

Kit d'intégration JAVA

Payline

Table des matières

| | | |
|----------|---|----------|
| 1 | Introduction..... | 3 |
| 1.1 | Objet du document..... | 3 |
| 1.2 | Public visé..... | 3 |
| 1.3 | Contenu du document | 3 |
| 1.4 | Liste des documents de référence | 3 |
| 1.5 | Avertissement..... | 4 |
| 1.6 | Contacts | 4 |
| 1.7 | Historique des révisions..... | 4 |
| 2 | Présentation du kit JAVA..... | 4 |
| 2.1 | Composition du kit | 4 |
| 2.2 | Structure et fonctions | 5 |
| 2.2.1 | Une application Web « webPayline.war »..... | 5 |
| 2.2.2 | La librairie « kitPayline.jar »..... | 5 |
| 2.2.3 | Le fichier de paramétrage « payline.properties »..... | 5 |
| 2.3 | Configuration du kit | 6 |
| 2.4 | Compatibilité du kit | 7 |
| 2.5 | Modification de l'application Web du client | 7 |
| 2.6 | Exemples..... | 8 |
| 2.6.1 | doAuthorization..... | 8 |

1 Introduction

1.1 Objet du document

Ce document décrit la composition et le fonctionnement du kit d'intégration JAVA. Il permet une mise en œuvre plus rapide de la solution Payline dans votre application.

Les kits sont proposés pour les plateformes JAVA, .NET et PHP qui couvrent la grande majorité des solutions e-commerce du marché.

Facile à utiliser, ils emploient les fonctionnalités monétiques de l'API Payline. Des classes permettent l'envoi de demandes, la lecture des réponses, et la mise en forme des données. Ces classes sont produites à partir du fichier de description WSDL de Payline.

1.2 Public visé

Ce document est destiné aux commerçants qui souhaitent intégrer dans leur système d'information des données issues de la solution de paiement Payline.

1.3 Contenu du document

Le chapitre «Présentation du kit JAVA» décrit la composition du kit, les fonctions Payline accessibles depuis le kit, la configuration nécessaire à son utilisation et les compatibilités logicielles de celui-ci.

1.4 Liste des documents de référence

Ce document complète le document de présentation fonctionnelle de Payline disponible sur le site Internet www.payline.com ou sur simple demande auprès du service d'assistance Payline.

1.5 Avertissement

Ce document est la propriété exclusive de Monext. Toute reproduction intégrale ou partielle, toute utilisation par des tiers, ou toute communication à des tiers, sans accord préalable écrit de Monext, est illicite.

1.6 Contacts

Vous avez besoin d'aide, de conseil ou vous souhaitez simplement nous poser une question. Contactez l'Assistance Payline par :

✉ Email : support@payline.com

Si vous découvrez une erreur dans cette documentation, vous pouvez nous envoyer un email en décrivant l'erreur ou le problème aussi précisément que possible. Merci de préciser la référence du document, sa date (indiquée sur la première page du document) et le(s) numéro(s) de page(s).

1.7 Historique des révisions

Le tableau ci-dessous liste les dernières modifications effectuées sur ce document.

| date | version | modifications |
|----------|---------|---|
| 26/10/09 | 1.0 | livraison version initiale |
| 14/03/11 | 1.1 | Nouveaux éléments de configuration |
| 06/04/11 | 1.2 | Précisions sur l'initialisation du contexte |
| 09/09/11 | 1.2.1 | Précision sur la configuration de la liste des contrats |

2 Présentation du kit JAVA

2.1 Composition du kit

Le kit JAVA est composé des éléments suivants :

- un fichier de paramétrage du kit de développement.
- une librairie de fonction qui permet d'utiliser les fonctions de l'API Payline
- un exemple d'application web utilisant la librairie.

2.2 Structure et fonctions

En décompressant un kit de développement, on trouve trois éléments :

2.2.1 Une application Web « webPayline.war »

L'archive « webPayline.war » contient une application Web présentant l'ensemble des fonctionnalités Payline disponible via le kit d'intégration. Chaque fonctionnalité se décline sous la forme de deux fichiers :

- un fichier HTML incluant le formulaire de présentation.
- un fichier JSP comportant le code JSP qui récupère l'ensemble des valeurs des champs transmis par le formulaire HTML et utilise la librairie pour effectuer la demande de Web Service.

Par exemple, la fonctionnalité « doWebPayment » qui permet la réalisation d'un paiement Web via le kit, se traduit par la présence des deux fichiers suivant :

- doWebPaymentForm.html
Ce fichier simule la page récapitulatif de la commande de votre client. Elle constitue l'étape 0 dans le processus de paiement web. Vous pouvez modifier les valeurs des champs du formulaire : Amount, Currency et OrderRef pour réaliser des paiements tests.
- doWebPayment.jsp
Ce fichier contient le code JSP qui récupère les informations transmises par le formulaire HTML et utilise la librairie pour initialiser une demande de paiement web.

2.2.2 La librairie « kitPayline.jar »

Cette librairie regroupe les classes qui décrivent les demandes, réponses et objets de l'API SOAP Payline. Une fois webPayline.war déployé, elle se trouve sous webPayline\WEB-INF\lib

- KitPayline.jar
Ce fichier contient la classe principale qui permet la création de messages SOAP ainsi que les autres classes spécifiques qui réalisent gèrent les demandes et réponses métier. Par exemple : doWebPayment, getWebPaymentDetails.

2.2.3 Le fichier de paramétrage « payline.properties »

Ce fichier regroupe l'ensemble de la configuration du kit : les paramètres propres à votre compte commerçant tel que l'identifiant commerçant, la clé d'accès aux services Payline, etc.

La description de ce fichier est décrite dans le chapitre ci-dessous « Configuration du kit ».

Une fois webPayline.war déployé, ce fichier se trouve sous webPayline\WEB-INF\classes.

2.3 Configuration du kit

Une fois le kit décompressé sur votre serveur, vous devez configurer les paramètres suivants dans le fichier « payline.properties » :

MERCHANT_ID : l'identifiant de votre compte commerçant,
ACCESS_KEY : la clé d'accès associé à votre compte commerçant,
CONTRACT_NUMBER : le numéro de contrat qui identifie votre point de vente et votre moyen de paiement par défaut,
RETURN_URL : l'URL de retour utilisée lorsque le paiement a été accepté,
CANCEL_URL : l'URL d'annulation utilisée lorsque le paiement a été refusé ou que votre client a annulé le paiement,
PRODUCTION : indicateur qui permet de basculer facilement de l'environnement d'homologation à la production.

Paramètres optionnels :

PAYMENT_ACTION : le code de la méthode de paiement à utiliser par défaut,
PAYMENT_MODE : le mode de paiement à utiliser par défaut,
PAYMENT_CURRENCY : le code ISO de la devise à utiliser par défaut,
ORDER_CURRENCY : le code ISO de la devise à utiliser par défaut,
SECURITY_MODE : le code du mode de sécurité à utiliser par défaut,
LANGUAGE_CODE : le code ISO de la langue à faire afficher par défaut,
NOTIFICATION_URL : l'URL de notification utilisée lorsque Payline vous notifie d'un paiement effectué,
SELECTED_CONTRACT_LIST : la liste des numéros de contrat à faire afficher, séparés par des « ; » (si vous en possédez plusieurs, sinon votre numéro de contrat unique). Laissez ce champ vide si vous souhaitez que tous vos contrats soient proposés.
CUSTOM_PAYMENT_PAGE_CODE : le code de personnalisation des pages de paiement Payline à utiliser par défaut.
PROXY_HOST : l'URL de votre proxy Internet,
PROXY_PORT : le port de communication de votre proxy Internet,
PROXY_LOGIN : l'identifiant utilisateur requis par votre proxy Internet,
PROXY_PASSWORD : le mot de passe utilisateur requis par de votre proxy Internet.
PRIMARY_HOMO_ENDPOINT_ADDRESS_WEB : L'url du endpoint des webservices Web d'homologation primaire
PRIMARY_HOMO_ENDPOINT_ADDRESS_DIRECT : L'url du endpoint des webservices Direct d'homologation primaire
PRIMARY_HOMO_ENDPOINT_ADDRESS_EXTENDED : L'url du endpoint des webservices Extended d'homologation primaire
SECONDARY_HOMO_ENDPOINT_ADDRESS_WEB : L'url du endpoint des webservices Web d'homologation secondaire
SECONDARY_HOMO_ENDPOINT_ADDRESS_DIRECT : L'url du endpoint des webservices Direct d'homologation secondaire
SECONDARY_HOMO_ENDPOINT_ADDRESS_EXTENDED : L'url du endpoint des webservices Extended d'homologation secondaire
PRIMARY_PROD_ENDPOINT_ADDRESS_WEB : L'url du endpoint des webservices Web de production primaire
PRIMARY_PROD_ENDPOINT_ADDRESS_DIRECT : L'url du endpoint des webservices Direct de production primaire
PRIMARY_PROD_ENDPOINT_ADDRESS_EXTENDED : L'url du endpoint des webservices Extended de production primaire

SECONDARY_PROD_ENDPOINT_ADDRESS_WEB : L'url du endpoint des webservices Web de production secondaire

SECONDARY_PROD_ENDPOINT_ADDRESS_DIRECT : L'url du endpoint des webservices Direct de production secondaire

SECONDARY_PROD_ENDPOINT_ADDRESS_EXTENDED : L'url du endpoint des webservices Extended de production secondaire

PRIMARY_CALL_TIMEOUT : La durée du timeout d'appel des webservices sur le site primaire

SECONDARY_CALL_TIMEOUT : La durée du timeout d'appel des webservices sur le site secondaire

PRIMARY_MAX_FAIL_RETRY : Le nombre de tentatives sur la chaine primaire par transaction

SECONDARY_MAX_FAIL_RETRY : Le nombre de tentatives sur la chaine secondaire par transaction

PRIMARY_REPLAY_TIMER : La durée d'attente avant le rejoue de la transaction sur le site primaire

SECONDARY_REPLAY_TIMER : La durée d'attente avant le rejoue de la transaction sur le site secondaire

PAYLINE_ERR_CODE : Les codes erreurs payline qui signifie l'échec de la transaction

PAYLINE_WS_SWITCH_ENABLE : Les noms des services web autorisés à basculer

PAYLINE_SWITCH_BACK_TIMER : La durée d'attente pour rebasculer en mode nominal

PRIMARY_TOKEN_PREFIX : Le préfixe du token sur le site primaire

SECONDARY_TOKEN_PREFIX : Le préfixe du token sur le site secondaire

PAYLINE_ERR_TOKEN : Les codes erreurs payline sur les préfixes des tokens

Attention : un redémarrage de votre serveur est nécessaire après toute modification de payline.properties pour la prise en compte des nouveaux paramètres.

2.4 Compatibilité du kit

Le kit JAVA est testé dans les environnements suivants :

- le système Windows XP Professionnel
- le serveur Tomcat 5.5
- Java 1.5 (testé avec java 1.6, nécessite l'utilisation du mécanisme endorsed : https://jaxb.dev.java.net/guide/Migrating_JAXB_2_0_applications_to_Java Confidential Page 7 Dernier enregistrement par JSERVAJEAN 15/11/2012SE_6.html#Using_JAXB_2_1_with_JavaSE_6)

2.5 Modification de l'application Web du client

Pour faire fonctionner la dernière version du kit JAVA certaines modifications de l'application web du client sont nécessaires :

Il faut tout d'abord intégrer dans les sources de l'application web la classe **AppListener.java** que vous trouverez dans les sources de l'application WebPayline (com.experian.payline.kit.listener).

Puis il faut déclarer dans le fichier web.xml le listener de la manière suivante :

```
<listener>
```

```
<listener-class>com.experian.payline.kit.Listener.AppListener</listener-  
class>  
</listener>
```

L'intérieur de la balise `listener-class` contient le chemin du fichier java en fonction de l'arborescence des packages.

Après ces modification le context `AppSwitcher` sera crée lors du démarrage de l'application sur le serveur d'application et pourra être récupérer avec la ligne de code suivante:

```
AppSwitcher appSwitcherFromContext =  
(AppSwitcher)Context.getAttribute("AppSwitcher");
```

2.6 Exemples

2.6.1 doAuthorization

La signature de la fonction `doAuthorization` a évolué avec la dernière version du kit JAVA. Un exemple d'utilisation est donné dans la `doAuthorization.jsp` sous `examples/direct/` :

```
DoAuthorizationResponse result = directPayment.doAuthorization(payment,  
order, buyer, card, privateDataList, authentication3DSecure, bank,  
Version, Context, ForceSwitchChain);
```

Les paramètres **bank**, **Version**, **Context** et **ForceSwitchChain** ont été introduits avec la version 1.0.11 du kit JAVA, développé pour la version 4.24 de Payline.

Pour un appel simple, **bank**, **Version** et **ForceSwitchChain** peuvent être valorisé à **null**. **Context** doit être valorisé avec la variable **appSwitcherFromContext** dont l'instanciation est expliquée dans le paragraphe précédent.