JAVA SDK

Content

Introduction
The functions
SDK configuration
Example
SDK compatibility
Pages linked

Introduction

SDK composition

This version is no longer maintained since the Payline 4.44.1 release

The JAVA SDK is composed of the following elements:

- an installation guide for the development SDK.
- a function library which allows you to use the functions of the Payline API.
- an example of a web application using the library.

Download the SDK:

webPaylin....44.1.war

Structure

Throughout this article, *xx* refers to the Payline release number.

The JAVA SDK is made available as a **webPayline_4** package **.** ***xx*** **.war** ready to be deployed on an Apache Tomcat server.

The web application contained in this package presents all the Payline functionalities available via the Integration SDK. Each feature comes in the form of two files:

- a file under the demos directory named <category> - <function_name> .jsp (or .html) containing the presentation form.
- a <function_name> .jsp file comprising the code which retrieves all the values transmitted by the form and uses the library to make the call to the web service.

For example, the “doWebPayment” functionality, which enables a Web payment to be made, results in the presence of the following two files:

- demos / web-doWebPayment.jsp: This file simulates the summary page of your customer's order. It constitutes step 0 in the web payment process. You can modify the values of the form fields: Amount, Currency and OrderRef to make test payments.

- web / doWebPayment.jsp: This file contains the JSP code which retrieves the information transmitted by the form and uses the library to initiate a web payment request.

The **webPayline_4.** ***xx*** **.war** contains the **payline-4** library **.** ***xx*** **.jar** .
This library groups together the classes that describe the objects of the SOAP Payline API and the functions used to make calls to Payline web services.
Once the **webPayline_4.** ***xx*** **.war** deployed, it is located under webPayline_4. *xx* \ WEB-INF \ lib

The **webPayline_4.** ***xx*** **.war** contains the **commons-codec-1.10.jar library** .
This library must be added to your environment to use the AJAX API. It contains the Base64 class used by the exchange functions with the Payline getToken servlet.
Once the **webPayline_4.** ***xx*** **.war** deployed, it is located under webPayline_4. *xx* \ WEB-INF \ lib

The **webPayline_4.** ***xx*** **.war** also contains a **payline.properties** file .
This file contains all the configuration of the SDK: the parameters specific to your merchant account such as the merchant identifier, the access key to Payline services, etc.
The parameters defined in this file are described in the chapter Configuring the SDK.
Once **webPayline_4.** ***xx*** **.war** deployed, payline.properties can be found under webPayline_4. *xx* \ WEB-INF \ classes

The functions

All Payline web services are covered by the JAVA SDK. The web service call functions are divided into 4 classes: *DirectPayment* , *ExtendedAPI* , *Wall etPayment* and *WebPayment* .

Call

The parameter passing between your store and these functions is done via the objects defined in the library. Examine the example scripts contained in the SDK to understand this mechanism.

Return

The response of a Payline web service is also made up of objects defined in the library. Examine the sample scripts in the SDK to understand how to read a response.

Special case:
If an exception occurs when calling an SDK function (for example due to a lack of communication with Payline due to the incorrect configuration of the authentication or proxy), a response containing the code**XXXXX**along with the exception label is returned.
This response is not returned by Payline, but by the bookstore.

SDK configuration

Once the SDK is unzipped on your server, you must configure the following parameters in the "payline.properties" file:

MERCHANT_ID: the identifier of your merchant account,
ACCESS_KEY: the access key associated with your merchant account,
CONTRACT_NUMBER: the contract number that identifies your point of sale and your default payment method,
RETURN_URL: the return URL used when the payment has been accepted,
CANCEL_URL: the cancellation URL used when the payment was refused or your customer canceled the payment,
PRODUCTION: indicator that allows to easily switch from the approval environment to production .

Optional parameters:

PAYMENT_ACTION: the code of the payment method to use by default,
PAYMENT_MODE: the payment method to use by default,
PAYMENT_CURRENCY: the ISO code of the currency to use by default,
ORDER_CURRENCY: the ISO code of the currency to use by default ,
SECURITY_MODE: the code of the security mode to use by default,
LANGUAGE_CODE: the ISO code of the language to display by default,
NOTIFICATION_URL: the notification URL used when Payline notifies you of a payment made,
SELECTED_CONTRACT_LIST:the list of contract numbers to display, separated by "; "(If you have more than one, otherwise your unique contract number). Leave this field blank if you want all of your contracts to be offered.
SECOND_CONTRACT_LIST: the list of contract numbers offered to the buyer in case of failure of the first payment attempt.
CUSTOM_PAGE_CODE: the code for customizing the Payline payment pages to use by default.
CUSTOM_TEMPLATE_URL: URL of your payment page personalization template
PROXY_HOST: the URL of your Internet proxy,
PROXY_PORT: the communication port of your Internet proxy,
PROXY_LOGIN:the user identifier required by your Internet proxy,
PROXY_PASSWORD: the user password required by your Internet proxy

Example

The *DirectPayment, ExtendedAPI* , *WalletPayment* and *WebPayment classes* containing the functions for calling Payline web services have two constructors:

- Without parameters: the configuration used is the one defined in the *payline.properties* file *;*
- With *ConnectParams* type parameter : this constructor ignores the *payline.properties* file and allows you to use parameters retrieved on the fly from your system.

doAuthorization

An example of a call to *doAuthorization* using the parameterless constructor of *DirectPayment* is provided in the *direct / doAuthorization.jsp* script of the *webPayline* application .

Here is a simple example of using the constructor with parameter:

```
ConnectParams params = new ConnectParams ( null , false , false , " 19406042904194");
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 );
```

SDK compatibility

The JAVA SDK is tested in the following environments:

- Windows XP Pro / Windows 7 Pro
- Tomcat 5.5 and 6.0.35
- Java 1.5 (tested with java 1.6, requires the use of the endorsed mechanism: https://jaxb.dev.java.net/guide /Migrating_JAXB_2_0_applications_to_JavaConfidentielPage 8Last registered by JSERVAJEAN 30/07/2013SE_6.html # Using_JAXB_2ava_1_6with)

Pages linked

- .NET SDK
- JAVA SDK
- JAVA version 2 SDK
- PHP SDK
- SDK & Plugins