03/10/2016

Sogenactif

Sogenactif Paypage Post

SOMMAIRE

1 Introduction

1.1 A propos de Sogenactif

Pour mieux appréhender la solution, nous vous conseillons de visiter le site

http://www.sogenactif.fr/

Tous les termes, acronymes, expressions spécifiques à Sogenactif et son contexte sont définis dans le document [**Glossaire**].

Merci de vous y référer chaque fois que nécessaire.

1.2 Objectif du présent document

L'objectif du présent document est d'expliquer la mise en œuvre de la solution Sogenactif Paypage POST et la mise en œuvre de tests initiaux de paiement.

Ce document est destiné à tous les Commerçants qui souhaitent souscrire à l'offre Sogenactif et utiliser un connecteur fondé sur les échanges HTTP(s) en mode POST entre les sites Web du Commerçant et les serveurs Sogenactif au moyen de la passerelle Sogenactif Paypage POST. Il s'adresse à l'équipe technique du Commerçant et non à l'équipe commerciale.

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

1.3 Prérequis

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 passerelle Sogenactif Paypage POST.

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

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

✍ Remarques

Si la clé est compromise, ou si vous supposez que c'est le cas, il relève de la responsabilité du Commerçant de renouveler sa clé secrète par l'intermédiaire de Sogenactif Download Extranet.

1.4 Fournisseur de service intermédiaire réalisant des transactions au nom d'un ou plusieurs commerçants

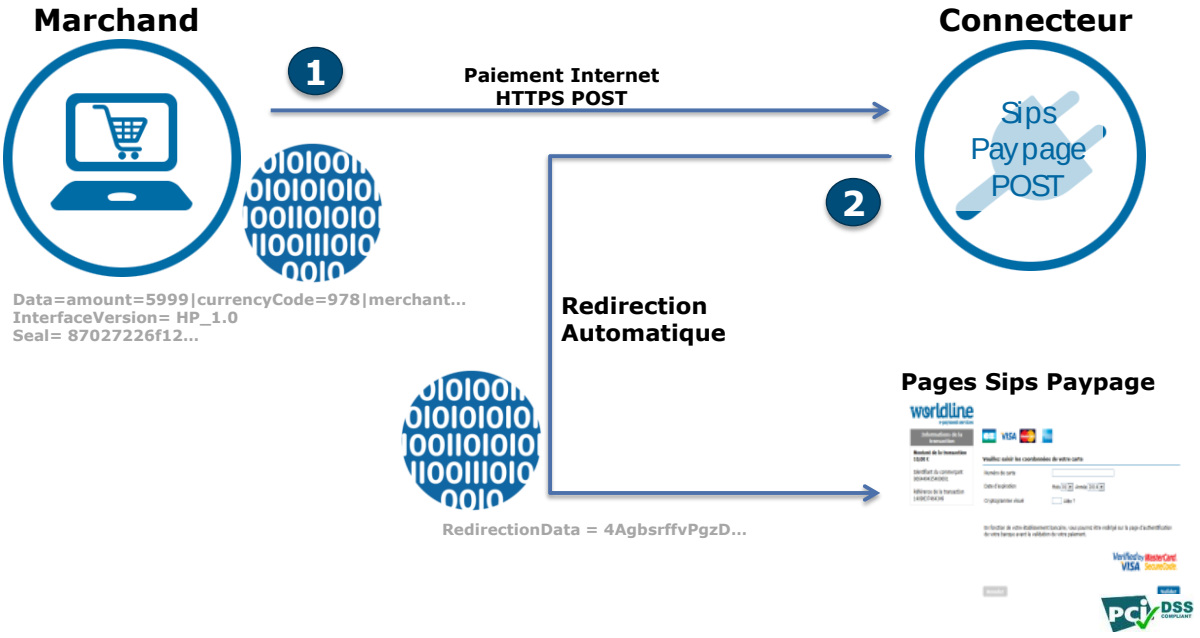
Si vous êtes un prestataire de service réalisant des opérations Sogenactif au nom d'un commerçant, le champ intermediateServiceProviderId doit être rempli avec votre identifiant unique de prestataire de service.

Le sceau (champ Seal) doit-être calculé avec votre propre clé secrète à la place de celle du commerçant que vous représentez.

Merci de vous référer au guide d'installation pour plus d'information sur les prestataires de services. De plus, Les détails du calcul du sceau sont disponibles dans le chapitre "Comment signer un message".

2 Processus de paiement

2.1 Principes généraux

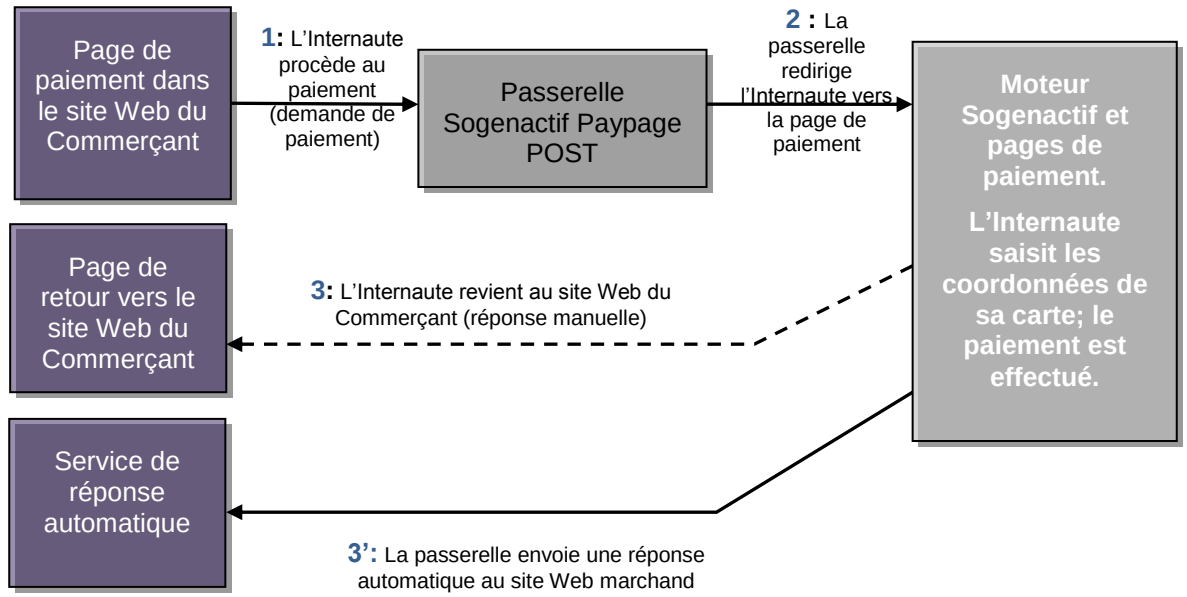


1 : Lorsque l'Internaute confirme le contenu de son panier, il est redirigé vers les serveurs de paiement Sogenactif. Ensuite, la demande de paiement est vérifiée et cryptée, si elle est valide (elle est nommée RedirectionData dans le système).

2 : L'Internaute est redirigé automatiquement vers les pages de paiement Sogenactif avec la demande cryptée. La demande est décryptée et la page de paiement Sogenactif 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 passerelle Sogenactif Paypage POST. SOCIÉTÉ GÉNÉRALE fournit au commerçant l'URL de la passerelle. 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 fonctionnera également.

Étape 2: La passerelle Sogenactif Paypage POST redirigera l'application appelante vers les pages de paiement Sogenactif. L'Internaute doit saisir les détails du moyen de paiement pour que le serveur de paiement Sogenactif 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 : Credit transfer ou PayPal). À la fin du processus de paiement, qu'il soit réussi ou non, deux réponses sont créées et envoyées à l'adresse URL précisée lors du 1er flux.

Il y a deux notifications séparées :

> **Étape 3:** *Les réponses manuelles* sont envoyées sous format HTTP(S) POST par le serveur de paiement à l'URL de la *réponse manuelle*. 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 pour cette raison que l'URL de *réponse manuelle* 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 Sogenactif mais cette fois-ci moyennant l'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 Sogenactif.

✍ **Remarques**

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 Sogenactif 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 Sogenactif Paypage.

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 3.1.3.
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 3.1.3

Un champ d'option supplémentaire est disponible :

Encode	Précise la méthode d'encodage utilisée dans le champ Data, tel que décrit au point 3.1.3
SealAlgorithm	Précise l'algorithme utilisé pour chiffrer le sceau, comme décrit dans la section 3.1.3 (voir les valeurs acceptées dans le dictionnaire des données)

✍ **Remarques**

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 un champ contient une liste d'objets complexes, sa représention 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 requête de paiement :

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

3.1.2 Syntaxe du champ Seal

La valeur du champ Seal est construite comme suit :

Pour l'ahorithm SHA-256 (valeur par défaut) :

- 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 ) )
```

Pour l'algorithme HMAC-SHA :

- Utilisation de la clé secrète partagée pour générer la variante HMAC du message
- Utilisation du champ Data uniquement (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 HMAC-SHA des octets obtenus

Cette procédure peut-être résumé comme suit :

```
HMAC-SHA256( UTF-8(Data), UTF-8(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 Requête de paiement

La demande de paiement est une requête HTTPS POST adressée à la passerelle de paiement. La manière la plus simple d'utiliser cette fonction est d'envoyer un formulaire HTML au moyen de la méthode POST.

4.1.1 CHAMPS PRÉVUS POUR LA requête DE PAIEMENT

Toutes les données impliquées dans la requête de paiement doivent être fournies, comme précisé au chapitre précédent.

La variable **InterfaceVersion** doit être fixée à **HP_2.14**.

Tous les réglages de la requête de paiement, son format et le caractère obligatoire ou facultatif des champs 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.normalreturnurl.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 passerelle Sogenactif Paypage 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.

✍ **Remarques**

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, des messages d'erreur beaucoup plus simples sont affichés sur la plate-forme de production. Ex « Erreur lors du traitement de la requête de paiement. Contactez le Commerçant ».

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'interfaces 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 Sogenactif	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 Réponses 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.

Les réponses de paiements sont des réponses HTTP POST envoyées aux l'URL automaticResponseUrl et normalReturnUrl.

Le commerçant doit mettre en place le système permettant de décoder les réponses, afin de connaitre la finalité du paiement.

Quatre champs sont définis dans les réponses et sont les suivants (sensible à la casse)

Nom des champs	Notes/règles
Data	Concaténation des champs en réponse
Encode	Type d'encodage utilisé
Seal	Signature du message
InterfaceVersion	Version de l'interface du connecteur

Table 1: Champs des réponses automatique et manuelle

SI la valeur du champ Encode est "base64" ou "base64url", le champ Data doit-être décodé en Base64/Base64Url pour retrouver la chaine des champs concaténée.

La chaîne concaténée est structurée comme suit : clé1=valeur1|clé2=valeur2…

Le sceau (champ Seal) des 2 réponses sont cryptés avec le même algorithme utilisé en entrée et fourni dans le champ sealAlgorithm. Si aucune valeur n'a été définie, la valeur SHA-256 est utilisée par défaut.

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 Sogenactif implique l'affichage d'un lien de redirection pour le client. Lorsque l'Internaute clique ce lien, le serveur Sogenactif 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 au point 8.2. 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. Premièrement, 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.

✍ **Remarques**

Les noms de paramètres utilisés dans la réponse manuelle sont sensibles à la casse.

La version actuelle d'**InterfaceVersion** est **HP_2.14**. Veuillez consultez le dictionnaire de données Sips pour une description complète des paramètres inclus dans la réponse.

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 Sogenactif 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 Sogenactif 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 Sogenactif n'attend aucune réponse après la transmission de la réponse automatique.

Comme dans le cas de la réponse manuelle, les champs de la réponse automatique sont décrits au point 8.2. 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.

✍ **Remarques**

Les noms de paramètres utilisés dans la réponse automatique sont sensibles à la casse.

La version actuelle d'**InterfaceVersion** est **HP_2.14**. Veuillez consultez le dictionnaire de données Sogenactif pour une description complète des paramètres inclus dans la réponse.

4.2.3 PROBLÈMES AVEC LA RÉCEPTION DES RÉPONSES SOGENACTIF

Ci-dessous, vous trouverez une liste des problèmes les plus couramment observés qui bloquent la réception des réponses automatiques et manuelles. Assurez-vous de les avoir vérifiés avant d'appeler le service d'assistance technique.

- Vérifiez si les adresses URL de réponse sont fournies dans la demande de paiement et s'ils sont valides. Pour le faire, vous pouvez tout simplement les copier et coller dans votre 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 de votre serveur Web.

- Si vous utilisez un port non standard, il doit être compris entre 80 et 9999 pour assurer la compatibilité avec Sogenactif.
- 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 Sogenactif n'est pas capable de signer le message de réponse. Cela s'applique, par exemple, à l'erreur « MerchantID inconnu » et au cas où la clé secrète est inconnue de Sogenactif.

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 de signer un message

La requête 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 Sogenactif 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.

✍ **Remarques**

Si la clé utilisée pour signer est compromise, ou si vous supposez que c'est le cas, il appartient au Commerçant de demander le renouvellement de sa clé en se connectant à Sogenactif Téléchargement.

5.2 Méthode utilisée 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

Pour Sha256

5.3.1 Php 5

```
<?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();
        }

    }
}
```

Pour Hmac-Sha256

5.3.4 Php 5

```
<?php

…

echo hash mac('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**.

5.3.5 Java

```
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public class ExampleHMACSHA256 {

        /**
         * 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
    {
      Mac hmacSHA256;

      hmacSHA256 = Mac.getInstance("HmacSHA256");
      SecretKeySpec keySpec = new SecretKeySpec(secretKey, "HmacSHA256");
      hmacSHA256.init(keySpec);

      return encodeHexString(hmacSHA256.doFinal(Data));
    }

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

}
```

5.3.6 .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;
            UTF8Encoding utf8 = new UTF8Encoding();
            Byte[] encodedBytes = utf8.GetBytes(sChaine);

            byte[] shaResult;

            HMAC hmac = new HMAC.Create("HMACSHA256");
            var key = "YourSecretKey";
            hmac.Key = utf8.GetBytes(key);
            hmac.Initialize();

            shaResult = hmac.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 peuvent être effectuées à l'aide de l'environnement de démonstration.

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

URL de démo du connecteur	https://payment-webinit.simu.sips-atos.com/paymentInit
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 essais.

✍ Remarques

Puisque l'identifiant Merchant 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 transactionReference soient préfixés par le nom de la future boutique qui sera utilisée dans l'environnement de production.

6.1 Test de transactions par carte

Si vous choisissez VISA, MASTERCARD ou MAESTRO, vous serez redirigé vers la page de saisie des informations de la carte où vous pouvez saisir les données détaillées de votre 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

- Vous pouvez 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 vous utilisez le numéro de carte 4100000000000005, la carte sera identifiée comme VISA et le paiement sera refusé (code de réponse 05).

6.2 Test de transactions iDeal

Si vous choisissez de tester iDeal, vous serez redirigé vers le serveur de simulation qui simule les transactions iDeal selon leur montant. Ensuite, vous retournerez au serveur de paiement qui affiche le ticket avec le résultat de la transaction.

Règles de 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 démarrer en production ?

L'étape suivante est de se connecter à l'environnement de production pour le démarrage réel.

Pour le faire, le Commerçant doit changer l'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 : https://payment-webinit-ws.sogenactif.com

Pour accéder à l'environnement de production, vous aurez besoin d'informations suivantes :

- l'identifiant du commerçant (**merchantID)** qui identifie le site de commerce en ligne sur le serveur de paiement Sogenactif,
- 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 le service d'assistance technique à la fin de l'étape de l'inscription.

Vous pouvez télécharger la version de la clé (**keyVersion)** et de la clé secrète (**secretKey)** depuis l'extranet https://telechargement.sogenactif.com en vous servant du nom de l'utilisateur et du mot de passe fournis par le service d'assistance technique à la fin de l'étape de l'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, jusqu'à ce que les montants soient crédités sur le compte du commerçant et débités du compte de l'acheteur.

Avant d'ouvrir effectivement sa boutique au public, le Commerçant peut envoyer une requête pour valider le paiement de bout en bout, jusqu'à ce que les montants soient crédités sur le compte du Commerçant et débités du compte de l'Acheteur.

8 Description des messages

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	
bypassDcc	Optionnel	HP_2.11	
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	

Nom du champ	Présence	Dans la version	Commentaires
hashAlgorithm2	Optionnel	HP_2.3	
hashSalt1	Optionnel	HP_2.1	
hashSalt2	Optionnel	HP_2.1	
holderAdditionalReference	Optionnel	HP_2.9	
intermediateServiceProviderId	Optional	HP_2.12	
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 certain 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	

Nom du champ	Présence	Dans la version	Commentaires
transactionReference	Optionnel	HP_1.0	
valueDate	Optionnel	HP_2.5	

Tableau 2: requête 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	
fraudData.riskManagementDynamicSettingList	Optionnel	HP_2.10	

Tableau 3: Détails des champs relatifs à la fraude

Données d'entrée RiskManagementDynamicSetting

Champ	Présence	As of version	Comments
riskManagementDynamicSetting. riskManagementDynamicParam	Obligatoire	HP_2.10	
riskManagementDynamicSetting. riskManagementDynamicValue	Obligatoire	HP_2.10	

Table 4: Details des champs RiskManagementDynamicSetting

8.1.3 Champs optionnels relatifs aux pages de paiement

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

Tableau 5: 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 6: 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 7: 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.addrOverride	Optionnel	HP_2.2	
paymentMeanData.paypal.dupCustom	Optionnel	HP_2.2	
paymentMeanData.paypal.dupDesc	Optionnel	HP_2.2	
paymentMeanData.paypal.dupFlag	Optionnel	HP_2.2	

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

Tableau 8: Champs relatifs à PayPal

Pour SDD

Champ	Présence	Dans la version	Commentaires
paymentMeanData.sdd.mandateAuthentMethod	Optionnel	HP_2.2	
paymentMeanData.sdd.mandateCertificationType	Optionnel	HP_2.5	
paymentMeanData.sdd.mandateUsage	Optionnel	HP_2.2	

Tableau 9: Champs relatifs à SDD

Pour Cofinoga3xcb

Champ	Présence	Dans la version	Commentaires
paymentMeanData.cofinoga3xcb.creditIndicator	Optionnel	HP_2.4	

Tableau 10: Champs relatifs à Cofinoga3xcb

Pour PassBe

Champ	Présence	Dans la version	Commentaires
paymentMeanData.passbe.settlementModeList	Optionnel	HP_2.5	

Tableau 11: Champs relatifs à PassBe

For Accord

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

Tableau 12: 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 13: Champs relatifs à Facilypay

For CetelemNxcb

Champ	Présence	Dans la version	Commentaires
paymentMeanData.cetelemNxcb.nxcbTransactionReference1	Optionnel	HP_2.9	
paymentMeanData.cetelemNxcb.nxcbTransactionReference2	Optionnel	HP_2.9	
paymentMeanData.cetelemNxcb.s10NxcbTransactionId1	Optionnel	HP_2.9	
paymentMeanData.cetelemNxcb.s10NxcbTransactionId2	Optionnel	HP_2.9	

Tableau 14: Champs relatifs à CetelemNxcb

Pour Presto

Champ	Présence	Dans la version	Commentaires
paymentMeanData.presto.financialProduct	Obligatoire	HP_2.10	Only mandatory for a presto transaction
paymentMeanData.presto.paymentMeanCustomerId	Obligatoire	HP_2.10	Only mandatory for a presto transaction
paymentMeanData.presto.prestoCardType	Optionnel	HP_2.10	

Table 15: Champs relatifs à Presto

Pour cofidis3x

Champ	Présence	Dans la version	Commentaires
cofidis3x.preScoreValue	Optionnel	HP_2.11	
cofidis3x.cofidisDisplayCancelButton	Optionnel	HP_2.11	
cofidis3x.cofidisPrivateData	Optionnel	HP_2.11	

Table 16: Champs relatifs à cofidis3x

Pour cofidis4x

Champ	Présence	Dans la version	Commentaires
cofidis4x.preScoreValue	Optionnel	HP_2.12	
cofidis4x.cofidisDisplayCancelButton	Optionnel	HP_2.12	
cofidis4x.cofidisPrivateData	Optionnel	HP_2.12	

Table 17: Champs relatifs à cofidis4x

For unEuroCom

Champ	Présence	Dans la version	Commentaires
unEuroCom.preScoreValue	Optionnel	HP_2.11	
unEuroCom.cofidisPrivateData	Optionnel	HP_2.11	

Table 18: Champs relatifs à unEuroCom

8.1.6 Champs optionnels relatifs au 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.s10TransactionIdsList	Optionnel	HP_2.7	
instalmentData.amountsList	Optionnel	HP_2.2	

Tableau 9 : Champs relatifs aux paiements récurrents

8.1.7 Champs optionnels pour les données de facturation

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	
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 19: Champs prévus pour l'élément billingAddress

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 20: Champs prévus pour l'élément billingContact

8.1.8 Champs optionnels pour les données client

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 21: Champs prévus pour l'élément customerAddress

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 22: Champs prévus pour l'élément customerContact

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	
customerData.pwd	Optionnel	HP_2.2	

Tableau 23: Champs prévus pour l'élément customerData

8.1.9 Champs optionnels pour les données de livraison

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 24: Champs prévus pour l'élément deliveryAddress

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 25: Champs prévus pour l'élément deliveryContact

deliveryData inputs

Champ	Présence	Dans la version	Commentaires
deliveryData.deliveryChargeAmount	Optionnel	HP_2.6	
deliveryData.deliveryMethod	Optionnel	HP_2.6	
deliveryData.deliveryMode	Optionnel	HP_2.6	
deliveryData.deliveryOperator	Optionnel	HP_2.6	
deliveryData.estimatedDeliveryDate	Optionnel	HP_2.6	
deliveryData.estimatedDeliveryDelay	Optionnel	HP_2.7	

Table 26: Champs prévus pour l'élément deliveryData

8.1.10 Champs optionnels pour les données Du titulaire de carte

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	

Champ	Présence	Dans la version	Commentaires
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 27: Champs prévus pour l'élément holderAddress

Données d'entrée holderContact

Champ	Présence	Dans la version	Commentaires
holderContact.email	Optionnel	HP_2.2	
holderContact.firstname	Optionnel	HP_2.2	
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 28: Champs prévus pour l'élément holderContact

8.1.11 Champs optionnels pour les données du panier

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	

Champ	Présence	Dans la version	Commentaires
shoppingCartDetail.shoppingCartTotalTaxAmount	Optionnel	HP_2.7	

Tableau 29: Champs prévus pour l'élément shoppingCartDetail

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.productTaxCategory	Optionnel	HP_2.11	
shoppingCartItem.productTaxRate	Optionnel	HP_2.6	
shoppingCartItem.productUnitAmount	Optionnel	HP_2.6	
shoppingCartItem.productUnitTaxAmount	Optionnel	HP_2.6	

Tableau 30: Champs prévus pour l'élément shoppingCartItem

8.1.12 Champs optionnels pour les ID de Transaction Sogenactif 1.0

Données d'entrée s10TransactionReference

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

Tableau 31: Champs prévus pour l'élément s10TransactionReference

8.1.13 Champs iotnnels pour les données de gestion du risque

Données d'entrée riskManagementCustomData

Champ	Présence	Dans la version	Commentaires
riskManagementCustomData. riskManagementCustomSequence	Optionnel	HP _2.9	
riskManagementCustomData. riskManagementCustomValue	Optionnel	HP _2.9	

Tableau 32: Champs prévus pour l'élément riskManagementCustomData

8.2 Réponses (automatiques et manuelles)

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

Champ	Dans la version	Commentaires
acquirerNativeResponseCode	HP_2.12	
acquirerResponseCode	HP_2.0	
acquirerResponseIdentifier	HP_2.8	
acquirerResponseMessage	HP_2.8	
additionalAuthorisationNumber	HP_2.8	
amount	HP_1.0	Valeur véhiculée dans la requête de paiement.
authorisationId	HP_1.0	Valeur véhiculée dans la requête de paiement.
captureDay	HP_1.0	Valeur véhiculée dans la requête de paiement.
captureLimiteDate	HP_2.3	
captureMode	HP_1.0	Valeur véhiculée dans la requête de paiement.
cardCSCResultCode	HP_2.0	
cardProductCode	HP_2.12	
cardProductName	HP_2.12	
cardProductProfile	HP_2.12	

Champ	Dans la version	Commentaires
complementaryCode*	HP_1.0	
complementaryInfo*	HP_2.0	
creditorId	HP_2.7	
currencyCode	HP_1.0	Valeur véhiculée dans la requête de paiement.
customerEmail	HP_2.0	Valeur véhiculée dans la requête de paiement. Seulement disponible en HP_2.0
customerId	HP_2.0	Valeur véhiculée dans la requête de paiement.
customerIpAddress	HP_2.0	Valeur véhiculée dans la requête de paiement.
customerMobilePhone	HP_2.1	Valeur véhiculée dans la requête de paiement. Seulement disponible en HP_2.1
dccAmount	HP_2.3	
dccCurrencyCode	HP_2.3	
dccExchangeRate	HP_2.3	
dccExchangeRateValidity	HP_2.3	
dccProvider	HP_2.3	
dccStatus	HP_2.3	
dccResponseCode	HP_2.3	
dueDate	HP_2.3	
guaranteeIndicator	HP_2.0	
hashPan1	HP_2.0	
hashPan2	HP_2.0	
holderAuthentMethod*	HP_2.4	
holderAuthentProgram	HP_2.5	
holderAuthentRelegation*	HP_2.0	
holderAuthentStatus*	HP_2.0	

Champ	Dans la version	Commentaires
instalmentAmountsList	HP_2.6	
instalmentDatesList	HP_2.6	
instalmentNumber	HP_2.6	
instalmentTransactionReferencesList	HP_2.6	
interfaceVersion*	HP_1.0	
invoiceReference	HP_2.10	
issuerCode	HP_2.12	
issuerCountryCode	HP_2.12	
issuerEnrollementIndicator*	HP_2.0	
issuerWalletInformation	HP_2.9	
keyVersion	HP_1.0	Valeur véhiculée dans la requête de paiement.
mandateAuthentMethod	HP_2.2	
mandateCertificationType	HP_2.7	
mandateId	HP_2.3	
mandateUsage	HP_2.2	
maskedPan*	HP_1.0	
merchantId	HP_1.0	Valeur véhiculée dans la requête de paiement.
merchantSessionId	HP_2.0	Valeur véhiculée dans la requête de paiement.
merchantTransactionDateTime	HP_2.0	Valeur véhiculée dans la requête de paiement.
merchantWalletID	HP_2.0	Valeur véhiculée dans la requête de paiement.
orderChannel	HP_2.0	Valeur véhiculée dans la requête de paiement.
orderId	HP_1.0	Valeur véhiculée dans la requête de paiement.

Champ	Dans la version	Commentaires
panEntryMode*	HP_2.4	
panExpiryDate*	HP_2.0	
paymentMeanBrand*	HP_1.0	
paymentMeanBrandSelectionStatus	HP_2.14	
paymentMeanData*	HP_2.2	
paymentMeanId	HP_2.6	
paymentMeanTradingName	HP_2.8	
paymentMeanType*	HP_1.0	
paymentPattern	HP_2.0	Valeur véhiculée dans la requête de paiement.
preAuthenticationColor	HP_2.10	
preAuthenticationInfo	HP_2.10	
preAuthenticationProfile	HP_2.10	
preAuthenticationProfileValue	HP_2.14	
preAuthenticationRuleResultList	HP_2.14	Une liste d'objet preAuthenticationRuleResult. Veuillez consulter 8.2.2 pour son contenu et 8.2.3 pour son format
preAuthenticationThreshold	HP_2.10	
preAuthenticationValue	HP_2.10	
preAuthorisationProfile	HP_2.14	
preAuthorisationProfileValue	HP_2.14	
preAuthorisationRuleResultList	HP_2.14	Une liste d'objet preAuthorisationRuleResult. Veuillez consulter 8.2.1 pour son contenu et 8.2.3 pour son format
responseCode	HP_1.0	
returnContext	HP_1.0	Valeur véhiculée dans la requête de paiement.
s10TransactionId	HP_2.9	

Champ	Dans la version	Commentaires
s10TransactionIdDate	HP_2.9	
s10transactionIdsList	HP_2.11	
scoreColor*	HP_2.0	
scoreInfo*	HP_2.0	
scoreProfile*	HP_2.0	
scoreThreshold*	HP_2.0	
scoreValue*	HP_2.0	
settlementMode	HP_2.7	
settlementModeComplement	HP_2.13	
statementReference*	HP_2.4	
tokenPan*	HP_2.0	
transactionActors	HP_2.2	Valeur véhiculée dans la requête de paiement.
transactionDateTime	HP_1.0	
transactionOrigin	HP_2.0	Valeur véhiculée dans la requête de paiement.
transactionReference	HP_1.0	Valeur véhiculée dans la requête de paiement.
walletType	HP_2.4	

Tableau 33: Champs prévus pour la réponse automatique/manuelle au paiement

***: CHAMPS RENSEIGNES S'ILS SONT DISPONIBLES, EN FONCTION DE L'ETAT DE LA TRANSACTION ET DU MOYEN DE PAIEMENT CHOISI.**

8.2.1 Données de PreAuthorisationRuleResult

Field	As of version	Comments
ruleCode	WS_2.14	
ruleType	WS_2.14	
ruleWeight	WS_2.14	
ruleSetting	WS_2.14	
ruleResultIndicator	WS_2.14	
ruleDetailedInfo	WS_2.14	

Table 34: Champs prévus pour l'élément PreAuthorisationRuleResult

8.2.2 Données de PreAuthenticationRuleResult

Field	As of version	Comments
ruleCode	WS_2.14	
ruleType	WS_2.14	
ruleWeight	WS_2.14	
ruleSetting	WS_2.14	
ruleResultIndicator	WS_2.14	
ruleDetailedInfo	WS_2.14	

Table 35: Champs prévus pour l'élément PreAuthenticationRuleResult

8.2.3 Format des Listes Complexes

Le format d'une liste d'objets complexes dans les réponses automatiques et manuelles est définie comme suit: (surligné en jaune)

amount=1000|currencyCode=978|objectNameList=[{"field1":"value1a",
"field2":"value2a","field3":"value3a"…},{"field1":"value1b",
"field2":"value2b","field3":"value3b"}…]|transactionReference=1452687287828

Le contenu de la liste sont enveloppés dans une paire de crochets [ ].

Chaque entrée de la liste est enveloppé dans une paire d'accolades { }.

Au sein de chaque entrée de la liste, chaque champ est représenté comme "nom_du_champ" = "valeur_du_champ". Notez que le nom et la valeur du champ sont tous deux enveloppés dans une paire de doubles guillemets "". Les paires de nom / valeur adjacents sont séparés par une virgule , .

Example :

amount=1000|currencyCode=978|preAuthenticationRuleResultList=[{“ruleDetailedInfo”:"SHIP_ZIP=A;BILL_ZIP=B",”ruleCode”:"A",”ruleType”:"BBB"},{“ruleDetailedInfo”:"SHIP_ZIP=C;BILL_ZIP=D",”ruleCode”:"A",”ruleType”:"BBB"}]|transactionReference=1452687287828

FIN DU DOCUMENT