

---

# LINE Pay Integration Guide

---

---

# Copyrights

---

Copyright © LINE Corp. All Rights Reserved.

This document is an intellectual property of LINE Corp.; unauthorized reproduction or distribution of this document, or any portion of it is prohibited by law.

This document is provided for information purpose only. LINE Corp. has endeavored to verify the completeness and accuracy of information contained in this document, but it does not take the responsibility for possible errors or omissions in this document. Therefore, the responsibility for the usage of this document or the results of the usage falls entirely upon the user, and LINE Corp. does not make any explicit or implicit guarantee regarding this.

Software products or merchandises mentioned in this document, including relevant URL information, conform to the copyright laws of their respective owners. The user is solely responsible for any results occurred by not complying with applicable laws.

LINE Corp. may modify the details of this document without prior notice.

---

# Document Information

---

## About this document

This Document explains how to develop LINE Pay Merchant's payments using LINE Pay.

## Audience

This document is intended for LINE Pay Merchant server/web developers with thorough understanding of HTTP communications and RESTful API.

## Contact Information

If you have any questions related to this document, or find any errors,

Please, feel free to contact us at:

LINE Pay Technical Support (pay\_tech@linecorp.com)

## Revision History

Version	Date	Changes made
1.0.0	2014.11.10	First Draft
1.0.1	2014.12.01	Add the Details of Preapproved Payment
1.0.2	2015.01.15	Add the Details of Authorization and Capture Process
1.0.3	2015.01.20	Add Error Code
1.0.4	2015.03.01	Add OTK Function, Add ConfirmUrl Browser Setting Function, Add Example Screen for Using Sandbox
1.0.5	2015.06.01	Add LINE Pay Payment Status Page Add oneTimeKey Description Add History of API Changes made Add LINE Pay IP Information in Appendix, QR/Barcode Test Methods
1.0.6	2015.07.30	Define returnCode allowing re-try
1.1.0	2017. 04. 10.	Add the common information of Endpoint and Header for API Change the order of API described in the document
1.1.1	2017. 11. 28.	Correct a wrong example
1.1.2	2018.01.12	Add extra fields in the Request Parameter of the Payment reserve API and modify its example. - extras.addFriends, extras.branchName

---

---

# Convention

---

## Source Code Marking

Text for Source Code is written in black on grey background.

```
COPYDATASTRUCT st;  
st.dwData = PURPLE_OUTBOUND_ENDING;  
st.cbData = sizeof(pp);  
st.lpData = &pp;  
::SendMes(GetTargetHwnd(), WM_COPYDATA, (WPARAM)this->m_hWnd, (LPARAM)&st);
```

---

# Table of Contents

---

<b>LINE Pay Integration Guide</b>	<b>1</b>
<b>Introduction</b>	<b>9</b>
<i>About LINE Pay</i>	9
<i>LINE Pay Merchant Registration Process</i>	10
<i>LINE Pay Features</i>	11
Normal Payment	11
Preapproved Payment	11
Common Features	11
LINE Pay Payment Status	12
<i>LINE Pay Integration Flow</i>	14
Normal Payment	14
Preapproved Payment	20
<b>How to Use LINE Pay APIs</b>	<b>21</b>
<i>Common Features</i>	21
LINE Pay Authentication	21
Request Header	21
Infra and Tech Support	21
<i>Get Payment Details API</i>	22
Get Payment Details Request	22
Get Payment Details API Response	22
Examples of Using Get Payment Details API	23
<i>Reserve Payment API</i>	25
Reserve Payment Request	25
Payment Reserve Response	28
Examples of Using Reserve Payment API	29
<i>Payment Confirm API</i>	30
Payment confirm Request	30
Payment ConfirmAPI Response	30
Examples of Payment confirm API	32
<i>Refund Payment API</i>	34
Refund Payment Request	34
Refund Payment Response	34
Examples of Using Refund Payment API	35
<i>Get Authorization Details API</i>	36
Get Authorization Details Request	36

---

Get Authorization Details Response .....	36
Examples of Using Get Authorization Details API .....	37
<i>Capture API</i> .....	38
Capture Request .....	38
Capture Response .....	38
Examples of Using Capture API .....	40
<i>Void Authorization API</i> .....	41
Void Authorization Request .....	41
Void Authorization Response .....	41
Examples of Using Void Authorization API .....	42
<i>Preapproved Payment API</i> .....	43
Preapproved Payment Request .....	43
Preapproved Payment Response .....	43
Examples of Using Preapproved Payment API .....	45
<i>Check regKey Status API</i> .....	46
Check regKey Status Request .....	46
Check regKey Status Response .....	46
Examples of Using Check regKey Status API .....	47
<i>Expire regKey API</i> .....	48
Expire regKey Request .....	48
Expire regKey Response .....	48
Examples of Expire regKey API .....	48
<b>Appendix</b> .....	<b>49</b>
<i>LINE Pay Error Codes</i> .....	49
<i>How to Use PaymentUrl App</i> .....	54
For Android Applications .....	54
For iOS Applications .....	55
<i>Detailed Explanation and Exceptional Case of ConfirmUrl</i> .....	56
Detailed Explanation of ConfirmUrl .....	56
Exceptional Case of ConfirmUrl .....	56
<i>When Calling confirmUrl from Server to Server</i> .....	57
ConfirmUrl Specification .....	57
Examples of Using confirmUrl .....	57
<i>Using Sandbox Methods</i> .....	59
Sandbox PC Payment .....	59
Sandbox Mobile Payment .....	60
Sandbox OneTimeKey Payment .....	60

---

# Tables and Figures

---

## List of Tables

Table 1 Get Payment Details API Request Parameter	22
Table 2 Get Payment Details API Response Body	22
Table 3 Reserve Payment API Request Parameter	25
Table 4 Payment Reserve Response Body	28
Table 5 Payment confirm API Request Parameter	30
Table 6 Payment confirm API Response Body	30
Table 7 Refund Payment API Request Parameter	34
Table 8 Refund Payment API Response Body	34
Table 9 Get Authorization Details API Request Parameter	36
Table 10 Get Authorization Details API Response Body	36
Table 11 Capture API Request Parameter	38
Table 12 Capture API Response Body	38
Table 13 Void Authorization API Response Body	41
Table 14 Preapproved Payment API Request Parameter	43
Table 15 Preapproved Payment API Response Body	43
Table 16 Check rekey Status API Request Parameter	46
Table 17 Check regKey Status API Response Body	46
Table 18 Expire regKey API Response	48
Table 19 LINE Pay Error Codes	49
Table 20 Error code by API	52
Table 21 confirmUrl Endpoint Specification	57
Table 22 confirmURL Parameters	57

## List of Figures

Figure 1 LINE Pay Payment Process	9
Figure 2 Merchant Registration Process	10
Figure 3 Payment reservation process for PC – Payment screen	14
Figure 4 Payment reservation for mobile - payment screen	15
Figure 5 Payment screen for PC –Payment completion	16
Figure 6 Payment screen for mobile – Payment completion	17
Figure 7 Calling ConfirmUrl from server to server	18

---

Figure 8 OneTimeKeyIssuance and Payment reserve_____	19
Figure 9 Preapproved Payment _____	20
Figure 10 Sample Screen of Sandbox PC Payment Process _____	59
Figure 11 Example Browser of Sandbox Mobile Payment Process _____	60



# Introduction

This chapter explains the basic concept of using LINE Pay.

## About LINE Pay

LINE Pay is a payment system that LINE users can use on LINE Pay Merchant's websites.

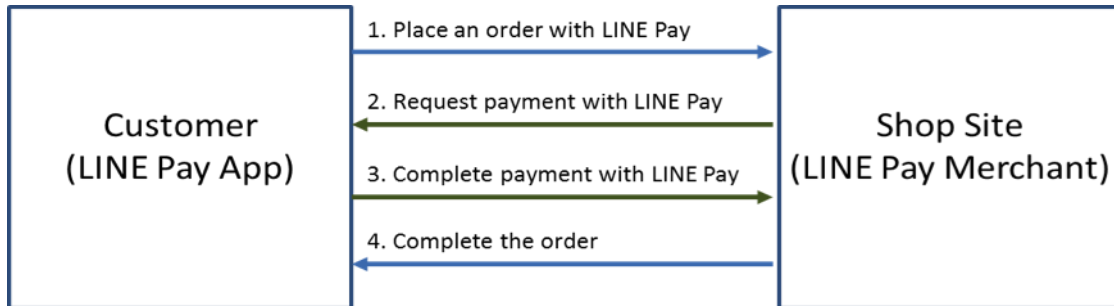


Figure 1 LINE Pay Payment Process

If a merchant is registered as a LINE Pay Merchant, it can attract LINE users around the world as its customers. Also, the merchant's revenue is expected to grow when its marketing channels are expanded through LINE.

For LINE Pay users to make payments using LINE Pay, the website must be registered as a LINE Pay Merchant.

## LINE Pay Merchant Registration Process

Once a merchant is registered as a LINE Pay Merchant, an integration key and test key are issued. Merchant registration process is as follows.

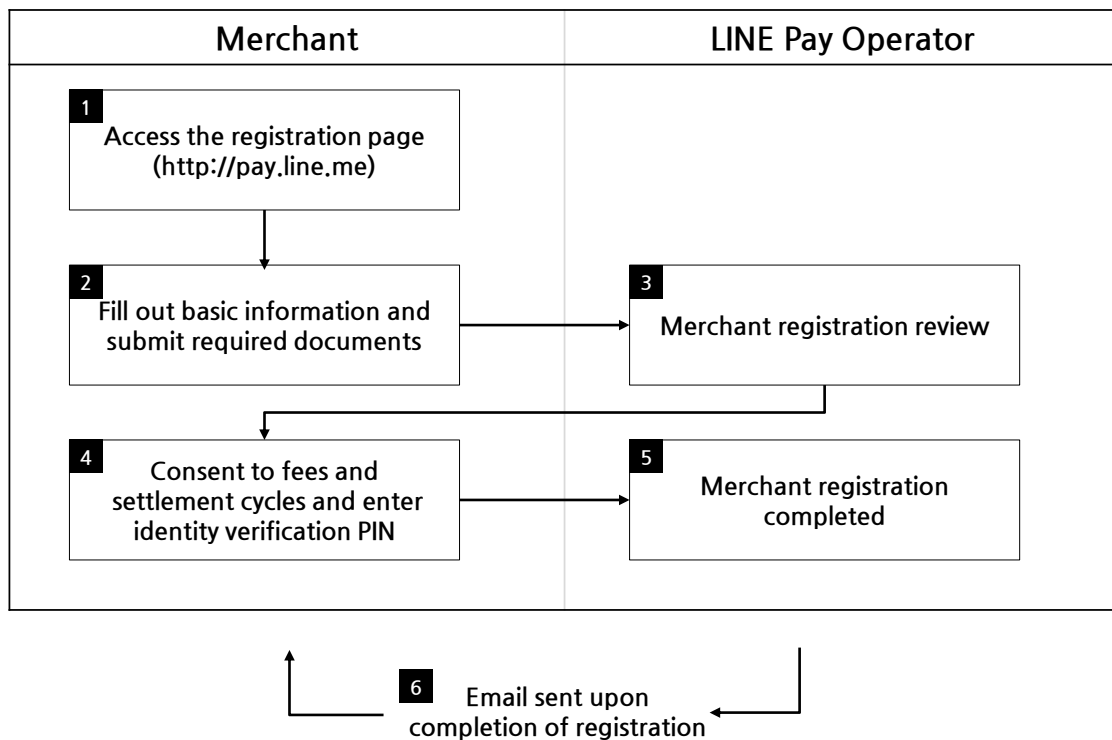


Figure 2 Merchant Registration Process

1. Access the Registration Page (<http://pay.line.me>).
2. Fill out basic information and submit required documents.
3. Merchant registration review
4. Consent to fees and settlement cycles and enter identity verification PIN.
5. Merchant registration completed
6. Email sent upon completion of registration

## LINE Pay Features

The following features are provided.

### Normal Payment

Normal payment is a payment type where a payment is made after LINE Pay users access LINE Pay, select a payment method and enter their payment password. The payment is completed when captured. When Merchants want to separate the authorization of payment from the capture of the authorized payment, they should call "Capture API". Depending on the payment method selected when the password is confirmed, the amount of money to be paid is deducted from the buyers' balance or credit card limit.

1. The Merchant server calls "Reserve Payment API".
2. The buyer selects a payment method and enters the payment password on the LINE Pay payment screen.
3. The Merchant server calls "Confirm Payment API".
4. The payment transaction is completed

### Preapproved Payment

Preapproved payment is made between Merchant server and LINE Pay server, without LINE Pay user intervention. When a LINE Pay user first makes a payment, LINE Pay issues a regKey for preapproved payment, to the Merchant, which goes through payment reservation and confirmation process, like the normal payment. Since then, the Merchant can use the issued regKey to complete following transactions by calling the Preapproved Payment API, without using the LINE app

1. First payment and regKey (for preapproved payment) issuance
  - a. The Merchant server calls the Reserve Payment API ("payType" : "PREAPPROVED")
  - b. The LINE Pay user is redirected to the confirmUrl (a parameter of the Reserve Payment API) after selecting a payment method and entering the payment password on the LINE Pay payment screen.
  - c. The Merchant server calls the Confirm Payment API and saves a regKey returned as response information.
  - d. The payment transaction is completed
2. Preapproved payment
  - a. The Merchant uses the regKey to call the preapproved payment API.
  - b. The payment transaction is completed.
3. Expire regKey Request to expire preapproved payment
  - a. Call API to expire the regKey
  - b. Expire preapproved payment

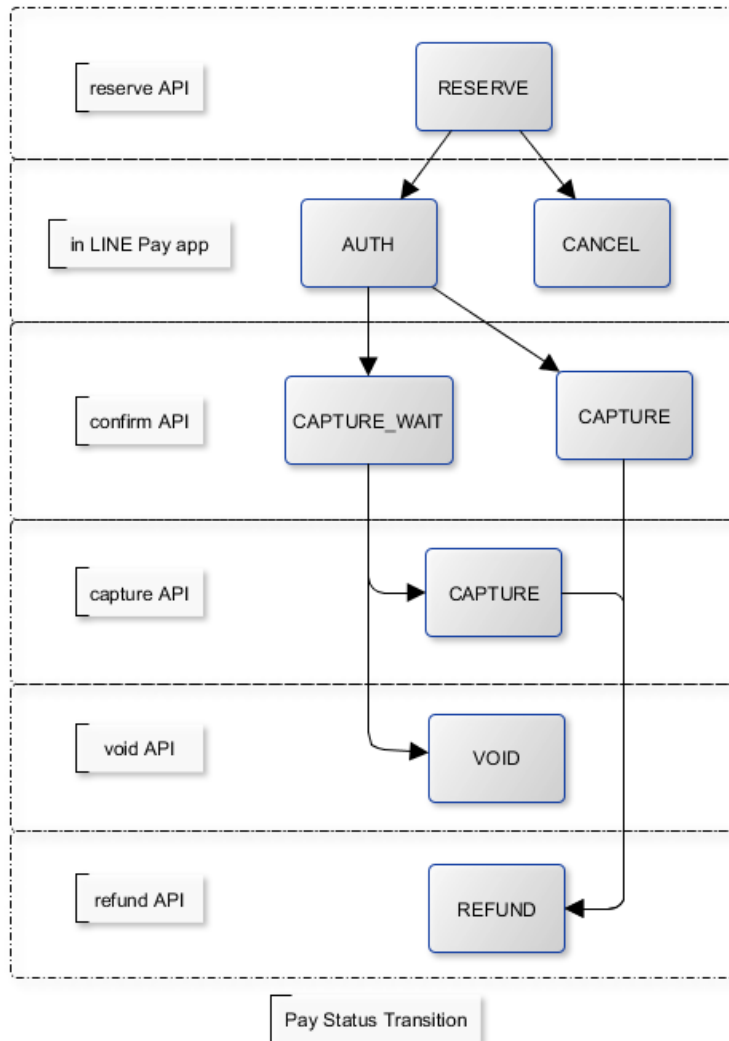
### Common Features

1. Get Payment details

2. Refund Request
3. Get Authorization details
4. Void Authorization Request

## LINE Pay Payment Status

This chapter explains about status from payment request to payment completion.



1. RESERVE : Calling Payment Reserve API, First Time Status
  2. AUTH : Select a Payment Method from LINE Pay app and an Authorization for a Payment Password from LINE Pay app succeeded.
  3. CANCEL : Cancelling a Payment from LINE Pay app
  4. CAPTURE\_WAIT : Capture Status shows Wait (When a "Capture" parameter of Reserve API is delivered as a "False".)
  5. CAPTURE : Capture
  6. VOID : Cancelling a Capture in Wait Status of a Capture.
  7. REFUND : Refund After a Capture
-

FAILED : LINE Pay is on a Failed Status during the processing API; It is occurred from Confirm API, Capture API, and Void API.

## LINE Pay Integration Flow

### Normal Payment

#### Request Payment – Payment Screen

The payment reservation process, which differs depending on LINE Pay users' access environment, is shown below:

##### 1. For PC

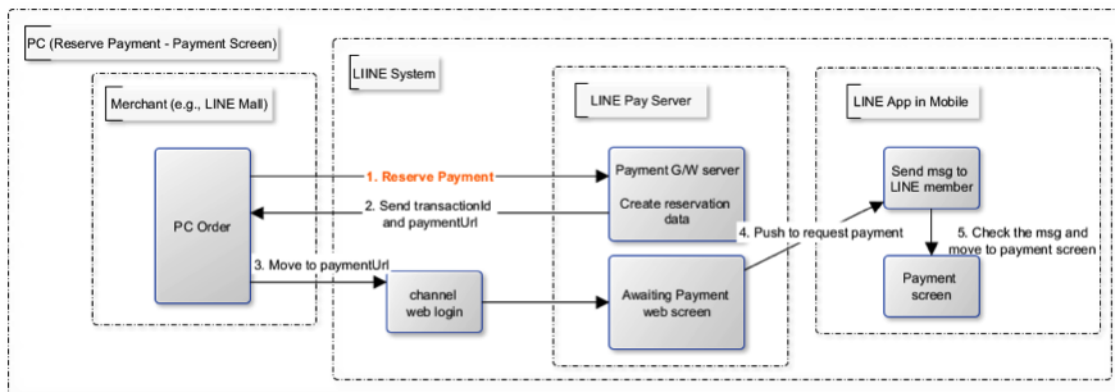


Figure 3 Payment reservation process for PC - Payment screen

1. The LINE Pay user selects a payment method from the Merchant's purchase order.
2. The Merchant calls LINE Pay's "Payment Reserve API" to create payment reservation information and gets a transaction Id, which is required information for payment confirmation and must be kept by the Merchant, and "paymentUrl."
3. The Merchant calls the paymentUrl (provided by LINE Pay) received in step b.
4. The LINE Pay user goes through channel web login<sup>1</sup> that checks if the user is a LINE user.
5. The LINE Pay server pushes a payment request to LINE after the user successfully logs in to LINE.
6. The LINE user gets notified of the payment request on the LINE app and moves to the payment screen.

<sup>1</sup> This is a process of verifying if the user is a LINE user before using LINE services. The channel web login process is skipped if the Merchant can access the LINE user information and forward mid when reserving payment..

## 2. For Mobile

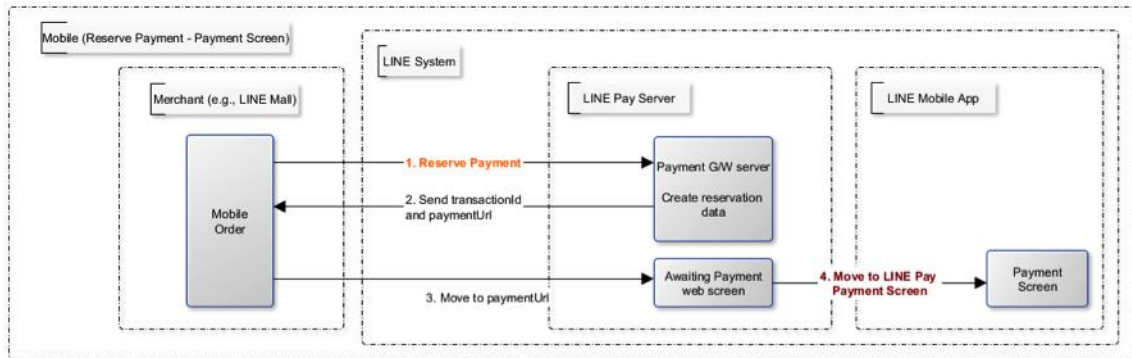


Figure 4 Payment reservation for mobile - payment screen

1. The LINE Pay user selects a payment method from the Merchant's purchase order.
2. The Merchant calls LINE Pay's "Reserve Payment API" to create payment reservation information and gets a transactionId, which is required information for payment confirmation and must be kept by the Merchant, and "paymentUrl."
3. The Merchant calls the paymentUrl (provided by LINE Pay) received in step b.
4. Using the paymentUrl, the Merchant redirects the LINE Pay user to the payment screen in the LINE app. The paymentUrl is divided into web and app, depending on the integration environment.
  - web: A web screen provided by LINE Pay. The Merchant checks whether the LINE app is installed before redirecting the user to the LINE Pay payment screen.
  - app: An app scheme URL to directly move to the LINE Pay payment screen. For how to check whether the LINE app is installed and available versions of LINE Pay (for Android only), please refer to [How to use PaymentUrl App Guide](#).

## Payment Screen - Payment Completion

After selecting a payment method (credit card or balance) to process the payment, the LINE Pay user moves to the "confirmUrl" (added by transactionId in LINE Pay) received from the Merchant when the payment is reserved. Then, the Merchant calls the "Payment confirm API" to complete the payment.

### 1. For PC

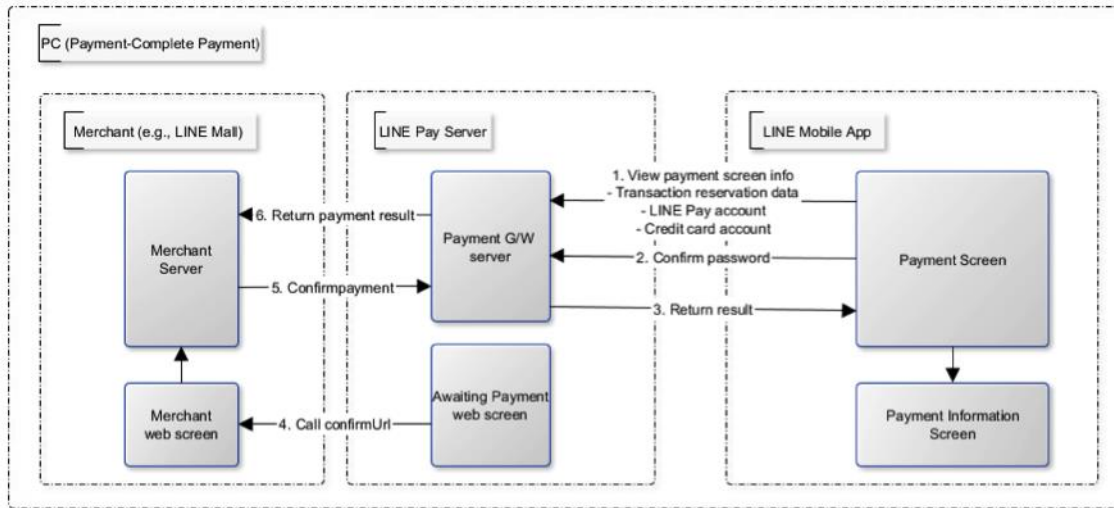


Figure 5 Payment screen for PC -Payment completion

1. The LINE Pay user selects a payment method on the LINE Pay payment screen and enters the password.
2. LINE Pay stores the payment method information and sets the payment status to 'authorized'.
3. The LINE Pay user checks the payment information screen.
4. When the transaction becomes payable on the Awaiting Payment screen, the user is redirected to the "confirmUrl" received from the Merchant when the payment is reserved.
5. The Merchant calls the Confirm Payment API to complete the payment.



## 2. For Mobile

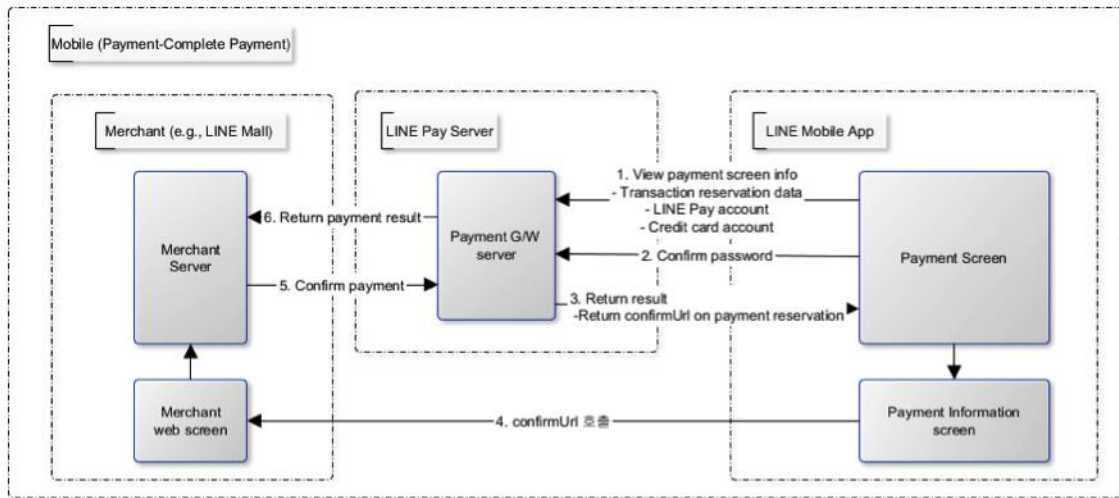


Figure 6 Payment screen for mobile - Payment completion

1. The LINE Pay user selects a payment method on the LINE Pay payment screen and enters the password.
2. LINE Pay stores the payment method information and sets the payment status to 'authorized'.
3. After checking the payment information screen on the LINE app, the LINE Pay user clicks 'OK' on the bottom to move to the "confirmUri" received when the payment is reserved.
4. The Merchant calls the Confirm Payment API to complete the payment.

### When the authorization of payment is separated from the capture of the authorized payment

The overall process is similar to those above, but the Merchant should set "capture" as "false" when calling the Reserve Payment API.

When the Merchant calls the Confirm Payment API, the payment status is saved as AUTHORIZATION.

When capturing the payment,

- To capture the payment: Call the Capture API to complete the payment.
- Not to capture the payment: Call the Void Authorization API to cancel the authorization of payment.

### When Calling confirmUrl from Server to Server

The payment can be made by only the communication between the Merchant server and the LINE Pay server, with delivering confirmUrl.

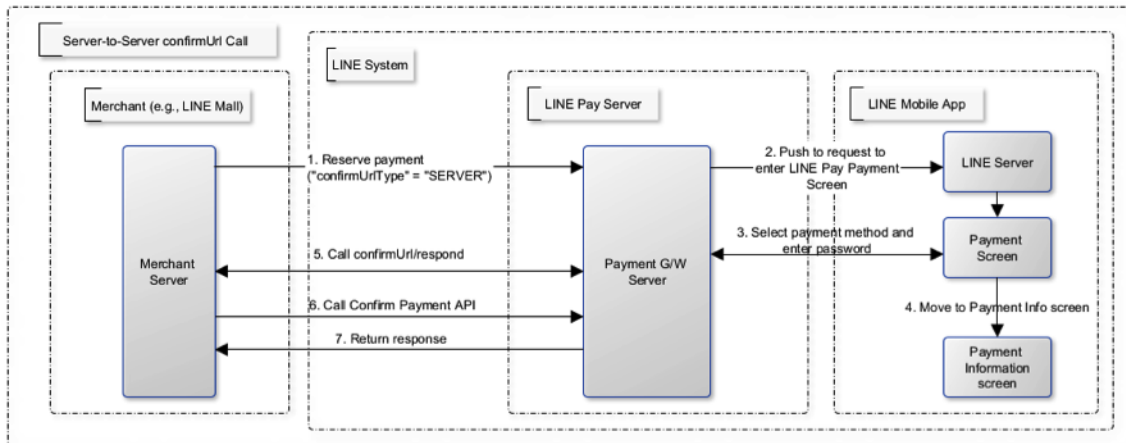


Figure 7 Calling ConfirmUrl from server to server

1. When reserving the payment, the Merchant should pass confirmUrlType : "SERVER".
2. The LINE Pay user selects a payment method and enters the password after entering the LINE Pay payment screen.
3. The LINE Pay server saves the payment information and calls the confirmUrl received from the Merchant when the payment is reserved.  
[\[Appendix\] "Please refer to the case of Calling Confirm Url from Server-to-Server"](#)
4. The Merchant server calls the Confirm Payment API to complete the payment. Please note that the Merchant server can call the Confirm Payment API only after the response for the ConfirmUrl is successfully sent.

### Case of Payment Process using oneTimeKey given by LINE Pay app oneTimeKey

LINE Pay app provides oneTimeKey to support an Offline Payment. oneTimeKey is used as a LINE Pay User's identification verification information and it is verified via Barcode or QR Code. Merchant reads a code, gets oneTimeKey, and processes a Payment.

Prior Conditions

oneTimeKey Valid Time : 10 minutes(From the time that user approaches to the oneTimeKey code page.)

oneTimeKey and Payment reserve will be deleted at the same time

Integration Flow

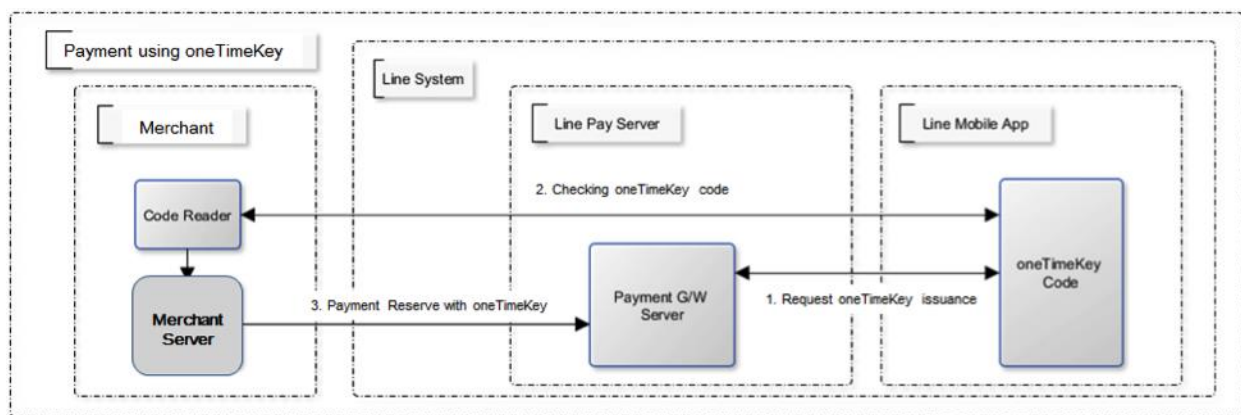


Figure 8 OneTimeKeyIssuance and Payment reserve

1. Merchant reads oneTimeKey code (barcode or QR Code) that LINE Pay user shows.
2. Merchant reserves Payment with oneTimeKey information that Merchant gets.
3. After Payment reserve process completes, LINE Pay user confirms Payment Request via LINE.
4. LINE Pay user selects a Payment method on the Payment Screen and Enters a Password on the same screen.
5. After LINE Pay user checks a payment information screen, For reserving a payment, the flow is different depending on confirmUrlType.
  - a. CLIENT(Default Value) : ConfirmUrl opens on an user's browser. After directing to the browser, calling a payment confirm API; it completes a payment.
  - b. SERVER : [\[Appendix\] Please refer to "When Calling ConfirmURL from Server-to-Server"](#)

Merchant Server calls Payment Confirm API and completes a Payment process. When incorrect response is occurred from ConfirmUrl, Merchant cannot call Confirm API

## Preapproved Payment

### Payment Request - Payment Screen

The overall flow is the same as the normal payment, except that "payType" should be set as "PREAPPROVED."

### Payment Screen - Payment Completion and regKey (for preapproved payment) Issuance

A regKey is delivered with the response information of payment confirmation additionally. The Merchant must save this key to use preapproved payment later.

### Preapproved Payment

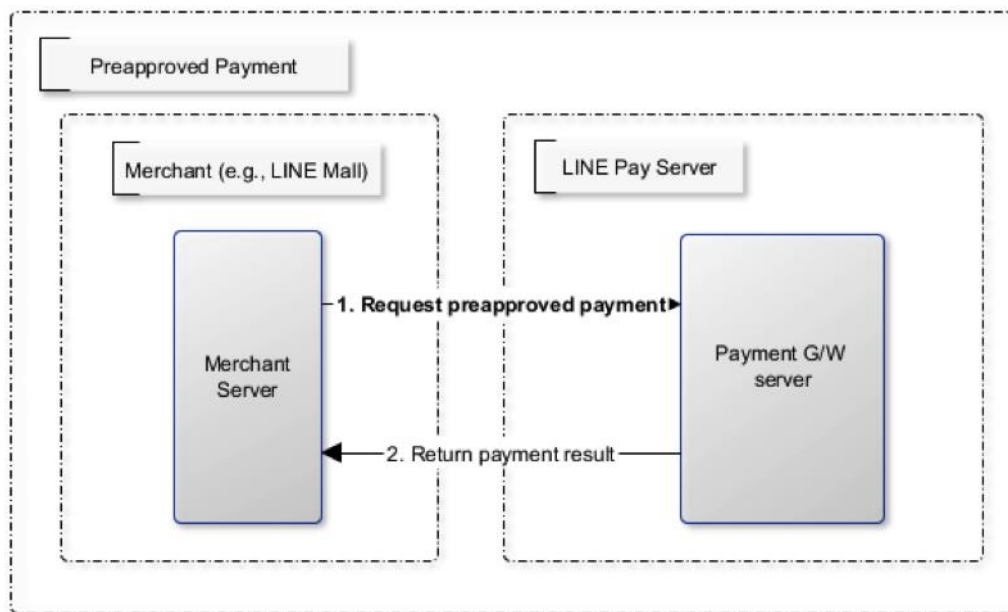


Figure 9 Preapproved Payment

When the payment is confirmed, the Merchant server calls "Preapproved Payment API " by using the regKey to make a payment. The LINE Pay user does not intervene during the payment process, but can be notified when the payment is completed.

### regKey (for Preapproved Payment) Expiration

You can call the Expire regKey API so that unnecessary regKeys expire. For more information, refer to "Expire regKey"

# How to Use LINE Pay APIs

This chapter details how to implement payments using LINE Pay. Before using LINE Pay APIs, Merchant's verification information can be viewed at Merchant Center<sup>2</sup> (<http://pay.line.me>) after evaluation process is complete.

## Common Features

### LINE Pay Authentication

Required authentication information for LINE Pay integration is as below.

- channel id
- channel secret key

Above information is sent by Header script and you can find them in each API specification.

### Request Header

Item	Data Type	Required?	Description
Content-Type	String	Y	application/json
X-LINE-ChannelId	String (10 Bytes)	Y	Payment Integration Information - Channel ID
X-LINE-ChannelSecret	String (32 Bytes)	Y	Payment Integration Information - Channel Secret Key
X-LINE-MerchantDeviceType	String (20 Bytes)	N	Offline Support - Device Type

### Infra and Tech Support

If you have questions for infra condition, technical support or facing internal error, please contact us via email ([pay\\_tech@linecorp.com](mailto:pay_tech@linecorp.com)).

Env.	URL	Description
Beta	<a href="https://lgw-pay.line-apps-beta.com">https://lgw-pay.line-apps-beta.com</a>	Testing environment that is available only in the LINE office network
Sandbox	<a href="https://sandbox-api-pay.line.me">https://sandbox-api-pay.line.me</a>	Environment for integration testing You can process the payment by Sandbox's web simulation payment page instead of LINE Pay app. Please refer to the appendix( <a href="#">Sandbox usage Guide</a> )

---

<sup>2</sup> Merchant Center is a channel in which LINE Pay Merchants can review transaction details and deposits, as well as update basic Merchant information and submit 1:1 inquiries. The current service status of a Merchant can be viewed in the Merchant's basic information.

Env.	URL	Description
Real	<a href="https://api-pay.line.me">https://api-pay.line.me</a>	Real Service Environment

## Get Payment Details API

Gets the details of payments made with LINE Pay. This API only gets the payments that have been captured.

### Get Payment Details Request

**GET /v2/payments**

- Connection Timeout : 5 seconds
- Read Timeout : 20 seconds

**Table 1 Get Payment Details API Request Parameter**

Item	Data Type	Required	Description
transactionId[]	number	N	A transaction ID issued by LINE Pay, for payment or refund.
orderId[]	String	N	Merchant Transaction Order ID

\* When the API is called, more than one parameter needs to be passed. You can check details of up to 100 transactions.

### Get Payment Details API Response

**Table 2 Get Payment Details API Response Body**

Item	Data Type	Description
returnCode	String (4byte)	Result Code <ul style="list-style-type: none"> <li>• 0000: Successful</li> <li>• 1104: Merchant not found.</li> <li>• 1105: This Merchant cannot use LINE Pay.</li> <li>• 1106: Header information error</li> <li>• 1150: Transaction record not found.</li> <li>• 1177: Exceeding max number(100) of transactions allowed</li> <li>• to retrieve</li> <li>• 9000 : Internal error</li> </ul>
returnMessage	String	Result messages or reason for failure
info[].transactionId	number	Transaction number (19 digits)
info[].transactionDate	String	Transaction date & time ( <a href="#">ISO 8601</a> )

Item	Data Type	Description
info[].transactionType	String	Transaction types <ul style="list-style-type: none"> <li>PAYMENT: Payment</li> <li>PAYMENT_REFUND: Refund</li> <li>PARTIAL_REFUND: Partion refund</li> </ul>
info[].productName	String	Merchant's order number
info[].merchantName	String	Merchant Name
info[].currency	String	Currency (ISO 4217)
Info[].authorizationExpiredDate	String	Expiration Date of Authorization ( <a href="#">ISO 8601</a> )
info[].payInfo[].method	String	Payment method used (Credit card: CREDIT_CARD, Balance: BALANCE, Discount: DISCOUNT)
info[].payInfo[].amount	number	Transaction amount (Amount transacted when the transaction Id was generated) The final transaction amount when retrieving the original transaction is $\text{sum}(\text{info}[].\text{payInfo}[].\text{amount}) - \text{sum}(\text{refundList}[].\text{refundAmount}).$

### Retrieving original transactions & When there is a refund

info[].refundList[].refundTransactionId	number	Refunded transaction number (19 digits)
info[].refundList[].transactionType	String	Transaction types <ul style="list-style-type: none"> <li>PAYMENT_REFUND: Refund</li> <li>PARTIAL_REFUND: Partial Refund</li> </ul>
info[].refundList[].refundAmount	number	Refund Amount
info[].refundList[].refundTransactionDate	String	Refunded transaction date & time ( <a href="#">ISO 8601</a> )

### When retrieving a refund

info[].originalTransactionId	Number	Original payment transaction number (19 digits)
------------------------------	--------	---

## Examples of Using Get Payment Details API

### Source 1 Get Payment Details API Request

```
GET https://sandbox-api-pay.line.me/v2/payments?transactionId=20140101123123123&orderId=1002045572 HTTP/1.1

Content-Type: application/json
X-LINE-ChannelId: 1414228945
X-LINE-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
X-LINE-MerchantDeviceProfileId: DEVICE_PROFILE_ID
```

## Source 2 Get Payment Details API Response (When retrieving original transactions)

```
{
  "returnCode": "0000",
  "returnMessage": "success",
  "info": [{
    "transactionId": "1020140728100001997",
    "transactionDate": "2014-07-28T09:48:43Z",
    "transactionType": "PAYMENT",
    "payInfo": [
      { "method": "BALANCE", "amount": 10 },
      { "method": "DISCOUNT", "amount": 10 }
    ],
    "productName": "tes production",
    "currency": "USD",
    "orderId": "20140101123123123",
    "refundList": [
      {
        "refundTransactionId": "20213901923123",
        "transactionType": "PARTIAL_REFUND",
        "refundAmount": -1,
        "refundTransactionDate": "2014-07-28T09:48:52Z"
      },
      {
        "refundTransactionId": "20213901923123",
        "transactionType": "PARTIAL_REFUND",
        "refundAmount": -1,
        "refundTransactionDate": "2014-07-28T09:49:28Z"
      },
      {
        "refundTransactionId": "20213901923123",
        "transactionType": "PARTIAL_REFUND",
        "refundAmount": -1,
        "refundTransactionDate": "2014-07-28T09:53:18Z"
      }
    ]
  }
]}
```

## Source 3 Get Payment Details API Response (When retrieving a refund)

```
{
  "returnCode": "0000",
  "returnMessage": "success",
  "info": [{
    "transactionId": "1020140728100001997",
    "transactionDate": "2014-07-28T09:48:43Z",
    "transactionType": "PARTIAL_REFUND",
    "amount": -5,
    "productName": "",
    "currency": "USD",
    "orderId": "20140101123123123",
    "originalTransactionId": "1020140728100001999"
  }
]}
```



## Reserve Payment API

Prior to processing payments with LINE Pay, the Merchant is evaluated if it is a normal Merchant store then the information is reserved for payment. When a payment is successfully reserved, the Merchant gets a "transaction Id" that is a key value used until the payment is completed or refunded.

### Reserve Payment Request

#### POST /v2/payments/request

- Connection Timeout : 5 seconds
- Read Timeout : 20 seconds

Table 3 Reserve Payment API Request Parameter

Item	Date Type	Required?	Description
productName	String (4000 Bytes)	Y	Product Name (charset: UTF-8)
productImageUrl	String (500 Bytes)	N	Product image URL Image URL to be displayed on the Payment screen <ul style="list-style-type: none"> <li>Size: 84 x 84 (Image to be displayed only on the Payment screen. Recommended to follow the guidelines)</li> </ul>
amount	number <sup>3</sup>	Y	Payment amount
currency	String (3byte)	Y	Payment currency (ISO 4217) Supported currencies are as follows: <ul style="list-style-type: none"> <li>USD</li> <li>JPY</li> <li>TWD</li> <li>THB</li> </ul>
mid	String (50byte)	N	LINE member ID <ul style="list-style-type: none"> <li>LINE user mid for making a payment</li> </ul>
oneTimeKey	String (12byte)	N	One Time Key Result of scanning and reading QR/Bar code information given by LINE Pay app is used as a LINE Pay user's mid .Valid time is 5 minutes and it will be deleted with reserve at the same time. Supports QR/BarCode above LINE 5.1 Version of LINE Pay app
confirmUrl	String	Y	Merchant's URL that the buyer is redirected to after selecting a payment method and entering the payment

<sup>3</sup> A numerical data type where decimal places are permitted.

Item	Date Type	Required?	Description
	(500byte)		<p>password in LINE Pay.</p> <ul style="list-style-type: none"> <li>On the redirected URL, Merchant can call Confirm Payment API and complete the payment</li> <li>LINE Pay passes an additional parameter, "transactionId."</li> </ul>
<div>Reference</div> <div><a href="#">Detailed Explanation and Exceptional Case of ConfirmUrl</a></div>			
confirmUrlType	String	N	<p>confirmUrl Type</p> <p>Type of URL that the buyer is redirected to after selecting a payment method and entering the payment password in LINE Pay</p> <ul style="list-style-type: none"> <li>CLIENT: A user based URL (default)</li> <li>SERVER: A server based URL. Users just need to check the payment information screen in LINE Pay which then notifies the Merchant server that the payment is available.</li> </ul>
checkConfirmUrlBrowser	Boolean	N	<p>When moved to confirmUrl, Check a browser</p> <ul style="list-style-type: none"> <li>true: When a browser calling a payment and a browser directing to confirmUrl are not identical, LINE Pay provides a Guide Page directing to a previous browse.</li> <li>false(Default Value): Directing to ConfirmUrl without checking a browser</li> </ul>
<div>Reference</div> <div><a href="#">Detailed Explanation and Exceptional Case of ConfirmUrl</a></div>			
cancelUrl	String (500byte)	N	<p>Payment Cancellation page URL</p> <ul style="list-style-type: none"> <li>The URL redirected to, from the LINE App Payment page when the LINE Pay user cancels payment (URL for the Merchant accessing via mobile devices to go to the Merchant's app or website when the payment is canceled)</li> <li>URL sent by Merchant is used as is</li> <li>No additional parameters sent by LINE Pay</li> </ul>
packageName	String	N	Information to avoid phishing during transition between

Item	Date Type	Required?	Description
	(4000byte)		apps in Android.
orderId	String (100byte)	Y	Merchant's order number corresponding to the payment request <ul style="list-style-type: none"> <li>A unique number managed by a Merchant</li> </ul>
deliveryPlacePhone	String (100byte)	N	Recipient contact (for Risk Management)
payType	String (12byte)	N	Payment types <ul style="list-style-type: none"> <li>NORMAL: Single payment (Default Value)</li> <li>PREAPPROVED: Preapproved payment</li> </ul>
langCd	String	N	Language codes on the payment waiting screen (paymentUrl). Supports a total of six languages. <ul style="list-style-type: none"> <li>ja: Japanese</li> <li>ko: Korean</li> <li>en: English</li> <li>zh-Hans: Chinese (Simplified)</li> <li>zh-Hant: Chinese (Traditional)</li> <li>th: Thai</li> <li>Language codes are not mandatory but if not received, multiple languages are supported based on the accept-language header.</li> <li>If an unsupported langCd is received, English ("en") is used by default.</li> <li>BCP-47 format: <a href="http://en.wikipedia.org/wiki/IETF_language_tag">http://en.wikipedia.org/wiki/IETF_language_tag</a></li> </ul>
capture	Boolean	N	Whether to capture or not <ul style="list-style-type: none"> <li>true: Payment authorization and capture are handled at once when the Confirm Payment API is called (default).</li> <li>false: A payment is completed only after it is authorized and then separately captured by calling "Capture API", when the Confirm Payment API is called</li> </ul>
<b>Extra fields</b>			
extras.addFriends	object[]	N	Add Friends List <ul style="list-style-type: none"> <li>type: Service type <ul style="list-style-type: none"> <li>"LINE_AT": line@</li> </ul> </li> <li>idList: id list (ID List (ID list registered at the LINE@/OA Management menu on Merchant Center))</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre>"addFriends" : [{   "type" : "LINE_AT",</pre> </div>

Item	Data Type	Required?	Description
			<pre>"idList" : ["@aaa", "@bbb"] }]</pre>
extras.branchName	String (200byte)	N	Branch Name where the payment is requested from (Only 100 letters will be displayed if it's exceeded.)

## Payment Reserve Response

Table 4 Payment Reserve Response Body

Item	Data Type	Description
returnCode	String (4byte)	Result Code <ul style="list-style-type: none"> <li>0000: Successful</li> <li>1104: Merchant not found.</li> <li>1105: This Merchant cannot use LINE Pay.</li> <li>1106: Header information error</li> <li>1124: Error in amount (scale).</li> <li>1133: Invalid oneTimeKey.</li> <li>1145: Payment in progress.</li> <li>1172: Existing same orderId</li> <li>1178: Unsupported currency</li> <li>1194: This Merchant cannot use preapproved payment.</li> <li>2101: Parameter error</li> <li>2102: JSON Data format error</li> <li>9000: Internal error</li> </ul>
returnMessage	String	Result messages or reason for failure. Examples are as follows: <ul style="list-style-type: none"> <li>Merchant unauthorized for payment</li> <li>Merchant verification information error</li> </ul>
info.transactionId	Number	Transaction number (19 digits)
info.paymentUrl.web	String	Web URL to go to after payment request <ul style="list-style-type: none"> <li>Used if payment request was made in Web environment</li> <li>URL to the LINE Pay payment waiting screen</li> <li>Redirected to the provided URL without any additional parameters</li> <li>When a pop-up browser from a Desktop is opened, Size               <ul style="list-style-type: none"> <li>- Width: 700px, Height : 546px</li> </ul> </li> </ul>
info.paymentUrl.app	String.	App URL to the Payment Screen <ul style="list-style-type: none"> <li>Used if payment request was made in an app</li> <li>Redirecting URL from Merchant's app to LINE Pay</li> </ul>

Item	Data Type	Description
		App
		Reference <a href="#">How to Use PaymentUrl App</a>
info.paymentAccessToken	String	Code Value (12 digits) using for input a code instead of using a Scanner on LINE app. (Supports a Scanner of LINE Pay app above LINE 5.1 Version)

## Examples of Using Reserve Payment API

### Source 4 Reserve Payment API Request

```
POST https://sandbox-api-pay.line.me/v2/payments/request HTTP/1.1

Content-Type: application/json
X-LINE-ChannelId: 1414228945
X-LINE-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
X-LINE-MerchantDeviceProfileId: DEVICE_PROFILE_ID

{
  "productName": "test product",
  "productImageUrl": "http://testst.com",
  "amount": 10,
  "currency": "USD",
  "mid": "os89dufgoiw8yer9021384rdfeq",
  "orderId": "20140101123456789",
  "confirmUrl": "naversearchapp://inappbrowser?url=
http%3A%2F%2FtestMall.com%2FcheckResult.nhn%3ForderId%3D20140101123456789",
  "cancelUrl": "naversearchapp://inappbrowser?url=
http%3A%2F%2FtestMall.com%2ForderSheet.nhn%3ForderId%3D20140101123456789",
  "capture": "true",
  "confirmUrlType": "CLIENT",
  "extras": {
    "addFriends": [
      {
        "type": "LINE_AT",
        "idList": ["@aaa", "@bbb"]
      }
    ],
    "branchName": "test_branch_1"
  }
}
```

### Source 5 Reserve Payment API Response

```
{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "transactionId": 123123123123,
    "paymentUrl": {
      "web": "http://web-pay.line.me/web/wait?transactionReserveId=blahblah",
      "app": "line://pay/payment/blahblah"
    },
    "paymentAccessToken": "187568751124"
  }
}
```

## Payment Confirm API

This API is used for a Merchant to complete its payment. The Merchant must call Confirm Payment API to actually complete the payment. However, when "capture" parameter is "false" on payment reservation, the payment status becomes AUTHORIZATION, and the payment is completed only after "Capture API" is called.

### Payment confirm Request

**POST** /v2/payments/{transactionId}/confirm

- Connection Timeout : 5 seconds
- Read Timeout : 40 seconds

Table 5 Payment confirm API Request Parameter

Item	Data Type	Required?	Description
amount	number	Y	Payment amount
currency	String (3byte)	Y	Payment currency (ISO 4217) Supported currencies are as follow: <ul style="list-style-type: none"> <li>• USD</li> <li>• JPY</li> <li>• TWD</li> <li>• THB</li> </ul>

### Payment Confirm API Response

Table 6 Payment confirm API Response Body

Item	Data Type	Description
returnCode	String (4byte)	Result Code <ul style="list-style-type: none"> <li>• 0000: Successful</li> <li>• 1101: The purchasing user suspended for transaction.</li> <li>• 1102: The purchasing user suspended for transaction.</li> <li>• 1104: Merchant not found.</li> <li>• 1105: This Merchant cannot use LINE Pay.</li> <li>• 1106: Header information error</li> <li>• 1110: Not available credit card.</li> <li>• 1124: Error in amount (scale).</li> <li>• 1141: Account status error.</li> <li>• 1142: Insufficient balance remains.</li> <li>• 1150: No transactions found.</li> <li>• 1152: Transaction has already been made.</li> <li>• 1153: Request amount is different from real amount.</li> </ul>

Item	Data Type	Description
		<ul style="list-style-type: none"> <li>1159: Omitted request payment information.</li> <li>1169: Payment method and password must be certificated by LINE Pay.</li> <li>1170: User's account remains have been changed.</li> <li>1172: Existing same orderId.</li> <li>1180: Expired the payment date.</li> <li>1198: Duplicated the request calling API.</li> <li>1199: Internal request error.</li> <li>1280: Temporary error occurred while making payment with credit card</li> <li>1281: Credit card payment error</li> <li>1282: Credit card authorization error</li> <li>1283: The payment has been rejected due to suspected fraud.</li> <li>1284: Credit card payment is temporarily not available.</li> <li>1285: Omitted credit card information</li> <li>1286: Incorrect credit card payment information</li> <li>1287: Credit card expiration date has passed.</li> <li>1288: Credit card has insufficient funds.</li> <li>1289: Maximum credit card limit exceeded</li> <li>1290: One-time payment limit exceeded</li> <li>1291: This card has been reported stolen.</li> <li>1292: This card has been suspended.</li> <li>1293: Invalid Card Verification Number (CVN)</li> <li>1294: This card is blacklisted.</li> <li>1295: Invalid credit card number</li> <li>1296: Invalid amount</li> <li>1298: The Credit card payment declined.</li> <li>9000: Internal error</li> </ul>
returnMessage	String	Result messages or reason for failure. Examples are as follows: <ul style="list-style-type: none"> <li>Merchant unauthorized for payment</li> <li>Merchant Authorization information error</li> </ul>
info.orderId	String	Order number sent from Merchant upon reserving a payment
info.transactionId	number	Transaction number (19 digits) received as a result after reserving a payment
info.authorizationExpirationDate	String	opt-authorization expiration date ( <a href="#">ISO 8601</a> ) <ul style="list-style-type: none"> <li>When the payment status is AUTHORIZATION (capture=false)</li> </ul>
info.payInfo[].method	String	Payment method used (Credit card: CREDIT_CARD, Balance: BALANCE , Discount: DISCOUNT)

Item	Data Type	Description
info.payInfo[].amount	number	Payment amount
info.payInfo[].creditCardNickname	String	For opt-Preapproved Payment, Credit card nickname <ul style="list-style-type: none"> <li>Initially registered credit card name managed by LINE Pay.</li> <li>If LINE Pay user has not set a nickname, blank text is displayed.</li> <li>The nickname can be changed when requested by the user in LINE Pay and changed name is not share with the Merchant.</li> </ul>
info.payInfo[].creditCardBrand	String	For opt-Preapproved Payment, Credit card brand <ul style="list-style-type: none"> <li>VISA</li> <li>MASTER</li> <li>AMEX</li> <li>DINERS</li> <li>JCB</li> </ul>
info.regKey	String	opt-Key for preapproved payment (15 digits)

## Examples of Payment confirm API

### Source 6 Payment Confirm API Request

```

POST https://sandbox-api-pay.line.me/v2/payments/sdhqiwouehrafdasrqoi123as/confirm HTTP/1.1

Content-Type: application/json
X-LINE-ChannelId: 1414228945
X-LINE-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
X-LINE-MerchantDeviceProfileId: DEVICE PROFILE ID

{
  "amount": 1000,
  "currency": "JPY"
}

```

### Source 7 Payment Confirm API Response (Balance payment when the payType is NORMAL)

```

{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "orderId": "order_210124213",
    "transactionId": 20140101123123123,
    "payInfo": [{
      "method": "BALANCE",
      "amount": 10
    }, {
      "method": "DISCOUNT",
      "amount": 10
    }]
  }
}

```



**Source 8 Payment Confirm API Response (Mixed Payment when the payType is NORMAL)**

```
{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "orderId": "order_210124213",
    "transactionId": 20140101123123123,
    "payInfo": [{
      "method": "BALANCE",
      "amount": 10
    }, {
      "method": "POINT",
      "amount": 5
    }]
  }
}
```

**Source 9 Payment Confirm API Response (When the payType is PREAPPROVED)**

```
{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "orderId": "order_210124213",
    "transactionId": 20140101123123123,
    "payInfo": [{
      "method": "CREDIT_CARD",
      "amount": 10,
      "creditCardNickname": "test",
      "creditCardBrand": "VISA"
    }],
    "regKey": "RKf930b19XTf1TX"
  }
}
```

## Refund Payment API

Requests refund of payments made with LINE Pay. To refund a payment, the LINE Pay user's payment transaction Id must be forwarded. A partial refund is also possible depending on the refund amount.

### Refund Payment Request

**POST** /v2/payments/{transactionId}/refund

- Connection Timeout : 5 seconds
- Read Timeout : 20seconds

Table 7 Refund Payment API Request Parameter

Item	Data Type	Required?	Description
refundAmount	number	N	Refund amount - Full refund if this parameter is not passed

### Refund Payment Response

Table 8 Refund Payment API Response Body

Item	Data Type	Description
returnCode	String (4byte)	Result Code <ul style="list-style-type: none"> <li>• 0000: Successful</li> <li>• 1101: The purchasing user suspended for transaction.</li> <li>• 1102: The purchasing user suspended for transaction.</li> <li>• 1104: Merchant not found.</li> <li>• 1105: This Merchant cannot use LINE Pay.</li> <li>• 1106: Header information error</li> <li>• 1124: Payment Amount error</li> <li>• 1150: Transaction record not found.</li> <li>• 1155: The transaction Id not eligible for Refund.</li> <li>• 1163: Exceeded the expiration for Refund.</li> <li>• 1164: Refund limit exceeded.</li> <li>• 1165: The transaction has already been refunded.</li> <li>• 1179: Status can not be processed.</li> <li>• 1198: Duplicated the request calling API.</li> <li>• 1199: Internal request error.</li> <li>• 9000: Internal error</li> </ul>
returnMessage	String	Result messages or reason for failure

Item	Data Type	Description
info.refundTransactionId	number	Refunded transaction number (newly issued number - 19 digits)
info.refundTransactionDate	String	Refunded transaction date & time ( <a href="#">ISO 8601</a> )

## Examples of Using Refund Payment API

### Source 10 Refund Payment API Request

```
POST https://sandbox-api-pay.line.me/v2/payments/sdhqiwouehrafdasrqoi123as/refund HTTP/1.1
Content-Type: application/json
X-LINE-ChannelId: 1414228945
X-LINE-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
X-LINE-MerchantDeviceProfileId: DEVICE_PROFILE_ID

{
  "refundAmount": 500
}
```

### Source 11 Refund Payment API Response (Balance payment when the payType is NORMAL)

```
{
  "returnCode": "0000",
  "returnMessage": "success",
  "info": {
    "refundTransactionId": 123123123123,
    "refundTransactionDate": "2014-01-01T06:17:41Z"
  }
}
```

## Get Authorization Details API

Gets the details authorized with LINE Pay. This API only gets data that is authorized or whose authorization is voided; the one that is already captured can be viewed by using "Get Payment Details API".

### Get Authorization Details Request

#### GET /v2/payments/authorizations

- Connection Timeout : 5 seconds
- Read Timeout : 20 seconds

Table 9 Get Authorization Details API Request Parameter

Item	Data Type	Required?	Description
transactionId[]	number	N	Transaction number issued by LINE Pay
orderId[]	String	N	Order number of Merchant

\* When this API is called, more than one parameter needs to be passed. You can check details of up to 100 transactions.

### Get Authorization Details Response

Table 10 Get Authorization Details API Response Body

Item	Data Type	Description
returnCode	String (4byte)	Results code <ul style="list-style-type: none"> <li>• 0000: Successful</li> <li>• 1104: Merchant not found.</li> <li>• 1105: This Merchant cannot use LINE Pay.</li> <li>• 1106: Header information error</li> <li>• 1150: Transaction record not found.</li> <li>• 1177: Exceeded max number of transactions (100) allowed to be retrieved</li> <li>• 9000: Internal error</li> </ul>
returnMessage	String	Result messages or reason for failure
info[].transactionId	number	Transaction number (19 digits)
info[].transactionDate	String	Transaction date ( <a href="#">ISO 8601</a> )
info[].transactionType	String	Transaction type <ul style="list-style-type: none"> <li>• PAYMENT: Payment</li> <li>• PAYMENT_REFUND: Refund</li> <li>• PARTIAL_REFUND: Partial refund</li> </ul>
info[].payStatus	String	Payment status

Item	Data Type	Description
		<ul style="list-style-type: none"> <li>AUTHORIZATION: Authorized</li> <li>VOIDED_AUTHORIZATION: Authorization voided "Void Authorization API" is called)</li> <li>EXPIRED_AUTHORIZATION: Authorization expired (The authorization period allowed by LINE Pay expired)</li> </ul>
info[].payInfo[].method	String	Payment method (Credit card: CREDIT_CARD, balance: BALANCE, Discount: DISCOUNT)
info[].payInfo[].amount	number	Transaction amount (when a transaction Id is created) The final transaction amount is sum(info[].payInfo[].amount) – sum(refundList[].refundAmount)
info[].productName	String	Product name
info[].currency	String	Currency (ISO 4217)
info[].orderId	String	Order number of Merchant
info[].authorizationExpiredDate	String	Authorization expiration date and time ( <a href="#">ISO 8601</a> )

## Examples of Using Get Authorization Details API

### Source 12 Get Authorization Details API Request

```
GET https://sandbox-api-pay.line.me/v2/payments/authorizations?transactionId=201612312312333401 HTTP/1.1

Content-Type: application/json
X-LINE-ChannelId: 1414228945
X-LINE-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
X-LINE-MerchantDeviceProfileId: DEVICE_PROFILE_ID
```

### Source 13 Get Authorization Details API Response

```
{
  "returnCode": "0000",
  "returnMessage": "success",
  "info": [{
    "transactionId": 201612312312333401,
    "transactionDate": "2014-07-28T09:48:43Z",
    "transactionType": "PAYMENT",
    "payInfo": [{
      "method": "BALANCE",
      "amount": 10
    }, {
      "method": "DISCOUNT",
      "amount": 10
    }],
    "productName": "tes production",
    "currency": "USD",
    "orderId": "20140101123123123",
    "payStatus": "AUTHORIZATION",
    "authorizationExpireDate": "2014-07-28T09:48:43Z"
  }]
}
```

## Capture API

If "capture" is "false" when the Merchant calls the "Reserve Payment API", the payment is completed only after the Capture API is called.

### Capture Request

**POST** /v2/payments/authorizations/{transactionId}/capture

- Connection Timeout : 5 seconds
- Read Timeout : 60 seconds

Table 11 Capture API Request Parameter

Item	Data Type	Required?	Description
amount	number	Y	Payment amount
currency	String (3byte)	Y	Payment currency (ISO 4217) The following currencies are supported. <ul style="list-style-type: none"><li>• USD</li><li>• JPY</li><li>• TWD</li><li>• THB</li></ul>

### Capture Response

Table 12 Capture API Response Body

Item	Data Type	Description
returnCode	String (4byte)	Result code <ul style="list-style-type: none"><li>• 0000: Successful</li><li>• 1104: Merchant not found.</li><li>• 1105: This Merchant cannot use LINE Pay.</li><li>• 1106: Header information error</li><li>• 1150: Transaction record not found.</li><li>• 1155: Wrong TransactionId</li><li>• 1170: User's account remains have been changed.</li><li>• 1172: Existing same orderId.</li><li>• 1179: Status can not be processed.</li><li>• 1180: Expired the payment date.</li><li>• 1183: Payment Amount Error</li><li>• 1184: Payment amount exceeds amount requested.</li></ul>

Item	Data Type	Description
		<ul style="list-style-type: none"> <li>1198: Duplicated the request calling API or Called Capture API during processing internal re-authorization. (Re-try in a few minutes)</li> <li>1199: Internal request error.</li> <li>1280: Temporary error occurred while making a payment with credit card</li> <li>1281: Credit card payment error</li> <li>1282: Credit card authorization error</li> <li>1283: The payment has been declined due to suspected fraud.</li> <li>1284: Credit card payment is temporarily not available.</li> <li>1285: Omitted credit card information</li> <li>1286: Incorrect credit card payment information</li> <li>1287: Credit card expiration date has passed.</li> <li>1288: Credit card has insufficient funds.</li> <li>1289: Maximum credit card limit exceeded.</li> <li>1290: One-time payment limit exceeded.</li> <li>1291: This card has been reported stolen.</li> <li>1292: This card has been suspended.</li> <li>1293: INVALID CARD VERIFICATION NUMBER (CVN)</li> <li>1294: This card is blacklisted.</li> <li>1295: Invalid credit card number</li> <li>1296: Invalid amount</li> <li>1298: The credit card payment declined.</li> <li>9000: Internal error</li> </ul> <p># When an Error code 1199,1280~1298 occurs, the transaction cancels automatically.</p>
returnMessage	String	Result messages or reason for failure. Examples are as follows: <ul style="list-style-type: none"> <li>Merchant unauthorized for payment</li> <li>Merchant verification information error</li> </ul>
info.orderId	String	Order number passed from the Merchant when a payment reservation is made.
info.transactionId	number	Transaction number (19 digits)
info.payInfo[].method	String	Payment method (Credit card: CREDIT_CARD, balance: BALANCE , Discount: DISCOUNT)
info.payInfo[].amount	number	Payment amount

## Examples of Using Capture API

### Source 14 Capture API Request

```
POST https://sandbox-api-pay.line.me/v2/payments/authorizations/sdhqiwouehrafdasrqoi123as/capture HTTP/1.1

Content-Type: application/json
X-LINE-ChannelId: 1414228945
X-LINE-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
X-LINE-MerchantDeviceProfileId: DEVICE_PROFILE_ID

{
  "amount": 1000,
  "currency": "JPY"
}
```

### Source 15 Capture API Response

```
{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "transactionId": 20140101123123123,
    "orderId": "order_210124213",
    "payInfo": [{
      "method": "BALANCE",
      "amount": 10
    }, {
      "method": "DISCOUNT",
      "amount": 10
    }]
  }
}
```



## Void Authorization API

Void a previously authorized payment. A payment that has been already captured can be refunded by using the "Refund Payment API"

### Void Authorization Request

**POST** /v2/payments/authorizations/{transactionId}/void

- Connection Timeout : 5 seconds
- Read Timeout : 20 seconds

### Void Authorization Response

Table 13 Void Authorization API Response Body

Item	Data Type	Description
returnCode	String (4byte)	Result code <ul style="list-style-type: none"> <li>• 0000: Successful</li> <li>• 1101: The purchasing user suspended for transaction.</li> <li>• 1102: The purchasing user suspended for transaction.</li> <li>• 1104: Merchant not found.</li> <li>• 1105: This Merchant cannot use LINE Pay.</li> <li>• 1106: Header information error</li> <li>• 1150: Transaction record not found.</li> <li>• 1155: Wrong TransactionId</li> <li>• 1165: Voided Transaction</li> <li>• 1170: User's account remains have been changed.</li> <li>• 1198: Duplicated the request calling API.</li> <li>• 1199: Internal request error.</li> <li>• 1900: Temporary Error. Please, try again later.</li> <li>• 1903: Temporary Error. Please, try again later.</li> <li>• 1999: It does not match the requested information. (When re-trying a request)</li> <li>• 9000: Internal error</li> </ul> # When an Error code 1900,1903,1999 occurs, allowing re-try.
returnMessage	String	Result messages or reason for failure. Examples are as follows: <ul style="list-style-type: none"> <li>• Merchant unauthorized for payment</li> <li>• Merchant verification information error</li> </ul>

## Examples of Using Void Authorization API

### Source 16 Void Authorization API Request

```
POST https://sandbox-api-pay.line.me/v2/payments/authorizations/sdhqiwouehrafdasrqoi123as/void HTTP/1.1

Content-Type: application/json
X-LINE-ChannelId: 1414228945
X-LINE-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
X-LINE-MerchantDeviceProfileId: DEVICE_PROFILE_ID
```

### Source 17 Void Authorization API Response

```
{
  "returnCode": "0000",
  "returnMessage": "OK"
}
```

## Preapproved Payment API

When the payment type of the Reserve Payment API was set as PREAPPROVED, a regKey is returned with the payment result. Preapproved Payment API uses this regKey to directly complete a payment without using the LINE app.

### Preapproved Payment Request

**POST** /v2/payments/preapprovedPay/{regKey}/payment

- Connection Timeout : 5 seconds
- Read Timeout : 40 seconds

Table 14 Preapproved Payment API Request Parameter

Item	Data Type	Required?	Description
productName	String (4000byte)	Y	Product name
amount	number	Y	Payment amount
currency	String (3byte)	Y	Payment currency (ISO 4217) Supported currencies are as follows: <ul style="list-style-type: none"> <li>• USD</li> <li>• JPY</li> <li>• TWD</li> <li>• THB</li> </ul>
orderId	String (100byte)	Y	Merchant's order number corresponding to the payment request. A unique number managed by a Merchant.
capture	Boolean	N	Whether to be captured or not <ul style="list-style-type: none"> <li>• true(Default Value): Authorization and capture are handled at once.</li> <li>• false: Handled Capture process and Called "Capture API" in order to complete a payment.</li> </ul>

### Preapproved Payment Response

Table 15 Preapproved Payment API Response Body

Item	Data Type	Description

Item	Data Type	Description
returnCode	String (4byte)	<p>Result code</p> <ul style="list-style-type: none"> <li>• 0000: Successful</li> <li>• 1101: The purchasing user suspended for transaction.</li> <li>• 1102: The purchasing user suspended for transaction.</li> <li>• 1104: Merchant not found.</li> <li>• 1105: This Merchant cannot use LINE Pay.</li> <li>• 1106: Header information error</li> <li>• 1110: Not available credit card.</li> <li>• 1124: Error in Amount (scale).</li> <li>• 1141: Account status error.</li> <li>• 1154: Preapproved payment account not available</li> <li>• 1172: Existing a transaction with the same order ID.</li> <li>• 1183: Payment amount must be greater than 0.</li> <li>• 1190: The regKey does not exist.</li> <li>• 1193: The regKey expired.</li> <li>• 1197: Already processing payment with regKey.</li> <li>• 1198: Duplicated the request calling API.</li> <li>• 1199: Internal request error.</li> <li>• 1280: Temporary error while making a payment with credit card</li> <li>• 1281: Credit card payment error</li> <li>• 1282: Credit card authorization error</li> <li>• 1283: The payment has been declined due to suspected fraud.</li> <li>• 1284: Credit card payment is temporarily not available.</li> <li>• 1285: Omitted credit card information</li> <li>• 1286: Incorrect credit card payment information</li> <li>• 1287: Credit card expiration date has passed.</li> <li>• 1288: Credit card has insufficient funds.</li> <li>• 1289: Maximum credit card limit exceeded</li> <li>• 1290: One-time payment limit exceeded</li> <li>• 1291: This card has been reported stolen.</li> <li>• 1292: This card has been suspended.</li> <li>• 1293: INVALID CARD VERIFICATION NUMBER (CVN)</li> <li>• 1294: This card is blacklisted.</li> <li>• 1295: Invalid credit card number</li> <li>• 1296: Invalid amount</li> <li>• 1298: The credit card payment declined.</li> <li>• 9000 : Internal error</li> </ul> <p># When an error code 1281~287, 1290~1294 occurs, expire the regKey</p>

Item	Data Type	Description
returnMessage	String	Result messages or reason for failure
info.transactionId	number	Transaction number (19 digits)
info.transactionDate	String	Transaction date and time ( <a href="#">ISO 8601</a> )
info.authorizationExpire Date	String	Expiration Date of Authorization ( <a href="#">ISO 8601</a> ) <ul style="list-style-type: none"> <li>Deliver for a transaction authorized only. (Capture=false)</li> </ul>

## Examples of Using Preapproved Payment API

### Source 18 Preapproved Payment API Request

```
POST https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/RK123asd213/payment HTTP/1.1
Content-Type: application/json
X-Line-ChannelId: 1414228945
X-Line-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27

{
  "productName": "Naver Music Monthly Pass",
  "amount": 10000,
  "currency": "JPY",
  "orderId": "testOrd2014121200000001"
}
```

### Source 19 Preapproved Payment API Response

```
{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "transactionId": 123123123123,
    "transactionDate": "2014-01-01T06:17:41Z"
  }
}
```

## Check regKey Status API

Checks if regKey is available before using the preapproved payment API.

### Check regKey Status Request

```
GET /v2/payments/preapprovedPay/{regKey}/check
```

- Connection Timeout : 5 seconds
- Read Timeout : 20 seconds

Table 16 Check rekey Status API Request Parameter

Item	Data Type	Required?	Description
creditCardAuth	Boolean	N	<div>Check Authorization for Credit Card minimum amount saved in regKey</div> <ul style="list-style-type: none"><li>• true :<ul style="list-style-type: none"><li>– Check Validity Data within LINE Pay</li><li>– Authorize a Credit Card minimum Amount</li><li>– Allow to activate a function of authorizing a Minimum amount in Merchant Information, it is required an inspection from LINE PAY Administrator.</li></ul></li><li>• false(Default Value) :<ul style="list-style-type: none"><li>– Check Validity data within LINE Pay</li></ul></li></ul>

### Check regKey Status Response

Table 17 Check regKey Status API Response Body

Item	Data Type	Description
returnCode	String(4byte)	<div>Result Code</div> <ul style="list-style-type: none"><li>• 0000: Successful</li><li>• 1101: The purchasing user suspended for transaction.</li><li>• 1102: The purchasing user suspended for transaction.</li><li>• 1104: Merchant not found.</li><li>• 1105: This Merchant cannot use LINE Pay.</li><li>• 1106: Header information error</li><li>• 1141: Can not authorize a credit card saved in regKey</li><li>• 1154: Preapproved payment account not available</li></ul>

Item	Data Type	Description
		<ul style="list-style-type: none"><li>1190: The regKey does not exist.</li><li>1193: The regKey expired.</li></ul>
returnMessage	String	Result messages or reason for failure

### Examples of Using Check regKey Status API

#### Source 20 Check regKey Status API Request

```
GET https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/RK7862119XTF1TT/check
HTTP/1.1

Content-Type: application/json
X-Line-ChannelId: 1414228945
X-Line-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
```

#### Source 21 Check regKey Status API Response

```
{
  "returnCode": "0000",
  "returnMessage": "OK"
}
```

## Expire regKey API

Expires the regKey information registered for preapproved payment. Once the API is called, the regKey is no longer used for preapproved payments.

### Expire regKey Request

```
POST /v2/payments/preapprovedPay/{regKey}/expire
```

- Connection Timeout : 5 seconds
- Read Timeout : 20 seconds

### Expire regKey Response

Table 18 Expire regKey API Response

Item	Data Type	Description
returnCode	String (4byte)	Result code <ul style="list-style-type: none"><li>• 0000: Successful</li><li>• 1104: Merchant not found.</li><li>• 1105: This Merchant cannot use LINE Pay.</li><li>• 1106: Header information error</li><li>• 1190: The regKey does not exist.</li><li>• 1193: The regKey expired.</li><li>• 9000: Internal error</li></ul>
returnMessage	String	Result messages or reason for failure

## Examples of Expire regKey API

### Source 22 Expire regKey API Request

```
POST https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/RK123asd213/expire HTTP/1.1
Content-Type: application/json
X-Line-ChannelId: 1414228945
X-Line-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
```

### Source 23 Expire regKey API Response

```
{
  "returnCode": "0000",
  "returnMessage": "OK"
}
```



# Appendix

## LINE Pay Error Codes

This section defines error codes of LINE Pay. returnMessage for an error code is provided in English, and if there is no message, hyphen (-) is shown.

**Table 19** LINE Pay Error Codes

Code	Description
1101	This user is not a LINE Pay user.
1102	The purchasing user suspended for transaction.
1104	Merchant not found.
1105	This Merchant cannot use LINE Pay.
1106	Header information error
1110	Not available credit card.
1124	Error in Amount (scale).
1133	Invalid oneTimeKey
1141	Account status error.
1142	Insufficient balance remains.
1145	Payment in progress.
1150	Transaction record not found.
1152	Transaction has already been made.
1153	Request amount is different from real amount.
1154	Preapproved payment account not available.
1155	The transaction Id not eligible for Refund.
1159	Omitted request payment information.
1163	Exceeded the expiration for Refund.
1164	Refund limit exceeded.
1165	The transaction has already been refunded.
1169	Payment method and password must be certificated by LINE Pay.
1170	User's account remains have been changed.
1172	Existing same orderId.

Code	Description
1177	Exceeded max. number of transactions (100) allowed to be retrieved.
1178	Unsupported currency.
1179	Status can not be processed.
1180	Expired the payment date
1183	Payment amount must be greater than 0.
1184	Payment amount exceeds amount requested.
1190	The regKey does not exist.
1193	The regKey expired.
1194	This Merchant cannot use Preapproved Payment.
1197	Already processing payment with regKey
1198	Duplicated the request calling API.
1199	Internal request error.
1280	Temporary error while making a payment with Credit Card
1281	Credit Card Payment Error
1282	Credit Card Authorization Error
1283	The payment has been declined due to suspected fraud.
1284	Payment amount must be greater than 0.
1285	Omitted credit card information
1286	Incorrect credit card payment information
1287	Credit card expiration date has passed.
1288	Credit card has insufficient funds.
1289	Maximum credit card limit exceeded.
1290	One-time payment limit exceeded.
1291	This card has been reported stolen.
1292	This card has been suspended.
1293	Invalid Card Verification Number (CVN)
1294	This card is blacklisted.
1295	Invalid credit card number
1296	Invalid amount

Code	Description
1298	The credit card payment declined.
2101	Parameter error
2102	JSON data format error
9000	Internal error

If an internal error, 1199, 9000 occurs, contact Technical Support at: (pay\_tech@linecorp.com).

Table 20 Error code by API

Error Code	reserve	confirm	Capture	Void	Refund	Authorization Details	Payment Details	Preapproved Payment	Check regKey	regKey Expired
1101		v		v	v			v	v	
1102		v		v	v			v	v	
1104	v	v	v	v	v	v	v	v	v	v
1105	v	v	v	v	v	v	v	v	v	v
1106	v	v	v	v	v	v	v	v	v	v
1110		v						v		
1124	v	v			v			v		
1133	v								v	
1141		v						v		
1142		v								
1145	v									
1150			v	v	v	v	v			
1152		v								
1153		v								
1154								v	v	
1155			v	v	v					
1159		v								
1163					v					
1164					v					
1165				v						
1169		v								
1170		v	v	v						
1172	v	v	v					v		
1177						v	v			
1178	v									
1179			v	v	v					
1180		v								
1183	v		v					v		
1184			v							
1190								v	v	v
1193								v	v	
1194	v									
1198		v	v	v	v			v		
1199		v	v	v	v			v		

Error Code	reserve	confirm	Capture	Void	Refund	Authorization Details	Payment Details	Preapproved Payment	Check regKey	regKey Expired
2101	v	v	v	v	v	v	v	v	v	v
2102	v	v	v	v	v	v	v	v	v	v
9000	v	v	v	v	v	v	v	v	v	v

Since the error codes from 1280 to 1298 are occurred from Credit Card company, they not necessarily mentioned in this document.

## How to Use PaymentUrl App

This section provides instructions for redirecting to LINE App after the Merchant's app requests a payment.

### For Android Applications

Use the following example code to check if the LINE app is installed and the available version of LINE Pay. If the LINE app is installed and the available version of LINE Pay is checked, the user is redirected to the LINE Pay payment screen.

```
int linePaySupportedVersion = 230;
String paymentUrl = "..."; // This is "paymentUrl.app" URL String.

Context context = getActivity();
try {
    PackageManager pm = context.getPackageManager();
    PackageInfo packageInfo = pm.getPackageInfo("jp.naver.line.android", 0);
    int versionCode = packageInfo.versionCode;
    if (linePaySupportedVersion <= versionCode) {
        launchUri(paymentUrl);
    } else {
        confirmLineInstall(context);
    }
} catch (NameNotFoundException e) {
    confirmLineInstall(context);
}

private void confirmLineInstall(Context context) {
    new AlertDialog.Builder(context)
        .setTitle("LINE Pay")
        .setMessage(getString(R.String.linepay_confirm))
        .setCancelable(false)
        .setPositiveButton(getString(R.String.linepay_install), new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                launchUri("market://details?id=jp.naver.line.android");
            }
        })
        .setNegativeButton(getString(R.String.linepay_cancel), new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .show();
}

private void launchUri(String uriString) {
    Uri uri = Uri.parse(uriString);
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
}
```

```
File: res/values/Strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    ...
    <String name="linepay_confirm">Supported by Android/iPhone LINE versions 4.8.0
or higher.</String>
    <String name="linepay_install">Get it now</String>
    <String name="linepay_cancel">cancel</String>
    ...
</resources>
```

## For iOS Applications

Use the following example code to check if the LINE app is installed. If the LINE app is installed, the user is redirected to the LINE Pay payment screen.

```
NSString* lineScheme = @"line://";
BOOL installed = [[UIApplication sharedApplication]
                 canOpenURL:[NSURL URLWithString:lineScheme]];
if (installed) {
    UIAlertView *alert =
        [[UIAlertView alloc] initWithTitle:@"LINE Pay"
        message:NSLocalizedString(@"linepay.confirm", nil)
        delegate:self
        cancelButtonTitle:NSLocalizedString(@"linepay.ok", nil) otherButtonTitles:nil];
    alert.tag = 1;
    [alert show];
} else {
    UIAlertView *alert =
        [[UIAlertView alloc] initWithTitle:@"LINE Pay"
        message:NSLocalizedString(@"linepay.confirm", nil)
        delegate:self
        cancelButtonTitle:NSLocalizedString(@"linepay.cancel", nil)
        otherButtonTitles:NSLocalizedString(@"linepay.install",
        nil), nil];
    alert.tag = 2;
    [alert show];
}

- (void)alertView:(UIAlertView*)alertView
clickedButtonAtIndex:(NSInteger)buttonIndex {
    if (alertView.tag == 1 && buttonIndex == 0) {
        NSString *paymentUrl = ...; // This is "paymentUrl.app" URL String.
        [self launchUrl:paymentUrl];
    } else if (alertView.tag == 2 && buttonIndex == 1) {
        [self launchUrl:@"itms-
apps://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware?id=443904275&mt=8"];
    }
}

- (void)launchUrl:(NSString*)urlString {
    NSURL *url = [NSURL URLWithString:urlString];
    [[UIApplication sharedApplication] openURL:url];
}
```

File : en.lproj/Localized.Strings

```
"linepay.confirm" = "Supported by Android/iPhone LINE versions 4.8.0 or higher.";
"linepay.ok" = "OK";
"linepay.cancel" = "Cancel";
"linepay.install" = "Get it now";
```

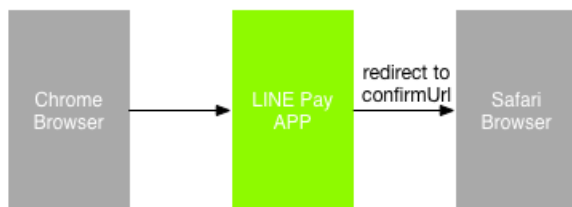
## Detailed Explanation and Exceptional Case of ConfirmUrl

### Detailed Explanation of ConfirmUrl

Once user completes Payment Authorization process (Select a Payment method and confirm a Payment Password) in LINE Pay app, User is redirected to callback Url. The Url does not represent a Payment Process completion. It is the Url for Merchant page allowing to Call confirm API given by LINE Pay and it will be completed a Final Payment Process.

### Exceptional Case of ConfirmUrl

When calling confirmUrl after user gets Payment Authorization from LINE Pay, the Browser requested payment and the Browser receiving ConfirmUrl may not be the same browser depending on the user's type of Operating System (OS).



#### When “checkConfirmUrlBrowser” for Payment reserve API is “true”,

LINE Pay checks out and determines the Browsers whether or not the browser starting Payment process is the same as the browser redirecting automatically according to the result of the following criteria,

- For the Same Browser : Move user to ConfirmUrl
- For Different Browser : Move user to Browser requested Payment

#### When “checkConfirmUrlBrowser” for Payment reserve API is “false (basic value)”,

When the processing already occurred based upon session data(e.g. Log in session) is saved in the browser history of merchant User's payment process can not be continued while calling ConfirmUrl. Therefore, to link to LINE Pay Payment ,Merchant should select the method below and implement it.

1. ConfirmUrl, which already called, does not authorize based upon session data. It is completed a Payment Process based upon Information in confirmUrl
2. Examples of Performing session data Process
  - a. Confirmed session data, Display Browser for Payment Result of Existing Merchant.
  - b. When Session data is unidentified, Merchant selects one of the 2 methods below and implements it..
    - Display Browser for Payment Result After session data is re-produced (e.g. Log in).
    - User is redirected to the Browser requested payment.



## When Calling confirmUrl from Server to Server

This section describes specification and examples of confirmUrl called from the LINE Pay server to Merchant server.

### When a merchant manages ACL(Access Control List)

A merchant should be registered/managed an IP from LINE Pay server. The following IP information is for each environment.

- Sandbox : 182.162.196.200
- Real : 211.249.40.1~211.249.40.30

## ConfirmUrl Specification

Table 21 confirmUrl Endpoint Specification

Item	Description
Protocol	HTTP
Method	GET
Request timeout	Connection: 5 seconds Read: 20 seconds

Table 22 confirmURL Parameters

파라미터	설명
orderId	Order number passed from the Merchant when a payment reservation is made.
transactionId	Transaction number returned as results when a payment reservation is made.

Merchants do not respond to the confirmUrl call, so LINE Pay determines whether it is successful or not by checking HTTP request Code. If the status code is not "200 OK," inform the LINE Pay user that the payment is not successfully completed.

## Examples of Using confirmUrl

Examples of using confirmUrl are as follows, where the confirmUrl provided by the Merchant is 'https://testmall.com/pay/result'.

### Request(LINE Pay -> Merchant)

```
http://testmall.com/pay/result?orderId=2014xxx1232132&transactionId=201408011234567890
```

### Response

```
// When successful
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Accept-Charset: utf-8
Content-Type: text/html
Content-Length: 2
Date: Sat, 11 Oct 2014 02:45:48 GMT

// When failed
HTTP/1.1 404 NOT_FOUND
```

```
Server:Apache-Coyote/1.1  
Accept-Charset:utf-8  
Content-Type:text/html  
Content-Length:4  
Date:Sat, 11 Oct 2014 02:45:48 GMT
```

## Using Sandbox Methods

For testing in the Sandbox of LINE Pay, the section, 'API Endpoint Specification', in this Integration Guide is stated that merchant should attempt to make a payment using sandbox only and merchant should create page for making a payment.

For attempting to make a payment in Sandbox environment, the payment process should be followed as shown below.

※ Requirement

Log-into sandbox is required from all of the channels such PC and mobile.

(To link to Real on the smart phone, User MID received from LINE will be used. This case does not require the channel log-in.)

### Sandbox PC Payment

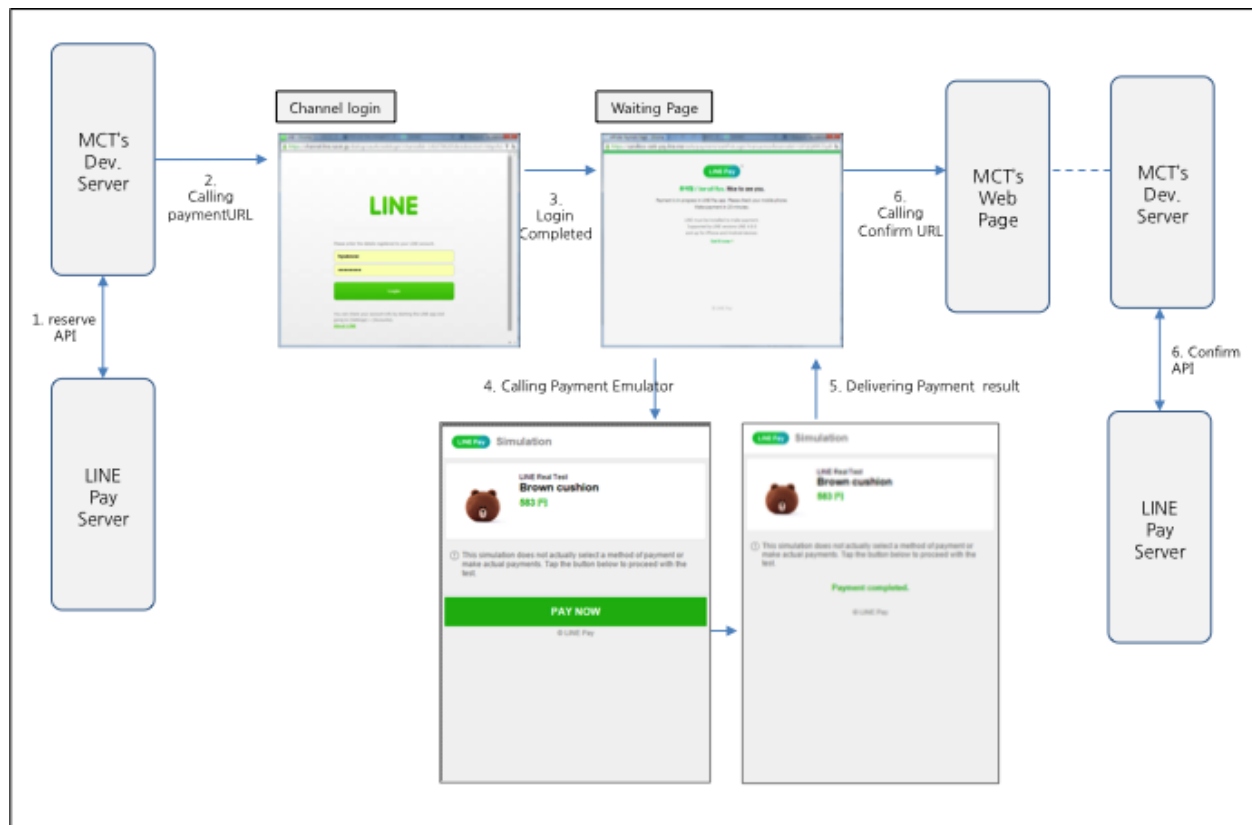


Figure 10 Sample Screen of Sandbox PC Payment Process

## Sandbox Mobile Payment

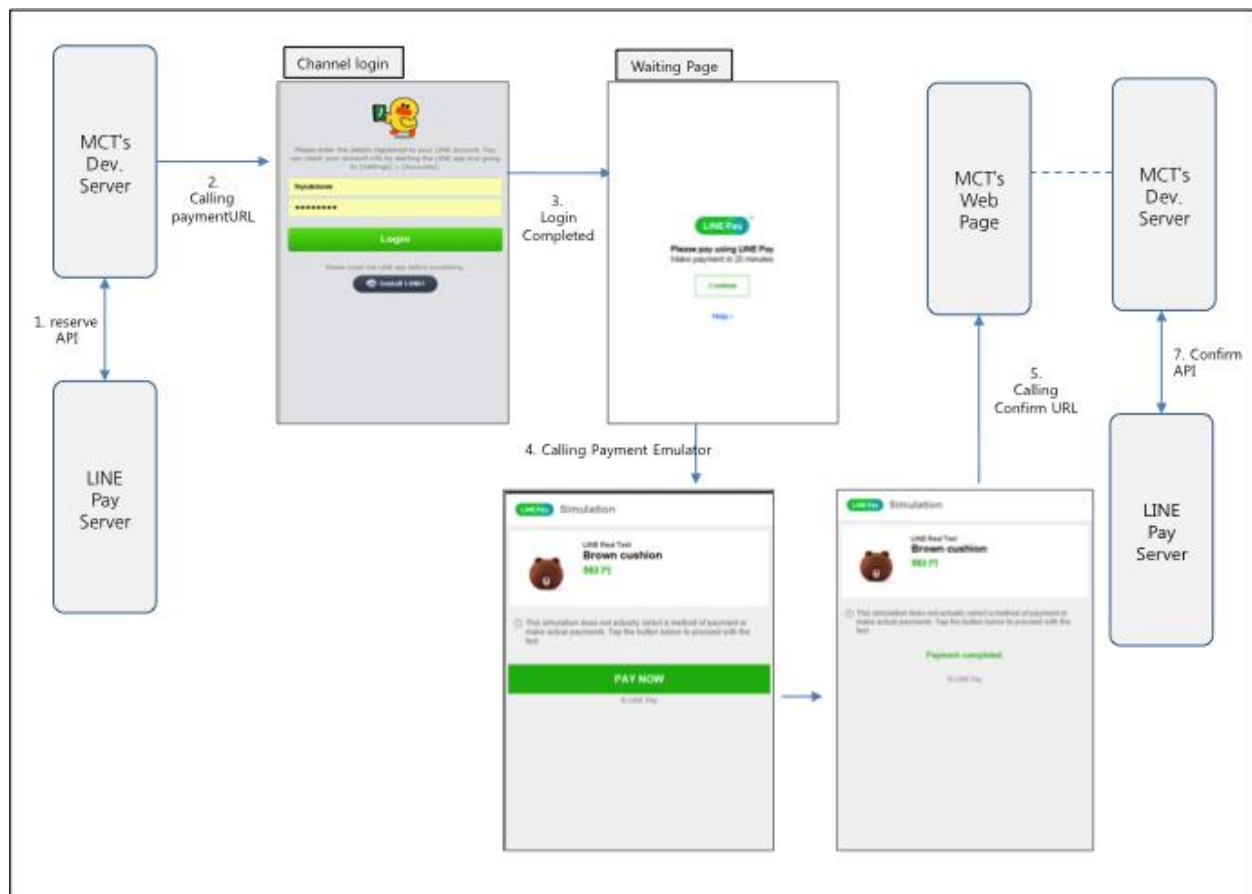


Figure 11 Example Browser of Sandbox Mobile Payment Process

## Sandbox OneTimeKey Payment

A merchant reads "QR/Bar Code", which an user shows from "my code" in LINE Pay app, and processes a payment. Sandbox offers "My Code" via web page it can execute for testing payment. A merchant reads My Code in Sandbox environment and calls Reserve API. Then, My Code Web page move directly to simulation page.

- Sandbox My code url : <https://sandbox-web-pay.line.me/web/sandbox/payment/otk>