Mercanet
paiement sécurisé
BNP PARIBAS

Mercanet

Guide Mercanet Essentiel POST

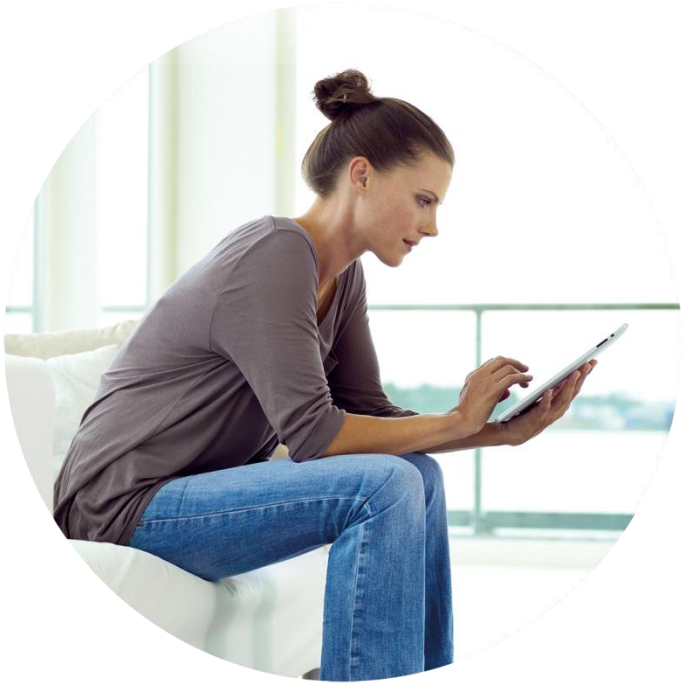


Table des matières

1 INTRODUCTION

L'objectif du présent document est d'expliquer aux Commerçants la mise en œuvre de la solution Mercanet essentiel POST et la mise en œuvre de tests initiaux de paiement.

Ce document est destiné à tous les Commerçants qui souhaitent utiliser un connecteur fondé sur les échanges HTTP(s) en mode POST entre leur site Web et les serveurs Mercanet au travers de l'offre Mecanet Essentiel POST.

Ce connecteur est conçu pour être prêt à l'emploi par le Commerçant.

✍ **A noter**

- Tous les termes, acronymes, expressions spécifiques à la Mercanet et son contexte sont définies dans le document : GLOSSAIRE
N'hésitez pas à vous y référer chaque fois que nécessaire

✍ **Important**

ATTENTION : Certains services décrits dans ce document peuvent ne pas encore être disponibles dans Mercanet.

1.1 PRINCIPES GENERAUX

Une connaissance élémentaire des standards relatifs aux langages de programmation Web pratiqués dans l'industrie, tels que Java, PHP ou .Net, est nécessaire pour développer un logiciel-client capable de se connecter à la Mercanet Essentiel Post.

Cette solution garantit que les échanges entre le site Web du Commerçant et les serveurs Mercanet sont sécurisés au moyen de clés secrètes.

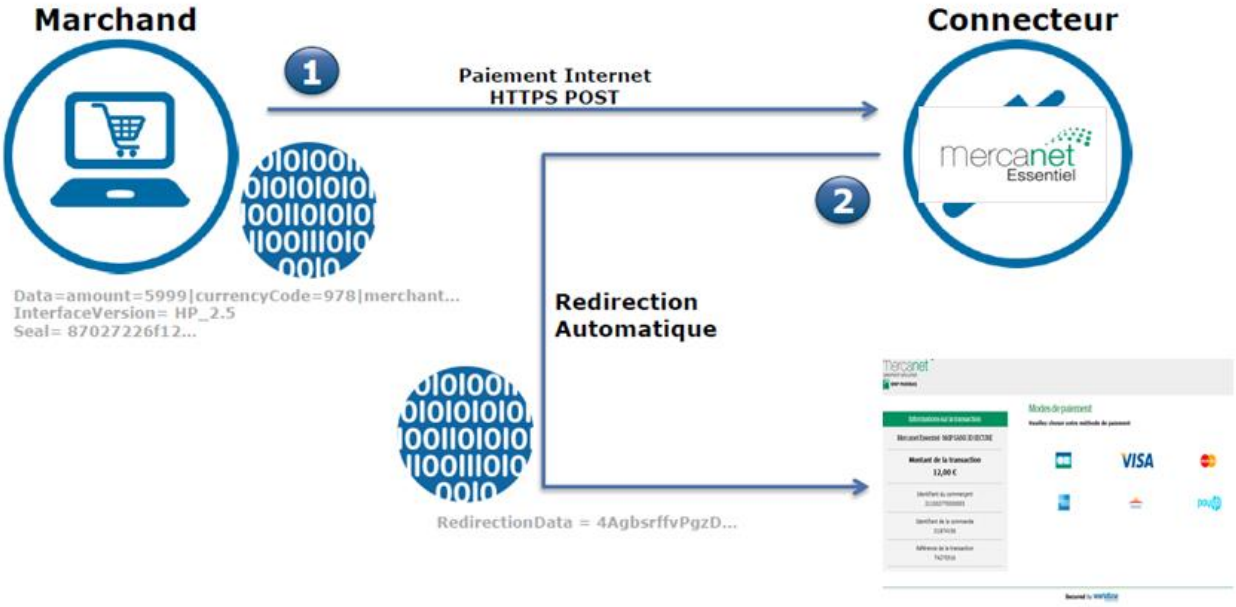
Le Commerçant est responsable de la sécurité du stockage et de la gestion de celles-ci.

✍ **Important**

Si la clé est compromise, ou si le Commerçant suppose que c'est le cas, il relève de sa responsabilité de renouveler sa clé secrète par l'intermédiaire de Mercanet Téléchargement, ou en contactant l'assistance Mercanet.

2 PROCESSUS DE PAIEMENT

2.1 PRINCIPES GENERAUX

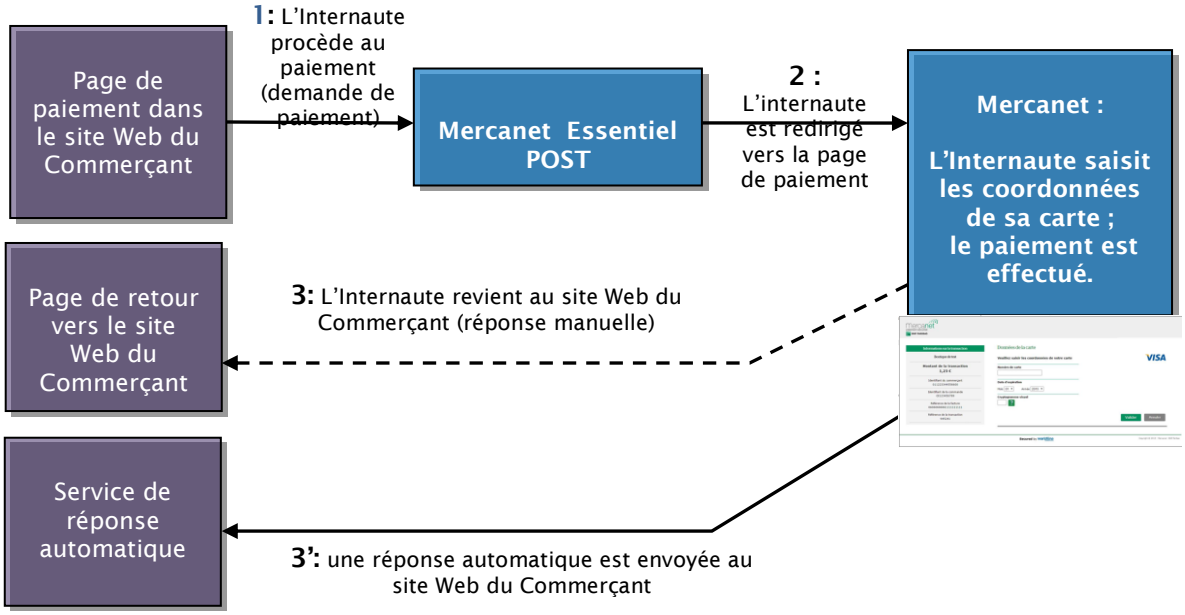


1 : Lorsque l'Internaute confirme le contenu de son panier, il est redirigé vers les serveurs de paiement Mercanet Ensuite, la demande de paiement est vérifiée et cryptée, si elle est valide, (elle est nommée RedirectionData dans le système).

2 : L'Internaute est redirigé automatiquement vers les pages de paiement Mercanet avec la demande cryptée. La demande est décryptée et la page de paiement invite l'Internaute à saisir les renseignements relatifs à son moyen de paiement.

2.2 FLUX DE PAIEMENT

Il y a trois flux à mettre en œuvre entre le site Web du Commerçant et le serveur de paiement pour intégrer la solution.



Étape 1: Lorsque l'Internaute procède à l'étape de paiement, une demande de paiement doit être envoyée à la Mercanet Essentiel POST, dont l'adresse URL est fournie au Commerçant. La meilleure méthode pour gérer cet appel est d'envoyer

un formulaire en mode POST via HTTPS. Toute autre solution capable d'envoyer une requête de cette nature fonctio**n**nera également.

Étape 2: Mercanet Essentiel Pots redirigera l'application appelante vers les pages de paiement Mercanet L'Internaute doit saisir les détails du moyen de paiement pour que le serveur de paiement Mercanet puisse prendre en charge la transaction. Il convient de noter que les détails du paiement peuvent être saisis directement sur le serveur qui propose le moyen de paiement (par exemple : PayPal). À la fin du processus de paiement, qu'il soit réussi ou non, deux réponses sont créées et envoyées à l'adresse URL si précisée lors du 1er flux.

Il y a deux procédures de notification séparées :

- Étape **3:** *Les Réponses manuelles* sont envoyées sous format HTTP(S) POST par le serveur de paiement à l'adresse URL de la *réponse normale*. Cette URL est précisée lors de la demande de paiement, lorsque l'Internaute clique le bouton « Revenir à la boutique » dans la page de paiement. C'est pourquoi l'adresse URL de *réponse normale* est en même temps la page de destination vers laquelle l'Internaute est redirigé à la fin du paiement. Il n'y a aucune garantie que l'Internaute clique ce lien. Par conséquent, il n'y a aucune garantie de recevoir la réponse manuelle.
- **Étape 3':** Les *réponses automatiques* sont envoyées indépendamment des réponses manuelles. Elles utilisent également les requêtes HTTP(s) POST envoyées par les serveurs de paiement Mercanet mais cette fois-ci moyennant l'adresse URL de la *réponse automatique* précisée lors de la demande de paiement. Cela signifie que le Commerçant recevra la réponse dès que le paiement est effectué dans les pages de paiement Mercanet

✍ **A noter**

Si le paiement a échoué, et dès que l'Internaute est redirigé vers le site Web du Commerçant, il n'est plus possible de revenir aux pages de paiement pour tenter de payer à nouveau ou pour corriger les données de carte. Le rôle du site Web du Commerçant est d'initialiser une nouvelle demande de paiement, en commençant par l'appel au connecteur Mercanet Essentiel.

3 DESCRIPTION DU PROTOCOLE

3.1 CHAMPS POST

Trois champs obligatoires sont renseignés dans les demandes de paiement et dans les réponses afférentes.

Data	Contient tous les renseignements relatifs à la transaction recueillis dans une chaîne de caractères telle que décrite au point 3.1.1.
InterfaceVersion	Version de l'interface du connecteur.
Seal	Utilisé pour valider l'intégrité des données échangées. Le champ Seal est calculé à l'aide du champ Data et du celui de la clé secrète, telle que décrite au point 3.1.2.

Un champ d'option supplémentaire est disponible :

Encode	Précise la méthode de codage utilisée dans le champ Data, tel que décrit au point 3.1.3.

✍ **Important**

Les noms de champs sont sensibles à la casse.

3.1.1 SYNTAXE DU CHAMP DATA

Le champ Data est construit conformément au format suivant :

<nom du champ 1>=<valeur du champ 1>|<nom du champ 2>=<valeur du champ 2>|<nom du champ 3>=<valeur du champ 3>|...etc.

Si le champ Data contient une liste d'objets complexes, sa représentation est construite conformément au format suivant :

< nom du champ 1 >=< valeur du champ 1>|<nom de la liste>={< nom du champ A1>=< valeur du champ A1>, < nom du champ A2>=< valeur du champ A2>},{< nom du champ B1>=< valeur du champ B1>, < nom du champ B2>=< valeur du champ B2>},{< nom du champ C1>=< valeur du champ C1>, < nom du champ C2>=< valeur du champ C2>}|< nom du champ 2>=< valeur du champ 2>

Tous les champs nécessaires pour la transaction (voir les détails dans le dictionnaire de données) doivent être inclus dans la chaîne de caractères. L'ordre des champs n'a pas d'importance.

Exemple d'une demande de paiement :

```
amount=55|currencyCode=978|merchantId=011223744550001|normalReturnUrl=http://www.normalreturnurl.com|transact
ionReference=534654|keyVersion=1
```

Exemple d'une requête de paiement avec une liste d'objets complexes :

```
amount=55|currencyCode=978|merchantId=011223744550001|normalReturnUrl=http://www.normalreturnurl.com|transact
ionReference=534654|
shoppingCartDetail.shoppingCartItemList={productName=apple,productDescription=red},{productName=pear,productD
escription=green},{productName=mango,productDescription=yellow}|keyVersion=1
```

3.1.2 SYNTAXE DU CHAMP SEAL

La valeur du champ Seal est construite comme suit :

- Concaténation du champ Data et de la clé secrète (encodée si l'option correspondante est choisie; voir point 3.1.3)
- Codage UTF-8 des données constituant le résultat de l'opération précédente
- Cryptage SHA256 des octets obtenus

Cette procédure peut être résumée comme suit :

```
SHA256( UTF-8(Data+secretKey ) )
```

3.1.3 SYNTAXE DU CHAMP ENCODE

Dans le cas où le champ Data comporte des caractères spéciaux, la valeur de ce champ doit être encodée.

Deux formats de codage sont permis : base64 ou base64Url.

Puisque le calcul de la signature se fait dans le champ Data, il convient de noter qu'après l'application du codage, c'est la valeur encodée du champ Data qui sera utilisée pour les besoins du calcul.

4 COMMENT EFFECTUER UN PAIEMENT

4.1 DEMANDE DE PAIEMENT

La demande de paiement est un appel HTTP POST adressée à la passerelle de paiement. La manière la plus simple d'appeler cette fonction est d'envoyer un formulaire HTML au moyen de la méthode POST.

4.1.1 CHAMPS PRÉVUS POUR LA DEMANDE DE PAIEMENT

Toutes les données impliquées dans la demande de paiement doivent être fournies, comme précisé au point 3 de ce chapitre.

La variable **InterfaceVersion** doit être fixée à **HP_2.9**.

Tous les réglages de la demande de paiement, leur format et leur caractère obligatoire ou facultatif sont décrits dans le dictionnaire de données au chapitre « Description des messages ».

4.1.2 EXEMPLE

Ci-dessous, voici un exemple du formulaire :

```
<form method="post" action="https://url.to.sips.server/paymentInit">
    <input type="hidden" name="Data"
value="amount=55|currencyCode=978|merchantId=011223744550001|normalReturnUrl=http://www.normalret
urnurl.com|transactionReference=534654|keyVersion=1">
    <input type="hidden" name="InterfaceVersion" value="HP_2.9">
    <input type="hidden" name="Seal"
value="21a57f2fe765e1ae4a8bf15d73fc1bf2a533f547f2343d12a499d9c0592044d4">
    <input type="submit" value="Proceed to payment">
  </form>
```

4.1.3 GESTION DES ERREURS

Tous les champs reçus par la Mercanet Essentiel POST à travers le connecteur font l'objet d'une vérification individuelle. Le tableau ci-dessous présente la liste des messages d'erreur pouvant s'afficher lors de cette étape ainsi que les solutions à mettre en œuvre.

✍ **A noter**

Les messages sont affichés seulement sur la plate-forme de simulation pour valider l'intégration du site Web du Commerçant. Pour des raisons de sécurité, les messages d'erreur beaucoup plus simples sont affichés sur la plate-forme de production. Ex « Erreur lors du traitement de la demande de paiement. Contactez votre support.

Message	Cause	Solution
Unknown version interface: <version>	La valeur <version> dans le champ POST InterfaceVersion est inconnue	Vérifier la version d'interface dans ce guide d'utilisation
Invalid keyword: <nom du paramètre>=<valeur du paramètre>	La demande contient un réglage <nom du paramètre> qui n'est pas prévu dans la demande de paiement	Vérifier les réglages de la demande de paiement dans le dictionnaire de données
Invalid field size: <nom du paramètre>=<valeur du paramètre>	La valeur du réglage <nom du paramètre> a une longueur incorrecte	Vérifier la longueur des réglages de la demande de paiement dans le dictionnaire de données
Invalid field value: <nom du paramètre>=<valeur du paramètre>	La valeur du réglage <nom du paramètre> a un format incorrect	Vérifier le format des réglages de la demande de paiement dans le dictionnaire de données
Mandatory field missing: <nom du paramètre>	Le réglage obligatoire <nom du paramètre> est manquant dans la demande de paiement	Vérifier les réglages obligatoires de la demande de paiement dans le dictionnaire de données
Unknown security version: <version>	La valeur <version> dans le réglage keyVersion est inconnue	Vérifier les versions des clés disponibles dans l'interface du Commerçant
Invalid signature	La vérification de la signature de la demande de paiement a échoué. Cela peut être causé par le calcul incorrect de la signature ou peut indiquer la falsification de certains champs après le calcul de la signature.	Vérifier les régulations concernant le calcul de la signature dans le dictionnaire de données
Transaction already processed: <référence de la transaction>	Une demande de paiement avec la même transactionReference a déjà été reçue et prise en charge par les serveurs Mercanet	Vérifier si le paramètre transactionReference est unique pour la transaction concernée
<Autres messages>	Dans le cas d'erreurs techniques, d'autres messages différents peuvent s'afficher	Contacter le service d'assistance technique

4.2 REPONSES AU PAIEMENT

Deux types de réponse sont prévus. Bien que les protocoles, formats et contenus des deux réponses soient exactement les mêmes, elles doivent être gérées de manière différente car elles répondent à deux besoins différents.

4.2.1 RÉPONSE MANUELLE

L'objectif principal de la réponse manuelle est de rediriger l'Internaute vers le site Web Marchand avec le résultat du paiement pour que le Commerçant puisse prendre la bonne décision concernant son client. Par exemple, dans le cas d'erreur, le Commerçant peut suggérer de retenter le paiement et de relancer le processus. Dans le cas de paiement réussi, le Commerçant peut afficher un message de remerciement et commencer à expédier les marchandises, si tel en est le besoin.

À la dernière étape, le processus de paiement Mercanet implique l'affichage d'un lien de redirection pour le client. Lorsque l'Internaute clique sur ce lien, le serveur Mercanet le redirige vers l'adresse URL contenue dans le champ normalReturnUrl fourni au début du processus de paiement. La redirection est une requête HTTP POST qui contient les réglages de la réponse, tels que décrits dans ce document. Il relève de la responsabilité du Commerçant de récupérer ces paramètres et vérifier la signature pour ainsi assurer l'intégrité des données de la réponse. De plus, le Commerçant est responsable d'afficher les messages pertinents (relatifs aux détails de la réponse) à son client.

Il est important de noter qu'il est impossible de garantir la réception de la réponse, celle-ci étant envoyée par le navigateur Web de l'Internaute. En effet, l'utilisateur final a la possibilité de ne pas cliquer le lien. De plus, la connexion qu'il utilise peut tout simplement éprouver un problème et bloquer la transmission de cette réponse. Par conséquent, celle-ci ne peut pas constituer la base unique pour les processus métier du Commerçant.

✍ **A noter**

Les noms de paramètres utilisés dans la réponse manuelle sont sensibles à la casse.
La version actuelle d'**InterfaceVersion** est **HP_2.12**. Veuillez consultez le dictionnaire de données pour une description complète des paramètres inclus dans la réponse.

Exemple de réponse manuelle avec la valorisation des champs Data, Seal et InterfaceVersion retournés dans la variable $_POST qui est un tableau :

```
Data=captureDay=0|captureMode=AUTHOR_CAPTURE|currencyCode=978|merchantId=002001000000001|orderCha
nnel=INTERNET|responseCode=00|transactionDateTime=2016-02-
16T14:35:57+01:00|transactionReference=test960354|keyVersion=1|acquirerResponseCode=00|amount=100
|authorisationId=351612|panExpiryDate=201701|paymentMeanBrand=VISA|paymentMeanType=CARD|complemen
taryCode=00|complementaryInfo=<RULE_RESULT CR=N SI=0 GC=0
/>,CARD_COUNTRY=FRA,<COUNTRY_COMBINATION CARD_COUNTRY=FRA IP_COUNTRY=FRA />,<CARD_INFOS BDOM=XXX
COUNTRY=FRA PRODUCTCODE=F NETWORK=VISA BANKCODE=20041 PRODUCTNAME=VISA CLASSIC PRODUCTPROFILE=XXX
/>|customerIpAddress=159.50.252.79|maskedPan=5017##########02|returnContext=123456|holderAuthentR
elegation=N|holderAuthentStatus=NO_AUTHENT|transactionOrigin=INTERNET|paymentPattern=ONE_SHOT|cus
tomerMobilePhone=null|mandateAuthentMethod=null|mandateUsage=null|transactionActors=null|mandateI
d=null|captureLimitDate=20160216|dccStatus=null|dccResponseCode=null|dccAmount=null|dccCurrencyCo
de=null|dccExchangeRate=null|dccExchangeRateValidity=null|dccProvider=null|statementReference=nul
l|panEntryMode=MANUAL|walletType=null|holderAuthentMethod=NO_AUTHENT_METHOD|holderAuthentProgram=
NO_AUTHENT|paymentMeanId=null|instalmentNumber=null|instalmentDatesList=null|instalmentTransactio
nReferencesList=null|instalmentAmountsList=null|settlementMode=null|mandateCertificationType=null
|valueDate=null|creditorId=null|acquirerResponseIdentifier=null|acquirerResponseMessage=null|paym
entMeanTradingName=null|additionalAuthorisationNumber=null

Seal=e421aa1a9b7de0d93477de7082562e6960e3832ed51e41974a1bee8fab9e1252

InterfaceVersion=HP_2.8
```

4.2.2 RÉPONSE AUTOMATIQUE

La réponse automatique est envoyée seulement si le champ automaticResponseUrl était envoyé dans la demande de paiement. Si tel est le cas, le serveur Mercanet envoie une réponse HTTP POST à l'adresse URL reçue. Les champs de la réponse sont identiques à ceux de la réponse manuelle. La seule différence entre les deux procédures est que la réponse automatique est envoyée directement par le serveur Mercanet sans passer par le navigateur Web de l'Internaute. Par conséquent, elle est bien plus fiable car elle sera toujours envoyée. L'autre conséquence, c'est que la procédure de réception de cette réponse ne doit pas tenter de répondre à l'application appelante. En principe, le serveur Mercanet n'attend aucune réponse après la transmission de la réponse automatique.

Comme pour la réponse manuelle, les champs de la réponse automatique sont décrits dans ce document. Il appartient au Commerçant de récupérer les réglages de la réponse, les enregistrer sous forme cryptée, vérifier la signature pour s'assurer de l'intégrité des champs de la réponse et, par conséquent, mettre à jour son système back office.

✍ **A noter**

Les noms de paramètres utilisés dans la réponse automatique sont sensibles à la casse.
La version actuelle d'**InterfaceVersion** est **HP_2.12**. Veuillez consultez le dictionnaire de données pour une description complète des paramètres inclus dans la réponse.

4.2.3 PROBLÈMES AVEC LA RÉCEPTION DES RÉPONSES MERCANET

Ci-dessous, une liste des problèmes les plus couramment observés qui bloquent la réception des réponses automatiques et manuelles. Le Commerçant doit les vérifier avant d'appeler le service d'assistance.

- Vérifier si les adresses URL de réponse sont fournies dans la demande de paiement et si elles sont valides. Pour le faire, le Commerçant peut tout simplement les copier et coller dans son navigateur.
- Les adresses URL fournies doivent être accessibles à distance, c'est-à-dire de l'Internet. Le contrôle d'accès (identifiant/mot de passe ou filtre IP) ou le pare-feu peuvent bloquer l'accès à votre serveur.
- L'accès aux adresses URL de réponse doit être confirmé dans le journal des notifications du serveur Web du Commerçant.
- Si le Commerçant utilise un port non standard, celui-ci doit être compris entre 80 et 9999 pour assurer la compatibilité avec Mercanet
- Il est impossible d'ajouter des paramètres du contexte aux adresses URL de réponse. Le champ orderID est prévu pour les paramètres supplémentaires. Éventuellement, le Commerçant peut se servir du champ sessionId pour retrouver les renseignements sur son client à la fin du processus de paiement.

4.2.4 GESTION DES ERREURS : PAS DE SIGNATURE DANS LA RÉPONSE

Dans certains cas d'erreurs, le serveur Mercanet n'est pas capable de signer le message de réponse. Cela s'applique, par exemple, à l'erreur « Identifiant MerchantID inconnu » et à la situation où la clé secrète est inconnue à la référence Mercanet

Pour ces raisons, le serveur de paiement enverra une réponse sans signature dans le champ Seal.

5 COMMENT SIGNER UN MESSAGE

5.1 LA RAISON POUR SIGNER UN MESSAGE

La demande de paiement contient les paramètres de la transaction et est envoyée par le navigateur Web de l'Internaute. Théoriquement, il est possible pour un pirate d'intercepter la demande et de changer les réglages avant que les données n'atteignent le serveur de paiement.
De ce fait, il est nécessaire de renforcer la sécurité pour assurer l'intégrité des paramètres de la transaction envoyée. La solution Mercanet répond à ce besoin par échange de signatures.
Un contrôle effectif de la signature comporte deux éléments :

- l'**intégrité** des messages de demande et de réponse ; l'absence de modifications lors de l'échange,
- l'**authentification** de l'émetteur et du destinataire, car ils se partagent la même clé secrète.

✍ A noter

Si la clé utilisée pour signer est compromise, ou si le Commerçant suppose que c'est le cas, il lui appartient d'en demander le renouvellement en se connectant à Mercanet Téléchargement ou en contactant l'assistance Mercanet.

5.2 METHODE UTILISEE POUR SIGNER UN MESSAGE

L'opération de signature est effectuée en calculant la valeur cryptée conformément aux paramètres de la transaction (champ Data). Ensuite, la clé secrète y est ajoutée. Toutes les chaînes de caractères sont converties en UTF-8 avant le cryptage.

L'algorithme de cryptage (SHA256) génère un résultat irréversible. En principe, lorsqu'un tel message est reçu, le destinataire doit recalculer la valeur cryptée pour la comparer à celle reçue. Toute différence indique que les données échangées ont été falsifiées.

Le résultat doit être envoyé sous forme hexadécimale dans le champ POST nommée Seal.

5.3 EXEMPLES DU CODE

5.3.1 PHP 5

```php
<?php
echo hash('sha256', $data.$secretKey);
?>
```

Le jeu de caractères UTF-8 doit être utilisé dans les champs Data et secretKey. Pour effectuer une conversion de ISO-8859-1 à UTF-8, faites appel à la fonction **utf8_encode**.

5.3.2 JAVA

```
import java.security.MessageDigest;

public class ExampleSHA256 {

        /**
         * table to convert a nibble to a hex char.
         */
        static final char[] hexChar = {
           '0' , '1' , '2' , '3' ,
           '4' , '5' , '6' , '7' ,
           '8' , '9' , 'a' , 'b' ,
           'c' , 'd' , 'e' , 'f'};

        /**
         * Fast convert a byte array to a hex string
         * with possible leading zero.
         * @param b array of bytes to convert to string
         * @return hex representation, two chars per byte.
         */
        public static String encodeHexString ( byte[] b )
           {
           StringBuffer sb = new StringBuffer( b.length * 2 );
           for ( int i=0; i<b.length; i++ )
              {
              // look up high nibble char
              sb.append( hexChar [( b[i] & 0xf0 ) >>> 4] );

              // look up low nibble char
              sb.append( hexChar [b[i] & 0x0f] );
              }
           return sb.toString();
           }

        /**
         * Computes the seal
         * @param Data the parameters to cipher
         * @param secretKey the secret key to append to the parameters
         * @return hex representation of the seal, two chars per byte.
         */
        public static String computeSeal(String Data, String secretKey) throws Exception
        {
          MessageDigest md = MessageDigest.getInstance("SHA-256");
          md.update((Data+secretKey).getBytes("UTF-8"));

          return encodeHexString(md.digest());
        }

        /**
         * @param args
         */
        public static void main(String[] args) {
                try {
                        System.out.println (computeSeal("parameters", "key"));
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

5.3.3 .NET

(Complété à l'aide d'un simple formulaire appelé « Form 1 » contenant deux champs de texte à renseigner : `txtSips`, `txtSecretKey` et un autre à afficher : `lblHEX`)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Security.Cryptography;

namespace ExampleDotNET
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdGO_Click(object sender, EventArgs e)
        {
            String sChaine = txtSips.Text + txtSecretKey.Text;
            UTF8Encoding utf8 = new UTF8Encoding();
            Byte[] encodedBytes = utf8.GetBytes(sChaine);

            byte[] shaResult;
            SHA256 shaM = new SHA256Managed();
            shaResult = shaM.ComputeHash(encodedBytes);

            lblHEX.Text = ByteArrayToHEX(shaResult);
        }

        private string ByteArrayToHEX(byte[] ba)
        {
            StringBuilder hex = new StringBuilder(ba.Length * 2);
            foreach (byte b in ba)
                hex.AppendFormat("{0:x2}", b);
            return hex.ToString();
        }

    }
}
```

6 COMMENT TESTER

Les étapes de tests et d'intégration doivent être effectuées à l'aide de **l'environnement de démonstration.**

Les détails techniques détaillées concernant l'utilisation de cet environnement sont décrits ci-dessous :

URL de démo du connecteur	cf. guide « **URL et Dictionnaire des données »**
ID du Commerçant	002001000000001
Version de la clé	1
Clé sécrète	002001000000001_KEY1

Dans l'environnement de simulation, le processus d'autorisation est simulé. Cela signifie qu'il n'est pas nécessaire d'utiliser des moyens de paiement réels pour effectuer les tests.

✍ A noter

Puisque l'identifiant Merchant ID est partagé entre tous les Commerçants/Prospects, il existe un risque de duplication de la transactionReference. Par conséquent, il est vivement recommandé que tous les noms transactionReference soient préfixés par le nom de la future boutique qui sera utilisée dans l'environnement de production.

6.1 TESTS DE TRANSACTIONS PAR CARTE

Si le Commerçant choisit VISA, MASTERCARD ou MAESTRO, il sera redirigé vers la page de renseignements sur la carte où il pourra saisir les données détaillées de sa carte.

Les règles de simulation suivantes s'appliquent à toutes les cartes :

- Le PAN doit comporter de 16 à 19 chiffres.
- Les six premiers chiffres du PAN déterminent le type de carte, conformément au tableau ci-dessous :

Type de carte	Début du numéro de carte
VISA	410000
MASTERCARD	510000
MAESTRO	500000

- Le Commerçant peut simuler tous les codes de réponse (cf. dictionnaire de données) en changeant les deux derniers chiffres.
- Le code de sécurité comporte trois ou quatre chiffres. Cette valeur est sans importance pour le résultat de la transaction.

Exemple: si le Commerçant utilise le numéro de carte 4100000000000005, la carte sera identifiée comme VISA et le paiement sera refusé (code de réponse 05).

6.2 TEST DE TRANSACTION IDEAL

Si le Commerçant fait le choix iDEAL, il sera redirigé vers le serveur qui simule les transactions iDEAL selon leur montant. Ensuite, il retournera au serveur de paiement qui affiche le ticket avec le résultat de la transaction.

Règles de la simulation d'un paiement iDEAL :

Montant de la transaction	Réponse de iDeal
2,00 EUR	Transaction annulée
3,00 EUR	Transaction expirée
4,00 EUR	Transaction non réalisée
5,00 EUR	Échec de la transaction
Autres cas	Transaction OK

7 COMMENT DEMARRER EN PRODUCTION ?

L'étape suivante est de se connecter à l'environnement de production pour un lancement réel.

Pour le faire, le Commerçant doit changer l'adresse URL du serveur de paiement et utiliser les ID du Commerçant reçus à l'étape de l'inscription.

7.1 IDENTIFIANT DU COMMERÇANT

L'adresse URL du serveur de paiement est précisée dans le guide « **URL et Dictionnaire des données ».**

Pour accéder à l'environnement de production, les informations suivantes sont nécessaires :

- l'identifiant du Commerçant (**merchantID)** qui identifie le site de commerce en ligne sur le serveur de paiement Mercanet,
- la version (**keyVersion)** de la clé sécrète,
- la clé sécrète (**secretKey)** utilisée pour signer les demandes et vérifier les réponses.

L'identifiant du Commerçant (**merchantID)** est fourni par l'assistance technique suite à l'inscription à Mercanet Essentiel.

Le Commerçant peut télécharger la version de la clé (**keyVersion)** et de la clé secrète (**secretKey)** depuis Mercanet Téléchargement (cf. chapitre 8) en se servant du nom de l'utilisateur et du mot de passe fournis suite à son inscription.

7.2 VALIDATION DANS L'ENVIRONNEMENT DE PRODUCTION

Dès que le Commerçant commence à utiliser son propre identifiant sur le serveur de production, toute transaction effectuée est une transaction réelle de bout en bout, à savoir un débit du compte de l'acheteur et un crédit du compte du Commerçant.

Avant d'ouvrir effectivement sa boutique au public, le Commerçant peut envoyer une requête pour valider le paiement (de faible montant) de bout en bout, et le rembourser si besoin.

Important

Avant d'ouvrir effectivement sa boutique au public, il est très fortement conseillé au Commerçant d'envoyer une requête pour valider un paiement (de faible montant) de bout en bout, pour vérifier en production le bon fonctionnement de Mercanet.
Il peut ensuite procéder à son rembourser si besoin.

8 DESCRIPTION DU MESSAGE

8.1 DEMANDE DE PAIEMENT

8.1.1 CHAMPS GÉNÉRIQUES

Nom du champ	Présence	Dans la version	Commentaires
amount	Obligatoire	HP_1.0	
automaticResponseUrl	Optionnel	HP_1.0	
billingFirstDate	Optionnel	HP_2.5	
captureDay	Optionnel	HP_1.0	
captureMode	Optionnel	HP_1.0	
currencyCode	Obligatoire	HP_1.0	
customer3DSTransactionDate	Optionnel	HP_2.5	
customerBillingNb	Optionnel	HP_2.5	
customerDeliverySuccessFlag	Optionnel	HP_2.5	
customerId	Optionnel	HP_2.0	
customerIpAddress	Optionnel	HP_2.1	
customerLanguage	Optionnel	HP_1.0	
customerPhoneValidationMethod	Optionnel	HP_2.5	
customerRegistrationDateOnline	Optionnel	HP_2.5	
customerRegistrationDateProxi	Optionnel	HP_2.5	
deliveryFirstDate	Optionnel	HP_2.5	
evidenceAcquisitionDate	Optionnel	HP_2.5	
evidenceNumber	Optionnel	HP_2.5	
evidenceType	Optionnel	HP_2.5	
expirationDate	Optionnel	HP_1.0	
hashAlgorithm1	Optionnel	HP_2.3	
hashAlgorithm2	Optionnel	HP_2.3	
hashSalt1	Optionnel	HP_2.1	
hashSalt2	Optionnel	HP_2.1	
holderAdditionalReference	Optionnel	HP_2.9	
invoiceReference	Optionnel	HP_2.0	
keyVersion	Obligatoire	HP_1.0	
mandateId	Optionnel	HP_2.5	
merchantId	Obligatoire	HP_1.0	
merchantSessionId	Optionnel	HP_2.0	
merchantTransactionDateTime	Optionnel	HP_2.0	
merchantWalletID	Optionnel	HP_2.2	
normalReturnUrl	Obligatoire	HP_1.0	
orderChannel	Optionnel	HP_2.1	
orderId	Optionnel	HP_1.0	
paymentMeanBrandList	Optionnel	HP_1.0	
paymentPattern	Optionnel	HP_2.1	Ce champ est obligatoire pour certains moyens de paiement. Se référer au guide d'implémentation du moyen de paiement concerné pour plus de détail.
returnContext	Optionnel	HP_2.0	
riskManagementCustomDataList	Optionnel	HP_2.9	
statementReference	Optionnel	HP_2.3	
templateName	Optionnel	HP_2.1	
transactionActors	Optionnel	HP_2.2	
transactionOrigin	Optionnel	HP_2.0	
transactionReference	Optionnel	HP_1.0	

Nom du champ	Présence	Dans la version	Commentaires
valueDate	Optionnel	HP_2.5	

Tableau 1 : Demande de paiement

8.1.2 CHAMPS OPTIONNELS RELATIFS À LA FRAUDE

Champ	Présence	Dans la version	Commentaires
fraudData.allowedCardArea	Optionnel	HP_2.1	
fraudData.allowedCardCountryList	Optionnel	HP_2.1	
fraudData.allowedIpArea	Optionnel	HP_2.1	
fraudData.allowedIpCountryList	Optionnel	HP_2.1	
fraudData.bypass3DS	Optionnel	HP_2.1	
fraudData.bypassCtrlList	Optionnel	HP_2.1	
fraudData.bypassInfoList	Optionnel	HP_2.1	
fraudData.deniedCardArea	Optionnel	HP_2.1	
fraudData.deniedCardCountryList	Optionnel	HP_2.1	
fraudData.deniedIpArea	Optionnel	HP_2.1	
fraudData.deniedIpCountryList	Optionnel	HP_2.1	

Tableau 2 : Détails des champs relatifs à la fraude

8.1.3 CHAMPS OPTIONNELS RELATIFS AUX PAGES DE PAIEMENT

Champ	Présence	Dans la version	Commentaires
paypageData.bypassReceiptPage	Optionnel	HP_2.0	

Tableau 3 : Détails des champs concernant le fonctionnement des pages de paiement

8.1.4 CHAMPS OPTIONNELS RELATIFS À L'AUTHENTIFICATION

Pour IssuerWalletPolicy

Champ	Presence	Dans la version	Commentaires
authenticationData.issuerWalletPolicy. check3DS	Optionnel	HP_2.2	
authenticationData.issuerWalletPolicy. checkCSC	Optionnel	HP_2.2	

Tableau 1: Détails des champs concernant l'authentification par Wallet

For CardAuthPolicy

Champ	Présence (M/O)	Dans la version	Commentaires
authenticationData.cardAuthPolicy. checkAVS	Optionnel	HP_2.8	
authenticationData.cardAuthPolicy. ignoreAddressCheckResult	Optionnel	HP_2.8	
authenticationData.cardAuthPolicy. ignorePostcodeCheckResult	Optionnel	HP_2.8	

Tableau 2: Détails des champs concernant l'authentification par carte

8.1.5 CHAMPS OPTIONNELS RELATIFS AUX MOYENS DE PAIEMENT

Pour PayPal

Champ	Présence	Dans la version	Commentaires
paymentMeanData.paypal.landingPage	Optionnel	HP_2.2	
paymentMeanData.paypal.addrOverride	Optionnel	HP_2.2	
paymentMeanData.paypal.invoiceId	Optionnel	HP_2.2	
paymentMeanData.paypal.dupFlag	Optionnel	HP_2.2	
paymentMeanData.paypal.dupDesc	Optionnel	HP_2.2	
paymentMeanData.paypal.dupCustom	Optionnel	HP_2.2	
paymentMeanData.paypal.dupType	Optionnel	HP_2.2	

Champ	Présence	Dans la version	Commentaires
paymentMeanData.paypal.mobile	Optionnel	HP_2.2	

Tableau 5 : Champs relatifs à PayPal

For Accord

Champ	Présence	Dans la version	Commentaires
paymentMeanData.accord.settlementMode	Optionnel	HP_2.6	

Tableau 3: Champs relatifs à Accord

For Facilypay

Champ	Présence	Dans la version	Commentaires
paymentMeanData.facilypay.depositRefundIndicator	Optionnel	HP_2.6	
paymentMeanData.facilypay.receiverType	Optionnel	HP_2.6	
paymentMeanData.facilypay.settlementMode	Optionnel	HP_2.6	
paymentMeanData.facilypay.settlementModeVersion	Optionnel	HP_2.6	

Tableau 4: Champs relatifs à Facilypay

For CetelemNxcb

Champ	Présence	Dans la version	Commentaires
paymentMeanData.cetelemNxcb.nxcbTransactionReference1	Optionnel	WS_2.9	
paymentMeanData.cetelemNxcb.nxcbTransactionReference2	Optionnel	WS_2.9	
paymentMeanData.cetelemNxcb.s10NxcbTransactionId1	Optionnel	WS_2.9	
paymentMeanData.cetelemNxcb.s10NxcbTransactionId2	Optionnel	WS_2.9	

Tableau 5: Champs relatifs à CetelemNxcb

8.1.6 CHAMPS OPTIONNELS POUR LE PAIEMENT ÉCHELONNÉ

Champ	Présence	Dans la version	Commentaires
instalmentData.number	Optionnel	HP_2.2	
instalmentData.datesList	Optionnel	HP_2.2	
instalmentData.transactionReferencesList	Optionnel	HP_2.2	
instalmentData.amountsList	Optionnel	HP_2.2	

Tableau 9 : Champs relatifs aux paiements récurrents

8.1.7 CHAMPS OPTIONNELS POUR LES DONNÉES DE FACTURATION

8.1.7.1 Données d'entrée billingAddress

Champ	Présence	Dans la version	Commentaires
billingAddress.addressAdditional1	Optionnel	HP_2.2	
billingAddress.addressAdditional2	Optionnel	HP_2.2	
billingAddress.addressAdditional3	Optionnel	HP_2.2	
billingAddress.city	Optionnel	HP_2.2	
billingAddress.company	Optionnel	HP_2.2	
billingAddress.country	Optionnel	HP_2.2	
billingAddress.postBox	Optionnel	HP_2.2	

Champ	Présence	Dans la version	Commentaires
billingAddress.state	Optionnel	HP_2.2	
billingAddress.street	Optionnel	HP_2.2	
billingAddress.streetNumber	Optionnel	HP_2.2	
billingAddress.zipCode	Optionnel	HP_2.2	

Tableau 10 : Champs prévus pour l'élément billingAddress

8.1.7.2 Données d'entrée billingContact

Champ	Présence	Dans la version	Commentaires
billingContact.email	Optionnel	HP_2.2	
billingContact.firstname	Optionnel	HP_2.2	
billingContact.gender	Optionnel	HP_2.2	
billingContact.lastname	Optionnel	HP_2.2	
billingContact.mobile	Optionnel	HP_2.2	
billingContact.phone	Optionnel	HP_2.2	
billingContact.title	Optionnel	HP_2.2	

Tableau 11 : Champs prévus pour l'élément billingContact

8.1.8 CHAMPS OPTIONNELS POUR LES DONNÉES CLIENT

8.1.8.1 Données d'entrée customerAddress

Champ	Présence	Dans la version	Commentaires
customerAddress.addressAdditional1	Optionnel	HP_2.2	
customerAddress.addressAdditional2	Optionnel	HP_2.2	
customerAddress.addressAdditional3	Optionnel	HP_2.2	
customerAddress.city	Optionnel	HP_2.2	
customerAddress.company	Optionnel	HP_2.2	
customerAddress.country	Optionnel	HP_2.2	
customerAddress.postBox	Optionnel	HP_2.2	
customerAddress.state	Optionnel	HP_2.2	
customerAddress.street	Optionnel	HP_2.2	
customerAddress.streetNumber	Optionnel	HP_2.2	
customerAddress.zipCode	Optionnel	HP_2.2	

Tableau 12 : Champs prévus pour l'élément customerAddress

8.1.8.2 Données d'entrée customerContact

Champ	Présence	Dans la version	Commentaires
customerContact.email	Optionnel	HP_2.2	
customerContact.firstname	Optionnel	HP_2.2	
customerContact.gender	Optionnel	HP_2.2	
customerContact.lastname	Optionnel	HP_2.2	
customerContact.mobile	Optionnel	HP_2.2	
customerContact.phone	Optionnel	HP_2.2	
customerContact.title	Optionnel	HP_2.2	

Tableau 13 : Champs prévus pour l'élément customerContact

8.1.8.3 Données d'entrée customerData

Champ	Présence	Dans la version	Commentaires
customerData.birthCity	Optionnel	HP_2.2	
customerData.birthCountry	Optionnel	HP_2.2	
customerData.birthDate	Optionnel	HP_2.2	
customerData.birthZipCode	Optionnel	HP_2.2	
customerData.nationalityCountry	Optionnel	HP_2.2	
customerData.newPwd	Optionnel	HP_2.2	

Champ	Présence	Dans la version	Commentaires
customerData.pwd	Optionnel	HP_2.2	

Tableau 14 : Champs prévus pour l'élément customerData

8.1.9 CHAMPS OPTIONNELS POUR LES DONNÉES DE LIVRAISON

8.1.9.1 Données d'entrée deliveryAddress

Champ	Présence	Dans la version	Commentaires
deliveryAddress.addressAdditional1	Optionnel	HP_2.2	
deliveryAddress.addressAdditional2	Optionnel	HP_2.2	
deliveryAddress.addressAdditional3	Optionnel	HP_2.2	
deliveryAddress.city	Optionnel	HP_2.2	
deliveryAddress.company	Optionnel	HP_2.2	
deliveryAddress.country	Optionnel	HP_2.2	
deliveryAddress.postBox	Optionnel	HP_2.2	
deliveryAddress.state	Optionnel	HP_2.2	
deliveryAddress.street	Optionnel	HP_2.2	
deliveryAddress.streetNumber	Optionnel	HP_2.2	
deliveryAddress.zipCode	Optionnel	HP_2.2	

Tableau 15 : Champs prévus pour l'élément deliveryAddress

8.1.9.2 Données d'entrée deliveryContact

Champ	Présence	Dans la version	Commentaires
deliveryContact.email	Optionnel	HP_2.2	
deliveryContact.firstname	Optionnel	HP_2.2	
deliveryContact.gender	Optionnel	HP_2.2	
deliveryContact.lastname	Optionnel	HP_2.2	
deliveryContact.mobile	Optionnel	HP_2.2	
deliveryContact.phone	Optionnel	HP_2.2	
deliveryContact.Title	Optionnel	HP_2.2	

Tableau 16 : Champs prévus pour l'élément deliveryContact

8.1.10 CHAMPS OPTIONNELS POUR LES DONNÉES DU TITULAIRE

8.1.10.1 Données d'entrée holderAddress

Champ	Présence	Dans la version	Commentaires
holderAddress.addressAdditional1	Optionnel	HP_2.2	
holderAddress.addressAdditional2	Optionnel	HP_2.2	
holderAddress.addressAdditional3	Optionnel	HP_2.2	
holderAddress.city	Optionnel	HP_2.2	
holderAddress.company	Optionnel	HP_2.2	
holderAddress.country	Optionnel	HP_2.2	
holderAddress.postBox	Optionnel	HP_2.2	
holderAddress.state	Optionnel	HP_2.2	
holderAddress.street	Optionnel	HP_2.2	
holderAddress.streetNumber	Optionnel	HP_2.2	
holderAddress.zipCode	Optionnel	HP_2.2	

Tableau 17 : Champs prévus pour l'élément holderAddress

8.1.10.2 Données d'entrée holderAddress

Champ	Présence	Dans la version	Commentaires
holderContact.email	Optionnel	HP_2.2	
holderContact.firstname	Optionnel	HP_2.2	

Champ	Présence	Dans la version	Commentaires
holderContact.gender	Optionnel	HP_2.2	
holderContact.lastname	Optionnel	HP_2.2	
holderContact.mobile	Optionnel	HP_2.2	
holderContact.phone	Optionnel	HP_2.2	
holderContact.title	Optionnel	HP_2.2	

Tableau 18 : Champs prévus pour l'élément holderContact

8.1.11 CHAMPS OPTIONNELS POUR LES DONNEES DU PANIER

8.1.11.1 Données d'entrée shoppingCartDetail

Champ	Présence	Dans la version	Commentaires
shoppingCartDetail.mainProduct	Optionnel	HP_2.6	
shoppingCartDetail.shoppingCartItemList	Optionnel	HP_2.6	Une liste de shoppingCartItem
shoppingCartDetail.shoppingCartTotalAmount	Optionnel	HP_2.6	
shoppingCartDetail.shoppingCartTotalQuantity	Optionnel	HP_2.6	
shoppingCartDetail.shoppingCartTotalTaxAmount	Optionnel	HP_2.7	

Tableau 6: Champs prévus pour l'élément shoppingCartDetail

8.1.11.2 Données d'entrée shoppingCartItem

Champ	Présence	Dans la version	Commentaires
shoppingCartItem.productCategory	Optionnel	HP_2.6	
shoppingCartItem.productCode	Optionnel	HP_2.6	
shoppingCartItem.productDescription	Optionnel	HP_2.6	
shoppingCartItem.productName	Optionnel	HP_2.6	
shoppingCartItem.productQuantity	Optionnel	HP_2.6	
shoppingCartItem.productSKU	Optionnel	HP_2.6	
shoppingCartItem.productTaxRate	Optionnel	HP_2.6	
shoppingCartItem.productUnitAmount	Optionnel	HP_2.6	
shoppingCartItem.productUnitTaxAmount	Optionnel	HP_2.6	

Tableau 7: Champs prévus pour l'élément shoppingCartItem

8.1.12 CHAMPS OPTIONNELS POUR LES ID DE TRANSACTION MERCANET ANCIENNE VERSION

8.1.12.1 Données d'entrée s10TransactionReference

Champ	Présence	Dans la version	Commentaires
s10TransactionReference.s10TransactionId	Optionnel	HP_2.7	
s10TransactionReference.s10TransactionIdDate	Optionnel	HP_2.7	

Tableau 8: Champs prévus pour l'élément s10TransactionReference

8.2 REPONSES (AUTOMATIQUES ET MANUELLES)

Le contenu des réponses Web automatiques et manuelles de Mercanet est identique. Le contenu lui-même peut varier selon le résultat du paiement (réussi ou autre).

Champ	Présence	Dans la version	Commentaires
acquirerNativeResponseCode	Optionnel	HP_2.12	
acquirerResponseCode	Obligatoire	HP_2.0	
acquirerResponseIdentifier	Optionnel	HP_2.8	
acquirerResponseMessage	Optionnel	HP_2.8	
additionalAuthorisationNumber	Optionnel	HP_2.8	
amount	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
authorisationId	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
captureDay	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
captureLimiteDate	Optionnel	HP_2.3	
captureMode	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
cardCSCResultCode	Obligatoire	HP_2.0	
cardProductCode	Optionnel	HP_2.12	
cardProductName	Optionnel	HP_2.12	
cardProductProfile	Optionnel	HP_2.12	
complementaryCode*	Optionnel	HP_1.0	
complementaryInfo*	Optionnel	HP_2.0	
creditorId	Optionnel	HP_2.7	
currencyCode	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
customerEmail	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement. Seulement disponible en HP_2.0
customerId	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
customerIpAddress	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
customerMobilePhone	Obligatoire	HP_2.1	Valeur véhiculée dans la requête de paiement. Seulement disponible en HP_2.1
dccAmount	Optionnel	HP_2.3	
dccCurrencyCode	Optionnel	HP_2.3	
dccExchangeRate	Optionnel	HP_2.3	
dccExchangeRateValidity	Optionnel	HP_2.3	
dccProvider	Optionnel	HP_2.3	
dccStatus	Optionnel	HP_2.3	
dccResponseCode	Optionnel	HP_2.3	
dueDate	Optionnel	HP_2.3	
guaranteeIndicator	Obligatoire	HP_2.0	
hashPan1	Obligatoire	HP_2.0	
hashPan2	Obligatoire	HP_2.0	
holderAuthentMethod*	Optionnel	HP_2.4	
holderAuthentProgram	Optionnel	HP_2.5	
holderAuthentRelegation*	Optionnel	HP_2.0	
holderAuthentStatus*	Optionnel	HP_2.0	
instalmentAmountsList	Optionnel	HP_2.6	
instalmentDatesList	Optionnel	HP_2.6	
instalmentNumber	Optionnel	HP_2.6	
instalmentTransactionReferencesList	Optionnel	HP_2.6	
interfaceVersion*	Optionnel	HP_1.0	
invoiceReference	Optionnel	HP_2.10	
issuerCode	Optionnel	HP_2.12	
issuerCountryCode	Optionnel	HP_2.12	
issuerEnrollementIndicator*	Optionnel	HP_2.0	
issuerWalletInformation	Optionnel	HP_2.9	
keyVersion	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
mandateAuthentMethod	Optionnel	HP_2.2	
mandateCertificationType	Optionnel	HP_2.7	
mandateId	Optionnel	HP_2.3	
mandateUsage	Optionnel	HP_2.2	

Champ	Présence	Dans la version	Commentaires
maskedPan*	Optionnel	HP_1.0	
merchantId	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
merchantSessionId	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
merchantTransactionDateTime	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
merchantWalletID	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
orderChannel	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
orderId	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
panEntryMode*	Optionnel	HP_2.4	
panExpiryDate*	Optionnel	HP_2.0	
paymentMeanBrand*	Optionnel	HP_1.0	
paymentMeanData*	Optionnel	HP_2.2	
paymentMeanId	Optionnel	HP_2.6	
paymentMeanTradingName	Optionnel	HP_2.8	
paymentMeanType*	Obligatoire	HP_1.0	
paymentPattern	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
preAuthenticationColor	Optionnel	HP_2.10	
preAuthenticationInfo	Optionnel	HP_2.10	
preAuthenticationProfile	Optionnel	HP_2.10	
preAuthenticationThreshold	Optionnel	HP_2.10	
preAuthenticationValue	Optionnel	HP_2.10	
responseCode	Obligatoire	HP_1.0	
returnContext	Optionnel	HP_1.0	Valeur véhiculée dans la requête de paiement.
s10TransactionId	Optionnel	HP_2.9	
s10TransactionIdDate	Optionnel	HP_2.9	
s10transactionIdsList	Optionnel	HP_2.11	
scoreColor*	Optionnel	HP_2.0	
scoreInfo*	Optionnel	HP_2.0	
scoreProfile*	Optionnel	HP_2.0	
scoreThreshold*	Optionnel	HP_2.0	
scoreValue*	Optionnel	HP_2.0	
settlementMode	Optionnel	HP_2.7	
statementReference*	Optionnel	HP_2.4	
tokenPan*	Optionnel	HP_2.0	
transactionActors	Obligatoire	HP_2.2	Valeur véhiculée dans la requête de paiement.
transactionDateTime	Obligatoire	HP_1.0	
transactionOrigin	Obligatoire	HP_2.0	Valeur véhiculée dans la requête de paiement.
transactionReference	Obligatoire	HP_1.0	Valeur véhiculée dans la requête de paiement.
valueDate	Optionnel	HP_2.7	
walletType	Optionnel	HP_2.4	

Tableau 9: Champs prévus pour la réponse automatique/manuelle au paiement

*: champs renseignés s'ils sont disponibles, en fonction de l'état de la transaction et du moyen de paiement choisi.

FIN DU DOCUMENT