
LINE Pay 技術連動ガイド

著作権

Copyright © LINE Corp. All Rights Reserved.

本書は、LINE 株式会社の知的財産であるため、いかなる場合にも LINE 株式会社の許可なく本書の一部または全部をコピー・転送・配布・変更して使用することはできません。

本書は、情報提供の目的に限って提供されます。LINE 株式会社は、本書に収められた情報については完全性と正確性を期しておりますが、記載内容の誤りや漏れによって発生する問題については責任を負いません。

したがって、本書の使用や使用した結果に伴う責任は全面的に利用者であり、LINE 株式会社はこれについて明示的または黙示的にいかなる保証もいたしません。

関連 URL 情報を含め、本書で言及した特定のソフトウェア商品および製品は、当該所有者が属している現地および国内外の関連法に従い、当該法律を遵守しないことにより発生するあらゆる結果に対する責任は全面的に利用者自身にあります。

LINE 株式会社は、本書の内容を予告なく変更する場合があります。

はじめに

本書について

本書では、LINE Pay 加盟店の決済を LINE Pay と連動する方法について説明します。

読者

本書の対象読者は、LINE Pay 加盟店のサーバーサイド開発者です。本書は、HTTP 通信と Restful API について理解していることを前提に記述されています。

連絡先

LINE Pay に関する問い合わせは下記までご連絡ください。

LINEPay 技術サポート(pay_tech@linecorp.com)

改版履歴

バージョン	日付	履歴
1.0.0	2014.11.10	初版作成
1.0.1	2014.12.01	継続決済使用可能有無確認 API を追加 付録追加
1.0.2	2015.01.15	売上確定を分離した仕様を追加
1.0.3	2015.01.20	結果コードを更新
1.0.4	2015.03.01	oneTimeKey 決済仕様の追加 ConfirmURL ブラウザ判定機能の追加 Sandbox 利用時の画面例を追加
1.0.5	2015.06.01	決済プロセスの説明を追加 oneTimeKey 説明を追加 結果コードの追加 API v1 から v2 へのアップデートについて変更履歴を追加 付録に LINE Pay Server の IP 情報、QR/バーコードテスト方法を追加
1.0.6	2015.07.30	リトライ可能 returnCode 新規定義
1.1.0	2017. 04. 10.	API の Endpoint と Header の共通情報の追加 表記の API の順序を変更

バージョン	日付	履歴
1.1.1	2017. 11. 28.	間違えた例を修正
1.1.2	2018.01.12	決済 resarve API のリクエストパラメータフィールドを追加および例を修正 - extras.addFriends, extras.branchName

表記規則

ソースコードの表記

本文書においてソースコードは、グレー背景に黒字で表記します。

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

目次

Copyright © LINE Corp. All Rights Reserved.	2
LINE Payについて	10
LINE Payとは	10
LINE Pay加盟店加入手順	11
LINE Payの機能	12
一般決済	12
継続決済	12
共通機能	12
LINE Pay 決済ステータス遷移	13
LINE Pay連動フロー	14
一般決済	14
継続決済	20
LINE Pay連動APIの使用方法	21
共通機能	21
LINE Pay認証	21
Request Header	21
インフラ・技術サポート.....	21
決済内訳照会API.....	22
決済内訳照会 Request.....	22
決済内訳照会 Response.....	22
決済内訳照会APIの例	23
決済Reserve API	25
決済Reserve Request	25
決済Reserve Response	28
決済reserve APIの例	29
決済Confirm API	31
決済Confirm Request	31
決済Confirm Response	31
決済confirm APIの例	33
払い戻しAPI.....	35
払い戻しRequest	35
払い戻しResponse	35
払い戻しAPIの例	36
オーソリ内訳照会API.....	37
オーソリ内訳照会 Request.....	37

オーソリ内訳照会 Response	37
オーソリ内訳照会APIの例	38
Capture API	40
Capture Request	40
Capture Response	40
capture APIの例	42
オーソリ無効処理 API	43
オーソリ無効処理 Request	43
オーソリ無効処理 Response	43
オーソリ無効処理APIの例	44
継続決済API	45
継続決済Request	45
継続決済Response	45
継続決済APIの例	47
regKey使用可否確認API	48
regKey使用可否確認 Request	48
regKey使用可否確認Response	48
regKey使用可否確認APIの例	49
regKey満了API	50
regKey満了Request	50
regKey満了Response	50
regKey満了APIの例	50
付録	51
LINE Payエラーコード	51
PaymentUrl app使用ガイド	56
Androidアプリのサンプル	56
iPhoneアプリのサンプル	57
confirmUrl 詳細説明及び例外状況への対応	58
confirmUrl 詳細説明	58
confirmUrl 例外状況への対応	58
ConfirmUrlをServer-to-Serverで連携する場合	59
ConfirmUrl詳細	59
ConfirmUrl例	59
Sandbox 利用方法	61
Sandbox PC 決済	61
Sandbox モバイル決済	62
Sandbox OneTimeKey 決済	62

図表一覧

表

表 1 決済内訳照会Request Parameter	22
表 2 決済内訳照会 Response Body	22
表 3 決済Reserve Request Parameter	25
表 4 決済reserve Response Body	28
表 5 決済Confirm Request Parameter	31
表 6 決済confirm Response Body	31
表 7 払い戻しRequest Parameter	35
表 8 払い戻しResponse Body	35
表 9 オートリ内訳照会 Request Parameter	37
表 10 オートリ内訳照会 Response Body	37
表 11 Capture Request Parameter	40
表 12 Capture Response Body	40
表 13 オートリ無効処理 Response Body	43
表 14 継続決済Request Parameter	45
表 15 継続決済Response Body	45
表 16 regKey使用可否確認 Request Parameter	48
表 17 regKey使用可否確認Response Body	48
表 18 regKey満了Response	50
表 19 LINE Payエラーコード	51
表 20 API固有のエラーコード	54
表 21 confirmUrl Endpoint	59
表 22 confirmURLパラメータ	59

図

図 1 LINE Payの決済プロセス	10
図 2 加盟店加入手順	11
図 3 PCからの決済reserve—決済画面	14
図 4 スマートフォン決済reserve—決済画面	15
図 5 PC決済画面—決済完了	16

図 6 スマートフォン決済画面—決済完了	17
図 7 confirmUrlをServer-to-Serverで連携	18
図 8 oneTimeKey発行と決済reserve	19
図 9 継続決済	20
図 10 Sandbox PC 決済画面の例	61
図 11 Sandbox モバイル決済画面の例	62

LINE Pay について

LINE Pay とLINE Pay の連動について基本的な内容を説明します。

LINE Pay とは

LINE Pay とは、LINE の会員が LINE Pay 加盟店のサイトで使用できる決済システムのことをいいます。

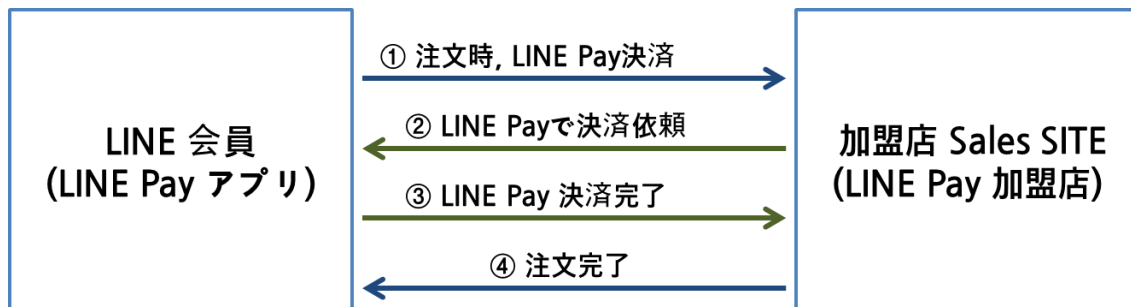


図 1 LINE Pay の決済プロセス

LINE Pay 加盟店に加入すると LINE を使用している世界中の LINE ユーザーたちを加盟店の顧客として誘致できます。また、LINE を通じて加盟店のマーケティングチャンネルが拡大し、売上高の増大も期待できるでしょう。

LINE Pay を使用するユーザーが LINE Pay 決済を行うには、決済するサイトが LINE Pay の加盟店に加入登録されている必要があります。

LINE Pay 加盟店加入手順

LINE Pay 加盟店に加入すると、連動キーとテストキーが発行されます。加盟店加入手順は以下のとおりです。

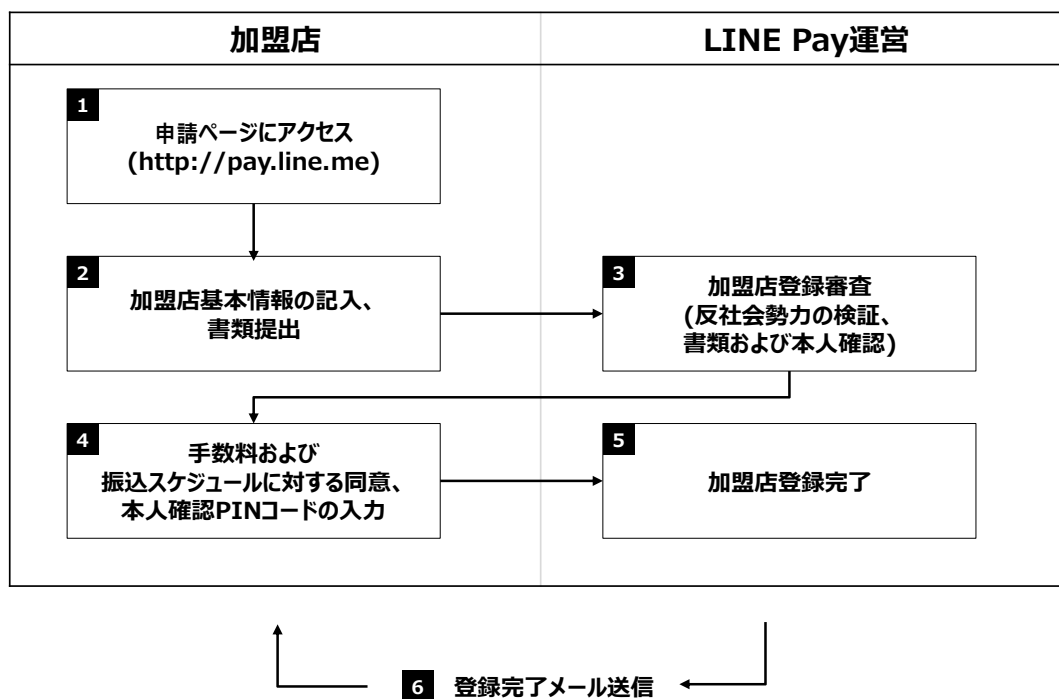


図 2 加盟店加入手順

1. 申請ページ(<http://pay.line.me>)にアクセスする。
2. 基本情報を記入し、必要な書類を提出する。
3. 加盟店の加入審査を行う。
4. 手数料と振込スケジュールに対する同意。
5. 加盟店加入が完了。
6. 登録完了メールが送信される。

LINE Pay の機能

LINE Pay の機能は、下記のように分類されています。

一般決済

一般決済とは、決済をリクエストするたびに LINE Pay 会員が LINE Pay にアクセスし、支払い手段を選択し決済パスワードを確認して取引を行う機能です。このとき、実際の決済完了は、売上(capture)を実行した時点となります。加盟店は売上とオーソリ(authorization)を分ける場合は「capture API」を呼び出す必要があります。「capture API」を実行すると、LINE Pay 会員がオーソリ時に選択した支払い手段により、残高の引落または、クレジットカード売上が確定します。

加盟店サーバーで「決済 reserve API」を呼び出す。

LINE Pay 決済画面で支払い手段を選択し、決済パスワードを確認する。

加盟店サーバーで「決済 confirm API」を呼び出す。

取引完了。

継続決済

継続決済とは、加盟店で決済が必要なタイミングに LINE Pay 会員の操作なく加盟店サーバーと LINE Pay サーバー間で決済を行う機能です。LINE Pay 会員が始めに決済方法を選択し承認すると、LINE Pay 側は加盟店に、次回以降継続決済を可能にする regKey(継続決済キー)を発行します。regKey(継続決済キー)の発行は、一般決済と同様に決済 reserve や決済 confirm のプロセスを実行する必要があります。決済 confirm のレスポンスで regKey が発行され、以降 LINE アプリでの操作なしに、加盟店と LINE Pay 間で継続決済 API のみで取引が行われます。継続決済をご利用頂くためには別途お申込と審査が必要になります。

1. 初めに、regKey(継続決済キー)の発行を行う。
 - a. 加盟店サーバーにて「決済 reserve API」を呼び出す(リクエスト情報のうち、“payType”：“PREAPPROVED”)。
 - b. LINE Pay 決済画面で支払い手段とパスワードを確認後に、confirmUrl(決済 reserve API のパラメータ)へ移動する。
 - c. 加盟店サーバーにて「決済 confirm API」を呼び出し、レスポンス情報の regKey を保存する。
 - d. 取引完了。
2. 継続決済
 - a. regKey で継続決済 API を呼び出す。
 - b. 取引完了。
3. 継続決済を停止するための regKey 満了リクエスト
 - a. regKey 満了リクエスト API の呼び出し
 - b. 継続決済を停止

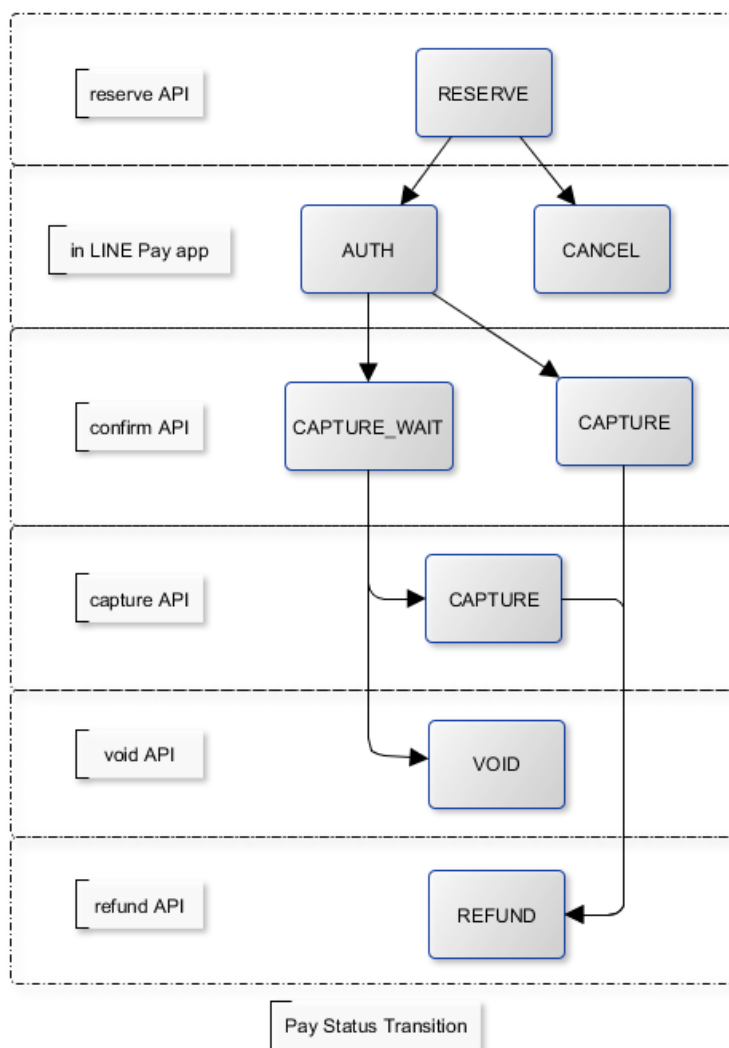
共通機能

1. 決済内訳照会
 2. 払い戻し請求
-

3. オーソリ内訳照会
4. オーソリ取消要求

LINE Pay 決済ステータス遷移

LINE pay の決済ステータスの流れを説明します。



1. RESERVE : 決済 Reserve API 実行
2. AUTH : LINE Pay app で決済手段の選択とパスワード認証
3. CANCEL : LINE Pay app で決済キャンセル
4. CAPTURE_WAIT : 売上確定待ち (reserve API の「capture」パラメータが「false」の場合)
5. CAPTURE : 売上確定
6. VOID : 売上確定待ち状態でのキャンセル
7. REFUND : 売上確定後の払い戻し

FAILED : LINE Pay API の処理中に失敗した状態で confirm、capture、void API で発生

LINE Pay 連動フロー

一般決済

決済依頼—決済画面

決済 reserve のプロセスは、以下のとおりです。LINE Pay 会員の加盟店のアクセス環境(PC、スマートフォン)によって連動フローが異なります。

1. PC の場合

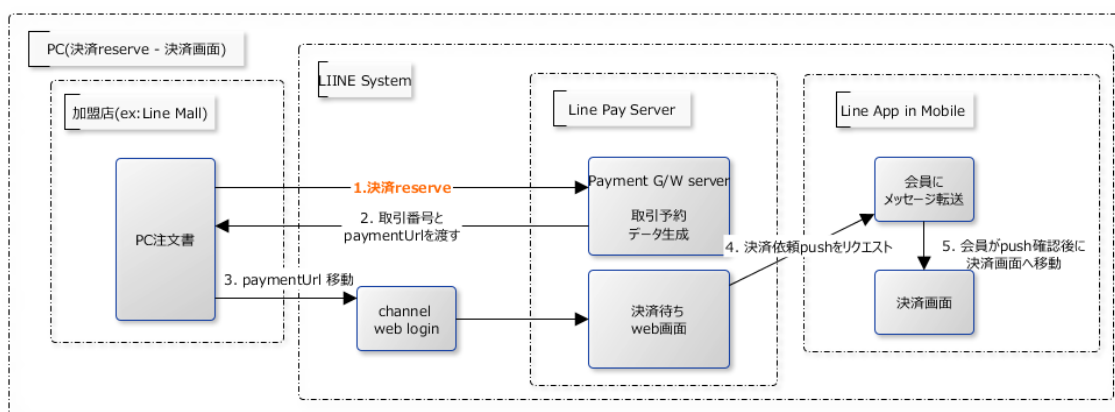


図 3 PC からの決済 reserve—決済画面

1. LINE Pay 会員が加盟店の支払い方法から LINE Pay を選択する。
2. 加盟店では、LINE Pay「決済 reserve API」を実行し、取引番号と“paymentUrl”を受け取る。取引番号は、決済 confirm の際に必ず必要な情報であり、加盟店で保存しておく必要がある。
3. 加盟店は、b で受け取った paymentUrl(LINE pay 提供)を呼び出す。
4. LINE の会員であるかどうかを確認するために、channel web login プロセスを経る。
5. ログインに成功すると、LINE Pay から LINE アプリに決済承認依頼を PUSH 通知する。
6. LINE 会員は、LINE アプリで決済内容を確認し決済画面へ移動する。

2. スマートフォンの場合

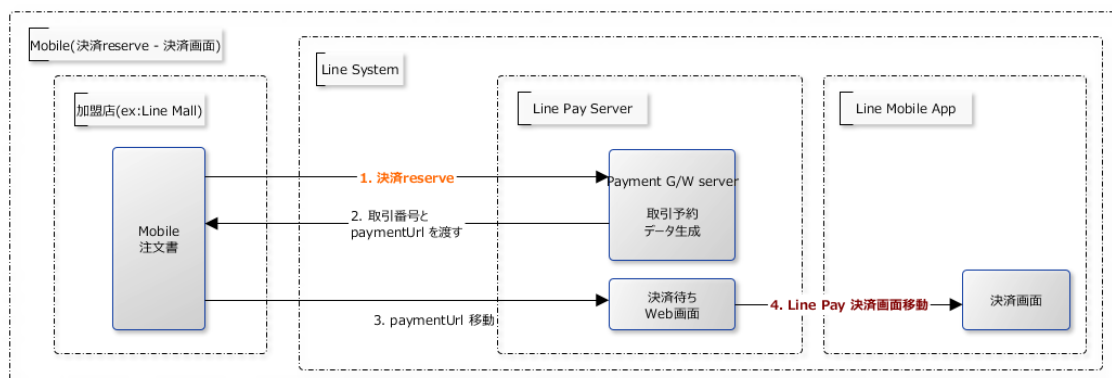


図 4 スマートフォン決済 reserve—決済画面

1. LINE 会員が加盟店の支払い手段として LINE Pay 支払いを選択する。
2. 加盟店では、LINE Pay「決済 reserve API」を実行し、取引番号と“paymentUrl”を受け取る。取引番号は、決済 confirm の際に必ず必要な情報であり、加盟店で保存しておく必要がある。
3. 加盟店は、b で受け取った paymentUrl(LINE pay 提供)を呼び出す。
4. paymentUrl を使って LINE アプリ内の決済画面へ移動する。以降のプロセスは、paymentUrl 加盟店の連動環境によって web とアプリで分かれている。
 - web: LINE Pay が提供する web 画面。LINE Pay 決済画面へ移動する前に LINE をインストールしているかどうか確認する。
 - アプリ: LINE Pay 決済画面へダイレクトに移動する app scheme url。デバイスの LINE インストール有無、LINE Pay 使用可能バージョン(Android のみ)をチェックし、LINE Pay 決済画面へ移動する。サンプルは、付録-「PaymentUrl app 使用ガイド」を参照。

決済画面—決済完了

LINE pay 会員は、LINE pay 決済画面で支払い手段(クレジットカードまたは残高)を選択し、決済に同意します。その後、決済 reserve 時に加盟店から受け取った“confirmUrl”(LINE Pay が transactionId パラメータを追加で設定します)へ移動します。加盟店へ移動後、LINE Pay に「決済 confirm API」を呼び出し、決済が完了します。

1. PC の場合

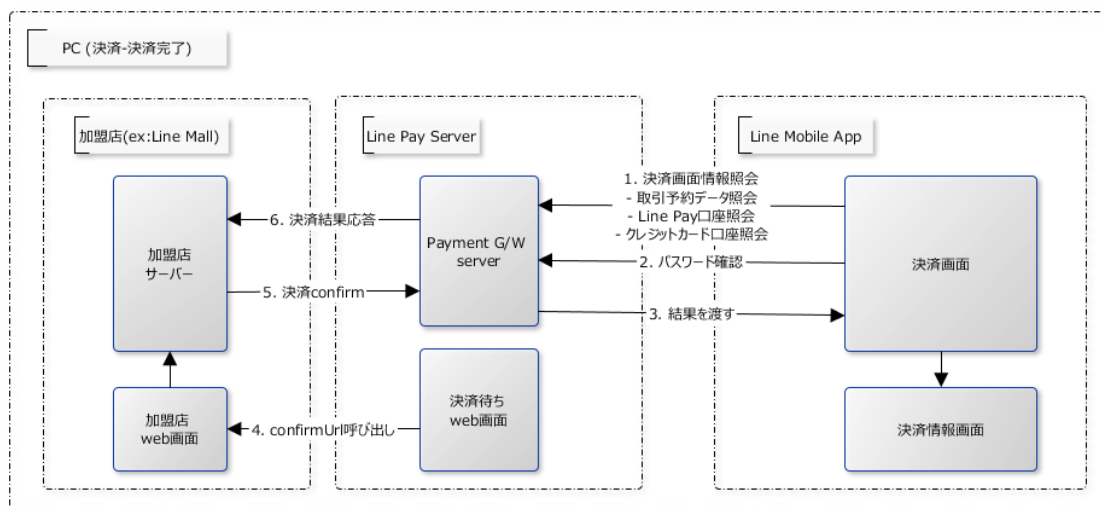


図 5 PC 決済画面—決済完了

1. LINE Pay 会員は、LINE Pay 決済画面で支払い手段を選択し、パスワードを入力する。
2. LINE Pay では、承認された支払い内容を保存して決済状態を変更する。
3. LINE Pay 会員は、LINE Pay で決済情報画面を確認する。
4. 決済待ち画面で決済可能な状態になったら、決済 reserve 時に加盟店が指定した“confirmUrl”へ移動する。
5. 加盟店は、LINE Pay に決済 confirm API を呼び出し、決済を完了させる。

2. スマートフォンの場合

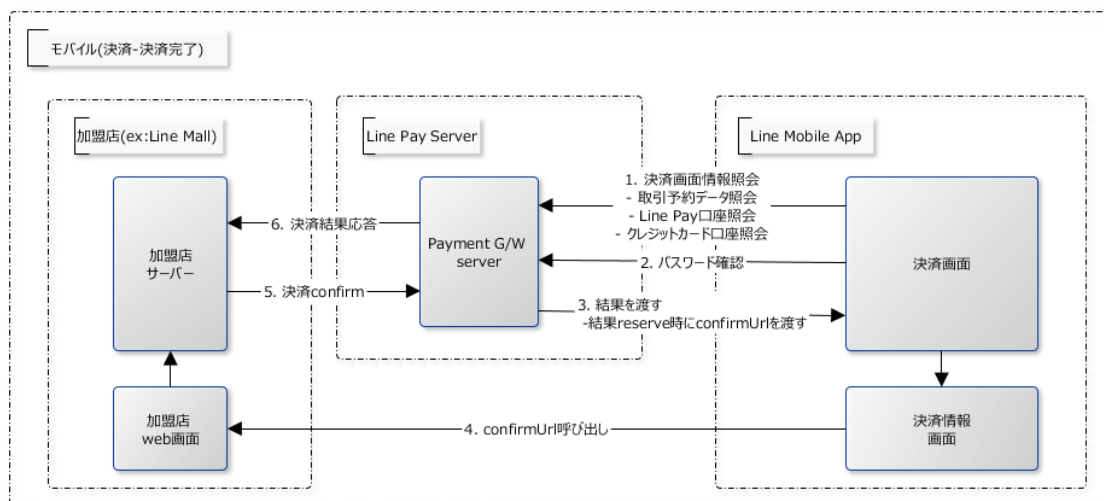


図 6 スマートフォン決済画面—決済完了

1. LINE Pay 会員は、LINE アプリ決済画面で支払い手段を選択し、パスワードを入力する。
2. LINE Pay では、承認された支払い内容を保存して決済状態を変更する。
3. LINE Pay 会員は、LINE アプリで決済情報画面を確認し、画面下のボタンをタップして決済 reserve 時に受け取った“confirmUrl”へ移動する。
4. 加盟店は、LINE Pay に決済 confirm API を呼び出し、決済を完了させる。

オーソリと売上を分けたい場合

オーソリと売上を分けるには、決済 reserve API を呼び出す際に“capture”パラメータを“false”にしてリクエストします。決済 confirm API が実行されると、決済はオーソリの状態となります。

加盟店は、売上処理時点で

- 売上確定を希望する場合は、capture API を呼び出して決済を完了させます。
- 売上確定を希望しない場合は、オーソリ無効 API を呼び出して決済のオーソリを取り消します。

confirmUrl を Server-to-Server で連携する場合

加盟店サーバーと LINE Pay サーバー間の通信のみで決済を行うことができます。

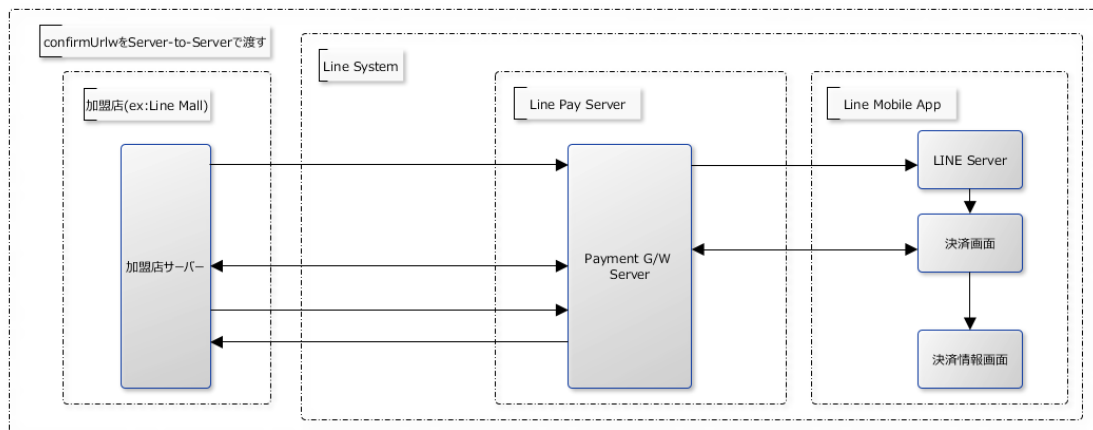


図 7 confirmUrl を Server-to-Server で連携

1. 加盟店は、決済 reserve 時に、confirmUrlType: “SERVER” を指定する。
2. LINE Pay 会員は、LINE Pay 決済画面で支払い手段を選択しパスワードを入力する。
3. LINE Pay サーバーは、決済内容を保存し決済 reserve 時に加盟店から指定された confirmUrl を呼び出す。
4. [付録] “ConfirmUrl を Server-to-Server で送信した場合”をご参照ください。
5. 加盟店サーバーは、決済 confirm API を呼び出して決済を完了する。confirmUrl に対する応答が正常に完了していなかった場合は、confirm API を呼び出すことができない。

oneTimeKey を利用して決済を行う場合

LINE Pay app では offline 決済のための oneTimeKey を提供します。oneTimeKey は LINE Pay 会員を特定する情報として Barcode もしくは QR Code として提供されます。加盟店はコードから oneTimeKey を取得して決済を行う事ができます。

前提条件

oneTimeKey 有効時間 : 10 分 (利用者が oneTimeKey コードを表示した時点から)

oneTimeKey は決済 reserve 実行と同時に削除される。

連動の流れ

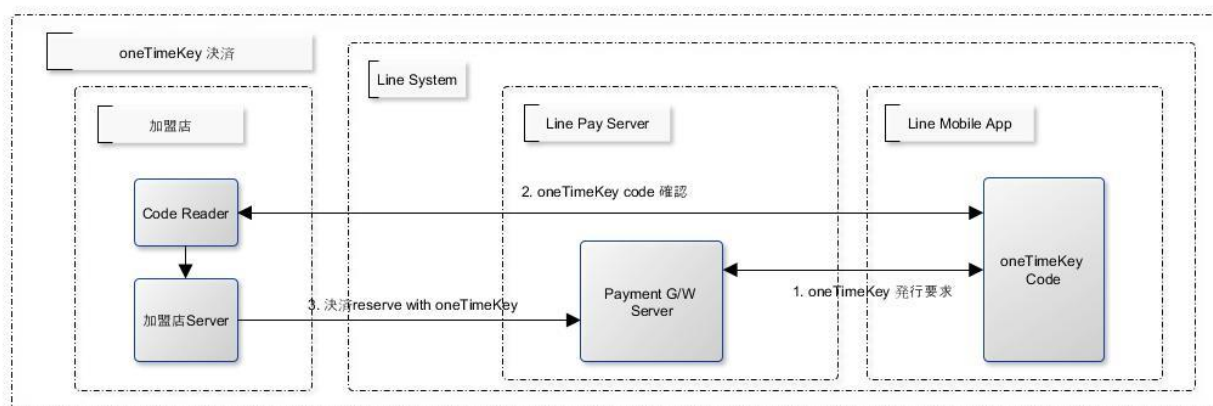


図 8 oneTimeKey 発行と決済 reserve

1. 加盟店は LINE Pay 会員が提示する oneTimeKey コード(barcode もしくは QR Code)を読取る。
2. 加盟店は取得した oneTimeKey 情報を利用し決済 reserve を実行する。
3. 決済 reserve が実行された後、LINE Pay 会員は LINE を通じて決済要請を確認する。
4. LINE Pay 会員は LINE Pay 決済画面で、決済手段を選択しパスワードを入力する。
5. LINE Pay 会員の決済承認後、加盟店への通知は決済 reserve API の confirmUrlType に従い行われる
 - a. CLIENT (デフォルト) : 会員端末の browser から confirmUrl に対して行なわれ、加盟店からの confirm API 呼出しによって決済完了となる。
 - b. SERVER : [\[付録\] "ConfirmUrl を Server-to-Server" に送信した場合を参照。](#)

加盟店サーバは決済 confirm API を実行し決済完了となる。confirmUrl 呼び出しに対する応答が正常に得られなかった場合には決済 confirm を実行することができない。

継続決済

決済リクエスト—決済画面

一般決済同様に決済 reserve API を利用します。リクエスト情報の“payType”に“PREAPPROVED”を指定してください。

決済画面—決済完了および regKey(継続決済キー)発行

一般決済同様に決済 confirm API を利用します。レスポンス情報で regKey が追加で設定されます。加盟店は、以後継続決済を行うために regKey を保存する必要があります。

継続決済

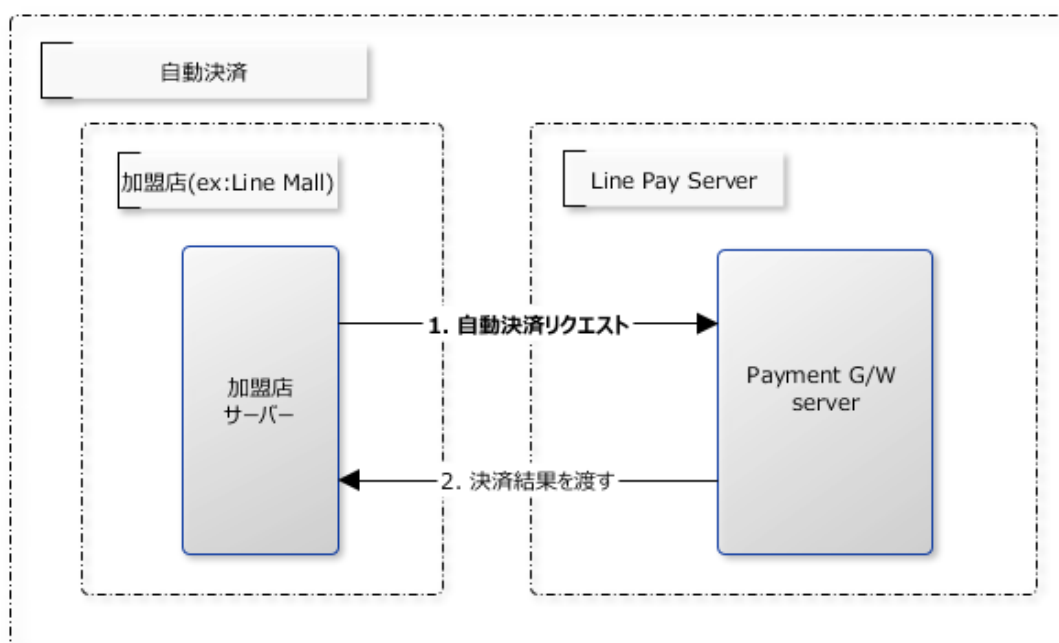


図 9 継続決済

決済 confirm API のレスポンスで取得した regKey を使って加盟店サーバーと LINE Pay サーバー間で「継続決済 API」を呼び出し、取引を行います。このとき、LINE Pay 会員の操作はありませんが、決済が完了したら LINE Pay 会員に決済完了を通知します。

regKey(継続決済キー)満了

regKey が不要となった（使用しない）場合は、「regKey 満了」とさせることができます。

LINE Pay 連動 API の使用方法

LINE Pay を連動する方法について詳しく説明します。連動を行う前に加盟店の認証情報を確認する必要がありますが、認証情報の確認は審査完了後に加盟店センター を通じて確認することができます。

共通機能

LINE Pay 認証

LINE Pay 連動時に必要な認証情報は、以下のとおりです。

- channel id
- channel secret key

すべて HTTP ヘッダーの情報として連携することになります。各 API の説明をご覧ください。

Request Header

項目	データタイプ	必須	説明
Content-Type	String	Y	application/json
X-LINE-ChannelId	String (10 Bytes)	Y	決済連動情報 - Channel ID
X-LINE-ChannelSecret	String (32 Bytes)	Y	決済連動情報 - Channel Secret Key
X-LINE-MerchantDeviceType	String (20 Bytes)	N	オフラインサポート- Device Type

インフラ・技術サポート

連動に関わる技術的な問い合わせや内部エラーが発生した場合は技術サポート(pay_tech@linecorp.com)までお問い合わせください。

環境	URL	説明
Beta	https://lgw-pay.line-apps-beta.com	LINE offline network 専用テスト環境
Sandbox	https://sandbox-api-pay.line.me	連動テストのための環境です。 Sandbox の Web シミュレーション決済ページにて、LINE Pay app の代わりに決済処理が可能です。 ご参照： Sandbox 利用ガイド
Real	https://api-pay.line.me	本番環境

決済内訳照会 API

LINE Pay の決済データを照会します。売上が確定されたデータのみ照会できます。

決済内訳照会 Request

GET /v2/payments

- Connection Timeout : 5 秒
- Read Timeout : 20 秒

表 1 決済内訳照会 Request Parameter

項目	データタイプ	必須	説明
transactionId[]	number	N	LINE Pay で発行した決済、または払い戻しの取引番号
orderId[]	String	N	加盟店の注文番号

* 照会をリクエストするときは、ひとつ以上のパラメータを指定しなければなりません。最大 100 件までの内訳を照会できます。

決済内訳照会 Response

表 2 決済内訳照会 Response Body

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> ● 0000: 成功 ● 1104: 加盟店がありません。 ● 1105: 加盟店が LINE Pay を利用できない状態です。 ● 1106: ヘッダー情報エラー ● 1150: 取引内訳がありません。 ● 1177: 照会可能な最大取引数を超えました(100 件)。 ● 9000 : 内部エラー
returnMessage	String	結果メッセージまたは、失敗の理由
info[].transactionId	number	取引番号(19 桁)
info[].transactionDate	String	取引日時 (ISO 8601)
info[].transactionType	String	取引区分 <ul style="list-style-type: none"> ● PAYMENT: 決済 ● PAYMENT_REFUND: 払い戻し

項目	データタイプ	説明
		<ul style="list-style-type: none"> PARTIAL_REFUND: 一部払い戻し
info[].productName	String	商品名
info[].merchantName	String	Merchant Name
info[].currency	String	通貨 (ISO 4217)
Info[].authorizationExpirationDate	String	オーソリ満了日時 (ISO 8601)
info[].payInfo[].method	String	決済時に使用した支払い手段 (クレジットカード: CREDIT_CARD, 残高: BALANCE, 割引: DISCOUNT)
info[].payInfo[].amount	number	取引金額(取引番号発生の際に取引された取引金額) 原決済取引を照会する際の最終取引金額は、 sum(info[].payInfo[].amount) - sum(refundList[].refundAmount)です。

原決済取引照会、および払い戻し取引がある場合

info[].refundList[].refund number TransactionId		払い戻し取引番号(19桁)
info[].refundList[].transactionType	String	取引区分 <ul style="list-style-type: none"> PAYMENT_REFUND : 払い戻し PARTIAL_REFUND : 一部払い戻し
info[].refundList[].refund number Amount		払い戻し金額
info[].refundList[].refund TransactionDate	String	払い戻し取引日時 (ISO 8601)

払い戻し取引の照会の場合

info[].originalTransaction Id	number	原決済取引番号 (19桁)
----------------------------------	--------	---------------

決済内訳照会 API の例

ソース 1 決済内訳照会 API リクエスト例

```
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
```

ソース 2 決済内訳照会 API 応答例(原決済取引照会の場合)

```
{
  "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"
      }
    ]
  }
]}
}
```

ソース 3 決済内訳照会 API 応答例(払い戻し取引照会の場合)

```
{
  "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"
  }
]}
}
```


決済 Reserve API

LINE Pay 決済を行う前に、加盟店の状態が正常であるかを判断し、決済のための情報を予約します。決済予約が成功したら、決済完了/払い戻しするまで使用する「取引番号」が発行されます。

決済 Reserve Request

POST /v2/payments/request

- Connection Timeout : 5 秒
- Read Timeout : 20 秒

表 3 決済 Reserve Request Parameter

項目	データタイプ	必須	説明
productName	String (4000 Bytes)	Y	商品名(charset: UTF-8)
productImageUrl	String (500 Bytes)	N	商品画像の URL 決済画面に表示する画像の URL <ul style="list-style-type: none"> ● サイズ: 84 x 84(決済画面でのみ表示。ガイド通りに表示することを推奨)
amount	number ¹	Y	決済金額
currency	String (3byte)	Y	決済通貨 (ISO 4217) 対応可能な通貨は以下のとおり。 <ul style="list-style-type: none"> ● USD ● JPY ● TWD ● THB
mid	String (50byte)	N	LINE member ID <ul style="list-style-type: none"> ● 決済を行う LINE 会員 mid
oneTimeKey	String (12byte)	N	One Time Key LINE Pay app で提供する QR/BarCode から取得し、LINE Pay 会員 mid にかわり会員を特定する情報となる。有効時間は 5 分で、reserveAPI が実行されると同時に削除される。 LINE Pay app の QR/ BarCode は LINE5.1 バージョンからサポートされます。

¹ 小数点が可能な数字型のデータタイプ

項目	データタイプ	必須	説明
confirmUrl	String (500byte)	Y	<p>LINE Pay で決済承認（決済方法選択、パスワード認証）後に遷移する加盟店 URL</p> <ul style="list-style-type: none"> 決済を完了するためには confirmUrl に遷移後、confirm API の呼び出しが必要となる。 url 移動時、transactionId 情報がパラメータで追加される。
<p>参考</p> <p>confirmUrl 詳細説明及び例外状況への対応</p>			
confirmUrlType	String	N	<p>confirmUrl タイプ</p> <p>LINE Pay で支払い手段を選択し、パスワードを確認後に移動する URL のタイプ</p> <ul style="list-style-type: none"> CLIENT: ユーザーベースの URL(デフォルト) SERVER: サーバーベースの URL。ユーザーは、LINE Pay で決済情報画面を確認することでシナリオが終了。LINE Pay から加盟店サーバーへ決済が可能な状態になったことを知らせる。
checkConfirmUrlBrowser	Boolean	N	<p>confirmUrl 遷移時の browser チェック可否</p> <ul style="list-style-type: none"> true: 決済を要請した browser と confirmUrl に移動した browser が異なる場合、元の browser で操作を行うように案内ページを LINE Pay で表示する。 false: 遷移先 browser の確認は行わず confirmUrl に直ぐに移動する。(デフォルト値)。
<p>参考</p> <p>confirmUrl 詳細説明及び例外状況への対応</p>			
cancelUrl	String (500byte)	N	<p>決済キャンセル Page の URL</p> <ul style="list-style-type: none"> LINE アプリ決済画面で LINE Pay 会員が決済をキャンセルした場合に移動する URL(キャンセル時に加盟店アプリまたは web に移動するための URL) 加盟店が指定した URL のまま移動する。 LINE Pay から追加で設定するパラメータなし。
packageName	String (4000byte)	N	<p>Android において、アプリ間切り替えの際 phishing を防止するための情報</p>

項目	データタイプ	必須	説明
orderId	String (100byte)	Y	決済予約に対する加盟店の注文番号 <ul style="list-style-type: none"> 加盟店で管理するユニークな番号。
deliveryPlacePhone	String (100byte)	N	受取人の連絡先 (for Risk Management)
payType	String (12byte)	N	決済タイプ <ul style="list-style-type: none"> NORMAL: 一般決済(デフォルト) PREAPPROVED: 継続決済
langCd	String	N	決済待ち画面(paymentUrl)言語コード。6 種の言語に対応。 <ul style="list-style-type: none"> ja : 日本語 ko : 韓国語 en : 英語 zh-Hans : 中国語(簡体字) zh-Hant : 中国語(繁体字) th : タイ語 必須情報ではない。指定されない場合には accept-language header を参照して多言語対応する。 対応していない langCd が指定された場合は、英語(en)をデフォルトで使用する。 BCP-47 형태: http://en.wikipedia.org/wiki/IETF_language_tag
capture	Boolean	N	売上処理 <ul style="list-style-type: none"> true: 決済 confirm API を呼び出した時にオーソリと売上を一度に処理する(デフォルト)。 false: 決済 confirm API を呼び出した時にオーソリのみ処理し、「capture API」を呼び出して取引を完了させる。
Extra fields			
extras.addFriends	object[]	N	友達リストを追加 <ul style="list-style-type: none"> - type: サービスタイプ <ul style="list-style-type: none"> - "LINE_AT": line@ - idList: ID List(加盟店センター内 LINE@/OA 管理メニューへ登録した ID リスト) <pre>"addFriends" : [{ "type" : "LINE_AT",</pre>

項目	データタイプ	必須	説明
			"idList" : ["@aaa", "@bbb"] }]
gmextras.branchName	String	N	決済を要求した店舗名(100文字まで表示) (200byte)

決済 Reserve Response

表 4 決済 reserve Response Body

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> 0000: 成功 1104: 加盟店がありません。 1105: 加盟店が LINE Pay を利用できない状態です。 1106: ヘッダー情報エラー 1124: 金額情報エラーです。(scale) 1133: 無効な oneTimeKey です。 1145: 決済中です。 1172: 同注文番号にて取引された履歴が存在しません 1178: この通貨には対応していません。 1194: 継続決済を使用できない加盟店です。 2101: パラメータエラー 2102: JSON データフォーマットエラー 9000: 内部エラー
returnMessage	String	結果メッセージまたは失敗理由。以下は例となります。 <ul style="list-style-type: none"> 決済不可能な加盟店 加盟店認証情報エラー
info.transactionId	Number	取引番号(19桁)
info.paymentUrl.web	String	決済予約後に移動するための Web URL <ul style="list-style-type: none"> 決済予約が Web で行われた場合に使用 LINE Pay 決済待ち画面に移動するための URL 指定された URL にそのまま移動 Desktop でポップアップ表示する場合の画面サイズ - Width: 700px, Height : 546px

項目	データタイプ	説明
info.paymentUrl.app	String.	決済画面へ移動するためのアプリの URL <ul style="list-style-type: none"> • アプリで決済予約が行われた場合に使用 • 加盟店アプリから LINE にアプリに移動をするための URL
参考 PaymentUrl app使用ガイド		
info.paymentAccessToken	String	LINE Pay app で Scanner を利用する代わりにコード番号を入力する場合に使用するコード値。(12桁) (LINE Pay app の Scanner は LINE5.1 からサポート)

決済 reserve API の例

ソース 4 決済 reserve API リクエストの例

```
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"
  }
}
```

ソース 5 決済 reserve API 応答の例

```
{
  "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"  
}
```

決済 Confirm API

加盟店が決済を最終的に完了させるための API です。加盟店で決済 confirm API を呼び出すことによって、実際の決済が完了します。決済 reserve 時に“capture”パラメータが“false”の場合、confirm API 実行時はオーソリ状態になるため、「capture API」実行時に決済完了となります。

決済 Confirm Request

POST /v2/payments/{transactionId}/confirm

- Connection Timeout : 5 秒
- Read Timeout : 40 秒

表 5 決済 Confirm Request Parameter

項目	データタイプ	必須	説明
amount	number	Y	決済金額
currency	String (3byte)	Y	決済通貨 (ISO 4217) 対応可能な通貨は以下のとおり。 <ul style="list-style-type: none"> • USD • JPY • TWD • THB

決済 Confirm Response

表 6 決済 confirm Response Body

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> • 0000: 成功 • 1101: 購入会員が取引不可の状態です。 • 1102: 購入会員が取引不可の状態です。 • 1104: 加盟店がありません。 • 1105: 加盟店が LINE Pay を利用できない状態です。 • 1106: ヘッダー情報エラー • 1110: 使用できないクレジットカードです。 • 1124: 金額情報のエラー • 1141: 口座の状態エラー • 1142: 口座の状態エラー

項目	データタイプ	説明
		<ul style="list-style-type: none"> 1150: 取引内訳がありません。 1152: すでに取引された決済です。 1153: 決済予約時の金額と要求された金額が異なります。 1159: 決済予約情報がありません。 1169: LINE Pay で決済手段の選択と認証を行う必要があります。 1170: 会員口座の残高が変動しました。 1172: すでに同じ注文番号で取引された履歴が存在します。 1180: 支払いの有効時間が経過しました。 1198: API の呼び出しが重複して発生しました。 1199: 内部要求エラー 1280: クレジットカード決済の一時的なエラー 1281: クレジットカード決済エラー 1282: クレジットカード承認エラー 1283: 不正使用が疑われるため決済が拒否されました。 1284: クレジットカード決済が一時的に中断されました。 1285: クレジットカード決済情報漏れ 1286: クレジットカードの決済情報が正しくありません。 1287: クレジットカードの有効期限が過ぎています。 1288: クレジットカード決済口座の残高が不足しています。 1289: クレジットカード利用限度額超過 1290: クレジットカード 1 件あたりの決済限度額超過 1291: 盗難報告されたカードです。 1292: 使用が停止されたカードです。 1293: CVN 入力エラー 1294: ブラックリストに登録されたカードです。 1295: クレジットカード番号が正しくありません。 1296: 処理できない金額です。 1298: カード使用が拒否されました。 9000: 内部エラー
returnMessage	String	<p>結果メッセージまたは、失敗の理由。以下は例となります。</p> <ul style="list-style-type: none"> 決済不可能な加盟店 加盟店認証情報エラー
info.orderId	String	決済予約時に加盟店が指定した注文番号
info.transactionId	number	決済予約時に発行された取引番号(19 桁)

項目	データタイプ	説明
info.authorizationExpirationDate	String	opt-オーソリ満了日時 (ISO 8601) <ul style="list-style-type: none"> オーソリ(capture=false)のみの決済の場合に連携する。
info.payInfo[].method	String	決済時に使用した支払い手段 (クレジットカード: CREDIT_CARD, 残高: BALANCE, 割引: DISCOUNT)
info.payInfo[].amount	number	決済金額
info.payInfo[].creditCardNickname	String	opt-継続決済時のクレジットカードのニックネーム <ul style="list-style-type: none"> LINE Pay で管理するクレジットカード名。登録の際に設定した名前。 LINE Pay 会員がニックネームを登録していない場合、空の文字列が設定される。 LINE Pay で会員が申請して変更可能で、変更内容は加盟店に共有されない。
info.payInfo[].creditCardBrand	String	opt-継続決済時のクレジットカードブランド <ul style="list-style-type: none"> VISA MASTER AMEX DINERS JCB
info.regKey	String	opt-継続決済のためのキー(15桁)

決済 confirm API の例

ソース 6 決済 confirm API リクエストの例

```
POST https://sandbox-api-pay.line.me/v2/payments/sdhqiwoehrafdasrqoi123as/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"
}
```

ソース 7 決済 confirm API 応答の例 (決済タイプ (payType) が NORMAL の場合の Balance 決済)

```
{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "orderId": "order_210124213",
```

```
"transactionId": 20140101123123123,
"payInfo": [{
  "method": "BALANCE",
  "amount": 10
}, {
  "method": "DISCOUNT",
  "amount": 10
}]
}
```

ソース 8 決済 confirm API 応答の例 (決済タイプ (payType) が NORMAL の場合ポイント複合決済)

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

ソース 9 決済 confirm API 応答の例 (決済タイプ (payType) が REAPPROVED の場合)

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

払い戻し API

LINE Pay で決済された取引の払い戻しをリクエストします。払い戻しの際には、LINE Pay 会員の決済取引番号を必ず指定しなければなりません。一部払い戻しも可能です。払い戻し可能期間は売上(capture)から 30 日間となります。

払い戻し Request

POST /v2/payments/{transactionId}/refund

- Connection Timeout : 5 秒
- Read Timeout : 20 秒

表 7 払い戻し Request Parameter

項目	データタイプ	必須	説明
refundAmount	number	N	払い戻し金額 - 送信されなかった場合は全額払い戻しする。

払い戻し Response

表 8 払い戻し Response Body

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> ● 0000: 成功 ● 1101: 購入会員が取引不可の状態です。 ● 1102: 購入会員が取引不可能な状態です。 ● 1104: 加盟店がありません。 ● 1105: 加盟店が LINE Pay を利用できない状態です。 ● 1106: ヘッダー情報エラー ● 1124: 金額情報のエラー ● 1150: 取引内訳がありません。 ● 1155: 無効な取引番号です。 ● 1163: 払い戻しはできません。(返金可能日超過) ● 1164: 払い戻し可能な金額を超えました。 ● 1165: すでに払い戻し済みの取引です。 ● 1179: 処理できない状態です。 ● 1198: API の呼び出しが重複して発生しました。

項目	データタイプ	説明
		<ul style="list-style-type: none"> 1199 : 内部要求エラー 9000 : 内部エラー
returnMessage	String	結果メッセージまたは、失敗の理由
info.refundTransactionId	number	払い戻し取引番号(新規発行-19桁)
info.refundTransactionDate	String	払い戻し取引日時 (ISO 8601)

払い戻し API の例

ソース 10 払い戻し API リクエスト例

```
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
}
```

ソース 11 払い戻し API 応答例 (決済タイプ (payType) が NORMAL の場合の Balance 決済)

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

オーソリ内訳照会 API

LINE Pay でのオーソリ履歴の内訳を照会する API です。オーソリ済み、またはオーソリ無効処理データのみ照会できます。売上が確定されたデータは「決済内訳照会 API」で照会できます。

オーソリ内訳照会 Request

GET /v2/payments/authorizations

- Connection Timeout : 5 秒
- Read Timeout : 20 秒

表 9 オーソリ内訳照会 Request Parameter

項目	データタイプ	必須	説明
transactionId[]	number	N	LINE Pay で発行した取引番号
orderId[]	String	N	加盟店の注文番号

*照会をリクエストするときは、ひとつ以上のパラメータを指定しなければなりません。最大 100 件までの内訳を照会できます。

オーソリ内訳照会 Response

表 10 オーソリ内訳照会 Response Body

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> ● 0000: 成功 ● 1104: 加盟店がありません。 ● 1105: 加盟店が LINE Pay を利用できない状態です。 ● 1106: ヘッダー情報エラー ● 1150: 取引内訳がありません。 ● 1177: 照会可能な最大取引数を超過しました(100 件)。 ● 9000 : 内部エラー
returnMessage	String	結果メッセージまたは、失敗の理由
info[].transactionId	number	取引番号(19 桁)
info[].transactionDate	String	取引日時 (ISO 8601)
info[].transactionType	String	取引区分 <ul style="list-style-type: none"> ● PAYMENT: 決済 ● PAYMENT_REFUND: 払い戻し ● PARTIAL_REFUND: 一部払い戻し

項目	データタイプ	説明
info[].payStatus	String	決済状態 <ul style="list-style-type: none"> AUTHORIZATION: オーソリ VOIDED_AUTHORIZATION: オーソリ無効(「オーソリ無効処理 API」を呼び出した状態) EXPIRED_AUTHORIZATION: オーソリ満了(LINE Pay で加盟店に許可したオーソリ期限が過ぎた場合)
info[].payInfo[].method	String	決済時に使用した支払い手段 (クレジットカード: CREDIT_CARD, 残高: BALANCE, 割引: DISCOUNT)
info[].payInfo[].amount	number	取引金額(取引番号発生の際に取引された取引金額) 原決済取引を照会する際の最終取引金額は、 sum(info[].payInfo[].amount) - sum(refundList[].refundAmount)です。
info[].productName	String	商品名
info[].currency	String	通貨 (ISO 4217)
info[].orderId	String	加盟店注文番号
info[].authorizationExpirationDate	String	オーソリ満了日時 (ISO 8601)

オーソリ内訳照会 API の例

ソース 12 オーソリ内訳照会 API リクエスト例

```
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
```

ソース 13 オーソリ内訳照会 API リクエスト例

```
{
  "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

決済 reserve API を呼び出す時に“capture”パラメータを“false”で指定した場合は、売上処理を行ったうえで決済を完了させることができます。決済 confirm API から 30 日以内に売上を確定させてください。期限を過ぎるとキャンセルとなります。

Capture Request

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

- Connection Timeout : 5 秒
- Read Timeout : 60 秒

表 11 Capture Request Parameter

項目	データタイプ	必須	説明
amount	number	Y	決済金額
currency	String (3byte)	Y	決済通貨 (ISO 4217) 対応可能な通貨は、以下のとおり。 <ul style="list-style-type: none"> ● USD ● JPY ● TWD ● THB

Capture Response

表 12 Capture Response Body

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> ● 0000: 成功 ● 1104: 加盟店がありません。 ● 1105: 加盟店が LINE Pay を利用できない状態です。 ● 1106: ヘッダー情報エラー ● 1150: 取引内訳がありません。 ● 1155: 無効な取引番号です。 ● 1170: 会員口座の残高が変動しました。 ● 1172: すでに同じ注文番号で取引された履歴が存在しません。 ● 1179: 処理できない状態です。 ● 1183: 金額のエラー

項目	データタイプ	説明
		<ul style="list-style-type: none"> 1184 : 金額のエラー 1198 : API の呼び出しが重複して発生しているか、内部の自動延長処理が実行中の場合。(リトライ可能) 1199 : 内部要求エラー 1280 : クレジットカード決済の一時的なエラー 1281 : クレジットカード決済エラー 1282 : クレジットカード承認エラー 1283 : 不正使用が疑われるため決済が拒否されました。 1284 : クレジットカード決済が一時的に中断されました。 1285 : クレジットカード決済情報漏れ 1286 : クレジットカードの決済情報が正しくありません。 1287 : クレジットカードの有効期限が過ぎています。 1288 : クレジットカード決済口座の残高が不足しています。 1289 : クレジットカード利用限度額超過 1290 : クレジットカード 1 件あたりの決済限度額超過 1291 : 盗難報告されたカードです。 1292 : 使用が停止されたカードです。 1293 : CVN 入力エラー 1294 : ブラックリストに登録されたカードです。 1295 : クレジットカード番号が正しくありません。 1296 : 処理できない金額です。 1298 : カード使用が拒否されました。 9000 : 内部エラー <p># エラーコード 1199、1280~1298 が発生した場合、取り引きは自動取り消しされる。</p>
returnMessage	String	結果メッセージまたは、失敗の理由。以下は例となります。 <ul style="list-style-type: none"> 決済不可能な加盟店 加盟店認証情報エラー
info.orderId	String	決済予約時に加盟店が指定した注文番号
info.transactionId	number	決済予約時に発行された取引番号(19桁)
info.payInfo[].method	String	決済時に使用した支払い手段 (クレジットカード: CREDIT_CARD, 残高: BALANCE, 割引: DISCOUNT)
info.payInfo[].amount	number	決済金額

capture API の例

ソース 14 capture API リクエストの例

```
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"
}
```

ソース 15 capture API 応答の例

```
{
  "returnCode": "0000",
  "returnMessage": "OK",
  "info": {
    "transactionId": 20140101123123123,
    "orderId": "order 210124213",
    "payInfo": [{
      "method": "BALANCE",
      "amount": 10
    }, {
      "method": "DISCOUNT",
      "amount": 10
    }
  ]
}
```

オーソリ無効処理 API

オーソリ状態の決済データを無効にします。決済 confirm API を呼び出してオーソリ状態まで進んだ場合に、オーソリを無効にする API です。オーソリ状態の決済のみ無効に処理できます。売上が確定された決済は、「払い戻し API」を使って払い戻し処理を行って下さい。

オーソリ無効処理 Request

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

- Connection Timeout : 5 秒
- Read Timeout : 20 秒

オーソリ無効処理 Response

表 13 オーソリ無効処理 Response Body

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> ● 0000: 成功 ● 1101: 購入会員が取引不可の状態です。 ● 1102: 購入会員が取引不可の状態です。 ● 1104: 加盟店がありません。 ● 1105: 加盟店が LINE Pay を利用できない状態です。 ● 1106: ヘッダー情報エラー ● 1150: 取引内訳がありません。 ● 1155: 無効な取引番号です。 ● 1165: すでに無効化され取引です。 ● 1170: 会員口座の残高が変動しました。 ● 1198: API の呼び出しが重複して発生しました。 ● 1199: 内部要求エラー ● 1900: 一時的エラーです。しばらくしてリトライして下さい。 ● 1902: 一時的エラーです。しばらくしてリトライして下さい。 ● 1999: 前回の要請情報と違います。(要請リトライ) ● 9000: 内部エラー # エラーコード 1900,1903,1999 発生の時リトライ可能。
returnMessage	String	結果メッセージまたは失敗の理由。以下は例となります。 <ul style="list-style-type: none"> ● 決済不可能な加盟店 ● 加盟店認証情報エラー

オーソリ無効処理 API の例

ソース 16 オーソリ無効処理 API リクエストの例

```
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
```

ソース 17 オーソリ無効処理 API 応答の例

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

継続決済 API

決済 reserve API で決済タイプ(type)が PREAPPROVED で決済された場合、決済結果の受信時に regKey を受け取ります。継続決済 API は、この regKey を利用し LINE アプリを介さずに直接決済する際に使用します。

継続決済 Request

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

- Connection Timeout : 5 秒
- Read Timeout : 40 秒

表 14 継続決済 Request Parameter

項目	データタイプ	必須	説明
productName	String (4000byte)	Y	商品名
amount	number	Y	決済金額
currency	String (3byte)	Y	決済通貨 (ISO 4217) 対応可能な通貨は、以下のとおり。 <ul style="list-style-type: none"> ● USD ● JPY ● TWD ● THB
orderId	String (100byte)	Y	決済に対する加盟店の注文番号。 加盟店で管理するユニークな番号。
capture	Boolean	N	売上確定を行うかどうか <ul style="list-style-type: none"> ● true (デフォルト) : オーソリと売上を一度に処理する。 ● false : オーソリまで行い「売上 API」の実行をもって取引が完了となる

継続決済 Response

表 15 継続決済 Response Body

項目	データタイプ	説明
----	--------	----

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> • 0000: 成功 • 1101: 購入した会員が取引不可能な状態です。 • 1102: 購入した会員が取引不可能な状態です。 • 1104: 加盟店がありません。 • 1105: 加盟店が LINE Pay を利用できない状態です。 • 1106: ヘッダー情報エラー • 1110: 使用できないクレジットカードです。 • 1124: 金額情報のエラー • 1141: 支払い口座の状態エラー • 1154: 継続決済口座の状態が正常ではありません。 • 1172: すでに同じ注文番号で取引された履歴が存在しません。 • 1183: 決済金額は 0 より大きくなければなりません。 • 1190: regKey がありません。 • 1193: regKey 満了 • 1198: API の呼び出しが重複して発生しました。 • 1199: 内部要求エラー • 1280: クレジットカード決済の一時的なエラー • 1281: クレジットカード決済エラー • 1282: クレジットカード承認エラー • 1283: 不正使用が疑われるため決済が拒否されました。 • 1284: クレジットカード決済が一時的に中断されました。 • 1285: クレジットカード決済情報漏れ • 1286: クレジットカードの決済情報が正しくありません。 • 1287: クレジットカードの有効期限が過ぎています。 • 1288: クレジットカード決済口座の残高が不足しています。 • 1289: クレジットカード利用限度額超過 • 1290: クレジットカード 1 件あたりの決済限度額超過 • 1291: 盗難報告されたカードです。 • 1292: 使用が停止されたカードです。 • 1293: CVN 入力エラー • 1294: ブラックリストに登録されたカードです。 • 1295: クレジットカード番号が正しくありません。 • 1296: 処理できない金額です。 • 1298: カード使用が拒否されました。
		<ul style="list-style-type: none"> • 9000: 内部エラー

error code 1281~1287、1290~1294 が発生した場合 regKey を期限切れに処理する。

項目	データタイプ	説明
returnMessage	String	結果メッセージまたは、失敗の理由
info.transactionId	number	取引番号(19桁)
info.transactionDate	String	取引日時 (ISO 8601)
info.authorizationExpirationDate	String	オーソリ有効期限日時 (ISO 8601)
reDate		<ul style="list-style-type: none">売上確定を別途行う場合 (capture= false) のみ連携される

継続決済 API の例

ソース 18 継続決済 API リクエスト例

```
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": " MUSIC月額利用料 ",
  "amount": 10000,
  "currency": "JPY",
  "orderId": "testOrd2014121200000001"
}
```

ソース 19 継続決済 API 応答例

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

regKey 使用可否確認 API

継続決済 API を使用する前に、regKey が使用可能な状態であるかどうかを確認します。

regKey 使用可否確認 Request

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

- Connection Timeout : 5 秒
- Read Timeout : 20 秒

表 16 regKey 使用可否確認 Request Parameter

項目	データタイプ	必須	説明
creditCardAuth	Boolean	N	regKey 登録されたクレジットカードに対して最小金額でオーソリを行うかどうか <ul style="list-style-type: none"> ● true : <ul style="list-style-type: none"> - LINE Pay 内部のデータ検証. - 最小金額でのオーソリ - 機能を有効するためには別途 LINE Pay への申請が別途必要となります。 ● false(デフォルト) : <ul style="list-style-type: none"> - LINE Pay 内部のデータ検証.

regKey 使用可否確認 Response

表 17 regKey 使用可否確認 Response Body

項目	データタイプ	説明
returnCode	String(4byte)	結果コード <ul style="list-style-type: none"> ● 0000: 成功 ● 1101: 購入会員が取引不可の状態です。 ● 1102: 購入した会員が取引できない状態です。 ● 1104: 加盟店がありません。 ● 1105: 加盟店が LINE Pay を利用できない状態です。 ● 1106: ヘッダー情報エラー ● 1141: 支払い口座の状態エラー ● 1154: 継続決済口座の状態が正常ではありません。

項目	データタイプ	説明
		<ul style="list-style-type: none">• 1190: regKey がありません• 1193: regKey 満了
returnMessage	String	結果メッセージまたは、失敗の理由

regKey 使用可否確認 API の例

ソース 20 regKey 使用可否確認 API リクエスト例

```
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
```

ソース 21 regKey の使用可否確認 API 応答例

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

regKey 満了 API

継続決済で登録された regKey 情報を満了させる API です。この API を呼び出した以降は、当該の regKey では継続決済することができなくなります。

regKey 満了 Request

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

- Connection Timeout : 5 秒
- Read Timeout : 20 秒

regKey 満了 Response

表 18 regKey 満了 Response

項目	データタイプ	説明
returnCode	String (4byte)	結果コード <ul style="list-style-type: none"> ● 0000: 成功 ● 1104: 加盟店がありません。 ● 1105: 加盟店が LINE Pay を利用できない状態です。 ● 1106: ヘッダー情報エラー ● 1190: regKey がありません。 ● 1193: regKey 満了 ● 9000 : 内部エラー
returnMessage	String	結果メッセージまたは、失敗の理由

regKey 満了 API の例

ソース 22 regKey 満了 API リクエスト例

```
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
```

ソース 23 regKey 満了 API 応答例

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

付録

LINE Pay エラーコード

LINE Pay で発生するエラーコードについて定義します。エラーコードの returnMessage は、英語で提供されます。メッセージがない場合は、ハイフン(-)が設定されます。

表 19 LINE Pay エラーコード

コード (returnCode)	説明
1101	LINE Pay 会員ではありません。
1102	購入した会員が取引不可能な状態です。
1104	加盟店がありません。
1105	加盟店が LINE Pay を利用できない状態です。
1106	ヘッダー情報エラー
1110	使用できないクレジットカードです
1124	金額情報エラーです。(scale)
1133	無効な oneTimeKey です。
1141	決済口座の状態エラー
1142	残高不足
1145	決済中です。
1150	取引内訳がありません。
1152	既に取り引された決済です。
1153	決済予約時の金額と要求された金額が異なります。
1154	継続決済口座の状態が正常ではありません。
1155	無効な取引番号です。
1159	決済情報がありません。
1163	払い戻し可能日が過ぎているため払い戻しができません。
1164	払い戻し可能な金額を超えました。
1165	すでに払い戻し済みの取引です。
1169	LINE Pay で決済手段の選択と認証を行う必要があります。
1170	会員口座の残高が変動しました。
1172	すでに同じ注文番号で取引された履歴が存在します。

コード (returnCode)	説明
1177	照会可能な最大取引数を超えました(100件)。
1178	この通貨には対応していません。
1179	処理できない状態です。
1180	支払いの有効時間が経過しました。
1183	決済金額は0より大きくなければなりません。
1184	お支払い金額を要求時の金額を超えています。
1190	regKeyがありません。
1193	regKey 満了。
1194	継続決済を使用できない加盟店です。
1197	regKey ですでに決済中です。
1198	リクエスト処理中です。
1199	内部要求エラー
1280	クレジットカード決済の一時的なエラー
1281	クレジットカード決済エラー。
1282	クレジットカード承認エラー。
1283	不正使用が疑われるため決済が拒否されました。
1284	クレジットカード決済が一時的に中断されました。
1285	クレジットカード決済情報漏れ。
1286	クレジットカードの決済情報が正しくありません。
1287	クレジットカードの有効期限が過ぎています。
1288	クレジットカードの決済口座の残高が不足しています。
1289	クレジットカードの利用限度額超過。
1290	クレジットカード1件あたりの決済限度額超過。
1291	盗難報告されたカードです。
1292	使用が停止されたカードです。
1293	CVN 入力エラー。
1294	ブラックリストに登録されたカードです。
1295	クレジットカード番号が正しくありません。
1296	処理できない金額です。

コード (returnCode)	説明
1298	カード使用が拒否されました。
2101	パラメータエラー。
2102	JSON データフォーマットエラー。
9000	内部エラー

内部エラー: 1199,9000 が発生した場合は、技術連動支援(pay_tech@linecorp.com)までお問い合わせください。

表 20 API 固有のエラーコード

エラー コード	reserve	confirm	売上	認証 無効	払い戻し	認証 履歴	支払 履歴	継続 決済	regKey 確認	regKey 満了
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		
2101	v	v	v	v	v	v	v	v	v	v

エラー コード	reserve	confirm	売上	認証 無効	払い戻し	認証 履歴	支払 履歴	継続 決済	regKey 確認	regKey 満了
2102	√	√	√	√	√	√	√	√	√	√
9000	√	√	√	√	√	√	√	√	√	√

エラーコード 1280～1298 は、カード会社で発生するエラーであり記載しない。

PaymentUrl app 使用ガイド

加盟店アプリにて決済をリクエストした以降に、LINE アプリでアプリに移動し決済を行う方法について説明します。

Android アプリのサンプル

下記のサンプルコードで LINE アプリのインストール有無と使用可能な LINE Pay のバージョンを確認できます。LINE アプリがインストールされていて、使用可能な LINE Pay のバージョンが確認できたら、LINE Pay 決済画面へ移動します。

```
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>
```


iPhone アプリのサンプル

下記のサンプルコードで LINE アプリのインストール有無を確認することができます。LINE アプリがインストールされていたら、LINE Pay 決済画面へ移動します。

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

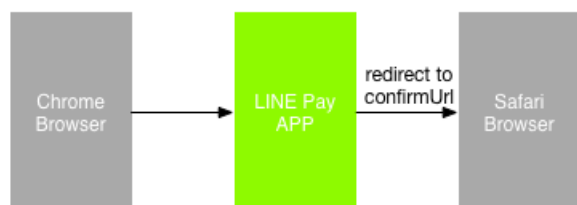
confirmUrl 詳細説明及び例外状況への対応

confirmUrl 詳細説明

利用者が LINE Pay app で決済承認(決済手段選択と暗証番号確認)後、呼び出される callback url 決済の完了を通知する url ではなく、LINE Pay 会員の決済承認が完了してことを通知する url で、決済を完了させるためには confirm API を呼び出す必要がある。

confirmUrl 例外状況への対応

LINE Pay での決済承認後、confirmUrl 呼び出しが、利用者の環境によっては決済を要請した Browser と異なる可能性がある。



決済 reserve API の “checkConfirmUrlBrowser”が “true”の場合

LINE Pay で browser を確認し、決済を始めた browser と confirmUrl が呼び出された browser が同じか判断する。判断結果により

- 同じである場合 : confirmUrl へ移動
- 異なる場合 : 決済を要請した browser への移動を案内する。元のブラウザで「続ける」ボタンを押すと confirmUrl が呼び出される

決済 reserve API の “checkConfirmUrlBrowser”が “false”(デフォルト値)の場合

加盟店で Browser の session data(ログイン session)に基づいて処理が行われる場合、confirmUrl 呼び出しの際、利用者の決済が正常に行われない場合もある。従って、加盟店は LINE Pay の決済連動時、下記のような対応を行う必要がある。

1. confirmUrl が呼び出される時、session data に基づいた処理を行わず、confirmUrl に含まれた情報で決済を完了。
。
2. session data に基づいた処理が必要な場合
 - a. session data の確認ができない場合、既存の加盟店決済結果画面表示
 - b. session data の確認ができない場合、下記2つ方法の中で対応策を選択し実現する必要がある
 - session data 再生成(ログイン) 後、決済結果画面を表示
 - 決済を要請した browser に戻るよう、利用者に案内

ConfirmUrl を Server-to-Server で連携する場合

LINE Pay から加盟店サーバーへ ConfirmUrl を呼び出す場合を定義します。

加盟店が ACL (Access Control List) を管理している場合

加盟店は LINE Pay サーバーの IP を登録/管理しなければならない。LINE Pay の環境固有の IP 情報は以下の通り。

- Sandbox :182.162.196.200
- Real : 211.249.40.1~211.249.40.30

ConfirmUrl 詳細

表 21 confirmUrl Endpoint

項目	説明
Protocol	HTTP
Method	GET
Request timeout	Connection: 5 秒 Read: 20 秒

表 22 confirmURL パラメータ

パラメータ	説明
orderId	決済 reserve の際に加盟店が指定した注文番号
transactionId	決済 reserve の際に発行された取引番号

加盟店は応答結果として別途情報を連携する必要はありません。LINE Pay では、HTTP 応答コードで成功有無を判断します。HTTP 応答コードが成功(200 OK)でない場合は、LINE Pay 会員へ決済が正常に処理されなかった旨を通知します。

ConfirmUrl 例

加盟店の提供する confirmUrl が `http://testmall.com/pay/result` だった場合の例は以下のとおりです。

リクエスト(LINE Pay -> 加盟店)

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

応答

```
// 成功した場合
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

// 失敗した場合
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
```

Sandbox 利用方法

LINE Pay Sandbox 環境にてテストする場合、本連動ガイドが提供する API Endpoint に記載された sandbox を利用し決済テストを実施できる。決済呼出ページは加盟店側で準備が必要となる。

Sandbox 環境で決済テストを実施する場合、次のように処理される。

※ 特記事項

Sandbox 環境では PC、モバイル共に、全てチャンネルログインを要求する。

(ただし、リアル環境ではスマートフォンの場合や、LINE ユーザー情報 (MID) を送信できる場合、チャンネルログインは表示されない。)

Sandbox PC 決済



図 10 Sandbox PC 決済画面の例

Sandbox モバイル決済

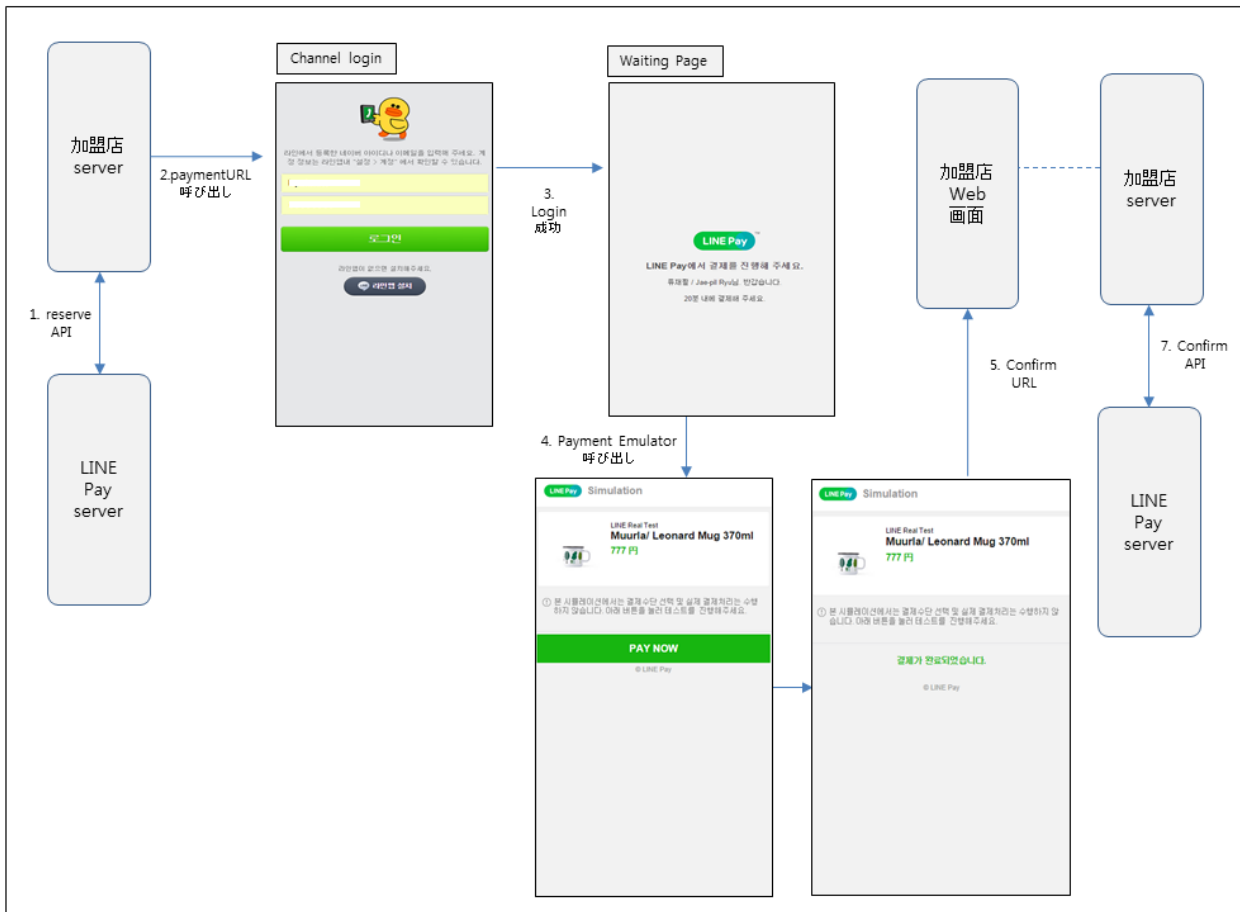


図 11 Sandbox モバイル決済画面の例

Sandbox OneTimeKey 決済

Sandbox でも WEB ページに表示された QR/ Bar Code から決済のテストをすることができます。

加盟店は QR/ Bar Code を読み取り決済 Reserve API を呼び出すと、QR/ Bar Code が表示された WEB ページが決済 simulation ページに移動する。

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