autorisation (suivant la norme ISO 4217) dans le cas où mode vaut FULL .	n3
date Date et heure de la demande d'autorisation dans le cas où mode vaut FULL .	dateTime ans..40
number Numéro de la demande d'autorisation dans le cas où mode vaut FULL .	n..2
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 45 : Objet *authorizationResponse*

captureResponse

L'objet **captureResponse** permet d'obtenir des informations à propos de la remise dans le cas où la transaction est remise.

captureResponse	Format
date Date et heure de remise.	dateTime ans..40
number Numéro de remise.	n3
reconciliationStatus Statut du rapprochement bancaire de la transaction.	n1
refundAmount Montant ayant déjà fait l'objet d'un remboursement dans sa plus petite unité monétaire.	n..12
refundCurrency Devise du montant ayant déjà fait l'objet d'un remboursement (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF) .	n3
chargeback Litige, impayé. Les valeurs possibles sont : <ul style="list-style-type: none"> • 0 : transaction ne faisant pas l'objet d'un litige. • 1 : transaction faisant l'objet d'un litige. 	

Tableau 46 : Objet *captureResponse*

customerResponse

L'objet **customerResponse** détaille de nombreuses informations à propos de l'acheteur.

La réponse se décompose en l'analyse de :

- **billingDetails**
Données de facturation de l'acheteur.
- **shippingDetails**
Données de livraison de l'acheteur. Les valeurs présentes dans cet objet seront identiques à celles de l'objet **billingDetails**.
- **extraDetails**

6.2.2.7.1. Objet billingDetails

billingDetails	
Attribut	Format
reference Référence de l'acheteur.	string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..	string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> • PRIVATE pour un particulier. • COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de l'acheteur.	string ans..255
district Quartier de l'acheteur.	string ans..127
zipCode Code postal de l'acheteur.	string ans..64
city Ville de l'acheteur.	string ans..128
state Etat/Région de l'acheteur.	string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none"> • DE pour l'Allemagne • IT pour l'Italie • PT pour le Portugal • GB pour le Royaume-Uni • JA pour le Japon • RU pour la Russie • ES pour l'Espagne • NL pour les Pays-Bas • SE pour la Suède • FR pour la France • PL pour la Pologne • CN pour la Chine 	string - a2
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> • de pour l'allemand • it pour l'italien • pt pour le portugais • en pour l'anglais • ja pour le japonais • ru pour le russe • es pour l'espagnol • nl pour le néerlandais • sv pour le suédois 	string - a2

billingDetails	
Attribut	Format
<ul style="list-style-type: none"> fr pour le français pl pour le polonais zh pour le chinois tr pour le turc 	
cellPhoneNumber Numéro de téléphone mobile	string ans..32
legalName Raison sociale de la société.	string ans..128

Tableau 47 : Objet billingDetails

6.2.2.7.2. Objet shippingDetails

shippingDetails	
Attribut	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de livraison.	string ans..255
address2 Complément d'adresse de livraison.	string ans..255
district Quartier de livraison.	string ans..127
zipCode Code postal de livraison.	string ans..64
city Ville de livraison.	string ans..128
state Etat/Région de livraison.	string ans..128
country Pays de livraison.	string - a2
deliveryCompanyName Informations sur le transporteur.	string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> STANDARD pour une livraison standard. EXPRESS pour une livraison express. 	string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont :	string (enum)

shippingDetails	
Attribut	Format
<ul style="list-style-type: none"> • RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. • RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). • RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. • PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). • ETICKET pour l'émission d'un billet électronique, téléchargement. 	
legalName Raison sociale de la société.	string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).	string ans..255

Tableau 48 : Objet shippingDetails

6.2.2.7.3. Objet extraDetails

extraDetails	Format
ipAddress L'adresse IP de l'acheteur.	string ans40

Tableau 49 : Objet extraDetails

markResponse

L'objet **markResponse** permet d'obtenir des informations sur la demande d'autorisation à 100 XPF .

markResponse	Format
amount Montant utilisé pour vérifier la validité de la carte, dans la plus petite unité monétaire .	n..12
currency Code de la monnaie utilisée pour vérifier la validité de la carte (suivant la norme ISO 4217) Ex : 953 pour le Franc CFP (XPF) .	n3
date Date et heure de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
number Numéro d'autorisation de la demande d'autorisation dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
result Résultat de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK . Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 50 : Objet markResponse

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

Cet objet se décompose :

- **authenticationRequestData**
Décrit le résultat de la demande d'inscription ainsi que le message encodé qui sera transmis par le navigateur de l'acheteur à l'ACS.
- **authenticationResultData**
Décrit les détails de l'authentification 3D Secure.

authenticationRequestData	Format
threeDSAcctId Certificat renvoyé par le Directory Server.	string
threeDSAcUrl Url de l'ACS à contacter.	string
threeDSBrand Réseau de la carte.	string
threeDSEncodedPareq Message PAREq encodé, prêt à envoyer à l'ACS.	string
threeDSEnrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none"> • Y pour un statut enrôlé. • N pour un statut non enrôlé. • U pour un statut inconnu. 	a1
threeDSRequestId Numéro de requête, à rappeler dans l'appel ENABLED_FINALIZE de l'attribut mode de l'objet threeDSRequest .	string

Tableau 51 : Objet authenticationRequestData

authenticationResultData	Format															
<div>transactionCondition Statut de l'authentification 3D Secure. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">COND_3D_SUCCESS Succès de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure et le porteur s'est authentifié correctement.COND_3D_FAILURE Echec de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas réussi à s'authentifier (mauvais mot de passe).COND_3D_ERROR Authentification en erreur. Le marchand participe au programme 3D Secure mais le serveur de la plateforme de paiement a rencontré un problème technique durant le processus d'authentification (lors de la vérification de l'inscription de la carte au programme 3D ou de l'authentification du porteur).COND_3D_NOTENROLLED Porteur non enrôlé. Le marchand participe au programme 3D Secure mais la carte du porteur n'est pas enrôlée.COND_3D_ATTEMPT Tentative d'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas eu à s'authentifier (le serveur de contrôle d'accès de la banque qui a émis la carte n'implémente que la génération d'une preuve de tentative d'authentification).COND_SSL 3D Secure non applicable. Le marchand n'est pas enrôlé à 3D Secure ou le canal de vente n'est pas couvert par cette garantie.</div>	string															
<div>enrolled Statut enrôlement du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut enrôlé.N pour un statut non enrôlé.U pour un statut inconnu.</div>	string															
<div>status Statut de l'authentification du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut authentifié 3 DS.N pour une erreur d'authentification.U pour une authentification impossible.A pour un essai d'authentification.</div>	a1															
<div>eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont :</div> <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>5</td><td>6</td><td>7</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>		status = Y	status = A	status = U	status = N	VISA - AMEX	5	6	7	-	MasterCard	02	01	-	-	string
	status = Y	status = A	status = U	status = N												
VISA - AMEX	5	6	7	-												
MasterCard	02	01	-	-												
<div>xid Numéro de transaction 3DS.</div>	string															
<div>cavvAlgorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">0 pour HMAC.1 pour CVV.</div>	n1															

authenticationResultData	Format
<ul style="list-style-type: none"> 2 pour CVV_ATN. 3 pour Mastercard SPA. 	
cavv Certificat de l'ACS.	string
signValid Signature de l'authentification 3DS.	string
brand Réseau de la carte.	string

Tableau 52 : Objet authenticationResultData

fraudManagementResponse

L'objet **fraudManagementResponse** permet d'obtenir les résultats des contrôles de gestion de risques.

La réponse se décompose en l'analyse des attributs :

- **riskControl**
Retourne le résultat du contrôle de gestion de risques effectué.
- **riskAnalysis**
Retourne le résultat de l'analyse de gestion de risques effectué par un système externe (Konduto, ClearSale, CyberSource,...).
- **riskAssessment**
Retourne le résultat de l'analyse de gestion des risques avancée effectuée par la plateforme de paiement.

L'attribut riskControl

Le format est le suivant : name1=result1;name2=result2

Nom	Format	Description
name	string	Nom de la règle de gestion de risque.
result	string	Résultat du contrôle.

Valeurs possibles pour 'name'	
CARD	Carte enregistrée en liste grise.
COUNTRY	Pays de l'acheteur enregistré en liste grise.
IPADDR	Adresse IP de l'acheteur enregistrée en liste grise.
AMOUNT	Montant maximum autorisé par commande est atteint.
BIN	Le code BIN de la carte est enregistré en liste grise.
ECB	Détection d'une e-carte bleue.
CARD_COMMERCIAL_NATIONAL	Détection d'une carte commerciale nationale.
CARD_COMMERCIAL_FOREIGN	Détection d'une carte commerciale étrangère.
CAS	Détection d'une carte à autorisation systématique.
COUNTRY_CONSISTENCY	Le pays d'origine de la carte, le pays de l'adresse IP de l'acheteur et le pays de l'acheteur ne correspondent pas.
NON_GUARANTEED_PAYMENT	Détection d'un paiement sans transfert de responsabilité.
IPADDR_COUNTRY	Le pays de l'adresse IP de l'acheteur est enregistré en liste grise.

Tableau 53 : Valeurs possibles pour 'name'

Valeurs possibles pour 'result'	
OK	Indique que le contrôle est correct.
KO	Indique que le contrôle est en erreur.

Tableau 54 : Valeurs possibles pour 'result'

L'attribut riskAnalysis

riskAnalysis	Format
score Score attribué à chaque transaction permettant d'évaluer le risque qui lui est associé.	string
resultCode Code renvoyé par un analyseur de risque externe. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour renvoyés par l'analyseur de risque externe .	string
status Statut de l'analyse de risque. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • P_SEND_OK, "Sent to clearsale and successfully processed" Succès • P_TO_SEND, "Transaction analysis is scheduled to be sent to risk analyzer" L'envoi est programmé • P_TO_SEND_KO, "Problem when tried to send to risk analyzer" Erreur de traitement • P_PENDING_AT_ANALYZER, "Analysis result is still being processed by the risk analyzer. We should keep checking/waiting for the analysis result" En cours de traitement par l'analyseur • P_MANUAL, "Analysis should be requested through user request (not automatically)" Attente d'envoi manuel • P_SKIPPED, "Analysis request discarded by current transaction status/problem" Ecarté • P_SEND_EXPIRED, "Analysis request expired" Expiré 	string
requestId Identifiant de l'analyse chez l'analyseur de risque.	string
extralInfo Pas de valorisation pour ClearSale. Pour CyberSource, cet attribut retourne tous les codes renvoyés par le DecisionManager. <ul style="list-style-type: none"> • key : nom de la donnée (son format est "string"). • value : valeur de la donnée (son format est "string"). Exemple : <ul style="list-style-type: none"> • COR-BA=Adresse de facturation corrigée ou corrigeable • A=Changements d'adresse excessifs. L'acheteur a changé d'adresse de facturation au moins deux fois au cours des six derniers mois. • etc. 	extInfo

Tableau 55 : Attribut riskAnalysis

L'attribut riskAssessment

Valeurs	Description
ENABLE_3DS	3D Secure activé.
DISABLE_3DS	3D Secure désactivé.
MANUAL_VALIDATION	La transaction est créée en validation manuelle. La remise du paiement est bloquée temporairement pour permettre au marchand de procéder à toutes les vérifications souhaitées.
REFUSE	La transaction est refusée.
RUN_RISK_ANALYSIS	Appel à un analyseur de risques externes sous condition que le marchand possède un contrat. Se référer à la description du champ vads_risk_analysis_result pour identifier la liste des valeurs possibles et leur description.
INFORM	Une alerte est remontée. Le marchand est averti qu'un risque est identifié.

Valeurs	Description
	Le marchand est informé via une ou plusieurs des règles du centre de notification (URL de notification, e-mail ou SMS).

extraResponse

L'objet **extraResponse** permet d'obtenir des informations supplémentaires à propos du paiement.

extraResponse	Format
paymentOptionCode Code de l'option de paiement utilisée lors du paiement.	string
paymentOptionOccNumb Nombre d'échéances lors d'un paiement en plusieurs fois. Par exemple, pour un paiement en trois fois, cet attribut sera valorisé à 3.	n..2

Tableau 56 : Objet extraResponse

6.3. Modifier les informations (panier) d'une transaction 'updatePaymentDetails'

updatePaymentDetails permet de modifier/mettre à jour les informations du panier.

updatePaymentDetails est uniquement disponible pour les transactions Klarna pour le moment, sous condition que celles-ci :

- possèdent un panier
- possèdent l'un des statuts suivants :
 - A valider et autoriser
 - A valider
 - A autoriser
 - En attente de remise
 - Remisé
- soient d'un montant global à la baisse ou constant (pas d'augmentation)

Requête à envoyer

La requête **updatePaymentDetails** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **updatePaymentDetails** prend en entrée un objet de type **updatePaymentDetails**.

Le type **updatePaymentDetails** est constitué des paramètres suivants :

Objet	Format	Requis
queryRequest	queryRequest	✓
shoppingCartRequest	shoppingCartRequest	✓

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

L'attribut à valoriser est le suivant :

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction. Remarque: cet attribut est utilisé pour remplacer les anciens attributs transactionId , sequenceNumber et creationDate . Cependant il est possible de continuer à utiliser ces anciens attributs pour des raisons de rétrocompatibilité. Pour plus d'informations, référez-vous au chapitre Gérer le rétrocompatibilité .	✓	string

Tableau 57 : Objet queryRequest

Les attributs **orderId**, **subscriptionId** et **paymentToken** ne sont pas aujourd'hui pris en considération pour cette opération.

shoppingCartRequest

L'objet **shoppingCartRequest** permet de transmettre le contenu du panier modifié.

Il doit contenir la totalité des attributs qui permettront de définir le panier définitif.

Remarque :

Si un attribut est supprimé dans la requête, il est supprimé définitivement. Pour pallier à toute erreur ou omission, tous les champs sont requis. Nous vous conseillons de les envoyer dans la requête même si l'un d'entre eux est valorisé à 0 ou vide.

shoppingCartRequest																																		
Attribut	Requis	Format																																
insuranceAmount Montant de l'assurance	✓	n..3																																
shippingAmount Frais d'expédition.	✓	n..3																																
cartItemInfo Champs personnalisables permettant d'ajouter les éléments du panier. L'attribut cartItemInfo est composé de sous objets : <ul style="list-style-type: none">productLabel : nom du produit. Son format est "string".productType : type de produit. Son format est "string (enum)". <table><tr><th>Valeur</th><th>Description</th></tr><tr><td>FOOD_AND_GROCERY</td><td>Produits alimentaires et d'épicerie</td></tr><tr><td>AUTOMOTIVE</td><td>Automobile / Moto</td></tr><tr><td>ENTERTAINMENT</td><td>Divertissement / Culture</td></tr><tr><td>HOME_AND_GARDEN</td><td>Maison et jardin</td></tr><tr><td>HOME_APPLIANCE</td><td>Equipement de la maison</td></tr><tr><td>AUCTION_AND_GROUP_BUYING</td><td>Ventes aux enchères et achats groupés</td></tr><tr><td>FLOWERS_AND_GIFTS </td><td>Fleurs et cadeaux</td></tr><tr><td>COMPUTER_AND_SOFTWARE</td><td>Ordinateurs et logiciels</td></tr><tr><td>HEALTH_AND_BEAUTY</td><td>Santé et beauté</td></tr><tr><td>SERVICE_FOR_INDIVIDUAL</td><td>Services à la personne</td></tr><tr><td>SERVICE_FOR_BUSINESS</td><td>Services aux entreprises</td></tr><tr><td>SPORTS</td><td>Sports</td></tr><tr><td>CLOTHING_AND_ACCESSORIES</td><td>Vêtements et accessoires</td></tr><tr><td>TRAVEL</td><td>Voyage</td></tr><tr><td>HOME_AUDIO_PHOTO_VIDEO</td><td>Son, image et vidéo</td></tr></table>	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	✓	cartItemInfo
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																																	

shoppingCartRequest			
Attribut		Requis	Format
Valeur	Description		
TELEPHONY	Téléphonie		

Tableau 59 : Valeurs associées à productType

- **productRef** : référence produit. Son format est "string".
- **productQty** : quantité de produit. Son format est "integer".
- **productAmount** : montant en centimes du produit. Son format est "string".
- **productVat** : montant de la taxe sur le produit. Son format est "string".

Exemple :

```
<cartItemInfo>
  <productLabel>CHIPS</productLabel>
  <productType>FOOD_AND_GROCERY</productType>
  <productRef>188545</productRef>
  <productQty>10</productQty>
  <productAmount>10000</productAmount>
</cartItemInfo>
```

Tableau 58 : Objet shoppingCartRequest

L'attribut **taxAmount** n'est pas aujourd'hui pris en considération pour cette opération.

Réponse en retour

La réponse à l'opération **updatePaymentDetails** est constituée d'un HEADER et d'un BODY de type **updatePaymentDetailsResponse**.

• HEADER

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

• BODY

La structure du message **updatePaymentDetailsResponse** est la suivante :

Nom	Type
updatePaymentDetailsResult	updatePaymentDetailsResult

La structure du message **updatePaymentDetailsResult** est la suivante :

Objet	Type
commonResponse	commonResponse
paymentResponse	paymentResponse
orderResponse	orderResponse
cardResponse	cardResponse
authorizationResponse	authorizationResponse
captureResponse	captureResponse
customerResponse	customerResponse
markResponse	markResponse
threeDSResponse	threeDSResponse
extraResponse	extraResponse
fraudManagementResponse	fraudManagementResponse
shoppingCartResponse	shoppingCartResponse

Remarque : l'objet **subscriptionResponse** est retourné mais n'est pas valorisé dans la réponse.

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none"> • La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction). • Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur. 	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none"> • AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue. • AUTHORISED_TO_VALIDATE A valider. La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la remise en banque. La transaction peut être validée tant que la date de remise n'est pas dépassée. Si cette date est dépassée, le paiement prend le statut Expiré (statut définitif). • WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement. • WAITING_AUTHORISATION_TO_VALIDATE A valider et à autoriser. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation sera déclenchée automatiquement à J-1 avant la date de remise en banque. Le paiement pourra être accepté ou refusé. La remise en banque est automatique. • CAPTURED La transaction est remise en banque. 	string
shopId Identifiant de la boutique.	n8
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none"> • EC pour le commerce électronique. • MOTO pour une commande par e-mail ou téléphone. • CC pour un centre d'appel. • OTHER pour un autre canal de vente. 	string (enum)

commonResponse		Format
submissionDate	Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40
contractNumber	Numéro de contrat commerçant utilisé.	string

Tableau 60 : Objet commonResponse

paymentResponse

L'objet **paymentResponse** détaille les informations sur la transaction.

paymentResponse	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none">• Soit cet identifiant est généré par la plateforme.• Soit cet identifiant est généré par le site marchand.	an..6
amount Montant de la transaction dans sa plus petite unité monétaire .	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	n3
effectiveAmount Montant de la transaction dans la devise réellement utilisée pour effectuer la remise en banque.	n..12
effectiveCurrency Devise réellement utilisée pour effectuer la remise en banque.	n3
expectedCaptureDate Date de remise en banque souhaitée. <u>Remarque</u> : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.	dateTime ans..40
operationType Type d'opération. Les valeurs possibles sont : <ul style="list-style-type: none">• 0 pour une opération de débit.• 1 pour une opération de remboursement.	n1
creationDate Date et heure de l'enregistrement de la transaction exprimée au format W3C.	dateTime ans..40
externalTransactionId Référence fournie par un tiers : numéro de transaction pour PayPal, Boletto, RRN pour Prism, etc...	string
liabilityShift Transfert de responsabilité. Les valeurs possibles sont : <ul style="list-style-type: none">• YES lorsque le paiement est garanti.• NO lorsque le paiement n'est pas garanti.	string
paymentType Type de paiement. Les valeurs possibles sont : <ul style="list-style-type: none">• SINGLE Paie ment en une seule fois.• INSTALLMENT Paie ment en plusieurs fois.• SPLIT Paie ment effectué avec plusieurs moyens de paiement.• SUBSCRIPTION Paie ment par alias ou lié à un abonnement.• RETRY Lors d'un paiement refusé, il est possible de réitérer la demande de paiement. Pour toute(s) réitération(s), le paiement est valorisé avec cette valeur. <u>Remarque</u> : Les valeurs INSTALLMENT et SPLIT peuvent être retournées uniquement si des paiements ont été créés via le formulaire de paiement.	string (enum)

paymentResponse	Format
sequenceNumber Numéro de séquence de la transaction. Sa valeur est fonction du contexte du paiement : <ul style="list-style-type: none"> Est valorisé à "1" pour un paiement unitaire. Est valorisé à "2" lors du rejeu d'un paiement initialement refusé. Prend la valeur du numéro d'échéance dans le cas d'un paiement en plusieurs fois créé à partir du formulaire de paiement. 	n..3
paymentError Complément d'information en cas d'erreur technique. Retourne un code d'erreur associé à l'erreur technique (voir chapitre Codes d'erreurs pouvant survenir durant le paiement).	n..3
nsu Apparaît lorsque isNsuRequested est valorisé à 1 dans la requête.	string

Tableau 61 : Objet paymentResponse

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
orderId Référence de la commande.	string an..64
extInfo Données personnalisées retournées en fonction des besoins. Exemple : extInfo.key, extInfo.value <ul style="list-style-type: none"> key : nom de la donnée (son format est "string"). value : valeur de la donnée (son format est "string"). 	extInfo

Tableau 62 : Objet orderResponse

cardResponse

L'objet **cardResponse** détaille les informations du moyen de paiement utilisé.

cardResponse	Format
number <ul style="list-style-type: none"> Numéro de carte masqué. Contient les 6 premiers chiffres du numéro, suivi par "XXXXXX" et enfin les 4 derniers chiffres. IBAN et BIC utilisés pour le paiement, séparés par un « _ » dans le cas d'un paiement par prélèvement. 	string
scheme Type de la carte.	string
brand Marque de la carte utilisée pour la transaction	string
country Code pays du pays d'émission de la carte (Code numérique ISO 3166).	ISO 3166
productCode Code produit de la carte.	an..3
bankCode Code banque de la banque émettrice.	n..5
expiryMonth Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).	n..2
expiryYear Année d'expiration sur 4 chiffres (ex : 2023).	n4

Tableau 63 : Objet cardResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
mode Spécifie de quelle manière est réalisée la demande d'autorisation. Deux valeurs possibles : <ul style="list-style-type: none"> • MARK Une autorisation de a été réalisée afin de vérifier la validité de la carte. Ce cas se présente lorsque la date de remise dépasse la période de validité d'une autorisation . • FULL Autorisation pour le montant total demandé dans la requête. 	string
amount Montant de l'autorisation dans la plus petite unité monétaire dans le cas où mode vaut FULL .	n..12
currency Code de la monnaie utilisée lors de la demande d'autorisation (suivant la norme ISO 4217) dans le cas où mode vaut FULL .	n3
date Date et heure de la demande d'autorisation dans le cas où mode vaut FULL .	dateTime ans..40
number Numéro de la demande d'autorisation dans le cas où mode vaut FULL .	n..2
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 64 : Objet *authorizationResponse*

captureResponse

L'objet **captureResponse** permet d'obtenir des informations à propos de la remise dans le cas où la transaction est remise.

captureResponse	Format
date Date et heure de remise.	dateTime ans..40
number Numéro de remise.	n3
reconciliationStatus Statut du rapprochement bancaire de la transaction.	n1
refundAmount Montant ayant déjà fait l'objet d'un remboursement dans sa plus petite unité monétaire.	n..12
refundCurrency Devise du montant ayant déjà fait l'objet d'un remboursement (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF) .	n3
chargeback Litige, impayé. Les valeurs possibles sont : <ul style="list-style-type: none"> • 0 : transaction ne faisant pas l'objet d'un litige. • 1 : transaction faisant l'objet d'un litige. 	

Tableau 65 : Objet *captureResponse*

customerResponse

L'objet **customerResponse** détaille de nombreuses informations à propos de l'acheteur.

La réponse se décompose en l'analyse de :

- **billingDetails**
Données de facturation de l'acheteur.
- **shippingDetails**
Données de livraison de l'acheteur. Les valeurs présentes dans cet objet seront identiques à celles de l'objet **billingDetails**.
- **extraDetails**

6.3.2.7.1. Objet billingDetails

billingDetails	
Attribut	Format
reference Référence de l'acheteur.	string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..	string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> • PRIVATE pour un particulier. • COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de l'acheteur.	string ans..255
district Quartier de l'acheteur.	string ans..127
zipCode Code postal de l'acheteur.	string ans..64
city Ville de l'acheteur.	string ans..128
state Etat/Région de l'acheteur.	string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none"> • DE pour l'Allemagne • IT pour l'Italie • PT pour le Portugal • GB pour le Royaume-Uni • JA pour le Japon • RU pour la Russie • ES pour l'Espagne • NL pour les Pays-Bas • SE pour la Suède • FR pour la France • PL pour la Pologne • CN pour la Chine 	string - a2
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> • de pour l'allemand • it pour l'italien • pt pour le portugais • en pour l'anglais • ja pour le japonais • ru pour le russe • es pour l'espagnol • nl pour le néerlandais • sv pour le suédois 	string - a2

billingDetails	
Attribut	Format
<ul style="list-style-type: none"> fr pour le français pl pour le polonais zh pour le chinois tr pour le turc 	
cellPhoneNumber Numéro de téléphone mobile	string ans..32
legalName Raison sociale de la société.	string ans..128

Tableau 66 : Objet billingDetails

6.3.2.7.2. Objet shippingDetails

shippingDetails	
Attribut	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de livraison.	string ans..255
address2 Complément d'adresse de livraison.	string ans..255
district Quartier de livraison.	string ans..127
zipCode Code postal de livraison.	string ans..64
city Ville de livraison.	string ans..128
state Etat/Région de livraison.	string ans..128
country Pays de livraison.	string - a2
deliveryCompanyName Informations sur le transporteur.	string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> STANDARD pour une livraison standard. EXPRESS pour une livraison express. 	string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont :	string (enum)

shippingDetails	
Attribut	Format
<ul style="list-style-type: none"> RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). ETICKET pour l'émission d'un billet électronique, téléchargement. 	
legalName Raison sociale de la société.	string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).	string ans..255

Tableau 67 : Objet shippingDetails

6.3.2.7.3. Objet extraDetails

extraDetails	Format
ipAddress L'adresse IP de l'acheteur.	string ans40

Tableau 68 : Objet extraDetails

markResponse

L'objet **markResponse** permet d'obtenir des informations sur la demande d'autorisation à 100 XPF .

markResponse	Format
amount Montant utilisé pour vérifier la validité de la carte, dans la plus petite unité monétaire .	n..12
currency Code de la monnaie utilisée pour vérifier la validité de la carte (suivant la norme ISO 4217) Ex : 953 pour le Franc CFP (XPF) .	n3
date Date et heure de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
number Numéro d'autorisation de la demande d'autorisation dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
result Résultat de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK . Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 69 : Objet markResponse

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

Cet objet se décompose :

- **authenticationRequestData**
Décrit le résultat de la demande d'inscription ainsi que le message encodé qui sera transmis par le navigateur de l'acheteur à l'ACS.
- **authenticationResultData**
Décrit les détails de l'authentification 3D Secure.

authenticationRequestData	Format
threeDSAcctId Certificat renvoyé par le Directory Server.	string
threeDSAcUrl Url de l'ACS à contacter.	string
threeDSBrand Réseau de la carte.	string
threeDSEncodedPareq Message PAREq encodé, prêt à envoyer à l'ACS.	string
threeDSEnrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none"> • Y pour un statut enrôlé. • N pour un statut non enrôlé. • U pour un statut inconnu. 	a1
threeDSRequestId Numéro de requête, à rappeler dans l'appel ENABLED_FINALIZE de l'attribut mode de l'objet threeDSRequest .	string

Tableau 70 : Objet authenticationRequestData

authenticationResultData	Format															
<p>transactionCondition Statut de l'authentification 3D Secure. Les valeurs possibles sont :</p> <ul style="list-style-type: none">• COND_3D_SUCCESS Succès de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure et le porteur s'est authentifié correctement.• COND_3D_FAILURE Echec de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas réussi à s'authentifier (mauvais mot de passe).• COND_3D_ERROR Authentification en erreur. Le marchand participe au programme 3D Secure mais le serveur de la plateforme de paiement a rencontré un problème technique durant le processus d'authentification (lors de la vérification de l'inscription de la carte au programme 3D ou de l'authentification du porteur).• COND_3D_NOTENROLLED Porteur non enrôlé. Le marchand participe au programme 3D Secure mais la carte du porteur n'est pas enrôlée.• COND_3D_ATTEMPT Tentative d'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas eu à s'authentifier (le serveur de contrôle d'accès de la banque qui a émis la carte n'implémente que la génération d'une preuve de tentative d'authentification).• COND_SSL 3D Secure non applicable. Le marchand n'est pas enrôlé à 3D Secure ou le canal de vente n'est pas couvert par cette garantie.	string															
<p>enrolled Statut enrôlement du porteur. Les valeurs possibles sont :</p> <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	string															
<p>status Statut de l'authentification du porteur. Les valeurs possibles sont :</p> <ul style="list-style-type: none">• Y pour un statut authentifié 3 DS.• N pour une erreur d'authentification.• U pour une authentification impossible.• A pour un essai d'authentification.	a1															
<p>eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont :</p> <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>5</td><td>6</td><td>7</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>		status = Y	status = A	status = U	status = N	VISA - AMEX	5	6	7	-	MasterCard	02	01	-	-	string
	status = Y	status = A	status = U	status = N												
VISA - AMEX	5	6	7	-												
MasterCard	02	01	-	-												
<p>xid Numéro de transaction 3DS.</p>	string															
<p>cavvAlgorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont :</p> <ul style="list-style-type: none">• 0 pour HMAC.• 1 pour CVV.	n1															

authenticationResultData	Format
<ul style="list-style-type: none"> 2 pour CVV_ATN. 3 pour Mastercard SPA. 	
cavv Certificat de l'ACS.	string
signValid Signature de l'authentification 3DS.	string
brand Réseau de la carte.	string

Tableau 71 : Objet authenticationResultData

fraudManagementResponse

L'objet **fraudManagementResponse** permet d'obtenir les résultats des contrôles de gestion de risques.

La réponse se décompose en l'analyse des attributs :

- riskControl**

Retourne le résultat du contrôle de gestion de risques effectué.

- riskAnalysis**

Retourne le résultat de l'analyse de gestion de risques effectué par un système externe (Konduto, ClearSale, CyberSource,...).

- riskAssessment**

Retourne le résultat de l'analyse de gestion des risques avancée effectuée par la plateforme de paiement.

L'attribut riskControl

Le format est le suivant : name1=result1;name2=result2

Nom	Format	Description
name	string	Nom de la règle de gestion de risque.
result	string	Résultat du contrôle.

Valeurs possibles pour 'name'	
CARD	Carte enregistrée en liste grise.
COUNTRY	Pays de l'acheteur enregistré en liste grise.
IPADDR	Adresse IP de l'acheteur enregistrée en liste grise.
AMOUNT	Montant maximum autorisé par commande est atteint.
BIN	Le code BIN de la carte est enregistré en liste grise.
ECB	Détection d'une e-carte bleue.
CARD_COMMERCIAL_NATIONAL	Détection d'une carte commerciale nationale.
CARD_COMMERCIAL_FOREIGN	Détection d'une carte commerciale étrangère.
CAS	Détection d'une carte à autorisation systématique.
COUNTRY_CONSISTENCY	Le pays d'origine de la carte, le pays de l'adresse IP de l'acheteur et le pays de l'acheteur ne correspondent pas.
NON_GUARANTEED_PAYMENT	Détection d'un paiement sans transfert de responsabilité.
IPADDR_COUNTRY	Le pays de l'adresse IP de l'acheteur est enregistré en liste grise.

Tableau 72 : Valeurs possibles pour 'name'

Valeurs possibles pour 'result'	
OK	Indique que le contrôle est correct.
KO	Indique que le contrôle est en erreur.

Tableau 73 : Valeurs possibles pour 'result'

L'attribut riskAnalysis

riskAnalysis	Format
score Score attribué à chaque transaction permettant d'évaluer le risque qui lui est associé.	string
resultCode Code renvoyé par un analyseur de risque externe. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour renvoyés par l'analyseur de risque externe .	string
status Statut de l'analyse de risque. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • P_SEND_OK, "Sent to clearsale and successfully processed" Succès • P_TO_SEND, "Transaction analysis is scheduled to be sent to risk analyzer" L'envoi est programmé • P_TO_SEND_KO, "Problem when tried to send to risk analyzer" Erreur de traitement • P_PENDING_AT_ANALYZER, "Analysis result is still being processed by the risk analyzer. We should keep checking/waiting for the analysis result" En cours de traitement par l'analyseur • P_MANUAL, "Analysis should be requested through user request (not automatically)" Attente d'envoi manuel • P_SKIPPED, "Analysis request discarded by current transaction status/problem" Ecarté • P_SEND_EXPIRED, "Analysis request expired" Expiré 	string
requestId Identifiant de l'analyse chez l'analyseur de risque.	string
extralInfo Pas de valorisation pour ClearSale. Pour CyberSource, cet attribut retourne tous les codes renvoyés par le DecisionManager. <ul style="list-style-type: none"> • key : nom de la donnée (son format est "string"). • value : valeur de la donnée (son format est "string"). Exemple : <ul style="list-style-type: none"> • COR-BA=Adresse de facturation corrigée ou corrigeable • A=Changements d'adresse excessifs. L'acheteur a changé d'adresse de facturation au moins deux fois au cours des six derniers mois. • etc. 	extInfo

Tableau 74 : Attribut riskAnalysis

L'attribut riskAssessment

Valeurs	Description
ENABLE_3DS	3D Secure activé.
DISABLE_3DS	3D Secure désactivé.
MANUAL_VALIDATION	La transaction est créée en validation manuelle. La remise du paiement est bloquée temporairement pour permettre au marchand de procéder à toutes les vérifications souhaitées.
REFUSE	La transaction est refusée.
RUN_RISK_ANALYSIS	Appel à un analyseur de risques externes sous condition que le marchand possède un contrat. Se référer à la description du champ vads_risk_analysis_result pour identifier la liste des valeurs possibles et leur description.
INFORM	Une alerte est remontée. Le marchand est averti qu'un risque est identifié.

Valeurs	Description
	Le marchand est informé via une ou plusieurs des règles du centre de notification (URL de notification, e-mail ou SMS).

extraResponse

L'objet **extraResponse** permet d'obtenir des informations supplémentaires à propos du paiement.

extraResponse	Format
paymentOptionCode Code de l'option de paiement utilisée lors du paiement.	string
paymentOptionOccNumb Nombre d'échéances lors d'un paiement en plusieurs fois. Par exemple, pour un paiement en trois fois, cet attribut sera valorisé à 3.	n..2

Tableau 75 : Objet extraResponse

shoppingCartResponse

L'objet **shoppingCartResponse** détaille le contenu du panier.

shoppingCartResponse		
Attribut		Format
cartItemInfo Champs personnalisables permettant d'ajouter les éléments du panier. L'attribut cartItemInfo est composé de sous objets :		cartItemInfo
<ul style="list-style-type: none">productLabel : nom du produit. Son format est "string".productType : type de produit. Son format est "string (enum)".		
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	
Tableau 76 : Valeurs associées à productType		
<ul style="list-style-type: none">productRef : référence produit. Son format est "string".productQty : quantité de produit. Son format est "integer".productAmount : montant en centimes du produit. Son format est "string".productVat : montant de la taxe sur le produit. Son format est "string".		
Exemple :		
<pre><cartItemInfo> <productLabel>CHIPS</productLabel> <productType>FOOD_AND_GROCERY</productType> <productRef>188545</productRef> <productQty>10</productQty> <productAmount>10000</productAmount> </cartItemInfo></pre>		

6.4. Annuler une transaction de paiement 'cancelPayment'

Annuler une transaction de paiement est réalisable grâce l'appel de l'opération **cancelPayment**.

cancelPayment permet d'annuler définitivement une transaction, non encore remisee, disposant d'un des statuts suivants :

- A valider
- A valider et autoriser
- En attente d'autorisation
- En attente de remise

Requête à envoyer

La requête **cancelPayment** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **cancelPayment** prend en entrée un objet de type **cancelPayment**.

Le type **cancelPayment** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Un seul attribut peut être valorisé si besoin :

commonRequest		
Attribut	Requis	Format
comment Commentaire libre.		string

Tableau 77 : Objet commonRequest

Le commentaire sera affiché dans l'historique de la transaction visible depuis le Back Office.

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

L'attribut à valoriser est le suivant :

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction. Remarque: cet attribut est utilisé pour remplacer les anciens attributs transactionId , sequenceNumber et creationDate . Cependant il est possible de continuer à utiliser ces anciens attributs pour des raisons de retrocompatibilité. Pour plus d'informations, référez-vous au chapitre Gérer le rétrocompatibilité .	✓	string

Tableau 78 : Objet queryRequest

Réponse en retour

La réponse à l'opération **cancelPayment** est faite par la plateforme de paiement suite à une demande d'annulation d'un paiement.

Elle est constituée d'un HEADER et d'un BODY de type **cancelPaymentResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **cancelPaymentResponse** est la suivante :

Nom	Type
cancelPaymentResult	cancelPaymentResult

La structure du message **cancelPaymentResult** est la suivante :

Objet	Type
commonResponse	commonResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• CANCELLED Annulée. La transaction est annulée par le marchand.	string
shopId Identifiant de la boutique.	n8
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• EC pour le commerce électronique.• MOTO pour une commande par e-mail ou téléphone.	string (enum)

commonResponse	Format
<ul style="list-style-type: none"> • CC pour un centre d'appel. • OTHER pour un autre canal de vente. 	
submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 79 : Objet commonResponse

6.5. Rechercher des paiements 'findPayments'

Rechercher des paiements est réalisable grâce l'appel de l'opération **findPayments**.

findPayments permet d'obtenir la liste de paiements qui correspondent aux critères de recherches saisis.

Requête à envoyer

L'opération **findPayments** est utilisée pour rechercher un ou plusieurs paiements.

La requête **findPayments** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **findPayments** prend en entrée un objet de type **findPayments**.

Le type **findPayments** est constitué du paramètre suivant :

Objet	Format	Requis
queryRequest	queryRequest	✓

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

Pour l'opération **findPayments**, seul l'attribut **orderId** doit être valorisé afin de trouver une liste de paiements.

queryRequest		
Attribut	Requis	Format
orderId Référence de la commande.	✓	ans..64

Tableau 80 : Objet queryRequest

Les attributs **paymentToken**, **subscriptionId** et **uuid** ne sont pas aujourd'hui pris en considération pour cette opération.

Exemple de requête à envoyer

Exemple de requête à envoyer :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:v5="http://v5.ws.vads.lyra.com/">
  <soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">
    <soapHeader:shopId>12345678</soapHeader:shopId>
    <soapHeader:requestId>1d37acc0-c5a7-4e32-b3df-168b9e2617e0</soapHeader:requestId>
    <soapHeader:timestamp>2015-04-01T12:21:09Z</soapHeader:timestamp>
    <soapHeader:mode>TEST</soapHeader:mode>
    <soapHeader:authToken>WFBz7scW8n/xYro5Od3iTUyFhr0Jw6Y4z1EhX71fR6U=</soapHeader:authToken>
  </soap:Header>
  <soap:Body>
    <v5:findPayments>
      <queryRequest>
        <orderId>TEST-01</orderId>
      </queryRequest>
    </v5:findPayments>
  </soap:Body>
</soap:Envelope>
```

Réponse en retour

La réponse à l'opération **findPayments** est faite par la plateforme de paiement suite à une recherche de un ou plusieurs paiements.

Elle est constituée d'un HEADER et d'un BODY de type **findPaymentsResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **findPaymentsResponse** est la suivante :

Nom	Type
findPaymentsResult	findPaymentsResult

La structure du message **findPaymentsResult** est la suivante :

Objet	Type
commonResponse	commonResponse
orderResponse	orderResponse
transactionItem	transactionItem

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

Les attributs contenant une valeur dans la réponse sont les suivants :

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
shopId Identifiant de la boutique.	n8

Tableau 81 : Objet commonResponse

Les attributs **transactionStatusLabel**, **paymentSource**, **submissionDate**, **contractNumber** et **paymentToken** ne sont pas valorisés dans la réponse.

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
orderId Référence de la commande.	string an..64
extInfo Données personnalisées retournées en fonction des besoins. <u>Exemple</u> : extInfo.key, extInfo.value <ul style="list-style-type: none">• key : nom de la donnée (son format est "string").• value : valeur de la donnée (son format est "string").	extInfo

Tableau 82 : Objet orderResponse

transactionItem

L'objet **transactionItem** détaille les informations de la transaction recherchée pour laquelle vous souhaitez des informations.

transactionItem	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• INITIAL En traitement. Ce statut est temporaire. Le statut définitif sera retourné aussitôt la synchronisation réalisée.• AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue.• AUTHORISED_TO_VALIDATE A valider. La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la remise en banque. La transaction peut être validée tant que la date de remise n'est pas dépassée. Si cette date est dépassée, le paiement prend le statut Expiré (statut définitif).• WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement.• WAITING_AUTHORISATION_TO_VALIDATE A valider et à autoriser. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation sera déclenchée automatiquement à J-1 avant la date de remise en banque. Le paiement pourra être accepté ou refusé. La remise en banque est automatique.• REFUSED Refusée. La transaction est refusée.• CAPTURED La transaction est remise en banque.• CANCELLED Annulée. La transaction est annulée par le marchand.• EXPIRED Expirée.	string

transactionItem	Format
<p>La date de remise est atteinte mais le marchand n'a pas validé la transaction.</p> <ul style="list-style-type: none"> UNDER_VERIFICATION (Spécifique à PayPal) En attente de vérification par PayPal. Cette valeur signifie que PayPal retient la transaction pour suspicion de fraude . Le paiement est alors dans l'onglet Paiement en cours. 	
<p>amount Montant de la transaction dans sa plus petite unité monétaire . <u>Remarque :</u> Ne doit pas être envoyé à vide ou être à 0. Si vous ne souhaitez pas modifier le statut de la transaction, vous devez valoriser l'attribut amount avec la valeur initiale. Si aucune information n'est modifiée, la requête sera rejetée avec un code erreur.</p>	n..12
<p>currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)</p>	n3
<p>expectedCaptureDate Date de remise en banque souhaitée. <u>Remarque :</u> si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.</p>	dateTime ans..40

Tableau 83 : Objet transactionItem

Exemple de réponse

Exemple de réponse :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">1d37acc0-c5a7-4e32-b3df-168b9e2617e0</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T12:21:09Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">m2+EUcE2+40LrPe/zpmo0W9BXqxAnTsA77OdesXCkiY=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:findPaymentsResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <findPaymentsResult>
        <requestId>1d37acc0-c5a7-4e32-b3df-168b9e2617e0</requestId>
        <commonResponse>
          <responseCode>0</responseCode>
          <responseCodeDetail>Action successfully completed</responseCodeDetail>
          <shopId>12345678</shopId>
        </commonResponse>
        <orderResponse>
          <orderId>TEST-01</orderId>
        </orderResponse>
        <transactionItem>
          <transactionUuid>a27d1907d7f74be1843318dce5875b99</transactionUuid>
          <transactionStatusLabel>AUTHORISED</transactionStatusLabel>
          <amount>1</amount>
          <currency>953</currency>
          <expectedCaptureDate>2015-04-01T14:07:54+02:00</expectedCaptureDate>
        </transactionItem>
        <transactionItem>
          <transactionUuid>170fd39a02c847feb5a5f750e8b320d2</transactionUuid>
          <transactionStatusLabel>AUTHORISED_TO_VALIDATE</transactionStatusLabel>
          <amount>1</amount>
          <currency>953</currency>
          <expectedCaptureDate>2015-04-01T14:18:58+02:00</expectedCaptureDate>
        </transactionItem>
      </findPaymentsResult>
    </ns2:findPaymentsResponse>
  </soap:Body>
</soap:Envelope>
```

6.6. Rembourser un acheteur 'refundPayment'

L'opération **refundPayment** permet de rembourser un acheteur.

Les transactions pouvant faire l'objet d'un remboursement ont le statut **Remisé**.

Requête à envoyer

L'opération **refundPayment** est utilisée pour effectuer un remboursement sur une transaction dont le statut est **Remisé**.

La requête **refundPayment** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **refundPayment** prend en entrée un objet de type **refundPayment**.

Le type **refundPayment** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
paymentRequest	paymentRequest	✓
queryRequest	queryRequest	✓

Remarque :

Si la carte est expirée lors de la demande de remboursement, une transaction refusée pour motif carte expirée sera créée.

La réponse contiendra les valeurs suivantes :

- **responseCode** : 0
- **transactionStatusLabel** : **REFUSED**

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Un seul attribut peut être valorisé si besoin :

commonRequest		
Attribut	Requis	Format
comment Commentaire libre.		string

Tableau 84 : Objet commonRequest

Le commentaire sera affiché dans l'historique de la transaction visible depuis le Back Office.

paymentRequest

L'objet **paymentRequest** permet de transmettre des informations liées au paiement.

Il possède les attributs suivants :

paymentRequest		
Attribut	Requis	Format
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none">• <u>Soit cet identifiant est généré par la plateforme.</u> Dans ce cas, ce paramètre ne doit pas être renseigné.• <u>Soit cet identifiant est généré par le site marchand.</u> Dans ce cas, ce paramètre doit être renseigné avec la valeur de l'identifiant souhaité. Attention, il incombe au site marchand de s'assurer de l'unicité des identifiants. Toute demande d'enregistrement contenant un identifiant déjà existant, sera rejetée, et retournera un code d'erreur 12. Remarque : cet attribut ne peut être envoyé à vide.		an..6
amount Montant de la transaction dans sa plus petite unité monétaire . Remarque : <ul style="list-style-type: none">• Ne doit pas être envoyé à vide ou être à 0.• Ne doit pas être supérieur au montant initial (cas du remboursement).	✓	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	✓	n3
expectedCaptureDate Date de remise demandée exprimée au format ISO 8601 défini par W3C. <i>Exemple</i> : 2016-07-16T19:20:00Z. Ce paramètre est utilisé pour effectuer un paiement différé. Si le nombre de jours entre la date de remise demandée et la date actuelle est supérieur à la durée de validité de l'autorisation, une autorisation de 100 XPF sera réalisée le jour de la transaction. Ceci afin de vérifier la validité de la carte. L'autorisation pour le montant total sera effectuée : <ul style="list-style-type: none">• fonctionnement par défaut : le jour de la date de remise en banque souhaitée,• fonctionnement avec autorisation anticipée : selon le moyen de paiement sélectionné, à J- le nombre de jours correspondant à la durée de validité d'une autorisation avant la date de remise en banque souhaitée. Si vous souhaitez être notifié du résultat de cette demande d'autorisation, vous devez configurer la règle de notification URL de notification sur autorisation par Batch dans le Back Office (Paramétrage > Règles de notifications). Remarque : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.		dateTime ans..40
manualValidation Permet de valider manuellement une transaction tant que la date de remise en banque souhaitée n'est pas dépassée. Pour cela, cet attribut doit être valorisé à 1 (validation manuelle). Valorisé à 0 , la validation sera automatique.		n1

Tableau 85 : Objet paymentRequest

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

L'attribut à valoriser est le suivant :

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction.	✓	string

queryRequest		
Attribut	Requis	Format
Remarque: cet attribut est utilisé pour remplacer les anciens attributs transactionId , sequenceNumber et creationDate . Cependant il est possible de continuer à utiliser ces anciens attributs pour des raisons de rétrocompatibilité. Pour plus d'informations, référez-vous au chapitre Gérer le rétrocompatibilité .		

Tableau 86 : Objet queryRequest

Réponse en retour

La réponse à l'opération **refundPayment** est faite par la plateforme de paiement suite à une demande de remboursement.

Elle est constituée d'un HEADER et d'un BODY de type **refundPaymentResponse**.

- HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- BODY**

La structure du message **refundPaymentResponse** est la suivante :

Nom	Type
refundPaymentResult	refundPaymentResult

La structure du message **refundPaymentResult** est la suivante :

Objet	Type
commonResponse	commonResponse
paymentResponse	paymentResponse
orderResponse	orderResponse
cardResponse	cardResponse
authorizationResponse	authorizationResponse
captureResponse	captureResponse
customerResponse	customerResponse
markResponse	markResponse
threeDSResponse	threeDSResponse
extraResponse	extraResponse
fraudManagementResponse	fraudManagementResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">INITIAL En traitement. Ce statut est temporaire. Le statut définitif sera retourné aussitôt la synchronisation réalisée.NOT_CREATED La transaction n'est pas créée et n'est pas visible dans le Back Office.AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue.AUTHORISED_TO_VALIDATE A valider. La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la remise en banque. La transaction peut être validée tant que la date de remise n'est pas dépassée. Si cette date est dépassée, le paiement prend le statut Expiré (statut définitif).WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement.WAITING_AUTHORISATION_TO_VALIDATE A valider et à autoriser. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation sera déclenchée automatiquement à J-1 avant la date de remise en banque. Le paiement pourra être accepté ou refusé. La remise en banque est automatique.REFUSED Refusée. La transaction est refusée.	string
shopId Identifiant de la boutique.	n8
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">EC pour le commerce électronique.MOTO pour une commande par e-mail ou téléphone.CC pour un centre d'appel.OTHER pour un autre canal de vente.	string (enum)
submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40

commonResponse	Format
contractNumber Numéro de contrat commerçant utilisé.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 87 : Objet commonResponse

paymentResponse

L'objet **paymentResponse** détaille les informations sur la transaction.

paymentResponse	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none"> • Soit cet identifiant est généré par la plateforme. • Soit cet identifiant est généré par le site marchand. 	an..6
amount Montant de la transaction dans sa plus petite unité monétaire .	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	n3
effectiveAmount Montant de la transaction dans la devise réellement utilisée pour effectuer la remise en banque.	n..12
effectiveCurrency Devise réellement utilisée pour effectuer la remise en banque.	n3
expectedCaptureDate Date de remise en banque souhaitée. Remarque : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.	dateTime ans..40
operationType Type d'opération. Les valeurs possibles sont : <ul style="list-style-type: none"> • 0 pour une opération de débit. • 1 pour une opération de remboursement. 	n1
creationDate Date et heure de l'enregistrement de la transaction exprimée au format W3C.	dateTime ans..40
externalTransactionId Référence fournie par un tiers : numéro de transaction pour PayPal, Boletto, RRN pour Prism, etc...	string
liabilityShift Transfert de responsabilité. Les valeurs possibles sont : <ul style="list-style-type: none"> • YES lorsque le paiement est garanti. • NO lorsque le paiement n'est pas garanti. 	string
paymentType Type de paiement. Les valeurs possibles sont : <ul style="list-style-type: none"> • SINGLE Paie ment en une seule fois. • INSTALLMENT Paie ment en plusieurs fois. • SPLIT Paie ment effectué avec plusieurs moyens de paiement. • SUBSCRIPTION 	string (enum)

paymentResponse	Format
<p>Païement par alias ou lié à un abonnement.</p> <ul style="list-style-type: none"> RETRY Lors d'un paiement refusé, il est possible de réitérer la demande de paiement. Pour toute(s) réitération(s), le paiement est valorisé avec cette valeur. <p><i>Remarque :</i> Les valeurs INSTALLMENT et SPLIT peuvent être retournées uniquement si des paiements ont été créés via le formulaire de paiement.</p> 	
<p>sequenceNumber Numéro de séquence de la transaction. Sa valeur est fonction du contexte du paiement :</p> <ul style="list-style-type: none"> Est valorisé à "1" pour un paiement unitaire. Est valorisé à "2" lors du rejeu d'un paiement initialement refusé. Prend la valeur du numéro d'échéance dans le cas d'un paiement en plusieurs fois créé à partir du formulaire de paiement. 	n..3
<p>paymentError Complément d'information en cas d'erreur technique. Retourne un code d'erreur associé à l'erreur technique (voir chapitre Codes d'erreurs pouvant survenir durant le paiement).</p>	n..3
<p>nsu Apparaît lorsque isNsuRequested est valorisé à 1 dans la requête.</p>	string

Tableau 88 : Objet paymentResponse

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
<p>orderId Référence de la commande.</p>	string an..64
<p>extInfo Données personnalisées retournées en fonction des besoins. <i>Exemple</i> : extInfo.key, extInfo.value</p> <ul style="list-style-type: none"> key : nom de la donnée (son format est "string"). value : valeur de la donnée (son format est "string"). 	extInfo

Tableau 89 : Objet orderResponse

cardResponse

L'objet **cardResponse** détaille les informations du moyen de paiement utilisé.

cardResponse	Format
<p>number</p> <ul style="list-style-type: none"> Numéro de carte masqué. Contient les 6 premiers chiffres du numéro, suivi par "XXXXXX" et enfin les 4 derniers chiffres. IBAN et BIC utilisés pour le paiement, séparés par un « _ » dans le cas d'un paiement par prélèvement. 	string
<p>scheme Type de la carte.</p>	string
<p>brand Marque de la carte utilisée pour la transaction</p>	string
<p>country Code pays du pays d'émission de la carte (Code numérique ISO 3166).</p>	ISO 3166
<p>productCode Code produit de la carte.</p>	an..3
<p>bankCode Code banque de la banque émettrice.</p>	n..5
<p>expiryMonth Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).</p>	n..2
<p>expiryYear</p>	n4

cardResponse	Format
Année d'expiration sur 4 chiffres (ex : 2023).	

Tableau 90 : Objet cardResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
mode Spécifie de quelle manière est réalisée la demande d'autorisation. Deux valeurs possibles : <ul style="list-style-type: none">• MARK Une autorisation de a été réalisée afin de vérifier la validité de la carte. Ce cas se présente lorsque la date de remise dépasse la période de validité d'une autorisation .• FULL Autorisation pour le montant total demandé dans la requête.	string
amount Montant de l'autorisation dans la plus petite unité monétaire dans le cas où mode vaut FULL .	n..12
currency Code de la monnaie utilisée lors de la demande d'autorisation (suivant la norme ISO 4217) dans le cas où mode vaut FULL .	n3
date Date et heure de la demande d'autorisation dans le cas où mode vaut FULL .	dateTime ans..40
number Numéro de la demande d'autorisation dans le cas où mode vaut FULL .	n..2
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 91 : Objet authorizationResponse

captureResponse

L'objet **captureResponse** permet d'obtenir des informations à propos de la remise dans le cas où la transaction est remise.

captureResponse	Format
date Date et heure de remise.	dateTime ans..40
number Numéro de remise.	n3
reconciliationStatus Statut du rapprochement bancaire de la transaction.	n1
refundAmount Montant ayant déjà fait l'objet d'un remboursement dans sa plus petite unité monétaire.	n..12
refundCurrency Devise du montant ayant déjà fait l'objet d'un remboursement (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF) .	n3
chargeback Litige, impayé. Les valeurs possibles sont : <ul style="list-style-type: none">• 0 : transaction ne faisant pas l'objet d'un litige.• 1 : transaction faisant l'objet d'un litige.	

Tableau 92 : Objet captureResponse

customerResponse

L'objet **customerResponse** détaille de nombreuses informations à propos de l'acheteur.

La réponse se décompose en l'analyse de :

- **billingDetails**

Données de facturation de l'acheteur.

- **shippingDetails**

Données de livraison de l'acheteur. Les valeurs présentes dans cet objet seront identiques à celles de l'objet **billingDetails**.

- **extraDetails**

Données techniques liées à l'acheteur.

6.6.2.7.1. Objet billingDetails

billingDetails	
Attribut	Format
reference Référence de l'acheteur.	string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..	string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none">• PRIVATE pour un particulier.• COMPANY pour une entreprise.	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de l'acheteur.	string ans..255
district Quartier de l'acheteur.	string ans..127
zipCode Code postal de l'acheteur.	string ans..64
city Ville de l'acheteur.	string ans..128
state Etat/Région de l'acheteur.	string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none">• DE pour l'Allemagne• IT pour l'Italie• PT pour le Portugal• GB pour le Royaume-Uni• JA pour le Japon• RU pour la Russie	string - a2

billingDetails	
Attribut	Format
<ul style="list-style-type: none"> ES pour l'Espagne FR pour la France NL pour les Pays-Bas PL pour la Pologne SE pour la Suède CN pour la Chine 	
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> de pour l'allemand en pour l'anglais es pour l'espagnol fr pour le français tr pour le turc it pour l'italien ja pour le japonais nl pour le néerlandais pl pour le polonais pt pour le portugais ru pour le russe sv pour le suédois zh pour le chinois 	string - a2
cellPhoneNumber Numéro de téléphone mobile	string ans..32
legalName Raison sociale de la société.	string ans..128

Tableau 93 : Objet billingDetails

6.6.2.7.2. Objet shippingDetails

shippingDetails	
Attribut	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de livraison.	string ans..255
address2 Complément d'adresse de livraison.	string ans..255
district Quartier de livraison.	string ans..127
zipCode Code postal de livraison.	string ans..64
city Ville de livraison.	string ans..128
state Etat/Région de livraison.	string ans..128
country	string - a2

shippingDetails	
Attribut	Format
Pays de livraison.	
deliveryCompanyName Informations sur le transporteur.	string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> • STANDARD pour une livraison standard. • EXPRESS pour une livraison express. 	string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont : <ul style="list-style-type: none"> • RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. • RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). • RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. • PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). • ETICKET pour l'émission d'un billet électronique, téléchargement. 	string (enum)
legalName Raison sociale de la société.	string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).	string ans..255

Tableau 94 : Objet shippingDetails

6.6.2.7.3. Objet extraDetails

extraDetails	Format
ipAddress L'adresse IP de l'acheteur.	string ans40

Tableau 95 : Objet extraDetails

markResponse

L'objet **markResponse** permet d'obtenir des informations sur la demande d'autorisation à 100 XPF .

markResponse	Format
amount Montant utilisé pour vérifier la validité de la carte, dans la plus petite unité monétaire .	n..12
currency Code de la monnaie utilisée pour vérifier la validité de la carte (suivant la norme ISO 4217) Ex : 953 pour le Franc CFP (XPF) .	n3
date Date et heure de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
number Numéro d'autorisation de la demande d'autorisation dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
result Résultat de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK . Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 96 : Objet markResponse

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

Cet objet se décompose :

- **authenticationRequestData**

Décrit le résultat de la demande d'enrôlement ainsi que le message encodé qui sera transmis par le navigateur de l'acheteur à l'ACS.

- **authenticationResultData**

Décrit les détails de l'authentification 3D Secure.

authenticationRequestData	Format
threeDSAcctId Certificat renvoyé par le Directory Server.	string
threeDSAcUrl Url de l'ACS à contacter.	string
threeDSBrand Réseau de la carte.	string
threeDSEncodedPareq Message PAREq encodé, prêt à envoyer à l'ACS.	string
threeDSEnrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	a1
threeDSRequestId Numéro de requête, à rappeler dans l'appel ENABLED_FINALIZE de l'attribut mode de l'objet threeDSRequest .	string

Tableau 97 : Objet authenticationRequestData

authenticationResultData	Format															
<div>transactionCondition Statut de l'authentification 3D Secure. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">COND_3D_SUCCESS Succès de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure et le porteur s'est authentifié correctement.COND_3D_FAILURE Echec de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas réussi à s'authentifier (mauvais mot de passe).COND_3D_ERROR Authentification en erreur. Le marchand participe au programme 3D Secure mais le serveur de la plateforme de paiement a rencontré un problème technique durant le processus d'authentification (lors de la vérification de l'inscription de la carte au programme 3D ou de l'authentification du porteur).COND_3D_NOTENROLLED Porteur non enrôlé. Le marchand participe au programme 3D Secure mais la carte du porteur n'est pas enrôlée.COND_3D_ATTEMPT Tentative d'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas eu à s'authentifier (le serveur de contrôle d'accès de la banque qui a émis la carte n'implémente que la génération d'une preuve de tentative d'authentification).COND_SSL 3D Secure non applicable. Le marchand n'est pas enrôlé à 3D Secure ou le canal de vente n'est pas couvert par cette garantie.</div>	string															
<div>enrolled Statut enrôlement du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut enrôlé.N pour un statut non enrôlé.U pour un statut inconnu.</div>	string															
<div>status Statut de l'authentification du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut authentifié 3 DS.N pour une erreur d'authentification.U pour une authentification impossible.A pour un essai d'authentification.</div>	a1															
<div>eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont :</div> <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>5</td><td>6</td><td>7</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>		status = Y	status = A	status = U	status = N	VISA - AMEX	5	6	7	-	MasterCard	02	01	-	-	string
	status = Y	status = A	status = U	status = N												
VISA - AMEX	5	6	7	-												
MasterCard	02	01	-	-												
<div>xid Numéro de transaction 3DS.</div>	string															
<div>cavvAlgorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">0 pour HMAC.1 pour CVV.</div>	n1															

authenticationResultData	Format
<ul style="list-style-type: none"> 2 pour CVV_ATN. 3 pour Mastercard SPA. 	
cavv Certificat de l'ACS.	string
signValid Signature de l'authentification 3DS.	string
brand Réseau de la carte.	string

Tableau 98 : Objet authenticationResultData

fraudManagementResponse

L'objet **fraudManagementResponse** permet d'obtenir les résultats des contrôles de gestion de risques.

La réponse se décompose en l'analyse des attributs :

- riskControl**

Retourne le résultat du contrôle de gestion de risques effectué.

- riskAnalysis**

Retourne le résultat de l'analyse de gestion de risques effectué par un système externe (Konduto, ClearSale, CyberSource,...).

- riskAssessment**

Retourne le résultat de l'analyse de gestion des risques avancée effectuée par la plateforme de paiement.

L'attribut riskControl

Le format est le suivant : name1=result1;name2=result2

Nom	Format	Description
name	string	Nom de la règle de gestion de risque.
result	string	Résultat du contrôle.

Valeurs possibles pour 'name'	
CARD	Carte enregistrée en liste grise.
COUNTRY	Pays de l'acheteur enregistré en liste grise.
IPADDR	Adresse IP de l'acheteur enregistrée en liste grise.
AMOUNT	Montant maximum autorisé par commande est atteint.
BIN	Le code BIN de la carte est enregistré en liste grise.
ECB	Détection d'une e-carte bleue.
CARD_COMMERCIAL_NATIONAL	Détection d'une carte commerciale nationale.
CARD_COMMERCIAL_FOREIGN	Détection d'une carte commerciale étrangère.
CAS	Détection d'une carte à autorisation systématique.
COUNTRY_CONSISTENCY	Le pays d'origine de la carte, le pays de l'adresse IP de l'acheteur et le pays de l'acheteur ne correspondent pas.
NON_GUARANTEED_PAYMENT	Détection d'un paiement sans transfert de responsabilité.
IPADDR_COUNTRY	Le pays de l'adresse IP de l'acheteur est enregistré en liste grise.

Tableau 99 : Valeurs possibles pour 'name'

Valeurs possibles pour 'result'	
OK	Indique que le contrôle est correct.
KO	Indique que le contrôle est en erreur.

Tableau 100 : Valeurs possibles pour 'result'

L'attribut riskAnalysis

riskAnalysis	Format
score Score attribué à chaque transaction permettant d'évaluer le risque qui lui est associé.	string
resultCode Code renvoyé par un analyseur de risque externe. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour renvoyés par l'analyseur de risque externe .	string
status Statut de l'analyse de risque. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • P_SEND_OK, "Sent to clearsale and successfully processed" Succès • P_TO_SEND, "Transaction analysis is scheduled to be sent to risk analyzer" L'envoi est programmé • P_TO_SEND_KO, "Problem when tried to send to risk analyzer" Erreur de traitement • P_PENDING_AT_ANALYZER, "Analysis result is still being processed by the risk analyzer. We should keep checking/waiting for the analysis result" En cours de traitement par l'analyseur • P_MANUAL, "Analysis should be requested through user request (not automatically)" Attente d'envoi manuel • P_SKIPPED, "Analysis request discarded by current transaction status/problem" Ecarté • P_SEND_EXPIRED, "Analysis request expired" Expiré 	string
requestId Identifiant de l'analyse chez l'analyseur de risque.	string
extralInfo Pas de valorisation pour ClearSale. Pour CyberSource, cet attribut retourne tous les codes renvoyés par le DecisionManager. <ul style="list-style-type: none"> • key : nom de la donnée (son format est "string"). • value : valeur de la donnée (son format est "string"). Exemple : <ul style="list-style-type: none"> • COR-BA=Adresse de facturation corrigée ou corrigeable • A=Changements d'adresse excessifs. L'acheteur a changé d'adresse de facturation au moins deux fois au cours des six derniers mois. • etc. 	extInfo

Tableau 101 : Attribut riskAnalysis

L'attribut riskAssessment

Valeurs	Description
ENABLE_3DS	3D Secure activé.
DISABLE_3DS	3D Secure désactivé.
MANUAL_VALIDATION	La transaction est créée en validation manuelle. La remise du paiement est bloquée temporairement pour permettre au marchand de procéder à toutes les vérifications souhaitées.
REFUSE	La transaction est refusée.
RUN_RISK_ANALYSIS	Appel à un analyseur de risques externes sous condition que le marchand possède un contrat. Se référer à la description du champ vads_risk_analysis_result pour identifier la liste des valeurs possibles et leur description.
INFORM	Une alerte est remontée. Le marchand est averti qu'un risque est identifié.

Valeurs	Description
	Le marchand est informé via une ou plusieurs des règles du centre de notification (URL de notification, e-mail ou SMS).

extraResponse

L'objet **extraResponse** permet d'obtenir des informations supplémentaires à propos du paiement.

extraResponse	Format
paymentOptionCode Code de l'option de paiement utilisée lors du paiement.	string
paymentOptionOccNumb Nombre d'échéances lors d'un paiement en plusieurs fois. Par exemple, pour un paiement en trois fois, cet attribut sera valorisé à 3.	n..2

Tableau 102 : Objet extraResponse

6.7. Dupliquer une transaction de paiement 'duplicatePayment'

L'opération **duplicatePayment** permet de créer une nouvelle transaction ayant exactement les mêmes caractéristiques que la transaction qui a servi de base à la duplication.

Les transactions pouvant faire l'objet d'une duplication doivent posséder un des statuts suivants :

- Remisé
- Expiré
- Annulé
- Refusé

Requête à envoyer

L'opération **duplicatePayment** est utilisée pour créer une nouvelle transaction ayant exactement les mêmes caractéristiques que la transaction qui a servi de base à la duplication.

La requête **duplicatePayment** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **duplicatePayment** prend en entrée un objet de type **duplicatePayment**.

Le type **duplicatePayment** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
paymentRequest	paymentRequest	✓
queryRequest	queryRequest	✓
orderRequest	orderRequest	

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Un seul attribut peut être valorisé si besoin :

commonRequest		
Attribut	Requis	Format
comment Commentaire libre.		string

Tableau 103 : Objet commonRequest

Le commentaire sera affiché dans l'historique de la transaction visible depuis le Back Office.

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

L'attribut à valoriser est le suivant :

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction. Remarque: cet attribut est utilisé pour remplacer les anciens attributs transactionId , sequenceNumber et creationDate . Cependant il est possible de continuer à utiliser ces anciens attributs pour des raisons de rétrocompatibilité. Pour plus d'informations, référez-vous au chapitre Gérer le rétrocompatibilité .	✓	string

Tableau 104 : Objet queryRequest

paymentRequest

L'objet **paymentRequest** permet de transmettre des informations liées au paiement.

Il possède les attributs suivants :

paymentRequest		
Attribut	Requis	Format
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none">• <u>Soit cet identifiant est généré par la plateforme</u>. Dans ce cas, ce paramètre ne doit pas être renseigné.• <u>Soit cet identifiant est généré par le site marchand</u>. Dans ce cas, ce paramètre doit être renseigné avec la valeur de l'identifiant souhaité. Attention, il incombe au site marchand de s'assurer de l'unicité des identifiants. Toute demande d'enregistrement contenant un identifiant déjà existant, sera rejetée, et retournera un code d'erreur 12. Remarque : cet attribut ne peut être envoyé à vide.		an..6
amount Montant de la transaction dans sa plus petite unité monétaire . <u>Remarque</u> : <ul style="list-style-type: none">• Ne doit pas être envoyé à vide ou être à 0.• Ne doit pas être supérieur au montant initial (cas du remboursement).	✓	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	✓	n3
expectedCaptureDate Date de remise demandée exprimée au format ISO 8601 définit par W3C. <i>Exemple</i> : 2016-07-16T19:20:00Z. Ce paramètre est utilisé pour effectuer un paiement différé. Si le nombre de jours entre la date de remise demandée et la date actuelle est supérieur à la durée de validité de l'autorisation , une autorisation de 100 XPF sera réalisée le jour de la transaction. Ceci afin de vérifier la validité de la carte. L'autorisation pour le montant total sera effectuée : <ul style="list-style-type: none">• fonctionnement par défaut : le jour de la date de remise en banque souhaitée,• fonctionnement avec autorisation anticipée : selon le moyen de paiement sélectionné, à J- le nombre de jours correspondant à la durée de validité d'une autorisation avant la date de remise en banque souhaitée. Si vous souhaitez être notifié du résultat de cette demande d'autorisation, vous devez configurer la règle de notification URL de notification sur autorisation par Batch dans le Back Office (Paramétrage > Règles de notifications). <u>Remarque</u> : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.		dateTime ans..40

paymentRequest		
Attribut	Requis	Format
manualValidation Permet de valider manuellement une transaction tant que la date de remise en banque souhaitée n'est pas dépassée. Pour cela, cet attribut doit être valorisé à 1 (validation manuelle). Valorisé à 0 , la validation sera automatique.		n1

Tableau 105 : Objet paymentRequest

orderRequest

L'objet **orderRequest** permet de transmettre des informations liées à la commande.

Il est composé de l'attribut suivant :

orderRequest		
Attribut	Requis	Format
orderId Référence de la commande.		string an..64
extInfo Champs personnalisables permettant d'ajouter des données supplémentaires (champ supplémentaire qui sera persisté dans la transaction et sera retourné dans la réponse). L'attribut extInfo est composé de sous objets : <ul style="list-style-type: none"> • key : nom de la donnée. Son format est "string". • value : valeur de la donnée. Son format est "string". <u>Exemple</u> : <extInfo><key>keyData</key><value>valuedata</value></extInfo>		extInfo

Tableau 106 : Objet orderRequest

Réponse en retour

La réponse à l'opération **duplicatePayment** est faite par la plateforme de paiement suite à une demande de duplication d'une transaction.

Elle est constituée d'un HEADER et d'un BODY de type **duplicatePaymentResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **duplicatePaymentResponse** est la suivante :

Nom	Type
duplicatePaymentResult	duplicatePaymentResult

La structure du message **duplicatePaymentResult** est la suivante :

Objet	Type
commonResponse	commonResponse
paymentResponse	paymentResponse
orderResponse	orderResponse
cardResponse	cardResponse
authorizationResponse	authorizationResponse
captureResponse	captureResponse
customerResponse	customerResponse
markResponse	markResponse
threeDSResponse	threeDSResponse
extraResponse	extraResponse
fraudManagementResponse	fraudManagementResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">INITIAL En traitement. Ce statut est temporaire. Le statut définitif sera retourné aussitôt la synchronisation réalisée.NOT_CREATED La transaction n'est pas créée et n'est pas visible dans le Back Office.AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue.AUTHORISED_TO_VALIDATE A valider. La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la remise en banque. La transaction peut être validée tant que la date de remise n'est pas dépassée. Si cette date est dépassée, le paiement prend le statut Expiré (statut définitif).WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement.WAITING_AUTHORISATION_TO_VALIDATE A valider et à autoriser. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation sera déclenchée automatiquement à J-1 avant la date de remise en banque. Le paiement pourra être accepté ou refusé. La remise en banque est automatique.REFUSED Refusée. La transaction est refusée.	string
shopId Identifiant de la boutique.	n8
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">EC pour le commerce électronique.MOTO pour une commande par e-mail ou téléphone.CC pour un centre d'appel.OTHER pour un autre canal de vente.	string (enum)
submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40

commonResponse	Format
contractNumber Numéro de contrat commerçant utilisé.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 107 : Objet commonResponse

paymentResponse

L'objet **paymentResponse** détaille les informations sur la transaction.

paymentResponse	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none"> • Soit cet identifiant est généré par la plateforme. • Soit cet identifiant est généré par le site marchand. 	an..6
amount Montant de la transaction dans sa plus petite unité monétaire .	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	n3
effectiveAmount Montant de la transaction dans la devise réellement utilisée pour effectuer la remise en banque.	n..12
effectiveCurrency Devise réellement utilisée pour effectuer la remise en banque.	n3
expectedCaptureDate Date de remise en banque souhaitée. Remarque : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.	dateTime ans..40
operationType Type d'opération. Les valeurs possibles sont : <ul style="list-style-type: none"> • 0 pour une opération de débit. • 1 pour une opération de remboursement. 	n1
creationDate Date et heure de l'enregistrement de la transaction exprimée au format W3C.	dateTime ans..40
externalTransactionId Référence fournie par un tiers : numéro de transaction pour PayPal, Boletto, RRN pour Prism, etc...	string
liabilityShift Transfert de responsabilité. Les valeurs possibles sont : <ul style="list-style-type: none"> • YES lorsque le paiement est garanti. • NO lorsque le paiement n'est pas garanti. 	string
paymentType Type de paiement. Les valeurs possibles sont : <ul style="list-style-type: none"> • SINGLE Paie ment en une seule fois. • INSTALLMENT Paie ment en plusieurs fois. • SPLIT Paie ment effectué avec plusieurs moyens de paiement. • SUBSCRIPTION 	string (enum)

paymentResponse	Format
<p>Païement par alias ou lié à un abonnement.</p> <ul style="list-style-type: none"> RETRY Lors d'un paiement refusé, il est possible de réitérer la demande de paiement. Pour toute(s) réitération(s), le paiement est valorisé avec cette valeur. <p><i>Remarque :</i> Les valeurs INSTALLMENT et SPLIT peuvent être retournées uniquement si des paiements ont été créés via le formulaire de paiement.</p> 	
<p>sequenceNumber Numéro de séquence de la transaction. Sa valeur est fonction du contexte du paiement :</p> <ul style="list-style-type: none"> Est valorisé à "1" pour un paiement unitaire. Est valorisé à "2" lors du rejeu d'un paiement initialement refusé. Prend la valeur du numéro d'échéance dans le cas d'un paiement en plusieurs fois créé à partir du formulaire de paiement. 	n..3
<p>paymentError Complément d'information en cas d'erreur technique. Retourne un code d'erreur associé à l'erreur technique (voir chapitre Codes d'erreurs pouvant survenir durant le paiement).</p>	n..3
<p>nsu Apparaît lorsque isNsuRequested est valorisé à 1 dans la requête.</p>	string

Tableau 108 : Objet paymentResponse

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
<p>orderId Référence de la commande.</p>	string an..64
<p>extInfo Données personnalisées retournées en fonction des besoins. <i>Exemple :</i> extInfo.key, extInfo.value</p> <ul style="list-style-type: none"> key : nom de la donnée (son format est "string"). value : valeur de la donnée (son format est "string"). 	extInfo

Tableau 109 : Objet orderResponse

cardResponse

L'objet **cardResponse** détaille les informations du moyen de paiement utilisé.

cardResponse	Format
<p>number</p> <ul style="list-style-type: none"> Numéro de carte masqué. Contient les 6 premiers chiffres du numéro, suivi par "XXXXXX" et enfin les 4 derniers chiffres. IBAN et BIC utilisés pour le paiement, séparés par un « _ » dans le cas d'un paiement par prélèvement. 	string
<p>scheme Type de la carte.</p>	string
<p>brand Marque de la carte utilisée pour la transaction</p>	string
<p>country Code pays du pays d'émission de la carte (Code numérique ISO 3166).</p>	ISO 3166
<p>productCode Code produit de la carte.</p>	an..3
<p>bankCode Code banque de la banque émettrice.</p>	n..5
<p>expiryMonth Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).</p>	n..2
<p>expiryYear</p>	n4

cardResponse	Format
Année d'expiration sur 4 chiffres (ex : 2023).	

Tableau 110 : Objet cardResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
mode Spécifie de quelle manière est réalisée la demande d'autorisation. Deux valeurs possibles : <ul style="list-style-type: none">• MARK Une autorisation de a été réalisée afin de vérifier la validité de la carte. Ce cas se présente lorsque la date de remise dépasse la période de validité d'une autorisation .• FULL Autorisation pour le montant total demandé dans la requête.	string
amount Montant de l'autorisation dans la plus petite unité monétaire dans le cas où mode vaut FULL .	n..12
currency Code de la monnaie utilisée lors de la demande d'autorisation (suivant la norme ISO 4217) dans le cas où mode vaut FULL .	n3
date Date et heure de la demande d'autorisation dans le cas où mode vaut FULL .	dateTime ans..40
number Numéro de la demande d'autorisation dans le cas où mode vaut FULL .	n..2
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 111 : Objet authorizationResponse

captureResponse

L'objet **captureResponse** permet d'obtenir des informations à propos de la remise dans le cas où la transaction est remise.

captureResponse	Format
date Date et heure de remise.	dateTime ans..40
number Numéro de remise.	n3
reconciliationStatus Statut du rapprochement bancaire de la transaction.	n1
refundAmount Montant ayant déjà fait l'objet d'un remboursement dans sa plus petite unité monétaire.	n..12
refundCurrency Devise du montant ayant déjà fait l'objet d'un remboursement (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF) .	n3
chargeback Litige, impayé. Les valeurs possibles sont : <ul style="list-style-type: none">• 0 : transaction ne faisant pas l'objet d'un litige.• 1 : transaction faisant l'objet d'un litige.	

Tableau 112 : Objet captureResponse

customerResponse

L'objet **customerResponse** détaille de nombreuses informations à propos de l'acheteur.

La réponse se décompose en l'analyse de :

- **billingDetails**

Données de facturation de l'acheteur.

- **shippingDetails**

Données de livraison de l'acheteur. Les valeurs présentes dans cet objet seront identiques à celles de l'objet **billingDetails**.

- **extraDetails**

Données techniques liées à l'acheteur.

6.7.2.7.1. Objet billingDetails

billingDetails	
Attribut	Format
reference Référence de l'acheteur.	string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..	string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none">• PRIVATE pour un particulier.• COMPANY pour une entreprise.	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de l'acheteur.	string ans..255
district Quartier de l'acheteur.	string ans..127
zipCode Code postal de l'acheteur.	string ans..64
city Ville de l'acheteur.	string ans..128
state Etat/Région de l'acheteur.	string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none">• DE pour l'Allemagne• IT pour l'Italie• PT pour le Portugal• GB pour le Royaume-Uni• JA pour le Japon• RU pour la Russie	string - a2

billingDetails	
Attribut	Format
<ul style="list-style-type: none"> ES pour l'Espagne FR pour la France NL pour les Pays-Bas PL pour la Pologne SE pour la Suède CN pour la Chine 	
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> de pour l'allemand en pour l'anglais es pour l'espagnol fr pour le français tr pour le turc it pour l'italien ja pour le japonais nl pour le néerlandais pl pour le polonais pt pour le portugais ru pour le russe sv pour le suédois zh pour le chinois 	string - a2
cellPhoneNumber Numéro de téléphone mobile	string ans..32
legalName Raison sociale de la société.	string ans..128

Tableau 113 : Objet billingDetails

6.7.2.7.2. Objet shippingDetails

shippingDetails	
Attribut	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de livraison.	string ans..255
address2 Complément d'adresse de livraison.	string ans..255
district Quartier de livraison.	string ans..127
zipCode Code postal de livraison.	string ans..64
city Ville de livraison.	string ans..128
state Etat/Région de livraison.	string ans..128
country	string - a2

shippingDetails	
Attribut	Format
Pays de livraison.	
deliveryCompanyName Informations sur le transporteur.	string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> • STANDARD pour une livraison standard. • EXPRESS pour une livraison express. 	string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont : <ul style="list-style-type: none"> • RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. • RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). • RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. • PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). • ETICKET pour l'émission d'un billet électronique, téléchargement. 	string (enum)
legalName Raison sociale de la société.	string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).	string ans..255

Tableau 114 : Objet shippingDetails

6.7.2.7.3. Objet extraDetails

extraDetails	Format
ipAddress L'adresse IP de l'acheteur.	string ans40

Tableau 115 : Objet extraDetails

markResponse

L'objet **markResponse** permet d'obtenir des informations sur la demande d'autorisation à 100 XPF .

markResponse	Format
amount Montant utilisé pour vérifier la validité de la carte, dans la plus petite unité monétaire .	n..12
currency Code de la monnaie utilisée pour vérifier la validité de la carte (suivant la norme ISO 4217) Ex : 953 pour le Franc CFP (XPF) .	n3
date Date et heure de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
number Numéro d'autorisation de la demande d'autorisation dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
result Résultat de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK . Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 116 : Objet markResponse

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

Cet objet se décompose :

- **authenticationRequestData**

Décrit le résultat de la demande d'inscription ainsi que le message encodé qui sera transmis par le navigateur de l'acheteur à l'ACS.

- **authenticationResultData**

Décrit les détails de l'authentification 3D Secure.

authenticationRequestData	Format
threeDSAcctId Certificat renvoyé par le Directory Server.	string
threeDSAcctsUrl Url de l'ACS à contacter.	string
threeDSBrand Réseau de la carte.	string
threeDSEncodedPareq Message PAREq encodé, prêt à envoyer à l'ACS.	string
threeDSEnrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	a1
threeDSRequestId Numéro de requête, à rappeler dans l'appel ENABLED_FINALIZE de l'attribut mode de l'objet threeDSRequest .	string

Tableau 117 : Objet authenticationRequestData

authenticationResultData	Format															
<div>transactionCondition Statut de l'authentification 3D Secure. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">COND_3D_SUCCESS Succès de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure et le porteur s'est authentifié correctement.COND_3D_FAILURE Echec de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas réussi à s'authentifier (mauvais mot de passe).COND_3D_ERROR Authentification en erreur. Le marchand participe au programme 3D Secure mais le serveur de la plateforme de paiement a rencontré un problème technique durant le processus d'authentification (lors de la vérification de l'inscription de la carte au programme 3D ou de l'authentification du porteur).COND_3D_NOTENROLLED Porteur non enrôlé. Le marchand participe au programme 3D Secure mais la carte du porteur n'est pas enrôlée.COND_3D_ATTEMPT Tentative d'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas eu à s'authentifier (le serveur de contrôle d'accès de la banque qui a émis la carte n'implémente que la génération d'une preuve de tentative d'authentification).COND_SSL 3D Secure non applicable. Le marchand n'est pas enrôlé à 3D Secure ou le canal de vente n'est pas couvert par cette garantie.</div>	string															
<div>enrolled Statut enrôlement du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut enrôlé.N pour un statut non enrôlé.U pour un statut inconnu.</div>	string															
<div>status Statut de l'authentification du porteur. Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">Y pour un statut authentifié 3 DS.N pour une erreur d'authentification.U pour une authentification impossible.A pour un essai d'authentification.</div>	a1															
<div>eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont :</div> <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>5</td><td>6</td><td>7</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>		status = Y	status = A	status = U	status = N	VISA - AMEX	5	6	7	-	MasterCard	02	01	-	-	string
	status = Y	status = A	status = U	status = N												
VISA - AMEX	5	6	7	-												
MasterCard	02	01	-	-												
<div>xid Numéro de transaction 3DS.</div>	string															
<div>cavvAlgorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont :</div> <div><ul style="list-style-type: none">0 pour HMAC.1 pour CVV.</div>	n1															

authenticationResultData	Format
<ul style="list-style-type: none"> 2 pour CVV_ATN. 3 pour Mastercard SPA. 	
cavv Certificat de l'ACS.	string
signValid Signature de l'authentification 3DS.	string
brand Réseau de la carte.	string

Tableau 118 : Objet authenticationResultData

extraResponse

L'objet **extraResponse** permet d'obtenir des informations supplémentaires à propos du paiement.

extraResponse	Format
paymentOptionCode Code de l'option de paiement utilisée lors du paiement.	string
paymentOptionOccNumb Nombre d'échéances lors d'un paiement en plusieurs fois. Par exemple, pour un paiement en trois fois, cet attribut sera valorisé à 3.	n..2

Tableau 119 : Objet extraResponse

fraudManagementResponse

L'objet **fraudManagementResponse** permet d'obtenir les résultats des contrôles de gestion de risques.

La réponse se décompose en l'analyse des attributs :

- **riskControl**
Retourne le résultat du contrôle de gestion de risques effectué.
- **riskAnalysis**
Retourne le résultat de l'analyse de gestion de risques effectué par un système externe (Konduto, ClearSale, CyberSource,...).
- **riskAssessment**
Retourne le résultat de l'analyse de gestion des risques avancée effectuée par la plateforme de paiement.

L'attribut riskControl

Le format est le suivant : name1=result1;name2=result2

Nom	Format	Description
name	string	Nom de la règle de gestion de risque.
result	string	Résultat du contrôle.

Valeurs possibles pour 'name'	
CARD	Carte enregistrée en liste grise.
COUNTRY	Pays de l'acheteur enregistré en liste grise.
IPADDR	Adresse IP de l'acheteur enregistrée en liste grise.
AMOUNT	Montant maximum autorisé par commande est atteint.
BIN	Le code BIN de la carte est enregistré en liste grise.
ECB	Détection d'une e-carte bleue.
CARD_COMMERCIAL_NATIONAL	Détection d'une carte commerciale nationale.
CARD_COMMERCIAL_FOREIGN	Détection d'une carte commerciale étrangère.
CAS	Détection d'une carte à autorisation systématique.

Valeurs possibles pour 'name'	
COUNTRY_CONSISTENCY	Le pays d'origine de la carte, le pays de l'adresse IP de l'acheteur et le pays de l'acheteur ne correspondent pas.
NON_GUARANTEED_PAYMENT	Détection d'un paiement sans transfert de responsabilité.
IPADDR_COUNTRY	Le pays de l'adresse IP de l'acheteur est enregistré en liste grise.

Tableau 120 : Valeurs possibles pour 'name'

Valeurs possibles pour 'result'	
OK	Indique que le contrôle est correct.
KO	Indique que le contrôle est en erreur.

Tableau 121 : Valeurs possibles pour 'result'

L'attribut riskAnalysis

riskAnalysis	Format
score Score attribué à chaque transaction permettant d'évaluer le risque qui lui est associé.	string
resultCode Code renvoyé par un analyseur de risque externe. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour renvoyés par l'analyseur de risque externe .	string
status Statut de l'analyse de risque. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • P_SEND_OK, "Sent to clearsale and successfully processed" Succès • P_TO_SEND, "Transaction analysis is scheduled to be sent to risk analyzer" L'envoi est programmé • P_TO_SEND_KO, "Problem when tried to send to risk analyzer" Erreur de traitement • P_PENDING_AT_ANALYZER, "Analysis result is still being processed by the risk analyzer. We should keep checking/waiting for the analysis result" En cours de traitement par l'analyseur • P_MANUAL, "Analysis should be requested through user request (not automatically)" Attente d'envoi manuel • P_SKIPPED, "Analysis request discarded by current transaction status/problem" Ecarté • P_SEND_EXPIRED, "Analysis request expired" Expiré 	string
requestId Identifiant de l'analyse chez l'analyseur de risque.	string
extraInfo Pas de valorisation pour ClearSale. Pour CyberSource, cet attribut retourne tous les codes renvoyés par le DecisionManager. <ul style="list-style-type: none"> • key : nom de la donnée (son format est "string"). • value : valeur de la donnée (son format est "string"). Exemple : <ul style="list-style-type: none"> • COR-BA=Adresse de facturation corrigée ou corrigable • A=Changements d'adresse excessifs. L'acheteur a changé d'adresse de facturation au moins deux fois au cours des six derniers mois. • etc. 	extInfo

Tableau 122 : Attribut riskAnalysis

L'attribut riskAssessment

Valeurs	Description
ENABLE_3DS	3D Secure activé.
DISABLE_3DS	3D Secure désactivé.
MANUAL_VALIDATION	La transaction est créée en validation manuelle. La remise du paiement est bloquée temporairement pour permettre au marchand de procéder à toutes les vérifications souhaitées.
REFUSE	La transaction est refusée.
RUN_RISK_ANALYSIS	Appel à un analyseur de risques externes sous condition que le marchand possède un contrat. Se référer à la description du champ vads_risk_analysis_result pour identifier la liste des valeurs possibles et leur description.
INFORM	Une alerte est remontée. Le marchand est averti qu'un risque est identifié. Le marchand est informé via une ou plusieurs des règles du centre de notification (URL de notification, e-mail ou SMS).

6.8. Valider une transaction de paiement 'validatePayment'

L'opération **validatePayment** permet d'autoriser la remise en banque d'une transaction à la date de présentation demandée dans le paiement original.

Les transactions pouvant faire l'objet d'une validation possèdent l'un des statuts suivants :

- A valider
- A valider et autoriser

Requête à envoyer

L'opération **validatePayment** est utilisée pour valider un paiement.

La requête **validatePayment** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **validatePayment** prend en entrée un objet de type **validatePayment**.

Le type **validatePayment** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Un seul attribut peut être valorisé si besoin :

commonRequest		
Attribut	Requis	Format
comment Commentaire libre.		string

Tableau 123 : Objet commonRequest

Le commentaire sera affiché dans l'historique de la transaction visible depuis le Back Office.

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

L'attribut à valoriser est le suivant :

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction. Remarque: cet attribut est utilisé pour remplacer les anciens attributs transactionId , sequenceNumber et creationDate . Cependant il est possible de continuer à utiliser ces anciens attributs pour des raisons de rétrocompatibilité. Pour plus d'informations, référez-vous au chapitre Gérer le rétrocompatibilité .	✓	string

Tableau 124 : Objet queryRequest

Réponse en retour

La réponse à l'opération **validatePayment** est faite par la plateforme de paiement suite à une validation d'un paiement. Elle est constituée d'un HEADER et d'un BODY de type **validatePaymentResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **validatePaymentResponse** est la suivante :

Nom	Type
validatePaymentResult	validatePaymentResult

La structure du message **validatePaymentResult** est la suivante :

Objet	Type
commonResponse	commonResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue.• WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement.• REFUSED	string

commonResponse	Format
Refusée. La transaction est refusée.	
shopId Identifiant de la boutique.	n8
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none"> • EC pour le commerce électronique. • MOTO pour une commande par e-mail ou téléphone. • CC pour un centre d'appel. • OTHER pour un autre canal de vente. 	string
submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 125 : Objet commonResponse

6.9. Remiser une transaction de paiement 'capturePayment'

L'opération **capturePayment** est spécifique à certains types de paiements effectués au Brésil.
Il permet de remiser un paiement dans la mesure où la transaction est en attente de remise.

Requête à envoyer

La requête **capturePayment** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **capturePayment** prend en entrée un objet de type **capturePayment**.

Le type **capturePayment** est composé d'un seul objet :

Objet	Format	Requis
settlementRequest	settlementRequest	✓

settlementRequest

L'objet **settlementRequest** permet de remiser une transaction par carte ou Boleto (Bresil).

settlementRequest		
Attribut	Requis	Format
transactionUuids Liste de transactions sur lesquelles une remise est demandée.	✓	string
commission Réservé à un usage spécifique (Brésil). Cet attribut est obligatoire pour Boleto. Commission payée à la banque pour le service de facturation.		n2
date Réservé à un usage spécifique (Brésil). Date de remise en banque exprimée au format ISO 8601 défini par W3C. Exemple : 2016-07-16T00:00:00Z. Les heures, minutes et secondes seront égales à zéro pour cet attribut.		dateTime ans..40

Tableau 126 : Objet settlementRequest

Réponse en retour

La réponse à l'opération **capturePayment** est faite par la plateforme de paiement suite à la remise d'un paiement.

Elle est constituée d'un HEADER et d'un BODY de type **capturePaymentResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **capturePaymentResponse** est la suivante :

Nom	Type
capturePaymentResult	capturePaymentResult

La structure du message **capturePaymentResult** est la suivante :

Objet	Type
commonResponse	commonResponse

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 127 : Objet commonResponse

6.10. Obtenir le détail d'une transaction de paiement 'getPaymentDetails'

L'opération **getPaymentDetails** permet de réaliser une demande de résultat d'un paiement pour en connaître ses différents attributs.

Requête à envoyer

La requête **getPaymentDetails** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **getPaymentDetails** prend en entrée un objet de type **getPaymentDetails**.

Le type **getPaymentDetails** est composé de deux objets :

Objet	Format	Requis
queryRequest	queryRequest	✓
extendedResponseRequest	extendedResponseRequest	

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

L'attribut à valoriser est le suivant :

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction. Remarque: cet attribut est utilisé pour remplacer les anciens attributs transactionId , sequenceNumber et creationDate . Cependant il est possible de continuer à utiliser ces anciens attributs pour des raisons de retrocompatibilité. Pour plus d'informations, référez-vous au chapitre Gérer le rétrocompatibilité .	✓	string

Tableau 128 : Objet queryRequest

extendedResponseRequest

L'objet **extendedResponseRequest** permet d'avoir une réponse adaptée au besoin du marchand.

L'attribut à valoriser est le suivant :

extendedResponseRequest		
Attribut	Requis	Format
isNsuRequested Les valeurs possibles : <ul style="list-style-type: none">• 0 : false.• 1 : true. Si valorisé à 1 alors la réponse contiendra un champ supplémentaire dans PaymentResponse .	✓	n1

Tableau 129 : Objet extendedResponseRequest

Réponse en retour

La réponse à l'opération **getPaymentDetails** est faite par la plateforme de paiement suite à une demande d'information sur un paiement.

Elle est constituée d'un HEADER et d'un BODY de type **getPaymentDetailsResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **getPaymentDetailsResponse** est la suivante :

Nom	Type
getPaymentDetailsResult	getPaymentDetailsResult

La structure du message **getPaymentDetailsResult** est la suivante :

Objet	Type
commonResponse	commonResponse
paymentResponse	paymentResponse
orderResponse	orderResponse
cardResponse	cardResponse
authorizationResponse	authorizationResponse
captureResponse	captureResponse
customerResponse	customerResponse
markResponse	markResponse
threeDSResponse	threeDSResponse
extraResponse	extraResponse
fraudManagementResponse	fraudManagementResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque : les objets **subscriptionResponse** et **tokenResponse** ne sont pas valorisés dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail	string

commonResponse	Format
<p>Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.</p>	
<p>transactionStatusLabel Libellé du statut de la transaction. Les valeurs possibles sont :</p> <ul style="list-style-type: none"> • AUTHORISED En attente de remise. La transaction est acceptée et sera remise en banque automatiquement à la date prévue. • AUTHORISED_TO_VALIDATE A valider. La transaction, créée en validation manuelle, est autorisée. Le marchand doit valider manuellement la remise en banque. La transaction peut être validée tant que la date de remise n'est pas dépassée. Si cette date est dépassée, le paiement prend le statut Expiré (statut définitif). • WAITING_AUTHORISATION En attente d'autorisation. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation et la remise en banque seront déclenchées automatiquement. • WAITING_AUTHORISATION_TO_VALIDATE A valider et à autoriser. La date de remise demandée est supérieure à la date de fin de validité d'une demande d'autorisation. Une autorisation de a été réalisée et acceptée par la banque émettrice. La demande d'autorisation sera déclenchée automatiquement à J-1 avant la date de remise en banque. Le paiement pourra être accepté ou refusé. La remise en banque est automatique. • REFUSED Refusée. La transaction est refusée. • CAPTURED La transaction est remise en banque. • CANCELLED Annulée. La transaction est annulée par le marchand. • EXPIRED Expirée. La date de remise est atteinte mais le marchand n'a pas validé la transaction. • UNDER_VERIFICATION (Spécifique à PayPal) En attente de vérification par PayPal. Cette valeur signifie que PayPal retient la transaction pour suspicion de fraude . Le paiement est alors dans l'onglet Paiement en cours. 	string
<p>shopId Identifiant de la boutique.</p>	n8
<p>paymentSource Origine de la transaction. Les valeurs possibles sont :</p> <ul style="list-style-type: none"> • EC pour le commerce électronique. • MOTO pour une commande par e-mail ou téléphone. • CC pour un centre d'appel. • OTHER pour un autre canal de vente. 	string
<p>submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).</p>	dateTime ans..40
<p>contractNumber Numéro de contrat commerçant utilisé.</p>	string
<p>paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).</p>	string

Tableau 130 : Objet commonResponse

paymentResponse

L'objet **paymentResponse** détaille les informations sur la transaction.

paymentResponse	Format
transactionUuid Référence unique de la transaction générée par la plateforme de paiement.	string ans32
transactionId Identifiant de la transaction lors de la création ou la modification d'une transaction de paiement. Sa valeur est unique sur une même journée. <ul style="list-style-type: none">• Soit cet identifiant est généré par la plateforme.• Soit cet identifiant est généré par le site marchand.	an..6
amount Montant de la transaction dans sa plus petite unité monétaire .	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	n3
effectiveAmount Montant de la transaction dans la devise réellement utilisée pour effectuer la remise en banque.	n..12
effectiveCurrency Devise réellement utilisée pour effectuer la remise en banque.	n3
expectedCaptureDate Date de remise en banque souhaitée. <u>Remarque</u> : si le délai avant remise est supérieur à 365 jours dans la requête de paiement, il est automatiquement repositionné à 365 jours.	dateTime ans..40
operationType Type d'opération. Les valeurs possibles sont : <ul style="list-style-type: none">• 0 pour une opération de débit.• 1 pour une opération de remboursement.	n1
creationDate Date et heure de l'enregistrement de la transaction exprimée au format W3C.	dateTime ans..40
externalTransactionId Référence fournie par un tiers : numéro de transaction pour PayPal, Boleto, RRN pour Prism, etc...	string
liabilityShift Transfert de responsabilité. Les valeurs possibles sont : <ul style="list-style-type: none">• YES lorsque le paiement est garanti.• NO lorsque le paiement n'est pas garanti.	string
paymentType Type de paiement. Les valeurs possibles sont : <ul style="list-style-type: none">• SINGLE Paie ment en une seule fois.• INSTALLMENT Paie ment en plusieurs fois.• SPLIT Paie ment effectué avec plusieurs moyens de paiement.• SUBSCRIPTION Paie ment par alias ou lié à un abonnement.• RETRY Lors d'un paiement refusé, il est possible de réitérer la demande de paiement. Pour toute(s) réitération(s), le paiement est valorisé avec cette valeur. <u>Remarque</u> : Les valeurs INSTALLMENT et SPLIT peuvent être retournées uniquement si des paiements ont été créés via le formulaire de paiement.	string (enum)

paymentResponse	Format
sequenceNumber Numéro de séquence de la transaction. Sa valeur est fonction du contexte du paiement : <ul style="list-style-type: none"> Est valorisé à "1" pour un paiement unitaire. Est valorisé à "2" lors du rejeu d'un paiement initialement refusé. Prend la valeur du numéro d'échéance dans le cas d'un paiement en plusieurs fois créé à partir du formulaire de paiement. 	n..3
paymentError Complément d'information en cas d'erreur technique. Retourne un code d'erreur associé à l'erreur technique (voir chapitre Codes d'erreurs pouvant survenir durant le paiement).	n..3
nsu Apparaît lorsque isNsuRequested est valorisé à 1 dans la requête.	string

Tableau 131 : Objet paymentResponse

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
orderId Référence de la commande.	string an..64
extInfo Données personnalisées retournées en fonction des besoins. Exemple : extInfo.key, extInfo.value <ul style="list-style-type: none"> key : nom de la donnée (son format est "string"). value : valeur de la donnée (son format est "string"). 	extInfo

Tableau 132 : Objet orderResponse

cardResponse

L'objet **cardResponse** détaille les informations du moyen de paiement utilisé.

cardResponse	Format
number <ul style="list-style-type: none"> Numéro de carte masqué. Contient les 6 premiers chiffres du numéro, suivi par "XXXXXX" et enfin les 4 derniers chiffres. IBAN et BIC utilisés pour le paiement, séparés par un « _ » dans le cas d'un paiement par prélèvement. 	string
scheme Type de la carte.	string
brand Marque de la carte utilisée pour la transaction	string
country Code pays du pays d'émission de la carte (Code numérique ISO 3166).	ISO 3166
productCode Code produit de la carte.	an..3
bankCode Code banque de la banque émettrice.	n..5
expiryMonth Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).	n..2
expiryYear Année d'expiration sur 4 chiffres (ex : 2023).	n4

Tableau 133 : Objet cardResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
mode Spécifie de quelle manière est réalisée la demande d'autorisation. Deux valeurs possibles : <ul style="list-style-type: none">• MARK Une autorisation de a été réalisée afin de vérifier la validité de la carte. Ce cas se présente lorsque la date de remise dépasse la période de validité d'une autorisation .• FULL Autorisation pour le montant total demandé dans la requête.	string
amount Montant de l'autorisation dans la plus petite unité monétaire dans le cas où mode vaut FULL .	n..12
currency Code de la monnaie utilisée lors de la demande d'autorisation (suivant la norme ISO 4217) dans le cas où mode vaut FULL .	n3
date Date et heure de la demande d'autorisation dans le cas où mode vaut FULL .	dateTime ans..40
number Numéro de la demande d'autorisation dans le cas où mode vaut FULL .	n..2
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 134 : Objet authorizationResponse

captureResponse

L'objet **captureResponse** permet d'obtenir des informations à propos de la remise dans le cas où la transaction est remise.

captureResponse	Format
date Date et heure de remise.	dateTime ans..40
number Numéro de remise.	n3
reconciliationStatus Statut du rapprochement bancaire de la transaction.	n1
refundAmount Montant ayant déjà fait l'objet d'un remboursement dans sa plus petite unité monétaire.	n..12
refundCurrency Devise du montant ayant déjà fait l'objet d'un remboursement (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF) .	n3
chargeback Litige, impayé. Les valeurs possibles sont : <ul style="list-style-type: none">• 0 : transaction ne faisant pas l'objet d'un litige.• 1 : transaction faisant l'objet d'un litige.	

Tableau 135 : Objet captureResponse

customerResponse

L'objet **customerResponse** détaille de nombreuses informations à propos de l'acheteur.

La réponse se décompose en l'analyse de :

- **billingDetails**

Données de facturation de l'acheteur.

- **shippingDetails**

Données de livraison de l'acheteur. Les valeurs présentes dans cet objet seront identiques à celles de l'objet **billingDetails**.

- **extraDetails**

Données techniques liées à l'acheteur.

6.10.2.7.1. Objet billingDetails

billingDetails	
Attribut	Format
reference Référence de l'acheteur.	string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..	string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none">• PRIVATE pour un particulier.• COMPANY pour une entreprise.	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de l'acheteur.	string ans..255
district Quartier de l'acheteur.	string ans..127
zipCode Code postal de l'acheteur.	string ans..64
city Ville de l'acheteur.	string ans..128
state Etat/Région de l'acheteur.	string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none">• DE pour l'Allemagne• IT pour l'Italie• PT pour le Portugal• GB pour le Royaume-Uni• JA pour le Japon• RU pour la Russie	string - a2

billingDetails	
Attribut	Format
<ul style="list-style-type: none"> ES pour l'Espagne FR pour la France NL pour les Pays-Bas PL pour la Pologne SE pour la Suède CN pour la Chine 	
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> de pour l'allemand en pour l'anglais es pour l'espagnol fr pour le français tr pour le turc it pour l'italien ja pour le japonais nl pour le néerlandais pl pour le polonais pt pour le portugais ru pour le russe sv pour le suédois zh pour le chinois 	string - a2
cellPhoneNumber Numéro de téléphone mobile	string ans..32
legalName Raison sociale de la société.	string ans..128

Tableau 136 : Objet billingDetails

6.10.2.7.2. Objet shippingDetails

shippingDetails	
Attribut	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de livraison.	string ans..255
address2 Complément d'adresse de livraison.	string ans..255
district Quartier de livraison.	string ans..127
zipCode Code postal de livraison.	string ans..64
city Ville de livraison.	string ans..128
state Etat/Région de livraison.	string ans..128
country	string - a2

shippingDetails	
Attribut	Format
Pays de livraison.	
deliveryCompanyName Informations sur le transporteur.	string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> • STANDARD pour une livraison standard. • EXPRESS pour une livraison express. 	string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont : <ul style="list-style-type: none"> • RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. • RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). • RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. • PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). • ETICKET pour l'émission d'un billet électronique, téléchargement. 	string (enum)
legalName Raison sociale de la société.	string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).	string ans..255

Tableau 137 : Objet shippingDetails

6.10.2.7.3. Objet extraDetails

extraDetails	Format
ipAddress L'adresse IP de l'acheteur.	string ans40

Tableau 138 : Objet extraDetails

markResponse

L'objet **markResponse** permet d'obtenir des informations sur la demande d'autorisation à 100 XPF .

markResponse	Format
amount Montant utilisé pour vérifier la validité de la carte, dans la plus petite unité monétaire .	n..12
currency Code de la monnaie utilisée pour vérifier la validité de la carte (suivant la norme ISO 4217) Ex : 953 pour le Franc CFP (XPF) .	n3
date Date et heure de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
number Numéro d'autorisation de la demande d'autorisation dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK .	dateTime ans..40
result Résultat de la demande d'autorisation réalisée dans le cas où l'attribut mode de l'objet authorizationResponse vaut MARK . Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 139 : Objet markResponse

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

Cet objet se décompose :

- **authenticationRequestData**

Décrit le résultat de la demande d'enrôlement ainsi que le message encodé qui sera transmis par le navigateur de l'acheteur à l'ACS.

- **authenticationResultData**

Décrit les détails de l'authentification 3D Secure.

authenticationRequestData	Format
threeDSAcctId Certificat renvoyé par le Directory Server.	string
threeDSAcUrl Url de l'ACS à contacter.	string
threeDSBrand Réseau de la carte.	string
threeDSEncodedPareq Message PAREq encodé, prêt à envoyer à l'ACS.	string
threeDSEnrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	a1
threeDSRequestId Numéro de requête, à rappeler dans l'appel ENABLED_FINALIZE de l'attribut mode de l'objet threeDSRequest .	string

Tableau 140 : Objet authenticationRequestData

authenticationResultData	Format															
transactionCondition Statut de l'authentification 3D Secure. Les valeurs possibles sont : <ul style="list-style-type: none">• COND_3D_SUCCESS Succès de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure et le porteur s'est authentifié correctement.• COND_3D_FAILURE Echec de l'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas réussi à s'authentifier (mauvais mot de passe).• COND_3D_ERROR Authentification en erreur. Le marchand participe au programme 3D Secure mais le serveur de la plateforme de paiement a rencontré un problème technique durant le processus d'authentification (lors de la vérification de l'inscription de la carte au programme 3D ou de l'authentification du porteur).• COND_3D_NOTENROLLED Porteur non enrôlé. Le marchand participe au programme 3D Secure mais la carte du porteur n'est pas enrôlée.• COND_3D_ATTEMPT Tentative d'authentification. Le marchand et le porteur de la carte sont inscrits au programme 3D Secure mais l'acheteur n'a pas eu à s'authentifier (le serveur de contrôle d'accès de la banque qui a émis la carte n'implémente que la génération d'une preuve de tentative d'authentification).• COND_SSL 3D Secure non applicable. Le marchand n'est pas enrôlé à 3D Secure ou le canal de vente n'est pas couvert par cette garantie.	string															
enrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	string															
status Statut de l'authentification du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut authentifié 3 DS.• N pour une erreur d'authentification.• U pour une authentification impossible.• A pour un essai d'authentification.	a1															
eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont : <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>5</td><td>6</td><td>7</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>		status = Y	status = A	status = U	status = N	VISA - AMEX	5	6	7	-	MasterCard	02	01	-	-	string
	status = Y	status = A	status = U	status = N												
VISA - AMEX	5	6	7	-												
MasterCard	02	01	-	-												
xid Numéro de transaction 3DS.	string															
cavvAlgorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont : <ul style="list-style-type: none">• 0 pour HMAC.• 1 pour CVV.	n1															

authenticationResultData	Format
<ul style="list-style-type: none"> 2 pour CVV_ATN. 3 pour Mastercard SPA. 	
cavv Certificat de l'ACS.	string
signValid Signature de l'authentification 3DS.	string
brand Réseau de la carte.	string

Tableau 141 : Objet authenticationResultData

fraudManagementResponse

L'objet **fraudManagementResponse** permet d'obtenir les résultats des contrôles de gestion de risques.

La réponse se décompose en l'analyse des attributs :

- riskControl**

Retourne le résultat du contrôle de gestion de risques effectué.

- riskAnalysis**

Retourne le résultat de l'analyse de gestion de risques effectué par un système externe (Konduto, ClearSale, CyberSource,...).

- riskAssessment**

Retourne le résultat de l'analyse de gestion des risques avancée effectuée par la plateforme de paiement.

L'attribut riskControl

Le format est le suivant : name1=result1;name2=result2

Nom	Format	Description
name	string	Nom de la règle de gestion de risque.
result	string	Résultat du contrôle.

Valeurs possibles pour 'name'	
CARD	Carte enregistrée en liste grise.
COUNTRY	Pays de l'acheteur enregistré en liste grise.
IPADDR	Adresse IP de l'acheteur enregistrée en liste grise.
AMOUNT	Montant maximum autorisé par commande est atteint.
BIN	Le code BIN de la carte est enregistré en liste grise.
ECB	Détection d'une e-carte bleue.
CARD_COMMERCIAL_NATIONAL	Détection d'une carte commerciale nationale.
CARD_COMMERCIAL_FOREIGN	Détection d'une carte commerciale étrangère.
CAS	Détection d'une carte à autorisation systématique.
COUNTRY_CONSISTENCY	Le pays d'origine de la carte, le pays de l'adresse IP de l'acheteur et le pays de l'acheteur ne correspondent pas.
NON_GUARANTEED_PAYMENT	Détection d'un paiement sans transfert de responsabilité.
IPADDR_COUNTRY	Le pays de l'adresse IP de l'acheteur est enregistré en liste grise.

Tableau 142 : Valeurs possibles pour 'name'

Valeurs possibles pour 'result'	
OK	Indique que le contrôle est correct.
KO	Indique que le contrôle est en erreur.

Tableau 143 : Valeurs possibles pour 'result'

L'attribut riskAnalysis

riskAnalysis	Format
score Score attribué à chaque transaction permettant d'évaluer le risque qui lui est associé.	string
resultCode Code renvoyé par un analyseur de risque externe. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour renvoyés par l'analyseur de risque externe .	string
status Statut de l'analyse de risque. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • P_SEND_OK, "Sent to clearsale and successfully processed" Succès • P_TO_SEND, "Transaction analysis is scheduled to be sent to risk analyzer" L'envoi est programmé • P_TO_SEND_KO, "Problem when tried to send to risk analyzer" Erreur de traitement • P_PENDING_AT_ANALYZER, "Analysis result is still being processed by the risk analyzer. We should keep checking/waiting for the analysis result" En cours de traitement par l'analyseur • P_MANUAL, "Analysis should be requested through user request (not automatically)" Attente d'envoi manuel • P_SKIPPED, "Analysis request discarded by current transaction status/problem" Ecarté • P_SEND_EXPIRED, "Analysis request expired" Expiré 	string
requestId Identifiant de l'analyse chez l'analyseur de risque.	string
extralInfo Pas de valorisation pour ClearSale. Pour CyberSource, cet attribut retourne tous les codes renvoyés par le DecisionManager. <ul style="list-style-type: none"> • key : nom de la donnée (son format est "string"). • value : valeur de la donnée (son format est "string"). Exemple : <ul style="list-style-type: none"> • COR-BA=Adresse de facturation corrigée ou corrigeable • A=Changements d'adresse excessifs. L'acheteur a changé d'adresse de facturation au moins deux fois au cours des six derniers mois. • etc. 	extInfo

Tableau 144 : Attribut riskAnalysis

L'attribut riskAssessment

Valeurs	Description
ENABLE_3DS	3D Secure activé.
DISABLE_3DS	3D Secure désactivé.
MANUAL_VALIDATION	La transaction est créée en validation manuelle. La remise du paiement est bloquée temporairement pour permettre au marchand de procéder à toutes les vérifications souhaitées.
REFUSE	La transaction est refusée.
RUN_RISK_ANALYSIS	Appel à un analyseur de risques externes sous condition que le marchand possède un contrat. Se référer à la description du champ vads_risk_analysis_result pour identifier la liste des valeurs possibles et leur description.
INFORM	Une alerte est remontée. Le marchand est averti qu'un risque est identifié.

Valeurs	Description
	Le marchand est informé via une ou plusieurs des règles du centre de notification (URL de notification, e-mail ou SMS).

extraResponse

L'objet **extraResponse** permet d'obtenir des informations supplémentaires à propos du paiement.

extraResponse	Format
paymentOptionCode Code de l'option de paiement utilisée lors du paiement.	string
paymentOptionOccNumb Nombre d'échéances lors d'un paiement en plusieurs fois. Par exemple, pour un paiement en trois fois, cet attribut sera valorisé à 3.	n..2

Tableau 145 : Objet extraResponse

6.11. Vérifier l'authentification 3D Secure 'verifyThreeDSEnrollment'

L'opération **verifyThreeDSEnrollment** permet de vérifier que la carte de l'acheteur est compatible avec l'authentification 3D Secure.

Requête à envoyer

La requête **verifyThreeDSEnrollment** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **verifyThreeDSEnrollment** prend en entrée un objet de type **verifyThreeDSEnrollment**.

Le type **verifyThreeDSEnrollment** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
paymentRequest	paymentRequest	✓
cardRequest	cardRequest	✓
techRequest	techRequest	
threeDSRequest	threeDSRequest	

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Il possède les attributs suivants :

commonRequest		
Attribut	Requis	Format
contractNumber Numéro de contrat commerçant utilisé.		string

Tableau 146 : Objet commonRequest

paymentRequest

L'objet **paymentRequest** permet de transmettre des informations liées au paiement.

Il possède les attributs suivants :

paymentRequest		
Attribut	Requis	Format
amount Montant de la transaction dans sa plus petite unité monétaire . <u>Remarque :</u> Ne doit pas être envoyé à vide ou être à 0. Si vous ne souhaitez pas modifier le statut de la transaction, vous devez valoriser l'attribut amount avec la valeur initiale. Si aucune information n'est modifiée, la requête sera rejetée avec un code erreur.	✓	n..12
currency Code de la devise de la transaction (norme ISO 4217). Ex : 953 pour le Franc CFP (XPF)	✓	n3

Tableau 147 : Objet paymentRequest

cardRequest

L'objet **cardRequest** permet de transmettre les informations sur la carte de paiement.

Deux cas de figure :

- Cas d'une vérification sans alias (identifiant)

Il possède les attributs suivants :

cardRequest		
Attribut	Requis	Format
number Numéro de la carte.	✓	string
expiryMonth Mois d'expiration de la carte, entre 1 et 12.	✓	n..2
expiryYear Année d'expiration de la carte sur 4 digits. <i>Exemple : 2016</i>	✓	n4

Tableau 148 : Objet cardRequest

- Cas d'une vérification avec alias (identifiant)

Il possède l'attribut suivant :

cardRequest		
Attribut	Requis	Format
paymentToken Identifiant unique (alias) associé à un moyen de paiement. <ul style="list-style-type: none">• Soit cet identifiant a été généré par la plateforme.• Soit cet identifiant a été généré par le site marchand.	✓	string ans..64

Tableau 149 : Objet cardRequest

techRequest

L'objet **techRequest** permet de transmettre des informations techniques à propos du navigateur de l'acheteur.

Cet objet doit obligatoirement être envoyé dans la requête.

Cependant, ses attributs sont facultatifs.

techRequest		
Attribut	Requis	Format
browserUserAgent Header « User-Agent » du navigateur de l'acheteur (HTTP/1.1 - RFC. 2616).		string
browserAccept Header « Accept » du navigateur de l'acheteur (HTTP/1.1 - RFC. 2616).		string
integrationType Nom et /ou version de la solution e-commerce utilisée.		string

Tableau 150 : Objet techRequest

threeDSRequest

L'objet **threeDSRequest** permet de transmettre des informations liées à 3D Secure.

Dans l'opération **verifyThreeDSEnrollment**, **threeDSRequest** est **spécifique à l'Inde**.

En Inde, des informations complémentaires sont requises pour procéder à l'authentification (exemple : numéro de téléphone de l'acheteur).

Pour cela, un attribut doit être envoyé :

threeDSRequest	
Attribut	Format
mpiExtension Données complémentaires suite à la vérification effectuée par le MPI du marchand. Est composé d'un sous objet extensionData de type extInfo . Exemple: extensionData.key, extensionData.value <ul style="list-style-type: none">key: nom de la donnée (son format est "string").value: valeur de la donnée (son format est "string").	extensionData

Exemple :

```
<threeDSRequest>
  <mpiExtension>
    <extensionData>
      <key>extensionType</key>
      <value>npc356</value>
    </extensionData>
    <extensionData>
      <key>phoneid</key>
      <value>910000000000</value>
    </extensionData>
  </mpiExtension>
</threeDSRequest>
```

Réponse en retour

La réponse à l'opération **verifyThreeDSEnrollement** est faite par la plateforme de paiement suite à une vérification de l'enrôlement de la carte de l'acheteur à 3D Secure.

Elle est constituée d'un HEADER et d'un BODY de type **verifyThreeDSEnrollementResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **verifyThreeDSEnrollementResponse** est la suivante :

Nom	Type
verifyThreeDSEnrollementResult	verifyThreeDSEnrollementResult

La structure du message **verifyThreeDSEnrollementResult** est la suivante :

Objet	Type
commonResponse	commonResponse
threeDSResponse	threeDSResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

Les attributs contenant une valeur dans la réponse sont les suivants :

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 151 : Objet commonResponse

Les attributs **shopId**, **paymentSource**, **submissionDate**, **contractNumber** et **paymentToken** ne sont pas valorisés dans cette opération.

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

- **authenticationRequestData**

Décrit le résultat de la demande d'enrôlement ainsi que le message encodé qui sera transmis par le navigateur de l'acheteur à l'ACS.

authenticationRequestData	Format
threeDSAcctId Certificat renvoyé par le Directory Server.	string
threeDSAcUrl Url de l'ACS à contacter.	string
threeDSBrand Réseau de la carte.	string
threeDSEncodedPareq Message PAREq encodé, prêt à envoyer à l'ACS.	string
threeDSEnrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	a1
threeDSRequestId Numéro de requête, à rappeler dans l'appel ENABLED_FINALIZE de l'attribut mode de l'objet threeDSRequest .	string

Tableau 152 : Objet authenticationRequestData

6.12. Vérifier le statut de l'authentification 3D Secure 'checkThreeDSAuthentication'

L'opération **checkThreeDSAuthentication** permet de vérifier le statut de l'authentification 3D Secure.

Requête à envoyer

La requête **checkThreeDSAuthentication** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **checkThreeDSAuthentication** prend en entrée un objet de type **checkThreeDSAuthentication**.

Le type **checkThreeDSAuthentication** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
threeDSRequest	threeDSRequest	✓

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Il possède les attributs suivants :

commonRequest		
Attribut	Requis	Format
contractNumber Numéro de contrat commerçant utilisé.		string

Tableau 153 : Objet commonRequest

threeDSRequest

L'objet **threeDSRequest** permet de transmettre des informations liées à 3D Secure.

Deux attributs doivent obligatoirement être envoyés :

threeDSRequest		
Attribut	Requis	Format
requestId Doit contenir la valeur retournée dans l'attribut threeDSRequestId de l'objet authenticationRequestData dans l'opération verifyThreeDSEnrollment .	✓	string
pares Message PaRes (Payer Authentication Response) renvoyé par l'ACS.	✓	string

Tableau 154 : Objet threeDSRequest

Réponse en retour

La réponse à l'opération **checkThreeDSAuthentication** est faite par la plateforme de paiement suite à une vérification du statut de l'authentification 3D Secure.

Elle est constituée d'un HEADER et d'un BODY de type **checkThreeDSAuthenticationResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **checkThreeDSAuthenticationResponse** est la suivante :

Nom	Type
checkThreeDSAuthenticationResult	checkThreeDSAuthenticationResult

La structure du message **checkThreeDSAuthenticationResult** est la suivante :

Objet	Type
commonResponse	commonResponse
threeDSResponse	threeDSResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès (à ne pas confondre avec le statut de la transaction).• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 155 : Objet commonResponse

threeDSResponse

L'objet **threeDSResponse** permet d'obtenir des informations à propos de l'authentification 3D Secure.

- **authenticationResultData**

Décrit les détails de l'authentification 3D-Secure.

authenticationResultData	Format															
enrolled Statut enrôlement du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut enrôlé.• N pour un statut non enrôlé.• U pour un statut inconnu.	string															
status Statut de l'authentification du porteur. Les valeurs possibles sont : <ul style="list-style-type: none">• Y pour un statut authentifié 3 DS.• N pour une erreur d'authentification.• U pour une authentification impossible.• A pour un essai d'authentification.	a1															
eci Indicateur de commerce Electronique. La valeur eci est fonction du statut de l'authentification 3DS et du type de carte. Les valeurs possibles sont : <table><tr><th></th><th>status = Y</th><th>status = A</th><th>status = U</th><th>status = N</th></tr><tr><td>VISA - AMEX</td><td>5</td><td>6</td><td>7</td><td>-</td></tr><tr><td>MasterCard</td><td>02</td><td>01</td><td>-</td><td>-</td></tr></table>		status = Y	status = A	status = U	status = N	VISA - AMEX	5	6	7	-	MasterCard	02	01	-	-	string
	status = Y	status = A	status = U	status = N												
VISA - AMEX	5	6	7	-												
MasterCard	02	01	-	-												
xid Numéro de transaction 3DS.	string															
cavvAlgorithm Algorithme de vérification de l'authentification du porteur (CAVV). Les valeurs possibles sont : <ul style="list-style-type: none">• 0 pour HMAC.• 1 pour CVV.• 2 pour CVV_ATN.• 3 pour Mastercard SPA.	n1															
cavv Certificat de l'ACS.	string															
signValid Signature de l'authentification 3DS.	string															
brand Réseau de la carte.	string															

Tableau 156 : Objet authenticationResultData

7. RÉALISER DES OPÉRATIONS SPÉCIFIQUES AUX PAIEMENTS PAR IDENTIFIANT

L'option paiement par identifiant permet de réaliser un certain nombre d'opérations telles que :

Opération	Nom de l'opération web service à utiliser
Créer un alias (identifiant)	createToken
Créer un alias (identifiant) à partir d'une transaction	createTokenFromTransaction
Réaliser des paiements récurrents (abonnements)	createSubscription
Modifier un alias (identifiant)	updateToken
Modifier un abonnement	updateSubscription
Récupérer le détail d'un alias (identifiant)	getTokenDetails
Récupérer le détail d'un abonnement	getSubscriptionDetails
Résilier un alias (identifiant)	cancelToken
Annuler un abonnement	cancelSubscription
Réactiver un alias (identifiant)	reactivateToken

Tableau 157 : Opérations disponibles avec l'option paiement par identifiant

Des exemples de codage sont proposés en annexe de ce document.

7.1. Créer un alias (identifiant) 'createToken'

L'opération **createToken** permet à un site marchand d'offrir à ses acheteurs la possibilité d'associer un alias (identifiant) à un ou plusieurs numéros de carte bancaire, dans le but de faciliter des paiements ultérieurs.

Cette opération permet :

- Des paiements rapides et sécurisés (paiement en un clic).
- D'effectuer des paiements périodiques ou abonnement.

Le partage d'identifiants

Il est possible de partager des identifiants (alias) entre plusieurs entités juridiques.

Les identifiants partagés entre plusieurs entités juridiques doivent être uniques et doivent être impérativement générés par la plateforme de paiement (en d'autres termes l'attribut **paymentToken** de l'objet **cardRequest** ne doit pas être renseigné).

Cependant, cette fonctionnalité est soumise à des conditions particulières. Veuillez contacter l'interlocuteur de votre plateforme de paiement pour en prendre connaissance.

Requête à envoyer

La requête **createToken** est constituée d'un HEADER et d'un BODY.

• HEADER

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

• BODY

L'opération **createToken** prend en entrée un objet de type **createToken**.

Le type **createToken** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
cardRequest	cardRequest	✓
customerRequest	customerRequest	✓

Note :

L'objet **shippingDetails** contenu dans le **customerRequest** n'est pas pris en compte lors de la création d'un alias.

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Il possède les attributs suivants :

commonRequest		
Attribut	Requis	Format
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• EC pour le commerce électronique. Paramètre utilisé par défaut si aucune valeur est renseignée ou si différente des valeurs possibles.• MOTO pour une commande par courrier ou téléphone.• CC pour un centre d'appel.• OTHER pour un autre canal de vente. Seule la valeur EC permet de créer une transaction avec 3D Secure. Les autres valeurs ne doivent être utilisées que pour de la vente à distance, pour laquelle le 3D Secure n'est pas applicable.		string
submissionDate Date et heure UTC de la transaction exprimée au format ISO 8601 défini par W3C. <i>Exemple : 2016-07-16T19:20:00Z</i> Si la valeur de cet attribut est trop éloignée de l'heure actuelle, la requête sera rejetée (code d'erreur 13).		dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé. Si ce champ est renseigné, veuillez à utiliser le bon contrat en fonction du réseau de la carte. Par exemple, un contrat CB ne peut être utilisé pour une transaction AMEX.		string
comment Commentaire libre.		string

Tableau 158 : Objet CommonRequest

cardRequest

L'objet **cardRequest** permet de transmettre les informations sur la carte de paiement.

Pour créer un alias, plusieurs attributs sont requis :

cardRequest		
Attribut	Requis	Format
number Numéro de la carte.	✓	string
scheme Types de cartes. Les valeurs possibles sont AMEX , CB , MASTERCARD , VISA , VISA_ELECTRON , VPAY , MAESTRO , E-CARTEBLEUE ou JCB .	✓	string
expiryMonth Mois d'expiration de la carte, entre 1 et 12.	✓	n..2
expiryYear Année d'expiration de la carte sur 4 digits. <i>Exemple : 2016</i>	✓	n4
cardSecurityCode Cryptogramme visuel à 3 chiffres (ou 4 pour Amex). Ce champ est obligatoire lorsque la carte dispose d'un code cryptogramme visuel. Si le CVV n'est pas transmis, la banque émettrice refusera le paiement. En revanche, ce champ est facultatif lorsque l'origine de la transaction est valorisée à MOTO.		string
cardHolderBirthday Date de naissance du porteur au format YYYY-MM-DD.	Requis selon le moyen de paiement	dateTime ans..64
paymentToken Identifiant unique (alias) associé à un moyen de paiement. <ul style="list-style-type: none">• Soit cet identifiant est généré par la plateforme. Dans ce cas, ce paramètre ne doit pas être renseigné.• Soit cet identifiant est généré par le site marchand. Dans ce cas, ce paramètre doit être renseigné avec la valeur de l'identifiant souhaité. Attention, il incombe au site marchand de s'assurer de l'unicité des identifiants. Toute demande d'enregistrement contenant un identifiant déjà existant, sera rejetée, et provoquera l'affichage d'un message d'erreur.		string ans..64

Tableau 159 : Objet cardRequest

customerRequest

L'objet **customerRequest** permet de transmettre des informations liées à la livraison, à la facturation et des données techniques liées à l'acheteur

Cet objet doit obligatoirement être envoyé dans la requête.


Il est composé des sous-objets suivants :

Format	Sous-objet	Requis
billingDetails Données de facturation de l'acheteur.	billingDetailsRequest	✓
shippingDetails Données de livraison de l'acheteur.	shippingDetailsRequest	
extraDetails Données techniques liées à l'acheteur.	extraDetailsRequest	

Tableau 160 : Sous-objets de customerRequest

7.1.2.3.1. Objet billingDetails

billingDetails possède les attributs suivants :

billingDetails		
Attribut	Requis	Format
reference Référence de l'acheteur.		string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..		string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 		string (enum)
firstName Nom de l'acheteur.		string ans..128
lastName Prénom de l'acheteur.		string ans..128
phoneNumber Numéro de téléphone de l'acheteur.		string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	 pour la création d'un alias	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>		string an..5
address Adresse de l'acheteur.		string ans..255
district Quartier de l'acheteur.		string ans..127
zipCode Code postal de l'acheteur.		string ans..64
city Ville de l'acheteur.		string ans..128
state Etat/Région de l'acheteur.		string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none"> DE pour l'Allemagne IT pour l'Italie PT pour le Portugal GB pour le Royaume-Uni JA pour le Japon RU pour la Russie ES pour l'Espagne NL pour les Pays-Bas SE pour la Suède FR pour la France PL pour la Pologne CN pour la Chine 		string - a2
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> de pour l'allemand it pour l'italien pt pour le portugais en pour l'anglais ja pour le japonais ru pour le russe es pour l'espagnol nl pour le néerlandais sv pour le suédois fr pour le français pl pour le polonais zh pour le chinois 		string - a2

billingDetails		
Attribut	Requis	Format
<ul style="list-style-type: none"> tr pour le turc 		
cellPhoneNumber Numéro de téléphone mobile		string ans..32
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).		string ans..255

Tableau 161 : Objet billingDetails

7.1.2.3.2. Objet shippingDetails

shippingDetails possède les attributs suivants :

shippingDetails		
Attribut	Requis	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 		string (enum)
firstName Nom de l'acheteur.		string ans..128
lastName Prénom de l'acheteur.		string ans..128
phoneNumber Numéro de téléphone de l'acheteur.		string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>		string an..5
address Adresse de livraison.		string ans..255
address2 Complément d'adresse de livraison.		string ans..255
district Quartier de livraison.		string ans..127
zipCode Code postal de livraison.		string ans..64
city Ville de livraison.		string ans..128
state Etat/Région de livraison.		string ans..128
country Pays de livraison.		string - a2
deliveryCompanyName Informations sur le transporteur.		string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> STANDARD pour une livraison standard. EXPRESS pour une livraison express. 		string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont :		string (enum)

shippingDetails		
Attribut	Requis	Format
<ul style="list-style-type: none"> RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). ETICKET pour l'émission d'un billet électronique, téléchargement. 		
legalName Raison sociale de la société.		string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).		string ans..255

Tableau 162 : Objet shippingDetails

7.1.2.3.3. Objet extraDetails

extraDetails possède les attributs suivants :

extraDetails		
Attribut	Requis	Format
ipAddress L'adresse IP de l'acheteur.		string ans40
fingerPrintId Identifiant unique de session. Spécifique au Brésil et à l'analyseur de fraude ClearSale. Codé sur 128 octets, peut être composé de majuscules ou de minuscules, chiffres ou tiret ([A-Z] [a-z], 0-9, _, -).		string ans128

Tableau 163 : Objet extraDetails

Exemple de requête à envoyer

Exemple de requête à envoyer :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:v5="http://v5.ws.vads.lyra.com/">
  <soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">
    <soapHeader:shopId>12345678</soapHeader:shopId>
    <soapHeader:requestId>78c944ce-511b-4e5a-b9cb-312e79666ed8</soapHeader:requestId>
    <soapHeader:timestamp>2015-04-01T12:24:56Z</soapHeader:timestamp>
    <soapHeader:mode>TEST</soapHeader:mode>
    <soapHeader:authToken>/a3609j1fU765pBBPEmTmjtjawL08dNVmYd0zVevHtA=</soapHeader:authToken>
  </soap:Header>
  <soap:Body>
    <v5:createToken>

      <commonRequest>
        <submissionDate>2015-04-01T12:24:56Z</submissionDate>
      </commonRequest>
      <cardRequest>
        <number>4970100000000003</number>
        <scheme>VISA</scheme>
        <expiryMonth>12</expiryMonth>
        <expiryYear>2018</expiryYear>
        <cardSecurityCode>123</cardSecurityCode>
      </cardRequest>
      <customerRequest>

        <billingDetails>
          <title>Mr</title>
          <type>PRIVATE</type>
          <firstName>Jean</firstName>
          <lastName>Dupont</lastName>
          <phoneNumber>123456789</phoneNumber>
          <email>jean.dupont@example.com</email>
          <streetNumber>15</streetNumber>
          <address>test address</address>
          <district>district</district>
          <zipCode>31000</zipCode>
        </billingDetails>
      </customerRequest>
    </v5:createToken>
  </soap:Body>
</soap:Envelope>
```

```

        <city>TOULOUSE</city>
        <state>state</state>
        <country>France</country>
        <cellPhoneNumber>0612345678</cellPhoneNumber>
        <legalName>Jean Dupont</legalName>
    </billingDetails>

    <shippingDetails>
        <type>PRIVATE</type>
        <firstName>Jean</firstName>
        <lastName>Dupont</lastName>
        <phoneNumber>123456789</phoneNumber>
        <streetNumber>1234</streetNumber>
        <address>street</address>
        <address2>street2</address2>
        <district>district</district>
        <zipCode>1234</zipCode>
        <city>City</city>
        <state>State</state>
        <country>France</country>
        <deliveryCompanyName>DELIVERYCOMP</deliveryCompanyName>
        <shippingSpeed>STANDARD</shippingSpeed>
        <shippingMethod>PACKAGE_DELIVERY_COMPANY</shippingMethod>
        <legalName>Jean Dupont</legalName>
    </shippingDetails>
</customerRequest>
</v5:createToken>
</soap:Body>
</soap:Envelope>

```

Réponse en retour

La réponse à l'opération **createToken** est faite par la plateforme de paiement lors de la création d'un alias (Identifiant compte acheteur) afin d'effectuer des paiements en un clic (paiement par identifiant).

Elle est constituée d'un HEADER et d'un BODY de type **createTokenResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **createTokenResponse** est la suivante :

Nom	Type
createTokenResult	createTokenResult

La structure du message **createTokenResult** est la suivante :

Objet	Type
commonResponse	commonResponse
authorizationResponse si la demande d'autorisation est refusée par la banque.	authorizationResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque :

Les objets **paymentResponse**, **orderResponse**, **cardResponse**, **captureResponse**, **customerResponse**, **markResponse**, **threeDSResponse**, **extraResponse**, **subscriptionResponse**, **shoppingCartResponse** et **fraudManagementResponse** ne sont pas valorisés dans la réponse.

Remarque :

Pour obtenir les détails du moyen de paiement utilisé, utilisez l'opération **getTokenDetails**. L'objet **cardResponse** est retourné dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">La valeur 0 indique que l'opération s'est déroulée avec succès.Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 164 : Objet commonResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

Un seul attribut est retourné dans la réponse :

authorizationResponse	Format
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 165 : Objet authorizationResponse

Exemple de réponse

Exemple de réponse :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">78c944ce-511b-4e5a-b9cb-312e79666ed8</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T12:24:56Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/Header/">padDTG41Q1UEh560px+dwKl3bgtjkkv6d2c4ahoQPJs=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:createTokenResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createTokenResult>
        <requestId>78c944ce-511b-4e5a-b9cb-312e79666ed8</requestId>
        <commonResponse>
          <responseCode>0</responseCode>
          <responseCodeDetail>Action successfully completed</responseCodeDetail>
          <paymentToken>cb059d56f8564674bc139a373a8daebb</paymentToken>
        </commonResponse>
      </createTokenResult>
    </ns2:createTokenResponse>
  </soap:Body>
</soap:Envelope>
```

7.2. Créer un alias (identifiant) à partir d'une transaction 'createTokenFromTransaction'

L'opération **createTokenFromTransaction** permet à un site marchand de créer un alias (identifiant) à partir d'une transaction existante.

Pour lutter contre les impayés, avant la création de l'alias, la validité du moyen de paiement utilisé pour la transaction d'origine est vérifiée sur la dernière transaction effectuée avec ce moyen de paiement.

Ainsi, il est possible que depuis une transaction de paiement valide, l'opération **createTokenFromTransaction** échoue dû à une transaction invalide plus récente.

Les données de facturation et de livraison de la transaction initiale ne seront par utilisées pour créer l'alias.

Requête à envoyer

La requête **createTokenFromTransaction** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **createTokenFromTransaction** prend en entrée un objet de type **createTokenFromTransaction**.

Le type **createTokenFromTransaction** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓
cardRequest	cardRequest	

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Il possède les attributs suivants :

commonRequest		
Attribut	Requis	Format
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• EC pour le commerce électronique. Paramètre utilisé par défaut si aucune valeur est renseignée ou si différente des valeurs possibles.• MOTO pour une commande par courrier ou téléphone.• CC pour un centre d'appel.• OTHER pour un autre canal de vente. Seule la valeur EC permet de créer une transaction avec 3D Secure. Les autres valeurs ne doivent être utilisées que pour de la vente à distance, pour laquelle le 3D Secure n'est pas applicable.		string
submissionDate Date et heure UTC de la transaction exprimée au format ISO 8601 défini par W3C. <i>Exemple : 2016-07-16T19:20:00Z</i>		dateTime ans..40

commonRequest		
Attribut	Requis	Format
Si la valeur de cet attribut est trop éloignée de l'heure actuelle, la requête sera rejetée (code d'erreur 13).		
contractNumber Numéro de contrat commerçant utilisé. Si ce champ est renseigné, veuillez à utiliser le bon contrat en fonction du réseau de la carte. Par exemple, un contrat CB ne peut être utilisé pour une transaction AMEX.		string
comment Commentaire libre.		string

Tableau 166 : Objet CommonRequest

queryRequest

L'objet **queryRequest** permet d'interroger un alias (identifiant de compte) pour en connaître ses différents attributs.

Seul l'attribut **uuid** est indispensable pour cette opération.

queryRequest		
Attribut	Requis	Format
uuid Référence unique de la transaction.	✓	string

Tableau 167 : Objet queryRequest

cardRequest

L'objet **cardRequest** permet de transmettre les informations sur la carte de paiement.

Seul l'attribut **paymentToken** peut être valorisé pour personnaliser l'alias.

cardRequest		
Attribut	Requis	Format
paymentToken Identifiant unique (alias) associé à un moyen de paiement. <ul style="list-style-type: none"> • Soit cet identifiant a été généré par la plateforme. • Soit cet identifiant a été généré par le site marchand. 		string ans..64

Tableau 168 : Objet cardRequest

Réponse en retour

La réponse à l'opération **createTokenFromTransaction** est faite par la plateforme de paiement lors de la création d'un alias (Identifiant) lors d'un paiement.

Elle est constituée d'un HEADER et d'un BODY de type **createTokenFromTransactionResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **createTokenFromTransactionResponse** est la suivante :

Nom	Type
createTokenFromTransactionResult	createTokenFromTransactionResult

La structure du message **createTokenFromTransactionResult** est la suivante :

Objet	Type
commonResponse	commonResponse
authorizationResponse	authorizationResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque :

Les objets **paymentResponse**, **orderResponse**, **cardResponse**, **captureResponse**, **customerResponse**, **markResponse**, **threeDSResponse**, **extraResponse**, **subscriptionResponse**, **shoppingCartResponse** et **fraudManagementResponse** ne sont pas valorisés dans la réponse.

Remarque :

Pour obtenir les détails du moyen de paiement utilisé, utilisez l'opération **getTokenDetails**. L'objet **cardResponse** est retourné dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">La valeur 0 indique que l'opération s'est déroulée avec succès.Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 169 : Objet commonResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

Un seul attribut est retourné dans la réponse :

authorizationResponse	Format
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 170 : Objet authorizationResponse

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

0 : Action réalisée avec succès

2 : Attribut invalide (l'identifiant de la boutique doit être le même que celui de la transaction)

10 : Transaction non trouvée

35 : Création de l'alias refusée (la transaction utilisée pour créer l'alias doit être réalisée avec succès et autorisée par la banque)

56 : PAN non trouvé (PAN de la transaction d'origine non trouvé)

7.3. Modifier un alias (identifiant) 'updateToken'

L'opération **updateToken** permet de modifier l'ensemble des informations enregistrées à propos de l'acheteur (ses données bancaires ou personnelles).

Requête à envoyer

La requête **updateToken** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **updateToken** prend en entrée un objet de type **updateToken**.

Le type **updateToken** est constitué des paramètres suivants :

Modification du moyen de paiement :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓
cardRequest	cardRequest	✓
customerRequest	customerRequest	

Modification des informations de l'acheteur :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓
cardRequest	cardRequest	
customerRequest	customerRequest	✓

Note:

L'objet **shippingDetails** contenu dans le **customerRequest** n'est pas pris en compte.

Modification du moyen de paiement et des informations de l'acheteur :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓
cardRequest	cardRequest	✓
customerRequest	customerRequest	✓

Note:

L'objet **shippingDetails** contenu dans le **customerRequest** n'est pas pris en compte.

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Il possède les attributs facultatifs suivants :

commonRequest		
Attribut	Requis	Format
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• EC pour le commerce électronique. Paramètre utilisé par défaut si aucune valeur est renseignée ou si différente des valeurs possibles.• MOTO pour une commande par courrier ou téléphone.• CC pour un centre d'appel.• OTHER pour un autre canal de vente. Seule la valeur EC permet de créer une transaction avec 3D Secure. Les autres valeurs ne doivent être utilisées que pour de la vente à distance, pour laquelle le 3D Secure n'est pas applicable.		string
submissionDate Date et heure UTC de la transaction exprimée au format ISO 8601 défini par W3C. <i>Exemple : 2016-07-16T19:20:00Z</i> Si la valeur de cet attribut est trop éloignée de l'heure actuelle, la requête sera rejetée (code d'erreur 13).		dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé. Si ce champ est renseigné, veuillez à utiliser le bon contrat en fonction du réseau de la carte. Par exemple, un contrat CB ne peut être utilisé pour une transaction AMEX.		string
comment Commentaire libre.		string

Tableau 171 : Objet CommonRequest

queryRequest

L'objet **queryRequest** permet d'interroger un alias (identifiant de compte) pour en connaître ses différents attributs.

Seul l'attribut **paymentToken** est indispensable pour cette opération.

queryRequest		
Attribut	Requis	Format
paymentToken Alias (paiement par identifiant).	✓	string ans..64

Tableau 172 : Objet queryRequest

cardRequest

L'objet **cardRequest** permet de transmettre les informations sur la carte de paiement.

Selon le type de paiement (paiement par alias ou paiement avec saisie des données bancaires), un ou plusieurs attributs sont requis.

cardRequest		
Attribut	Requis	Format
number Numéro de la carte.	✓	string
scheme Types de cartes. Les valeurs possibles sont AMEX , CB , MASTERCARD , VISA , VISA_ELECTRON , VPAY , MAESTRO , E-CARTEBLEUE ou JCB .	✓	string
expiryMonth Mois d'expiration de la carte, entre 1 et 12.	✓	n..2
expiryYear Année d'expiration de la carte sur 4 digits. <i>Exemple : 2016</i>	✓	n4
cardSecurityCode Cryptogramme visuel à 3 chiffres (ou 4 pour Amex). Ce champ est obligatoire lorsque la carte dispose d'un code cryptogramme visuel. Si le CVV n'est pas transmis, la banque émettrice refusera le paiement. En revanche, ce champ est facultatif lorsque l'origine de la transaction est valorisée à MOTO.		string
cardHolderBirthday Date de naissance du porteur au format YYYY-MM-DD.	Requis selon le moyen de paiement	dateTime ans..64

Tableau 173 : Objet cardRequest

customerRequest

L'objet **customerRequest** permet de transmettre des informations liées à la livraison, à la facturation et des données techniques liées à l'acheteur

Cet objet doit obligatoirement être envoyé dans la requête.

Il est composé des sous-objets suivants :

Format	Sous-objet	Requis
billingDetails Données de facturation de l'acheteur.	billingDetailsRequest	✓
shippingDetails Données de livraison de l'acheteur.	shippingDetailsRequest	
extraDetails Données techniques liées à l'acheteur.	extraDetailsRequest	

Tableau 174 : Sous-objets de **customerRequest**

7.3.1.4.1. Objet billingDetails

billingDetails possède les attributs suivants :

billingDetails		
Attribut	Requis	Format
reference Référence de l'acheteur.		string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..		string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none">• PRIVATE pour un particulier.• COMPANY pour une entreprise.		string (enum)
firstName Nom de l'acheteur.		string ans..128
lastName Prénom de l'acheteur.		string ans..128
phoneNumber Numéro de téléphone de l'acheteur.		string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	✓ pour la création d'un alias	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>		string an..5
address Adresse de l'acheteur.		string ans..255
district Quartier de l'acheteur.		string ans..127
zipCode Code postal de l'acheteur.		string ans..64
city Ville de l'acheteur.		string ans..128
state Etat/Région de l'acheteur.		string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles :		string - a2

billingDetails		
Attribut	Requis	Format
<ul style="list-style-type: none"> DE pour l'Allemagne GB pour le Royaume-Uni ES pour l'Espagne FR pour la France IT pour l'Italie JA pour le Japon NL pour les Pays-Bas PL pour la Pologne PT pour le Portugal RU pour la Russie SE pour la Suède CN pour la Chine 		
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> de pour l'allemand en pour l'anglais es pour l'espagnol fr pour le français tr pour le turc it pour l'italien ja pour le japonais nl pour le néerlandais pl pour le polonais pt pour le portugais ru pour le russe sv pour le suédois zh pour le chinois 		string - a2
cellPhoneNumber Numéro de téléphone mobile		string ans..32
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).		string ans..255

Tableau 175 : Objet billingDetails

7.3.1.4.2. Objet shippingDetails

shippingDetails possède les attributs suivants :

shippingDetails		
Attribut	Requis	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 		string (enum)
firstName Nom de l'acheteur.		string ans..128
lastName Prénom de l'acheteur.		string ans..128
phoneNumber Numéro de téléphone de l'acheteur.		string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>		string an..5
address Adresse de livraison.		string ans..255
address2 Complément d'adresse de livraison.		string ans..255
district Quartier de livraison.		string ans..127

shippingDetails		
Attribut	Requis	Format
zipCode Code postal de livraison.		string ans..64
city Ville de livraison.		string ans..128
state Etat/Région de livraison.		string ans..128
country Pays de livraison.		string - a2
deliveryCompanyName Informations sur le transporteur.		string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> • STANDARD pour une livraison standard. • EXPRESS pour une livraison express. 		string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont : <ul style="list-style-type: none"> • RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. • RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). • RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. • PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). • ETICKET pour l'émission d'un billet électronique, téléchargement. 		string (enum)
legalName Raison sociale de la société.		string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).		string ans..255

Tableau 176 : Objet shippingDetails

7.3.1.4.3. Objet extraDetails

extraDetails possède les attributs suivants :

extraDetails		
Attribut	Requis	Format
ipAddress L'adresse IP de l'acheteur.		string ans40
fingerPrintId Identifiant unique de session. Spécifique au Brésil et à l'analyseur de fraude ClearSale. Codé sur 128 octets, peut être composé de majuscules ou de minuscules, chiffres ou tiret ([A-Z] [a-z], 0-9, _, -).		string ans128

Tableau 177 : Objet extraDetails

Réponse en retour

La réponse à l'opération **updateToken** est faite par la plateforme de paiement à une demande de modification sur un alias/identifiant compte acheteur.

Elle est constituée d'un HEADER et d'un BODY de type **updateTokenResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **updateTokenResponse** est la suivante :

Nom	Type
updateTokenResult	updateTokenResult

La structure du message **updateTokenResult** est la suivante :

Objet	Type
commonResponse	commonResponse
authorizationResponse si des modifications sur le moyen de paiement sont réalisées.	authorizationResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque : les objets **paymentResponse**, **orderResponse**, **cardResponse**, **captureResponse**, **customerResponse**, **markResponse**, **threeDSResponse**, **extraResponse**, **subscriptionResponse** et **fraudManagementResponse** ne sont pas valorisés dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">La valeur 0 indique que l'opération s'est déroulée avec succès.Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 178 : Objet commonResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
mode Spécifie de quelle manière est réalisée la demande d'autorisation. Deux valeurs possibles : <ul style="list-style-type: none">MARK Une autorisation de a été réalisée afin de vérifier la validité de la carte.	string
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 179 : Objet authorizationResponse

7.4. Récupérer le détail d'un alias (identifiant) 'getTokenDetails'

L'opération **getTokenDetails** permet de récupérer un certain nombre d'informations liées à un alias (identifiant).

Requête à envoyer

La requête **getTokenDetails** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **getTokenDetails** prend en entrée un objet de type **getTokenDetails**.

Le type **getTokenDetails** est composé d'un seul objet :

Objet	Format	Requis
queryRequest	queryRequest	✓

queryRequest

L'objet **queryRequest** permet d'interroger un alias (identifiant de compte) pour en connaître ses différents attributs.

Seul l'attribut **paymentToken** est indispensable pour cette opération.

queryRequest		
Attribut	Requis	Format
paymentToken Alias (paiement par identifiant).	✓	string ans..64

Tableau 180 : Objet queryRequest

Réponse en retour

La réponse à l'opération **getTokenDetails** est faite par la plateforme de paiement suite à une demande d'information sur un token.

Elle est constituée d'un HEADER et d'un BODY de type **getTokenDetailsResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **getTokenDetailsResponse** est la suivante :

Nom	Type
getTokenDetailsResult	getTokenDetailsResult

La structure du message **getTokenDetailsResult** est la suivante :

Objet	Type
commonResponse	commonResponse
cardResponse	cardResponse
customerResponse	customerResponse
authorizationResponse	authorizationResponse
tokenResponse	tokenResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque : les objets **paymentResponse**, **orderResponse**, **captureResponse**, **markResponse**, **subscriptionResponse**, **extraResponse** et **fraudManagementResponse** ne sont pas pris en considération dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 181 : Objet commonResponse

cardResponse

L'objet **cardResponse** détaille les informations du moyen de paiement utilisé.

cardResponse	Format
number <ul style="list-style-type: none">Numéro de carte masqué. Contient les 6 premiers chiffres du numéro, suivi par "XXXXXX" et enfin les 4 derniers chiffres.IBAN et BIC utilisés pour le paiement, séparés par un « _ » dans le cas d'un paiement par prélèvement.	string
scheme Type de la carte.	string
brand Marque de la carte utilisée pour la transaction	string
country Code pays du pays d'émission de la carte (Code numérique ISO 3166).	ISO 3166
productCode Code produit de la carte.	an..3
bankCode Code banque de la banque émettrice.	n..5
expiryMonth Mois d'expiration entre 1 et 12 (ex: 3 pour mars, 10 pour octobre).	n..2
expiryYear Année d'expiration sur 4 chiffres (ex : 2023).	n4

Tableau 182 : Objet cardResponse

authorizationResponse

L'objet **authorizationResponse** permet d'obtenir des informations sur la demande d'autorisation.

authorizationResponse	Format
date Date et heure de la demande d'autorisation dans le cas où mode vaut FULL .	dateTime ans..40
number Numéro de la demande d'autorisation dans le cas où mode vaut FULL .	an..6
result Code retour de la demande d'autorisation retourné par la banque émettrice en cas de refus. Les valeurs possibles sont listées dans le chapitre Gérer les codes de retour d'une demande d'autorisation .	n..2

Tableau 183 : Objet authorizationResponse

customerResponse

L'objet **customerResponse** détaille de nombreuses informations à propos de l'acheteur.

La réponse se décompose en l'analyse de :

- **billingDetails**

Données de facturation de l'acheteur.

- **shippingDetails**

Données de livraison de l'acheteur. Les valeurs présentes dans cet objet seront identiques à celles de l'objet **billingDetails**.

- **extraDetails**

Données techniques liées à l'acheteur.

7.4.2.4.1. Objet billingDetails

billingDetails	
Attribut	Format
reference Référence de l'acheteur.	string n..80
title Civilité de l'acheteur. Exemples de valeurs possibles : Monsieur, Madame, etc..	string n..80
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none">• PRIVATE pour un particulier.• COMPANY pour une entreprise.	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
email E-mail de l'acheteur. Paramètre obligatoire lors de la création d'un alias.	string ans..150
streetNumber Numéro de rue de l'acheteur. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de l'acheteur.	string ans..255
district Quartier de l'acheteur.	string ans..127
zipCode Code postal de l'acheteur.	string ans..64
city Ville de l'acheteur.	string ans..128
state Etat/Région de l'acheteur.	string ans..128
country Pays de l'acheteur selon la norme ISO 3166 . Exemples de valeurs possibles : <ul style="list-style-type: none">• DE pour l'Allemagne• IT pour l'Italie• PT pour le Portugal• GB pour le Royaume-Uni• JA pour le Japon• RU pour la Russie	string - a2

billingDetails	
Attribut	Format
<ul style="list-style-type: none"> ES pour l'Espagne FR pour la France NL pour les Pays-Bas PL pour la Pologne SE pour la Suède CN pour la Chine 	
language Langue de l'acheteur selon la norme ISO 639-1 . Exemples de valeurs possibles : <ul style="list-style-type: none"> de pour l'allemand en pour l'anglais es pour l'espagnol fr pour le français tr pour le turc it pour l'italien ja pour le japonais nl pour le néerlandais pl pour le polonais pt pour le portugais ru pour le russe sv pour le suédois zh pour le chinois 	string - a2
cellPhoneNumber Numéro de téléphone mobile	string ans..32
legalName Raison sociale de la société.	string ans..128

Tableau 184 : Objet billingDetails

7.4.2.4.2. Objet shippingDetails

shippingDetails	
Attribut	Format
type Type d'acheteur. Les valeurs possibles sont : <ul style="list-style-type: none"> PRIVATE pour un particulier. COMPANY pour une entreprise. 	string (enum)
firstName Nom de l'acheteur.	string ans..128
lastName Prénom de l'acheteur.	string ans..128
phoneNumber Numéro de téléphone de l'acheteur.	string ans..32
streetNumber Numéro de rue pour la livraison. <i>Remarque : si cet attribut est présent dans la requête, il ne peut être envoyé vide.</i>	string an..5
address Adresse de livraison.	string ans..255
address2 Complément d'adresse de livraison.	string ans..255
district Quartier de livraison.	string ans..127
zipCode Code postal de livraison.	string ans..64
city Ville de livraison.	string ans..128
state Etat/Région de livraison.	string ans..128
country	string - a2

shippingDetails	
Attribut	Format
Pays de livraison.	
deliveryCompanyName Informations sur le transporteur.	string ans..128
shippingSpeed Mode de livraison sélectionné. Les valeurs possibles sont : <ul style="list-style-type: none"> • STANDARD pour une livraison standard. • EXPRESS pour une livraison express. 	string (enum)
shippingMethod Méthode de livraison utilisée. Les valeurs possibles sont : <ul style="list-style-type: none"> • RECLAIM_IN_SHOP pour le retrait de la marchandise en magasin. • RELAY_POINT pour l'utilisation d'un réseau de points de retrait tiers (Kiala, Alveol, etc). • RECLAIM_IN_STATION pour le retrait dans un aéroport, une garde ou une agence de voyage. • PACKAGE_DELIVERY_COMPANY pour la livraison par transporteur (Colissimo, UPS, etc). • ETICKET pour l'émission d'un billet électronique, téléchargement. 	string (enum)
legalName Raison sociale de la société.	string ans..128
identityCode Permet d'identifier de façon unique chaque citoyen au sein d'un pays. Par exemple, au Brésil, ClearSale impose que ce champ soit valorisé avec le CPF/ CNPJ (format numérique, de longueur comprise entre 11 et 20 digits).	string ans..255

Tableau 185 : Objet shippingDetails

7.4.2.4.3. Objet extraDetails

extraDetails	Format
ipAddress L'adresse IP de l'acheteur.	string ans40

Tableau 186 : Objet extraDetails

tokenResponse

L'objet **tokenResponse** permet d'obtenir des informations sur la date de création et/ou de résiliation d'un alias.

tokenResponse	Format
creationDate Date de création de l'alias.	dateTime ans..40
cancellationDate Date de résiliation de l'alias.	dateTime ans..40

Tableau 187 : Objet tokenResponse

7.5. Résilier un alias (identifiant) 'cancelToken'

L'opération **cancelToken** permet de désactiver/résilier un alias (identifiant) permettant d'effectuer des paiements.

Remarque :

Un alias résilié / désactivé suite à cette opération résilie tous les abonnements liés à cet alias.

Requête à envoyer

La requête **cancelToken** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **cancelToken** prend en entrée un objet de type **cancelToken**.

Le type **cancelToken** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Seul l'attribut **submissionDate** peut être valorisé si besoin. Cet attribut est facultatif.

commonRequest		
Attribut	Requis	Format
submissionDate Date et heure UTC de la résiliation exprimée au format ISO 8601 définit par W3C. <i>Exemple : 2016-07-16T19:20:00Z</i> Si la valeur de cet attribut est trop éloignée de l'heure actuelle, la requête sera rejetée (code d'erreur 13).		dateTime ans..40

Tableau 188 : Objet commonRequest

Les attributs **paymentSource**, **contractNumber** et **comment** ne sont pas aujourd'hui pris en considération pour cette opération.

queryRequest

L'objet **queryRequest** permet d'interroger un alias (identifiant de compte) pour en connaître ses différents attributs.

Seul l'attribut **paymentToken** est indispensable pour cette opération.

queryRequest		
Attribut	Requis	Format
paymentToken Alias (paiement par identifiant).	✓	string ans..64

Tableau 189 : Objet queryRequest

Réponse en retour

La réponse à l'opération **cancelToken** est faite par la plateforme de paiement suite à une demande de désactivation d'un identifiant (token en anglais) permettant d'effectuer des paiements.

Elle est constituée d'un HEADER et d'un BODY de type **cancelTokenResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **cancelTokenResponse** est la suivante :

Nom	Type
cancelTokenResult	cancelTokenResult

La structure du message **cancelTokenResult** est la suivante :

Objet	Type
commonResponse	commonResponse

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 190 : Objet commonResponse

7.6. Réactiver un alias (identifiant) 'reactivateToken'

L'opération **reactivateToken** permet de réactiver un alias (identifiant).

Requête à envoyer

La requête **reactivateToken** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **reactivateToken** prend en entrée un objet de type **reactivateToken**.

Le type **reactivateToken** est composé de l'objet suivant :

Objet	Format	Requis
queryRequest	queryRequest	✓

queryRequest

L'objet **queryRequest** permet d'interroger un alias (identifiant de compte) pour en connaître ses différents attributs.

Seul l'attribut **paymentToken** est indispensable pour cette opération.

queryRequest		
Attribut	Requis	Format
paymentToken Alias (paiement par identifiant).	✓	string ans..64

Tableau 191 : Objet queryRequest

Réponse en retour

La réponse à l'opération **reactivateToken** est faite par la plateforme de paiement suite à une réactivation d'un identifiant/alias (token en anglais).

Elle est constituée d'un HEADER et d'un BODY de type **reactivateTokenResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **reactivateTokenResponse** est la suivante :

Nom	Type
reactivateTokenResult	reactivateTokenResult

La structure du message **reactivateTokenResult** est la suivante :

Objet	Type
commonResponse	commonResponse

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 192 : Objet commonResponse

7.7. Réaliser des paiements récurrents (abonnements)

'createSubscription'

L'opération **createSubscription** permet de réaliser des paiements récurrents (abonnements). Elle nécessite l'utilisation d'un alias déjà existant et valide.

Cet alias peut être créé via :

- l'opération **createToken**,
ou
- par formulaire (voir **Guide d'implémentation du formulaire de paiement**).

Lorsque l'opération **createSubscription** est créée, la plateforme de paiement traite automatiquement les échéances à réaliser. Pour être notifié du résultat d'une échéance, le marchand doit paramétrer la règle **URL de notification à la création d'un abonnement** depuis son Back Office (voir chapitre **Paramétrer les notifications** du **Guide d'implémentation du formulaire de paiement...**).

Erreurs fréquentes lors de la création de paiements récurrents :

- L'alias (identifiant) fourni n'existe pas, est résilié ou est suspendu.
- La date d'effet (attribut **effectDate** de l'objet **subscriptionRequest**) se situe dans le passé.
Un code d'erreur sera renvoyé.

Requête à envoyer

La requête **createSubscription** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **createSubscription** prend en entrée un objet de type **createSubscription**.

Le type **createSubscription** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
orderRequest	orderRequest	
subscriptionRequest	subscriptionRequest	✓
cardRequest	cardRequest	✓

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Il possède les attributs suivants :

commonRequest		
Attribut	Requis	Format
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• EC pour le commerce électronique. Paramètre utilisé par défaut si aucune valeur est renseignée ou si différente des valeurs possibles.• MOTO pour une commande par courrier ou téléphone.• CC pour un centre d'appel.• OTHER pour un autre canal de vente. Seule la valeur EC permet de créer une transaction avec 3D Secure. Les autres valeurs ne doivent être utilisées que pour de la vente à distance, pour laquelle le 3D Secure n'est pas applicable.		string
submissionDate Date et heure UTC de la transaction exprimée au format ISO 8601 défini par W3C. <i>Exemple : 2016-07-16T19:20:00Z</i> Si la valeur de cet attribut est trop éloignée de l'heure actuelle, la requête sera rejetée (code d'erreur 13).		dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé. Si ce champ est renseigné, veillez à utiliser le bon contrat en fonction du réseau de la carte. Par exemple, un contrat CB ne peut être utilisé pour une transaction AMEX.		string
comment Commentaire libre.		string

Tableau 193 : Objet CommonRequest

orderRequest

L'objet **orderRequest** permet de transmettre des informations liées à la commande.

Il est composé de l'attribut suivant :

orderRequest		
Attribut	Requis	Format
orderId Référence de la commande.		string an..64
extInfo Champs personnalisables permettant d'ajouter des données supplémentaires (champ supplémentaire qui sera persisté dans la transaction et sera retourné dans la réponse). L'attribut extInfo est composé de sous objets : <ul style="list-style-type: none">• key : nom de la donnée. Son format est "string".• value : valeur de la donnée. Son format est "string". <i>Exemple : <extInfo><key>keyData</key><value>valuedata</value></extInfo></i>		extInfo

Tableau 194 : Objet orderRequest

subscriptionRequest

L'objet **subscriptionRequest** permet de transmettre des informations liées à l'abonnement.

Il possède les attributs suivants :

subscriptionRequest		
Attribut	Requis	Format
effectDate Date d'effet au format W3C. <i>Exemple</i> : 2016-07-16T19:20Z La date ne peut pas être dans le passé.	✓	dateTime ans..40
amount Montant de l'abonnement dans sa plus petite unité monétaire.	✓	n..12
currency Code de la devise (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF)).	✓	n3
initialAmount Montant des échéances de l'abonnement (dans sa plus petite unité monétaire) pour la ou les premières échéances si ces dernières sont différentes du montant de l'abonnement amount .	✓	n..12
initialAmountNumber Nombre d'échéances auxquelles il faut appliquer le montant initialAmount . Cet attribut devient obligatoire si initialAmount est valorisé.	✓	int
rrule Description de la règle de l'abonnement. La valeur attendue dans cet attribut est une chaîne de caractères suivant la spécification iCalendar , ou Internet Calendar, décrite dans la RFC5545 (voir http://tools.ietf.org/html/rfc5545). Pour des raisons techniques, il n'est pas possible de définir des périodes d'abonnement inférieures à une journée. Les mots clés "SECONDLY" / "MINUTELY" / "HOURLY" ne sont donc pas pris en compte. Exemples : <ul style="list-style-type: none">Pour définir des échéances de paiement ayant lieu le dernier jour de chaque mois, pendant 12 mois, la règle s'écrit : RRULE:FREQ=MONTHLY;BYMONTHDAY=28,29,30,31;BYSETPOS=-1;COUNT=12 Cette règle signifie que si le mois courant ne contient pas de 31, alors le moteur prendra en compte le 30. Si le 30 n'existe pas, alors il prendra en compte le 29 et ainsi de suite jusqu'au 28. Une autre version de cette règle : RRULE:FREQ=MONTHLY;COUNT=5;BYMONTHDAY=-1Pour définir des échéances de paiement ayant lieu le 10 de chaque mois, pendant 12 mois, alors la règle d'abonnement s'écrit de la manière suivante : RRULE:FREQ=MONTHLY;COUNT=12;BYMONTHDAY=10Pour définir des échéances de paiement ayant lieu chaque trimestre, jusqu'au 31/12/2016 : RRULE:FREQ=YEARLY;BYMONTHDAY=-1;BYMONTH=1,4,7,10;UNTIL=20161231 Les échéances auront lieu chaque 1er de janvier, avril, juillet et octobre. Leur nombre total dépend de la date d'effet de l'abonnement (voir paramètre vads_sub_effect_date). Pour plus de détails et d'exemples vous pouvez consulter le site http://recurrence.sourceforge.net/.	✓	string
subscriptionId Identifiant de l'abonnement		string
description Description de l'abonnement.		string

Tableau 195 : Objet subscriptionRequest

cardRequest

L'objet **cardRequest** permet de transmettre les informations sur la carte de paiement.

Seul l'attribut **paymentToken** doit être valorisé.


cardRequest		
Attribut	Requis	Format
paymentToken Identifiant unique (alias) associé à un moyen de paiement. <ul style="list-style-type: none">• Soit cet identifiant a été généré par la plateforme.• Soit cet identifiant a été généré par le site marchand.	 pour le paiement par identifiant	string ans..64

Tableau 196 : Objet cardRequest

Exemple de requête à envoyer

Exemple de requête à envoyer

La requête ci-dessous est un exemple pour illustrer un abonnement avec identifiant.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:v5="http://v5.ws.vads.lyra.com/">
  <soap:Header xmlns:soapHeader="http://v5.ws.vads.lyra.com/Header">
    <soapHeader:shopId>12345678</soapHeader:shopId>
    <soapHeader:requestId>4e977101-9551-4dd8-9296-a4fdad0b5fe4</soapHeader:requestId>
    <soapHeader:timestamp>2015-04-01T12:28:49Z</soapHeader:timestamp>
    <soapHeader:mode>TEST</soapHeader:mode>
    <soapHeader:authToken>rk/ufZhrnsq2yrJMhehRWu9UQ9jbU9RSt2MsP0czeIA=</soapHeader:authToken>
  </soap:Header>
  <soap:Body>
    <v5:createSubscription>
      <commonRequest>
        <paymentSource>EC</paymentSource>
        <submissionDate>2015-04-01T12:28:49Z</submissionDate>
      </commonRequest>
      <orderRequest>
        <orderId>TEST-ORDER</orderId>
      </orderRequest>
      <subscriptionRequest>
        <subscriptionId>TEST-SUBSCRIPTION-01</subscriptionId>
        <effectDate>2015-04-01T12:28:49Z</effectDate>
        <amount>10</amount>
        <currency>953</currency>
        <initialAmount>10</initialAmount>
        <initialAmountNumber>100</initialAmountNumber>
        <rrule>RRULE:FREQ=MONTHLY;COUNT=10;BYMONTHDAY=10</rrule>
      </subscriptionRequest>
      <cardRequest>
        <paymentToken>cb059d56f8564674bc139a373a8daebb</paymentToken>
      </cardRequest>
    </v5:createSubscription>
  </soap:Body>
</soap:Envelope>
```

Réponse en retour

La réponse à l'opération **createSubscription** est faite par la plateforme de paiement à une demande de création d'un abonnement.

Elle est constituée d'un HEADER et d'un BODY de type **createSubscriptionResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **createSubscriptionResponse** est la suivante :

Nom	Type
createSubscriptionResult	createSubscriptionResult

La structure du message **createSubscriptionResult** est la suivante :

Objet	Type
commonResponse	commonResponse
subscriptionResponse	subscriptionResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque : les objets **paymentResponse**, **orderResponse**, **cardResponse**, **captureResponse**, **authorizationResponse**, **customerResponse**, **markResponse**, **threeDSResponse**, **extraResponse** et **fraudManagementResponse** ne sont pas pris en considération dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

Les attributs contenant une valeur dans la réponse sont les suivants :

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 197 : Objet commonResponse

Les attributs **shopId**, **paymentSource**, **submissionDate**, **contractNumber** et **paymentToken** ne sont pas valorisés dans cette opération.

subscriptionResponse

L'objet **subscriptionResponse** détaille l'ensemble des informations constituant un abonnement. Dans cette opération, seul l'attribut **subscriptionId** est retourné si ce dernier a été envoyé dans la requête.

subscriptionResponse	Format
subscriptionId Identifiant de l'abonnement	string

Tableau 198 : Objet subscriptionResponse

Exemple de réponse

Exemple de réponse :

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <shopId xmlns="http://v5.ws.vads.lyra.com/Header/">12345678</shopId>
    <requestId xmlns="http://v5.ws.vads.lyra.com/Header/">4e977101-9551-4dd8-9296-
a4fdad0b5fe4</requestId>
    <timestamp xmlns="http://v5.ws.vads.lyra.com/Header/">2015-04-01T12:28:49Z</timestamp>
    <mode xmlns="http://v5.ws.vads.lyra.com/Header/">TEST</mode>
    <authToken xmlns="http://v5.ws.vads.lyra.com/
Header/">cOS5ykWUf7lerOaIrAmkfmNFD7ZbqvEWHiqUEmIngUU=</authToken>
  </env:Header>
  <soap:Body>
    <ns2:createSubscriptionResponse xmlns:ns2="http://v5.ws.vads.lyra.com/">
      <createSubscriptionResult>
        <requestId>4e977101-9551-4dd8-9296-a4fdad0b5fe4</requestId>
        <commonResponse>
          <responseCode>0</responseCode>
          <responseCodeDetail>Action successfully completed</responseCodeDetail>
        </commonResponse>
        <authorizationResponse/>
      </createSubscriptionResult>
    </ns2:createSubscriptionResponse>
  </soap:Body>
</soap:Envelope>
```

7.8. Modifier un abonnement 'updateSubscription'

L'opération **updateSubscription** permet de modifier :

- Les données relatives de l'acheteur.
- Les échéances de paiement : un montant, une devise, une date d'échéance, un statut, etc.

Cette opération ne peut être appelée si la date d'effet est atteinte.

Requête à envoyer

La requête **updateSubscription** est constituée d'un HEADER et d'un BODY.

• HEADER

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

• BODY

L'opération **updateSubscription** prend en entrée un objet de type **updateSubscription**.

Le type **updateSubscription** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓
subscriptionRequest	subscriptionRequest	✓
paymentRequest	paymentRequest	

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Plusieurs attributs, facultatifs, peuvent être spécifiés dans la requête.

commonRequest		
Attribut	Requis	Format
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none">• EC pour le commerce électronique. Paramètre utilisé par défaut si aucune valeur est renseignée ou si différente des valeurs possibles.• MOTO pour une commande par courrier ou téléphone.• CC pour un centre d'appel.• OTHER pour un autre canal de vente. Seule la valeur EC permet de créer une transaction avec 3D Secure. Les autres valeurs ne doivent être utilisées que pour de la vente à distance, pour laquelle le 3D Secure n'est pas applicable.		string
contractNumber Numéro de contrat commerçant utilisé. Si ce champ est renseigné, veuillez à utiliser le bon contrat en fonction du réseau de la carte. Par exemple, un contrat CB ne peut être utilisé pour une transaction AMEX.		string

Tableau 199 : Objet commonRequest

L'attribut **comment** n'est pas aujourd'hui pris en considération pour cette opération.

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

Il possède les attributs suivants :

queryRequest		
Attribut	Requis	Format
paymentToken Alias (paiement par identifiant).	✓	string ans..64
subscriptionId Identifiant de l'abonnement.	✓	string

Tableau 200 : Objet queryRequest

subscriptionRequest

L'objet **subscriptionRequest** permet de transmettre des informations liées à l'abonnement.

Il possède les attributs suivants :

subscriptionRequest		
Attribut	Requis	Format
effectDate Date d'effet au format W3C. <i>Exemple</i> : 2016-07-16T19:20Z La date ne peut pas être dans le passé.	✓	dateTime ans..40
amount Montant de l'abonnement dans sa plus petite unité monétaire.	✓	n..12
currency Code de la devise (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF)).	✓	n3
initialAmount Montant des échéances de l'abonnement (dans sa plus petite unité monétaire) pour la ou les premières échéances si ces dernières sont différentes du montant de l'abonnement amount .	✓	n..12
initialAmountNumber Nombre d'échéances auxquelles il faut appliquer le montant initialAmount . Cet attribut devient obligatoire si initialAmount est valorisé.	✓	int
rrule Description de la règle de l'abonnement. La valeur attendue dans cet attribut est une chaîne de caractères suivant la spécification iCalendar , ou Internet Calendar, décrite dans la RFC5545 (voir http://tools.ietf.org/html/rfc5545). Pour des raisons techniques, il n'est pas possible de définir des périodes d'abonnement inférieures à une journée. Les mots clés "SECONDLY" / "MINUTELY" / "HOURLY" ne sont donc pas pris en compte. Exemples : <ul style="list-style-type: none">Pour définir des échéances de paiement ayant lieu le dernier jour de chaque mois, pendant 12 mois, la règle s'écrit : RRULE:FREQ=MONTHLY;BYMONTHDAY=28,29,30,31;BYSETPOS=-1;COUNT=12 Cette règle signifie que si le mois courant ne contient pas de 31, alors le moteur prendra en compte le 30. Si le 30 n'existe pas, alors il prendra en compte le 29 et ainsi de suite jusqu'au 28. Une autre version de cette règle : RRULE:FREQ=MONTHLY;COUNT=5;BYMONTHDAY=-1Pour définir des échéances de paiement ayant lieu le 10 de chaque mois, pendant 12 mois, alors la règle d'abonnement s'écrit de la manière suivante : RRULE:FREQ=MONTHLY;COUNT=12;BYMONTHDAY=10Pour définir des échéances de paiement ayant lieu chaque trimestre, jusqu'au 31/12/2016 : RRULE:FREQ=YEARLY;BYMONTHDAY=-1;BYMONTH=1,4,7,10;UNTIL=20161231 Les échéances auront lieu chaque 1er de janvier, avril, juillet et octobre. Leur nombre total dépend de la date d'effet de l'abonnement (voir paramètre vads_sub_effect_date).	✓	string

subscriptionRequest		
Attribut	Requis	Format
Pour plus de détails et d'exemples vous pouvez consulter le site http://recurrence.sourceforge.net/ .		
subscriptionId Identifiant de l'abonnement		string
description Description de l'abonnement.		string

Tableau 201 : Objet subscriptionRequest

Réponse en retour

La réponse à l'opération **updateSubscription** est faite par la plateforme de paiement à une demande de modification sur un abonnement.

Elle est constituée d'un HEADER et d'un BODY de type **updateSubscriptionResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **updateSubscriptionResponse** est la suivante :

Nom	Type
updateSubscriptionResult	updateSubscriptionResult

La structure du message **updateSubscriptionResult** est la suivante :

Objet	Type
commonResponse	commonResponse

Remarque : les objets **paymentResponse**, **orderResponse**, **cardResponse**, **authorizationResponse**, **captureResponse**, **customerResponse**, **markResponse**, **threeDSResponse**, **extraResponse**, **subscriptionResponse** et **fraudManagementResponse** ne sont pas pris en considération dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 202 : Objet commonResponse

7.9. Récupérer le détail d'un abonnement 'getSubscriptionDetails'

L'opération **getSubscriptionDetails** permet de rechercher un abonnement pour en connaître ses différents attributs.

Requête à envoyer

La requête **getSubscriptionDetails** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **getSubscriptionDetails** prend en entrée un objet de type **getSubscriptionDetails**.

Le type **getSubscriptionDetails** est composé de l'objet suivant :

Objet	Format	Requis
queryRequest	queryRequest	✓

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

Il possède les attributs suivants :

queryRequest		
Attribut	Requis	Format
paymentToken Alias (paiement par identifiant).	✓	string ans..64
subscriptionId Identifiant de l'abonnement.	✓	string

Tableau 203 : Objet queryRequest

Réponse en retour

La réponse à l'opération **getSubscriptionDetails** est faite par la plateforme de paiement suite à une demande d'information sur un abonnement.

Elle est constituée d'un HEADER et d'un BODY de type **getSubscriptionDetailsResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **getSubscriptionDetailsResponse** est la suivante :

Nom	Type
getSubscriptionDetailsResult	getSubscriptionDetailsResult

La structure du message **getSubscriptionDetailsResult** est la suivante :

Objet	Type
commonResponse	commonResponse
orderResponse	orderResponse
subscriptionResponse	subscriptionResponse

Les données retournées dans la réponse dépendent des objets et attributs envoyés dans la requête.

Cependant, quelle que soit l'opération, l'attribut **responseCode** de l'objet **CommonResponse** doit avant tout être analysé :

- La valeur 0 indique que l'opération s'est déroulée avec succès.
- Une valeur différente de 0 implique une analyse de l'attribut **responseCodeDetails**. Ce dernier précise l'origine de l'erreur.

Remarque : les objets **paymentResponse**, **cardResponse**, **authorizationResponse**, **captureResponse**, **customerResponse**, **markResponse**, **extraResponse**, **fraudManagementResponse** et **tokenResponse** ne sont pas pris en considération dans la réponse.

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

Les attributs contenant une valeur dans la réponse sont les suivants :

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string
shopId Identifiant de la boutique.	n8

commonResponse	Format
paymentSource Origine de la transaction. Les valeurs possibles sont : <ul style="list-style-type: none"> • EC pour le commerce électronique. • MOTO pour une commande par e-mail ou téléphone. • CC pour un centre d'appel. • OTHER pour un autre canal de vente. 	string (enum)
submissionDate Date et heure UTC de la transaction exprimée au format W3C (exemple : 2016-07-16T19:20:00Z).	dateTime ans..40
contractNumber Numéro de contrat commerçant utilisé.	string
paymentToken Alias ou Identifiant du compte acheteur (paiement par identifiant).	string

Tableau 204 : Objet commonResponse

L'attribut **transactionStatusLabel**, n'est pas valorisé dans cette opération.

orderResponse

L'objet **orderResponse** détaille la commande.

orderResponse	Format
orderId Référence de la commande.	string an..64
extInfo Données personnalisées retournées en fonction des besoins. <u>Exemple</u> : extInfo.key, extInfo.value <ul style="list-style-type: none">• key : nom de la donnée (son format est "string").• value : valeur de la donnée (son format est "string").	extInfo

Tableau 205 : Objet orderResponse

subscriptionResponse

L'objet **subscriptionResponse** détaille l'ensemble des informations constituant un abonnement.

subscriptionResponse	Format
subscriptionId Identifiant de l'abonnement	string
effectDate Date d'effet au format W3C. <i>Exemple</i> : 2016-07-16T19:20Z La date ne peut pas être dans le passé.	dateTime ans..40
cancelDate Date d'annulation d'une échéance de paiement.	dateTime ans..40
initialAmount Montant des échéances de l'abonnement (dans sa plus petite unité monétaire) pour la ou les premières échéances si ces dernières sont différentes du montant de l'abonnement amount .	
rrule Description de la règle de l'abonnement. Exemple: <ul style="list-style-type: none">• RRULE:FREQ=MONTHLY;COUNT=12;BYMONTHDAY=10 Echéances de paiement ayant lieu le 10 de chaque mois, pendant 12 mois.	string
description Description de l'abonnement.	string
initialAmountNumber Numéro de l'échéance du montant initial de l'abonnement. Cet attribut devient obligatoire si initialAmount est valorisé.	int
pastPaymentsNumber Nombre de versements antérieurs.	int
totalPaymentsNumber Nombre total de paiements.	int
amount Montant de l'abonnement dans sa plus petite unité monétaire.	n..12
currency Code de la devise (Code monnaie ISO 4217 Ex : 953 pour le Franc CFP (XPF)).	n3

Tableau 206 : Objet subscriptionResponse

7.10. Annuler un abonnement 'cancelSubscription'

L'opération **cancelSubscription** permet de désactiver/résilier un abonnement à une date précise.

Remarque :

La requête **cancelSubscription** doit être envoyée au minimum 8 jours avant la date d'échéance de l'abonnement.

Requête à envoyer

La requête **cancelSubscription** est constituée d'un HEADER et d'un BODY.

- **HEADER**

Valorisez le HEADER afin de transmettre la valeur des attributs **shopId**, **requestId**, **timestamp**, **mode** et **authToken** (voir chapitre L'en-tête).

- **BODY**

L'opération **cancelSubscription** prend en entrée un objet de type **cancelSubscription**.

Le type **cancelSubscription** est constitué des paramètres suivants :

Objet	Format	Requis
commonRequest	commonRequest	
queryRequest	queryRequest	✓

commonRequest

L'objet **commonRequest** permet de transmettre des informations générales sur une opération.

Seul l'attribut **submissionDate** peut être valorisé si besoin. Cet attribut est facultatif.

commonRequest		
Attribut	Requis	Format
submissionDate Date et heure UTC de la résiliation exprimée au format ISO 8601 défini par W3C. <i>Exemple : 2016-07-16T19:20:00Z</i> Si la valeur de cet attribut est trop éloignée de l'heure actuelle, la requête sera rejetée (code d'erreur 13).		dateTime ans..40

Tableau 207 : Objet commonRequest

Les attributs **paymentSource**, **contractNumber** et **comment** ne sont pas aujourd'hui pris en considération pour cette opération.

queryRequest

L'objet **queryRequest** permet d'interroger une transaction pour en connaître ses différents attributs.

Il possède les attributs suivants :

queryRequest		
Attribut	Requis	Format
paymentToken Alias (paiement par identifiant).	✓	string ans..64
subscriptionId Identifiant de l'abonnement.	✓	string

Tableau 208 : Objet queryRequest

Réponse en retour

La réponse à l'opération **cancelSubscription** est faite par la plateforme de paiement suite à une demande de désactivation d'un abonnement.

Elle est constituée d'un HEADER et d'un BODY de type **cancelSubscriptionResponse**.

- **HEADER**

Le HEADER est transmis par la plateforme de paiement.

Vérifiez la valeur du jeton d'authentification (voir chapitre Vérifier l'en-tête SOAP dans la réponse).

- **BODY**

La structure du message **cancelSubscriptionResponse** est la suivante :

Nom	Type
cancelSubscriptionResult	cancelSubscriptionResult

La structure du message **cancelSubscriptionResult** est la suivante :

Objet	Type
commonResponse	commonResponse

commonResponse

L'objet **commonResponse** permet d'obtenir des informations générales sur une opération.

commonResponse	Format
responseCode Référez-vous au chapitre Gérer les erreurs applicatives . Premier attribut à analyser quelle que soit l'opération. <ul style="list-style-type: none">• La valeur 0 indique que l'opération s'est déroulée avec succès.• Une valeur différente de 0 implique une analyse de l'attribut responseCodeDetails. Ce dernier précise l'origine de l'erreur.	n..2
responseCodeDetail Détail de l'erreur si l'attribut responseCode est différent de 0. Reportez-vous au chapitre Gérer les erreurs applicatives pour plus d'informations.	string

Tableau 209 : Objet commonResponse

8. EXEMPLES EN PHP

Pour tous les exemples en PHP proposés en annexes, deux fichiers doivent être créés en amont pour être utilisés dans les requêtes. Ils sont communs à toutes les opérations.

- Un fichier "v5.php" pour la définition des objets :

```
<?php

class commonRequest {
    public $paymentSource; // string
    public $submissionDate; // dateTime
    public $contractNumber; // string
    public $comment; // string
}

class commonResponse {
    public $responseCode; // int
    public $responseCodeDetail; // string
    public $transactionStatusLabel; // string
    public $shopId; // string
    public $paymentSource; // string
    public $submissionDate; // dateTime
    public $contractNumber; // string
    public $paymentToken; // string
}

class cardRequest {
    public $number; // string
    public $scheme; // string
    public $expiryMonth; // int
    public $expiryYear; // int
    public $cardSecurityCode; // string
    public $cardHolderBirthDay; // dateTime
    public $paymentToken; // string
}

class customerRequest {
    public $billingDetails; // billingDetailsRequest
    public $shippingDetails; // shippingDetailsRequest
    public $extraDetails; // extraDetailsRequest
}

class billingDetailsRequest {
    public $reference; // string
    public $title; // string
    public $type; // custStatus
    public $firstName; // string
    public $lastName; // string
    public $phoneNumber; // string
    public $email; // string
    public $streetNumber; // string
    public $address; // string
    public $district; // string
    public $zipCode; // string
    public $city; // string
    public $state; // string
    public $country; // string
    public $language; // string
    public $cellPhoneNumber; // string
    public $legalName; // string
    public $identityCode; // string
}

class shippingDetailsRequest {
    public $type; // custStatus
    public $firstName; // string
    public $lastName; // string
    public $phoneNumber; // string
    public $streetNumber; // string
    public $address; // string
    public $address2; // string
    public $district; // string
    public $zipCode; // string
    public $city; // string
    public $state; // string
    public $country; // string
    public $deliveryCompanyName; // string
    public $shippingSpeed; // deliverySpeed
    public $shippingMethod; // deliveryType
    public $legalName; // string
    public $identityCode; // string
}
```

```

class extraDetailsRequest {
    public $ipAddress; // string
    public $fingerPrintId; // string
}

class shoppintCartRequest {
    public $insuranceNumber; // long
    public $shippingAmount; // long
    public $taxAmount; // long
    public $cartItemInfo; // cartItemInfo
}

class cartItemInfo {
    public $productLabel; // string
    public $productType; // productType
    public $productRef; // string
    public $productQty ; // int
    public $productAmount; // string
    public $productVat; // string
}

class paymentRequest {
    public $transactionId; // string
    public $amount; // long
    public $currency; // int
    public $expectedCaptureDate; // dateTime
    public $manualValidation; // int
    public $paymentOptionCode; // string
    public $retryUuid; // string
}

class paymentResponse {
    public $transactionId; // string
    public $amount; // long
    public $currency; // int
    public $effectiveAmount; // long
    public $effectiveCurrency; // int
    public $expectedCaptureDate; // dateTime
    public $manualValidation; // int
    public $operationType; // int
    public $creationDate; // dateTime
    public $externalTransactionId; // string
    public $liabilityShift; // string
    public $sequenceNumber; // int
    public $paymentType; // paymentType
    public $paymentError; // int
}

class orderResponse {
    public $orderId; // string
    public $extInfo; // extInfo
}

class extInfo {
    public $key; // string
    public $value; // string
}

class cardResponse {
    public $number; // string
    public $scheme; // string
    public $brand; // string
    public $country; // string
    public $productCode; // string
    public $bankCode; // string
    public $expiryMonth; // int
    public $expiryYear; // int
}

class authorizationResponse {
    public $mode; // string
    public $amount; // long
    public $currency; // int
    public $date; // dateTime
    public $number; // string
    public $result; // int
}

class captureResponse {
    public $date; // dateTime
    public $number; // int
    public $reconciliationStatus; // int
    public $refundAmount; // long
    public $refundCurrency; // int
    public $chargeback; // boolean
}

class customerResponse {

```

```

    public $billingDetails; // billingDetailsResponse
    public $shippingDetails; // shippingDetailsResponse
    public $extraDetails; // extraDetailsResponse
}

class billingDetailsResponse {
    public $reference; // string
    public $title; // string
    public $type; // custStatus
    public $firstName; // string
    public $lastName; // string
    public $phoneNumber; // string
    public $email; // string
    public $streetNumber; // string
    public $address; // string
    public $district; // string
    public $zipCode; // string
    public $city; // string
    public $state; // string
    public $country; // string
    public $language; // string
    public $cellPhoneNumber; // string
    public $legalName; // string
}

class shippingDetailsResponse {
    public $type; // custStatus
    public $firstName; // string
    public $lastName; // string
    public $phoneNumber; // string
    public $streetNumber; // string
    public $address; // string
    public $address2; // string
    public $district; // string
    public $zipCode; // string
    public $city; // string
    public $state; // string
    public $country; // string
    public $deliveryCompanyName; // string
    public $shippingSpeed; // deliverySpeed
    public $shippingMethod; // deliveryType
    public $legalName; // string
    public $identityCode; // string
}

class extraDetailsResponse {
    public $ipAddress; // string
}

class markResponse {
    public $amount; // long
    public $currency; // int
    public $date; // dateTime
    public $number; // string
    public $result; // int
}

class threeDSResponse {
    public $authenticationRequestData; // authenticationRequestData
    public $authenticationResultData; // authenticationResultData
}

class authenticationRequestData {
    public $threeDSAcctId; // string
    public $threeDSAcUrl; // string
    public $threeDSBrand; // string
    public $threeDSEncodedPareq; // string
    public $threeDSEnrolled; // string
    public $threeDSRequestId; // string
}

class authenticationResultData {
    public $brand; // string
    public $enrolled; // string
    public $status; // string
    public $eci; // string
    public $xid; // string
    public $cavv; // string
    public $cavvAlgorithm; // string
    public $signValid; // string
    public $transactionCondition; // string
}

class extraResponse {
    public $paymentOptionCode; // string
    public $paymentOptionOccNumber; // int
}

class fraudManagementResponse {

```

```

    public $riskControl; // riskControl
    public $riskAnalysis; // riskAnalysis
    public $riskAssessments; // riskAssessments
}

class shoppingCartResponse {
    public $cartItemInfo; // cartItemInfo
}

class riskControl {
    public $name; // string
    public $result; // string
}

class riskAnalysis {
    public $score; // string
    public $resultCode; // string
    public $status; // vadRiskAnalysisProcessingStatus
    public $requestId; // string
    public $extraInfo; // extInfo
}

class riskAssessments {
    public $results; // string
}

class techRequest {
    public $browserUserAgent; // string
    public $browserAccept; // string
}

class orderRequest {
    public $orderId; // string
    public $extInfo; // extInfo
}

class createPayment {
    public $commonRequest; // commonRequest
    public $threeDSRequest; // threeDSRequest
    public $paymentRequest; // paymentRequest
    public $orderRequest; // orderRequest
    public $cardRequest; // cardRequest
    public $customerRequest; // customerRequest
    public $techRequest; // techRequest
    public $shoppingCartRequest; // shoppingCartRequest
}

class threeDSRequest {
    public $mode; // threeDSMode
    public $requestId; // string
    public $pares; // string
    public $brand; // string
    public $enrolled; // string
    public $status; // string
    public $eci; // string
    public $xid; // string
    public $cavv; // string
    public $algorithm; // string
}

class createPaymentResponse {
    public $createPaymentResult; // createPaymentResult
}

class createPaymentResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $threeDSResponse; // threeDSResponse
    public $extraResponse; // extraResponse
    public $subscriptionResponse; // subscriptionResponse
    public $fraudManagementResponse; // fraudManagementResponse
    public $shoppingCartResponse; // shoppingCartResponse
}

class cancelToken {
    public $commonRequest; // commonRequest
    public $queryRequest; // queryRequest
}

class cancelTokenResponse {
    public $cancelTokenResult; // cancelTokenResult
}

```

```

class cancelTokenResult {
    public $commonResponse; // commonResponse
}

class queryRequest {
    public $uuid; // string
    public $orderId; // string
    public $subscriptionId; // string
    public $paymentToken; // string
}

class wsResponse {
    public $requestId; // string
}

class createToken {
    public $commonRequest; // commonRequest
    public $cardRequest; // cardRequest
    public $customerRequest; // customerRequest
}

class createTokenResponse {
    public $createTokenResult; // createTokenResult
}

class createTokenResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $threeDSResponse; // threeDSResponse
    public $extraResponse; // extraResponse
    public $subscriptionResponse; // subscriptionResponse
    public $fraudManagementResponse; // fraudManagementResponse
    public $shoppingCartResponse; // shoppingCartResponse
}

class subscriptionResponse {
    public $subscriptionId; // string
    public $effectDate; // dateTime
    public $cancelDate; // dateTime
    public $initialAmount; // long
    public $rrule; // string
    public $description; // string
    public $initialAmountNumber; // int
    public $pastPaymentNumber; // int
    public $totalPaymentNumber; // int
    public $amount; // long
    public $currency; // int
}

class getTokenDetails {
    public $queryRequest; // queryRequest
}

class getTokenDetailsResponse {
    public $getTokenDetailsResult; // getTokenDetailsResult
}

class getTokenDetailsResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $subscriptionResponse; // subscriptionResponse
    public $extraResponse; // extraResponse
    public $fraudManagementResponse; // fraudManagementResponse
    public $threeDSResponse; // threeDSResponse
    public $tokenResponse; // tokenResponse
}

class updateSubscription {
    public $commonRequest; // commonRequest
    public $queryRequest; // queryRequest
    public $subscriptionRequest; // subscriptionRequest
}

class subscriptionRequest {
    public $subscriptionId; // string
    public $effectDate; // dateTime
    public $amount; // long

```

```

    public $currency; // int
    public $initialAmount; // long
    public $initialAmountNumber; // int
    public $rrule; // string
    public $description; // string
}

class updateSubscriptionResponse {
    public $updateSubscriptionResult; // updateSubscriptionResult
}

class updateSubscriptionResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $threeDSResponse; // threeDSResponse
    public $extraResponse; // extraResponse
    public $subscriptionResponse; // subscriptionResponse
    public $fraudManagementResponse; // fraudManagementResponse
}

class capturePayment {
    public $settlementRequest; // settlementRequest
}

class settlementRequest {
    public $transactionUuids; // string
    public $commission; // double
    public $date; // dateTime
}

class capturePaymentResponse {
    public $capturePaymentResult; // capturePaymentResult
}

class capturePaymentResult {
    public $commonResponse; // commonResponse
}

class findPayments {
    public $queryRequest; // queryRequest
}

class findPaymentsResponse {
    public $findPaymentsResult; // findPaymentsResult
}

class findPaymentsResult {
    public $commonResponse; // commonResponse
    public $orderResponse; // orderResponse
    public $transactionItem; // transactionItem
}

class transactionItem {
    public $transactionUuid; // string
    public $transactionStatusLabel; // string
    public $amount; // long
    public $currency; // int
    public $expectedCaptureDate; // dateTime
}

class refundPayment {
    public $commonRequest; // commonRequest
    public $paymentRequest; // paymentRequest
    public $queryRequest; // queryRequest
}

class refundPaymentResponse {
    public $refundPaymentResult; // refundPaymentResult
}

class refundPaymentResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $threeDSResponse; // threeDSResponse
    public $extraResponse; // extraResponse
    public $fraudManagementResponse; // fraudManagementResponse
}

```

```

class verifyThreeDSEnrollment {
    public $commonRequest; // commonRequest
    public $paymentRequest; // paymentRequest
    public $cardRequest; // cardRequest
    public $techRequest; // techRequest
}

class verifyThreeDSEnrollmentResponse {
    public $verifyThreeDSEnrollmentResult; // verifyThreeDSEnrollmentResult
}

class verifyThreeDSEnrollmentResult {
    public $commonResponse; // commonResponse
    public $threeDSResponse; // threeDSResponse
}

class reactivateToken {
    public $queryRequest; // queryRequest
}

class reactivateTokenResponse {
    public $reactivateTokenResult; // reactivateTokenResult
}

class reactivateTokenResult {
    public $commonResponse; // commonResponse
}

class createSubscription {
    public $commonRequest; // commonRequest
    public $orderRequest; // orderRequest
    public $subscriptionRequest; // subscriptionRequest
    public $cardRequest; // cardRequest
}

class createSubscriptionResponse {
    public $createSubscriptionResult; // createSubscriptionResult
}

class createSubscriptionResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $threeDSResponse; // threeDSResponse
    public $extraResponse; // extraResponse
    public $subscriptionResponse; // subscriptionResponse
    public $fraudManagementResponse; // fraudManagementResponse
    public $shoppingCartResponse; // shoppingCartResponse
}

class cancelSubscription {
    public $commonRequest; // commonRequest
    public $queryRequest; // queryRequest
}

class cancelSubscriptionResponse {
    public $cancelSubscriptionResult; // cancelSubscriptionResult
}

class cancelSubscriptionResult {
    public $commonResponse; // commonResponse
}

class updatePayment {
    public $commonRequest; // commonRequest
    public $queryRequest; // queryRequest
    public $paymentRequest; // paymentRequest
}

class updatePaymentResponse {
    public $updatePaymentResult; // updatePaymentResult
}

class updatePaymentResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $threeDSResponse; // threeDSResponse
}

```

```

    public $extraResponse; // extraResponse
    public $subscriptionResponse; // subscriptionResponse
    public $fraudManagementResponse; // fraudManagementResponse
}

class validatePayment {
    public $commonRequest; // commonRequest
    public $queryRequest; // queryRequest
}

class validatePaymentResponse {
    public $validatePaymentResult; // validatePaymentResult
}

class validatePaymentResult {
    public $commonResponse; // commonResponse
}

class cancelPayment {
    public $commonRequest; // commonRequest
    public $queryRequest; // queryRequest
}

class cancelPaymentResponse {
    public $cancelPaymentResult; // cancelPaymentResult
}

class cancelPaymentResult {
    public $commonResponse; // commonResponse
}

class checkThreeDSAuthentication {
    public $commonRequest; // commonRequest
    public $threeDSRequest; // threeDSRequest
}

class checkThreeDSAuthenticationResponse {
    public $checkThreeDSAuthenticationResult; // checkThreeDSAuthenticationResult
}

class checkThreeDSAuthenticationResult {
    public $commonResponse; // commonResponse
    public $threeDSResponse; // threeDSResponse
}

class getPaymentDetails {
    public $queryRequest; // queryRequest
}

class getPaymentDetailsResponse {
    public $getPaymentDetailsResult; // getPaymentDetailsResult
}

class getPaymentDetailsResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $subscriptionResponse; // subscriptionResponse
    public $extraResponse; // extraResponse
    public $fraudManagementResponse; // fraudManagementResponse
    public $threeDSResponse; // threeDSResponse
    public $tokenResponse; // tokenResponse
}

class duplicatePayment {
    public $commonRequest; // commonRequest
    public $paymentRequest; // paymentRequest
    public $orderRequest; // orderRequest
    public $queryRequest; // queryRequest
}

class duplicatePaymentResponse {
    public $duplicatePaymentResult; // duplicatePaymentResult
}

class duplicatePaymentResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse

```

```

    public $threeDSResponse; // threeDSResponse
    public $extraResponse; // extraResponse
    public $fraudManagementResponse; // fraudManagementResponse
}

class updateToken {
    public $commonRequest; // commonRequest
    public $queryRequest; // queryRequest
    public $cardRequest; // cardRequest
    public $customerRequest; // customerRequest
}

class updateTokenResponse {
    public $updateTokenResult; // updateTokenResult
}

class updateTokenResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $threeDSResponse; // threeDSResponse
    public $extraResponse; // extraResponse
    public $subscriptionResponse; // subscriptionResponse
    public $fraudManagementResponse; // fraudManagementResponse
}

class getPaymentUuid {
    public $legacyTransactionKeyRequest; // legacyTransactionKeyRequest
}

class legacyTransactionKeyRequest {
    public $transactionId; // string
    public $sequenceNumber; // int
    public $creationDate; // dateTime
}

class getPaymentUuidResponse {
    public $legacyTransactionKeyResult; // legacyTransactionKeyResult
}

class legacyTransactionKeyResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
}

class getSubscriptionDetails {
    public $queryRequest; // queryRequest
}

class getSubscriptionDetailsResponse {
    public $getSubscriptionDetailsResult; // getSubscriptionDetailsResult
}

class getSubscriptionDetailsResult {
    public $commonResponse; // commonResponse
    public $paymentResponse; // paymentResponse
    public $orderResponse; // orderResponse
    public $cardResponse; // cardResponse
    public $authorizationResponse; // authorizationResponse
    public $captureResponse; // captureResponse
    public $customerResponse; // customerResponse
    public $markResponse; // markResponse
    public $subscriptionResponse; // subscriptionResponse
    public $extraResponse; // extraResponse
    public $fraudManagementResponse; // fraudManagementResponse
    public $threeDSResponse; // threeDSResponse
    public $tokenResponse; // tokenResponse
}

class paymentType {
    const SINGLE = 'SINGLE';
    const INSTALLMENT = 'INSTALLMENT';
    const SPLIT = 'SPLIT';
    const SUBSCRIPTION = 'SUBSCRIPTION';
    const RETRY = 'RETRY';
}

class custStatus {
    const PRIVATE = 'PRIVATE';
    const COMPANY = 'COMPANY';
}

class deliverySpeed {
    const STANDARD = 'STANDARD';
}

```

```

    const EXPRESS = 'EXPRESS';
}

class deliveryType {
    const RECLAIM_IN_SHOP = 'RECLAIM_IN_SHOP';
    const RELAY_POINT = 'RELAY_POINT';
    const RECLAIM_IN_STATION = 'RECLAIM_IN_STATION';
    const PACKAGE_DELIVERY_COMPANY = 'PACKAGE_DELIVERY_COMPANY';
    const ETICKET = 'ETICKET';
}

class vadRiskAnalysisProcessingStatus {
    const P_TO_SEND = 'P_TO_SEND';
    const P_SEND_KO = 'P_SEND_KO';
    const P_PENDING_AT_ANALYZER = 'P_PENDING_AT_ANALYZER';
    const P_SEND_OK = 'P_SEND_OK';
    const P_MANUAL = 'P_MANUAL';
    const P_SKIPPED = 'P_SKIPPED';
    const P_SEND_EXPIRED = 'P_SEND_EXPIRED';
}

class threeDSMode {
    const DISABLED = 'DISABLED';
    const ENABLED_CREATE = 'ENABLED_CREATE';
    const ENABLED_FINALIZE = 'ENABLED_FINALIZE';
    const MERCHANT_3DS = 'MERCHANT_3DS';
}

class productType {
    const FOOD_AND_GROCERY = 'FOOD_AND_GROCERY';
    const AUTOMOTIVE = 'AUTOMOTIVE';
    const ENTERTAINMENT = 'ENTERTAINMENT';
    const HOME_AND_GARDEN = 'HOME_AND_GARDEN';
    const HOME_APPLIANCE = 'HOME_APPLIANCE';
    const AUCTION_AND_GROUP_BUYING = 'AUCTION_AND_GROUP_BUYING';
    const FLOWERS_AND_GIFTS = 'FLOWERS_AND_GIFTS';
    const COMPUTER_AND_SOFTWARE = 'COMPUTER_AND_SOFTWARE';
    const HEALTH_AND_BEAUTY = 'HEALTH_AND_BEAUTY';
    const SERVICE_FOR_INDIVIDUAL = 'SERVICE_FOR_INDIVIDUAL';
    const SERVICE_FOR_BUSINESS = 'SERVICE_FOR_BUSINESS';
    const SPORTS = 'SPORTS';
    const CLOTHING_AND_ACCESSORIES = 'CLOTHING_AND_ACCESSORIES';
    const TRAVEL = 'TRAVEL';
    const HOME_AUDIO_PHOTO_VIDEO = 'HOME_AUDIO_PHOTO_VIDEO';
    const TELEPHONY = 'TELEPHONY';
}
?>

```

- Un fichier **"function.php"** pour les fonctions :

```

<?php
function getAuthToken ($requestId , $timestamp, $key)
{
    $data = "";
    $data = $requestId.$timestamp;
    $authToken = hash_hmac("sha256", $data, $key, true);
    $authToken = base64_encode ($authToken);
    //var_dump($authToken);
    return $authToken;
}

function gen_uuid() {
    if (function_exists('random_bytes')) {
        // PHP 7
        $data = random_bytes(16);
    } elseif (function_exists('openssl_random_pseudo_bytes')) {
        // PHP 5.3, Open SSL required
        $data = openssl_random_pseudo_bytes(16);
    } else {
        return sprintf(
            '%04x%04x-%04x-%04x-%04x%04x%04x',
            mt_rand(0, 0xffff),
            mt_rand(0, 0xffff),
            mt_rand(0, 0xffff),
            mt_rand(0, 0xffff) | 0x4000,
            mt_rand(0, 0x3fff) | 0x8000,
            mt_rand(0, 0xffff),
            mt_rand(0, 0xffff),
            mt_rand(0, 0xffff)
        );
    }
}

$data[6] = chr(ord($data[6]) & 0x0f | 0x40); // set version to 100
$data[8] = chr(ord($data[8]) & 0x3f | 0x80); // set bits 6 & 7 to 10

return vsprintf('%s%s-%s-%s-%s%s%s', str_split(bin2hex($data), 4));
}

```

```

function setHeaders ($shopId, $requestId, $timestamp, $mode, $authToken, $key, $client) {
    //Création des en-têtes shopId, requestId, timestamp, mode et authToken
    $ns = 'http://v5.ws.vads.lyra.com/Header/';
    $headerShopId = new SOAPHeader ( $ns, 'shopId', $shopId );
    $headerRequestId = new SOAPHeader ( $ns, 'requestId', $requestId);
    $headerTimestamp = new SOAPHeader ( $ns, 'timestamp', $timestamp );
    $headerMode = new SOAPHeader ( $ns, 'mode', $mode );
    $authToken = getAuthToken ($requestId, $timestamp, $key);

    $headerAuthToken = new SOAPHeader ( $ns, 'authToken', $authToken );
    //Ajout des en-têtes dans le SOAP Header
    $headers = array (
        $headerShopId,
        $headerRequestId,
        $headerTimestamp,
        $headerMode,
        $headerAuthToken
    );

    $client->__setSoapHeaders ( $headers );
}

function setJsessionId($client){
    $cookie=$_SESSION['JSESSIONID'];
    $client->__setCookie('JSESSIONID', $cookie);
    return $cookie;
}

/**
 *
 *
 * @param $client
 * @return string $JSESSIONID
 */
function getJsessionId($client){
    //récupération de l'entête de la réponse
    $header=($client->__getLastResponseHeaders());

    if(!preg_match("#JSESSIONID=([A-Za-z0-9\._]+)#",$header, $matches)){
        return "Aucun ID de Session Renvoyé." ; //Ce cas ne devrait jamais se présenter;
        die;
    }

    $JSESSIONID = $matches[1];
    $_SESSION['JSESSIONID'] = $JSESSIONID;
    //print_r($JSESSIONID);

    return $JSESSIONID;
}

function formConstructor ($threeDsAcUrl,$threeDsrequestId,$threeDsEncodedPareq,
$threeDsServerResponseUrl){

$msg= ( '
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>3DS</title>
<script type="text/javascript">
<!--
        function submitForm(){
            document.redirectForm.submit();
        }
    -->
</script>
</head>

<body id="lyra" onLoad="setTimeout('\submitForm()'.\'\',500);">
<div id="container">
<div id="paymentSolutionInfo">
<div id="title">&nbsp;</div>
</div>

<hr class="ensureDivHeight"/>
<br/>
<br/>

<br/>
<br/>
<br/>
<form name="redirectForm" action="'. $threeDsAcUrl. '" method="POST">
    <input type="hidden" name="PaReq" value="'. $threeDsEncodedPareq. '" />
    <input type="hidden" name="TermUrl" value="'. $threeDsServerResponseUrl. '" />
    <input type="hidden" name="MD" value="'. $threeDsrequestId. '" />

    <noscript><input type="submit" name="Go" value="Click to continue"/></noscript>
</form>

```

```
<div id="backToBoutiqueBlock"> </div>
<div id="footer"> </div>
</div>
</body>
</html>'
);

echo $msg;

}

?>
```

8.1. createPayment

Création d'une transaction de paiement avec authentification 3D Secure

Quatre fichiers sont requis pour créer une transaction de paiement avec authentification 3D Secure :

- un fichier pour les fonctions "**function.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour la définition des objets "**v5.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour l'opération **createPayment** :

```
<?php
include_once 'v5.php'; // Fichier comportant la définition des différentes objets
include_once 'function.php'; // Fichier comportant l'ensemble des fonctions utiles
(génération de l'uuid, etc...)

//Initialisation des variables
$shopId = "123456789";
$key = "123456789123456";
$mode = "TEST";
$wsdl = "https://epaync.nc/vads-ws/v5?wsdl";

//Exemple d'Initialisation d'un client SOAP avec gestion du SNI
/*
$client = new soapClient($wsdl, $options = array('trace'=>1,
    'exceptions'=> 0,
    'encoding' => 'UTF-8', 'soapaction' => '',
    'uri' => 'http://v5.ws.vads.lyra.com/',
    'cache_wsdl' => WSDL_CACHE_NONE,
    //Proxy parameters
    'proxy_host' => 'my.proxy.host',
    'proxy_port' => 3128,
    'stream_context' => stream_context_create (array('ssl' => array(
    'SNI_enabled' => true,
    'SNI_server_name' => 'epaync.nc'))))
    ));
*/

//Exemple d'Initialisation d'un client SOAP sans proxy
$client = new soapClient($wsdl, $options = array(
    'trace'=>1,
    'exceptions'=> 0,
    'encoding' => 'UTF-8',
    'soapaction' => ''
));

//Génération du header
$requestId = gen_uuid ();
$timestamp = gmdate ( "Y-m-d\TH:i:s\Z" );
$authToken = base64_encode(hash_hmac('sha256',$requestId.$timestamp, $key, true));
$headers ($shopId, $requestId, $timestamp, $mode, $authToken, $key, $client);

//Génération du body
$commonRequest = new commonRequest;
$commonRequest->paymentSource = 'EC';
$commonRequest->submissionDate = new DateTime('now',new DateTimeZone('UTC'));

$threeDSRequest = new threeDSRequest;
$threeDSRequest->mode = "ENABLED_CREATE";

$paymentRequest = new paymentRequest;
$paymentRequest->amount = "2990";
$paymentRequest->currency = "953";
$paymentRequest->manualValidation = '0';

$orderRequest = new orderRequest;
$orderRequest->orderId = "myOrder";

$cardRequest = new cardRequest;
$cardRequest->number = "4970100000000000";
$cardRequest->scheme = "VISA";
$cardRequest->expiryMonth = "12";
$cardRequest->expiryYear = "2023";
$cardRequest->cardSecurityCode = "123";
$cardRequest->cardHolderBirthDay = "2008-12-31";

$customerRequest = new customerRequest;
$customerRequest->billingDetails = new billingDetailsRequest;
$customerRequest->billingDetails->email="test@exemple.com";
```

```

$customerRequest->extraDetails = new extraDetailsRequest;

$techRequest = new techRequest;

//Appel de l'opération createPayment
try {
    $createPaymentRequest = new createPayment;
    $createPaymentRequest->commonRequest = $commonRequest;
    $createPaymentRequest->threeDSRequest = $threeDSRequest;
    $createPaymentRequest->paymentRequest = $paymentRequest;
    $createPaymentRequest->orderRequest = $orderRequest;
    $createPaymentRequest->cardRequest = $cardRequest;
    $createPaymentRequest->customerRequest = $customerRequest;
    $createPaymentRequest->techRequest = $techRequest;
    $createPaymentRequest->commonRequest->submissionDate = $createPaymentRequest->
commonRequest->submissionDate->format(dateTime::W3C);
    $createPaymentResponse= new createPaymentResponse();
    $createPaymentResponse = $client->createPayment($createPaymentRequest);
} catch (SoapFault $fault) {

//Gestion des exceptions
    trigger_error("SOAP Fault: (faultcode: {$fault->faultcode}, faultstring: {$fault->
faultstring})", E_USER_ERROR);
}

/* Affichage des logs XML à remplacer par une écriture dans un fichier de log.
*
* ATTENTION VOUS NE DEVEZ PAS ENREGISTRER LES NUMEROS DE CARTE DANS VOS LOGS
*/
echo "<hr> [Request Header] <br/>", htmlspecialchars($client->__getLastRequestHeaders()),
"<br/>";
echo "<hr> [Request] <br/>", htmlspecialchars($client->__getLastRequest()), "<br/>";
echo "<hr> [Response Header]<br/>", htmlspecialchars($client->__getLastResponseHeaders()),
"<br/>";
echo "<hr> [Response]<br/>", htmlspecialchars($client->__getLastResponse()), "<br/>";
echo '<hr>';
echo "<hr> [Response SOAP Headers]<br/>";

//Analyse de la réponse
//Récupération du SOAP Header de la réponse afin de stocker les en-têtes dans un tableau
(ici $responseHeader)
$dom = new DOMDocument;
$dom->loadXML($client->__getLastResponse(), LIBXML_NOWARNING);
$path = new DOMXPath($dom);
$headers = $path->query('//*[local-name()="Header"]/*');
$responseHeader = array();
foreach($headers as $headerItem) {
    $responseHeader[$headerItem->nodeName] = $headerItem->nodeValue;
}

//Calcul du jeton d'authentification de la réponse
$authTokenResponse = base64_encode(hash_hmac('sha256',$responseHeader['timestamp'].
$responseHeader['requestId'],$key,true));
if ($authTokenResponse !== $responseHeader['authToken']){
    //Erreur de calcul ou tentative de fraude
    echo 'Erreur interne rencontrée';
}
else{
    //Analyse de la réponse
    if ($createPaymentResponse->createPaymentResult->commonResponse->responseCode != "0"){
        //process error
    }
    else{
        //Process terminé avec succès
        //Test de la présence du transactionStatusLabel:
        if (isset ($createPaymentResponse->createPaymentResult->commonResponse->
transactionStatusLabel)){
            //La carte est non enrôlée ou 3DS Désactivé
            // Le paiement est accepté
            // Le code ci-dessous doit être modifié pour intégrer les mises à jour de base de
données etc..
            switch ($createPaymentResponse->createPaymentResult->commonResponse->
transactionStatusLabel) {
                case "AUTHORISED":
                    echo "paiement accepté";
                    break;
                case "WAITING AUTHORISATION":
                    echo "paiement accepté";
                    break;
                case "AUTHORISED_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                case "WAITING AUTHORISATION_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                // Le paiement est refusé
                default:
                    echo "paiement refusé";
                    break;
            }
        }
    }
}

```

```

    }
    else{
        // si absent = la transaction n'est pas créée, on est donc dans le cas d'une carte
        enrôlée
        // on procède alors à la génération du formulaire de redirection 3DS

        //On récupère l'identifiant de session afin de maintenir la session lors de l'analyse
        de la réponse de l'acs
        $cookie = getJsessionId($client);

        // On stocke l'identifiant de session dans le champ MD. Ce champ sera renvoyé inchangé
        par l'ACS
        $MD=setJsessionId($client)."+".$createPaymentResponse->createPaymentResult-
>threeDSResponse->authenticationRequestData->threeDSRequestId;

        //On initialise les autres champs nécessaire à la redirection vers l'ACS
        $threeDsAcUrl = $createPaymentResponse->createPaymentResult->threeDSResponse-
>authenticationRequestData->threeDSAcUrl;
        $threeDsEncodedPareq = $createPaymentResponse->createPaymentResult->threeDSResponse-
>authenticationRequestData->threeDSEncodedPareq;
        $threeDsServerResponseUrl = "http://127.0.0.1/webservices/ws-v5/retour3DS.php";

        //ATTENTION en mode TEST, l'identifiant de session doit être ajouté à l'URL de l'ACS
        pour maintenir la session HTTP
        $JSESSIONID=setJsessionId($client);
        if ($mode == "TEST"){
            $threeDsAcUrl = $threeDsAcUrl.";jsessionid=".$JSESSIONID;
        }
        formConstructor ($threeDsAcUrl,$MD,$threeDsEncodedPareq,$threeDsServerResponseUrl);
    }
}
?>

```

- un fichier pour traiter le retour de l'authentification 3D Secure **"retour3DS.php"** :

```

<?php
include_once 'v5.php'; // Fichier comportant la définition des différentes objets
include_once 'function.php';// Fichier comportant l'ensemble des fonctions utiles
(génération de l'uuid, etc...)

//Intialisation des variables
$shopId = "12345678";
$key = "123456789123456";
$mode = "TEST";
$wsdl = "https://epaync.nc/vads-ws/v5?wsdl";

//Récupération de la réponse de l'ACS
//On retrouve l'identifiant de session dans le champ MD afin de maintenir la session HTTP
if (isset ($ _POST['MD']) AND (isset ($ _POST['PaRes']))) {
    list($JSESSIONID, $threeDSRequestId) = explode(" +", $ _POST['MD']);

    //On supprime les espaces et les retours à la ligne du message PaRes
    $spares = str_replace("\r\n", "", $ _POST['PaRes'], $count);

    //Exemple d'Initialisation d'un client SOAP avec gestion du SNI
    /*
    $client = new soapClient($wsdl, $options = array('trace'=>1,
        'exceptions'=> 0,
        'encoding' => 'UTF-8', 'soapaction' => '',
        'uri' => 'http://v5.ws.vads.lyra.com/',
        'cache_wsdl' => WSDL_CACHE_NONE,
        //Proxy parameters
        'proxy_host' => 'my.proxy.host',
        'proxy_port' => 3128,
        'stream_context' => stream_context_create (array('ssl' => array(
            'SNI_enabled' => true,
            'SNI_server_name' => 'epaync.nc'))))
    );
    */

    //Exemple d'Initialisation d'un client SOAP sans proxy
    $client = new soapClient($wsdl, $options = array(
        'trace'=>1,
        'exceptions'=> 0,
        'encoding' => 'UTF-8',
        'soapaction' => ''
    ));

    //Génération du header
    $requestId = gen_uuid ();
    $timestamp = gmdate ( "Y-m-d\TH:i:sZ" );
    $authToken = base64_encode(hash_hmac('sha256',$requestId.$timestamp, $key, true));
    setHeaders ($shopId, $requestId, $timestamp, $mode, $authToken, $key, $client);

    //Génération du body
    $commonRequest = new commonRequest;

```

```

$commonRequest->submissionDate = new DateTime('now',new DateTimeZone('UTC'));

$threeDSRequest = new threeDSRequest;
$threeDSRequest->mode = "ENABLED_FINALIZE";
$threeDSRequest->requestId = $threeDSRequestId;
$threeDSRequest->pares = $pares;

$createPaymentRequest = new createPayment;
$createPaymentRequest->commonRequest = $commonRequest;
$createPaymentRequest->threeDSRequest = $threeDSRequest;

$createPaymentRequest->commonRequest->submissionDate = $createPaymentRequest->
commonRequest->submissionDate->format(dateTime::W3C);

try {
    //Maintenance de la session HTTP
    $client->__setCookie('JSESSIONID', $JSESSIONID);

    //Appel de l'opération createPayment
    $createPaymentResponse= new createPaymentResponse();
    $createPaymentResponse = $client->createPayment($createPaymentRequest);

} catch (SoapFault $fault) {
    //gestion des exceptions
    trigger_error("SOAP Fault: (faultcode: {$fault->faultcode}, faultstring: {$fault->
faultstring})", E_USER_ERROR);
}

/*
 * Affichage des logs XML à remplacer par une écriture dans un fichier de log.
 */
echo "<hr> [Request Header] <br/>", htmlspecialchars($client->__getLastRequestHeaders()),
"<br/>";
echo "<hr> [Request] <br/>", htmlspecialchars($client->__getLastRequest()), "<br/>";
echo "<hr> [Response Header]<br/>", htmlspecialchars($client->
__getLastResponseHeaders()), "<br/>";
echo "<hr> [Response]<br/>", htmlspecialchars($client->__getLastResponse()), "<br/>";
echo '<hr>';

//Analyse de la réponse
//Récupération du SOAP Header de la réponse afin de stocker les en-têtes dans un tableau
(ici $responseHeader)
$dom = new DOMDocument;
$dom->loadXML($client->__getLastResponse(), LIBXML_NOWARNING);
$path = new DOMXPath($dom);
$headers = $path->query('//*[local-name()="Header"]/*');
$responseHeader = array();
foreach($headers as $headerItem) {
    $responseHeader[$headerItem->nodeName] = $headerItem->nodeValue;
}

//Calcul du jeton d'authentification de la réponse.

$authTokenResponse = base64_encode(hash_hmac('sha256',$responseHeader['timestamp'].
$responseHeader['requestId'],$key, true));
if ($authTokenResponse !== $responseHeader['authToken']){

    //Erreur de calcul ou tentative de fraude
    echo 'Erreur interne rencontrée';
}
else{
    //Analyse de la réponse
    //Vérification du responseCode
    if ($createPaymentResponse->createPaymentResult->commonResponse->responseCode != "0"){

        //process error
        echo 'erreur interne';
    }
    else{
        //Process terminé avec succès
        //test de la présence du transactionStatusLabel:
        if (isset ($createPaymentResponse->createPaymentResult->commonResponse->
transactionStatusLabel)){

            // Le paiement est accepté
            // Le code ci-dessous doit être modifié pour intégrer les mises à jour de base de
données etc..
            switch ($createPaymentResponse->createPaymentResult->commonResponse->
transactionStatusLabel) {
                case "AUTHORISED":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTHORISATION":
                    echo "paiement accepté";
                    break;
                case "AUTHORISED_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTHORISATION_TO_VALIDATE":

```

```
        echo "paiement accepté";
        break;
    // Le paiement est refusé
    default:
        echo "paiement refusé";
        break;
    }
}
else{
    echo 'erreur interne';
}
}
}
else{
    //retour du 3DS sans paramètre ou accès direct à la page de retour 3DS
    echo 'error';
}
?>
```

8.2. findPayments

Recherche d'un paiement

Trois fichiers sont requis pour rechercher un paiement :

- un fichier pour les fonctions "**function.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour la définition des objets "**v5.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour l'opération **findPayments** :

```
<?php
include_once 'v5.php'; // Fichier comportant la définition des différentes objets
include_once 'function.php'; // Fichier comportant l'ensemble des fonctions utiles
(génération de l'uuid, etc...)

//Initialisation des variables
$shopId = "12345678";
$key = "1234567891234567";
$mode = "TEST";
$wsdl = "https://epaync.nc/vads-ws/v5?wsdl";

//Exemple d'Initialisation d'un client SOAP avec gestion du SNI
/*
$client = new soapClient($wsdl,
$options = array('trace'=>1, 'exceptions'=> 0,
'encoding' => 'UTF-8','soapaction' => '',
'uri' => 'http://v5.ws.vads.lyra.com/',
'cache_wsdl' => WSDL_CACHE_NONE,
//Proxy parameters
'proxy_host' => 'my.proxy.host',
'proxy_port' => 3128,
'stream_context' => stream_context_create (array('ssl' => array(
'SNI_enabled' => true,
'SNI_server_name' => 'epaync.nc'))
));
*/

//Exemple d'Initialisation d'un client SOAP sans proxy
$client = new soapClient($wsdl, $options = array(
'trace'=>1,
'exceptions'=> 0,
'encoding' => 'UTF-8',
'soapaction' => ''
));

//Génération du header
$requestId = gen_uuid ();
$timestamp = gmdate ( "Y-m-d\TH:i:s\Z" );
$authToken = base64_encode(hash_hmac('sha256',$requestId.$timestamp, $key, true));
setHeaders ($shopId, $requestId, $timestamp, $mode, $authToken, $key, $client);

//Génération du body
$queryRequest = new queryRequest;
$queryRequest->orderId ="myOrder";

try {

    $findPaymentsRequest = new findPayments;

    $findPaymentsRequest->queryRequest = $queryRequest;

    $findPaymentsResponse = $client->findPayments($findPaymentsRequest);
} catch (SoapFault $fault) {

//Gestion des exceptions
trigger_error("SOAP Fault: (faultcode: {$fault->faultcode}, faultstring: {$fault->faultstring})", E_USER_ERROR);
}

/* Affichage des logs XML à remplacer par une écriture dans un fichier de log.
*
* ATTENTION VOUS NE DEVEZ PAS ENREGISTRER LES NUMEROS DE CARTE DANS VOS LOGS
*/
echo "<hr> [Request Header] <br/>", htmlspecialchars($client->__getLastRequestHeaders()),
"<br/>";
echo "<hr> [Request] <br/>", htmlspecialchars($client->__getLastRequest()), "<br/>";
```

```

        echo "<hr> [Response Header]<br/>", htmlspecialchars($client->__getLastResponseHeaders()), "<br/>";
        echo "<hr> [Response]<br/>", htmlspecialchars($client->__getLastResponse()), "<br/>";
        echo '<hr>';
        echo "<hr> [Response SOAP Headers]<br/>";

//Analyse de la réponse
//Récupération du SOAP Header de la réponse afin de stocker les en-têtes dans un tableau
(ici $responseHeader)
$dom = new DOMDocument;
    $dom->loadXML($client->__getLastResponse(), LIBXML_NOWARNING);
    $path = new DOMXPath($dom);
    $headers = $path->query('//*[local-name()="Header"]/*');
$responseHeader = array();
foreach($headers as $headerItem) {
    $responseHeader[$headerItem->nodeName] = $headerItem->nodeValue;
}

//Calcul du jeton d'authentification de la réponse
$authTokenResponse = base64_encode(hash_hmac('sha256',$responseHeader['timestamp'].
$responseHeader['requestId'],$key, true));
if ($authTokenResponse !== $responseHeader['authToken']){
    //Erreur de calcul ou tentative de fraude
    echo 'Erreur interne rencontrée';
}
else{

//Analyse de la réponse
if ($findPaymentsResponse->findPaymentsResult->commonResponse->responseCode != "0"){
    //process error
}
else{
    //Process terminé avec succès
    //test de la présence du transactionStatusLabel:
    if (isset ($findPaymentsResponse->findPaymentsResult->commonResponse->transactionStatusLabel)){
        //la carte est non enrôlée ou 3DS Désactivé

        // Le paiement est accepté
        switch ($findPaymentsResponse->findPaymentsResult->commonResponse->transactionStatusLabel) {

            case "AUTHORISED":
                echo "paiement accepté";
                break;
            case "WAITING AUTORISATION":
                echo "paiement accepté";
                break;
            case "AUTHORISED_TO_VALIDATE":
                echo "paiement accepté";
                break;
            case "WAITING_AUTORISATION_TO_VALIDATE":
                echo "paiement accepté";
                break;
            // Le paiement est refusé
            default:
                echo "paiement refusé";
                break;

        }

    }
    else{
        // si absent = la transaction n'est pas créée, on est donc dans le cas d'une carte enrôlée
        // on procède alors à la génération du formulaire de redirection 3DS
        //On récupère l'identifiant de session afin de maintenir la session lors de l'analyse de la réponse de l'acs
        $cookie = getSessionId($client);
    }
}
}
}

```

8.3. createToken

Création d'un alias

Trois fichiers sont requis pour créer un alias :

- un fichier pour les fonctions "**function.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour la définition des objets "**v5.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour l'opération **createToken** :

```
<?php
include_once 'v5.php'; // Fichier comportant la définition des différentes objets
include_once 'function.php'; // Fichier comportant l'ensemble des fonctions utiles
(génération de l'uuid, etc...)

//Initialisation des variables
$shopId = "12345678";
$key = "1234567891234567";
$mode = "TEST";
$wsdl = "https://epaync.nc/vads-ws/v5?wsdl";

//Exemple d'Initialisation d'un client SOAP avec gestion du SNI
/*
$client = new soapClient($wsdl,
$options = array('trace'=>1, 'exceptions'=> 0,
    'encoding' => 'UTF-8', 'soapaction' => '',
    'uri' => 'http://v5.ws.vads.lyra.com/',
    'cache_wsdl' => WSDL_CACHE_NONE,
    //Proxy parameters
    'proxy_host' => 'my.proxy.host',
    'proxy_port' => 3128,
    'stream_context' => stream_context_create (array('ssl' => array(
        'SNI_enabled' => true,
        'SNI_server_name' => 'epaync.nc'))))
));
*/

//Exemple d'Initialisation d'un client SOAP sans proxy
$client = new soapClient($wsdl, $options = array(
    'trace'=>1,
    'exceptions'=> 0,
    'encoding' => 'UTF-8',
    'soapaction' => ''
));

//Génération du header
$requestId = gen_uuid ();
$timestamp = gmdate ( "Y-m-d\TH:i:s\Z" );
$authToken = base64_encode(hash_hmac('sha256',$requestId.$timestamp, $key, true));
$headers ($shopId, $requestId, $timestamp, $mode, $authToken, $key, $client);

//Génération du body
$commonRequest = new commonRequest;
$commonRequest->submissionDate = new DateTime('now',new DateTimeZone('UTC'));

$cardRequest = new cardRequest;
$cardRequest->number = "4970100000000003";
$cardRequest->scheme = "VISA";
$cardRequest->expiryMonth = "12";
$cardRequest->expiryYear = "2023";
$cardRequest->cardSecurityCode = "123";

$customerRequest = new customerRequest;
$customerRequest->billingDetails = new billingDetailsRequest;
$customerRequest->billingDetails->email="nom.prenom@exemple.com";

$customerRequest->extraDetails = new extraDetailsRequest;
$customerRequest->extraDetails->sendMail ="1";
$customerRequest->extraDetails->ipAddress ="127.0.0.1";

//Appel de l'opération createToken
try {

    $createTokenRequest = new createToken;
    $createTokenRequest->commonRequest = $commonRequest;
    $createTokenRequest->cardRequest = $cardRequest;
    $createTokenRequest->customerRequest = $customerRequest;

    $createTokenRequest->commonRequest->submissionDate = $createTokenRequest->commonRequest->
    >submissionDate->format(dateTime::W3C);
```

```

    $createTokenResponse = $client->createToken($createTokenRequest);
} catch (SoapFault $fault) {

//Gestion des exceptions

    trigger_error("SOAP Fault: (faultcode: {$fault->faultcode}, faultstring: {$fault->faultstring})", E_USER_ERROR);
}

/* Affichage des logs XML à remplacer par une écriture dans un fichier de log.
 *
 * ATTENTION VOUS NE DEVEZ PAS ENREGISTRER LES NUMEROS DE CARTE DANS VOS LOGS
 */
echo "<hr> [Request Header] <br/>", htmlspecialchars($client->__getLastRequestHeaders()),
"<br/>";
echo "<hr> [Request] <br/>", htmlspecialchars($client->__getLastRequest()), "<br/>";
echo "<hr> [Response Header]<br/>", htmlspecialchars($client->__getLastResponseHeaders()), "<br/>";
echo "<hr> [Response]<br/>", htmlspecialchars($client->__getLastResponse()), "<br/>";
echo '<hr>';
echo "<hr> [Response SOAP Headers]<br/>";

//Analyse de la réponse
//Récupération du SOAP Header de la réponse afin de stocker les en-têtes dans un tableau
(ici $responseHeader)
$dom = new DOMDocument;
    $dom->loadXML($client->__getLastResponse(), LIBXML_NOWARNING);
    $path = new DOMXPath($dom);
    $headers = $path->query('//*[local-name()="Header"]/*');
$responseHeader = array();
foreach($headers as $headerItem) {
    $responseHeader[$headerItem->nodeName] = $headerItem->nodeValue;
}

//Calcul du jeton d'authentification de la réponse
$authTokenResponse = base64_encode(hash_hmac('sha256',$responseHeader['timestamp'].
$responseHeader['requestId'],$key, true));
if ($authTokenResponse !== $responseHeader['authToken']){
    //Erreur de calcul ou tentative de fraude
    echo 'Erreur interne rencontrée';
}
else{

    //Analyse de la réponse
    if ($createTokenResponse->createTokenResult->commonResponse->responseCode != "0"){

        //process error
    }
    else{
        //Process terminé avec succès
        //Test de la présence du transactionStatusLabel:
        if (isset ($createTokenResponse->createTokenResult->commonResponse->transactionStatusLabel)){
            //La carte est non enrôlée ou 3DS Désactivé

            // Le paiement est accepté
            // Le code ci-dessous doit être modifié pour intégrer les mises à jour de base de
            données etc..

            switch ($createTokenResponse->createTokenResult->commonResponse->transactionStatusLabel) {
                case "AUTHORISED":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTORISATION":
                    echo "paiement accepté";
                    break;
                case "AUTHORISED_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTORISATION_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                // Le paiement est refusé
                default:
                    echo "paiement refusé";
                    break;
            }
        }
    }
}
}
?>

```

8.4. createSubscription

L'opération **createSubscription** permet de réaliser des paiements récurrents (abonnements). Elle nécessite l'utilisation d'un alias déjà existant et valide.

Cet alias peut être créé via :

- l'opération **createToken**,
ou
- par formulaire (voir **Guide d'implémentation du formulaire de paiement**)

Trois fichiers sont requis pour créer un alias :

- un fichier pour les fonctions "**function.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour la définition des objets "**v5.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour l'opération **createSubscription** :

```
<?php
include_once 'v5.php'; // Fichier comportant la définition des différentes objets
include_once 'function.php'; // Fichier comportant l'ensemble des fonctions utiles
(génération de l'uuid, etc...)

//Initialisation des variables
$shopId = "12345678";
$key = "1234567891234567";
$mode = "TEST";
$wsdl = "https://epaync.nc/vads-ws/v5?wsdl";

//Exemple d'Initialisation d'un client SOAP avec gestion du SNI
/*
$client = new soapClient($wsdl,
$options = array('trace'=>1, 'exceptions'=> 0,
'encoding' => 'UTF-8', 'soapaction' => '',
'uri' => 'http://v5.ws.vads.lyra.com/',
'cache_wsdl' => WSDL_CACHE_NONE,
//Proxy parameters
'proxy_host' => 'my.proxy.host',
'proxy_port' => 3128,
'stream_context' => stream_context_create (array('ssl' => array(
'SNI_enabled' => true,
'SNI_server_name' => 'epaync.nc'))))
);
*/

//Exemple d'Initialisation d'un client SOAP sans proxy
$client = new soapClient($wsdl, $options = array(
'trace'=>1,
'exceptions'=> 0,
'encoding' => 'UTF-8',
'soapaction' => ''
));

//Génération du header
$requestId = gen_uuid ();
$timestamp = gmdate ( "Y-m-d\TH:i:s\Z" );
$authToken = base64_encode(hash_hmac('sha256',$requestId.$timestamp, $key, true));
$headers ($shopId, $requestId, $timestamp, $mode, $authToken, $key, $client);

//Génération du body

$commonRequest = new commonRequest;
$commonRequest->submissionDate = new DateTime('now',new DateTimeZone('UTC'));

$orderRequest = new orderRequest;
$orderRequest->orderId = "myFirstSubscription";

$subscriptionRequest = new subscriptionRequest;
$subscriptionRequest->effectDate = "2016-07-10T18:00:00Z";
$subscriptionRequest->amount = "30000";
$subscriptionRequest->currency = "953";
$subscriptionRequest->initialAmount = "1000";
$subscriptionRequest->initialAmountNumber = "1";
$subscriptionRequest->rrule = "RRULE:FREQ=MONTHLY;COUNT=12;BYMONTHDAY=10";
```

```

$cardRequest = new cardRequest;
$cardRequest->paymentToken = "MyToken";

//Appel de l'opération createSubscription
try {
    $createSubscriptionRequest = new createSubscription;
    $createSubscriptionRequest->commonRequest = $commonRequest;
    $createSubscriptionRequest->orderRequest = $orderRequest;
    $createSubscriptionRequest->cardRequest = $cardRequest;
    $createSubscriptionRequest->subscriptionRequest = $subscriptionRequest;

    $createSubscriptionRequest->commonRequest->submissionDate = $createSubscriptionRequest->commonRequest->submissionDate->format(dateTime::W3C);

    $createSubscriptionResponse = $client->createSubscription($createSubscriptionRequest);
} catch (SoapFault $fault) {

//Gestion des exceptions

    trigger_error("SOAP Fault: (faultcode: {$fault->faultcode}, faultstring: {$fault->faultstring})", E_USER_ERROR);
}

/* Affichage des logs XML à remplacer par une écriture dans un fichier de log.
 *
 * ATTENTION VOUS NE DEVEZ PAS ENREGISTRER LES NUMEROS DE CARTE DANS VOS LOGS
 */
echo "<hr> [Request Header] <br/>", htmlspecialchars($client->__getLastRequestHeaders()),
"<br/>";
echo "<hr> [Request] <br/>", htmlspecialchars($client->__getLastRequest()), "<br/>";
echo "<hr> [Response Header]<br/>", htmlspecialchars($client->__getLastResponseHeaders()), "<br/>";
echo "<hr> [Response]<br/>", htmlspecialchars($client->__getLastResponse()), "<br/>";
echo '<hr>';
echo "<hr> [Response SOAP Headers]<br/>";

///Analyse de la réponse
//Récupération du SOAP Header de la réponse afin de stocker les en-têtes dans un tableau
(ici $responseHeader)
    $dom = new DOMDocument;
    $dom->loadXML($client->__getLastResponse(), LIBXML_NOWARNING);
    $xpath = new DOMXPath($dom);
    $headers = $xpath->query('//*[local-name()="Header"]/*');
    $responseHeader = array();
    foreach($headers as $headerItem) {
        $responseHeader[$headerItem->nodeName] = $headerItem->nodeValue;
    }

//Calcul du jeton d'authentification de la réponse
$authTokenResponse = base64_encode(hash_hmac('sha256',$responseHeader['timestamp'].
$responseHeader['requestId'],-$key, true));
if ($authTokenResponse !== $responseHeader['authToken']){
    //Erreur de calcul ou tentative de fraude
    echo 'Erreur interne rencontrée';
}
else{

    //Analyse de la réponse
    if ($createSubscriptionResponse->createSubscriptionResult->commonResponse->responseCode != "0"){
        //process error
    }
    else{

        //Process terminé avec succès
        //Test de la présence du transactionStatusLabel:
        if (isset ($createSubscriptionResponse->createSubscriptionResult->commonResponse->transactionStatusLabel)){
            //La carte est non enrôlée ou 3DS Désactivé

            // Le paiement est accepté
            // Le code ci-dessous doit être modifié pour intégrer les mises à jour de base de données etc..

            switch ($createPaymentResponse->createPaymentResult->commonResponse->transactionStatusLabel){
                case "AUTHORISED":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTHORISATION":
                    echo "paiement accepté";
                    break;
                case "AUTHORISED_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTHORISATION_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
            }
            // Le paiement est refusé

```

```
        default:
            echo "paiement refusé";
            break;
        }
    }
}
?>
```

8.5. cancelSubscription

Trois fichiers sont requis pour résilier un abonnement :

- un fichier pour les fonctions "**function.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour la définition des objets "**v5.php**" (contenu disponible au chapitre **Annexe**).
- un fichier pour l'opération **cancelSubscription** :

```
<?php
include_once 'v5.php'; // Fichier comportant la définition des différentes objets
include_once 'function.php'; // Fichier comportant l'ensemble des fonctions utiles
(génération de l'uuid, etc...)

//Initialisation des variables
$shopId = "12345678";
$key = "1234567891234567";
$mode = "TEST";
$wsdl = "https://epaync.nc/vads-ws/v5?wsdl";

//Exemple d'Initialisation d'un client SOAP sans proxy
$client = new soapClient($wsdl, $options = array(
    'trace'=>1,
    'exceptions'=> 0,
    'encoding' => 'UTF-8',
    'soapaction' => ''
));

//Génération du header
$requestId = gen_uuid ();
$timestamp = gmdate ( "Y-m-d\TH:i:s\Z" );
$authToken = base64_encode(hash_hmac('sha256',$requestId.$timestamp, $key, true));
$headers ($shopId, $requestId, $timestamp, $mode, $authToken, $key, $client);

//Génération du body

$commonRequest = new commonRequest;

$queryRequest = new queryRequest;
$queryRequest->paymentToken = "c975d6af1d5e478da3570d43494d86d2";
$queryRequest->submissionDate = "2015-08-28T09:21:34+00:00";
$queryRequest->subscriptionId = "20150828i6VFSA";

// Appel de l'opération ccancelSubscription
try {
    $cancelSubscriptionRequest = new cancelSubscription;
    $cancelSubscriptionRequest->commonRequest = $commonRequest;
    $cancelSubscriptionRequest->queryRequest = $queryRequest;

    $cancelSubscriptionResponse = $client->cancelSubscription($cancelSubscriptionRequest);
} catch (SoapFault $fault) {

//Gestion des exceptions

    trigger_error("SOAP Fault: (faultcode: {$fault->faultcode}, faultstring: {$fault->faultstring})", E_USER_ERROR);
}

/* Affichage des logs XML à remplacer par une écriture dans un fichier de log.
 *
 * ATTENTION VOUS NE DEVEZ PAS ENREGISTRER LES NUMEROS DE CARTE DANS VOS LOGS
 */
echo "<hr> [Request Header] <br/>", htmlspecialchars($client->__getLastRequestHeaders()),
"<br/>";
echo "<hr> [Request] <br/>", htmlspecialchars($client->__getLastRequest()), "<br/>";
echo "<hr> [Response Header]<br/>", htmlspecialchars($client->
__getLastResponseHeaders()), "<br/>";
echo "<hr> [Response]<br/>", htmlspecialchars($client->__getLastResponse()), "<br/>";
echo '<hr>';
echo "<hr> [Response SOAP Headers]<br/>";

///Analyse de la réponse
//Récupération du SOAP Header de la réponse afin de stocker les en-têtes dans un tableau
(ici $responseHeader)
$dom = new DOMDocument;
$dom->loadXML($client->__getLastResponse(), LIBXML_NOWARNING);
$path = new DOMXPath($dom);
$headers = $path->query('//*[local-name()="Header"]/*');
$responseHeader = array();
foreach($headers as $headerItem) {
    $responseHeader[$headerItem->nodeName] = $headerItem->nodeValue;
}
```

```

//Calcul du jeton d'authentification de la réponse
$authTokenResponse = base64_encode(hash_hmac('sha256',$responseHeader['timestamp'].
$responseHeader['requestId'],$key, true));
if ($authTokenResponse != $responseHeader['authToken']){
    //Erreur de calcul ou tentative de fraude
    echo 'Erreur interne rencontrée';
}
else{
    //Analyse de la réponse
    if ($cancelSubscriptionResponse->cancelSubscriptionResult->commonResponse->responseCode != "0"){
        //process error
    }
    else{
        //Process terminé avec succès
        //Test de la présence du transactionStatusLabel:
        if (isset ($cancelSubscriptionResponse->cancelSubscriptionResult->commonResponse->transactionStatusLabel)){
            //La carte est non enrôlée ou 3DS Désactivé

            // Le paiement est accepté
            // Le code ci-dessous doit être modifié pour intégrer les mises à jour de base de données etc..

            switch ($cancelSubscriptionResponse->cancelSubscriptionResult->commonResponse->transactionStatusLabel){
                case "AUTHORISED":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTHORISATION":
                    echo "paiement accepté";
                    break;
                case "AUTHORISED_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                case "WAITING_AUTHORISATION_TO_VALIDATE":
                    echo "paiement accepté";
                    break;
                // Le paiement est refusé
                default:
                    echo "paiement refusé";
                    break;
            }
        }
    }
}
}
?>

```


9.1. Codes réponse d'une demande d'autorisation (authorizationResponse.result)

Le tableau ci-dessous liste les différentes valeurs associées aux moyens de paiement :

- **CB**
- **Visa**
- **MasterCard**

Valeur	Description	Motif frauduleux	Valeur	Description	Motif frauduleux
0	Transaction approuvée ou traitée avec succès		38	Date de validité de la carte dépassée	
2	Contacter l'émetteur de carte		41	Carte perdue	OUI
3	Accepteur invalide	OUI	43	Carte volée	OUI
4	Conserver la carte	OUI	51	Provision insuffisante ou crédit dépassé	
5	Ne pas honorer	OUI	54	Date de validité de la carte dépassée	OUI
7	Conserver la carte, conditions spéciales	OUI	55	Code confidentiel erroné	
8	Approuver après identification		56	Carte absente du fichier	OUI
12	Transaction invalide	OUI	57	Transaction non permise à ce porteur	OUI
13	Montant invalide	OUI	58	Transaction non permise à ce porteur	
14	Numéro de porteur invalide	OUI	59	Suspicion de fraude	OUI
15	Emetteur de carte inconnu	OUI	60	L'accepteur de carte doit contacter l'acquéreur	
17	Annulation acheteur		61	Montant de retrait hors limite	
19	Répéter la transaction ultérieurement		63	Règles de sécurité non respectées	OUI
20	Réponse erronée (erreur dans le domaine serveur)		68	Réponse non parvenue ou reçue trop tard	
24	Mise à jour de fichier non supportée		75	Nombre d'essais code confidentiel dépassé	
25	Impossible de localiser l'enregistrement dans le fichier		76	Porteur déjà en opposition, ancien enregistrement conservé	OUI
26	Enregistrement dupliqué, ancien enregistrement remplacé		90	Arrêt momentané du système	
27	Erreur en « edit » sur champ de liste à jour fichier		91	Émetteur de cartes inaccessible	
28	Accès interdit au fichier		94	Transaction dupliquée	
29	Mise à jour impossible		96	Mauvais fonctionnement du système	
30	Erreur de format		97	Échéance de la temporisation de surveillance globale	
31	Identifiant de l'organisme acquéreur inconnu	OUI	98	Serveur indisponible routage réseau demandé à nouveau	
33	Date de validité de la carte dépassée	OUI	99	Incident domaine initiateur	
34	Suspicion de fraude	OUI			

9.2. Codes réponse de l'analyseur de risque externe (`fraudManagementResponse.riskAnalysis.resultCode`)

Valeurs communes à tous les analyseurs de risques	
INVALID_CREDENCIAL	Problème de paramétrage du contrat d'analyse de risques.
COMUNICACION_PROBLEM	Impossible de communiquer avec l'analyseur de risques.
DATA_PROCESSING_PROBLEM	Problème lors du traitement de l'envoi ou de la réponse d'analyse de risques.
MISSING_MANDATORY_ORDER_INFO	Des données relatives à la commande sont manquantes.
MISSING_MANDATORY_SHIPPING_INFO	Des données relatives à la livraison sont manquantes.
MISSING_MANDATORY_SHIPPING_ADDRES	Des données relatives à l'adresse de livraison sont manquantes.
MISSING_MANDATORY_BILLING_INFO	Des données relatives à la facturation sont manquantes.
MISSING_MANDATORY_BILLING_ADDRESS	Des données relatives à l'adresse de facturation sont manquantes.
MISSING_MANDATORY_CARD_INFO	Des données concernant le moyen de paiement sont manquantes.
MISSING_MANDATORY_CUSTOMER_INFO	Des données concernant l'acheteur sont manquantes.

Tableau 210 : Valeurs communes à tous les types d'analyseurs de risques

ClearSale	
APA	La transaction est automatiquement approuvée selon les paramètres définis.
APM	La transaction est manuellement approuvée par un analyste.
RPM	La commande est refusée en raison du manque d'informations sur l'acheteur en accord avec la politique appliquée.
AMA	En attente d'analyse manuelle. La commande est dans une file d'attente pour analyse.
ERR	Erreur
NVO	Nouvelle commande. En attente de traitement et de classification.
SUS	Commande suspendue manuellement. La commande est suspendue pour suspicion de fraude.
CAN	Commande annulée. La commande est annulée par l'acheteur.
FRD	Fraude confirmée avec l'opérateur de la carte de crédit ou du titulaire de la carte.
RPA	Commande refusée automatiquement. La commande est refusée en application des paramètres de l'analyseur de fraude externe.
RPP	Commande refusée automatiquement. La commande est refusée en application de la politique client ou ClearSale.

Tableau 211 : Valeurs retournées par ClearSale

CyberSource	
100	La transaction s'est effectuée avec succès.
101	La transaction est refusée. Un ou plusieurs champs sont manquants.
102	La transaction est refusée. Un ou plusieurs champs contient des données invalides.
150	Erreur.
151	Erreur. La requête a été reçue mais le délai a été dépassé. Cette erreur n'inclue pas les dépassements de délais entre le client et le serveur.
152	Erreur. La requête a été reçue mais un service n'a pas terminé à temps.
202	Refusée. Carte expirée.
231	Refusée. Numéro de compte invalide.
234	Refusé. Un problème est survenu avec la configuration CyberSource du marchand.
400	Refusée. Le score de la fraude dépasse le seuil de tolérance.
480	La commande est marquée afin d'être examinée par le Decision Manager.
481	La commande a été rejetée par le Decision Manager.

Tableau 212 : Valeurs retournées par Cybersource

Konduto	
APPROVE	Konduto recommande d'accepter la transaction. Si aucune règle ne contredit cette recommandation, le statut de la transaction sera AUTHORISED.
DECLINE	Konduto recommande de refuser la transaction.

Konduto	
	Le statut de la transaction sera REFUSED.
REVIEW	<p>Konduto recommande de vérifier la transaction. En fonction du résultat de l'authentification 3D-Secure, le statut de la transaction sera :</p> <ul style="list-style-type: none"> • AUTHORISED_TO_VALIDATE si le porteur s'est authentifié avec succès. • REFUSED en cas d'échec d'authentification du porteur.

Tableau 213 : Valeurs retournées par Konduto

9.3. Codes d'erreurs pouvant survenir durant le paiement (paymentResponse.paymentError)

Dans la réponse d'une opération web service, l'objet **paymentError** contient une valeur qui détermine la raison d'un paiement refusé.

Le tableau ci-dessous liste les différentes valeurs :

Code	Message
1	La transaction n'a pas été trouvée.
2	La transaction n'a pas été trouvée.
3	Cette action n'est pas autorisée sur une transaction ayant ce statut {0}.
4	Cette transaction n'est pas autorisée dans ce contexte.
5	La transaction existe déjà.
6	Montant de transaction invalide.
7	Cette action n'est plus possible pour une transaction créée à cette date.
8	La date d'expiration de la carte ne permet pas cette action.
9	CVV obligatoire pour la carte.
10	Le montant de remboursement est supérieur au montant initial.
11	La somme des remboursements effectués est supérieure au montant initial.
12	La duplication d'un crédit (remboursement) n'est pas autorisée.
13	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
14	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
15	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
16	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
17	Le téléparamétrage du contrat Aurore a échoué.
18	L'analyse de la réponse Cetelem a échoué.
19	Devise inconnue.
20	Type de carte invalide.
21	Aucun contrat trouvé pour ce paiement. Veuillez modifier les données ou contacter votre gestionnaire en cas d'échecs répétés.
22	Boutique non trouvée.
23	Contrat ambiguë
24	Contrat invalide
25	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
26	Numéro de carte invalide
27	Numéro de carte invalide
28	Numéro de carte invalide
29	Numéro de carte invalide
30	Numéro de carte invalide (Luhn)
31	Numéro de carte invalide (longueur)
32	Numéro de carte invalide (non trouvé)
33	Numéro de carte invalide (non trouvé)
34	Contrôle carte à autorisation systématique en échec.
35	Contrôle e-Carte Bleue en échec.
36	Le contrôle des risques a provoqué le refus de la transaction.
37	Interruption non gérée lors du processus de paiement.
38	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
39	Refus 3D Secure pour la transaction.
40	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
41	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
42	Une erreur interne est survenue lors de la consultation du numéro de carte.
43	Une erreur interne est survenue lors de la consultation du numéro de carte.
45	Devise invalide pour la modification.
46	Le montant est supérieur au montant autorisé.

Code	Message
47	La date de présentation souhaitée est postérieure à la date de validité de l'autorisation.
48	La modification requise est invalide.
49	Définition du paiement multiple invalide.
50	Boutique inconnue.
51	Cours inconnu.
52	Le contrat est clos depuis le {0}.
53	La boutique {0} est close depuis le {1}.
54	Paramètre rejeté pouvant contenir des données sensibles {0}.
55	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
57	Erreur lors de la récupération de l'alias.
58	Le statut de l'alias n'est pas compatible avec cette opération
59	Erreur lors de la récupération de l'alias.
60	Alias existant.
61	Alias invalide
62	Création d'un alias refusée.
63	Abonnement déjà existant.
64	Cet abonnement est déjà résilié.
65	Cet abonnement est invalide.
66	La règle de récurrence n'est pas valide.
67	Création de l'abonnement refusée.
69	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
70	Code pays invalide.
71	Paramètre du service web invalide.
72	Refus d'autorisation par Cofinoga.
73	Refus de l'autorisation à 100 XPF.
74	Configuration de paiement invalide.
75	L'opération a été refusée par PayPal.
77	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
78	Identifiant de transaction non défini.
79	Identifiant de transaction déjà utilisé.
80	Identifiant de transaction expiré.
81	Contenu du thème config invalide.
82	Le remboursement n'est pas autorisé.
83	Montant de transaction en dehors des valeurs permises.
85	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
87	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
91	Aucune option de paiement activée pour ce contrat.
92	Une erreur est survenue lors du calcul du canal de paiement.
93	Une erreur est survenue lors du retour de l'acheteur sur la page de finalisation de paiement.
94	Une erreur technique est survenue.
96	Une erreur est apparue lors de la remise de cette transaction.
97	Date de remise trop éloignée.
98	Date de transaction invalide.
99	Une erreur est survenue lors du calcul de l'origine du paiement.
100	Contrôle carte commerciale en échec.
101	Refusé car première échéance refusée.
103	Le statut de la transaction n'a pas pu être synchronisé avec le système externe
104	Une erreur est apparue lors de la remise de cette transaction.
105	Une erreur de sécurité est apparue lors du processus 3DS de cette transaction.
106	Devise non supportée pour ce contrat et/ou cette boutique.
107	La carte associée à l'alias n'est plus valide.
108	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
109	Délai d'attente dépassé lors de la redirection de l'acheteur.
110	Carte de paiement non supportée par le contrat.

Code	Message
111	Refus des transactions sans transfert de responsabilité.
112	L'annulation n'est pas autorisée.
113	La duplication n'est pas autorisée.
115	Le remboursement n'est pas autorisé.
116	Paieement manuel non autorisé pour cette carte.
118	Paieement manuel en plusieurs fois non autorisé pour cette carte.
119	La date soumise est invalide.
120	L'option de paiement de la transaction initiale n'est pas applicable.
124	Carte inactive.
125	Paieement refusé par l'acquéreur.
126	Cette action n'est pas possible car la séquence de paiement n'est pas terminée.
136	Refus des transactions dérivées, sans transfert de responsabilité sur la transaction primaire.
137	La transaction est un doublon.
138	Le remboursement partiel n'est pas possible sur cette transaction.
139	Remboursement refusé.
141	L'analyseur de risque a rejeté cette transaction.
142	Le type de carte utilisé n'est pas valide pour le mode de paiement demandé.
143	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
144	Une transaction en mode production a été marquée en mode test chez l'acquéreur.
145	Une transaction en mode test a été marquée en mode production chez l'acquéreur.
146	Code sms invalide.
147	Le module de gestion de fraudes a demandé le refus de cette transaction.
148	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande. La transaction n'a pas été créée.
149	La durée de la session de paiement a expiré (cas de l'acheteur qui est redirigé vers l'ACS et qui ne finalise pas l'authentification 3D Secure).
150	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande. La transaction n'a pas été créée.
151	Une transaction Facily Pay ne peut pas être annulée/modifiée/remboursée entre 23h30 et 5h30
152	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande
153	Une erreur technique est survenue lors de l'appel au service Banque Accord
155	La transaction Facily Pay n'a pu être annulée/modifiée/remboursée : l'état de la transaction ne permet pas de réaliser l'action demandée. Rappel concernant une transaction Facily Pay : un remboursement doit respecter un délai de deux jours après la remise, le délai entre deux remboursements est d'un jour, un remboursement partiel est limité à 20 jours, un remboursement total est limité à 6 mois.
158	Suite à un incident technique, nous ne sommes pas en mesure de traiter votre demande.
159	Le montant est inférieur au montant minimum autorisé (minimum={0} {1}).
160	Il est impossible de rembourser une transaction impayée.
165	Le type de document d'identité est présent, mais son numéro est absent.
166	Le numéro de document d'identité est présent, mais son type est absent.
167	Le type du document d'identité est inconnu.
168	Le numéro du document d'identité est invalide.
169	Les données spécifiques devant être transmises à l'acquéreur sont invalides.
170	Le paiement différé n'est pas autorisé.
171	Le nombre de mois pour le paiement différé n'est pas autorisé.
172	La cinématique de paiement sélectionnée est invalide.
175	Annulation impossible, veuillez tenter un remboursement
176	Remboursement impossible, veuillez tenter une annulation.
177	Aucune réponse à la demande d'autorisation n'a été reçue dans le délai imparti,
178	Annulation impossible, la transaction a déjà été annulée
179	Le status de la transaction est inconnue.
182	L'identifiant national du client est absent.
183	Le format de l'identifiant national du client est incorrect.

9.4. Codes produits VISA / MASTERCARD (cardResponse.productCode)

VISA	Désignation
A	Visa Traditional
B	Visa Traditional Rewards
C	Visa Signature
D	Visa Signature Preferred
E	Proprietary ATM
F	Visa Classic
G	Visa Business
G1	Visa Signature Business
G2	Reserved
G3	Visa Business Enhanced
H	Reserved
I	Visa Infinite
J	Reserved
J1	Reserved
J2	Reserved
J3	Visa Healthcare
J4	Reserved
K	Visa Corporate T&E
K1	Visa GSA Corporate T&E
L	Electron
N	Visa Platinum
N1	TBA
P	Visa Gold
Q	Private Label
Q1	Reserved
R	Proprietary
S	Visa Purchasing
S1	Visa Purchasing
S2	Visa Purchasing
S3	Visa Purchasing
S4	Government Services Loan
S5	Commercial Transport EBT
S6	Business Loan
S7	Visa Distribution
T	Reserved
U	Visa TravelMoney
V	Visa VPay
W	Reserved
X	Reserved
Y	Reserved
Z	Reserved

MASTERCARD	Désignation
MPN	MASTERCARD PREPAID DEBIT STANDARD-INSURANCE
MPO	MASTERCARD PREPAID DEBIT STANDARD-OTHER
MPP	MASTERCARD PREPAID CARD
MPR	MASTERCARD PREPAID DEBIT STANDARD-TRAVEL
MPT	MASTERCARD PREPAID DEBIT STANDARD-TEEN
MPV	MASTERCARD PREPAID DEBIT STANDARD-VERNMENT
MPW	DEBIT MASTERCARD BUSINESS CARD PREPAID WORK B2B

MASTERCARD	Désignation
MPX	MASTERCARD PREPAID DEBIT STANDARD-FLEX BENEFIT
MPY	MASTERCARD PREPAID DEB STANDARD-EMPLOYEE INCENTIVE
MRG	MASTERCARD PREPAID CARD
MRH	MASTERCARD UNKNOWN PRODUCT
MRW	PREPAID MASTERCARD BUSINESS CARD
MSG	PREPAID MAESTRO CONSUMER RELOADABLE CARD
MSI	MAESTRO CARD
MWB	WORLD MASTERCARD FOR BUSINESS CARD
MWE	WORLD ELITE MASTERCARD CARD
DLS	DEBIT MASTERCARD CARD-DELAYED DEBIT
MCB	MASTERCARD BUSINESSCARD CARD
MCC	MASTERCARD CREDIT CARD (MIXED BIN)
MVOIR	MASTERCARD FLEET CARD
MCG	LD MASTERCARD CARD
MCO	MASTERCARD CORPORATE CARD
MCP	MASTERCARD PURCHASING CARD
MCS	STANDARD MASTERCARD CARD
MCW	WORLD MASTERCARD CARD
MDG	LD DEBIT MASTERCARD CARD
MDH	WORLD DEBIT EMBOSSED MASTERCARD CARD
MDP	PLATINUM DEBIT MASTERCARD CARD
MDS	DEBIT MASTERCARD CARD
MIU	DEBIT MASTERCARD UNEMBOSSED
MNW	MASTERCARD WORLD CARD
MOC	MASTERCARD UNKNOWN PRODUCT
MPG	DEBIT MASTERCARD STANDARD PREPAID-GENERAL SPEND
MPC	MASTERCARD PROFESSIONAL CARD
MPL	PLATINUM MASTERCARD CARD
MPP	MASTERCARD PREPAID CARD
MRG	MASTERCARD PREPAID CARD
MRO	MASTERCARD REWARDS ONLY
MRW	PREPAID MASTERCARD BUSINESS CARD
MSB	MAESTRO SMALL BUSINESS CARD
MSI	MAESTRO CARD
MSO	MAESTRO PREPAID OTHER CARD
MSW	PREPAID MAESTRO CORPORATE CARD
OLS	MAESTRO-DELAYED DEBIT
TCB	MASTERCARD BUSINESS CARD-IMMEDIATE DEBIT
TCC	MASTERCARD (MIXED BIN)-IMMEDIATE DEBIT
TCG	LD MASTERCARD CARD-IMMEDIATE DEBIT
TCS	MASTERCARD STANDARD CARD-IMMEDIATE DEBIT
TCW	WORLD SIGNIA MASTERCARD CARD-IMMEDIATE DEBIT
TIU	MASTERCARD UNEMBOSSED — IMMEDIATE DEBIT
TNW	MASTERCARD NEW WORLD-IMMEDIATE DEBIT
TPL	PLATINUM MASTERCARD-IMMEDIATE DEBIT
WBE	MASTERCARD UNKNOWN PRODUCT

CB	Désignation
1	Carte nationale de retrait
2	Carte nationale de retrait et de paiement
3	Carte nationale de paiement
4	Carte nationale de paiement et de retrait à autorisation systématique
5	Carte nationale de paiement à autorisation systématique

9.5. Liste des devises supportées (paymentRequest.currency)

Devise	Codification ISO 4217	Nombre de chiffres après le séparateur décimal
Dollar australien (AUD)	036	2
Yen (JPY)	392	0
Dollar néo-zélandais (NZD)	554	2
Dollar des États-Unis (USD)	840	2
Franc CFP (XPF)	953	0

9.6. Liste des types de carte supportées (cardRequest.scheme / cardResponse.scheme)

Valeurs possibles pour cardRequest.scheme

Valeur	Description
AMEX	Carte de crédit American Express
BANCONTACT	Carte Maestro Bancontact Mistercash
CB	CB
E-CARTEBLEUE	e-Carte Bleue
JCB	Carte JCB
MAESTRO	Maestro
MASTERCARD	MasterCard
VISA	Visa
VISA_ELECTRON	Visa Electron
VPAY	Carte V Pay

Tableau 214 :

Valeurs possibles pour cardResponse.scheme

Valeur	Description
AMEX	Carte de crédit American Express
BANCONTACT	Carte Maestro Bancontact Mistercash
CB	CB
E-CARTEBLEUE	e-Carte Bleue
JCB	Carte JCB
MAESTRO	Maestro
MASTERCARD	MasterCard
PAYPAL	PayPal
PAYPAL_SB	PayPal - Sandbox
VISA	Visa
VISA_ELECTRON	Visa Electron
VPAY	Carte V Pay