CASTLES TECHNOLOGY

Castles – Saturn1000

Payment Application Integration

Confidential

Version 0.18

2024/12/12

Castles Technology Co., Ltd.

6F, No. 207-5, Sec. 3, Beixin Rd., XindianDistrict,

New Taipei City 23143, Taiwan R.O.C.

http://www.castech.com.tw

WARNING

Information in this document is subject to change without prior notice.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of **Castles Technology Co., Ltd.**

All trademarks mentioned are proprietary of their respective owners.

Revision History

Version	*Date*	*Descriptions*	*Author*
0.1	2021/08/20	Initial draft.	Jack Li
0.2	2021/08/23	Modify 2.3 command format Add 2.4 communication protocol Modify 3.1 – 3.7 with command detail Add appendix-A JSON key list Add appendix-B return code list	Jack Li
0.3	2021/08/25	Modify “2.4 Communication Protocol” description Modify “3.1 Get Data” request info Modify “3.5 Settlement” request and response info Modify “3.7 Auth-Complete” request info Modify “Appendix-A JSON Key List” Modify “Appendix-B Return Code List”	Jack Li
0.4	2021/08/27	Modify “3.5 Settlement” request and response info Modify “3.7 Auth-Complete” request info Modify “Appendix-A JSON Key List” Modify “Appendix-B Return Code List”	Jack Li
0.5	2021/09/27	1. add “1.3 POS and ECR Integration” 2. modify description of “2.1 Start Payment Service” 3. add “2.1.1 Start Service from Menu” 4. modify description of “2.1.2 Start Service from the 3rd party application” 5. modify description of “2.2 Socket Connection” 6. add error code of “Appendix-B Return Code List” E000000D settlement fail E000000E repeated transaction ID (txnPosTxnId) E000000F printer out of paper E0000010 printer error	Jack Li

0.6	2021/10/1	1. add “3.8 Display” 2. add “3.9 Return to Idle” 3. add “3.10 Scan Barcode” 4. update “Appendis A – JSON Key List” txnMsgLine1 txnMsgLine2 ... txnMsgLine8 txnBarcodeFormat txnBarcodeData txnBarcodeImgOffset txnTimeout	Jack Li
0.7	2022/04/29	Modify “2.4 Communication Protocol” description about socket connection	Jack Li
0.8	2022/05/26	Modify “Appendix-B Return Code List”	Jack Li
0.9	2022/6/17	Add Sample reqeuest command to each transaction Modify “3.1 Get Data” request and response info Modify “3.2 Sale” request and response info Modify “3.3 Void” request and response info Modify “3.4 Refund” request and response info Modify “3.6 Pre-Auth” request and response info Modify “3.7 Auth-Complete” request and response info	Jack Li
0.10	2022/07/19	1. add error code of “Appendix-B Return Code List” E0000002A Please re-try Settle	Jack Li
0.11	2023/02/08	1. add Chapter 2.1.3 “Checking Service from the 3rd Party Application”	Jack Li
0.12	2023/02/09	Corrected description	Bob Parry

0.13	2023/03/24	1. Add service package name in responses of Checking Service 2. Add command “reprint” to reprint receipts 3. Add txnCoupon in sale command	Jack Li
0.13.1	2023/05/05	Add chapter-4 for ITS commands	Jack
0.13.2	2023/05/05	1. Change title of chapter 3 and 4 for socket server and client mode 2. Update 4.6 Host Payment – Void 3. Update “txnEmvTagList” description	Jack
0.13.3	2023/05/15	1. Update “txnEmvTagList” description 2. Add “Appendix C – EMV Tag List” 3. Fix typo	Jack
0.14	2023/12/11	1. Update “return2Idle” infomation	Jack
0.15	2024/06/11	1. Add “txnInvoiceNum“, “txnRrn” , “txnBatchNum” ■ 3.2. Sale ■ 3.3. Void ■ 3.4. Refund (Return) ■ 3.5. Settlement ■ 3.6. Pre-Auth (Auth) ■ 3.7. Auth-Complete (Capture) ■ 4.5. Host Payment – Sale ■ 4.5. Host Payment – Void ■ 4.5. Host Payment – Refund ■ “The Transaction JSON Key” 2. Add “txnAID“, “txnTC” ■ 3.2. Sale ■ 3.3. Void ■ 3.4. Refund (Return) ■ 3.6. Pre-Auth (Auth) ■ 3.7. Auth-Complete (Capture) ■ 4.5. Host Payment – Sale ■ 4.5. Host Payment – Void ■ 4.5. Host Payment – Refund ■ “The Transaction JSON Key”	David
0.16	2024/06/24	1. Update Appendix B – Return Code List	Wayne

		2. Add “HostDAGetCount”,”HostDAUpload” trasnaction	
0.17	2024/07/18	1. Add description of each service type in 2.1.1	
0.18	2024/12/12	1. Add cashbackRequest 2. Add status Notification 3. Add infSN parameter in getData	Wayne

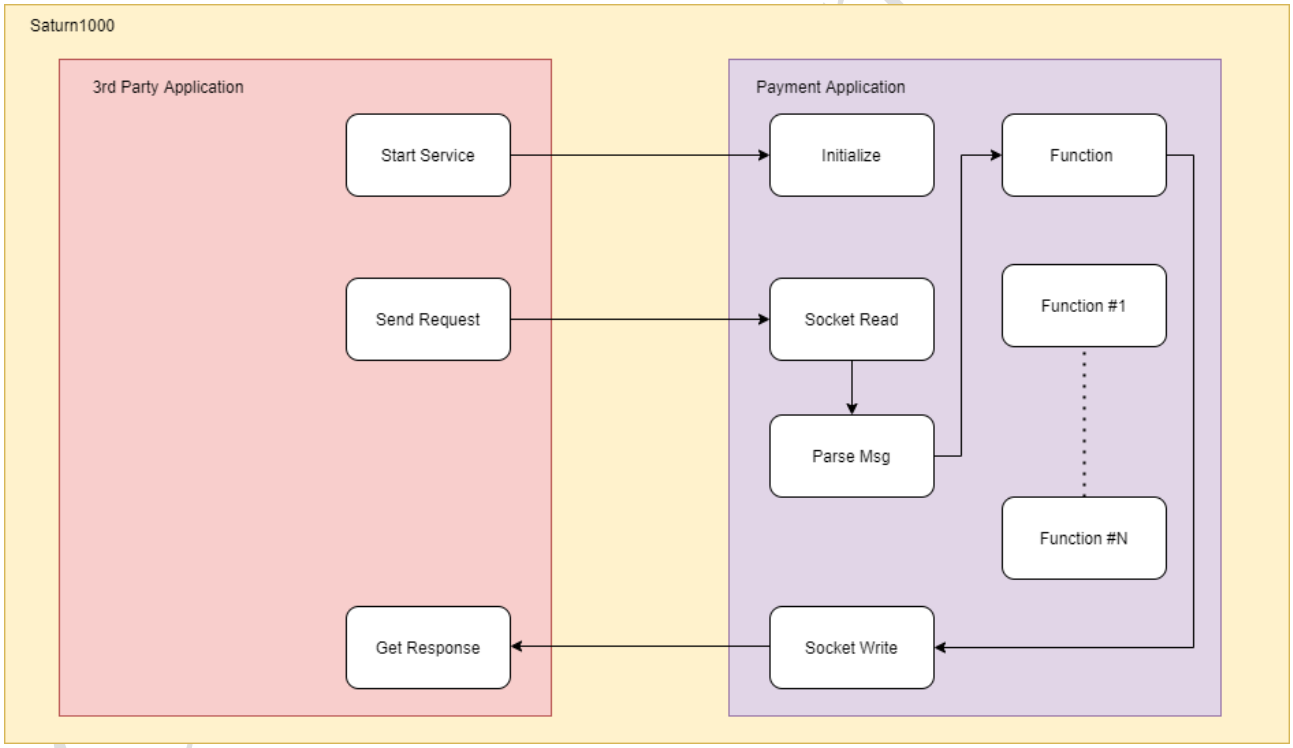
Table of Contents

1. Introduction

The Saturn1000 is an Android PCI-PTS certified terminal and Castles provides a payment application to run on it. This document will show you how to quickly integrate with the CastlesPay payment application on the Castles Saturn1000 terminal.

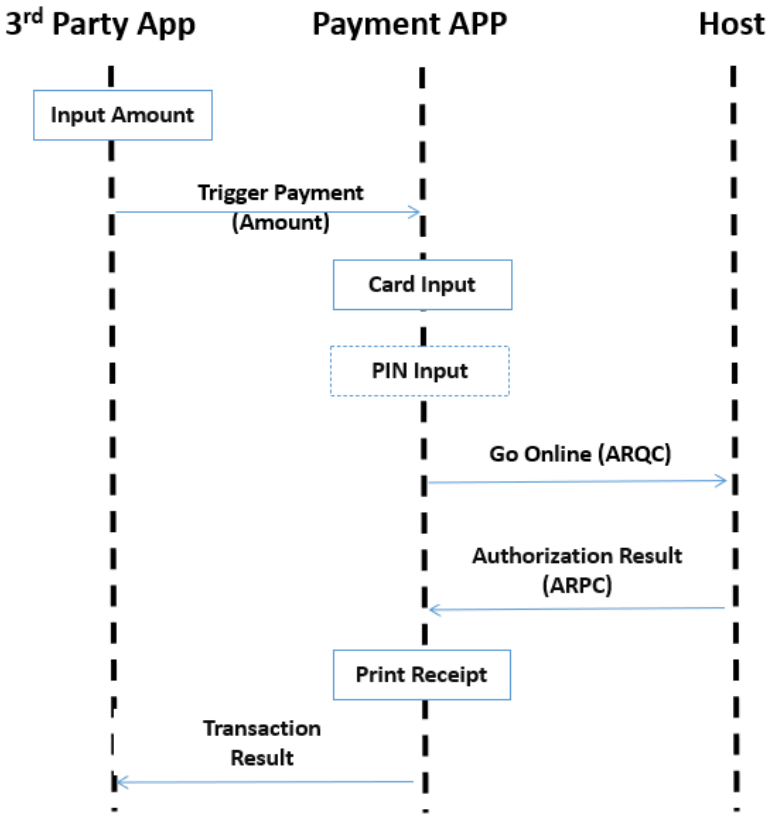
1.1. Architecture

The payment application consists of a payment service which can receive the command from a socket connection. The 3rd party application can trigger the payment service and the socket server will be initialized automatically. After that, the 3rd party application can use the socket connection to send the payment request command (e.g. “sale” request) to the payment application and the payment application will display the present card UI with the amount from the 3rd party application.



1.2. Transaction Walkthrough

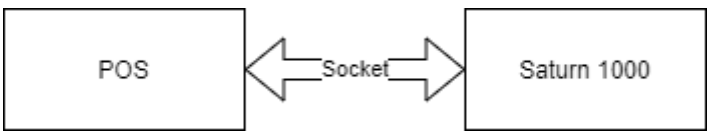
Below is an example which shows how an online transaction works betweens the 3rd party application, payment application and the host.



Note: “PIN input” is optional depending on CVM request.

1.3. POS and ECR Integration

The payment service also supports external POS (Point of Sale) or ECR (Electronic Cash Register) integration. A POS/ECR can send a request via the socket to the payment application, using the same interface as a 3rd party application on the terminal. The only difference is the use of a different IP/Port for socket connection. Please check chapter “2.1.2 Start Service from Menu” for more detail.



2. Implementation

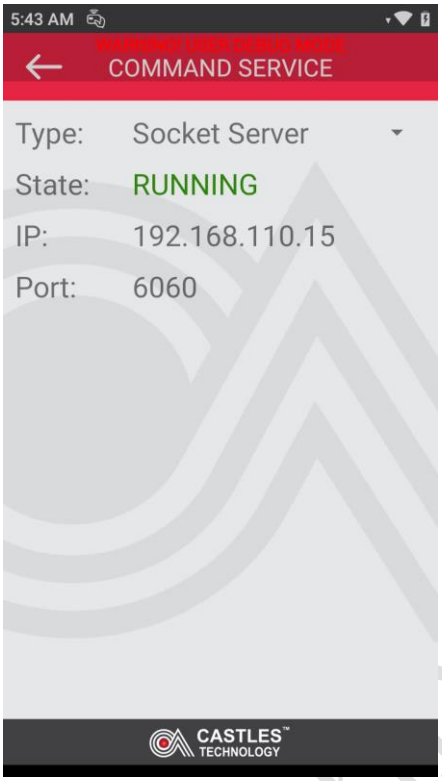
2.1. Start Payment Service

To enable Payment Service, the user can enable the payment service from the menu, or it can be enabled via the TMS (CasHUB) or from the 3rd party application.

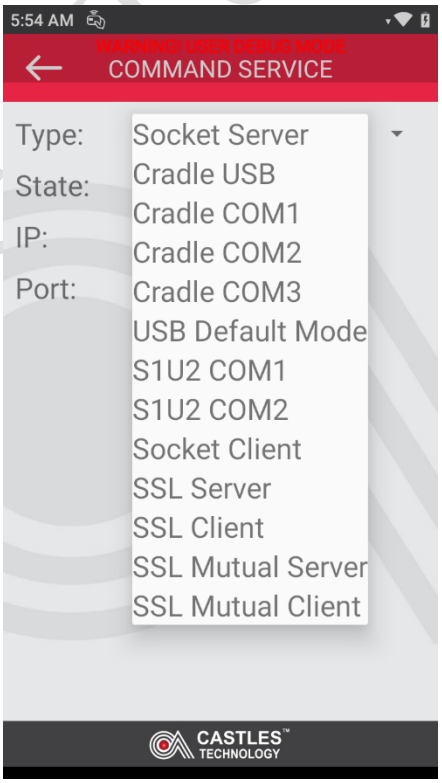
2.1.1. Start Service from Menu

The “CMD SERVICE” button can be found in the payment applicatoin‘s menu page, as shown in image below:

Select the “CMD SERVICE” button:



The user can see information about the service: service state, server IP and port. The service enable flag can be configured via the TMS system. If the flag is set to “true”, the payment application will start the service automatically after initialization.



The user can change the service type by using the drop down list. There are several options to send the command to terminal by setting:

- **Socket Server**

The terminal will be the socket server. Other apps can connect to the AP by using the IP/Port set on the “Command Service” page.

- **Cradle USB**

Users need to connect to the S1F2 Cradle’s wifi first, and then send commands through Cradle’s micro USB. Note: Please verify that this is available for your hardware before committing to using this method.

- **Cradle COM1/COM2/COM3**

Users need to connect to the S1F2 Cradle’s wifi first, and then send commands through Cradle’s COM1/COM2/COM3 by setting. Note: Please verify that this is availableyou’re your hardware before committing to using this method.

- **USB Default Mode**

Users can send commands through the device’s USB. Note: Please verify that this is availableyou’re your hardware before committing to using this method.

- **S1U2 COM1/COM2**

Users can send commands through the S1U2 COM1/COM2. (S1U2 only)

- **Socket Client**

Terminal will be the socket client. The parameters “POSServerIP” and “POSServerPort” need to be set first. The terminal will continually to try to connect to the set IP/Port after the app has initialized.

- **SSL Server**

The terminal must contain certificate files, and parameters “POSSslCrtName” and “POSSslKeyName” need to be set. Other apps can connect to the app by using the IP/Port set on the “Command Service” page.

- **SSL Client**

The terminal must contain certificate files, and the parameter “CAINFOList” needs to be set. The terminal will continually to try to connect to the set IP/Port after the appp has initialized.

- **SSL Mutual Server**

The terminal must contain certificate files, and parameters “POSSslCrtName”, “POSSslKeyName” and “CAINFOList” need to be set. Other apps can connect to the AP by using the IP/Port on set the “Command Service” page.

- **SSL Mutual Client**

The terminal must contain certificate files, and the parameters “POSSslCrtName”, “POSSslKeyName” and “CAINFOList” need to be set. The terminal will continually to try to connect to the set IP/Port after the app has initialized.

2.1.2. Start Service from the 3rd Party Application

The payment service can be triggered by the method below.

```
final String serviceName = "com.castles.attendap.service.MultiAppService";
final String packageName = "com.castles.attendap";
Intent intent = context.getPackageManager().getLaunchIntentForPackage(packageName);
intent.setClassName(packageName,serviceName);
if(!isMyServiceRunning(serviceName)){
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        context.startForegroundService(intent);
    }
    else {
        context.startService(intent);
    }
}
```

Note. Do not call `startForegroundService()` directly in `onCreate()`. That will cause the APP to crash.

2.1.3. Checking Service from the 3rd Party Application

Check Service broadcast is available as shown below from the following CastlesPay app versions onward:
EVO: 1.2.21.11 or later
APACS: 1.2.16.7 or later
Elavon: 1.2.9.9 or later
ServiceDemo:1.2 version or later will contain a demo for this feature.
The 3rd party application can send a broadcast “android.intent.action.service_status”

```
sendBroadcast(new Intent(
        "android.intent.action.check_service"));
```

If service is alive, the service will send response broadcast “android.intent.action.service_status”.
The payment service should create a broadcast receiver to receive this broadcast:

```
BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        //analysis Intent
    }
};
IntentFilter filter = new IntentFilter();
filter.addAction("android.intent.action.service_status");
registerReceiver(broadcastReceiver,filter);
```

The broadcast intent will contain a bundle. Read the bundle and parse the information from the services:

```
@Override
public void onReceive(Context context, Intent intent) {
    Log.d("ServiceReceiver", "receive new broadcast:" +
intent.getAction());
    if
(intent.getAction().equals("android.intent.action.service_status")) {
        Bundle bundle = intent.getExtras();
        String name = bundle.getString("serviceName");
        String status = bundle.getString("serviceStatus");
       String packageName = bundle.getString("servicePackageName");
    }
}
```

The bundle will contain two items of information:

"serviceName" will show the service's APP name(e.g. CastlesPay_EVO)

"serviceStatus" will show the service's status ("processing" if the service is performing a transaction, or "available" if the service ready and idle).

"servicePackageName" will show the service's package name (e.g. com.castles.attendap_evo.service.MultiAppService)

There will be multiple responses if two or more CastlesPay services are available on the device.

2.2. Socket Connetion

When acting as a server, the payment service will start the socket server. The 3rd party application needs to develop the socket client function to connect to the socket server by using below setting. Note that this is a simple IP socket, not a web socket, which has additional protocols on top.
Default Server Ports are listed below. However they are configurable if necessary.

APACS

Type of Application	Server IP	Server Port
Application on the terminal	Local IP (“127.0.0.1”)	“8080”
External POS/ECR application	Assigned IP (check by 2.1.1)	“8080”

EVO

Type of Application	Server IP	Server Port
Application on the terminal	Local IP (“127.0.0.1”)	“9090”
External POS/ECR application	Assigned IP (check by 2.1.1)	“9090”

ELAVON

Type of Application	Server IP	Server Port
Application on the terminal	Local IP (“127.0.0.1”)	“7070”
External POS/ECR application	Assigned IP (check by 2.1.1)	“7070”

Note: Socket server will take around 5s for initialization. The 3rd party application can only connect to the socket server when it’s ready.

2.3. Command Format

The payment socket service uses a JSON format command. All JSON values use a “string” format. A command sample is below.

```
{
    "txnPosTxnId":    "000001",
    "txnType":        "sale",
    "txnAmtBase":     "100.00",
    "txnAmtTip":      "15.00"
}
```

2.4. Communication Protocol

There are some rules for the command packet

- “txnPosTxnId” should be unique per request packet, the value is “000001” ~ “999999”, the response packet will return the same value as sent in the request packet
- Only one request can be sent at a time. If a 2nd request is sent when the 1st one is not finished, the payment application will return a busy status.

The only exception to this is the return2Idle command

2.5. Status Notification

If the payment application has enabled Status Notifications in config, he payment application will send messages to indicate the current status and activity during a transaction.

See the command detail for more information.

3. Command Detail – Socket Server Mode

All commands use the JSON format of key and value. Keys set as Mandatory must appear in messages. Keys set as Optional may or not may not appear in messages. Applications must allow for unexpected key values that may appear in future versions.

3.1. Get Data

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used to identify a response message
txnType	M	“getData”

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“getData”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnHostInfo	O	Array of info for each host, content txnAcquirerName, txnTid and txnMid
infAppVersion	O	The application version
infEmvVersion	O	The EMV (contact) lib version
infEmvclVersion	O	The EMVCL (contactless) lib version
infAndroidVersion	O	The android system FW version
infSecureModuleVersion	O	The secure module FW version
infSN	O	Device serial number

Sample

Transaction	Description

GetData Request	{
	"txnPosTxnId": "000006",
	"txnType": "getData"
	}

GetData Response	{
	"txnHostInfo": [
	{
	"txnAcquirerName": "AIB",
	"txnMid": "010003364",
	"txnTid": "98400255"
	},
	{
	"txnAcquirerName": "AMEX",
	"txnMid": "6085468148",
	"txnTid": "26458666"
	},
	{
	"txnAcquirerName": "JCB",
	"txnMid": "00000000",
	"txnTid": "00000000"
	}
	],
	"infAndroidVersion": "1.3.1.129611",
	"infAppVersion": "1.2.11",
	"infEmvVersion": "VR9B03-20220512",
	"infEmvclVersion": "VR9V03-20220512",
	"infSecureModuleVersion": "10A2540",
	"txnDateTime": "20220623074703",
	"txnPosTxnId": "000007",
	"txnType": "getData"
	}

3.2. Sale

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“sale”
txnAmtBase	M	base amount of transaction, e.g. “100.00”
txnAmtTip	M	tip amount, e.g. “15.00”, if no tip yet, put “0.00”

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“sale”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnMaskedCardNum	M	Masked card number
txnApprovalCode	O	Host approval code (auth code)
txnHostMsg	O	The message from host server
txnStan	O	STAN, system trace audit number
txnAmtTrans	O	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	O	The final tip amount
txnAmtBase	O	The base transaction amount, excluding any tip or cashback amounts
txnAmtCashback	O	The cashback amount
txnCardBrand	O	Credit/debit cards “VISA” “MasterCard” “American Express” “Discover”

		“JCB” “UnionPay”
txnEntryMode	O	“msr” “chip” “ctls” “manual” “barcode”
txnTid	O	Terminal ID provided by host, TID
txnMid	O	Merchant ID provided by host MID
txnCashierId	O	Cashier ID (cashier num)
txnTableId	O	Table ID (table num)
txnDccAmt	O	Converted amount after performing DCC
txnDccRate	O	Converion rate of DCC
txnDccCurrency	O	Target currency of DCC
txnCoupon	O	The Coupon amount
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

Sample

Transaction	Description

Sale Request	{
	"txnPosTxnId": "000001",
	"txnType": "sale",
	"txnAmtBase": "1.00"
	}

Sale Response	{ "txnAmtBase": "1.00", "txnAmtCashback": "0.00", "txnAmtTip": "0.00", "txnAmtTrans": "1.00", "txnApprovalCode": "005551", "txnCardBrand": "MasterCard", "txnDateTime": "20220623072100", "txnEntryMode": "MSR",

	"txnHostMsg": "AUTH CODE:005551", "txnMaskedCardNum": "5413 33** **** 0037 ", "txnMid": "010003364", "txnPosTxnId": "000011", "txnReturnCode": "00000000", "txnStan": "0002", "txnTid": "98400255", "txnType": "sale" }

3.3. Void

There are different conditions for “VOID” transaction for different acquiring hosts. Some hosts place a time limit the “VOID” transaction, which can be performed within 30s of the last transaction.

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“void”
txnStan	M	The STAN of the transaction for void

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“void”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnMaskedCardNum	M	Masked card number
txnApprovalCode	O	Host approval code (auth code)
txnHostMsg	O	The message from host server
txnStan	O	STAN, system trace audit number
txnAmtTrans	O	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	O	The tip amount
txnAmtBase	O	The base transaction amount, any excluding tip or cashback amounts
txnAmtCashback	O	The cashback amount
txnCardBrand	O	Credit/debit cards “VISA”

		“MasterCard” “American Express” “Discover” “JCB” “UnionPay”
txnEntryMode	O	“msr” “chip” “ctls” “manual” “barcode”
txnTid	O	Terminal ID provided by host, TID
txnMid	O	Merchant ID provided by host MID
txnCashierId	O	Cashier ID (cashier num)
txnTableId	O	Table ID (table num)
txnDccAmt	O	Converted amount after performing DCC
txnDccRate	O	Converion rate of DCC
txnDccCurrency	O	Target currency of DCC
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

Sample

Transaction	Description

Void Request	{
	"txnPosTxnId": "000002",
	"txnType": "void",
	"txnStan": "0002"
	}

Void Response	{ "txnAmtBase": "1.00", "txnAmtCashback": "0.00", "txnAmtTip": "0.00", "txnAmtTrans": "1.00", "txnApprovalCode": "005551", "txnCardBrand": "MasterCard",

	"txnDateTime": "20220623072100", "txnEntryMode": "MSR", "txnHostMsg": "AUTH CODE:005551", "txnMaskedCardNum": "5413 33** **** 0037 ", "txnMid": "010003364", "txnPosTxnId": "000002", "txnReturnCode": "E8000000", "txnStan": "0002", "txnTid": "98400255", "txnType": "void" }

3.4. Refund (Return)

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“refund”
txnAmtTrans	M	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“refund”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnMaskedCardNum	M	Masked card number
txnApprovalCode	O	Host approval code (auth code)
txnHostMsg	O	The message from host server
txnStan	O	STAN, system trace audit number
txnAmtTrans	O	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	O	The tip amount
txnAmtBase	O	The base transaction amount, excluding any tip or cashback amounts
txnAmtCashback	O	The cashback amount
txnCardBrand	O	Credit/debit cards “VISA” “MasterCard”

		“American Express” “Discover” “JCB” “UnionPay”
txnEntryMode	O	“msr” “chip” “ctls” “manual” “barcode”
txnTid	O	Terminal ID provided by host, TID
txnMid	O	Merchant ID provided by host MID
txnCashierId	O	Cashier ID (cashier num)
txnTableId	O	Table ID (table num)
txnDccAmt	O	Amount after execute DCC
txnDccRate	O	Convert rate of DCC
txnDccCurrency	O	Target currency of DCC
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

Sample

Transaction	Description

Refund Request	{
	“txnPosTxnId”: “000003”,
	“txnType”: “refund”,
	“txnAmtTrans”: “1.00”
	}

Refund Response	{ "txnAmtBase": "1.00", "txnAmtCashback": "0.00", "txnAmtTip": "0.00", "txnAmtTrans": "1.00", "txnApprovalCode": "", "txnCardBrand": "MasterCard", "txnDateTime": "20220623072350",

	"txnEntryMode": "MSR", "txnHostMsg": "REFUND", "txnMaskedCardNum": "5413 33** **** 0037 ", "txnMid": "010003364", "txnPosTxnId": "000003", "txnReturnCode": "00000000", "txnStan": "0004", "txnTid": "98400255", "txnType": "refund" }

3.5. Settlement

Due to the payment application may connect to multiple payment hosts or acquiring banks, the response of settlement transaction will have multiple layers of JSON string. The example is like below.


```
{
        " txnSettleInfo ": [
          {
            "txnAcquirerName": "AIB",
            "txnApprovalCode": "95",
            "txnDateTime": " 20210827175413",
            "txnMid": "00000000",
            "txnPosTxnId": "000001",
            "txnReturnCode": "00000000",
            "txnStan": "0095",
            "txnTid": "00000000",
            "txnTotalRefundAmt": "0.00",
            "txnTotalRefundCnt": "0",
            "txnTotalSaleAmt": "0.00",
            "txnTotalSaleCnt": "0",
            "txnTotalSettleAmt": "0.00",
            "txnTotalSettleCnt": "0",
            "txnType": "settlement"
          },
          {
            "txnAcquirerName": "AMEX",
            "txnApprovalCode": "96",
            "txnDateTime": " 20210827175413",
            "txnMid": "00000000",
            "txnPosTxnId": "000001",
            "txnReturnCode": "00000000",
            "txnStan": "0096",
            "txnTid": "00000000",
            "txnTotalRefundAmt": "0.00",
            "txnTotalRefundCnt": "0",
            "txnTotalSaleAmt": "0.00",
```

```
            "txnTotalSaleCnt": "0",
            "txnTotalSettleAmt": "0.00",
            "txnTotalSettleCnt": "0",
            "txnType": " settlement "
        }
    ],
    "txnDateTime": " 20210827175413",
    "txnPosTxnId": "000001",
    "txnReturnCode": "00000000",
    "txnStan": "0095",
    "txnType": " settlement "
}
```


Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“settlement”

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“settlement”
txnReturnCode	M	The return code of the transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnSettleInfo	O	Array of settlement info for each host
txnApprovalCode	O	Host approval code (auth code)
txnHostMsg	O	The message from host server
txnStan	O	STAN, system trace audit number

txnTid	O	Terminal ID provided by host, TID
txnMid	O	Merchant ID provided by host MID
txnTotalSaleAmt	O	Total sale amount
txnTotalSaleCnt	O	Total sale count
txnTotalRefundAmt	O	Total refund amount
txnTotalRefundCnt	O	Total refund count
txnTotalSettleAmt	O	Total settle amount
txnTotalSettleCnt	O	Total settle count
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number

Sample:

Transaction	Description

Settlement Request	{
	"txnPosTxnId": "000006",
	"txnType": "settlement"
	}

Settlement Response	{ "txnSettleInfo": [{ "txnAcquirerName": "AIB", "txnApprovalCode": "", "txnDateTime": "20220623073107", "txnHostMsg": "COMPLETED", "txnMid": "010003364", "txnPosTxnId": "000006", "txnReturnCode": "00000000", "txnStan": "0007", "txnTid": "98400255", "txnTotalRefundAmt": "1.00", "txnTotalRefundCnt": "1", "txnTotalSaleAmt": "1.00", "txnTotalSaleCnt": "1", "txnTotalSettleAmt": "2.00", "txnTotalSettleCnt": "0",

```
            "txnType": "settlement"
        },
        {
            "txnAcquirerName": "AMEX",
            "txnDateTime": "20220623073415",
            "txnHostMsg": "Connection failed",
            "txnMid": "6085468148",
            "txnPosTxnId": "000006",
            "txnReturnCode": "E0000007",
            "txnStan": "0008",
            "txnTid": "26458666",
            "txnTotalRefundAmt": "0.00",
            "txnTotalRefundCnt": "0",
            "txnTotalSaleAmt": "0.00",
            "txnTotalSaleCnt": "0",
            "txnTotalSettleAmt": "0.00",
            "txnTotalSettleCnt": "0",
            "txnType": "settlement"
        }
    ],
    "txnDateTime": "20220623073107",
    "txnPosTxnId": "000006",
    "txnReturnCode": "E000000D",
    "txnStan": "0007",
    "txnType": "settlement"
}
```

3.6. Pre-Auth (Auth)

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“preAuth”
txnAmtBase	M	base amount of transaction, e.g. “100.00”

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“preAuth”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnMaskedCardNum	M	Masked card number
txnApprovalCode	O	Host approval code (auth code)
txnHostMsg	O	The message from host server
txnStan	O	STAN, system trace audit number
txnAmtTrans	O	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	O	The tip amount
txnAmtBase	O	The base transaction amount, excluding any tip or cashback amounts
txnAmtCashback	O	The cashback amount
txnCardBrand	O	Credit/debit cards “VISA” “MasterCard” “American Express” “Discover” “JCB”

		“UnionPay”
txnEntryMode	O	“msr” “chip” “ctls” “manual” “barcode”
txnTid	O	Terminal ID provided by host, TID
txnMid	O	Merchant ID provided by host MID
txnCashierId	O	Cashier ID (cashier num)
txnTableId	O	Table ID (table num)
txnDccAmt	O	Amount after execute DCC
txnDccRate	O	Convert rate of DCC
txnDccCurrency	O	Target currency of DCC
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

Sample:

Transaction	Description

PreAuth Request	{
	"txnPosTxnId": "000004",
	"txnType": "preAuth",
	"txnAmtBase": "1.00"
	}

PreAuth Response	{ "txnAmtBase": "1.00", "txnAmtCashback": "0.00", "txnAmtTip": "0.00", "txnAmtTrans": "1.00", "txnApprovalCode": "005552", "txnCardBrand": "MasterCard", "txnDateTime": "20220623072455", "txnEntryMode": "MSR", "txnHostMsg": "AUTH CODE:005552", "txnMaskedCardNum": "5413 33** **** 0037 ",

	"txnMid": "010003364", "txnPosTxnId": "000004", "txnReturnCode": "00000000", "txnStan": "0005", "txnTid": "98400255", "txnType": "preAuth" }

3.7. Auth-Complete (Capture)

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“authComplete”
txnAmtPreAuth	M	The pre-auth amount
txnAmtAuthComplete	M	The final capture amount
txnApprovalCode	M	Host approval code (auth code)

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“authComplete”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnMaskedCardNum	M	Masked card number
txnApprovalCode	O	Host approval code (auth code)
txnHostMsg	O	The message from host server
txnStan	O	STAN, system trace audit number
txnAmtTrans	O	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	O	The tip amount
txnAmtBase	O	The base transaction amount, excluding any tip or cashback amounts
txnAmtCashback	O	The cashback amount
txnCardBrand	O	Credit/debit cards “VISA” “MasterCard”

		“American Express” “Discover” “JCB” “UnionPay”
txnEntryMode	O	“msr” “chip” “ctls” “manual” “barcode”
txnTid	O	Terminal ID provided by host, TID
txnMid	O	Merchant ID provided by host MID
txnCashierId	O	Cashier ID (cashier num)
txnTableId	O	Table ID (table num)
txnDccAmt	O	Amount after execute DCC
txnDccRate	O	Convert rate of DCC
txnDccCurrency	O	Target currency of DCC
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

Sample:

Transaction	Description

AuthComplete Request	{
	"txnPosTxnId": "000005",
	"txnType": "authComplete",
	"txnAmtPreAuth": "1.00",
	"txnAmtAuthComplete": "1.00",
	"txnApprovalCode": "006838"
	}

AuthComplete Response	{ "txnAmtBase": "1.00", "txnAmtCashback": "0.00", "txnAmtTip": "0.00", "txnAmtTrans": "1.00", "txnApprovalCode": "005551",

	"txnCardBrand": "MasterCard", "txnDateTime": "20220623072542", "txnEntryMode": "MSR", "txnMaskedCardNum": "5413 33** **** 0037 ", "txnMid": "010003364", "txnPosTxnId": "000005", "txnReturnCode": "00000000", "txnStan": "0006", "txnTid": "98400255", "txnType": "authComplete" }

3.8. Display

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“display”
txnMsgLine1	O	Line-1 message
txnMsgLine2	O	Line-2 message
…	O	
txnMsgLine8	O	Line-8 message
txnBarcodeFormat	O	“QR_CODE”: display QR code “CODE_39”: display code 39 “CODE_93” : display code 93 “CODE_128” : display code 128 “EAN_8” : display code EAN_8 “EAN_13” : display code EAN_13
txnBarcodeData	O	The data of barcode or QR code to display
txnBarcodeImgOffset	O	The barcode or QR code image display offset by lines The default offset is “2” E.g. “2” means display the image under the msg line 2
txnTimeout	M	The timeout for the display in mini-second, after the timeout, the terminal will return to idle display

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“display”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”

Sample:

Transaction	Description

Display Request	{
	"txnPosTxnId": "000006",
	"txnMsgLine1": "000001",
	"txnMsgLine2": "000002",
	"txnBarcodeFormat": "QR_CODE",
	"txnBarcodeData": "castles",
	"txnBarcodeImgOffset": "2",
	"txnTimeout": "1000",
	"txnType": "display"
	}

Display Response	{
	"txnDateTime": "20220623080918",
	"txnPosTxnId": "000005",
	"txnReturnCode": "00000000",
	"txnType": "display"
	}

3.9. Return to Idle

Return to Idle works in two situations:

1. Display: close Display and return to idle

2. Sale, Void, Refund, Pre-Auth, Auth-Complete (Capture): If the current txn is still in the Swipe/insert/Tap card period, sending Return to Idle can cancel the txn and return to idle mode. The app needs to close the txn socket and create a new socket to send the Return to Idle command.

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“return2Idle”

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“return2Idle”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”

Sample:

Transaction	Description

ReturnToIdle Request	{
	"txnPosTxnId": "000001",
	"txnType": "return2Idle"
	}

ReturnToIdle Response	{
	"txnDateTime": "20220623081342",
	"txnPosTxnId": "000002",
	"txnReturnCode": "00000000",
	"txnType": "return2Idle"
	}

3.10. Scan Barcode

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“scanBarcode”
txnTimeout	M	The timeout for scanning in milliseconds

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“scanBarcode”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnBarcodeData	O	The data of barcode or QR code

3.11. Reprint

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“reprint”
txnStan	M	The STAN of the transaction for reprint receipt

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“reprint”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”

3.12. CashbackRequest

CashbackRequest is a request from the payment application for the merchant to set the cashback value. If the Cashback POS message is enabled ("POSCashbackCheck": "Y") and the payment application finds that the card supports cashback, the payment application will send this message to the 3rd party application and wait for a response up to POSCashbackTimeout seconds.

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used to identify the response message
txnType	M	“CashbackRequest”
txnDateTime	M	“yyyymmddhhmmss”
txnReturnCode	M	"FFFFFFFF" payment application status feedback

Response

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“CashbackResponse”
txnAmtCashback	M	Cashback amount
txnDateTime	M	“yyyymmddhhmmss”

3.13. Status

Status is the payment application status notification sent only if "POSStatusNotify": "verbose" is set in config. There is no associated response message.

Request

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“status”
txnDateTime	M	“yyyymmddhhmmss”
txnStatus	M	Current Status - see below
txnReturnCode	M	"FFFFFFFF" payment application status feedback

The expected txnStatus values are:
CARD_ENTRY_REQUESTED
CARD_INSERTED
CARD_TAPPED
CARD_SWIPED
MANUAL_INPUT
QRCODE_TXN
DONATION_REQUESTED
DONATION_ACCEPTED
DONATION_DECLINED
TIP_REQUESTED
TIP_ACCEPTED
TIP_DECLINED
LOYALTY_IN_PROGRESS
PIN_REQUESTED
PIN_ENTRY_FAIL
PIN_ENTRY_SUCCESS
COMMUNICATE_WITH_HOST

4. Command Detail – Socket Client Mode

The following messages are only supported on particular CastlesPay application versions. Please check if they are supported for your planned installation before trying to use them.

4.1. Device Connect

Request (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“devConnect”
txnDateTime	M	“yyyymmddhhmmss”
infAppVersion	M	The application version
infEmvVersion	M	The EMV (contact) lib version
infEmvclVersion	M	The EMVCL (contactless) lib version
infAndroidVersion	M	The android system FW version
infSecureModuleVersion	M	The secure module FW version
devIpAddrHex	M	Device IP address in HEX format string e.g. 192.168.1.100 is sent as “C0A80164”.
devSerialNum	M	Device serial number
tmsLastReturnCode	M	TMS last return code “” means no update performed
tmsLastDateTime	M	TMS last update date time “” means no update performed

Response (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“devConnect”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail

4.2. Host Disconnect

Request (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“hostDisconnect”

Response (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“hostDisconnect”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail

Socket will be disconnected in 1s after returning the reponse.

4.3. User Decline

Request (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“userDecline”

Response (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“userDecline”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail

4.4. User Cancel

Request (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“userCancel”

Response (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“userCancel”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail

4.5. Host Payment – Sale

Request (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“hostSale”
txnAmtBase	M	base amount of transaction, e.g. “100.00”
txnAmtTip	M	tip amount, e.g. “15.00”, if no tip, please put “0.00”
txnRrn	M	Transaction reference number
txnSupplierId	M	Transaction supplier ID
txnReceiptHeader	M	Transaction receipt header
txnReceiptFooter	M	Transaction receipt footer

Response (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package

txnType	M	“hostSale”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnMaskedCardNum	M	Masked card number
txnApprovalCode	M	Host approval code (auth code)
txnHostMsg	M	The message from host server
txnStan	M	STAN, system trace audit number
txnAmtTrans	M	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	M	The tip amount
txnAmtBase	M	The base transaction amount, excluding any tip or cashback amounts
txnAmtCashback	M	The cashback amount
txnCardBrand	M	Credit/debit cards “VISA” “MasterCard” “American Express” “Discover” “JCB” “UnionPay”
txnEntryMode	M	“msr” “chip” “ctls” “manual” “barcode”
txnTid	M	Terminal ID provided by host, TID
txnMid	M	Merchant ID provided by host MID
txnCashierId	M	Cashier ID (cashier num)
txnTableId	M	Table ID (table num)
txnDccAmt	M	Converted amount after performing DCC
txnDccRate	M	Converion rate of DCC
txnDccCurrency	M	Target currency of DCC

txnCoupon	M	The coupon amount
txnEmvTagList	M	The transaction related EMV TLV tags, e.g. “5F2A0208405F3401009F0608A000000025010403…” please check Appendix-C for detail
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

4.6. Host Payment – Void

To cancel/void the previous transaction.

Request (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“hostVoid”

Response (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“hostVoid”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

4.7. Host Payment – Refund

Request (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“hostRefund”
txnAmtTrans	M	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnRrn	M	Transaction reference number
txnSupplierId	M	Transaction supplier ID
txnReceiptHeader	M	Transaction receipt header
txnReceiptFooter	M	Transaction receipt footer

Response (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“hostRefund”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail
txnDateTime	M	“yyyymmddhhmmss”
txnMaskedCardNum	M	Masked card number
txnApprovalCode	M	Host approval code (auth code)
txnHostMsg	M	The message from host server
txnStan	M	STAN, system trace audit number
txnAmtTrans	M	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	M	The tip amount
txnAmtBase	M	The base transaction amount, excluding any tip or

		cashback amounts
txnAmtCashback	M	The cashback amount
txnCardBrand	M	Credit/debit cards “VISA” “MasterCard” “American Express” “Discover” “JCB” “UnionPay”
txnEntryMode	M	“msr” “chip” “ctls” “manual” “barcode”
txnTid	M	Terminal ID provided by host, TID
txnMid	M	Merchant ID provided by host MID
txnCashierId	M	Cashier ID (cashier num)
txnTableId	M	Table ID (table num)
txnDccAmt	M	Converted amount after performing DCC
txnDccRate	M	Converion rate of DCC
txnDccCurrency	M	Target currency of DCC
txnCoupon	M	The coupon amount
txnEmvTagList	M	The transaction related EMV TLV tags, e.g. “5F2A0208405F3401009F0608A000000025010403…” please check Appendix-C for detail
txnInvoiceNum	O	Invoice Number
txnRrn	O	Reference Number
txnBatchNum	O	Batch Number
txnAID	O	Application Identifier
txnTC	O	Transaction Certificate

4.8. Host Deferred Auth Get Count

Request (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“hostDAGetCount”

Response (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“hostDAGetCount”
txnDACount	M	Counts of current deferred auth doesn’t send to host
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail

4.9. Host Deferred Auth Upload

Request (Server -> Terminal)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” unique transaction ID used for identify response message
txnType	M	“hostDAUpload”

Response (Terminal -> Server)

Key	M/O	Description
txnPosTxnId	M	"000001" ~ “999999” the same value as request package
txnType	M	“hostDAUpload”
txnReturnCode	M	The return code of transaction result, please ref. appendix B for detail

Appendix A – JSON Key List

All JSON value format is string (ascii) format for easy implementation

The Transaction JSON Key

Key Name	length	Description
txnReturnCode	8	The transaction return code
txnHostMsg	VAR	Host message
txnStan	VAR	STAN, system trace audit number
txnApprovalCode	6	Host approval code (auth code)
txnTotalSaleAmt	VAR	Total sale amount
txnTotalSaleCnt	VAR	Total sale count
txnTotalRefundAmt	VAR	Total refund amount
txnTotalRefundCnt	VAR	Total refund count
txnTotalSettleAmt	VAR	Total settle amount
txnTotalSettleCnt	VAR	Total settle count
txnType	VAR	Transaction type “sale” “preAuth” “authComplete” “refund” “void” “settlement” “getData”
txnAmtTrans	VAR	Final total transaction amount with 2 decimal digits. The value includes tip and cashback amout e.g. “150.00”. Amount is equal to txnAmtBase + txnAmtTip + txnAmtCashback
txnAmtTip	VAR	The tip amount
txnAmtBase	VAR	The base transaction amount, excluding any tip or cashback amounts
txnAmtCashback	VAR	The cashback amount
txnAmtPreAuth	VAR	The pre-auth amount
txnAmtAuthComplete	VAR	The auth-complete amount
txnCardBrand	VAR	Credit/debit cards “VISA”

		“MasterCard” “American Express” “Discover” “JCB” “UnionPay”
txnDateTime	14	“yyyymmddhhmmss”
txnEntryMode	VAR	“msr” “chip” “ctls” “manual” “barcode”
txnTid	8	Terminal ID provided by host, TID
txnMid	15	Merchant ID provided by host MID
txnCashierId	VAR	Cashier ID (cashier num)
txnTableId	VAR	Table ID (table num)
txnPosTxnId	6	POS app transaction unique ID “000001” ~ “999999”
txnSettleInfo	VAR	array of settlement info for each host
txnTimeout	VAR	The timeout in mini-second
txnBarcodeData	VAR	Barcode or QR code data
txnBarcodeFormat	VAR	Barcode or QR code format
txnBarcodeImgOffset	1	Barcode or QR code image offset
txnMsgLine1 txnMsgLine2 ... txnMsgLine8	VAR	The msg to display on different line
txnMaskedCardNum	VAR	Masked PAN
txnDccAmt	VAR	Amount after execute DCC
txnDccRate	VAR	Convert rate of DCC
txnDccCurrency	VAR	Target currency of DCC
txnHostInfo	VAR	Array of info for each host, contents txnAcquirerName, txnTid and txnMid
txnRrn	VAR	Transaction reference number
txnSupplierId	VAR	Supplier ID
txnReceiptHeader	VAR	The receipt header info
txnReceiptFooter	VAR	The receipt footer info
txnEmvTagList	VAR	The transaction related EMV TLV tags,

		e.g. “5F2A0208405F3401009F0608A000000025010403…” please check Appendix-C for detail
tmsLastReturnCode	VAR	TMS last return code “” means no update performed
tmsLastDateTime	VAR	TMS last update date time “” means no update performed
txnInvoiceNum	VAR	Invoice Number
txnBatchNum	VAR	Batch Number
txnAID	VAR	Application Identifier
txnTC	VAR	Transaction Certificate
txnDACount	VAR	Counts of current deferred auth doesn’t send to host
txnStatus	VAR	Current transaction status

The Infomation JSON Key

Key Name	length	Description
infAppVersion	VAR	The application version
infEmvVersion	VAR	The EMV (contact) lib version
infEmvclVersion	VAR	The EMVCL (contactless) lib version
infAndroidVersion	VAR	The android system FW version
infSecureModuleVersion	VAR	The secure module FW version
devIpAddrHex	VAR	Device IP address in HEX format string e.g. 192.168.1.100 is sent as “C0A80164”
devSerialNum	VAR	Device serial number
infSN	VAR	Device serial number

Appendix B – Return Code List

Return Code	Description
00000000	success or approval
E0000000	**TERMINAL ERROR OFFSET**
E0000001	initialization fail (bad config file…etc)
E0000002	invalid parameter or invalid command format (JSON format)
E0000003	unsupported function
E0000004	device busy
E0000005	network error
E0000006	poll card timeout, no card detected (default is 120s)
E0000007	host response timeout, no response from host (default is 60s)
E0000008	user cancel
E0000009	declinded by local (card or terminal), if declined by host, please ref. E8xxxxxx error code
E000000A	read card fail
E000000B	contactless collision (multiple card)
E000000C	transaction not found
E000000D	settlement fail
E000000E	repeated transaction ID (txnPosTxnId)
E000000F	printer out of paper
E0000010	printer error
E0000011	Transaction already voided
E0000012	Transaction declined by card
E0000013	Trasnaction declined by signature failed
E0000014	AVS declined
E0000015	Permission Denied
E0000016	Advice failed
E0000017	Bad MAC
E0000018	Reversal Fail
E0000019	DCC Cancel
E000001A	Barcode Not Created
E000001B	Barcode Insufficient Memory
E000001C	Barcode Already Created

E000001D	Barcode Folder Path Error
E000001E	Barcode Deviceid Error
E000001F	Barcode Devicetype Error
E0000020	Barcode Customerid Error
E0000021	Barcode Deactivation Not Allowed
E0000022	Barcode Curl Failure
E0000023	Barcode Flexera Failure
E0000024	Barcode Invalid Key
E0000025	Barcode Clock Windback Detected
E0000026	Barcode No License Available
E0000027	Barcode Flexera Com
E0000028	Barcode Flexera No Response
E0000029	Barcode Not Activated
E000002A	Please re-try Settle
E000002B	Amount input limit
E000002C	Trasnaction not accepted
E000002D	Payment Service Not Activate
E000002E	Main Service Not Draw Overlay Permission
E000002F	Payment Service Not Draw Overlay Permission
E8000000	**HOST ERROR OFFSET**
FFFFFFFF	*STATUS FEEDBACK from payment application*
....	…

Appendix C – EMV Tag List

The transaction related EMV TLV tags as below,

Currency code (5F2A)

Card Sequence Num (5F34)

CID from 1st GenAC (9F28)

AID (9F06)

Application Cryptogram (9F26)

Cryptogram Information Data (Normal CID) (9F27)

Issuer Application Data (9F10)

Unpredictable Number (9F37)

Application Transaction Counter (9F36)

Ternminal Verification Results (95)

AIP (82)

Terminal Country Code (9F1A)

CVM Results (9F34)

TSI (9B)

IAC Default (9F0D)

IAC Denial (9F0E)

IAC Online (9F0F)

TAC Default (E0)

TAC Denial (E1)

TAC Online (E2)

Reason Online Code (E4)

Application Usage Control (9F07)

Application Version Number (9F09)

Authorisation Response Code (8A)

Terminal Type (9F35)

Terminal Capabilities (9F33)

Contactless Form Factor (9F6E - Mastercard & Visa or 9F67 - Amex)

Contactless Discretionary Data (9F7C - Visa)