VeriTrans4G 開発ガイド別冊
LINE Pay ご利用ガイド

Ver. 1.0.8 （2023 年 9 月～）

Copyright © 2023 DG Financial Technology, Inc., a Digital Garage company. All rights reserved.

目次

第 1 章　本ガイドの内容 3
1-1 本ガイドの内容 3
1-2 著作権、および問い合わせ先 3
1-3 改訂履歴 3

第 2 章　LINE Pay サービス概要 5
2-1 LINE Pay サービスの概要 5
2-2 LINE Pay サービスのパターン 6

第 3 章　MDK の処理概要 8
3-1 機能一覧（決済要求コマンド） 8
3-2 決済処理シーケンス 9

第 4 章　結果通知受信処理 18
4-1 結果通知受信処理の対象 18

第 5 章　インターフェイス詳細 19
5-1 LINE Pay 19
5.1.1 申込 19
5.1.2 消費者ブラウザから店舗側の完了画面へリダイレクトする時のクエリパラメータ 22
5.1.3 売上 23
5.1.4 取消 23
5-2 共通 25
5.2.1 検索 25
5-3 結果通知電文 29
5.3.1 結果通知電文の概要 29
5.3.2 LINE Pay の結果通知電文 29

第 6 章　その他　補足事項 31
6-1 売上およびキャンセルに関する注意事項 31
6.1.1 売上およびキャンセルの要求可能期間 31
6-2 検索（Search）に関する補足（詳細コマンドタイプ/詳細オーダー決済状態） 31
6.2.1 決済確認方式：サーバ間通信の場合の検索結果について 32
6-3 スマートフォン向けのサイトおよび独自アプリと LINE Pay アプリとの連携について 33
6.3.1 appUrlScheme の指定方法 33
6.3.2 redirectAppURL の利用方法 35
6.3.3 checkUseBrowser の利用方法 38
6-4 LINE メッセージについて 39

第 7 章　導入・テストに関する補足 40
7-1 決済申込時の画面遷移をシミュレートする 40

7.1.1 PC またはスマートフォン（ブラウザ）での画面遷移 40
7.1.2 スマートフォン（加盟店アプリ）を利用した場合の画面遷移 42
7-2 決済確認方法がサーバ間通信で oneTimeKey 利用時の決済申込をシミュレートする 43
7-3 決済申込・売上・取消コマンドのエラーをシミュレートする 43

第1章 本ガイドの内容

1-1 本ガイドの内容

本ガイドは、株式会社 DG フィナンシャルテクノロジーが提供する VeriTrans4G の LINE Pay サービスをインターネット店舗等に導入するための、WEB アプリケーション開発者向けのガイドです。

1-2 著作権、および問い合わせ先

［著作権］

本ドキュメントの著作権は株式会社 DG フィナンシャルテクノロジーが保有しています。

Copyright © 2023 DG Financial Technology, Inc., a Digital Garage company. All rights reserved.

［お問い合わせ先］

株式会社 DG フィナンシャルテクノロジー ベリトランス テクニカルサポート

電子メール： tech-support@veritrans.jp

1-3 改訂履歴

2017/04 ：Ver. 1.0.0 リリース

※ 以下、「VeriTrans3G 開発ガイド別冊 LINE Pay ご利用ガイド」 Ver 1.0.3 からの更新分を記載します。

「6-3 LINE から消費者に送信されるメール」を削除

2017/07 ：Ver 1.0.1 リリース

「6-4 スマートフォン向けのサイトおよび独自アプリと LINE Pay アプリとの連携について」の paymentConfirmType の説明の誤りを修正

2018/01 ：Ver 1.0.2 リリース

「5-3 結果通知電文」の「通知件数」の「説明」を修正

（1 度に通知可能な件数を 1,000 件から 100 件に変更）

2018/05 ：Ver 1.0.3 リリース

LINE Pay クレジットカード連携サービスの提供中止に伴い以下より関連記述を削除。

- 「2-1 LINE Pay サービスの概要」よりクレジットカードでの支払いに関する記述を削除。
- 「3-1 機能一覧（決済要求コマンド）」の表注釈よりクレジットカードの与信に関する記述を削除。
- 「6-1 売上およびキャンセルに関する注意事項」よりクレジットカード支払い時のキャンセルに関する記述を削除。
- 「6-2 チャージバックについて」を削除。

「2-1 LINE Pay サービスの概要」に LINE Pay チャージ方法に関する記述を追加。

2019/04 ：Ver 1.0.4 リリース

「6-4 LINE メッセージについて」を追加

2019/06 ：Ver 1.0.5 リリース

「5.1.1 申込」 要求電文：LinepayAuthorizeRequestDto の oneTimeKey の桁数を、

半角英数字 12 桁以内から半角英数字 19 桁以内に変更。

2019/10 ：Ver 1.0.6 リリース

「6.2.1 決済確認方式：サーバ間通信の場合の検索結果について」の内容を変更

2020/07 ：Ver 1.0.7 リリース

「3-2 決済処理シーケンス」の申込結果通知(PUSH)L の処理説明を変更

「5.1.1 申込」の pushUrl の説明を変更

2023/09 ：Ver 1.0.8 リリース

「1.2.1 著作権、および問い合わせ先」内容をベリトランス（株）から株式会社 DG フィナンシャルテクノロジーへ変更

「3-2 決済処理シーケンス」

以下のシーケンスに結果通知(PUSH)の受信を強調する文言を追加

【LINE Pay：決済申込時の処理（PC＋スマートフォン）】

【LINE Pay：決済申込時の処理（スマートフォンアプリから LINE Pay に連動） ※ブラウザを介するケース】

「5.1.2 消費者ブラウザから店舗側の完了画面へリダイレクトする時のクエリパラメータ」

結果通知取得の重要性を追記

「この改ざんチェックは必須ではありませんが」の「必須ではありませんが」を削除

VeriTrans4G LINE Pay ご利用ガイド

「5.2.1 検索」応答電文：serviceType を追加、階層を修正
「5.3.2 LINE Pay の結果通知電文」の表中の通知件数分の項番を（4～10）に修正
「6-2 検索（Search）に関する補足（詳細コマンドタイプ/詳細オーダー決済状態）」の決済中止の successDetailTxnType を QuitAuth→Init に修正

第2章 LINE Pay サービス概要

2-1 LINE Pay サービスの概要

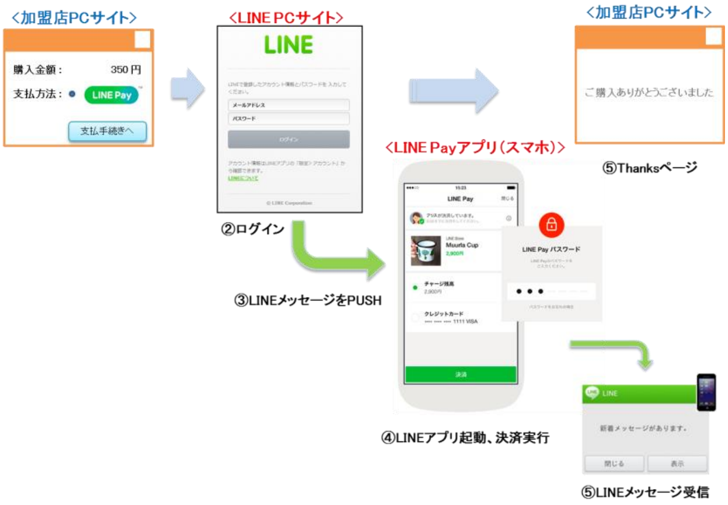
LINE Pay サービスは、加盟店様のサイトで購入された商品の代金を、LINE のユーザ ID と支払い用の LINE Pay パスワードで決済できるサービスです。LINE Pay チャージ残高で決済ができます。

- LINE Pay 利用登録を行った LINE ユーザが利用できる、パスワードのみの簡単な支払い方法です。
- 事前にチャージした、LINE Pay チャージ残高による支払いが可能です。
    - チャージ方法は銀行口座からのチャージ、LINE Pay カードやアプリの QR コード/バーコードを利用した店舗レジでのチャージ、ATM でのチャージなど様々な方法が提供されています。利用可能なチャージ方法やチャージ手順については LINE Pay サイトの情報やアプリをご参照ください。
- LINE アプリをインストールしたスマートフォンで決済を行います。
    - PC サイトから決済申込みを行い、決済実行はスマートフォンを利用するという形態も可能です。
    - 店舗のレジで、LINE アプリのバーコード/QR コードを利用した決済も可能です。(LINE ver5.1 からサポートされます。)
- 商材はデジタルコンテンツ/物販に対応しています。

決済の画面遷移のイメージを以下に示します。

(スマートフォンで決済)

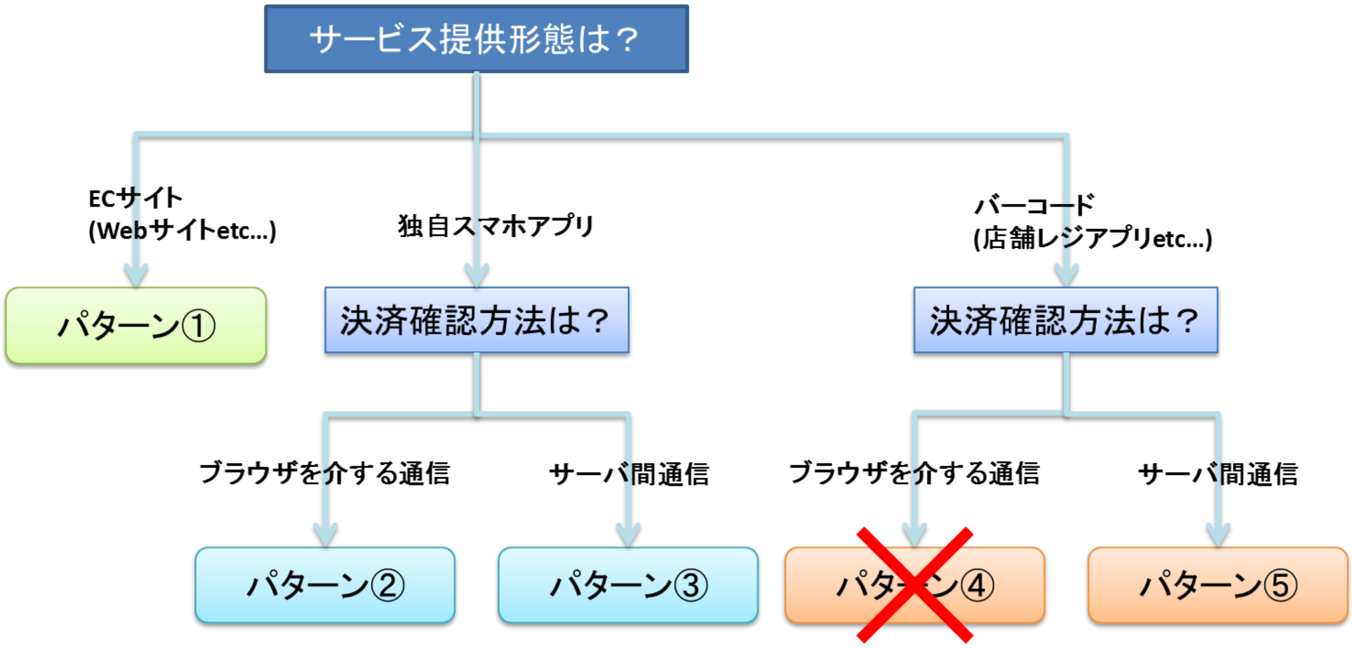
(PC＋スマートフォンで決済)



（店舗（レジ）＋スマートフォンで決済）

2-2 LINE Pay サービスのパターン

LINE Pay サービスでご提供できるサービスのパターン例を以下に示します。



パターン例の補足

パターン①　：　主に消費者が Web ブラウザを利用して、決済申込をするパターン。
PC の Web ブラウザもスマートフォンの Web ブラウザも同様になります。

パターン②　：　加盟店様独自アプリをご利用で、アプリ内ブラウザ（WebView）にて決済確認を行うパターン

パターン③　：　加盟店様独自アプリをご利用で、結果通知（もしくは検索コマンド）にて決済確認を行うパターン

パターン④　：　店舗のレジなどで、消費者のアプリからバーコードを読み取り、レジアプリ内のブラウザ（WebView）にて決済確認を行うパターン**（※本機能は、現在ご利用いただけません。）**

パターン⑤　：　店舗のレジなどで、消費者のアプリからバーコードを読み取り、結果通知（もしくは検索コマンド）にて決済確認を行うパターン

上記のパターン例ごとの本ドキュメントの関連について、下表に示します。

〇:必須　　△:ご利用の方式によっては必要　　×:不要

章番号	パターン①	パターン②	パターン③	パターン④	パターン⑤
3-1	〇				
3-2					
決済申込時の処理	PC＋スマートフォン、スマートフォン	スマートフォンアプリから LINE Pay に連動 ※ブラウザを介するケース	スマートフォンアプリから LINE Pay に連動 ※サーバ間通信のケース	店舗(レジ)から LINE Pay に連動 ※ブラウザを介するケース	店舗(レジ)から LINE Pay に連動 ※サーバ間通信のケース
売上処理	〇				
キャンセル処理	〇				
4-1	△(※1)				
5-1					
5.1.1	〇	〇	〇 ※「決済確認方法」を指定する	〇 ※「ワンタイムキー」を指定する	〇 ※「ワンタイムキー」「決済確認方法」を指定する
5.1.2	〇	〇	×	〇	×
5.1.3	〇				
5.1.4	〇				
5-2	△(※2)				
5-3	△(※1)				
6-1	〇				
6-2	△(※2)				
6.2.1	×	×	△	×	△
6-3					
6.3.1	△(※3)	〇	×	〇	×
6.3.2	×	〇	×	〇	×
6.3.3	△(※4)	△(※4)	×	△(※4)	×
7-1					
7.1.1	〇	×	×	×	×
7.1.2	×	〇	〇	〇	×
7-2	×	×	×	×	〇
7-3	〇				

※1:結果通知をご利用の場合　　※2:検索コマンドをご利用の場合

※3:利用ブラウザを指定する場合

※4:申込時のブラウザ(アプリ内ブラウザ含む)と決済完了時のブラウザが異なると問題がある場合(例:セッションをご利用の場合など。)

第3章 MDK の処理概要

3-1 機能一覧（決済要求コマンド）

使用可能な決済要求コマンドを下記表に示します。

決済要求種類 /コマンド	デジタルコンテンツ	物販	概要
申込（与信） ／Authorize	○	○	決済の申し込み（オーソリ）を要求します。
申込（与信＋売上） ／Authorize	○	○	決済の申し込み（オーソリ）と売上を同時に要求します。
売上 ／Capture	○	○	申込済み取引の売上確定を要求します。 オーソリ時の金額を超えない範囲で、売上金額を指定できます（部分売上）。
キャンセル ／Cancel	○	○	決済を取消（キャンセル）します。 売上前の取消については、与信（オーソリ）の取消となります。 売上後の取消については、売上金額を超えない範囲で、取消金額を指定できます（部分取消）[※1]。

※1 部分取消は、決済金額の残りが 0 円になるまで繰り返し実行できます。

3-2 決済処理シーケンス

LINE Pay サービスでは、消費者のアクセス環境（PC, スマートフォン）や決済方法（EC サイト, 店舗）、決済確認方法（ブラウザ経由, サーバ間通信）によって LINE Pay への連動フローが異なります。
各フローの詳細については、以下をご参照ください。

■ EC サイトを利用した決済フロー

- ➢ EC サイトを利用した決済フローについては、アクセス環境（PC, スマートフォン）によって少し異なります。
（加盟店様サイトと決済サーバーの間のインターフェイスについて、違いはほとんどありません。）
  - ✧ PC からのアクセスでは、消費者は PC のブラウザで LINE の認証を行った後、スマートフォンの LINE アプリ（LINE Pay アプリ）で決済を実行します。その後、PC のブラウザで自動的に加盟店様の決済完了画面に遷移します。
  - ✧ スマートフォンからのアクセスでは、加盟店様のサイトから LINE Pay の決済 URL に遷移後に、自動的に起動した LINE アプリで決済を実行します。決済実行後は、LINE アプリがブラウザを起動し、加盟店様の決済完了画面に遷移します。

■ 加盟店様独自アプリを利用した決済フロー

- ➢ 加盟店独自のスマートフォンアプリから LINE Pay に連動することもできます。本章では概略フローのみを示しますので、詳しくは「6-3 スマートフォン向けのサイトおよび独自アプリと LINE Pay アプリとの連携について」をご参照ください。

■ 店舗等でバーコードや QR コードを利用した決済フロー

- ➢ 加盟店様の店舗のレジ等で消費者のスマートフォンの LINE Pay アプリに表示されているバーコード（QR コード）を読取り、そのコード値を MDK のリクエストパラメータに設定して決済申込みをすることができます。（バーコード/QR コードは LINE5.1 バージョンからサポートされます。）

- ◆ 各決済フローにおいて、加盟店様側で決済が成功したかどうかを確認する方法を選択できます。
  - ✓ ブラウザを介する決済確認・・・主に、EC サイトで決済をする場合や、加盟店独自アプリ（店舗レジアプリを含む）でアプリ内ブラウザを利用されている場合。
  - ✓ サーバ間通信での決済確認・・・主に、加盟店独自アプリ（店舗レジアプリを含む）でブラウザを利用しない場合。
- ◆ 決済の成立には、必ず LINE アプリでの消費者の操作が必要になりますが、決済サーバーから LINE Pay システムへの決済 Reserve 要求後、消費者が LINE アプリの操作を完了するまでに 20 分以上経過すると、LINE Pay システム側で当該取引は無効となりますので、20 分以内に決済して頂くよう消費者にご案内下さい。

【LINE Pay：決済申込時の処理（PC＋スマートフォン）】

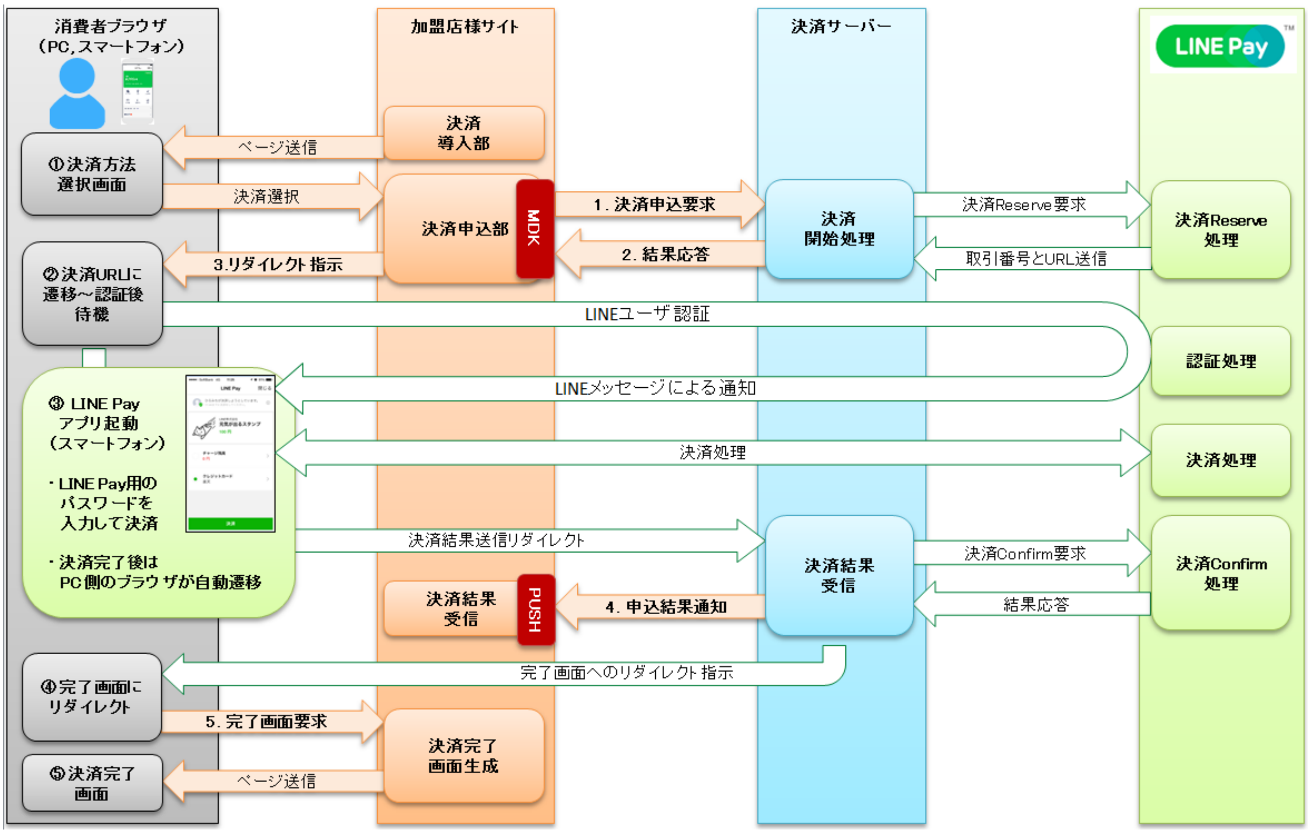


図 3-2-1 MDK 利用時システム処理概要図（PC による LINE Pay 申込）

【LINE Pay：決済申込時の処理（スマートフォン）】

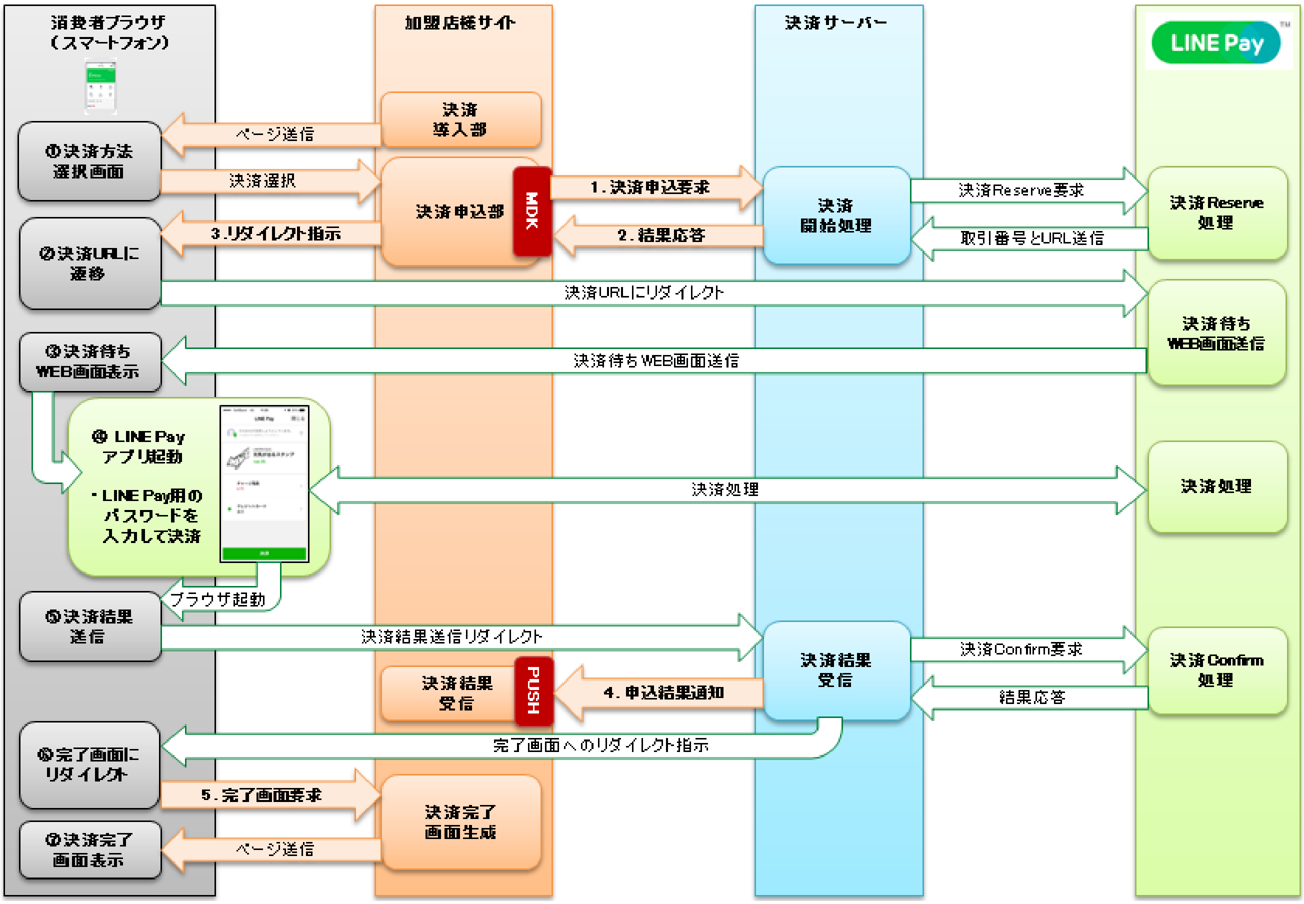


図 3-2-2 MDK 利用時システム処理概要図（スマホによる LINE Pay 申込）

No.	基本機能	処理説明
1	決済申込要求	EC サイトに渡された決済申込情報を電文化し、決済サーバーに送信します。
2	結果応答の受信	決済サーバーから返戻された結果を受信します。結果が成功（決済可能）の場合、「3. リダイレクト指示」を実施します。消費者ブラウザに返すリダイレクト先の URL が決済サーバーより返戻された結果に含まれます。
3	リダイレクト指示	決済サーバーにより返戻されたリダイレクト先の URL（決済 URL）を用いて、消費者ブラウザに LINE Pay の決済 URL へのリダイレクト指示を送信します。(※1)
4	申込結果通知(PUSH)	この通知は、決済申込により与信が成功した場合のみ通知されます。 リクエスト時に指定した PUSH 通知先 URL（未指定の場合は、MAP（Merchant Administration Portal）により設定した値）に、決済サーバーより POST で通知されますので、決済結果を受け取り、注文データ等に反映します。 **完了画面要求は、消費者の端末（PC やスマートフォン）のネットワーク環境の問題や、誤操作（ブラウザやタブを閉じてしまう等）により、正しく行われない可能性がありますので、必ずこの通知を受信してください。** なお、この通知と完了画面要求のリダイレクト結果が加盟店に届く順番は保証されていません。どちらが先に届いても問題とならないようにご対応ください。
5	完了画面要求	消費者ブラウザから結果を受信し、完了画面を送信します。 消費者側の状況により通信断や誤ってブラウザを閉じる、といったケースではこのシーケンスは発生しませんが、4.申込結果通知（PUSH）を受信していれば、決済結果を取りこぼすことはありません。

※1 HTTP レスポンスヘッダ（Location ヘッダ）にリダイレクト先 URL を設定し、HTTP ステータスコード＝302 でリダイレクトするか、JavaScript を利用してリダイレクト先 URL に自動遷移させてください。

【LINE Pay：決済申込時の処理（スマートフォンアプリから LINE Pay に連動） ※ブラウザを介するケース】

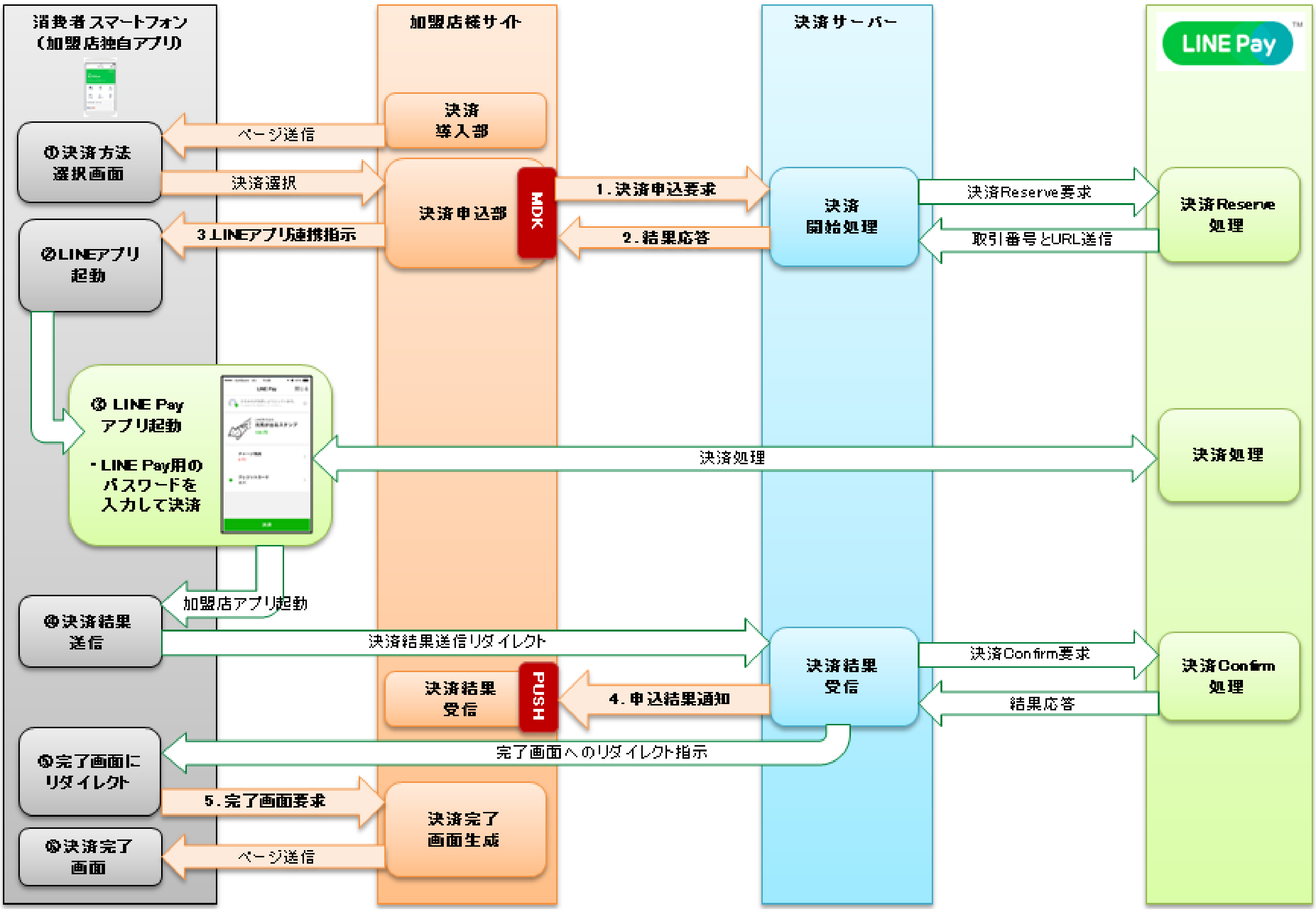


図 3-2-3 MDK 利用時システム処理概要図（スマホアプリによる LINE Pay 申込 ※ブラウザを介するケース）

No.	基本機能	処理説明
1	決済申込要求	加盟店独自のスマートフォンアプリから EC サイトに渡された決済申込情報を電文化し、決済サーバーに送信します。
2	結果応答の受信	決済サーバーから返戻された結果を受信します。結果が成功（決済可能）の場合、「3.LINE アプリ連携指示」を実行します。LINE アプリを起動するための URL スキームが決済サーバーより返戻された結果に含まれています。
3	LINE アプリ連携指示	決済サーバーにより返戻された LINE アプリを起動するための URL スキームをスマートフォン上で動作する加盟店アプリに送信します。 加盟店アプリでは、この URL スキームを利用して LINE アプリを起動します。 LINE アプリ（LINE Pay アプリ）で決済が行われると、再度加盟店アプリが呼び出されますので、アプリ起動時に引き渡された決済サーバーの URL に結果を送信します。
4	申込結果通知(PUSH)	この通知は、決済申込により与信が成功した場合のみ通知されます。 リクエスト時に指定した PUSH 通知先 URL（未指定の場合は、MAP（Merchant Administration Portal）により設定した値）に、決済サーバーより POST で通知されますので、決済結果を受け取り、注文データ等に反映します。 **完了画面要求は、消費者の端末（PC やスマートフォン）のネットワーク環境の問題や、誤操作（ブラウザやタブを閉じてしまう等）により、正しく行われない可能性がありますので、必ずこの通知を受信してください。** なお、この通知と完了画面要求のリダイレクト結果が加盟店に届く順番は保証されていませ

		ん。どちらが先に届いても問題とならないようにご対応ください。
5	完了画面要求	加盟店アプリから結果を受信し、完了画面を送信します。 消費者側の状況により通信断や誤ってアプリを閉じる、といったケースではこのシーケンスは発生しませんが、4.申込結果通知（PUSH）を受信していれば、決済結果を取りこぼすことはありません。

【LINE Pay：決済申込時の処理（スマートフォンアプリから LINE Pay に連動） ※サーバ間通信のケース】

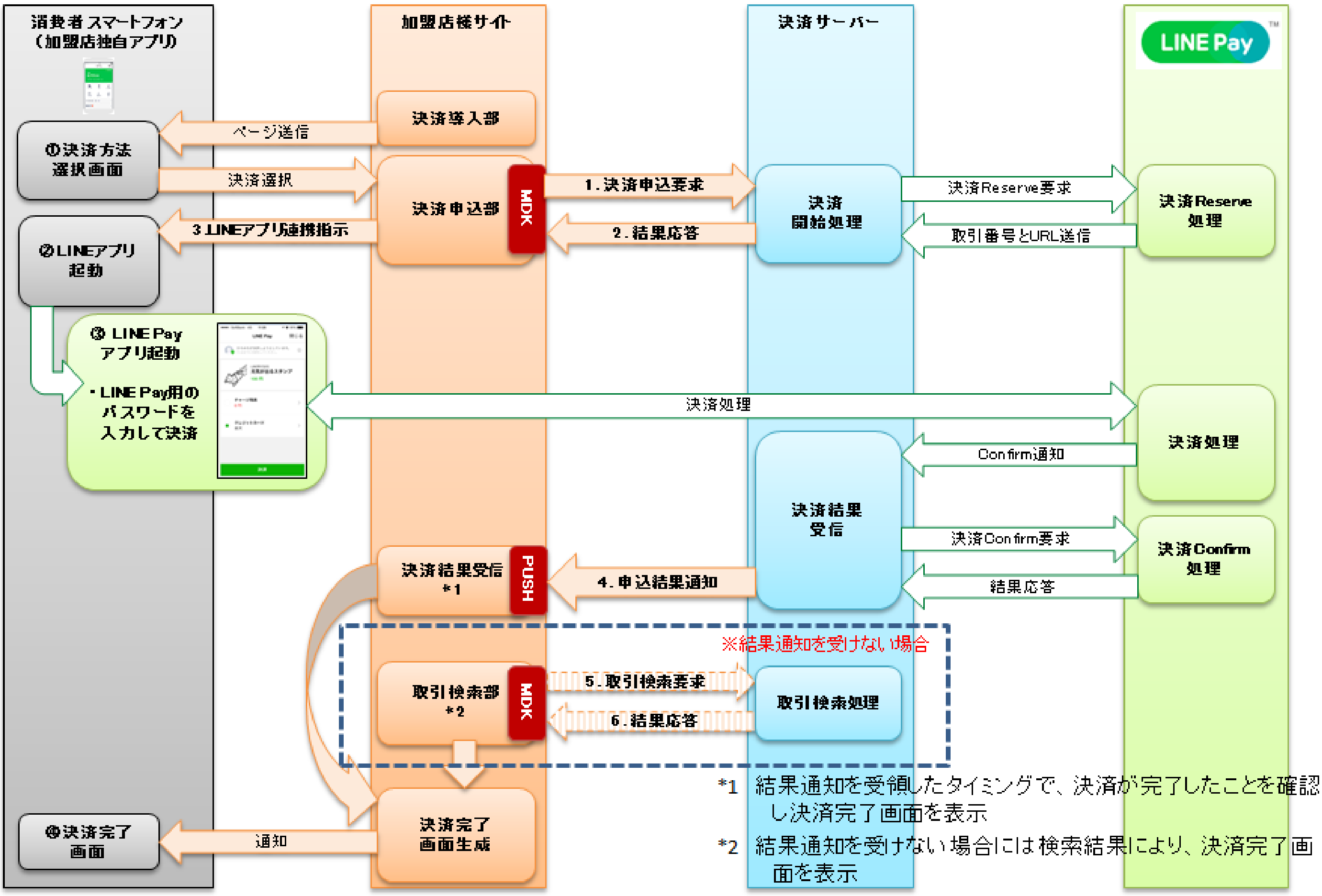


図 3-2-4 MDK 利用時システム処理概要図（スマホアプリによる LINE Pay 申込　※サーバ間通信のケース）

No.	基本機能	処理説明
1	決済申込要求	加盟店独自のスマートフォンアプリから EC サイトに渡された決済申込情報を電文化し、決済サーバーに送信します。
2	結果応答の受信	決済サーバーから返戻された結果を受信します。結果が成功（決済可能）の場合、「3.LINE アプリ連携指示」を実行します。LINE アプリを起動するための URL スキームが決済サーバーより返戻された結果に含まれています。
3	LINE アプリ連携指示	決済サーバーにより返戻された LINE アプリを起動するための URL スキームをスマートフォン上で動作する加盟店アプリに送信します。 加盟店アプリでは、この URL スキームを利用して LINE アプリを起動します。
4	申込結果通知(PUSH)	この通知は、決済申込の結果が通知されます。（成功、失敗共に通知されます。） リクエスト時に指定した PUSH 通知先 URL（未指定の場合は、MAP（Merchant Administration Portal）により設定した値）に、決済サーバーより POST で通知されますので、決済結果を受け取り、注文データ等に反映し完了処理を行います。
5,6	取引検索要求、結果応答	申込結果通知を受領しない場合、決済申込をした取引を検索し、決済の状態を確認します。 まだ決済が完了（成功又は失敗）していない場合は、再度確認します。 決済が完了（成功又は失敗）している場合は、注文データなどに反映し完了処理を行います。 ※決済完了の確認方法は、「6.2.1 決済確認方式：サーバ間通信の場合の検索結果について」を参照してください。

【LINE Pay：決済申込時の処理（店舗（レジ）から LINE Pay に連動） ※ブラウザを介するケース】

（※本機能については、現在ご利用頂けません。）

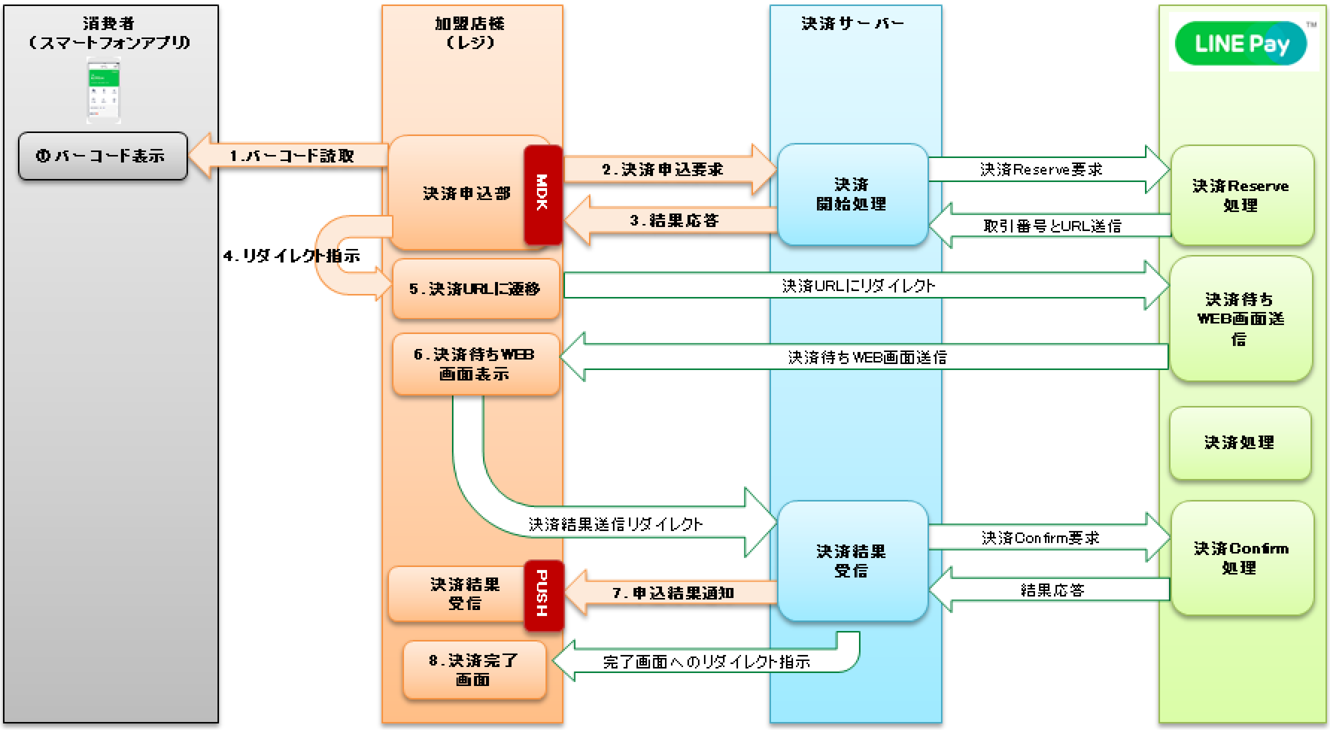


図 3-2-5 MDK 利用時システム処理概要図（店舗（レジ）での LINE Pay 申込 ※ブラウザを介するケース）

No.	基本機能	処理説明
1	バーコード読取	加盟店様のレジアプリ等で、消費者の LINE アプリに表示されているバーコードを読み取ります。
2	決済申込要求	加盟店様で取得したバーコードと決済申込情報を電文化し、決済サーバーに送信します。
3	結果応答の受信	決済サーバーから返戻された結果を受信します。結果が成功（決済可能）の場合、「4. リダイレクト指示」を実施します。レジアプリの内部ブラウザでリダイレクトする先の URL が決済サーバーより返戻された結果に含まれます。
4	リダイレクト指示	決済サーバーにより返戻されたリダイレクト先の URL（決済 URL）を用いて、レジアプリの内部ブラウザに LINE Pay の決済 URL へのリダイレクト指示を送信します。（※1）
5	決済 URL に遷移	LINE Pay 側の画面に遷移し、決済が完了するまで待機画面となります。
6	決済待ち WEB 画面表示	決済が完了すると待機画面から決済サーバーにリダイレクトします。
7	申込結果通知(PUSH)	この通知は、決済申込により与信が成功した場合のみ通知されます。 リクエスト時に指定した PUSH 通知先 URL（未指定の場合は、MAP（Merchant Administration Portal）により設定した値）に、決済サーバーより POST で通知されますので、決済結果を受け取り、注文データ等に反映します。
8	完了画面要求	加盟店アプリから結果を受信し、完了画面を送信します。 消費者側の状況により通信断や誤ってアプリを閉じる、といったケースではこのシーケンスは発生しませんが、7.申込結果通知（PUSH）を受信していれば、決済結果を取りこぼすことはありません。

※1 HTTP レスポンスヘッダ（Location ヘッダ）にリダイレクト先 URL を設定し、HTTP ステータスコード＝302 でリダイレクトするか、JavaScript を利用してリダイレクト先 URL に自動遷移させてください。

【LINE Pay：決済申込時の処理（店舗（レジ）から LINE Pay に連動）　※サーバ間通信のケース】

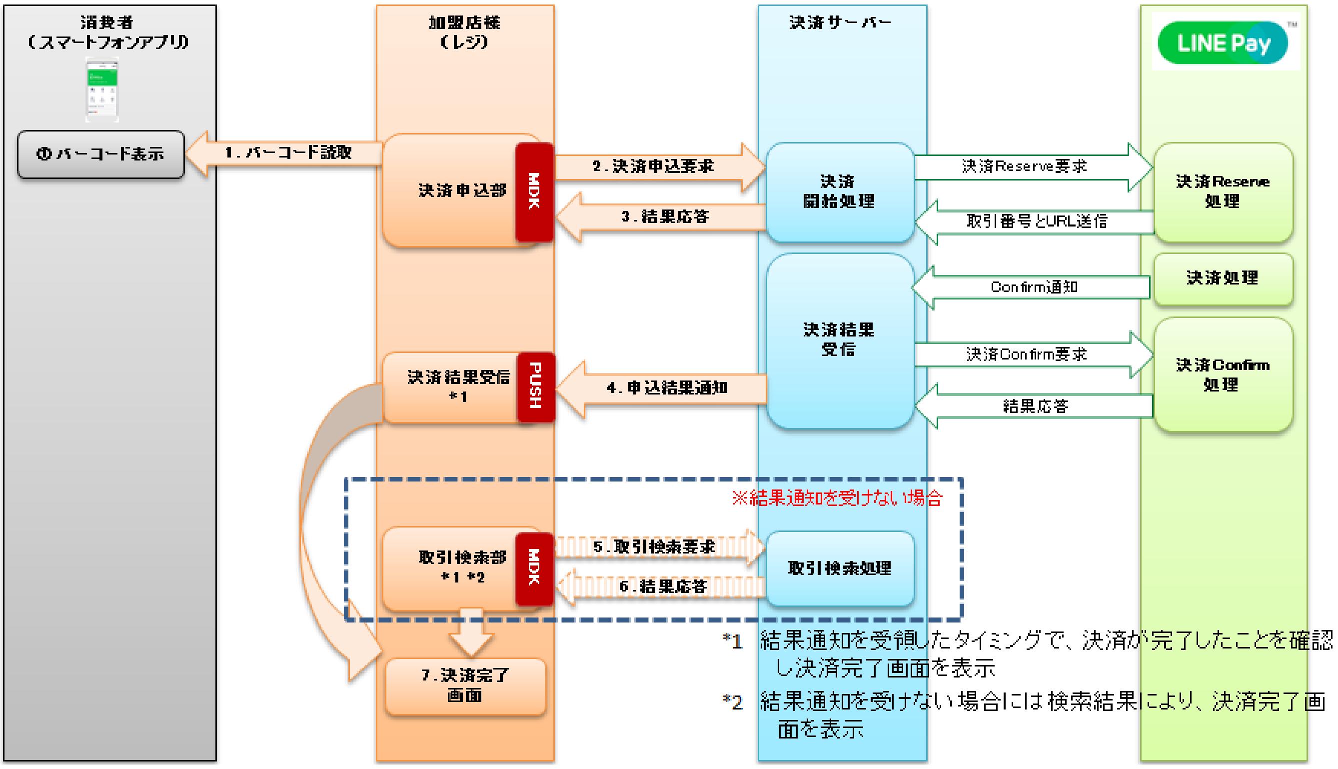


図 3-2-6 MDK 利用時システム処理概要図（店舗（レジ）での LINE Pay 申込　※サーバ間通信のケース）

No.	基本機能	処理説明
1	バーコード読取	加盟店様のレジアプリ等で、消費者の LINE アプリに表示されているバーコードを読み取ります。
2	決済申込要求	加盟店様で取得したバーコードと決済申込情報を電文化し、決済サーバーに送信します。
3	結果応答の受信	決済サーバーから返戻された結果を受信します。結果が成功（決済可能）の場合、LINE Pay と消費者の間で認証＆決済処理が行われているため、加盟店様側のレジアプリ等では決済待機画面を表示します。
4	申込結果通知(PUSH)	この通知は、決済申込の結果が通知されます。（成功、失敗共に通知されます。） リクエスト時に指定した PUSH 通知先 URL（未指定の場合は、MAP（Merchant Administration Portal）により設定した値）に、決済サーバーより POST で通知されますので、決済結果を受け取り、注文データ等に反映し完了処理を行います。
5,6	取引検索要求、結果応答	申込結果通知を受領しない場合、決済申込をした取引を検索し、決済の状態を確認します。 まだ決済が完了（成功又は失敗）していない場合は、再度確認します。 決済が完了（成功又は失敗）している場合は、注文データなどに反映し完了処理を行います。 ※決済完了の確認方法は、「6.2.1 決済確認方式：サーバ間通信の場合の検索結果について」を参照してください。
7	決済完了画面	検索結果や結果通知により、決済完了の確認が取れた場合、完了画面を表示します。

【LINE Pay：売上処理】

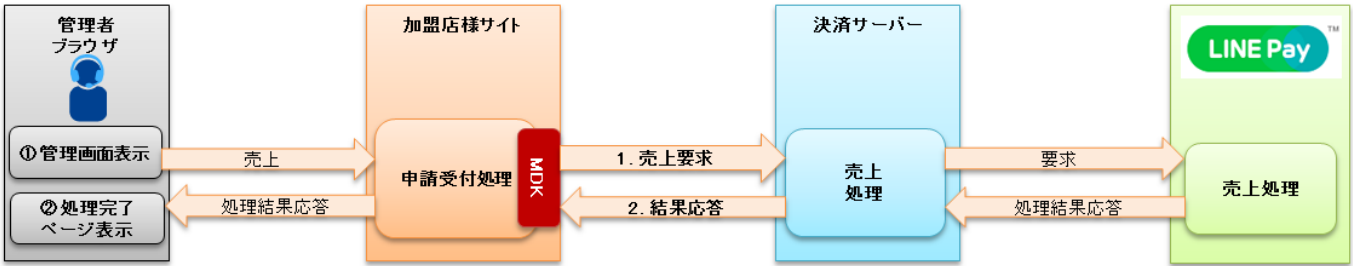


図 3-2-7 MDK 利用時システム処理概要図（LINE Pay 売上要求）

No.	基本機能	処理説明
1	売上要求	売上の対象となる取引情報を電文化し、決済サーバーに送信します。
2	結果応答	決済サーバーより結果応答を受信します。

【LINE Pay：キャンセル処理】

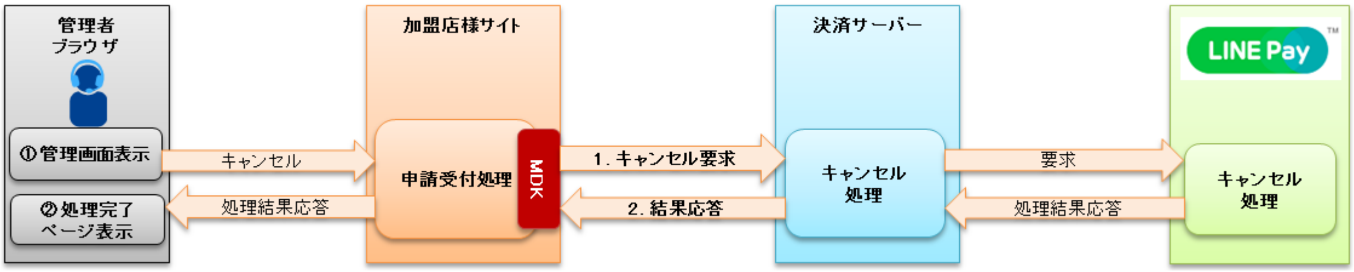


図 3-2-8 MDK 利用時システム処理概要図（LINE Pay キャンセル要求）

No.	基本機能	処理説明
1	取消要求	キャンセルの対象となる取引情報を電文化し、決済サーバーに送信します。
2	結果応答	決済サーバーより結果応答を受信します。

第4章 結果通知受信処理

4-1 結果通知受信処理の対象

LINE Pay の結果通知受信処理の対象は以下の機能です。

決済サービス名	決済申込（与信/与信売上）	売上	キャンセル
LINE Pay	○	－	－

表 1 通知対象一覧

通信電文仕様（共通部）および店舗受信後応答仕様は、『開発ガイド』をご参照ください。

LINE Pay 固有の通知電文インターフェイスにつきましては、「5.3.2 LINE Pay の結果通知電文」をご参照ください。

第5章　インターフェイス詳細

ここでは LINE Pay サービスで利用する MDK インターフェイス項目を説明します。

■「設定」欄の内容は以下の通りです。

要求電文 … 必須項目：○　任意項目：△　設定不可：×　その他条件付：※、※n（条件は説明欄、または欄外に記入）

応答電文 … 必ず返戻：○　処理成功時のみ返戻：△　返戻なし：×　その他条件付：※

■orderId（取引 ID）について

店舗で任意に採番してください。申込処理毎に付ける必要があります。他の取引 ID と重複しないよう採番してください。他決済サービスとも重複できません。

また、テスト取引で使用した取引 ID を、本番取引で再度使用することはできません。

取引 ID には、半角英数字以外に"-"（ハイフン）、"_"（アンダースコア）も使用可能です。

■応答電文について

応答電文には、本書に記載されていないフィールド（パラメータ）も存在しますが、通常は、記載されているフィールド以外は加盟店様の方で意識する必要はございません。

■「設定」欄のタイトルについて

設定欄のタイトル（B/S）は、決済確認方法を示しています。決済確認方法によって、設定可能な項目が異なりますのでご注意ください。

B：ブラウザを介した通信　S：サーバ間通信

5-1 LINE Pay

5.1.1 申込

要求電文　：　LinepayAuthorizeRequestDto

フィールド名	項目名	書式・制限	説明	設定 B	設定 S
orderId	取引 ID	半角英数字 100 桁以内	店舗側で採番した取引 ID を指定	○	○
amount	決済金額	半角数字 7 桁以内	決済金額（合計金額）を指定	○	○
withCapture	売上フラグ	右記参照	"true"： 与信同時売上 "false"： 与信のみ （デフォルト値） ※指定が無い場合は、与信のみとなります。	△	△
itemId	商品 ID	半角英数字 64 桁以内	店舗側で発番する、商品またはサービスの管理用番号を指定 ・ LINE Pay システムには連携されません。	△	△
itemName	商品名	全角半角 4000byte 以内	商品名を指定 ※LINE が提供する消費者向けの画面やメール本文に表示されます。	○	○
itemImageUrl	商品画像 URL	URL に使用可能な半角文字 256 桁以内	決済画面に表示される商品画像の URL を指定。 サイズ：84×84	△	△

要求電文 ： LinepayAuthorizeRequestDto

フィールド名	項目名	書式・制限	説明	設定	
				B	S
checkUseBrowser	使用ブラウザ判定	右記参照	"true"： 決済を要請した browser と LINE アプリから遷移する browser が異なる場合、LINE Pay で決済を要請した browser に戻るよう案内ページを提供する。 "false"： LINE Pay で決済を要請した browser と LINE アプリから遷移する browser の確認をしない ※指定がない場合は、"false"となります。 ※詳しくは、「6.3.3checkUseBrowser の利用方法」を参照してください。	△	△
appUrlScheme	アプリ起動 URL スキーム	URL に使用可能な半角文字 256 桁以内	LINE Pay アプリが起動するブラウザまたは独自アプリケーションを起動するための URL スキーム。 スマートフォン用のサイトまたはスマートフォン上で動作する独自アプリケーションから LINE Pay を利用する際には指定して下さい。 LINE Pay アプリで決済後に決済サーバーに遷移するための URL を、決済サーバー側で付与するためのプレースホルダ"<url>"を埋め込んだ URL スキームを指定して下さい。 ※詳しくは、「6.3.1appUrlScheme の指定方法」を参照してください。	※	×
useOriginalApp	独自アプリ起動時のオプション指定	右記参照	"0"： オプション指定なし（デフォルト値） "1"： <url>を URL エンコーディングする このオプションを指定した場合は、appUrlScheme に指定したプレースホルダ<url>に決済サーバー側で URL エンコーディングした URL 文字列をセットします。	△	×
mid	LINE member ID	半角英数字 50 桁以内	LINE ユーザを特定する固有 ID。 ※現在は使用できません。	×	×
packageName	packageName	半角英数字 4000 桁以内	Android でアプリ起動時に指定する packageName。 ※詳しくは、「6.3.1 appUrlScheme の指定方法」を参照してください。	△	×
successUrl	決済完了時 URL	URL に使用可能な半角文字 256 桁以内	決済成功時に、店舗側サイトに画面遷移を戻すための URL を指定（クエリパラメータ指定可） ※未指定の場合は、MAP（Merchant Administrator Portal）から登録申請した値を使用。	△	×
cancelUrl	決済キャンセル時 URL	URL に使用可能な半角文字 256 桁以内	決済キャンセル時に、店舗側サイトに画面遷移を戻すための URL を指定（クエリパラメータ指定可） ※未指定の場合は、MAP（Merchant Administrator Portal）から登録申請した値を使用。	△	×
errorUrl	決済エラー時 URL	URL に使用可能な半角文字 256 桁以内	決済エラー時に、店舗側サイトに画面遷移を戻すための URL を指定（クエリパラメータ指定可） ※未指定の場合は、MAP（Merchant Administrator Portal）から登録申請した値を使用。	△	×
pushUrl	結果通知受信 URL	URL に使用可能な半角文字 256 桁以内	結果通知のプッシュ URL を指定（クエリパラメータ指定可） ※未指定の場合は、MAP（Merchant Administration Portal）により設定した値を使用	△	△

要求電文 ： LinepayAuthorizeRequestDto

フィールド名	項目名	書式・制限	説明	設定	
				B	S
oneTimeKey	ワンタイムキー	半角英数字 19 桁以内	店舗レジなどで QR コードもしくはバーコード読み取りによる決済を実行する場合に指定します。 LINE Pay アプリで提供する QR/BarCode を取得し、その値を指定してください。この値の有効期間は 5 分間で、有効期間内に申込みをしなければなりません。 また、決済成立後に同じ値を設定した場合や、不正な値を設定した場合には、IG08 を返却します。 QR/BarCode は LINE 5.1 バージョンからサポートされます。 ※この項目を指定する場合、paymentConfirmType は、”1”：サーバ間通信を指定してください。	△	△
paymentConfirmType	決済確認方法	右記参照	決済確認の方式を指定します。 ”0”： ブラウザを介する通信（デフォルト値） ”1”： サーバ間通信	△	○

応答電文 ： LinepayAuthorizeResponseDto

フィールド名	項目名	書式・制限	説明	設定
serviceType	決済サービスタイプ	半角英数字 10 桁以内	要求電文を送信した決済サービスタイプ	○
mstatus	処理結果コード	半角英数字 32 文字以内	”success”：正常終了 ”failure”：異常終了	○
vResultCode	詳細結果コード	半角英数字 16 文字	処理の結果を詳細に表すコード 4 桁ずつ 4 つのブロックで構成され、各ブロックでサービス毎の処理結果を表します。 詳細は『結果コード一覧』を参照下さい。	○
merrMsg	エラーメッセージ	文字列 1024 バイト以内	処理結果を日本語で表示します。	○
marchTxn	電文 ID	文字列 100 桁以内	決済サーバーにて決済処理電文（内部処理も含む）毎に付与する ID 1つの取引 ID に対して、複数の ID が付与されます。	○
orderId	取引 ID	半角英数字 100 文字以内	決済要求時に店舗様にて任意に採番し送信された取引 ID	○
linepayOrderId	LINEPay 取引番号	半角英数字 19 文字以内	LINEPay システム側で発番された取引番号 ※mstatus=failure の場合は設定されません。	△
custTxn	取引毎に付く ID	文字列 100 桁以内	決済サーバーがオーダー（取引 ID）と紐付ける為に採番する ID	○
txnVersion	MDK バージョン	半角英数字 5 桁	電文のバージョン 問題発生時などに用いますが、通常ご利用になることはありません。	○
redirectWebUrl	リダイレクト用 WebURL	文字列	LINE Pay が提供する決済 URL。 PC から決済申込を行った場合、またはスマートフォン向けの WEB サイトから（加盟店独自のアプリを起動しないで）決済申込を行った場合は、こちらの URL にそのまま遷移させます。POST、GET のどちらも利用可能です。 ※要求電文にて oneTimeKey かつ paymentConfirmType=”1” を設定した場合や、処理失敗時には設定されません。	※

応答電文 ： LinepayAuthorizeResponseDto				
フィールド名	項目名	書式・制限	説明	設定
redirectAppUrl	LINE にアプリ間移動するための URL（URL Scheme）	文字列	LINE Pay アプリの決済画面へ移動する URL Scheme。 加盟店独自のアプリから LINE Pay 決済画面に移動する場合はこちらの URL を使用します。 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、この URL で LINE アプリに移動してください。 詳しくは、「6-3 スマートフォン向けのサイトおよび独自アプリと LINE Pay アプリとの連携について」をご参照ください。 ※要求電文にて oneTimeKey かつ paymentConfirmType="1" を設定した場合や 、処理失敗時には設定されません。	※

5.1.2 消費者ブラウザから店舗側の完了画面へリダイレクトする時のクエリパラメータ

※申込のリクエストで、paymentConfirmType="1"（サーバー間通信）の場合はリダイレクトしません。

リダイレクト受信内容：決済サーバーから消費者ブラウザ経由で店舗へリダイレクト(GET)される内容				
フィールド名	項目名	書式・制限	説明	設定
mstatus	処理結果コード	半角英数字 32 文字以内	"success" ：正常終了 "failure" ：異常終了 "pending" ：保留	○
vResultCode	詳細結果コード	半角英数 4 桁	処理の結果を詳細に表すコード 詳細は『結果コード一覧』を参照下さい。	○
orderId	取引 ID	半角英数字 100 文字以内	決済要求時に店舗様にて任意に採番し送信された取引 ID	○
txnType	トランザクションタイプ	文字列 右記参照	通知対象となるトランザクションのタイプが設定されます。 "Authorize"	○
linepayOrderId	LINEPay 取引番号	半角英数字 19 文字以内	LINEPay システム側で発番された取引番号 ※mstatus=failure, pending の場合は設定されません。	△
vAuthInfo	改ざんチェック用 ハッシュ値	文字列 右記参照	下記文字列を連結し、SHA-256 によって算出したハッシュ値 ・マーチャント CCID ・パラメータ値の連結文字列（authParams が示す順序で連結） ・パスワード 尚、パラメータ値を連結する際は、パラメータ名や区切り文字は含めず、パラメータ値のみを連結しています。連結した文字列をバイナリに変換する際の文字エンコーディングは、UTF-8 を使用しています。	○
authParams	ハッシュ値算出 パラメータ順序	文字列 右記参照	vAuthInfo のハッシュ値を算出する元とした文字列の、パラメータの連結順序を示す値 パラメータ名のカンマ区切り文字列を Base64 エンコードしています。デコードを行うと文字列が復元されます。 例） "orderId,vResultCode,mstatus" "mstatus,orderId,vResultCode" （順序は固定ではないため、リクエスト受信のたびに動的に処理する必要があります。）	○

（重要）

✧ これらのパラメータはブラウザを経由して連携されるため、画面遷移が正常に行われないケースでは取得することができません。そのため、決済サーバーから直接送信される「5-3 結果通知電文」の受信と組み合わせて確実に結果を取得するようにしてください。

✧ vAuthInfo および authParams を利用した改ざんチェックについて

決済サーバーから消費者ブラウザを経由して店舗側の結果画面に遷移(リダイレクト)しますが、ここで店舗側システムが受け取ったクエリパラメータが改ざんされていないことを検証するためのパラメータが、vAuthInfo と authParams です。

店舗側システムで算出したハッシュ値が、クエリパラメータより取得した vAuthInfo と一致している場合は、パラメータは改ざんされていない、とみなすことができます。

この改ざんチェックは、悪意を持った第三者によって、不正なリダイレクト電文を受信する可能性がありますので、実装を強く推奨しています。

実装方法の詳細につきましては、弊社より提供しているサンプルプログラムをご参照ください。

5.1.3 売上

要求電文 ： LinepayCaptureRequestDto

フィールド名	項目名	書式・制限	説明	設定
orderId	取引 ID	半角英数字 100 桁以内	売上対象の取引 ID を指定	○
amount	売上金額	半角数字 7 桁以内	売上金額を指定。 ※未指定時は全額売上となります。	△

応答電文 ： LinepayCaptureResponseDto

フィールド名	項目名	書式・制限	説明	設定
serviceType	決済サービスタイプ	半角英数字 10 桁以内	要求電文を送信した決済サービスタイプ	○
mstatus	処理結果コード	半角英数字 32 文字以内	″success″：正常終了 ″failure″：異常終了 ″pending″：保留	○
vResultCode	詳細結果コード	半角英数字 16 文字	処理の結果を詳細に表すコード 4 桁ずつ 4 つのブロックで構成され、各ブロックでサービス毎の処理結果を表します。 詳細は『結果コード一覧』を参照下さい。	○
merrMsg	エラーメッセージ	文字列 1024 バイト以内	処理結果を日本語で表示します。	○
marchTxn	電文 ID	文字列 100 桁以内	決済サーバーにて決済処理電文(内部処理も含む)毎に付与する ID 1つの取引 ID に対して、複数の ID が付与されます。	○
orderId	取引 ID	半角英数字 100 文字以内	決済要求時に店舗様にて任意に採番し送信された取引 ID	○
custTxn	取引毎に付く ID	文字列 100 桁以内	決済サーバーがオーダー(取引 ID)と紐付ける為に採番する ID	○
txnVersion	MDK バージョン	半角英数字 5 桁	電文のバージョン 問題発生時などに用いますが、通常ご利用になることはありません。	○
captureDatetime	売上日時	文字列 14 桁	YYYYMMDDhhmmss 形式	△
balance	残高	半角数字 7 桁以内	現在の決済金額を返します。	△

5.1.4 取消

要求電文 ： LinepayCancelRequestDto

フィールド名	項目名	書式・制限	説明	設定

orderId	取引 ID	半角英数字 100 桁以内	上記「orderId(取引 ID)について」参照	○
amount	減額金額	半角数字 7 桁以内	減額する金額を指定。 ※未指定時は全額取消となります。 ※売上が未成立の取引の場合は、全額のみ指定可です。	△

応答電文 ： LinepayCancelResponseDto

フィールド名	項目名	書式・制限	説明	設定
serviceType	決済サービスタイプ	半角英数字 10 桁以内	要求電文を送信した決済サービスタイプ	○
mstatus	処理結果コード	半角英数字 32 文字以内	″success″：正常終了 ″failure″：異常終了 ″pending″：保留	○
vResultCode	詳細結果コード	半角英数字 16 文字	処理の結果を詳細に表すコード 4 桁ずつ 4 つのブロックで構成され、各ブロックでサービス毎の処理結果を表します。 詳細は『結果コード一覧』を参照下さい。	○
merrMsg	エラーメッセージ	文字列 1024 バイト以内	処理結果を日本語で表示します。	○
marchTxn	電文 ID	文字列 100 桁以内	決済サーバーにて決済処理電文(内部処理も含む)毎に付与する ID 1つの取引 ID に対して、複数の ID が付与されます。	○
orderId	取引 ID	半角英数字 100 文字以内	決済要求時に店舗様にて任意に採番し送信された取引 ID	○
custTxn	取引毎に付く ID	文字列 100 桁以内	決済サーバーがオーダー(取引 ID)と紐付ける為に採番する ID	○
txnVersion	MDK バージョン	半角英数字 5 桁	電文のバージョン 問題発生時などに用いますが、通常ご利用になることはありません。	○
cancelDatetime	取消日時	文字列 14 桁	YYYYMMDDhhmmss 形式	△
balance	残高	半角数字 7 桁以内	現在の決済金額を返します。	△

5-2 共通

5.2.1 検索

■「設定」欄の内容は以下の通りとなります。

要求電文 … 必須項目：○　任意項目：△　設定不可：×　その他条件付：※（条件は説明欄に記入）

応答電文 … 必ず返戻：○　該当取引存在時に返戻：△　返戻なし：×　その他条件付：※

■複数指定は 0～の添字を指定します。

例）exparam.serviceTypeCd[0]=card&exparam.serviceTypeCd[1]=linepay

■ワイルドカードは値の一部と"*"を組み合わせて検索します。"*"のみの指定はできません。

例）exparam.searchParameters.common.orderId=123*

■通常の検索の他、マスタ情報取得が可能です。マスタ情報取得時のインターフェイスは後記します。

要求電文 ： SearchRequestDto

※ 以下は、共通の検索要求フィールドです。

検索フィールド名	検索項目名	書式・制限	複数指定	ﾜｲﾙﾄﾞｶｰﾄﾞ	説明	設定
requestId	リクエスト ID	半角英数字記号 128 文字以内			リクエスト ID を指定します。指定した場合は、それ以外のパラメータは指定できなくなります。	△
serviceTypeCd	決済サービスタイプ	右記参照	○		検索対象の決済を指定します。未指定の場合は、全決済が検索対象となります。 "linepay"： LINE Pay	△
newerFlag	最新トランザクションフラグ	右記参照			成功、失敗に関係なく、最新トランザクションのみを検索する場合に使用します。 "true"： 1 取引内の最新トランザクションのみ検索 "false"： 全てのトランザクションを検索 ※指定しない場合は"false"になります。	△
containDummyFlag	ダミー決済対象フラグ	右記参照			ダミー取引も検索する場合に使用します。 "true"： ダミー取引も検索する "false"： ダミー取引は検索しない ※指定しない場合は"false"になります。	△
maxCount	検索最大件数	1～1000			取得したい検索結果の最大件数です。未指定の場合、最大値となります。 ※短時間で大量の検索を繰り返すような処理はサーバに負荷が掛かりますのでご遠慮頂けますようお願いします。	△
common　共通						
orderId	取引 ID	半角英数字 100 桁以内		○	検索したい取引 ID を指定します。	△

	orderStatus	取引決済状態	右記参照	○		以下のいずれかを指定します。 ″ initial ″：初期状態 ″ end ″：終了 ″ end_presentation ″：画面遷移正常終了 ″ pending ″：保留 ″ validation_error ″：検証エラー ″ expired ″：期限切れ ″ error ″：エラー	△
	command	コマンド	右記参照	○		決済で使用するコマンドを指定します。 ″ Authorize ″：与信、申込 ″ Capture ″：売上 ″ Cancel ″：取消	△
	mstatus	ステータスコード	右記参照	○		決済結果として返戻されるステータスコードを指定します。 ″ success ″：成功 ″ failure ″：失敗 ″ pending ″：保留	△
	txnDatetime.from	取引日（From）	文字列 12 桁			取引日時の範囲（From）を指定します。 YYYYMMDDhhmm 形式	△
	txnDatetime.to	取引日（To）	文字列 12 桁			取引日時の範囲（To）を指定します。 YYYYMMDDhhmm 形式	△
	amount.from	金額（From）	数字 12 桁以内			決済金額の範囲（From）を指定します。	△
	amount.to	金額（To）	数字 12 桁以内			決済金額の範囲（To）を指定します。	△
※　以下は、LINE Pay 固有の検索要求フィールドです。							
linepay	LINE Pay						
	detailOrderType	詳細オーダー決済状態	右記参照	○		※「detailOrderType」の詳細は「6-2 検索（Search）に関する補足（詳細コマンドタイプ／詳細オーダー決済状態）」を参照	△
	detailCommandType	詳細コマンドタイプ	右記参照	○		※「detailCommandType」の詳細は「6-2 検索（Search）に関する補足（詳細コマンドタイプ／詳細オーダー決済状態）」を参照	△
	itemId	商品番号	半角英数字 64 桁以内		○	決済申込時に指定した商品番号	△

応答電文 ： SearchResponseDto

※ 以下は、共通の検索結果フィールドです。

検索フィールド名	検索項目名	書式・制限	説明	設定
result	処理結果	―		○
serviceType	サービスタイプ	右記参照	″search″	○
mstatus	処理結果コード	半角英数字 32 文字以内	処理の結果ステータスが格納されます。 “success”：正常終了 “failure”：異常終了	○
vResultCode	詳細結果コード	半角英数字 16 文字	処理の結果を詳細に表すコードとなります。 詳細は『結果コード一覧』を参照下さい。	○
merrMsg	エラーメッセージ	文字列 1024 バイト以内	処理結果を日本語で表示します。	△
overMaxCountFlag	最大件数超えフラグ	右記参照	検索対象データが要求電文で指定した検索最大件数より多いかどうかを表します。 ″true″： 最大件数以上 ″false″： 最大件数未満	△
searchCount	検索結果件数	0～1000	検索結果件数（オーダー件数）が格納されます。	△
orderInfos	オーダー情報リスト		複数のオーダー情報（orderInfo）が格納されます。	△
orderInfo	オーダー情報	―	検索条件に該当した取引の情報が該当件数分繰り返されます。0～1000 件（要求電文で指定した検索最大件数まで）となります。	△
index	インデックス	0～999	検索された情報のインデックスが格納されます。	△
serviceTypeCd	決済サービスタイプ	右記参照	決済の種類が格納されます。 ″linepay″： LINE Pay	△
orderId	取引 ID	文字列	取引の取引 ID が格納されます。	△
orderStatus	取引決済状態	右記参照	決済の状態が格納されます。 ″ initial ″：初期状態 ″ end ″：終了 ″ end_presentation ″：画面遷移正常終了 ″ pending ″：保留 ″ validation_error ″：検証エラー ″ expired ″：期限切れ ″ error ″：エラー ※このフィールドは、取引 ID に関連する注文データの状態を完全に表現するものではありません。サービスタイプによっては詳細な状態遷移を別のフィールドに保持している場合がありますので、店舗側システムの用途に合わせて successDetailTxnType や、決済固有の状態フィールドをご参照ください。	△
lastSuccessTxnType	最終成功トランザクションタイプ	文字列	直近の成功したコマンド名が格納されます。	△
successDetailTxnType	詳細トランザクションタイプ	文字列	取引の詳細な状態が格納されます。 検索要求電文の各決済の detailOrderType 参照。	△
properOrderInfo	固有オーダー情報	―	各決済サービスの固有オーダー情報が格納されます。後述の一覧参照。	△

transactionInfos	決済トランザクションリスト	－	複数の決済トランザクション情報(transactionInfo)が格納されます。	△
transactionInfo	決済トランザクション情報	－	検索条件に該当した取引の情報が該当件数分繰り返されます。	△
txnId	トランザクション管理 ID	文字列	決済サーバーが採番する管理 ID となります。	△
command	コマンド	文字列	実行されたコマンド名が格納されます。 ※MDK で要求したコマンドだけでなく、決済サーバーの内部処理コマンドも含まれます。	△
mstatus	処理結果コード	半角英数字 32 文字以内	処理の結果ステータスが格納されます。 “success”:正常終了 “failure”:異常終了 “pending”:保留	△
vResultCode	詳細結果コード	文字列 16 桁	処理の結果を詳細に表すコードとなります。 詳細は『結果コード一覧』を参照下さい。	△
txnDatetime	取引日時	文字列 23 桁	取引日時が格納されます。 YYYY-MM-DD hh:mi:ss.mmm 形式	△
amount	金額	半角数字 12 桁以内	決済した金額が格納されます。	△
properTransactionInfo	固有トランザクション情報	－	各決済サービスの固有トランザクション情報が格納されます。後述の一覧参照。	△

☆ 以下の情報は、上記「固有オーダー情報」「固有トランザクション情報」内の階層にセットされます。

※　以下は、LINE Pay 固有の検索結果フィールドです。

properOrderInfo	LINE Pay　固有オーダー情報			
与信同時売上フラグ	withCapture	文字列	決済申込時に指定した与信同時売上フラグ	△
商品名	itemName	文字列	決済申込時に指定した商品名	△
商品番号	itemId	文字列	決済申込時に指定した商品番号	△
申込金額	authorizeAmount	半角数字 7 桁以内	決済申込時に指定した金額	△
残高	balance	半角数字 7 桁以内	残高(現在の決済金額)	△
決済申込日時	authorizeDatetime	文字列 14 桁	決済申込日時 YYYYMMDDhhmmss 形式	△
LINE Pay 取引番号	linepayOrderId	文字列 19 桁以内	LINE Pay 側で発番された取引番号	△

properTransactionInfo	LINE Pay　固有トランザクション情報			
LINE Pay エラーコード	linepayErrorCode	文字列	LINE Pay 側システムから返却されたエラーコードが格納されます。	△
詳細コマンドタイプ	detailCommandType	文字列	※「detailCommandType」の詳細は「6-2 検索(Search)に関する補足(詳細コマンドタイプ/詳細オーダー決済状態)」を参照	△
LINE Pay への要求日時	linepayRequestDatetime	文字列 14 文字	LINE Pay への要求日時が格納されます。 YYYYMMDDhhmmss 形式	△
LINE Pay からの返戻日時	linepayResponseDatetime	文字列 14 文字	LINE Pay からの返戻日時が格納されます。 YYYYMMDDhhmmss 形式	△

5-3 結果通知電文

5.3.1 結果通知電文の概要

実装方法の詳細につきましては、弊社より提供しているサンプルプログラムをご参照ください。

■ **機能概要**

決済サーバーから店舗システムへ決済の結果通知を行います。

通知は、HTTP の POST を利用します。

※ プロトコルは HTTP(Port:80)、HTTPS(Port:443)に対応しています。

処理の概要については『開発ガイド』をご参照ください。

■ **HMAC について**

POST にてデータを通知するとき、リクエストヘッダーに HMAC を設定します。

また、HMAC 値は、リクエストボディから算出します。

アルゴリズムは「HmacSHA256」が設定されます。

フィールド名	設定値
content-hmac	h={アルゴリズム名};s={CCID};v={HMAC 値}

表 2 リクエストヘッダーに設定する HMAC フィールド

■ **結果通知受信処理結果について**

決済サーバーからの通知に対して店舗側システムが HTTP ステータスコード"200"を返戻した場合、受信処理が正常に終了したものと判断します。"200"以外のコードを返した場合は、受信失敗と判断し、一定期間、通知を繰り返します。

※ 規定の回数失敗した場合は、通知処理が停止しますのでご注意ください。

5.3.2 LINE Pay の結果通知電文

LINE Pay における結果通知電文の通知タイミングは、決済確認方式によって異なります。

【決済確認方式:ブラウザを介する通信 の場合】

LINE Pay 側で決済申込みが成立したタイミング(決済サーバーからの要求が成功したタイミング)となります。この時点では、店舗側の完了画面 URL への遷移はまだ行われていません。

消費者が画面遷移の途中で離脱した場合や、通信環境の問題で画面遷移時にタイムアウトが発生した場合には、最終遷移先の完了画面 URL まで到達しない場合がありますので、このような場合にも、決済サーバーからの通知を店舗側システムで受信して頂くことで、決済が成立したことを検知することができます。

【決済確認方式:サーバ間通信 の場合】

LINE Pay 側で決済申込の結果が確定したタイミングとなります。「ブラウザを介する通信」とは異なり、申込みが成功した場合も失敗した場合も、通知します。

No.	通知機能	通知タイミング	注意事項
1	決済申込完了通知 (Authorize) ※ブラウザを介する通信	決済申込が成立(成功)したタイミングで送信されます。	決済申込が失敗した場合は送信されません。

No.	通知機能	通知タイミング	注意事項
2	決済申込完了通知 (Authorize) ※サーバ間通信	決済申込が確定(成功又は失敗)したタイミングで送信されます。	

表 3 LINE Pay の結果通知機能

以下に、LINE Pay の結果通知項目を示します。

項番	フィールド名	項目名	書式・制限	説明
1	numberOfNotify	通知件数	半角数字 4 桁以内	1 度に通知可能な件数は 100 件、 101 件以上は次回通知
2	pushTime	送信時刻	半角数字 14 桁	決済サーバーから通知した時刻 yyyyMMddHHmmss 形式
3	pushId	識別 ID	半角数字 8 桁	プッシュ処理を行うたびに採番される ID。 注) 他の決済サービスで使用された ID と重複する場合があります。
通知件数分下記の項番(4～10)を繰り返す。尚、フィールド名の後ろに 4 ケタの連番(0000～0999)を付与する。				
4	orderId	取引 ID	半角英数字 100 桁以内	取引 ID
5	txnType	トランザクションタイプ	※右記の説明を参照	"Authorize":決済申込完了通知
6	txnTime	処理日時	半角数字 14 桁	yyyyMMddHHmmss 形式
7	vResultCode	詳細結果コード	半角英数字 4 桁	処理結果コードの詳細は、 『結果コード一覧』をご参照ください。
8	mstatus	処理ステータス	半角英数字 8 桁以内	"success":正常終了 "failure":異常終了(サーバ間通信の場合のみ) "pending":保留(サーバ間通信の場合のみ)
9	linepayOrderId	LINE Pay 取引番号	半角英数字 19 桁以内	LINE Pay システムが発番した取引番号
10	dummy	ダミー決済フラグ	半角数字 1 桁	ダミー取引を示す場合は"1"を設定する。

(注意事項)

- ✧ pushId(識別 ID)は、他の決済サービスで使用された Id と重複する場合がありますので、ユニークキーとして処理しないようにしてください。
- ✧ 項目の並び順は、必ずしも表の順序とは一致しません。

第6章 その他 補足事項

6-1 売上およびキャンセルに関する注意事項

6.1.1 売上およびキャンセルの要求可能期間

LINE Pay における、売上およびキャンセルの要求可能期間を下表に示します。

要求種類	要求可能期間
売上確定	決済成立(オーソリ)から 30 日間
キャンセル	売上前: 決済成立(オーソリ)から 30 日間 売上後: 売上日(または部分キャンセル実行日)から 30 日間

表 4 売上およびキャンセルの要求可能期間

- 売上確定は、オーソリが成立していても失敗する可能性があります。LINE Pay では必ず売上が成功した後に商品の発送やサービス提供を行うようにしてください。

6-2 検索(Search)に関する補足(詳細コマンドタイプ/詳細オーダー決済状態)

決済サーバーでは、システム内部データとして詳細コマンド(処理要求の種類)と詳細な決済状態(その取引がどのような状態にあるか)を管理しています。

通常の運用では、店舗様システムではこれらの内部情報を利用する必要はありませんが、店舗様システムにおいて取引の状態検索が必要なケースや、障害等のお問い合わせ時の参照情報として、これらの内部データを Search コマンドで公開しています。

下表に Search コマンドにて利用可能な内部データの一覧を示します。

詳細コマンドタイプ		コマンド成功時の詳細オーダー決済状態	
論理名	値	論理名	successDetailTxnType 値
決済認可	PreAuth	決済申込	Init
決済中止	QuitAuth	決済中止	Init
与信	Auth	与信	Auth
取消(与信)	VoidAuth	取消(与信)	VoidAuth
与信売上	AuthCapture	与信売上	AuthCapture
取消(与信売上)	VoidAuthCapture	取消(与信売上)	VoidAuthCapture
売上	PostAuth	売上確定	PostAuth
取消(売上)	VoidPostAuth	取消(売上)	VoidPostAuth
与信無効	ExpiredAuth	与信無効	ExpiredAuth

表 5 詳細コマンドタイプと詳細オーダー決済状態

注) 一覧に記載していないコマンドタイプと状態も存在します。

6.2.1 決済確認方式：サーバ間通信の場合の検索結果について

決済確認方式:サーバ間通信の場合は、結果通知か検索結果により決済の『成功／失敗』を判断します。

検索結果で決済の成功／失敗を判定するには、取引 ID で検索(Search)を実施し、検索結果の「successDetailTxnType」を確認します。

- successDetailTxnType が Auth または AuthCapture となっている場合は、決済成功です。
- successDetailTxnType が Init または QuitAuth の場合は、決済失敗です。

なお、決済後に売上確定や与信取消などの処理を実施している場合は、successDetailTxnType は PostAuth や VoidAuth になりますので、「表 5 詳細コマンドタイプと詳細オーダー決済状態」を参考に判定してください。

※決済申し込み後、消費者が LINE アプリの操作を完了させるまでに、最大で 20 分間時間をかけることができますので、検索は申し込み 20 分後に実施してください。

6-3 スマートフォン向けのサイトおよび独自アプリと LINE Pay アプリとの連携について

※本項は、独自アプリで決済をし、申込のリクエストで paymentConfirmType="0"（ブラウザを介する通信）を指定する加盟店様が対象となります。

6.3.1 appUrlScheme の指定方法

スマートフォン向けのサイトまたは、スマートフォン上で動作する独自アプリケーションから LINE Pay を利用する場合には、LINE Pay アプリで決済を実行後に、ブラウザまたはアプリに戻るための URL スキームの指定が必要です。申込電文（LinepayAuthorizeRequestDto）の appUrlScheme に指定して下さい。

appUrlScheme には、プレースホルダ"<url>"を埋めこんだ文字列を指定して下さい。決済サーバー側で、LINE Pay アプリで決済後の遷移先の URL（決済サーバーの URL）を設定して LINE Pay に連携します。URL の"https://"は省略されますのでご注意ください。

主要ブラウザの URL スキームと、独自アプリケーションの場合の指定方法の例を以下に示します。

起動アプリ（ブラウザ）	Android		iOS	
Safari	appUrlScheme	ー	appUrlScheme	デフォルトで Safari が起動するため指定不要
	packageName	ー		
Chrome	appUrlScheme	ー	appUrlScheme	googlechromes:<url>
	packageName	com.android.chrome		
Firefox	appUrlScheme	ー	appUrlScheme	ー
	packageName	org.mozilla.firefox		
Opera	appUrlScheme	ー	appUrlScheme	ー
	packageName	com.opera.browser		
Opera mini	appUrlScheme	ー	appUrlScheme	opera-https:<url>
	packageName	com.opera.mini.android		
独自アプリ	appUrlScheme	**独自のスキーム名**://"<url>"**を含む独自アプリの指定パラメータ** 例） myApp://<url>	appUrlScheme	**独自のスキーム名**:"<url>"**を含む独自アプリの指定パラメータ** 例）myApp:<url>
	packageName	**独自アプリのパッケージ名**		

表 6 appUrlScheme と packageName の指定方法

ブラウザを起動する場合は、利用者からのアクセス時の user-agent によって利用ブラウザを判別し、適切な URL スキームを設定してください。Android の場合には、packageName を設定してください。

独自アプリを起動する場合は、プレースホルダ<url>を取得し、アプリ内ブラウザを利用して<url>に遷移してください。遷移後は、決済サーバーにて決済の成否判定を行い、加盟店様が指定した URL に遷移します。また、独自アプリを起動する場合で、申込電文（LinepayAuthorizeRequestDto）の **useOrignalApp に"1"**を設定した場合は、LINE Pay アプリから起動された際のパラメータのプレースホルダ<url>には URL エンコーディングされた文字列が設定されますので、URL デコードしてご利用ください。

（独自アプリ起動時の LINE Pay 取引番号の連携について）

LINE Pay アプリが URL スキームを使ってアプリを呼び出す際には、appUrlScheme に LINE Pay 取引番号が付与されます。

&transactionId=取引番号 の形式で末尾に付与されます。

LINE Pay 取引番号は、申込の応答電文（LinepayAuthorizeResponseDto.linepayOrderId）にて連携していますので、加盟店の独自アプリ側でチェックのために利用することができます。実際の URL スキームの例を以下に示します。

（Android の例）

myApp://api.veritrans.co.jp/tercerog/webinterface/…..**&transactionId=2015029910000274310**

（iOS の例）

myApp:api.veritrans.co.jp/tercerog/webinterface/…..**&transactionId=2015029910000274910**

注）useOrignalApp = 1 で URL エンコーディング設定を行った場合、&transactionId=… よりも前の、<url>で指定した部分のみがエンコーディングされます。

6.3.2 redirectAppURL の利用方法

加盟店独自のスマートフォンアプリから決済申込を実行後、LINE アプリに移動して決済を行う方法について説明します。

尚、各 OS での実装はバージョン等で異なることがありますので下記は実装例として参照してください。

Android アプリの例

以下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 = "…"; // ここに redirectAppURL を設定してください。
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 アプリの例

以下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: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 = ...; // ここに redirectAppURL を設定してください。
        [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";
```

※iOS10 以降のアプリ連動時の注意事項

加盟店独自のスマートフォンアプリから LINE アプリに移動して決済申込を行う際 CustomURLScheme を呼び出して LINE アプリに移動しますが、iOS10 以降では CustomURLScheme を使ったアプリ呼び出しが制限されています。

その為、加盟店様が独自アプリを提供している場合、以下のガイドラインに従って実装する必要があります。

具体的な対応方法については、以下のいずれかを検討してください。

方法1. LSApplicationQueriesSchemes の管理リストに LINE を追加する

File : info.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
…
<dict>
  <key>LSApplicationQueriesSchemes</key>
  <array>
    <string>line</string>
  </array>
</dict>
…
</plist>
```

方法2. UIWebViewDelegate 実装(※方法1での対応が不可能な場合)

```
import UIKit

class ViewController: UIViewController {

    let webView = UIWebView.init()

    override func viewDidLoad() {
        super.viewDidLoad()
        webView.frame = self.view.frame

        // setup delegate
        webView.delegate = self

        self.view.addSubview(webView)

  // merchant develop code
    }

}

extension ViewController: UIWebViewDelegate {
    // http あるいは https ではなく外部スキームの場合、無条件に実行するように設定
    func webView(_ webView: UIWebView, shouldStartLoadWith request: URLRequest, navigationType:
UIWebViewNavigationType) -> Bool {
        if let url = request.url, url.scheme != "http" && url.scheme != "https" {
            UIApplication.shared.open(url, options: [:], completionHandler: nil)
            return false
        }
        return true
    }
}
```

6.3.3 checkUseBrowser の利用方法

LINE Pay システムで決済申込時のブラウザを識別できない場合、決済申込のブラウザとは異なるブラウザ（OS のデフォルトブラウザ）で決済結果の画面に遷移することがあります。

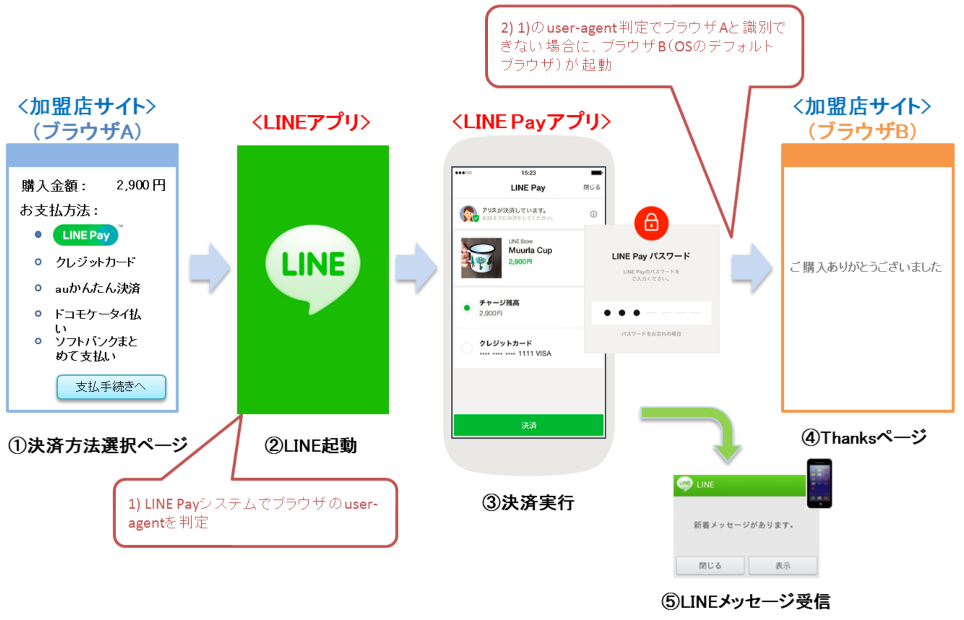
このような画面遷移をした場合に、加盟店様サイトで決済が成立しないなどの問題が生じる際に checkUseBrowser をご利用ください。ただし、消費者のスマートフォン操作（画面遷移）が、通常とは少し異なりますので実際の動作をご確認の上、ご利用をご検討下さい。

※この機能は、LinepayAuthorizeResponseDto の「redirectWebUrl」をご利用の場合のみ適用されます。

「redirectAppUrl」をご利用の際には適用されませんので、ご注意ください。

以下に画面遷移の例を示します。

checkUseBrowser に"false"を設定した場合の遷移（checkUseBrowser が未設定の場合も同様）



checkUseBrowser に"true"を設定した場合の遷移

6-4 LINE メッセージについて

消費者への完了メッセージ(LINE メッセージ)の対象は以下の機能です。

決済サービス名	決済申込 (与信/与信売上)	売上	キャンセル
LINE Pay	○	―	○

表 7LINE メッセージ送信対象一覧

第7章 導入・テストに関する補足

決済サーバーに接続し、テストを実施するための各種手続き・手順の詳細につきましては、『導入テストガイド』を参照してください。

7-1 決済申込時の画面遷移をシミュレートする

弊社では、LINE Pay のテストのための疑似環境を提供しています。疑似環境では、LINE アプリを利用した動作確認はできませんが、決済サーバーと加盟店システム間のインターフェイスは本番環境と同等のため、加盟店様システムの開発・テストにご活用頂けます。

アクセス環境（PC／スマートフォンのブラウザ、または加盟店独自アプリ）の違いを完全に再現することはできませんが、決済申込（Authorize）のレスポンスとして返戻された URL を呼び分けることで、ブラウザベースの遷移と加盟店独自アプリでの遷移をシミュレーションすることができます。

なお、Authorize コマンドで oneTimeKey を指定し且つ paymentConfirmType に”1”（サーバ間通信）を指定した場合には、redirectWebUrl と redirectAppUrl を返却しないため、以下の画面遷移ではなく 7-2 の方法でシミュレートできます。

7.1.1 PC またはスマートフォン（ブラウザ）での画面遷移

Authorize コマンドの結果として返戻されたリダイレクト URL（LinepayAuthorizeResponseDto.redirectWebUrl）に消費者のブラウザを遷移させると、以下の疑似環境が表示されます。

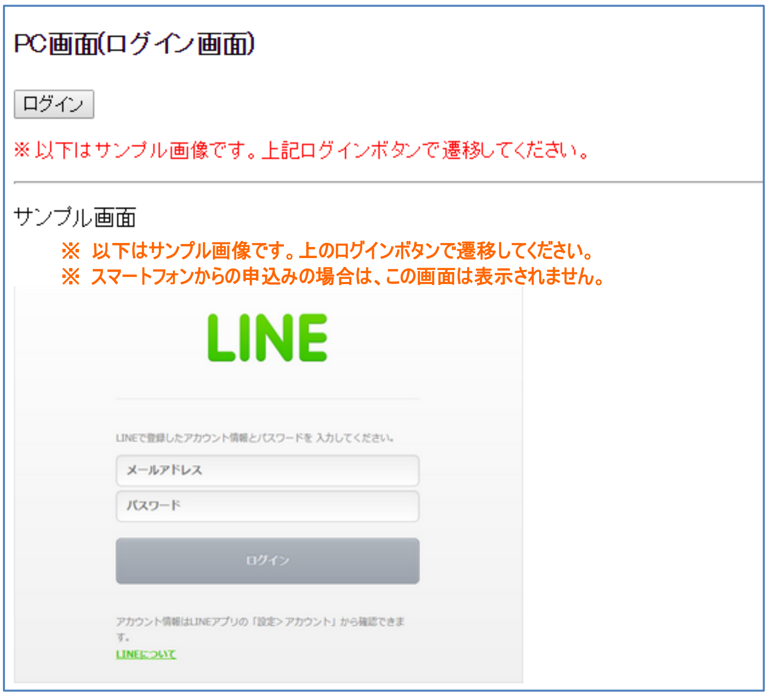


図 9 疑似環境（ダミーLINE Pay－PC ログイン画面）

LINE の認証画面のイメージが表示されますので、そのまま「ログイン」ボタンをクリックして次の画面に遷移します。

この画面は、スマートフォンの場合や oneTimeKey を指定した場合は表示されません。

図 10 疑似環境（ダミーLINE Pay－決済画面）

上記画面イメージの上半分には、PC（またはスマホ）のブラウザに表示される、待機画面を表示しています。

本番の決済では、この待機画面が表示された後に LINE Pay アプリ（スマートフォン）を使って決済を行います。

疑似環境では画面の下半分にスマホ画面のイメージを表示していますので、ここで「購入」または「キャンセル」を選択することで、以降の遷移を確認できます。

paymentConfirmType が”0”（ブラウザを介する通信）の場合、加盟店の決済完了画面（成功時の URL）またはキャンセル時の URL に遷移させることができます。

paymentConfirmType が”1”（サーバ間通信）の場合、決済申込完了通知か、検索（Search）にて取引の状態を確認することができます。

加盟店のエラー画面への遷移は、「7-3 決済申込・売上・取消コマンドのエラーをシミュレートする」を参照してください。

7.1.2 スマートフォン（加盟店アプリ）を利用した場合の画面遷移

Authorize コマンドの結果として返戻された URL（LinepayAuthorizeResponseDto.redirectAppUrl）に加盟店アプリから移動すると以下の疑似環境が表示されます。

実際の動作では LINE アプリが起動しますが、疑似環境ではブラウザ上に LINE アプリのイメージが表示されます。

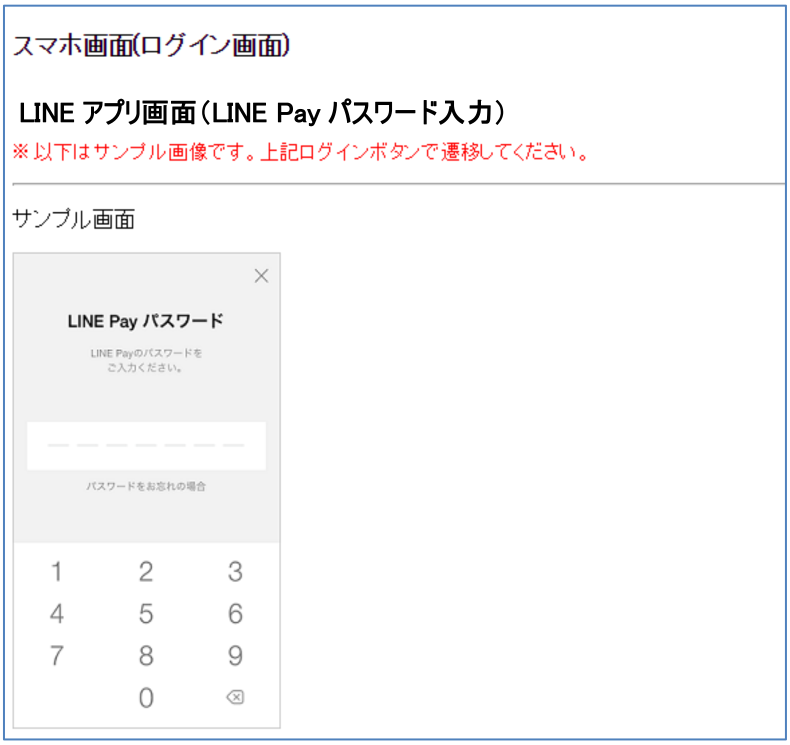


図 11 疑似環境（ダミーLINE Pay－LINE アプリ－パスワード入力）

図 12 疑似環境（ダミーLINE Pay－LINE アプリ－決済画面）

7-2 決済確認方法がサーバ間通信で oneTimeKey 利用時の決済申込をシミュレートする

Authorize コマンドで oneTimeKey を指定し且つ paymentConfirmType に"1"(サーバ間通信)を指定した場合、画面遷移が行われないため、以下の流れで決済申込をシミュレートすることができます。

1. Authorize コマンドを要求(oneTimeKey に任意の値を指定し、paymentConfirmType に"1"(サーバ間通信)を指定)
2. 約 30 秒後にステータスが変更
   - 2.1 結果通知 URL を指定している場合には、決済申込完了通知を送信しますので通知内容を確認してください。
   - 2.2 結果通知 URL を指定していない場合には、検索(Search)で確認してください。
3. 取引の状態を確認し、加盟店様の仕様に従って決済が完了した旨を消費者へお伝えください。

7-3 決済申込・売上・取消コマンドのエラーをシミュレートする

Authorize、Capture および Cancel コマンドの要求金額("amount")の下一桁の値を調整することで、意図的にエラーを発生させることができます。

要求金額の下一桁と、返戻される vResultCode の対応表を以下に記載します。

<table>
<tr><th>要求金額の下 1 桁</th><th>Authorize (申込み)</th><th>エラーURL への遷移</th><th>サーバ間通信時の結果通知</th><th>Capture (売上)</th><th>Capture (部分売上)</th><th>Cancel (取消)</th><th>Cancel (部分取消)</th></tr>
<tr><td>0</td><td rowspan="10">I001(成功)</td><td rowspan="3">I001(成功)</td><td rowspan="3">I001(成功)</td><td rowspan="6">I001(成功)</td><td rowspan="6">I001(成功)</td><td rowspan="8">I001(成功)</td><td rowspan="8">I001(成功)</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td>IG02(ｴﾗｰ)</td><td>IG02(ｴﾗｰ)</td></tr>
<tr><td>4</td><td rowspan="6">I001(成功)</td><td rowspan="6">I001(成功)</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td><td>IG03(ｴﾗｰ)</td><td>IG03(ｴﾗｰ)</td></tr>
<tr><td>7</td><td rowspan="3">I001(成功)</td><td rowspan="3">I001(成功)</td></tr>
<tr><td>8</td><td>IG03(ｴﾗｰ)</td><td>IG03(ｴﾗｰ)</td></tr>
<tr><td>9</td><td>I001(成功)</td><td>I001(成功)</td></tr>
</table>

- 金額未指定の Capture、Cancel の場合、全額が要求されたものとして動作します。
- 下一桁が 3 の取引については、Authorize コマンドの要求は成功しますが、その後の消費者画面の遷移で加盟店のエラーURL に遷移します。
- 上記の金額を利用したシミュレート以外に、oneTimeKey を利用して「oneTimeKey が無効」のエラーを発生させることができます。以下が、その条件となります。

  Authorize コマンドの oneTimeKey に「999999999999」を指定した場合に、IG08 のエラーコードを返却します。
- 返戻されるエラーコード(vResultCode)につきましては、今後、予告なしに変更させて頂く場合がございます。