T05Pay Terminal

Connection Documentation

Version 1.2

Changelog

Date	Changes
30th June 2021	Version 1.2 Add Pre-Authorization Payment
28th April 2021	Version 1.1 Add Voiding of Latest Transaction
5th March 2020	Changed Deep Link Communication Method
24th Feb 2020	Added Deep Link Communication
7th Feb 2020	Version 1.0 Initial Document

Socket Connection

Setting up the connection

- Set a **static IP address** for the POS (e.g. `192.168.0.230`)

- Ensure that there is internet connection

- Enable **Terminal Mode** on the POS

- Connect to the POS's IP via WebSocket with the **port 8888** (e.g. `ws://192.168.0.230:8888`)

Payment Request

Once T05Pay is switched into **Terminal Mode**, the application will begin to listen for payment requests through the web socket. The payment request needs to be in the following stringified JSON format:

```
{"customOrderId": "47lq5d", "customPrintData": "CustomPrint ????",
"total": 10}
```

customOrderId

A unique identifier provided by the client to identify the transaction.

Type	Required
string	Yes

customPrintData

Additional Text to be printed on the receipt.

Type	Required
string	No

total

Amount to be paid for the payment in cents.

Type	Required
number	Yes

Successful Payment

Once payment is successful on the T05Pay application, a stringified JSON object will be sent back to the client.

```
{status: "Approved", customOrderId: "customOrderId", transactionId:
"transactionId", amountInCents: paymentAmount, paymentType:
"paymentMethod", date: "transactionTime"}
```

- **status** - Status of the Payment.

- **customOrderId** - Identifier sent by the client.

- **transactionId** - 2C2P's unique ID for the transaction.

- **amountInCents** - Amount in cents paid during payment.

- **paymentType** - Type of Payment

- **date** - Timestamp when payment is successful

- **pan** - Masked number of the card used for payment (Applicable only to card payments)

Void Latest Payment

To void the latest payment, the request needs to be in the following stringified JSON format:

```
{"action": "voidLatest"}
```

Error Handling

If an invalid payment request is sent to the application, a response will be sent back to the client in the following stringified JSON object:

```
{status: "fail", message: "message"}
```

Handling Disconnection

If the connection between the T05Pay app and the client is broken, the payment (**A**) will still proceed on. The client should **attempt payment with the same customOrderId**, and the results of payment (**A**) will be returned to the client. This is to prevent duplicate payment requests.

If socket connection is re-established after payment is completed and the client tries to make a payment request again with the same customOrderId. Instead of attempting another payment, the application will query the server for the payment status and immediately responds to the client in the following format:

```
{status: "Approved", customOrderId: "customOrderId", transactionId:
transactionId, amountInCents: paymentAmount, paymentType:
"paymentMethod", date: "transactionTime"}
```

- **status** - Status of the Payment.

- **customOrderId** - Identifier sent by the client.

- **transactionId** - 2C2P's unique ID for the transaction.

- **amountInCents** - Amount in cents paid during payment.

- **paymentType** - Type of Payment

- **date** - Timestamp when payment is successful

- **pan** - Masked number of the card used for payment (Applicable only to card payments)

Deep Link Connection

Reference: Getting a Result from an Activity

Payment Request

To establish a Deep Link connection to the T05Pay Application for payment, call the `startActivityForResult` method with the following `Intent`:

```
Intent intent = new Intent("com.t05.t05pay.PAY");
intent.setDataAndType(null, "text/plain");
```

To send Payment data through the Intent, add a URLEncoded query string as an extra data in the following format:

```
String queryString = URLEncoder.encode("amount=" +
                        amount +
                        "&customOrderId=" +
                        customOrderId +
                        "&printEnabled=" +
                        printEnabled +
                        "&customPrintData=" +
                        customPrintData, "UTF-8");
```

- **amount** - Amount to be paid for the payment in cents.

- **customOrderId** - A unique identifier provided by the client to identify the transaction.

- **printEnabled** - If true, T05Pay Application will print the Payment Receipt (Only applicable if the device allows printing) (Accepted Values: true/false)

- **customPrintData** - Additional Text to be printed on the receipt. (Only applicable if the device allows printing & printEnabled is true)

NOTE: It is recommended for customOrderId or printEnabled to not contain & and = characters as they are special characters used in the query string.

Successful Payment

On a successful payment, the T05Pay Application will return to the client application through `setResult` with the following Intent:

```java
Intent result = new Intent("com.t05.t05pay.PAY_RESULT", null);
```

An extra data named `response` will be in the Intent containing the stringified JSON object response. To extract the string value, use the following in `onActivityResult`:

```java
String response = intent.getStringExtra("response");
```

The stringified JSON object will be in the following format on a **successful payment** along with the result code RESULT_OK:

```
{status: "Approved", customOrderId: customOrderId, transactionId:
transactionId, amountInCents: amountInCents, paymentType: paymentType, date:
transactionTime, pan: pan}
```

- **status** - Status of the Payment.

- **customOrderId** - Identifier sent by the client.

- **transactionId** - 2C2P's unique ID for the transaction.

- **amountInCents** - Amount in cents paid during payment.

- **paymentType** - Type of Payment

- **date** - Timestamp when payment is successful

- **pan** - Masked number of the card used for payment (Applicable only to card payments)

Pre-Authorization Payment Flow

Pre-Authorize Payment

Request

To establish a Deep Link connection to the T05Pay Application for pre-authorization payment, call the `startActivityForResult` method with the following `Intent`:

```
Intent intent = new Intent("com.t05.t05pay.PRE_AUTH");
intent.setDataAndType(null, "text/plain");
```

To send Payment data through the Intent, add a URLEncoded query string as an extra data in the following format:

```
String queryString = URLEncoder.encode("maxAmount=" +
                        Integer.parseInt(maxAmount)
                        "&preAuthText=" +
                        preAuthText, "UTF-8");
```

- **maxAmount** – Amount to be used for the initial pre-authorization

- **preAuthText** – Customized text to be displayed in payment application

Response

On a successful payment, the T05Pay Application will return to the client application through `setResult` with the following Intent:

```
Intent result = new Intent("com.t05.t05pay.PRE_AUTH_RESULT", null);
```

An extra data named `response` will be in the Intent containing the stringified JSON object response. To extract the string value, use the following in `onActivityResult`:

```
String response = intent.getStringExtra("response");
```

The stringified JSON object will be in the following format on a **successful payment** along with the result code RESULT_OK:

```
{status: "Approved", message, sessionId}
```

- **status** – Status of the Payment

- **message** – Custom message that is shown when payment fails

- **sessionId** – Custom ID to be used for capturing of funds

Capture Funds

Request

To establish a Deep Link connection to the T05Pay Application for pre-authorization payment, call the `startActivityForResult` method with the following `Intent`:

```java
Intent intent = new Intent("com.t05.t05pay.POST_AUTH");
intent.setDataAndType(null, "text/plain");
```

To send Payment data through the Intent, add a URLEncoded query string as an extra data in the following format:

```java
String queryString = URLEncoder.encode("amount=" +
                        Integer.parseInt(amount)
                        "&sessionId=" +
                        sessionId, "UTF-8");
```

- **amount** – Amount to be paid for the payment in cents

- **sessionId** – Custom ID to be used for capturing of funds

Response

On a successful payment, the T05Pay Application will return to the client application through `setResult` with the following Intent:

```java
Intent result = new Intent("com.t05.t05pay.POST_AUTH_RESULT", null);
```

An extra data named `response` will be in the Intent containing the stringified JSON object response. To extract the string value, use the following in `onActivityResult`:

```java
String response = intent.getStringExtra("response");
```

The stringified JSON object will be in the following format on a **successful payment** along with the result code RESULT_OK:

```
{status: "Approved", amountInCents, message, sessionId, paymentType, date, pan}
```

- **status** – Status of the Payment.

- **amountInCents** – Amount paid for in cents

- **sessionId** – Custom ID to be used for capturing of funds

- **paymentType** – Type of payment used

- **date** – Date of transaction

- **pan** – Masked pan of card if applicable

- **message** – Custom message that is shown when payment fails

Void Latest Payment

To void the latest transaction, call the `startActivityForResult` method with the following Intent:

```
Intent intent = new Intent("com.t05.t05pay. VOID_LATEST");
intent.setDataAndType(null, "text/plain");
```

Error Handling

If an invalid payment request is sent to the application, the following stringified JSON object will be sent back along with the result code RESULT_CANCELED:

```
{status: "Failed", customOrderId: "customOrderId", message: "message"}
```