



Kit d'intégration JAVA

sommaire

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	5
2.1. Composition du kit	5
2.2. Structure	5
2.2.1. Une application web « webPayline.war ».....	5
2.2.2. La librairie « kitPayline »	6
2.2.3. Le fichier de paramétrage « payline.properties ».....	6
2.3. Fonctions	6
2.3.1. Appel	6
2.3.2. Retour.....	6
2.4. Configuration du kit	7
2.5. Exemples	7
2.5.1. doAuthorization.....	8
2.6. Compatibilité du kit.....	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	Précisions sur la configuration de la liste des contrats
30/07/13	1A	Suppression des paramètres de bascule Nouveaux exemples

2. Présentation du kit JAVA

2.1. Composition du kit

Le kit JAVA est composé des éléments suivants :

- une notice d'installation 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

En décompressant le kit de développement JAVA, 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 sous le répertoire *demos* nommé **<catégorie>-<nom_fonction>.jsp** (ou **.html**) contenant le formulaire de présentation.
- un fichier **<nom_fonction>.jsp** comportant le code qui récupère l'ensemble des valeurs transmises par le formulaire et utilise la librairie pour effectuer l'appel au web service.

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

- *demos/web-doWebPayment.jsp*
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.
- *web/doWebPayment.jsp*
Ce fichier contient le code JSP qui récupère les informations transmises par le formulaire et utilise la librairie pour initialiser une demande de paiement web.

2.2.2. La librairie « kitPayline »

Cette librairie regroupe les classes qui décrivent les objets de l'API SOAP Payline et les fonctions permettant de réaliser les appels aux web services Payline.

Une fois *webPayline.war* déployé, elle se trouve sous `webPayline\WEB-INF\lib`

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.

Les paramètres définis dans ce fichier sont décrits dans le chapitre *Configuration du kit*.

Une fois *webPayline.war* déployé, *payline.properties* se trouve sous `webPayline\WEB-INF\classes`.

2.3. Fonctions

L'ensemble des web services Payline est couvert par le kit JAVA. Les fonctions d'appel des web services sont réparties dans 4 classes : *DirectPayment*, *ExtendedAPI*, *WalletPayment* et *WebPayment*.

2.3.1. Appel

Le passage de paramètre entre votre boutique et ces fonctions se fait via les objets définis dans la librairie. Examinez les scripts d'exemple contenus dans le kit pour comprendre ce mécanisme.

2.3.2. Retour

La réponse d'un web service Payline est également composée d'objets définis dans la librairie. Examinez les scripts d'exemple contenus dans le kit pour comprendre la lecture d'une réponse.

Cas particulier :

Si lors de l'appel à une fonction du kit une exception se produit (par exemple en raison d'une absence de communication avec Payline du fait du mauvais paramétrage de l'authentification ou de proxy), une réponse contenant le code **XXXXX** ainsi que le libellé de l'exception est renvoyée.

Cette réponse n'est pas retournée par Payline, mais par la librairie.

2.4. 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.
SECOND_CONTRACT_LIST : la liste des numéros de contrats proposés à l'acheteur en cas d'échec du premier essai de paiement.
CUSTOM_PAGE_CODE : le code de personnalisation des pages de paiement Payline à utiliser par défaut.
CUSTOM_TEMPLATE_URL : URL de votre template de personnalisation des pages de paiement
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

2.5. Exemples

Les classes *DirectPayment*, *ExtendedAPI*, *WalletPayment* et *WebPayment* contenant les fonctions d'appel aux web services Payline possèdent deux constructeurs :

- Sans paramètre : la configuration utilisée est celle définie dans le fichier *payline.properties*
- Avec paramètre de type *ConnectParams* : ce constructeur ne tient pas compte du fichier *payline.properties* et permet d'utiliser des paramètres récupérés à la volée depuis votre système

2.5.1.doAuthorization

Un exemple d'appel à *doAuthorization* avec utilisation du constructeur sans paramètre de *DirectPayment* est fourni dans le script *direct/doAuthorization.jsp* de l'application *webPayline*.

Voici un exemple simple de l'utilisation du constructeur avec paramètre :

```
ConnectParams params = new ConnectParams(null, false, false, "19406042904914",
"A8*****oj", "proxy-host", "8080", null, null);
DirectPayment directPayment = new DirectPayment(params);
Payment payment = new Payment();
payment.setAmount("990");
payment.setCurrency("978");
payment.setAction("101");
payment.setMode("CPT");
payment.setContractNumber("CB");
Order order = new Order();
Date dNow = new Date();
order.setRef("KIT_"+dNow.getTime());
order.setAmount(payment.getAmount());
order.setCurrency(payment.getCurrency());
java.text.SimpleDateFormat s = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
order.setDate(s.format(dNow));
Card card = new Card();
card.setType("VISA");
card.setNumber("1111222233334444");
card.setExpirationDate("0114");
card.setCvx("123");
DoAuthorizationResponse res = directPayment.doAuthorization(payment, order, null,
card, null, null, null, null);
```

2.6. Compatibilité du kit

Le kit JAVA est testé dans les environnements suivants :

- Windows XP Pro / Windows 7 Pro
- Tomcat 5.5 et 6.0.35
- 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 8 Dernier enregistrement par JSERVAJEAN 30/07/2013SE_6.html#Using_JAXB_2_1_with_JavaSE_6)