Systempay
Solutions d'avance

Définition des Webservices Systempay

Version 1.2a

Rédaction, Vérification, Approbation

Rédaction		Vérification		Approbation	
Nom	**Date/Visa**	**Nom**	**Date/Visa**	**Nom**	**Date/Visa**
Lyra-Network	03/05/2012	Lyra-Network	03/05/2012	Lyra-Network	03/05/2012

Historique du document

Version	Auteur	Date	Commentaires
1.2a	Lyra-Network	03/05/2012	▪ Ajout de la méthode sendVEReqAndbuildPAReqByIdentifierTx ▪ Suppression des méthodes : ▪ buildPAReqByOrderIdentifierTx ▪ sendVEReqByOrderIdentifierTx ▪ sendVEReqTx ▪ buildPAReqTx
1.2	Lyra-Network	14/12/2011	▪ Ajout de nouveaux codes d'erreur ▪ Ajout de la méthode buildPAReqByOrderIdentifierTx ▪ Modification apportées sur le maintien de la session HTTP en mode TEST
1.1	Lyra-Network	20/06/2011	▪ Ajout des méthodes : ▪ sendVEReqAndbuildPAReqTx ▪ sendVEReqByOrderIdentifierTx ▪ Précisions ajoutées concernant le maintien de la session HTTP
1.0	Lyra-Network	20/06/2011	Version initiale.

Systempay
Solutions d'avance

SOMMAIRE

1. Présentation

Ce document présente les webservices 3-D Secure qui permettent de réaliser des vérifications sur l'enrôlement d'une carte et de rediriger le porteur vers l'ACS de la banque émettrice.
Les résultats des requêtes présentées dans ce document serviront à renseigner le paramètre **ThreeDSResult** lors d'une requête de paiement.
(cf Guide_d_implementation_standard_WebService.pdf)

Ces webservices ont été développés suivant le protocole SOAP (Simple Object Access Protocol) et sont décrits par le fichier wsdl suivant :

https://paiement.systempay.fr/vads-ws/threeds-v1?wsdl

Afin de sécuriser les échanges, les webservices (SOAP) sont cryptés grâce au protocole HTTPS. De plus un mécanisme de signature a été mis en place afin de valider et d'authentifier l'échange des données. (cf. 4)

2. Principe de fonctionnement

Pour obtenir une authentification 3-D Secure, vous devez suivre la cinématique suivante :

- demande de vérification d'enrôlement,
- sauvegarde du champ **requestId**,
- vérification de l'état d'enrôlement
  - Si la carte est enrôlée, on continue en 3-D Secure
  - Sinon : fin de la procédure 3-D S, on procède au paiement en mode SSL.
- construction du message **PaReq**,
- redirection du navigateur client vers l'ACS,
- récupération des paramètres POST '**MD**' et '**Pares**' sur la page de retour,
- analyse du **PaRes :**
  - Si l'authentification est correcte, on procède au paiement en précisant le résultat de l'authentification dans le champ '**threeDsResult**'
  - Sinon, on rejette le paiement.

3. Description des types

3.1. ThreeDSecureResponse

Ce type permet de décrire la réponse de l'ensemble des webservices 3DS.
Elle sera complétée par un type de réponse spécifique à chaque requête.

Nom du champ	Type	Description
errorCode	int	Code d'erreur
errorDetail	String	Détail de l'erreur si le champ errorCode est différent de 0
timestamp	long	Timestamp permettant la génération de signature unique
signature	String	Signature de la Réponse (cf. ci-dessous)

La signature permet de valider l'intégrité de la réponse, le calcul de cette signature se fait en prenant les paramètres dans l'ordre suivant :

errorCode, errorDetail, timestamp

3.2. PaResInfo

Il s'agit de la réponse renvoyée par le serveur lors d'une requête **analysePaResTx**.
Elle contient le résultat de l'authentification 3-D Secure

Nom du champ	Type	Description
brand	String	Brand de la carte (« VISA » ou « MASTERCARD »)
enrolled	String	Statut enrôlement porteur : • « Y » : Enrôlé • « N » : Non enrôlé • « U » : Inconnu
status	String	Statut authentification du porteur : • « Y » : Authentifié 3DS • « N » : Erreur Authentification • « U » : Authentification impossible • « A » : Essai d'authentification
eci	String	ECI
xid	String	Numéro de transaction marchand
cavv	String	Certificat de l'ACS
cavvAlgorithm	String	Algorithme CAVV : • « 0 » : HMAC • « 1 » : CVV • « 2 » : CVV_ATN • « 3 » : Mastercard SPA

La signature permet de valider l'intégrité de la réponse, le calcul de cette signature se fait en prenant les paramètres dans l'ordre suivant :

errorCode, errorDetail, timestamp, brand, enrolled, status, eci, xid, cavv, cavvAlgorithm

3.3. VEResPAReqInfo

Réponse renvoyée lors d'une requête **sendVEReqAndbuildPAReqTx**. Elle contient le résultat de la demande d'enrôlement ainsi que le message encodé qui sera transmis par le navigateur client à l'ACS.

Nom du champ	Type	Description
requestId	String	numéro de requête, à rappeler dans les appels analyzePAResTx
enrolled	String	Statut d'enrôlement du porteur : • « Y » : Enrôlé • « N » : Non enrôlé • « U » : Inconnu
acsUrl	String	Url de l'ACS à contacter, présent uniquement si enrolled=Y
acctId	String	certificat renvoyé par le Directory Server
encodedPareq	String	message PAReq encodé, prêt à envoyer à l'ACS
brand	String	Réseau de carte

La signature permet de valider l'intégrité de la réponse, le calcul de cette signature se fait en prenant les paramètres dans l'ordre suivant :

errorCode, errorDetail, timestamp, requestId, enrolled, acsUrl, acctId, encodedPareq

3.4. Description des codes erreur

errorCode	Description
0	Action réalisée avec succès
1	Action non autorisée
2	Mauvaise signature
3	Aucune brand localisée
4	Erreur lors de la détermination de la plage de la carte
5	Aucun contrat adéquat trouvé
6	Spécification du contrat ambigüe, plusieurs sont disponibles
7	Marchand non enrôlé
8	Signature de l'ACS invalide
9	Erreur technique
10	Mauvais paramètres
11	Format de date incorrect
12	3D Secure désactivé
13	Identifiant non trouvé
14	PAN non trouvé
98	Erreur de traitement sur l'ACS
99	Erreur inconnue

4. Signature

Un certificat est nécessaire pour dialoguer avec la plateforme de paiement. Il est mis à disposition de toutes les personnes habilitées à la consultation des certificats dans votre outil de gestion de caisse à l'emplacement suivant : Paramètres / Boutique / Certificat. Il existe deux certificats différents : un pour la plateforme de test et un pour la plateforme de production.

La signature sera générée comme suit :

- Création d'une chaîne de caractère représentant la concaténation des paramètres, séparés par le caractère "+".
- Ajout à cette chaîne d'un "certificat " numérique (de test ou de production selon le contexte).
- Hachage de la chaîne résultante avec l'algorithme SHA1.

La plateforme de paiement effectuera obligatoirement la vérification de la signature. Il est de la responsabilité du commerçant de vérifier à son tour la signature transmise en retour.

L'ordre des champs doit être respecté.

Les champs de type numérique ne doivent pas avoir de 0 à gauche du digit le plus significatif.

Les champs de type bool prennent les valeurs suivantes :

- 1 pour vrai (true)
- 0 pour faux (false)

Les champs de type String non renseignés seront vides.

En mode TEST, en cas de mauvais calcul de signature, le code erreur renvoyé est « BAD_SIGNATURE » et, la chaine de caractère utilisée pour la signature côté serveur est alors renvoyée dans le champ **errorDetail**.

5. Description des méthodes et cinématique

Les WebServices 3D-S décrits dans ce document doivent respecter la cinématique suivante :

- Demande de vérification d'enrôlement (VEReq)
- Construction du message codé (PAReq)
- Soumission d'un formulaire auto posté contenant le PAReq
- Analyse de la réponse renvoyée par l'ACS (PARes)

Important :

L'architecture de la plateforme de paiement reposant sur un ensemble de serveurs avec répartition de charge, il est nécessaire que chaque requête associée à un même paiement soit réalisée avec la même session HTTP afin d'assurer la continuité du processus.

Pour cela, à chaque requête sendVEReqAndbuildPAReqTx, une session est créée coté serveur.
L'ID de la session est renvoyé dans l'entête HTTP de la réponse. Il devra être retourné dans les requêtes analysePAResTx suivantes.

Note concernant le mode TEST :

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

Ceci devra se faire en concaténant :

- L'url de l'ACS obtenue dans la réponse VEResPAReqInfo
- L'identifiant de session renvoyé dans l'entête http, séparés par « **;jsessionid=** »

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

5.1. Maintien de la session HTTP entre chaque requête

- En java

Une fois le client WebService créé (port), utiliser la méthode signalée en gras dans le code suivant :

```
Service service = Service.create(wsdlURL, qname);
ThreeDSecure port = service.getPort(ThreeDSecure.class);
((BindingProvider)
port).getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
```

Cela permet au serveur de ne pas ignorer les infos de session associées à la requête http et de maintenir un cookie avec l'ID de session.

- En PHP

Après l'appel à la fonction sendVEReqAndBuildPAReqTx, la session est créée côté serveur et renvoyée dans les headers HTTP de la réponse. Pour les méthodes suivantes de la même transaction, il faut récupérer ce header et le transmettre en tant que cookie dans la requête HTTP.
Voici un exemple de code pour récupérer l'id de session et le transmettre :

```
/* La méthode ci-dessous permet de récupérer l'entête HTTP de la réponse */
$header = $client->__getLastResponseHeaders();

/* Dans la chaîne de caractère obtenue, nous recherchons la présence de l'ID de la
session HTTP, stockée dans l'élément "JSESSIONID" : */

if(!preg_match("#JSESSIONID=([A-Za-z0-9\.]+)#",$header, $matches)){
      return "Aucun ID de Session Renvoyé.";//Cas d'erreur technique;
}
$cookie = $matches[1];

/*La méthode ci-dessous permet de spécifier un cookie qui sera envoyé dans chaque
entête http */
$client->__setCookie ("JSESSIONID", $cookie);
```

Il est donc nécessaire de stocker cet id de session car lors du retour de l'ACS, dé corrélé des requêtes sendVEReqAndbuildPAReqTx, il faut également l'envoyer (cf. 5.5).

5.2. sendVEReqAndbuildPAReqTx

Cette fonction permet de faire une demande de vérification d'enrôlement auprès des Directory Servers VISA ou MasterCard et de générer le message envoyé par le navigateur du client lors de la requête d'authentification du payeur (PAReq) vers l'ACS.
Cette fonction prend en entrée les paramètres suivants :

Nom du champ	Type	Description	Obligatoire
shopId	String	Identifiant de la boutique	✓
contractNumber	String	Numéro de contrat commerçant	✓
ctxMode	String	Contexte de sollicitation de la plateforme de paiement ("TEST", "PRODUCTION")	✓
cardNumber	String	Numéro de carte du porteur	✓
browserUserAgent	String	Header « User-Agent » du navigateur du client	
browserAccept	String	Header « Accept » du navigateur du client	
purchaseAmount	String	Montant de la transaction en plus petite unité monétaire	✓
purchaseCurrency	String	Devise (Code monnaie ISO 4217, Euro : 978)	✓
cardExpiry	String	Date d'expiration de la carte, format YYMM	✓
wsSignature	String	Signature (cf ci-dessous)	✓

Le calcul de la signature se fait en prenant les paramètres dans l'ordre suivant :
shopId, , contractNumber, ctxMode, cardNumber, browserUserAgent, browserAccept, purchaseAmount, purchaseCurrency, cardExpiry

Cette fonction retourne une réponse du type **VEResPAReqInfo** contenant le résultat de la demande d'enrôlement ainsi que le message PAReq codé dans le cas où le champ **enrolled** est valorisé à « **Y** »

Si le champ **enrolled** est valorisé à 'Y' alors la cinématique d'authentification 3-D Secure peut continuer.

La transaction pourra continuer en mode SSL dans les cas suivants :

- L'errorCode est à 0 et le champ **enrolled** est différent de Y : dans ce cas le paramètre **threeDsResult** de la requête create (ws standard) prendra les valeurs suivantes :

Nom du champ	Type	Valeur
brand	String	Ne pas envoyer
enrolled	String	Statut d'enrôlement du porteur : • "N" : Non enrôlé ou • "U" : Inconnu
authStatus	String	Ne pas envoyer
eci	String	Ne pas envoyer
xid	String	Ne pas envoyer
cavv	String	Ne pas envoyer
cavvAlgorithm	String	Ne pas envoyer

- L'errorCode vaut '7', '9' ou '99' : dans ce cas le paramètre **threeDsResult** ne doit pas être envoyé dans la requête create.

5.3. sendVEReqAndbuildPAReqByIdentifierTx

Cette fonction permet de faire une demande de vérification d'enrôlement auprès des Directory Servers VISA ou MasterCard en passant en paramètre l'identifiant d'un compte carte et de générer le message envoyé par le navigateur du client lors de la requête d'authentification du payeur (PAReq) vers l'ACS.
Cette fonction prend en entrée les paramètres suivants :

Nom du champ	Type	Description	Obligatoire
shopId	String	Identifiant de la boutique	✓
contractNumber	String	Numéro de contrat commerçant	✓
ctxMode	String	Contexte de sollicitation de la plateforme de paiement ("TEST", "PRODUCTION")	✓
identifier	String	Identifiant du compte carte	✓
browserUserAgent	String	Header « User-Agent » du navigateur du client	✓
browserAccept	String	Header « Accept » du navigateur du client	✓
purchaseAmount	String	Montant de la transaction en plus petite unité monétaire	✓
purchaseCurrency	String	Devise (Code monnaie ISO 4217, Euro : 978)	✓
wsSignature	String	Signature (cf ci-dessous)	✓

Le calcul de la signature se fait en prenant les paramètres dans l'ordre suivant :
shopId, contractNumber, ctxMode, orderId, browserUserAgent, browserAccept, purchaseAmount, purchaseCurrency

Cette fonction retourne une réponse du type **VEResPAReqInfo** contenant le résultat de la demande d'enrôlement ainsi que le message PAReq codé dans le cas où le champ **enrolled** est valorisé à « **Y** ».

5.4. Redirection vers l'ACS

Une fois l'encodedPareq récupéré, il faut rediriger le navigateur du client vers son ACS, en renvoyant une page HTML avec un formulaire POST auto soumis.

L'url de l'ACS est utilisée comme action du POST, la valeur est celle retournée par l'appel de la méthode sendVEReqAndBuildPaReqTx (acsUrl).

Il faut également disposer d'une url de retour sur le serveur pour récupérer la réponse de l'ACS (elle aussi par POST).

Ce formulaire doit obligatoirement contenir les champs suivants :

Nom du champ	Type	Description	Obligatoire
PaReq	String	Message codé, retourné par l'appel de la méthode sendVEReqAndBuildPAReqTx	✓
TermUrl	String	URL de retour dans laquelle sera analysé le retour de l'authentification 3-D Secure	✓
MD	String	'Merchant DATA', par commodité il est conseillé de valoriser ce champ avec le requestId afin de reprendre facilement le traitement	✓

Note concernant le mode TEST :

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

Ceci devra se faire en concaténant :

- L'url de l'ACS obtenue dans la réponse VEResPAReqInfo
- L'identifiant de session renvoyé dans l'entête http, séparés par « **;jsessionid=** »

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

Exemple :

*<form name="Form" method="post" action="https://paiement.systempay.fr/vads-payment/acs.silent_authenticate.a**;jsessionid=B420BF68835F6563FB6E4B289ABB9080.bdxvad3**" >*
...

</form>

Exemple de page de redirection

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8"/>
<title>---</title>
<script type="text/javascript">
<!--
    function submitForm(){
        document.redirectForm.submit();
    }
 -->
</script>
</head>

<body onLoad="setTimeout(\'submitForm()'.'\',500);">
<span class="message">redirection ACS</span>
<br/>
<br/>
<br/>
<form name="redirectForm" action="acsUrl" method="POST">
            <input type="hidden" name="PaReq" value="encodedPareq"/>
            <input type="hidden" name="TermUrl" value="url_de_retour"/>
            <input type="hidden" name="MD" value="requestId" />
            <noscript><input type="submit" name="Go" value="Click to continue"/></noscript>
</form>
</body>
</html>
```

5.5. Récupération de la réponse de l'ACS

Il est nécessaire de mettre en place une URL de retour de l'ACS pour que celui-ci puisse nous renvoyer les données du PARes.
Les paramètres renvoyés par l'ACS sont les suivants :

- PaRes : contient le message PARes
- MD : contient le Merchant Data envoyé lors de l'appel à l'ACS

Il convient alors d'extraire du champ **MD** les valeurs de l'id de session et du **requestId** pour ensuite les utiliser lors de l'appel à la méthode **analyzePAResTx**.

Exemple de page de retour :

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

```
<?php
   session_start();
?>
<html>
<head></head>
<body>
<?php
$PaRes = $_POST['PaRes'];
list($JSESSIONID, $requestId) = explode("+",$_POST['MD']);

//Initialisation du client SOAP
$client = new soapclient($wsdl,array('trace' =>1));

//Définition du cookie qui sera envoyé avec la requête SOAP
$client-> __setCookie('JSESSIONID', $JSESSIONID);


Appel de la méthode analysePAResTx
...


</body>
</html>
```

5.6. analyzePAResTx

Cette fonction permet de soumettre à la plateforme de paiement le message PaRes reçu après l'authentification 3D-S.
Cette fonction prend en entrée les paramètres suivants :

Nom du champ	Type	Description	Obligatoire
shopId	String	Identifiant de la boutique	✓
contractNumber	String	Numéro de contrat commerçant	✓
ctxMode	String	Contexte de sollicitation de la plateforme de paiement ("TEST", "PRODUCTION")	✓
requestId	String	numéro de requête	✓
pares	String	Message PaRes encodé, reçu de l'ACS	✓
wsSignature	String	Signature (cf ci-dessous)	✓

Le calcul de la signature se fait en prenant les paramètres dans l'ordre suivant :
shopId, contractNumber, ctxMode, requestId, pares

Cette fonction retourne une réponse du type **PAResInfo**. Elle contient le message PAReq codé.
Cette réponse est ensuite utilisée pour renseigner le champ **threeDsResult** lors d'une requête de paiement :

Si le champ '**status**' est valorisé à :

- 'N' ou 'U' : Rejet de la transaction : l'internaute ne s'est pas authentifié correctement
- 'Y' ou 'A' : L'authentification est correcte, la demande de paiement peut être effectuée en transmettant le résultat de l'authentification dans le champ **threeDsResult**.

Remarque :

Le message PARes peut comporter des caractères de retour à la ligne ('CR', 'LF' ou '\r','\n').
Ces caractères sont remplacés par un simple LF par certains framework au moment de l'appel au WS. C'est le cas notamment en ASP.NET.
Afin de ne pas rencontrer des problèmes de calcul de signature, il est conseillé de supprimer les retours à la ligne avant le calcul de signature.

Cette suppression n'altère pas l'intégrité du message PARes.