

WbPayライブラリ仕様書

Ver. 1.0

PAYTREE(チャネルペイメントサービス株式会社様)対応版

株式会社 日本ウェルバ

2020年6月1日

目次

1. 概要
 - 1.1 目的
 - 1.2 対象OS
 - 1.3 提供形態
 - 1.4 関連文書
 - 1.5 QRコード決済システムフロー概要
 - 1.6 ライブラリの使い方概要
2. 設定
 - 2.1 ディレクトリ構成
 - 2.2 ネットワーク
 - 2.3 デジタル署名
 - 2.4 各種コード
 - 2.5 ログ
 - 2.6 コンフィギュレーションファイル
3. ライブラリリファレンス
 - 3.1 クラス
 - 3.2 Enum
 - 3.3 デリゲート
4. リクエスト詳細
 - 4.1 支払
 - 4.2 返金
 - 4.3 取消
 - 4.4 入金
5. テストモード
6. 取引ステータス
7. リクエストシーケンス
8. パラメーター一覧
9. エラーコード一覧

1. 概要

1.1 目的

本ライブラリは、チャネルペイメントサービス株式会社が提供する、QRコード決済のためのGatewayサービス(PAYTREE)との通信機能を提供します。

POSや券売機などのアプリケーションは、本ライブラリを通してPAYTREEとの通信を行い、決済プラットフォームのQRコード決済を利用できます。

1.2 対象OS

本ソフトウェアは以下のOSでの動作をサポートします。

Windows .Net Framework 4.5.2 以上

1.3 提供形態

本ライブラリは、DLLの形で提供されます。

本ライブラリを利用するアプリケーションから参照可能はフォルダーに配置してください。

DLL名 WbPay.dll

なお、本ライブラリのクラスはすべて以下の名前空間に属します。

名前空間： WbPay

1.4 関連文書

本仕様書に記載のない事項は、以下の文書に従います。

チャネルペイメントサービス株式会社発行仕様書

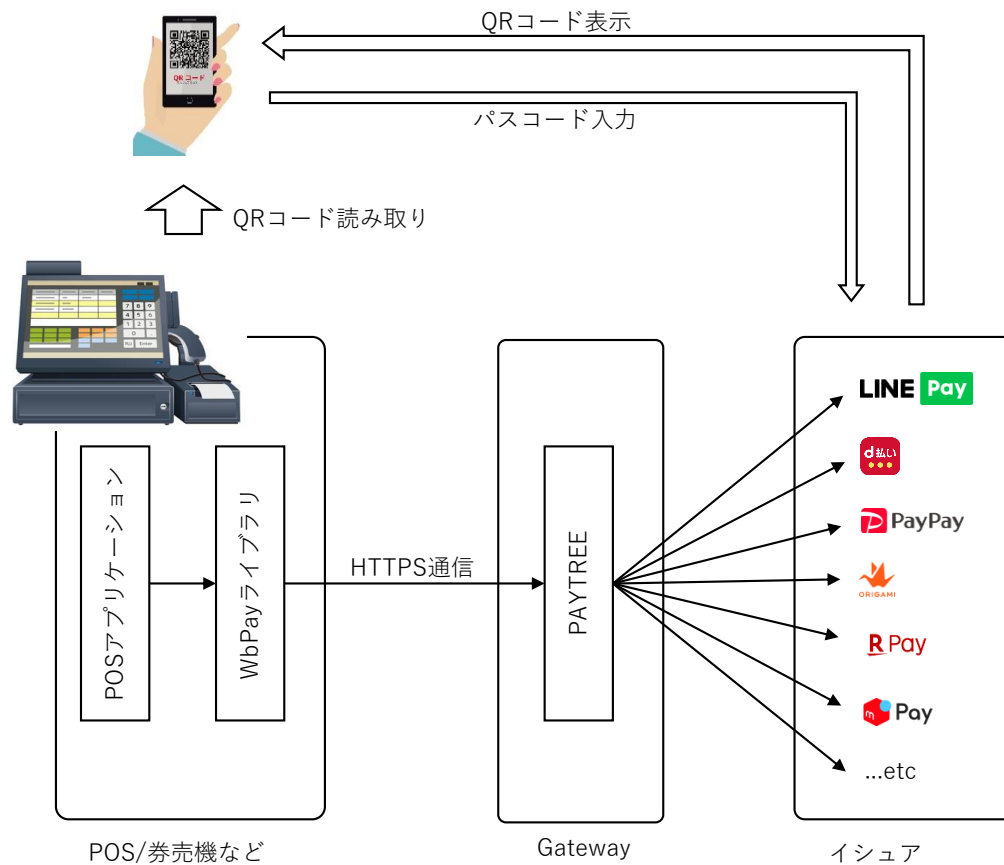
PAYTREEサービスAPI仕様書 API規約V3.9 2019年12月26日

POS-PAYTREE_IF仕様書 r27

CPS AWSインフラ全体構成

PAYTREEシーケンスr4.2(BtoB版)

1.5 QRコード決済システムフロー概要



本仕様書での記述について

チャネルペイメントサービス株式会社様が提供するGatewayを「Gateway」「チャネルペイメントサービス様」「CPS」「PAYTREE」「サーバ」と記述している場合があります。

イシューアを「決済プラットフォーム」「決済チャネル」「プロセッサ」と記述している場合があります。

1.6 ライブラリの使い方概要

(1) 処理の流れ

- ① WbCPSPayのインスタンスの生成
- ② リクエストメソッドの実行

WbCPSPayには以下のリクエストメソッドがあります。

| | |
|---------|----|
| Pay | 支払 |
| Refund | 返金 |
| Reverse | 取消 |
| Deposit | 入金 |

リクエストの結果はイベントで通知されます。イベントには以下の2種類があります。

| | |
|---------------|-------------|
| CompleteEvent | リクエスト完了イベント |
| ErrorEvent | エラーイベント |

必要に応じてリクエストを制御するメソッドも使用します。

| | |
|------------|-------------------|
| PayConfirm | 支払時の顧客承認が必要な場合の制御 |
|------------|-------------------|

- ③ インスタンスをクローズする。

アプリケーションの終了時は必ずクローズしてください。

(2) リクエストメソッドの処理について

リクエストメソッドでは、Gatewayにリクエストを送信し、レスポンスを受信するだけでなく、エラー時の回復処理や、エラー回復出来なかった場合に必要な後処理など、一連のリクエストシーケンスを実行しています。

そのため、イベントで通知される結果は、リクエストの最終結果となります。

アプリケーションでは、エラー時のサーバに対する後処理などを考慮する必要はありません。また、リトライなどにより、プログラム上でエラーを回復することもできません。

回復が必要なエラーが発生した場合は、エラーイベントで返される推奨エスカレーションに従った対応(コールセンタへの連絡など)が必要になります。

(3) 支払時に顧客承認が必要な場合の処理について

支払時に顧客承認が必要な場合、支払リクエストは一旦リクエスト完了イベントで「顧客支払待ち」を通知後、顧客のパスコード入力を待機します。

レジ担当者がお客様のパスコード入力を確認後、PayConfirmで支払ステータス確認を指示します。

支払確認のポーリングモードを指定することができます。ポーリングモードを指定することにより、レジ担当者によるお客様のパスコード入力確認が不要になります。

ポーリングモードでは、パスコード入力待機中に一定時間間隔でサーバに対してポーリングを行い、支払完了が確認された時点でコンプリーイベントで通知または、タイムアウト時にエラーイベントで通知を行います。

但し、ポーリングによる処理をする場合は、あらかじめ時間間隔や回数についてGateway及び決済プラットフォームとの協議が必要ですので、ご注意ください。

2. 設定

2.1 ディレクトリ構成

本ライブラリを実行するために、以下のディレクトリが必要になります。
これらのディレクトリは変更できません。

| | |
|-------------|-------------------|
| Wellbaホーム | C:\Wellba |
| アプリケーションホーム | (Wellbaホーム)\WbPay |
| ログディレクトリ | (アプリケーションホーム)\Log |

これらのディレクトリには、コンフィギュレーションファイルなどを置くほか、ライブラリが動作に必要なファイルを作成しますので、読み込み、書き込み、ファイル作成のアクセス権が必要です。
また、ライブラリが作成したファイルを、変更したり削除すると、正常に動作しなくなりますのでご注意ください。

2.2 ネットワーク

本ライブラリはInternet経由してPAYTREEに接続しますので、Internetへの接続、通信が可能なネットワーク環境が必要です。

また、接続先はFQDN(Fully Qualified Domain Name)で指定しますので、DNSによる名前解決ができる必要があります。

2.3 デジタル署名

PAYTREEとの通信データにはデジタル署名を添付します。そのための、デジタル署名作成のための鍵をあらかじめ用意する必要があります。

デジタル署名の方式はSHA1WithRSAです。

鍵はPKCS1形式(2048)で、PEMファイルで作成します。

公開鍵はPAYTREE側で受信データ検証のために必要となりますので、あらかじめキャナルペイメントサービス株式会社様に送付しPAYTREEに登録してもらう必要があります。

また、本ライブラリで、受信データ検証のために必要なPAYTREEの公開鍵もあらかじめ入手しておく必要があります。

鍵ファイルは、アプリケーションホームに置いてください。

2.4 各種コード

以下のコードは、PAYTREEとの通信に必要なものですので、あらかじめ準備しておく必要があります。

マーチャントID (固定長12桁)

キャナルペイメントサービス株式会社様より加盟店毎に発行されるIDです。

端末識別番号 (英数字固定長13桁)

加盟店で任意に設定するコードです。

企業コード(5桁) + "0"(1桁) + 店舗コード(6桁) + 端末番号(1桁)

加盟店、店舗、端末を一意に識別できるように決める必要があります。

店舗番号 (英数字32桁以下)

加盟店で任意に設定するコードです。

端末識別番号中の店舗コードと異なるコードでも問題ありません。

ただし、このコードはあらかじめPAYTREEに登録する必要があります。

端末番号 (英数字32桁以下)

加盟店で任意に設定するコードです。

端末識別番号中の端末番号と異なるコードでも問題ありません。

2.5 ログ

ライブラリーの実行状態や実行時のエラーをログファイルに記録します。

ログファイルは、実行日毎の作成されます。

ログの有効期間を指定することにより、古くなったログファイルは自動的に削除されます。

記録内容は、ログレベルを指定することにより変えることができます。

ログの記録順が、実際の事象の発生順と異なることがあります。その場合は、記録されている発生時刻で順序を判断してください。

ディレクトリ

(アプリケーションホーム)¥Log

ファイル名

WbCPSPay.(年月日).log

ログレベル

ログレベルは、コンフィギュレーションファイルに指定します。

ログレベルは、レベルの値または名称どちらでも指定できます。

指定したレベルより上位のレベル(レベル値が小さい)の内容が出力されます。

通常の運用では、ログレベルの指定は Sequence を推奨します。

| レベル(値、名称) | ログ内容 |
|---------------|---|
| 0 Basic | 実行開始、終了など実行の基本情報を記録します。 |
| 1 Error | エラーを記録します。エラーにはプログラム上のエラーと、CPSサーバが返すエラーレスポンスが含まれます。 |
| 2 Warning | 実行に影響のないデータ不備などを記録します。 |
| 3 Sequence | リクエストメソッドの実行毎に、リクエスト内容とレスポンス内容を記録します。 |
| 4 Transaction | CPSサーバとの通信トランザクションをすべて記録します。 |

2.6 コンフィギュレーションファイル

リクエスト送信に必要なパラメータや、ライブラリの動作の設定をします。

設定は Key = Value 形式で記述します。

コメントを記述できます。"#"から行末までをコメントとして扱います。

文字コードはUTF-8を使用し、Key、Valueには"#"を除く英数字記号が使用可能です。

設定には必須と任意があり、必須設定がないとエラーになります。

ファイル

(アプリケーションホーム)¥WbCPSPay.conf

Key

PAYTREE接続URL(必須)

ServerURL

PASYTREEサーバURL

各APIのPath(必須)

ApiPathPay

支払API

ApiPathPayV1

支払(自動判別)API

ApiPathRefund

返金API

ApiPathReverse

取消API

ApiPathConfirm

確認API

ApiPathConfirmV1

確認(自動判別)API

Http設定(任意)

HttpTimeout

HTTPリクエストのタイムアウト時間(単位：秒)

このタイムアウト時間には名前解決に必要な時間が含まれています。

DNSの応答に最大15secかかることがあります。

15秒以下に設定すると、Httpリクエストが必ずタイムアウトになる場合がありますので、15秒以下にはしないで下さい。

(規定値：100秒)

IgnoreCertificateError

SLL証明書エラーを無視する[yes , no]

サーバのSSL証明書が信頼できない場合の動作を設定します。

(規定値：no)

通信データのデジタル署名用の鍵ファイル(必須)

PrivateKeyFile

秘密鍵ファイル

リクエストデータの署名作成用の自サイトの秘密鍵

PublicKeyFile

公開鍵ファイル

レスポンスデータの検証用のサーバの公開鍵

リクエストパラメータ(必須)

| | |
|--------------------|--|
| MerchantId | マーチャントID(固定長12桁) |
| TerminalUniquId | 端末識別番号(固定長13桁) |
| BranchCode | 店舗番号(32桁以下) |
| TerminalCode | 端末番号(32桁以下) |
| Locale | 国際化コード(30桁以下) (固定: JAPAN) ※固定コード以外の設定はしないで下さい。 |
| TimeZone | タイムゾーン(4桁以下) (固定: p9) ※固定コード以外の設定はしないで下さい。 |
| CurrencyCode = JPY | 正札通貨(10桁以下) (固定: JPY) ※固定コード以外の設定はしないで下さい。 |
| AppVersion | サービス接続元のバージョン情報 (固定: windows-WbCPSPay-1.0) ※固定コード以外の設定はしないで下さい。 |

支払顧客承認待ち動作設定 (任意)

| | |
|---------------------------|--|
| PayConfirmPollingMode | 顧客承認待ちポーリングモード[yes、no] 顧客承認待ち時のポーリング動作の有無を設定します。 (規定値: no) |
| PayConfirmPollingWait | 顧客承認待ち確認待ち時間(単位: 秒) 顧客承認待ち時の、支払確認までの待ち時間を指定します。 (規定値: 5秒) |
| PayConfirmPollingInterval | 顧客承認待ちポーリング時間間隔(単位: 秒) 顧客承認待ち時の、支払確認実行の時間間隔を指定します。 (規定値: 3秒) |
| PayConfirmPollingTimeout | 顧客承認待ちタイムアウト時間(単位: 秒) 顧客承認待ち時の、タイムアウト時間を指定します。 (規定値: 60秒) |

通信エラー時の確認リクエスト繰り返し設定(任意)

| | |
|------------------------|---|
| RequestConfirmInterval | 確認リクエストの送信間隔(単位: 秒) リクエスト結果が通信エラーなどで不明の場合の、確認リクエストの送信までの待機時間。 (規定値: 1秒) |
| RequestConfirmNum | 確認リクエストの送信回数 通信エラーなどの場合の確認リクエスト再送信の回数。 通常1を設定します。決済プラットフォームにより回数制限がある場合がありますので、3以上を設定する場合は、必ずPAYTREE様、決済プロットフォーム様と相談の上設定してください。 決済プラットフォームによって設定を変えることはできません。 (規定値: 1回) |

ログ出力設定(任意)

| | |
|-----------|--|
| LogLevel | ログレベル(値、名称) (規定値: Error) |
| LogExpire | ログの有効期間(単位: 日) ログファイルを自動削除するまでの期間を設定します。 0は削除しない (規定値: 0) |

テストモード(任意)

AppTestMode テストモードを有効にする。[yes , no]

テストモードを設定します。

(規定値：no)

TestSequenceDelay 遅延時間(単位：秒)

サーバーとの通信をシミュレートするために、レスポンスを受信するまでの遅延時間を挿入することができます。

0を設定すると、遅延時間は挿入されません。

(規定値：1秒)

TestClassPay 支払テストクラス

テストで使用する支払レスポンスの種類を設定します。

(規定値：0)

TestClassRefund 返金テストクラス

テストで使用する返金レスポンスの種類を設定します。

(規定値：0)

TestClassReverse 取消テストクラス

テストで使用する取消レスポンスの種類を設定します。

(規定値：0)

TestClassDeposit 入金テストクラス

テストで使用する入金レスポンスの種類を設定します。

(規定値：0)

3. ライブラリーリファレンス

3.1 クラス

WbCPSPay クラス

PAYTREEのサービスを利用するためのクラスです。

アプリケーションからの要求を、リクエストの形でPAYTREEへ送信し、決済プラットフォームの決済結果をPAYTREEからレスポンスとして受け取ります。

各ホストマシン(PCなど)では、同時に複数のWbCPSPayプロセスを動作させることはできません。

継承

Object

コンストラクター

public WbCPSPay()

ひとつのホストマシンの中では、同時にひとつのインスタンスしか生成することができません。

WbCPSPayインスタンスの生成は比較的重い処理ですので、インスタンスの生成はアプリケーションプログラム起動時に行い、アプリケーションプログラム終了まで、そのひとつのインスタンスを利用することを推奨します。

ただし、複数のアプリケーションプログラムから、同時にWbCPSPayを利用する可能性がある場合は、アプリケーション間で競合が発生しないようしないよう、インスタンスの生成、Closeのタイミングを十分検討してください。

例外

WbPayException

WbCPSPayがすでに動作中です。

コンフィギュレーションファイルの記述にエラーがあります。

.Net framework の標準例外

ログファイルがオープンできない。

コンフィギュレーションファイルがオープンできない。

その他のエラー

メソッド

public void Close()

WbCPSPayインスタンスを終了し、リソースを解放します。

一度Closeしたインスタンスを再度Openすることはできません。

Close後は、他のWbCPSPayインスタンスを生成することができるようになります。

public void Pay(WbCPSPayRequest request)

支払リクエストをPAYTREEに送信します。

処理は非同期で実行されるため、このメソッドは、すぐに戻ります。

リクエストメソッドの同時実行はできません。

また、リクエストシーケンス実行中も実行できません。

処理の内容については、リクエスト詳細を参照してください。

例外

ArgumentNullException

requestがnullです。

WbPayException

リクエストメソッドが実行中

実行中のリクエストシーケンスが完了していない。

必須リクエストパラメータが未指定、桁数オーバーなど

public void PayConfirm(bool confirm_flag)

支払(顧客承認有)の場合の顧客支払待ちの制御をします。

リクエストメソッドのとの同時実行はできません。

支払リクエストシーケンス実行中で、顧客支払待ちの場合のみ実行できます。

パラメータ

bool confirm_flag

true を指定した場合、確認を実行します。確認結果はEventで通知されます。

falseを指定した場合、顧客承認待ちを終了します。顧客承認は完了していない場合

は、エラーイベントで支払失敗が通知されます。顧客承認待ちポーリングモードの場合

もポーリングを中止します。

例外

WbPayException

リクエストメソッドが実行中

public void Refund(WbCPSRefundRequest request)

返金リクエストをPAYTREEに送信します。

処理は非同期で実行されるため、このメソッドは、すぐに戻ります。

リクエストメソッドの同時実行はできません。

また、リクエストシーケンス実行中も実行できません。

処理の内容については、リクエスト詳細を参照してください。

例外

ArgumentNullException

requestがnullです。

WbPayException

リクエストメソッドが実行中

実行中のリクエストシーケンスが完了していない。

必須リクエストパラメータが未指定、桁数オーバーなど

public void Reverse(WbCPSReverseRequest request)

取消リクエストをPAYTREEに送信します。

処理は非同期で実行されるため、このメソッドは、すぐに戻ります。

リクエストメソッドの同時実行はできません。

また、リクエストシーケンス実行中も実行できません。

処理の内容については、リクエスト詳細を参照してください。

例外

ArgumentNullException

requestがnullです。

WbPayException

リクエストメソッドが実行中

実行中のリクエストシーケンスが完了していない。

必須リクエストパラメータが未指定、桁数オーバーなど

public void Deposit(WbCPSDepositRequest request)

入金リクエストをPAYTREEに送信します。

処理は非同期で実行されるため、このメソッドは、すぐに戻ります。

リクエストメソッドの同時実行はできません。

また、リクエストシーケンス実行中も実行できません。

処理の内容については、リクエスト詳細を参照してください。

例外

ArgumentNullException

requestがnullです。

WbPayException

リクエストメソッドが実行中

実行中のリクエストシーケンスが完了していない。

必須リクエストパラメータが未指定、桁数オーバーなど

イベント

PAYTREEからのレスポンスは、イベントでアプリケーションに通知されます。

イベントハンドラー内より、リクエストメソッドを実行することは可能です。

ただし、リクエストメソッド呼び出し時に、すでに実行中のリクエストメソッドが未完了であったり、リクエストシーケンスが未完了の場合は、例外が発生します。

ひとつのリクエストに対して、複数のイベントが発生する場合がありますが、通常は同時に発生することはありません。

複数のイベントが発生する場合は、イベントハンドラーを抜けた後、次のイベントが発生します。

ただし、イベントハンドラーの処理がある一定時間以上継続している場合、イベントハンドラーを抜ける前に、次のイベントが発生する場合があります。

イベントハンドラーの処理時間が、イベントの発生間隔と比較して長い場合、イベント通知順が前後する場合があります。

public event WbCPSCompleteEventHandler CompleteEvent

PAYTREEへのリクエストに対して、決済成功のレスポンスを受け取ったことを通知します。

詳細は、4.リクエスト詳細を参照してください。

public event WbCPSErrorEventHandler ErrorEvent

PAYTREEへのリクエストに対して、決済失敗またはエラーのレスポンスを受け取ったことを通知します。

詳細は、4.リクエスト詳細を参照してください。

WbCPSPayRequest クラス

支払リクエストで送信するデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSPayResponse クラス

支払リクエストに対して受信したレスポンスデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSRefundRequest クラス

支払リクエストで送信するデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSRefundResponse クラス

支払リクエストに対して受信したレスポンスデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSReverseRequest クラス

返金リクエストで送信するデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSReverseResponse クラス

取消リクエストに対して受信したレスポンスデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSDepositRequest クラス

入金リクエストで送信するデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSDepositResponse クラス

入金リクエストに対して受信したレスポンスデータを表します。

継承

Object

フィールド

フィールドについては、リクエスト詳細を参照してください。

WbCPSResponseError クラス

リクエストがエラーとなった場合の、エラー情報を表します。

継承

Object

フィールド

public int Type

エラーの種類を表します。

- | | |
|---|--------------------|
| 2 | PAYTREEネット通信エラー |
| 5 | WbCPSPayライブラリーエラー |
| 6 | WbCPSPayポストプロセスエラー |
| 9 | PAYTREEエラー |

public string Code

エラーコード

詳細はエラーコード一覧を参照してください。

public string SubCode

エラーサブコード

PAYTREEプロセッサエラーのエラーコードです。

詳細はエラーコード一覧を参照してください。

public string Message

エラーメッセージ

詳細はエラーコード一覧を参照してください。

public WbCPSPAction RequestAction

エラーに対する、推奨エスカレーションを表します。

詳細はエラーコード一覧を参照してください。

WbPayException クラス

継承

Exception

プロパティー

public int Type

エラーの種類を取得します。

- | | |
|---|--------------------|
| 2 | PAYTREEネット通信エラー |
| 5 | WbCPSPayライブラリーエラー |
| 6 | WbCPSPayポストプロセスエラー |
| 9 | PAYTREEエラー |

public string Code

エラーコードを取得します。

詳細はエラーコード一覧を参照してください。

public string SubCode

エラーサブコードを取得します。

PAYTREEプロセッサエラーのエラーコードです。

詳細はエラーコード一覧を参照してください。

public string Message

例外を説明するメッセージを取得します。

(継承元：Exception)

3.2 Enum

public enum WbCPSPayType

支払チャンネル(決済プラットフォーム)を表します。

フィールド

| | |
|----------------------|-----------------------|
| AutoDetect | 自動判別 |
| LinePay | LINE Pay |
| RegionalElectricMony | 地域電子マネー |
| OrigamiPay | Origami Pay |
| RakutenPay | 楽天ペイ |
| PayPay | PayPay |
| dPayment | d払い |
| CpayAlipay | Cpay(Alipay) |
| CpayWechat | Cpay(Wechat) |
| MerPay | メルペイ |
| AuPay | au Pay |
| JCBAlipay | JCP-Alipay (※リクエスト不可) |
| QUOCardPay | QUO カード Pay |
| YuchoPay | ゆうちょ Pay |
| KPlus | K Plus |
| ResonaWallet | りそなウォレット |
| EPOSPay | EPOS Pay |
| JCoinPay | J-Coin Pay |
| pring | pring |
| atone | atone |
| aftee | aftee |
| BankPay | BankPay |
| OnepayAlipay | Onepay(Alipay) |
| OnepayWechat | Onepay(Wechat) |
| OnepayGNAlipay | Onepay(GN-Alipay) |
| OnepayGNAlipayHK | Onepay(GN-AlipayHK) |
| OnepayGNkakaopay | Onepay(GN-kakaopay) |
| SmartCode | SmartCode |

public enum WbCPSCurrencyType

正札通貨を表します。

フィールド

| | |
|-----|-------------|
| CNY | 人民元 |
| GBP | ポンド |
| HKD | 香港ドル |
| USD | 米ドル |
| JPY | 日本円 |
| CAD | カナダドル |
| AUD | オーストラリア ドル |
| EUR | ユーロ |
| NZD | ニュージーランド ドル |
| KRW | 韓国ウォン |

public enum WbCPSValueType

入金種類を表します。

フィールド

| | |
|--------|------|
| Basic | 基本 |
| Bonus | ボーナス |
| Coupon | クーポン |
| Point | ポイント |

public enum WbCPSAction

エスカレーション種類をあわわします。

フィールド

| | |
|------------|--------------------|
| VendorCall | アプリケーションベンダー |
| CPSCall | CPSコールセンター |
| Customer | 顧客から決済アプリカスタマーサービス |

public enum WbCPSRequestType

リクエスト種類を表します。

フィールド

| | |
|---------|---------|
| Pay | 支払リクエスト |
| Refund | 返金リクエスト |
| Reverse | 取消リクエスト |
| Confirm | 確認リクエスト |
| Deposit | 入金リクエスト |

3.3 デリゲート

public delegate void WbCPSCompleteEventHandler(WbCPSRequestType type, Object response)

CompleteEventを処理するメソッドです。

パラメータ

WbCPSRequestType type

レスポンスの元となるリクエストの種類を表す。

Object response

レスポンスデータ

responseのクラスは、リクエストの種類により変わります。

詳細は、4.リクエスト詳細を参照してください。

public delegate void WbCPSErrorEventHandler(WbCPSRequestType type, WbCPSResponseError error)

ErrorEventを処理するメソッドです。

パラメータ

WbCPSRequestType type

レスポンスの元となるリクエストの種類を表す。

WbCPSResponseError error

エラーの内容を表します。

4. リクエスト詳細

4.1 支払

顧客が提示したバーコード／QRコードで、指定金額の支払を要求します。

支払失敗の原因が、リクエスト内容ではなく、通信エラーなどの場合は、再度、支払を実行して下さい。

支払失敗時の取消処理が失敗した場合は、CPSコールセンター対応が必要です。

顧客承認が必要な支払の場合は顧客のパスコード入力を待機し、PayConfirmの指示により支払ステータスの確認を行います。

顧客承認待ちの終了は、パスコードが入力され支払が完了したとき、顧客がパスコード入力を中止したとき、タイムアウト、その他のエラーの場合です。

(1) リクエストシーケンス

- ① リクエストの送信
- ② レスポンスの受信
- ③ 通信エラーやその他必要な場合の結果再確認
- ④ エラーによる支払失敗の場合の取消処理

(2) イベント

CompleteEvent イベント

支払成功、顧客支払待ちの場合にCompleteEventが発生します。

支払成功、顧客支払待ちの区別は、レスポンスパラメータのTransStatusの値で判断します。

顧客承認が必要な支払の場合は、顧客支払待ち開始時とパスコード入力後の支払完了時の2回CompleteEventが発生します。

ErrorEvent イベント

PAYTREEがリクエスト成功のレスポンスを返した場合でも、支払失敗に相当する場合は、ErrorEventが発生します。

1回のPay呼び出しに対して、複数のErrorEventが発生する場合があります。

顧客承認待ち時にPayConfirm実行により支払ステータスの確認を行った場合、パスコード未入力で顧客支払待ちが継続している場合は、顧客支払待ちをErrorEventで通知します。

(3) リクエストデータ(WbCPSPayRequest クラス)

public WbCPSPayType PayType (必須)

支払チャンネル

自動判別を指定した場合は、Gatewayにて自動判別されます。

public decimal Amount (15桁以下) (必須)

正札金額。小数点以下は切り捨てられます。

public string ReceiptNo (32桁以下) (任意)

レシート番号

public string UserCode (512桁以下) (必須)

ユーザーのバーコードまたはQRコード

(QRコードはバーコードと同じコード体系、またはEMV体型の全コード)

public string Remark (128桁以下) (任意)

備考

(4) レスポンスデータ(WbCPSPayResult クラス)

```
public string OrderId                (32桁)
    オーダーID(PAYTREE内で採番されるトランザクション番号)
public string TransTime              (14桁)
    取引日時(YYYYMMDDHHMMSS)
public decimal Amount                (15桁以下)
    支払金額。小数点以下は発生しない。
public int TransStatus
    取引ステータス
public WbCPSPayType PayType
    支払チャンネル
```

4.2 返金

支払済みのトランザクションに対して、全額または一部の返金を要求します。

返金可能な実効期間は決済プラットフォームにより設定されます。

返金の際に手数料を含むか否かは決済プラットフォームとの契約によります。

返金失敗の原因が、リクエスト内容ではなく、通信エラーなどの場合は、再度、返金を実行して下さい。

返金失敗の原因が通信エラー以外の場合は、管理画面(決済プラットフォームまたはCPSが用意している)にて返金処理をしてください。

(1) リクエストシーケンス

- ① リクエストの送信
- ② レスポンスの受信
- ③ 通信エラーやその他必要な場合の結果再確認

(2) イベント

CompleteEvent イベント

返金成功の場合にCompleteEventが発生します。

ErrorEvent イベント

PAYTREEがリクエスト成功のレスポンスを返した場合でも、返金失敗に相当する場合は、ErrorEventが発生します。

(3) リクエストデータ(WbCPSRefundRequest クラス)

| | | |
|-----------------------------|----------|------|
| public decimal RefundAmount | (15桁以下) | (必須) |
| 返金金額 | | |
| 小数点以下は切り捨てられます。 | | |
| public string OrderId | (32桁以下) | (必須) |
| 元オーダーのオーダーID | | |
| public string RefundReason | (256桁以下) | (任意) |
| 返金理由 | | |
| public string Remark | (256桁以下) | (任意) |
| 備考 | | |

(4) レスポンスデータ(WbCPSRefundResult クラス)

| | |
|-----------------------------|---------|
| public string OrderId | (32桁以下) |
| 元オーダーのオーダーID | |
| public int TransStatus | |
| 取引ステータス | |
| public decimal RefundAmount | (15桁以下) |
| 返金金額 | |
| 小数点以下は発生しない | |
| public string TransTime | (14桁) |
| 取引日時(YYYYMMDDHHMMSS) | |

4.3 取消

支払中や支払済みのトランザクションに対して、取消を要求します。

店舗システムが、支払処理中に発生したエラーなどにより支払処理を中止する場合に、取消を要求します。
(POS中止処理)

支払完了後の返品などにより、払い戻しが発生する場合には取消を要求することはできません。その場合は、返金を使用してください。

取消失敗の原因が、リクエスト内容ではなく、通信エラーなどの場合は、再度、取消を実行して下さい。

取消失敗の原因が通信エラー以外の場合は、CPSコールセンター対応が必要になります。

(1) リクエストシーケンス

- ① リクエストの送信
- ② レスポンスの受信
- ③ 通信エラーやその他必要な場合の結果再確認

(2) イベント

CompleteEvent イベント

取消成功の場合にCompleteEventが発生します。

ErrorEvent イベント

PAYTREEがリクエスト成功のレスポンスを返した場合でも、取消失敗に相当する場合は、ErrorEventが発生します。

(3) リクエストデータ(WbCPSReverseRequest クラス)

```
public string OrderId                (32桁以下)    (必須)
    元オーダーのオーダーID
```

(4) レスポンスデータ(WbCPSReverseResult クラス)

```
public string OrderId                (32桁以下)
    元オーダーのオーダーID
```

```
public int TransStatus
```

取引ステータス

```
public decimal Amount                (15桁以下)
```

取消金額

小数点以下は発生しない。

取消成功の場合でも、Amountが0となっている場合があります。

```
public string TransTime              (14桁)
```

取引日時(YYYYMMDDHHMMSS)

4.4 入金

決済プラットフォームへチャージの要求をします。

チャージサービスの実施には、加盟店様と決済プラットフォームとの契約が必要となります。

(現在、チャージサービスの対応は LINEPay のみ)

入金失敗の原因が、リクエスト内容ではなく、通信エラーなどの場合は、再度、入金を実行して下さい。

入金失敗時の取消処理が失敗した場合は、CPSコールセンター対応が必要です。

(1) リクエストシーケンス

- ① リクエストの送信
- ② レスポンスの受信
- ③ 通信エラーやその他必要な場合の結果再確認
- ④ エラーによるリクエスト失敗の場合の取消処理

(2) イベント

CompleteEvent イベント

入金成功の場合にCompleteEventが発生します。

ErrorEvent イベント

PAYTREEがリクエスト成功のレスポンスを返した場合でも、入金失敗に相当する場合は、ErrorEventが発生します。

1回のDeposit呼び出しに対して、複数のErrorEventが発生する場合があります。

エラーの詳細は、エラー一覧参照。

(3) リクエストデータ(WbCPSDepositRequest クラス)

| | |
|---------------------------------|--------------|
| public WbCSPayType PayType | (必須) |
| 支払チャンネル | |
| 自動判別は指定できません。 | |
| public WbCPSValueType ValueType | (必須) |
| 入金種類。現時点では「基本」のみをサポートしている。 | |
| public decimal Amount | (15桁以下) (必須) |
| 入金金額 | |
| 小数点以下は切り捨てられます。 | |
| public string ReceiptNo | (32桁以下) (任意) |
| レシート番号 | |
| public string UserCode | (32桁以下) (必須) |
| 取引対象のカード番号 | |

(4) レスポンスデータ(WbCPSDepositResult クラス)

| | |
|----------------------------------|---------|
| public string OrderId | (32桁) |
| オーダーID(PAYTREE内で採番されるトランザクション番号) | |
| public string TransTime | (14桁) |
| 取引日時(YYYYMMDDHHMMSS) | |
| public decimal Amount | (15桁以下) |
| 入金金額 | |
| public int TransStatus | |
| 取引ステータス | |

5 テストモード

テストモードでは、実際にCPSサーバとの通信は行わずに、サーバからのレスポンスをシミュレートします。

テストモードに設定することにより、サーバとの通信を行わずに、アプリケーションのテストができます。レスポンスの種類は、テストクラスを設定することにより、処理ごとに変更することができます。

テストモードでは、単にサーバのレスポンスをシミュレートしているだけで、リクエストデータの内容に関するチェックなどは行っていない。

また、エラーについては、テストのために典型的なパターンを用意しているだけです。すべてのエラーを網羅しているわけではありません。

テストクラス

支払

支払の履歴は、WbCPSPayのインスタンスが存在する間は、保存されます。

返金、取消、確認は支払履歴が保存されているオーダーについてのみ可能です。

- 0 成功
- 1 顧客承認→成功
- 2 ネットワークエラー
- 3 ユーザコードエラー
- 4 残高不足
- 5 サービス停止中
- 6 顧客承認→支払中止
- 7 顧客承認→支払タイムアウト

支払確認

テストクラスの設定はありません。

支払が「顧客支払待ち」となったオーダーについてのみ、支払確認が可能です。

返金

支払が成功したオーダーについてのみ、返金ができます。

- 0 成功
- 1 ネットワークエラー

取消

支払が成功したオーダーについてのみ、取消ができます。

- 0 成功

入金

- 0 成功