SLIM CD Developers Guide

For Shopping Cart Integrators

Contents

[SLIM CD Integration Options 3](#_Toc68869695)

[Other Considerations 3](#_Toc68869696)

[Advantages/Disadvantages 3](#_Toc68869697)

[Direct API calls 4](#_Toc68869698)

[Advantages 4](#_Toc68869699)

[Disadvantages 4](#_Toc68869700)

[Hosted Payment Pages 5](#_Toc68869701)

[Advantages 5](#_Toc68869702)

[Disadvantages 5](#_Toc68869703)

[Hosted JavaScript 6](#_Toc68869704)

[Advantages 6](#_Toc68869705)

[Disadvantages 6](#_Toc68869706)

[Surcharge/Convenience Fee Considerations 7](#_Toc68869707)

[3D-Secure Considerations 7](#_Toc68869708)

[Cart Configuration Screen 8](#_Toc68869709)

[URL Configuration Items 8](#_Toc68869710)

[User Configuration Items 8](#_Toc68869711)

[Transaction Configuration Items 8](#_Toc68869712)

[Hosted Payment Pages Integration 8](#_Toc68869713)

[SLIM CD Developer Documentation Links 9](#_Toc68869714)

[Developer Website 9](#_Toc68869715)

[Developer Videos 9](#_Toc68869716)

[Direct APIs and Data Fields 9](#_Toc68869717)

[PCI Considerations 9](#_Toc68869718)

[Hosted Payment Pages 9](#_Toc68869719)

[Hosted JavaScript 9](#_Toc68869720)

[User Credentials 9](#_Toc68869721)

[More Direct Links 10](#_Toc68869722)

[Sandbox Accounts 10](#_Toc68869723)

[Hosted Payment Pages APIs 10](#_Toc68869724)

[Hosted JavaScript API & Samples 10](#_Toc68869725)

[ProcessTransaction Documentation 10](#_Toc68869726)

[GetClientSites Documentation 11](#_Toc68869727)

[Notes on Surcharge 12](#_Toc68869728)

[Notes on Convenience Fees 12](#_Toc68869729)

[Notes on CheckMask 13](#_Toc68869730)

[Notes on Level 3 Indicator 14](#_Toc68869731)

[API Access Credentials 14](#_Toc68869732)

[Sample Code Downloads 14](#_Toc68869733)

[Marketplace Considerations 14](#_Toc68869734)

# SLIM CD Integration Options

SLIM CD’s APIs offer three distinct approaches to integration:

* Direct API calls from server-to-server, with the cardnumber transmitted (host server is in scope for PCI)
* Hosted Payment Pages, where the consumer’s browser is redirected to SLIMCD.COM to complete the transaction and then the consumer’s browser is redirected back to a page on the host server (host server is out of scope for PCI)
* Hosted JavaScript, where JavaScript from SLIMCD.COM is included into the payment page. The payment page must have certain characteristics. The payment fields (cardnumber, etc.) must not be sent back to the cart. And, the form’s “submit” must be intercepted by the JavaScript so that any cardholder data on the page is transmitted to SLIMCD.COM, and the response provides token data used to replace the cardholder data.

The integration requirements of the shopping cart may limit or exclude one or more of the above choices. The merchant’s desire to avoid PCI scope for their web server may also influence the decision on which technology to choose. If the cart’s requirements do not support the Hosted Payment Page or the cart does not allow customizations required to support the JavaScript, a direct API call (and the resulting requirement of PCI in-scope) may be the only option.

# Other Considerations

In today’s rapidly evolving payment space, different payment methods or features are becoming more popular. Examples include the ability to surcharge credit cards, add a convenience fee to all forms of payment, implement 3D-Secure fraud protection, add payment types (like ACH Check processing), etc.

The Hosted Payment Pages can support credit cards and checks. The Hosted Page can automatically perform processing for surcharge, including the identification of credit cards vs debit/check cards and only applying the surcharge when allowed. The Hosted Payment Pages support convenience fees for credit cards, debit cards, ACH checks, etc. The Hosted Payment Page can also support 3D Secure, a VISA/MC feature that shifts fraud liability to the cardholder by redirecting the consumer to their bank’s page to be validated before completing the credit card charge.

If other technology is used (Direct API calls or Hosted JavaScript), the developer must add each of the above features, as desired.

# Advantages/Disadvantages

Each of the above approaches offers advantages and disadvantages. The below sections detail the advantages and disadvantages of each integration approach. This information should be carefully considered and measured against the integration requirements of a specific shopping cart.

## Direct API calls

Direct API calls provide a method of performing an HTTP POST from the cart to the SLIM CD web services and receiving a synchronous response with approval or decline information. The cart will display a form (or forms) to collect all of the payment information. The payment itself is processed with a single web service from the cart platform to the SLIM CD ProcessTransaction web service.

### Advantages

* Direct API calls are easiest to implement because processing a payment is a simple server-to-server call with the cardnumber or checking account information.
* SLIM CD provides libraries that wrap web service calls for most popular languages
* Web services are available that can accept name/value pairs, JSON or XML, and return either JSON or XML

### Disadvantages

* The Direct API call must pass the cardnumber, which means the hosting server will “touch/store/transmit” the cardholder’s data and therefore is “in scope” for PCI
* If the merchant wishes to accept ACH checks, the developer must also collect this information on the form
* If the merchant desires surcharge or convenience fee functionality, the developer must call additional web services (see below) before processing the payment, and must be prepared to present “confirmation screens” to the cardholder before proceeding.
* If the merchant wishes to perform 3D Secure, the developer must perform a separate integration to a 3D Secure provider (including whatever certification they require)

## Hosted Payment Pages

A Hosted Payment Page requires the page template to be created on SLIM CD’s website. The page will contain the form and form fields, as well as many “control” features (such as payment types accepted, 3D Secure, etc)

A Hosted Payment Page is “controlled” by a Secure Session. A Session is “created” by calling the CreateSession API. The merchant credentials, name of the form, transaction type is required. Other information can be provided, such as the cardholder’s name, address, etc. Pages can be designed to display and/or edit this billing information, and the page should be customized to meet the requirements of the cart’s work flow. Hosted Payment Pages have two methods of communicating the results back to the cart. A POSTBACK URL can be configured in the page, which specifies a URL to receive data in a “silent post” (server-to-server data transfer with no user interaction). At the end of the process, a REDIRECT URL can be specified whereby the consumer’s browser is transferred back to the cart URL. The cart can also call CheckSession to pull transaction results for the session.

### Advantages

* The merchant’s web server will be out of scope for PCI
* The Hosted Payment Page can run as a stand-alone page or inside an <iframe>
* Credit cards and ACH checks are supported
* Surcharge support is supported
* Convenience fees are supported
* 3D Secure is supported
* The look and feel of the page can be customized
* The fields on the form can be customized and additional fields can be created to capture additional information (if desired)
* Secure Sessions’ Web services are available that can accept name/value pairs, JSON or XML, and return either JSON or XML

### Disadvantages

* The pages use tables and are not responsive/mobile friendly
* The developer must implement a method of retrieving the results from what happened on the page (by the silent POSTBACK URL or the Redirect URL and a CheckSession call)
* The POSTBACK URL is called for each payment attempt, so special consideration may need to be made to accept repeated updates for a single transaction (as multiple cards are attempted)
* The developer must decide how to handle either the “redirect” at the end of the process, or how to close the page/iframe once payment is complete. The developer will need to consider the best aesthetics and user-experience
* The Hosted Payment Page may have either SURCHARGE/CONVENINECE FEE enabled, which means the final amount may be greater than what was requested
* If the developer allows “split tender” across multiple forms of payment, it will be up to the developer to configure the page to allow “partial approvals” for less than the requested amount and handle all transaction flows to prompt for additional forms of payment OR void the partial approval. (Partial approvals are NOT enabled by default)
* SLIM CD does have a separate one-time “setup fee” for adding a Hosted Payment Page to a merchant’s account

## Hosted JavaScript

Hosted JavaScript requires the cart’s payment page to include a link to SLIM CD’s JavaScript code. The cart checkout page must be built with certain considerations to avoid the transmission of payment information to the cart itself. The developer must be prepared to have the payment page’s JavaScript convert the payment information into a “token”, and the code-behind for the page must accept that token and then perform a server-to-server Direct API call to actually process the transaction.

### Advantages

* The merchant’s web server will be out of scope for PCI
* The cart and the developer are in full control of the look-and-feel of the page

### Disadvantages

* Credit cards and ACH checks can be supported, but it is up to the developer to handle coding for these use-cased
* Surcharge support must be done via API calls, as well as any confirmation screens
* Convenience fees support must be done via API calls, as well as any confirmation screens
* 3D Secure may be difficult to implement this model, but may be possible with other JavaScript libraries of a 3D-Secure provider.

# Surcharge/Convenience Fee Considerations

Hosted Payment Pages have surcharge and convenience fee functionality “built in”. When using a Hosted Payment Page, the developer must be aware that the payment page could change the total amount of the payment and the “approved amount” may be greater than was requested. The amount of the surcharge will be returned in a separate field when the POSTBACK or the CheckSession is performed.

* If the merchant desires surcharge functionality, the developer must
	+ Call a web service to see if surcharge is enabled
	+ If a credit card is used (not ACH/Checks), call a web service to calculate the surcharge and, based on the response from the web service, prompt the user to accept the surcharge and new total
	+ Use the new the total and surcharge amount when performing the API call to the payment web service (ProcessTransaction)
* If the merchant wishes to charge convenience fees, the developer must
	+ Call a web service to see if convenience fee is enabled
	+ Call a web service to calculate the convenience fee and prompt the user to accept the new total
	+ Use the new the total and surcharge amount when performing the API call to the payment web service (ProcessTransaction)

# 3D-Secure Considerations

3D-Secure allows the merchant protection against chargebacks or disputes caused by fraudulent transactions. SLIM CD’s Hosted Payment Pages are already integrated with Paay.com.

Direct API calls – For direct API calls, simply provide the information retrieved from the 3D-Secure provider. However, acquiring that information requires a full integration with the 3D-Secure provider. This can be a considerable undertaking to integrate with a 3D-Secure provider or the provider may have tools to make this task manageable. See each provider for more information.

Hosted Payment Pages – Slim CD’s Hosted Payment Pages have built-in support for Paay.com’s 3D-Secure platform

Hosted JavaScript – In this case, the server-to-server calls to process the payment occur with the Direct API call (above), so the same considerations apply.

# Cart Configuration Screen

The shopping cart should have a configuration screen that allows SLIM CD Merchant Settings to be configured and saved.

## URL Configuration Items

Transaction Base URL – Used for sending transactions with ProcessTransaction

 <https://trans.slimcd.com>

Stats Base URL – Used to pull reports or display Hosted Payment Pages

 <https://stats.slimcd.com>

## User Configuration Items

UserName – Can be a ClientID, API ACCCESS CREDENTIAL or a SLIM CD “clerk” Username

Password – Blank for API ACCESS CREDENTIAL. Otherwise the password for the clientid/user

ClientID – Numeric Slim CD Client ID. (Can be zero for some API ACCESS CREDENTIALS)

SiteID – Numeric Slim CD Site ID. (Can be zero for some API ACCESS CREDENTIALS)

PriceID – Numeric Slim CD Price ID. (Can be zero for some API ACCESS CREDENTIALS)

Key - Software License Key (per ClientID). Email support@slimcd.com to request one per client.

## Transaction Configuration Items

Transaction Type should contain 2 options: AUTH or SALE, and default to SALE.

## Hosted Payment Pages Integration

This is only needed if you’re using Hosted Payment Pages.

Form Name – Name of the form to display.

(Note that at this time the PostBackURL and RedirectURL are coded into the form. Future enhancements may make it possible to send the PostbackURL & RedirectURL to CreateSession)

# SLIM CD Developer Documentation Links

Slim CD has a full developer’s website and a variety of sample code. We recommend a screen-sharing conference call with SLIM CD so we can help identify the best solution based on the requirements of the shopping cart. If you are familiar with the cart’s capabilities, we can present SLIM CD’s integration options and save you considerable time when performing an integration.

## Developer Website

Main Website: <https://developer.slimcd.com/>

Hosted Payment Pages: <https://developer.slimcd.com/hosted-pagessecure-sessions/>

<https://developer.slimcd.com/secure-sessions-flow/>

Hosted JavaScript: <https://developer.slimcd.com/your-page-with-javascript/>

API Reference: <https://developer.slimcd.com/reference/>

## Developer Videos

A variety of videos are available on various subjects.

* Overview of available APIs : <https://youtu.be/osNHVjCw6YQ>

### Direct APIs and Data Fields

* Direct API Connections: <https://youtu.be/bLbzeoHEf30>
* SLIM CD API Additional Data Fields: <https://youtu.be/uXkdA95BfJU>

### PCI Considerations

* PCI Scope Reduction Options: <https://youtu.be/xkCbRtgEekw>
* Acquiring and using Payment Tokens: <https://youtu.be/ZBrvMCmtQK0>

### Hosted Payment Pages

* Hosted Payment Pages Overview: <https://youtu.be/AHnzKFVOnKw>
* Hosted Page Flow Control: <https://youtu.be/j8LmI9edbbA>
* Hosted Page Design: <https://youtu.be/PbuEe_ZULI0>
* Hosted Page Form Fields: <https://youtu.be/I8DJBH9Dlto>

### Hosted JavaScript

* Hosted JavaScript Overview: <https://youtu.be/lEeHPi74Mok>

### User Credentials

* API Access Credentials: <https://youtu.be/4h-kzoBmJ7w>
* SlimCD Merchant Reporting Hierarchy Considerations: <https://youtu.be/NJt0VNb4k88>

## More Direct Links

### Sandbox Accounts

“Shared” test accounts: <https://developer.slimcd.com/test-accounts/>

### Hosted Payment Pages APIs

 CreateSession: <https://developer.slimcd.com/createsession/>

 ShowSession: <https://developer.slimcd.com/showsession/>

 CheckSession: <https://developer.slimcd.com/checksession/>

### Hosted JavaScript API & Samples

 YourPage with JavaScript: <https://developer.slimcd.com/your-page-with-javascript/>

### ProcessTransaction Documentation

ProcessTransaction can be used for credit cards or checks. It can accept either a cardnumber or a token.

 ProcessTransaction: <https://developer.slimcd.com/processtransaction-2/>

## GetClientSites Documentation

It should not be necessary to call GetClientSites, but we provide information herein because doing so can improve the integration into SLIM CD. It may be easier for the developer to have local configuration items, or to set the local items’ defaults based on this call. SLIM CD’s products use this call so that the end-user experience is completely controlled by our web services in a single portal, which we do recommend whenever possible.

The Entry Point GetClientSites provides information on a particular SiteID’s configuration on SLIM CD. A developer can call GetClientSites to determine if surcharge or convenience fee is enabled, what fields are needed to process checks, or the types of cards accepted by a merchant. (Note that an API ACCESS CREDENTIAL will need LOGIN permissions to call this web service)

GetClientSites: <https://developer.slimcd.com/getuserclientsites/>

Here is a sample response in XML:

<reply>

<response>Success</response>

<responsecode>0</responsecode>

<description>OK</description>

<responseurl>https://stats.slimcd.com/wswebservices/login.asmx/GetUserClientSite3</responseurl>

<datablock>

<SiteList>

 <Site>

<ClientID>1032</ClientID>

<SiteID>228226448</SiteID>

<PriceID>74</PriceID>

<MerchantID> </MerchantID>

<BusinessType>2</BusinessType>

<CreditRoute>TST</CreditRoute>

<CheckRoute>CER</CheckRoute>

<GiftRoute>FTG</GiftRoute>

<CardTypes>VMADJCBT</CardTypes>

<CheckMask>11111111</CheckMask>

<ClientName>RETAIL TEST ACCOUNT</ClientName>

<SiteName>RETAIL TEST ACCOUNT</SiteName>

<DBA> </DBA>

<Address>610 N University Drive</Address>

<Address2> </Address2>

<City>PARKLAND</City>

<State>FL</State>

<Zip>33071</Zip>

<Country>840</Country>

<Phone>(954) 752-9309</Phone>

<Email>wureal2@slimcd.com</Email>

<surcharge\_percentage>3</surcharge\_percentage>

<amex\_percentage>3</amex\_percentage>

<force\_surcharge>False</force\_surcharge>

<skip\_surcharge\_binlookup>False</skip\_surcharge\_binlookup>

<sitetype>E</sitetype>

<level3indicator>True</level3indicator>

<conveniencefee\_enabled>False</conveniencefee\_enabled>

<conveniencefee\_settings>0</conveniencefee\_settings>

<conveniencefee\_receiptlabel>Convenience Fee:</conveniencefee\_receiptlabel>

<ECC/>

<CER/>

<GTI/>

<CGW/>

<GMS/>

<VER/>

 </Site>

</SiteList>

</datablock>

</reply>

Items that may be of interest for shopping cart integration are highlighted in yellow above.

Notes on Surcharge

 If the surcharge percent and/or AMEX percent are non-zero, surcharge is enabled.  The FORCE SURCHARGE is an indicator that says don’t let users manually enter a surcharge amount (force the auto-calculated call to GETSURCHARGE).  The skip\_surcharge\_binlookup is used for Governments that are allowed to surcharge ALL cards (including DEBIT cards, not just credit cards)

Example in XML:

<surcharge\_percentage>3</surcharge\_percentage>

<amex\_percentage>3</amex\_percentage>

<force\_surcharge>False</force\_surcharge>

<skip\_surcharge\_binlookup>False</skip\_surcharge\_binlookup>

See the SLIM CD documentation on Surcharge for more information on calling the GetSurcharge web service.

Notes on Convenience Fees

 If the conveniencefee\_enabled indicator is True, then convenience fee is enabled.  The conveniencefee\_settings can provide insight into the need to call GetConveniceFee for a particular payment method. It’s not necessary to do this, as it will return 0 convenience fee for that payment type if it’s not enabled.

Example in XML:

<conveniencefee\_enabled>False</conveniencefee\_enabled>

<conveniencefee\_settings>0</conveniencefee\_settings>

<conveniencefee\_receiptlabel>Convenience Fee:</conveniencefee\_receiptlabel>

|  |  |
| --- | --- |
| Position | Meaning |
| 1 | Y=Enabled for Credit Card |
| 2 | Y=Enabled for Electronic Check/ACH |
| 3 | Y=Enabled for Paper Checks (in person) |
| 4 | Y=Enabled for PIN Debit |
| 5 | Y=Enabled for Gift Cards |
| 6 | Y=Enabled for Cash Payments |

The conveniencefee\_receiptlabel also includes the text to show on the screen as the “prompt” for a convenience fee. Note that this will come back as part of GetConvenienceFee, so it is not necessary to get this information from GetClientSites.

Notes on CheckMask

The CheckMask indicates which fields on the form should be displayed for the particular check processor. It is also acceptable to display all fields all the time, without hiding fields that are not needed. Note that StateCode, Driver’s License and Birth Date are mainly for in-person check verification/guarantee, and are not normally used for ACH processing.

Example in XML:

<CheckMask>11111111</CheckMask>

|  |  |
| --- | --- |
| Position | Meaning |
| 2 | Show Account Number Entry Field |
| 3 | Show Check Number Entry Field |
| 4 | Show Route Number Entry Field |
| 5 | Show State Code Entry Field (list can be provided by SLIM CD, but it’s 2-character states, plus SS#, etc. |
| 6 | Show Driver’s License Entry Field |
| 7 | Show Birthdate entry field |
| 8 | Show Personal Data entry fields |

Notes on card types:

Example XML:

<CardTypes>VMADJCBT</CardTypes>

* V=Visa
* M=MasterCard
* A=Amex
* D=Discover, J=JCB, C=Carte Blanche
* B=PIN DEBIT
* T=EBT

Note that D, J and C are all mapped to Discover (they use Discover routing). So a Discover card can be any of the D, J or C cards. The JavaScript, PHP and .NET libraries all have examples to determine a card’s type based on the cardnumber.

Notes on Level 3 Indicator

If the level 3 indicator is on, the developer can send line items for each item in the cart. See the level 3 documentation for more information.

Example in XML:

<level3indicator>True</level3indicator>

## API Access Credentials

Log into <https://www.slimcd.com> with test account credentials and click on CONFIG/API ACCESS CREDENTIALS to create a credential.

## Sample Code Downloads

Download PHP, .NET and other sample code here: <https://stats.slimcd.com/developers/downloads>

Note that there are sections for Hosted Payment Pages Samples, PHP libraries and samples and for JavaScript.

# Marketplace Considerations

Many carts have a Marketplace or plug-in portal for webmasters or merchants to access when acquiring plugins or integrations. Each cart will have its own installation requirements and plugin submission requirements. It is outside the scope of this document to include specific information on such details.