SmartConnect Integration Guide

for Point-of-Sale Software Vendors

Version: 1.3

www.smartpaydev.com 0800 4 SMARTPAY (0800 476 278) info@smartpaydev.com

Confidentiality and Intellectual Property Rights Statement

This document is a part of a software development kit (SDK) you have downloaded from the smartpay.com website or have received from a third party by some other means. You, as the reader, are aware that all information, whatsoever, contained in this SDK is confidential (Confidential Information) and is the property of Smartpay Limited (Smartpay). or its licensors and as such protected by law. This document or any information it contains, whether text, numbers, tables, pictures, graphs or other, may not be disclosed, copied, reproduced or distributed to a third party without the prior consent of Smartpay.

You may only use this SDK to help you connect to Smartpay payment solutions. You may not directly or indirectly, in any way, reveal, report, publish, disclose, transfer or otherwise use or exploit any of the Confidential Information except as specifically authorized by Smartpay. Furthermore, you may not reverse engineer any of the contents of the SDK or use any Confidential Information to compete with Smartpay or obtain advantage vis-à-vis in any commercial activity.

Misuse or unauthorized distribution of this document or the information it contains may result in Smartpay seeking legal measures through the appropriate authorities.

No part of this document may be copied or reproduced in any form without prior written consent from Smartpay.

The information in this document is subject to change without notice.

Disclaimer

Smartpay Limited has taken great effort to verify the accuracy of this document, but we assume no responsibility for any technical inaccuracies or typographical errors. In no event, shall Smartpay Limited be liable for any direct, indirect, special or incidental damage resulting from, arising out of or in connection with the use of the information.

Overview

The purpose of this document is to outline how a (generic) Point-Of-Sale software integrates with SmartConnect APIs to provide a seamless and integrated payment experience to the customer.
Note that while this is the desired way to integrate with any given POS vendor, given the circumstances and limitations of the system – SmartConnect can try to accommodate a slightly different flow, if needed. This can be discussed with the development team on a case by case basis.

What is SmartConnect?

SmartConnect is an efficient, scalable & secure cloud-based API platform. It pairs any API-enabled payment initiator (such as cloud/on-prem POS software), running on any device (PC, Mac, tablet, phone), to any compatible payment terminal device (such as PAX, Verifone, Ingenico, CBA Albert, and more - mobile phones and other devices).
SmartConnect has an Open API architecture that enables 3rd parties to integrate and grow the eco-system with ease. Data is collected and stored in real-time, empowering data APIs for various insights and analytics.

The Integration Process

The integration process is divided into three steps:

1) The Store, that owns one or more Payment Terminals, needs to be registered as a “Merchant” in the SmartConnect database. SmartConnect has APIs exposed for this, and the API call to do so is typically something the terminal vendor does through their “TMS” (Terminal Management System).
   Note that if a business entity owns more than one physical stores (on different locations), each one of them will have their own Merchant Id.
   The process of setting up a Merchant is outside the scope of this document, and the POS software vendor assumes that this has already been completed.

2) A **Register** in the POS software, that belongs to a **Store**, needs to be paired to the given Payment Terminal. One Register can only be paired to one Payment Terminal, and vice-versa, one Payment Terminal can only be paired to one Register, at a given point in time.
   The pairing process is explained in the Pairing with the Terminal section.

3) Once a Register is paired to the Payment Terminal, it can utilize SmartConnect APIs to initiate transactions.
   The APIs are explained in more detail in the Initiating a Transaction section.

Pairing with the Terminal

There are three main actors in the pairing process:

- Terminal – the Payment Terminal device.
- SmartConnect – the cloud based API platform that connects the Transaction Initiator to the Payment Terminal.
- POS Software – the Transaction Initiator.

The flow has two key steps:

1) The pairing process is initiated on the Payment Terminal, which connects to SmartConnect APIs and retrieves a temporary **pairing code**. This code will be displayed on the terminal for a set amount of time.
2) The POS Software needs to make an API call and supply the **pairing code** together with the **Register Id** and some other meta data.

The POS software needs to have a **UI component** that enables the user to input the given pairing code. This pairing code is then sent via an API call to SmartConnect. The API call will return a successful response once pairing is complete. The POS software does not need to persist any Payment Terminal metadata in its database.

To initiate pairing from the POS, execute the following HTTP request:

```
PUT https://api-dev.smart-connect.cloud/POS/Pairing/{pairingCode}
```

The pairingCode is the code displayed on the Payment Terminal.
Using the HTTPS protocol is mandatory. Note that the api-dev subdomain should be used in a development environment only. To use in a production environment, use the api subdomain:

```
PUT https://api.smart-connect.cloud/POS/Pairing/{pairingCode}
```

The following parameters need to be supplied in the request body, with the content type **application/x-www-form-urlencoded**:

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSRegisterName	Main Register	Register Name	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
ContactName	John Doe		**No**
Address	123 Some Street		**No**
City	Auckland		**No**
State	Auckland		**No**
ZipCode	1023		**No**
Email	john.doe@toys4nz.co.nz		**No**

Phone	09 1234567	**No**

Initiating a Transaction

Once pairing is complete, the Register can initiate a transaction to the Payment Terminal by calling the SmartConnect API. The POS does not need to know the device Id or any other Terminal information, SmartConnect will automatically know which Payment Terminal to invoke from the cloud.

To start the transaction process, the POS needs to execute the following HTTP request:

```
POST https://api-dev.smart-connect.cloud/POS/Transaction
```

Using the HTTPS protocol is mandatory. Note that the **api-dev** subdomain should be used in a development environment only. To use in a production environment, use the **api** subdomain:

```
POST https://api.smart-connect.cloud/POS/Transaction
```

Parameters need to be supplied in the request body, with the content type **application/x-www-form-urlencoded**:

The Register Id and the Store Name need to match the respective parameters supplied when pairing the Terminal.

The following transaction types are available:

Logon

This function requests the terminal to perform a logon operation with an acquirer.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Acquirer.Logon		**Yes**

Settlement Inquiry

This function requests the terminal to perform a settlement inquiry operation with an acquirer.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Acquirer.Settlement.Inquiry		**Yes**
Date {DATE}	20170613	The inquiry date	**No**

Settlement Cutover

This function requests the terminal to perform a settlement cutover operation with an acquirer.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Acquirer.Settlement.Cutover		**Yes**

Purchase

This function requests the terminal to initiate a purchase transaction for a specific amount.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Card.Purchase		**Yes**
AmountTotal {AMOUNT}	500		**Yes**

Purchase + Cash

This function requests the terminal to initiate a purchase transaction with a cash amount.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Card.PurchasePlusCash		**Yes**
AmountTotal {AMOUNT}	500	The total amount of the transaction, **including** the cash amount.	**Yes**
AmountCash {AMOUNT}	100	Cash portion of the AmountTotal	**Yes**

Cash Advance

This function requests the terminal to initiate a cash-out only transaction.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**

TransactionType	Card.CashAdvance	**Yes**
AmountTotal {AMOUNT}	500	**Yes**

Refund

This function requests the terminal to initiate a refund transaction.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Card.Refund		**Yes**
AmountTotal {AMOUNT}	500		**Yes**

Authorise

This function requests the terminal to initiate an auth transaction, to reserve funds in the cardholder's account.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Card.Authorise		**Yes**
AmountAuth {AMOUNT}	500		**Yes**
TransactionReference	AUTH01	Reference string, alphanumeric, up to 8 characters *	**No**

* *The TransactionReference is used to look up and retrieve the original auth information when finalising. If a TransactionReference is not supplied by the POS, the terminal will prompt for this at the beginning of the transaction. If the TransactionReference matches an existing stored transaction, the terminal will prompt whether to continue. If YES, the existing auth transaction will be updated; if NO, the transaction will be cancelled. In all cases, the value used for the transaction is returned in the result message (for accepted transactions) and that value must be used when finalizing.*

Finalise

This function requests the terminal to finalise an auth transaction, debiting funds from the cardholder's account.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Card.Finalise		**Yes**
AmountFinal {AMOUNT}	500		**Yes**
TransactionReference	AUTH01	The reference string for the original auth, from the auth result.	**Yes**

Get Transaction Result

This function is used to retrieve the result of the last financial transaction.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Journal.GetTransResult		**Yes**

Reprint Receipt

This function is used to retrieve the result of the last financial transaction.

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Journal.ReprintReceipt		**Yes**

Get Terminal Status

This function requests the current status of the terminal. {TERMINAL- STATUS}

Parameter Name	*Example Value*	*Comment*	*Mandatory*
POSRegisterID	7444ae07-dc63-e49c-33e3-59a7c108cc80	Unique Register Id	**Yes**
POSBusinessName	toys4nz	Store name	**Yes**
POSVendorName	Till2Go	POS software vendor	**Yes**
TransactionType	Terminal.GetStatus		**Yes**

Surcharging/Tipping

Surcharge and tip are generally applied at the terminal during the customer input phase of a transaction. As a result, a POS may request a transaction for $x, but the total amount processed by the terminal and host may be $x+y. Where it is expected that a surcharge and/or tip may apply, it is incumbent on the POS to check for amount fields in the response to determine what the final transaction amount is.

The absence of amount fields in the response is indicative that the processed amount is the same as the requested amount.

API Response

After the Transaction has finished on the Terminal (regardless of the outcome, or if it times out), the HTTP response will contain a JSON object, which will have the structure as per the table below.

Arguments and Result types

Type Name	*Values*	*Description*
{AMOUNT}		Amounts are specified in the minor currency units of the default currency of the terminal. For example, NZD$19.95 becomes "1995".
{DATE}		Must be in the format YYYYMMDD.
{RESULT}	One of: OK CANCELLED FAILED FAILED-INTERFACE	 The function was processed successfully. The function was cancelled. The function failed. The function failed because of a communications or interface error.
{TRANS-RESULT}	One of: OK-ACCEPTED OK-DECLINED OK-UNAVAILABLE CANCELLED FAILED FAILED-INTERFACE	 The transaction was accepted. The transaction was declined. The transaction could not be processed at this time. The transaction was cancelled by the operator or customer, or by the terminal. The transaction failed. The transaction failed because of a communications or interface error.
{TERMINAL-STATUS}	One or more of:* READY NOT_INITIALIZED NOT_LOGGED_ON REVERSAL_PENDING OFFLINE	 The terminal is ready to process transactions. The terminal requires initialization. The terminal is not logged on to the acquirer. The terminal has a reversal pending for a previous transaction. The terminal is currently operating in EOV/EFB (offline) mode.

**Should more than one state be applicable at the same time, the terminal may return multiple states as a comma-separated string; e.g. "NOT_LOGGED_ON,REVERSAL_PENDING,OFFLINE".*

Parameter Name	*Type*	*Comment*	*Presence*
root	object		**Always**
→ transactionId	string	Unique transaction identifier.	**Always**
→ transactionTimeStamp	string	Date and time of the transaction, as generated on the server (UTC time).	**Always**
→ merchantId	string	Unique merchant (store) identifier.	**Always**
→ deviceId	string	Unique device identifier.	**Always**
→ transactionStatus	string	Status of the transaction, one of the following values: • **PENDING** – the result of the transaction is pending from the terminal (does not happen when invoked from the iframe) • **COMPLETED** – the transaction is successfully completed on the Terminal	**Always**
→ data	object		Optional
→ TransactionResult	string	Transaction outcome, one of the following values: **{TRANS-RESULT}**	**Always**
→ Receipt	string	Receipt data, as generated by the Terminal.	Optional
→ RequestId	string	This will be the same as the transaction Id.	Optional
→ AcquirerRef	string	System trace/audit number.	Optional
→ AccountType	string	The account type selected on the Terminal, if applicable.	Optional
→ Timestamp	string	Date and time of the transaction, from the Terminal (local time).	Optional
→ Result	string	An indication of whether the request was processed by the Terminal, one of the following values: **{RESULT}**	**Always**
→ Function	string	Terminal transaction type invoked.	**Always**
→ AuthId	string	For transactions authorized by a third party (e.g. Paymark), this will be the authorization id.	Optional
→ CardPan	string	The last four digits of the card used on the Terminal.	Optional
→ AmountTotal	string	The total amount paid on the Terminal.	Optional
→ Merchant	string	The merchant Id, as configured on the Terminal.	Optional
→ CardType	string	The type of card, as returned by the Terminal.	Optional
→ TerminalRef	string	The Terminal Id, as configured on the Terminal.	Optional
→ AmountSurcharge	string	The surcharge amount (part of the total amount), if configured and entered on the Terminal.	Optional
→ AmountTip	string	The tip amount (part of the total amount), if configured and entered on the Terminal.	Optional

An example response:


```
{
  "transactionId": "3ec7084a-296f-4d2e-93bc-53b0860bbc68",
  "transactionTimeStamp": "201702231038363427",
  "merchantId": "0f8fad5b-d9cb-469f-a165-70867728950e",
  "deviceId": "9999999991",
  "transactionStatus": "COMPLETED",
  "data": {
    "TransactionResult": "OK-ACCEPTED",
    "Receipt": "PAYMARK PKMS         \n        TEST TERMINAL         \n        182
Wairau Rd         \n\n*-----------EFTPOS-----------*\nTERMINAL
00906604\nTIME             23FEB17 10:38\nTRAN 000135             CREDIT\nEMV TEST
CARD\nCARD                  ....0138\nCONTACTLESS\nVisa Debit\nRID: A000000003\nPIX:
1010\nARQC: CCD5BD398510E46E\nTVR: 0000000000\nATC: 0322\nTSI: 0000\nAUTH 123456\nREF NO
000144      \nPURCHASE               NZ$ 5.00\nTOTAL                NZ$ 5.00\n\n
ACCEPTED           \n******DUPLICATE  RECEIPT******\n*----------------------------*",
    "RequestId": "3ec7084a-296f-4d2e-93bc-53b0860bbc68",
    "AcquirerRef": "000135",
    "AccountType": "CREDIT",
    "Timestamp": "20170223103846",
    "Result": "OK",
    "Function": "Card.Purchase",
    "AuthId": "123456",
    "CardPan": "....0138",
    "AmountTotal": "500",
    "Merchant": "1",
    "CardType": "EMV TEST CARD",
    "TerminalRef": "00906604",
    "PosData": "Pos data"
  }
}
```