Architecture du module as\_systempay  
pour la mise en œuvre de SystemPay

Les deux logiciels Galette et Joomla s’interfacent avec le module commun SystemPay, le but est de n’avoir qu’un seul code à générer et donc à maintenir, pour faire les paiements en ligne et aussi de minimiser les développements spécifiques à SystemPay dans nos programmes. Il est toujours préférable de maximiser les indépendances des logiciels.

C’est aussi une facilité, et permettra de diviser les tests par 2, que Joomla et Galette se concentre sur les IHM, plutôt que sur les fonctions de paiement, difficiles à mettre au point.

# Retour de notification,

C’est sur cette adresse, que nous savons que l’opération s’est bien déroulée. L’adresse de ce retour doit être paramétrée dans la transaction d’appel vers SystemPay. Le code exécuté lors de ce retour doit appeler le code du module commun (création de l’objet et appel de la fonction).

Lors d’un retour de notification, Galette ou Joomla indiqueront à Payé l’abonnement ou le paiement ayant fait l’objet de la transaction.

Le module commun enregistre les principaux paramètres de la transaction. Il appelle également le module Compta qui va générer les lignes de comptabilité dans la table dédiée à cet effet. Les paramètres étant déjà enregistrés sur le back end, il ne faut conserver que l’essentiel, cela évitera de devoir nettoyer les bases périodiquement, ni même de devoir conserver les N° de CB.

La table est créée sous MySQL. Le mécanisme d’insertion des transactions dans la table est fonctionnel.

# Retour vers la boutique,

L’adresse de ce retour doit être paramétrée dans la transaction d’appel vers SystemPay. Le module commun ne fait aucune opération sur un retour boutique.

# Partie comptabilité,

Joomla et Galette envoient leurs informations au module commun, et c’est celui-ci qui les traite et les met en BdD. Toujours dans un souci de communaliser le code et de n’avoir qu’un seul code à gérer. Les infos à passer sont rudimentaires, il s’agit de :

* La date de la transaction (date seule sans heure, déjà fourni par SystemPay)
* Montant total (déjà fourni par SystemPay)
* Tiers ( ID du USER)
* Catégories (codé en format texte : code compta + montant ventilé, autant de fois que d’activités Galette)
* Eventuellement une ligne commentaire

Elles sont passées dans le champ vads\_order\_info.

vads\_order\_info = ga|756\_BF\_45/756\_AS\_24|Ligne de commentaire

ga indique que Galette est émetteur de la transaction (sinon jo …)

Le caractère « | » s épare les données Emetteur, Ventilation et Commentaire

La partie Ventilation contient la ventilation du paiement 756\_BF\_45/756\_AS\_24

Chaque caractère « / » sépare une partie de la ventilation et chaque partie est séparée par « \_ ». Ces parties sont composées de 2 caractères indiquant le code de la section, le code comptable et le montant de la ventilation.

# Appel du module commun pour passer un ordre de transaction

Le mécanisme est le même que pour SystemPay, il faut générer une Form contenant les champs vads\_ et les passer en mode POST (voir fichier index.html). Il vaudrait mieux que cette form contienne également le champ signature.

# Retour de notification vers Galette

Le module commun envoie en mode POST (à analyser par $\_REQUEST) les informations de paiement effectué à Galette (comme Systempay le fait)

Joomla

Galette

Module commun SystemPay

SystemPay

Module commun SystemPay

# Liste des fichiers du module as\_systempay

## Dans le répertoire log

### sp\_log.txt

* Contient les Log de ces scripts
* Ce fichier doit être effacé manuellement

## Dans le répertoire configuration

Ces fichiers sont dans un répertoire spécifique car ils ne doivent pas être déposés sur des systèmes de gestion de configuration accessible en mode public.

### sp\_configuration.txt

* Contient la configuration propre à SystemPay, cette configuration est appelée par as\_systempay, mais certains paramètres de configuration peuvent être surchargés lors de l’envoi de la transaction.
* Ce fichier doit être modifié pour passer de TEST à PRODUCTION

### sp\_db\_config.php

* Contient les constantes permettant les accès à la base de données
* ET les constantes de débogage

## Dans le répertoire courant

### sp\_include.php

* Contient tous les « define » de ce script. Chaque numéro de define est lié aux ID des messages de la table tsp\_systempay\_msg. **Attention à bien maintenir les valeurs des define en fonction des ID de la table !**

### index.html

* formulaire HTML de saisie des informations Utilisateurs de paiement, appelle ensuite le formulaire de paiement. Ce formulaire n’est à utiliser que pour les tests.

### sp\_outils.php

* Contient les fonctions statiques, de débogage, envoi de mail, de formatage de dates, hors fonctions liées à systempay

### sp\_form\_paiement.php

* Génère une forme contenant toutes les informations à destination de SystemPay
* Envoie la forme

### sp\_retour\_notification.php

* Cette page est appelée par le code développé sous Joomla, sous Galette ou sous tout autre fichier php. Elle est appelée le système de notification de Systempay en fin de paiement.
* Analyse les données $\_REQUEST et vérifie la validité de la transaction renvoyée par SuytemPay.
* Ecrit en BdD un enregistrement avec les principaux paramètres fournis par SystemPay.

### sp\_lire\_msg\_bd.php

* Lit les messages fournis par la documentation SystemPay en français ou en anglais dans la BdD

### sp\_paiement.php

* Lit et interprète toutes les informations d'un $\_REQUEST de retour de SystemPay et les place dans l'objet

### sp\_compta.php

* Interprète le champ « vads\_order\_info » du $\_REQUEST de retour de notification SystemPay et les génère les lignes de comptabilité correspondantes dans la table tb\_systempay\_oper.

### sp\_notif\_joomla.php

* Ce fichier est donné à titre d’exemple de ce qui doit être fait par Joomla ou Galette, il doit alors être copié dans le répertoire Joomla ou appelé avec des « include » pointant sur ce répertoire.
* Les « include » de ce fichier désignent le chemin de as\_systempay de manière absolue et non relative.
* Il appelle TraiterNotification, cette procédure analyse les paramètres de retour de SystemPay et enregistre la transaction en base de données et en compta. Cette procédure renvoie true si la signature est valide

### sp\_retour\_boutique.php

* Ce fichier est donné à titre d’exemple, il peut être appelé lors du retour boutique (voir fichier de configuration ou configuration du BackOffice ou paramètre de transaction. En règle générale, Galette ou Joomla auront leurs propres fichiers de retour boutique.

## Dans le répertoire sql

### sp\_Vue\_Oper.sql

* Contient le code SQL pour la création de la vue
* Ainsi que pour les tables sql tb\_systempay\_msg (ensemble des messages fournis par la documentation de SystemPay) et tb\_systempay\_oper (ensemble des transactions effectuées)

## Dans le répertoire compteur

### sp\_compteur.txt

* Chaque transaction doit avoir un identifiant unique. Le numéro de l’identifiant est stocké dans ce fichier, et est incrémenté à chaque fois qu’un utilisateur effectue un paiement. Il s’agit d’un entier.

## Dans le répertoire documentation

### sp\_architecture.docx

* Ce fichier.