

クレジットカード決済
トークン方式
3Dセキュア2.0
(複数カード情報登録ワンタッチ課金)

■概要

- :複数カード情報登録ワンタッチ課金について……………P.3
- :対応ブラウザ……………P.4

■ユーザー、カード情報仕様

- :ユーザー登録仕様……………P.6
- :ユーザー参照仕様……………P.7
- :ユーザー更新仕様……………P.8
- :ユーザー削除仕様……………P.9
- :カード情報登録仕様……………P.10
- :カード情報参照仕様……………P.11
- :カード情報更新仕様……………P.12
- :カード情報削除仕様……………P.13
- :ユーザー・カード情報一括登録決済(商品登録なし)仕様…P.14
- :ユーザー・カード情報一括登録決済(商品登録あり)仕様…P.15
- :一括登録決済結果通知のパラメータ仕様……………P.16

■決済処理(ワンタッチ課金)仕様

- :ワンタッチ課金(商品登録なし)仕様……………P.18
- :ワンタッチ課金(商品登録あり)仕様……………P.19
- :ワンタッチ課金結果通知のパラメータ仕様……………P.20

■トークン作成、3Dセキュア2.0認証仕様

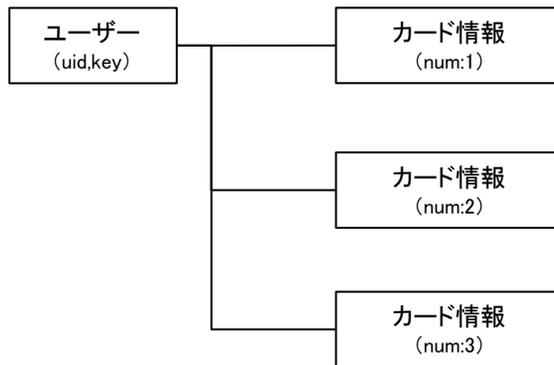
- :ポップアップ方式 実装方法(サンプル)……………P.22
- :カスタマイズ方式 実装方法(サンプル)……………P.24
- :トークン作成(ポップアップ方式)仕様……………P.26
- :トークン作成(カスタマイズ方式)仕様……………P.27
- :3Dセキュア2.0認証
 - カード情報登録、カード情報更新)仕様……………P.28
 - ユーザー・カード情報一括登録決済(商品登録なし)仕様…P.29
 - ユーザー・カード情報一括登録決済(商品登録あり)仕様…P.30

複数カード情報登録ワンタッチ課金について

■複数カード情報登録ワンタッチ課金とは

複数カード情報登録ワンタッチ課金は、事前に「ユーザー」とそれに紐づく「カード情報」を登録しその登録した情報を用いて決済(ワンタッチ課金)を行います。

1ユーザーに対して複数のカード情報を登録することができます。



決済時に都度カード情報を入力しなくてよくなるので、加盟店様の任意のタイミングで決済を行うことができます。

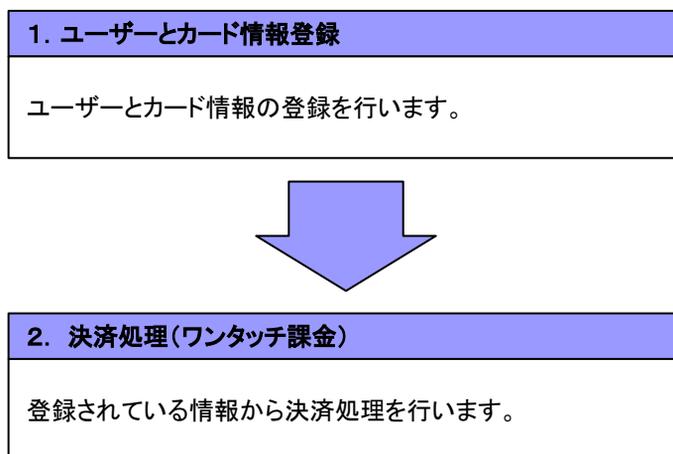
■トークン作成、3Dセキュア2.0認証

カード情報を登録、変更する際には **トークン作成** と **3Dセキュア2.0認証** が必要です。

トークン作成と3Dセキュア2.0認証の画面フロー、処理フローについては以下の仕様書をご覧ください。

「クレジットカード決済 トークン方式 3Dセキュア2.0(概要、処理フロー、画面フロー)」

■処理フロー



対応ブラウザ

対応ブラウザは以下です。

- Microsoft Edge 最新版
- Google Chrome 最新版
- Firefox 最新版
- Safari 最新版

※JavaScriptの利用できない端末(一部フィーチャーフォン等)では、トークン決済を行う事が出来ません。



ユーザー、カード情報仕様

■リクエスト

・接続先

https://credit.j-payment.co.jp/gateway/carduser.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「0」ユーザー登録	○	半角数字(1)
ユーザーID	uid	作成するユーザーのID	○	半角英数(50)
メールアドレス	em	メールアドレス	※1	半角英数(254)
電話番号	pn	電話番号	※1	半角数字(15)
その他データ		key=value形式で任意で設定するパラメータ ※2		1024バイト以内

<※1>

メールアドレス、電話番号のどちらかが必須となります。

<※2>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,,012345,vnjpBycDO5

項目	フィールド	詳細	形式
結果	rst	1:正常 2:エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーのID	半角英数(50)
キー	key	ユーザーIDに紐づくキー	半角英数(10)
その他データ		リクエスト時に指定したその他データ	1024バイト以内

■リクエスト

・接続先

<https://credit.j-payment.co.jp/gateway/carduser.aspx>

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「3」ユーザー参照	○	半角数字(1)
ユーザーID	uid	登録時に指定したユーザーID	○	半角英数(50)
その他データ		key=value形式で任意で設定するパラメータ ※1		1024バイト以内

<※1>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,,012345,vnjpBycDO5,sample@sample.jp,09011111111

項目	フィールド	詳細	形式
結果	rst	1:正常 2:エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーのID	半角英数(50)
キー	key	ユーザーIDに紐づくキー	半角英数(10)
メールアドレス	em	メールアドレス	半角英数(254)
電話番号	pn	電話番号	半角数字(15)
その他データ		リクエスト時に指定したその他データ	1024バイト以内

■リクエスト

・接続先

https://credit.j-payment.co.jp/gateway/carduser.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「1」ユーザー更新	○	半角数字(1)
ユーザーID	uid	登録時に指定したユーザーID	○	半角英数(50)
キー	key	登録時に発行されたユーザーIDに紐づくキー	○	半角英数(100)
メールアドレス	em	メールアドレス	※1	半角英数(254)
電話番号	pn	電話番号	※1	半角数字(15)
その他データ		key=value形式で任意で設定するパラメータ ※2		1024バイト以内

<※1>

メールアドレス、電話番号のどちらかが必須となります。

※すでにそのユーザーに紐づくメールアドレス、電話番号が登録済みの場合は除く

<※2>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,,012345,vnjpBycDO5

項目	フィールド	詳細	形式
結果	rst	1:正常 2:エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーのID	半角英数(50)
キー	key	リクエスト時に指定したキー	半角英数(10)
その他データ		リクエスト時に指定したその他データ	1024バイト以内

ユーザー削除 仕様

■リクエスト

・接続先

<https://credit.j-payment.co.jp/gateway/carduser.aspx>

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「2」ユーザー削除	○	半角数字(1)
ユーザーID	uid	登録時に指定したユーザーID	○	半角英数(50)
キー	key	登録時に発行されたユーザーIDに紐づくキー	○	半角英数(100)
その他データ		key=value形式で任意で設定するパラメータ ※1		1024バイト以内

<※1>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,,012345,vnjpBycDO5

項目	フィールド	詳細	形式
結果	rst	1:正常 2:エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーのID	半角英数(50)
キー	key	リクエスト時に指定したキー	半角英数(10)
その他データ		リクエスト時に指定したその他データ	1024バイト以内

■リクエスト

・接続先

https://credit.j-payment.co.jp/gateway/carduser_token.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

※カード情報の登録の際には **トークン作成** と **3Dセキュア2.0認証** が必須です。後述の「トークン作成、3Dセキュア2.0認証仕様」をご覧ください。

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「4」カード情報登録	○	半角数字(1)
ユーザーID	uid	カード情報を登録するユーザーID ※1	○	半角英数(50)
キー	key	カード情報を登録するユーザーのキー	○	半角英数(10)
トークン情報	tkn	トークン(トークン作成処理で取得)	○	半角英数記号
その他データ		key=value形式で任意で設定するパラメータ ※2		1024バイト以内

<※1>

ユーザー情報からメールアドレス、電話番号を取得し、カード登録を行います。その為、ユーザーに紐づくメールアドレス、電話番号が登録されていない場合、エラーとなります。

<※2>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,,012345,vnjpBycDO5,1

項目	フィールド	詳細	形式
結果	rst	1:正常 2:エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーID	半角英数(50)
キー	key	リクエスト時に指定したキー	半角英数(10)
カード情報連番	num	カードを識別する為の連番	半角数字
その他データ		リクエスト時に指定したその他データ	1024バイト以内

■リクエスト

・接続先

<https://credit.j-payment.co.jp/gateway/carduser.aspx>

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「6」カード情報参照	○	半角数字(1)
ユーザーID	uid	カード情報を参照するユーザーのID	○	半角英数(50)
キー	key	カード情報を参照するユーザーのキー	○	半角英数(10)
カード情報連番	num	カード情報を参照するカード情報連番	○	半角数字
その他データ		key=value形式で任意で設定するパラメータ ※1		1024バイト以内

<※1>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,012345,vnjpBycDO5,1,*****9999,2512,sample,sample

項目	フィールド	詳細	形式
結果	rst	1: 正常 2: エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーID	半角英数(50)
キー	key	リクエスト時に指定したキー	半角英数(10)
カード情報連番	num	リクエスト時に指定したカード情報連番	半角数字
カード番号	cn	登録されているクレジットカード番号 ※下4桁のみ 例「*****9999」	記号(10~12)+ 半角数字(4)
有効期限	ed	登録されているクレジットカードの有効期限(YYMM形式)	半角数字(4)
カード名義(名)	fn	登録されているカード名義(名)	半角英数
カード名義(性)	ln	登録されているカード名義(性)	半角英数
その他データ		リクエスト時に指定したその他データ	1024バイト以内

カード情報更新 仕様

■リクエスト

・接続先

https://credit.j-payment.co.jp/gateway/carduser_token.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

※カード情報の更新の際には **トークン作成** と **3Dセキュア2.0認証** が必須です。
後述の「トークン作成、3Dセキュア2.0認証仕様」をご覧ください。

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「4」カード情報更新	○	半角数字(1)
ユーザーID	uid	カード情報を更新するユーザーID ※1	○	半角英数(50)
キー	key	カード情報を更新するユーザーのキー	○	半角英数(10)
カード情報連番	num	カード情報を更新するカード情報連番	○	半角数字
トークン情報	tkn	トークン(トークン作成処理で取得)	○	半角英数記号
その他データ		key=value形式で任意で設定するパラメータ ※2		1024バイト以内

<※1>

ユーザー情報からメールアドレス、電話番号を取得し、カード更新を行います。
その為、ユーザーに紐づくメールアドレス、電話番号が登録されていない場合、エラーとなります。

<※2>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,,012345,vnjpBycDO5,1

項目	フィールド	詳細	形式
結果	rst	1:正常 2:エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーID	半角英数(50)
キー	key	リクエスト時に指定したキー	半角英数(10)
カード情報連番	num	リクエスト時に指定したカード情報連番	半角数字
その他データ		リクエスト時に指定したその他データ	1024バイト以内

カード情報削除 仕様

■リクエスト

・接続先

https://credit.j-payment.co.jp/gateway/carduser.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

・リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
変更タイプ	cmd	「5」カード情報削除	○	半角数字(1)
ユーザーID	uid	カード情報を削除するユーザーID	○	半角英数(50)
キー	key	カード情報を削除するユーザーのキー	○	半角英数(10)
カード情報連番	num	カード情報を削除するカード情報連番	○	半角数字
その他データ		key=value形式で任意で設定するパラメータ ※1		1024バイト以内

<※1>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

■レスポンス

・形式

HTML出力 カンマ区切り

・サンプル

1,,012345,vnjpBycDO5,1

項目	フィールド	詳細	形式
結果	rst	1:正常 2:エラー	半角数字
エラーコード	ec	エラー時(rst=2)のときのみ返却	半角英数
ユーザーID	uid	リクエスト時に指定したユーザーID	半角英数(50)
キー	key	リクエスト時に指定したキー	半角英数(10)
カード情報連番	num	リクエスト時に指定したカード情報連番	半角数字
その他データ		リクエスト時に指定したその他データ	1024バイト以内

ユーザー・カード情報一括登録決済(商品登録なし)仕様

ユーザーとカード情報を一括で登録しながら商品登録なしの決済を行います。

■接続先

https://creditj-payment.co.jp/gateway/gateway_token.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

■リクエスト方式

GET もしくは POST

※一括登録決済の際には **トークン作成**と**3Dセキュア2.0認証**が必須です。
後述の「トークン作成、3Dセキュア2.0認証仕様」をご覧ください。

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
ジョブタイプ	jb	決済処理方法 「AUTH」(仮売上) ※1 「CAPTURE」(仮実同時売上)	○	半角英字
決済結果返信方法	rt	「0」キックバック 「1」レスポンス	○	半角数字(1)
店舗オーダー番号	cod	オーダーを識別する為に任意で設定するパラメータ		50バイト以内
ユーザーID	uid	作成するユーザーのID	○	半角英数(50)
トークン情報	tkn	トークン(トークン作成処理で取得)	○	半角英数記号
メールアドレス	em	メールアドレス	※4	半角英数(254)
電話番号	pn	電話番号	※4	半角数字(15)
商品金額	am	商品金額 ※2	○	半角数字
税金額	tx	税金額 ※2	○	半角数字
送料	sf	送料 ※2	○	半角数字
その他データ		key=value形式で任意で設定するパラメータ ※3		1024バイト以内

<※1>

AUTH(仮売上)を指定した場合、決済後90日以内に実売上を行ってください。
実売上を行って頂かないとお客様への請求が確定いたしません。
CAPTURE(仮実同時売上)を指定した場合、決済時に自動的に実売上が行われます。

<※2>

【決済金額+税金額+送料】の値がカード会社に送信されます。
商品金額に送料、税額を含む場合、tx sf は0を指定してください。

<※3>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

<※4>

メールアドレス、電話番号のどちらかが必須となります。

■レスポンス

後述の「一括登録決済結果通知仕様」を参照

ユーザー・カード情報一括登録決済(商品登録あり)仕様

ユーザーとカード情報を一括で登録しながら商品登録ありの決済を行います。

■接続先

https://credit.j-payment.co.jp/gateway/gateway_token.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

■リクエスト方式

GET もしくは POST

※一括登録決済の際には **トークン作成**と**3Dセキュア2.0認証**が必須です。
後述の「トークン作成、3Dセキュア2.0認証仕様」をご覧ください。

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
決済結果返信方法	rt	「0」キックバック 「1」レスポンス	○	半角数字(1)
店舗オーダー番号	cod	オーダーを識別する為に任意で設定するパラメータ		50バイト以内
ユーザーID	uid	作成するユーザーのID	○	半角英数(50)
トークン情報	tkn	トークン(トークン作成処理で取得)	○	半角英数記号
メールアドレス	em	メールアドレス	※2	半角英数(254)
電話番号	pn	電話番号	※2	半角数字(15)
商品コード	iid	商品コード	○	半角英数(50)
その他データ		key=value形式で任意で設定するパラメータ ※1		1024バイト以内

<※1>
「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

<※2>
メールアドレス、電話番号のどちらかが必須となります。

■レスポンス

後述の「一括登録決済結果通知のパラメータ仕様」を参照

一括登録決済結果通知のパラメータ仕様

■キックバック(rt=0)仕様

・形式

URLパラメータ

・通知先

管理画面の「決済結果通知設定」の「決済結果通知URL」で設定したURL

※通知の成功判定を行っているため、通知先ではContentLengthが0以上が出力されるよう実装してください。

・サンプル

決済結果通知URL?gid=10000001&rst=1&ap=0001112&ec=&god=10000002&cod=test &am=1000&tx=80&sf=500&ta=1580&uid= 012345
&key=vnpjBycDO5&num=1

■レスポンス(rt=1)仕様

・形式

HTML出力 カンマ区切り

・サンプル

1095729,1,TestMod,,205685,,1000,2,4,1006,,012345,vnpjBycDO5,1

項目	フィールド	詳細	レスポンス出力順	形式
決済番号	gid	決済ごとに発行される番号	1	半角数字
決済結果	rst	1:決済成功 2:決済失敗	2	半角数字(1)
カード会社承認番号	ap	カード会社から発行される承認番号	3	半角英数(6~7)
エラーコード	ec	エラーコード	4	半角英数(12)
オーダーコード	god	決済ごとに発行されるコード	5	半角数字
店舗オーダー番号	cod	決済リクエスト時に指定したcod	6	50バイト以内
決済金額	am	決済リクエスト時に指定したam	7	半角数字
税金額	tx	決済リクエスト時に指定したtx	8	半角数字
送料	sf	決済リクエスト時に指定したsf	9	半角数字
合計金額	ta	am,tx,sfの合計値	10	半角数字
発行ID	id	弊社システムより発行されたID ※1	11	半角英数(50字以内)
発行パスワード	ps	弊社システムより発行されたパスワード ※1	12	半角英数(50字以内)
ユーザーID	uid	リクエスト時に指定したユーザーのID	13	半角英数(50)
キー	key	ユーザーIDに紐づくキー	14	半角英数(10)
カード情報連番	num	カードを識別する為の連番	15	半角数字
その他		決済リクエスト時に指定したその他データ	16	1024バイト以内

<※1>

コンテンツ(ID/PW)の商品登録した商品で、商品登録あり決済をした場合のみ返却されます。
ユーザー、カード情報一括登録のuid,keyとは関係ありませんのでご注意ください。

決済処理(ワンタッチ課金)仕様

ワンタッチ課金(商品登録なし)仕様

■接続先

https://credit.j-payment.co.jp/gateway/onetouch.aspx

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

■リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
決済結果返信方法	rt	「0」キックバック 「1」レスポンス	○	半角数字(1)
処理タイプ	cmd	「4」	○	半角数字(1)
店舗オーダー番号	cod	オーダーを識別する為に任意で設定するパラメータ		50バイト以内
ユーザーID	uid	カード情報を使用するユーザーID	○	半角英数(50)
キー	key	カード情報を使用するユーザーのキー	○	半角英数(10)
カード情報連番	num	カード情報を使用するカード情報連番	○	半角数字
ジョブタイプ	jb	「AUTH」(仮売上) 「CAPTURE」(仮同時売上)	○	半角英字
商品金額	am	商品金額	○	半角数字
税金額	tx	税金額	○	半角数字
送料	sf	送料	○	半角数字
メールアドレス	em	メールアドレス ※1		半角英数(254)
電話番号	pn	電話番号 ※1		半角数字(15)
支払方法	md	未指定の場合は「一括払い」となります 「10」一括払い 「61」分割払い ※2 「80」リボ払い ※2		半角数字(2)
分割回数	pt	「3,5,6,10,12,15,18,20,24」から選択	※4	半角数字
その他データ		key=value形式で任意で設定するパラメータ ※3		1024バイト以内

<※1>

指定の場合は、登録されているユーザーの情報が更新されます。

<※2>

分割払い・リボ払いは契約時のみ指定可

<※3>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

<※4>

md=61(分割払い)指定時は必須

■レスポンス

後述の「ワンタッチ課金結果通知のパラメータ仕様」を参照

ワンタッチ課金(商品登録あり)仕様

■接続先

<https://credit.j-payment.co.jp/gateway/onetouch.aspx>

※接続元IP認証を行っておりますため、加盟店様のサーバ(固定されたIPアドレス)からリクエストしてください。
エンドユーザー様のブラウザからリクエストした場合ER003(送信元IPエラー)が発生します。

■リクエスト方式

GET もしくは POST

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
決済結果返信方法	rt	「0」キックバック 「1」レスポンス	○	半角数字(1)
処理タイプ	cmd	「4」	○	半角数字(1)
店舗オーダー番号	cod	オーダーを識別する為に任意で設定するパラメータ		50バイト以内
ユーザーID	uid	カード情報を使用するユーザーID	○	半角英数(50)
キー	key	カード情報を使用するユーザーのキー	○	半角英数(10)
カード情報連番	num	カード情報を使用するカード情報連番	○	半角数字
商品コード	iid	商品コード	○	半角英数(50)
メールアドレス	em	メールアドレス ※1		半角英数(254)
電話番号	pn	電話番号 ※1		半角数字(15)
支払方法	md	未指定の場合は「一括払い」となります 「10」一括払い 「61」分割払い ※2 「80」リボ払い ※2		半角数字(2)
分割回数	pt	「3,5,6,10,12,15,18,20,24」から選択	※4	半角数字
その他データ		key=value形式で任意で設定するパラメータ ※3		1024バイト以内

<※1>

指定の場合は、登録されているユーザーの情報が更新されます。

<※2>

分割払い・リボ払いは契約時のみ指定可

<※3>

「改行コード」「スペース」を送信される場合、必ずエンコード処理を行ってください

<※4>

md=61(分割払い)指定時は必須

■レスポンス

後述の「ワンタッチ課金結果通知のパラメータ仕様」を参照

ワンタッチ課金結果通知のパラメータ仕様

決済結果通知の仕様は以下の通りです。

■キックバック(rt=0)仕様

・形式

URLパラメータ

・通知先

管理画面の「決済結果通知設定」の「決済結果通知URL」で設定したURL

※通知の成功判定を行っているため、通知先ではContentLengthが0以上が出力されるよう実装してください。

・サンプル

決済結果通知URL?gid=10000001&rst=1&ap=0001112&ec=&god=10000002&cod=test &am=1000&tx=80&sf=500&ta=1580

■レスポンス(rt=1)仕様

・形式

HTML出力 カンマ区切り

・サンプル

1095729,1,TestMod,,205685,,1000,2,4,1006,..

項目	フィールド	詳細	レスポンス出力順	形式
決済番号	gid	決済ごとに発行される番号	1	半角数字
決済結果	rst	1:決済成功 2:決済失敗	2	半角数字(1)
カード会社承認番号	ap	カード会社から発行される承認番号	3	半角英数(6~7)
エラーコード	ec	エラーコード ※1	4	半角英数(12)
オーダーコード	god	決済ごとに発行されるコード	5	半角数字
店舗オーダー番号	cod	決済リクエスト時に指定したcod	6	50バイト以内
決済金額	am	決済リクエスト時に指定したam	7	半角数字
税金額	tx	決済リクエスト時に指定したtx	8	半角数字
送料	sf	決済リクエスト時に指定したsf	9	半角数字
合計金額	ta	am,tx,sfの合計値	10	半角数字
発行ID	id	弊社システムより発行されたID ※2	11	半角英数(50字以内)
発行パスワード	ps	弊社システムより発行されたパスワード ※2	12	半角英数(50字以内)
その他		決済リクエスト時に指定したその他データ	13	1024バイト以内

<※1>

カード会社通信後エラー(Gから始まるエラーコード)のみが対象です。

<※2>

コンテンツ(ID/PW)の商品登録した商品で、商品登録あり決済をした場合のみ返却されます。

ユーザー、カード情報一括登録のuid,keyとは関係ありませんのでご注意ください。

トークン作成、3Dセキュア2.0認証仕様

ポップアップ方式 実装方法(サンプル)

HTMLの実装

①jQuery, CPToken.js, EMV3DSAdapter.jsを読み込みます。

```
<head>
<meta charset="utf-8" />
<script type="text/javascript"
src="https://credit.j-payment.co.jp/gateway/js/jquery.js" ></script> <!--※1-->
<script type="text/javascript"
src="https://credit.j-payment.co.jp/gateway/js/CPToken.js" ></script><!--※2-->
<script type="text/javascript"
src="https://credit.j-payment.co.jp/gateway/js/EMV3DSAdapter.js" ></script><!--※3-->
</head>
```

※1 加盟店様側でjQueryの読み込み部分を実装されている場合は不要です。

※2 トークン決済用

※3 3Dセキュア2.0用

②トークン用Hidden要素、トークンポップアップ用要素、3Dセキュア用要素を追加します。

```
<form id="mainform" method="POST" action="https://credit.j-payment.co.jp/gateway/carduser_token.aspx" >
  <input type="hidden" value="" id="aid" name="aid">
  <input type="hidden" value="4" id="cmd" name="cmd">
  <input type="hidden" value="" id="uid" name="uid">
  <input type="hidden" value="" id="key" name="key">
  <!--トークン作成処理後にid:tkn要素に値がセットされます-->
  <input type="hidden" value="" id="tkn" name="tkn" >
  <!-- トークンポップアップ表示用 -->
  <div id="CARD_INPUT_FORM" ></div>
  <!-- 3Dセキュアポップアップ表示用 -->
  <div id="EMV3DS_INPUT_FORM" ></div>
  <input type="button" value="登録する" onclick="doPurchase()" />
</form>
```

次のページへ続く

ポップアップ方式 実装方法(サンプル)

Javascriptの実装

```
/**
 * 1. クレジットカードトークンの作成
 * トークン作成用の関数を準備します。
 */
function doPurchase() {

    // トークン作成処理を呼び出します。
    CPToken.CardInfo({
        aid: '000000' // 店舗IDを設定します。
    }, execAuth); // 3Dセキュア2.0認証用関数をコールバックにセットします。
}

/**
 * 2. 3Dセキュア2.0の認証を実行
 * 3Dセキュア2認証用関数の準備をします。
 * 引数はresultCode, errMsgの2つを受け取れるようにします。
 */
function execAuth(resultCode, errMsg) {
    if (resultCode != "Success") {
        // 戻り値がSuccess以外の場合はエラーメッセージを表示します
        window.alert(errMsg);
    } else {

        // 3Dセキュア2.0認証処理を呼び出します。
        ThreeDSAdapter.authenticate({
            aid: '000000', // 店舗IDを設定します。
            tkn: $("#tkn").val(), // トークン作成後にtkn要素に値が入力されます
            uid: '000000', // ユーザーIDを設定します。
        }, execPurchase); // 決済実行用の関数をコールバックにセットします。
    }
}

/**
 * 3. 決済処理の実行
 * 決済実行用の関数を準備します。
 * 引数はresultCode, errMsgの2つを受け取れるようにします。
 */
function execPurchase(resultCode, errMsg) {
    if (resultCode != "Success") {
        // 戻り値がSuccess以外の場合はエラーメッセージが返却されます。
        window.alert(errMsg);
    } else {
        // 決済リクエストを実行する処理の実装をします。
        $("#mainform").submit();
    }
}
```

カスタマイズ方式 実装方法(サンプル)

HTMLの実装

①jQuery, CPToken.js, EMV3DSAdapter.jsを読み込みます。

```
<head>
<meta charset="utf-8" />
<script type="text/javascript"
src="https://credit.j-payment.co.jp/gateway/js/jquery.js" ></script> <!--※1-->
<script type="text/javascript"
src="https://credit.j-payment.co.jp/gateway/js/CPToken.js" ></script><!--※2-->
<script type="text/javascript"
src="https://credit.j-payment.co.jp/gateway/js/EMV3DSAdapter.js" ></script><!--※3-->
</head>
```

※1 加盟店様側でjQueryの読み込み部分を実装されている場合は不要です。

※2 トークン決済用

※3 3Dセキュア2.0用

②トークン用Hidden要素、トークンポップアップ用要素、3Dセキュア用要素を追加します。

```
<form id="mainform" method="POST" action="https://credit.j-payment.co.jp/gateway/carduser_token.aspx" >
  <input type="hidden" value="" id="aid" name="aid">
  <input type="hidden" value="4" id="cmd" name="cmd">
  <input type="hidden" value="" id="uid" name="uid">
  <input type="hidden" value="" id="key" name="key">
  <!-- カード番号 -->
  <input type="text" value="" id="cn" name="cn">
  <!-- カード有効期限 -->
  <input type="text" value="" id="ed_year" name="ed_year">
  <input type="text" value="" id="ed_month" name="ed_month">
  <!-- カード名義 -->
  <input type="text" value="" id="fn" name="fn">
  <input type="text" value="" id="ln" name="ln">
  <!-- トークン作成処理後にid:tkn要素に値がセットされます -->
  <input type="hidden" value="" id="tkn" name="tkn" >
  <!-- 3Dセキュアポップアップ表示用 -->
  <div id="EMV3DS_INPUT_FORM"></div>
  <input type="button" value="購入する" onClick="doPurchase()" >
</form>
```

次のページへ続く

カスタマイズ方式 実装方法(サンプル)

Javascriptの実装

```
/**
 * 1. クレジットカードトークンの作成
 * トークン作成用の関数を準備します。
 */
function doPurchase() {
    // トークン作成処理を呼び出します。
    CPToken.TokenCreate ({
        aid: '000000', // 店舗IDを設定します。
        cn: $("#cn").val(),
        ed: $("#ed_year").val() + $("#ed_month").val(),
        fn: $("#fn").val(),
        ln: $("#ln").val()
    }, execAuth); // 3Dセキュア2.0認証用関数をコールバックにセットします。
}

/**
 * 2. 3Dセキュア2.0の認証を実行
 * 3Dセキュア2認証用関数の準備をします。
 * 引数はresultCode, errMsgの2つを受け取れるようにします。
 */
function execAuth(resultCode, errMsg) {
    if (resultCode != "Success") {
        // 戻り値がSuccess以外の場合はエラーメッセージを表示します
        window.alert(errMsg);
    } else {

        // 3Dセキュア2.0認証処理を呼び出します。
        ThreeDSAdapter.authenticate({
            aid: '000000', // 店舗IDを設定します。
            tkn: $("#tkn").val(), // トークン作成後にtkn要素に値が入力されます
            uid: '000000', // ユーザーIDを設定します。
        }, execPurchase); // 決済実行用の関数をコールバックにセットします。
    }
}

/**
 * 3. 決済処理の実行
 * 決済実行用の関数を準備します。
 * 引数はresultCode, errMsgの2つを受け取れるようにします。
 */
function execPurchase(resultCode, errMsg) {
    if (resultCode != "Success") {
        // 戻り値がSuccess以外の場合はエラーメッセージが返却されます。
        window.alert(errMsg);
    } else {
        // カード情報を消去
        $("#cn").val("");
        $("#ed_year").val("");
        $("#ed_month").val("");
        $("#fn").val("");
        $("#ln").val("");

        // 決済リクエストを実行する処理の実装をします。
        $("#mainform").submit();
    }
}
```

トークン作成(ポップアップ方式)仕様

トークン作成(ポップアップ方式)の仕様は以下の通りです。

■接続先

<https://credit.j-payment.co.jp/gateway/js/CPToken.js>

■ファイル

CPToken.js

■関数

CPToken.CardInfo(tokenRequest,callback)

・CardInfo引数仕様

引数名	説明	備考
tokenRequest	トークン作成に必要なパラメータ	Json形式
callback	実行後にコールバックされる関数 ※ThreeDSAdapter.authenticate関数を指定してください。	

・tokenRequest仕様

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)

■レスポンス

なし(処理完了時にHTMLのid=tkn要素のvalueにトークンの文字列が設定されます)

■その他

弊社のJavaScript内で使用しているため、実装の際に「rpemvtds」クラスを使用しないでください。

トークン作成(カスタマイズ方式)仕様

トークン作成(カスタマイズ方式)の仕様は以下の通りです。

■接続先

https://creditj-payment.co.jp/gateway/js/CPToken.js

■ファイル

CPToken.js

■関数

CPToken.TokenCreate(tokenRequest, callback)

・TokenCreate引数仕様

引数名	説明	備考
tokenRequest	トークン作成に必要なパラメータ	Json形式
callback	実行後にコールバックされる関数 ※ThreeDSAdapter.authenticate関数を指定してください。	

・tokenRequest仕様

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
カード番号	cn	カード番号	○	半角数字
カード有効期限	ed	カードの有効期限(YMMM形式)	○	半角数字
カード名義(名)	fn	カードの名義(名)	○	半角英数記号(43) ※1
カード名義(姓)	ln	カードの名義(姓)	○	半角英数記号(43) ※1
セキュリティコード	cvv	VISA・MASTER・JCB・DINERSは カード裏面の3桁の番号 AMEXのみカード表面の4桁の番号	※2	半角数字

<※1>

- ・記号はピリオド、ハイフン(いずれも半角)
- ・カード名義(名)と(姓)の合計が44文字以内

<※2>

セキュリティコードを利用の場合は必須(ご利用には契約が必要です)

■レスポンス

なし(処理完了時にHTMLのid=tkn要素のvalueにトークンの文字列が設定されます)

■その他

弊社のJavaScript内で使用しているため、実装の際に「rpemvtds」クラスを使用しないでください。

3Dセキュア2.0認証(カード情報登録,カード情報更新)仕様

3Dセキュア2.0認証(カード情報登録,カード情報更新)の仕様は以下の通りです。

■接続先

<https://credit.j-payment.co.jp/gateway/js/EMV3DSAdapter.js>

■ファイル

EMV3DSAdapter.js

■関数

ThreeDSAdapter.authenticate(authenticateRequest,callback)

・ authenticate引数仕様

引数名	説明	備考
authenticateRequest	3DS認証に必要なパラメータ	Json形式
callback	実行後にコールバックされる関数 ※決済処理を行う関数を指定してください。	

・ authenticateRequest仕様

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
トークン情報	tkn	トークン(トークン作成処理で取得)	○	半角英数記号
ユーザーID	uid	カード情報を登録するユーザーのID	○	半角英数記号

・ callback仕様

項目	説明	備考
結果コード	成功時:Success 失敗時:エラーコード	“Success”はAPI呼び出しの成功を意味し、3DS認証の成功を表すものではありません。 3DS認証の成否は、後続で実施する決済処理のレスポンスで出力されます。
メッセージ	結果コードの詳細メッセージ	

■レスポンス

なし

■その他

弊社のJavaScript内で使用しているため、実装の際に「rpemvtds」クラスを使用しないでください。

3Dセキュア2.0認証 ユーザー・カード情報一括登録決済(商品登録なし)仕様

3Dセキュア2.0認証 ユーザー・カード情報一括登録決済(商品登録なし)仕様は以下の通りです。

■接続先

<https://credit.j-payment.co.jp/gateway/js/EMV3DSAdapter.js>

■ファイル

EMV3DSAdapter.js

■関数

ThreeDSAdapter.authenticate(authenticateRequest,callback)

・ authenticate引数仕様

引数名	説明	備考
authenticateRequest	3DS認証に必要なパラメータ	Json形式
callback	実行後にコールバックされる関数 ※決済処理を行う関数を指定してください。	

・ authenticateRequest仕様

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
トークン情報	tkn	トークン(トークン作成処理で取得)	○	半角英数記号
商品金額	am	商品金額 ※1	○	半角数字
税金額	tx	税金額 ※1	○	半角数字
送料	sf	送料 ※1	○	半角数字
メールアドレス	em	メールアドレス ※1	※2	半角英数(254)
電話番号	pn	電話番号 ※1	※2	半角数字(15)

< ※1 >

ユーザー・カード情報一括登録決済で指定した値と同じ値を指定してください。

< ※2 >

メールアドレス、電話番号のどちらかが必須となります。

< ※3 >

こちらでuidパラメータを指定した場合エラーになりますのでご注意ください。

・ callback仕様

項目	説明	備考
結果コード	成功時: Success 失敗時: エラーコード	“Success”はAPI呼び出しの成功を意味し、3DS認証の成功を表すものではありません。 3DS認証の成否は、後続で実施する決済処理のレスポンスで出力されます。
メッセージ	結果コードの詳細メッセージ	

■レスポンス

なし

■その他

弊社のJavaScript内で使用しているため、実装の際に「rpemvtds」クラスを使用しないでください。

3Dセキュア2.0認証 ユーザー・カード情報一括登録決済(商品登録あり)仕様

3Dセキュア2.0認証 ユーザー・カード情報一括登録決済(商品登録あり)の仕様は以下の通りです。

■接続先

<https://credit.j-payment.co.jp/gateway/js/EMV3DSAdapter.js>

■ファイル

EMV3DSAdapter.js

■関数

ThreeDSAdapter.authenticate(authenticateRequest,callback)

・ authenticate引数仕様

引数名	説明	備考
authenticateRequest	3DS認証に必要なパラメータ	Json形式
callback	実行後にコールバックされる関数 ※決済処理を行う関数を指定してください。	

・ authenticateRequest仕様

項目	フィールド	詳細	必須	形式
店舗ID	aid	契約時に発行される店舗のID	○	半角数字(6)
トークン情報	tkn	トークン(トークン作成処理で取得)	○	半角英数記号
商品コード	iid	商品コード ※1	○	半角英数(50)
メールアドレス	em	メールアドレス ※1	※2	半角英数(254)
電話番号	pn	電話番号 ※1	※2	半角数字(15)

<※1>

ユーザー・カード情報一括登録決済と同じ値を指定してください。

<※2>

メールアドレス、電話番号のどちらかが必須となります。

<※3>

こちらでuidパラメータを指定した場合エラーになりますのでご注意ください。

・ callback仕様

項目	説明	備考
結果コード	成功時: Success 失敗時: エラーコード	“Success”はAPI呼び出しの成功を意味し、3DS認証の成功を表すものではありません。 3DS認証の成否は、後続で実施する決済処理のレスポンスで出力されます。
メッセージ	結果コードの詳細メッセージ	

■レスポンス

なし

■その他

弊社のJavaScript内で使用しているため、実装の際に「rpemvtds」クラスを使用しないでください。