

PrIn FinTech Co., Ltd. (HQ)
SJ Infinite I Business Complex,
21st Floor Vibhavadi Rangsit Road,
Chompol, Chatuchak, Bangkok, Thailand. 10900
Tax ID: 0105535090912



V.1.0.2

CHILLCREDIT MERCHANT INTEGRATION

MANUAL DOCUMENT [ภาษาไทย]



 +66.2.107.7788	 @chillpay	 chillpay payment gateway
 CS.CHILLPAY	 @ChillPay_VI	 help@chillpay.co



สารบัญ

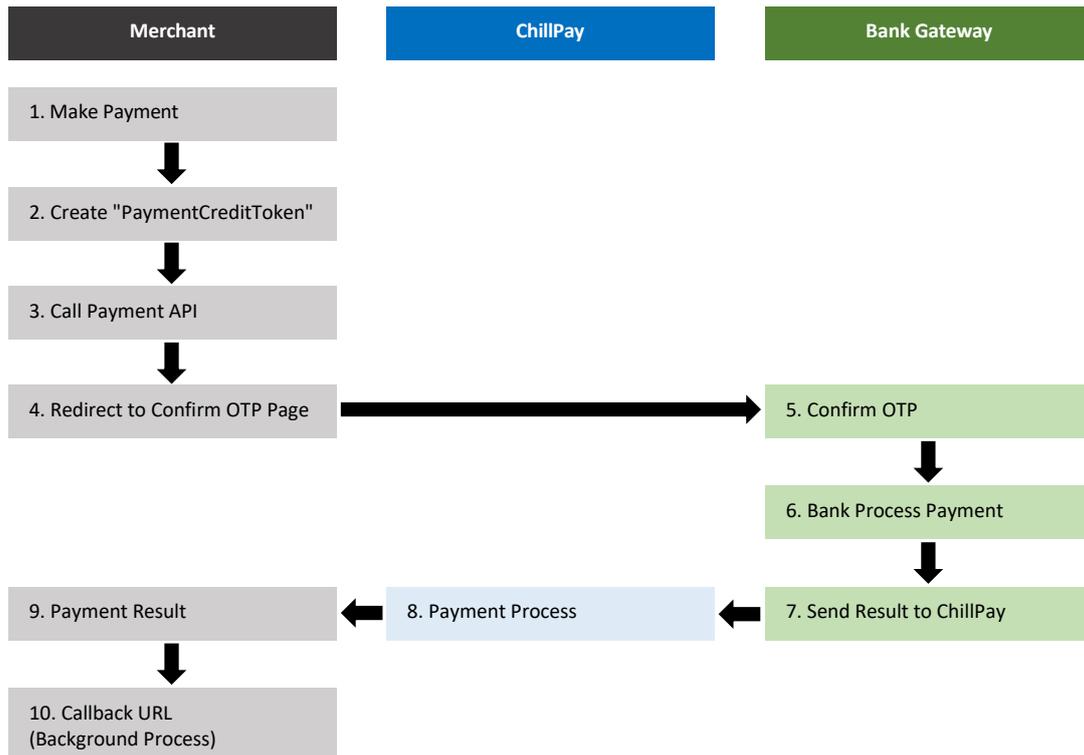
A. คู่มือในการพัฒนาระบบ เพื่อเชื่อมต่อกับบริการ ChillCredit	3
1. การเชื่อมต่อบริการ Credit Card Inline (รองรับเฉพาะการจ่ายเงินโดยใช้ Credit Card เท่านั้น)	3
1.1 System Flow ของการเชื่อมต่อ Credit Card Inline	3
1.2 การเชื่อมต่อกับระบบ Credit Card Inline.....	4
2. การเชื่อมต่อบริการ ChillCredit Public Api	46
2.1 API สำหรับ CreatePaymentCreditToken.....	46
B. ข้อมูลการทดสอบบน Sandbox	50
C. ภาคผนวก	51
ภาคผนวก ก.	51
ภาคผนวก ข.	51
ภาคผนวก ค.	53
ภาคผนวก ง.....	53
ภาคผนวก จ.	54

A. คู่มือในการพัฒนาระบบ เพื่อเชื่อมต่อกับบริการ ChillCredit

1. การเชื่อมต่อบริการ Credit Card Inline (รองรับเฉพาะการจ่ายเงินโดยใช้ Credit Card เท่านั้น)

วิธีการเชื่อมต่อแบบ Credit Card Inline (รองรับเฉพาะการจ่ายเงินโดยใช้ Credit Card เท่านั้น) เป็นรูปแบบการเชื่อมต่อที่หน้าเว็บร้าน การสามารถออกแบบ หรือปรับแต่ง UI ให้สามารถทำหน้ารับข้อมูลบัตรเครดิตฝังอยู่ในหน้าเว็บของร้านค้าเองได้เลย (ไม่ต้องมีการ redirect ไป หน้ากรอกบัตรของ ChillPay)

1.1 System Flow ของการเชื่อมต่อ Credit Card Inline



1. ลูกค้า (End User) ทำรายการชำระเงิน พร้อมกรอกข้อมูลบัตรเครดิตที่หน้าของร้านค้า และกดชำระเงิน
2. ก่อนที่ข้อมูลการทำรายการของลูกค้า จะถูกส่งไปที่ Server ของร้านค้า, Script ของระบบ ChillPay ที่ฝังอยู่ที่หน้าเว็บของร้านค้า จะทำการยิง API เพื่อส่งข้อมูลบัตรไปเก็บไว้ที่ระบบ Credit Card Inline และสร้างเป็น Token (เรียกว่า PaymentCreditToken) เพื่อใช้ในกระบวนการการจ่ายเงิน พร้อมกับแนบ Token นี้ส่งไปหาร้านค้าพร้อมข้อมูลการทำรายการของลูกค้าด้วย
3. เมื่อ Server ของร้านค้าได้รับข้อมูลการทำรายการของลูกค้าพร้อม Token ก็จะใช้ข้อมูลนี้เพื่อเรียก API Payment มาที่ระบบ ChillPay ที่เป็น API สำหรับ Inline Payment โดยเฉพาะ (เรียกว่า ChillPay DirectCreditApi) โดยร้านค้าจะได้ Response กลับเป็น URL สำหรับใช้ในการ Redirect ไปยังหน้ากรอก OTP (ซึ่งเป็นหน้าของธนาคาร)
4. ระบบของร้านค้า Redirect ไปยังหน้ากรอก OTP
5. ลูกค้าทำการ Confirm การทำรายการชำระเงิน โดยการกรอก OTP และ Submit เพื่อ Process การจ่ายเงินต่อ
6. ธนาคาร Process การจ่ายเงิน
7. ธนาคารมีการส่งผลลัพธ์กลับมาทางฝั่ง ChillPay พร้อม Redirect กลับมาที่ระบบ ChillPay
8. ระบบ ChillPay ส่งผลลัพธ์กลับไปหาร้านค้า โดยมี 2 ทาง คือการ Redirect ไปยังหน้าแสดงผลการชำระเงินของร้าน และ การยิง Callback เพื่อส่งรายละเอียดผลการชำระเงินกลับไปหาร้านค้า (เป็น Background Process)
9. ร้านค้าแสดงผลการชำระเงินให้ลูกค้าเห็น
10. ร้านค้าได้รับรายละเอียดผลการชำระเงินจาก Callback ที่ระบบ ChillPay ส่งกลับไปหา

1.2 การเชื่อมต่อกับระบบ Credit Card Inline

การเชื่อมต่อกับระบบ Credit Card Inline จะแบ่งออกเป็น 2 ส่วนโดย ส่วนแรก คือการฝัง Script ของระบบ Credit Card Inline ลงไปที่หน้าเว็บของร้านค้าเพื่อให้สามารถรอกบัตรเครดิตได้ที่หน้าเว็บของร้านค้าเลย และส่วนที่สองคือ Server ของร้านค้าจะทำการยิง API เพื่อทำรายการชำระเงินกับระบบ ChillPay ผ่าน API เฉพาะสำหรับระบบ Credit Card Inline ซึ่งจะมีรายละเอียดดังนี้

1.2.1 การฝัง Script ของระบบ Credit Card Inline ลงไปที่หน้าเว็บของร้านค้า โดยนำเอา JavaScript ของฝั่ง ChillPay ไปใช้งาน โดยหน้าจ่ายเงินของร้านค้าต้องมีการแก้ไขดังนี้

1.2.1.1 เพิ่ม tag div ที่จำเป็นสำหรับเป็นช่องที่จะใช้แสดงส่วนที่เป็น Input ของบัตรเครดิต ซึ่งจะมีทั้งหมด 5 ตัว

```
<div id=""ccinline-card-name""></div>
<div id=""ccinline-card-number""></div>
<div id=""ccinline-card-expiry""></div>
<div id=""ccinline-card-cv"" ></div>
<div id=""ccinline-card-remember""></div>
<div id=""ccinline-card-select""></div>
```

- โดย tag เหล่านี้ไม่จำเป็นต้องอยู่ที่เดียวกัน แต่ส่วนสำคัญคือ จะต้องมีย div id "ccinline-card-name", "ccinline-card-number", "ccinline-card-expiry", "ccinline-card-cv" และ "ccinline-card-remember" ซึ่งเป็น div ที่จำเป็นต้องมี เพื่อให้ script ของ ระบบ Credit Card Inline ทำงานได้อย่างถูกต้อง
- ส่วน div id "ccinline-card-select" จะเป็น optional มีหรือไม่มีก็ได้ ขึ้นกับว่าต้องการใช้งานหรือไม่ โดย div นี้จะเป็นการใช้งาน UI ที่ให้ลูกค้า (End User) สามารถเลือกใช้บัตรที่เคยจำไว้ได้ [ดูรายละเอียดเพิ่มเติมที่หัวข้อ 1.2.4 การใช้งาน Card Select UI](#)
- ร้านค้าสามารถปรับแต่ง HTML Properties ของตัว div นี้หรือภายนอกได้ตามที่ต้องการ

1.2.1.2 เพิ่ม Tag Script เพื่อเรียกใช้งาน script ของระบบ Credit Card Inline โดย Script นี้ควรจะอยู่ใน Form ของการ Payment ของร้านค้าเอง เพื่อที่ว่าเมื่อลูกค้าทำการจ่ายเงิน Script จะสามารถสร้าง Token และแนบ Token ลงไปใน Form เพื่อส่งกลับไปหา Server ของร้านค้าได้

```
<script
  src=""https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
  data-merchant-code=""XXX-MERCHANTCODE-XXX"
  data-api-key=""XXXXXXXX-API-KEY-XXXXXXXX" >
</script>
```

- โดย Default แล้ว Script จะทำการแทรกการทำงานเมื่อมีการ Submit Form โดยจะทำการยิง API ไปที่ระบบ Credit Card ของ ChillPay เพื่อสร้าง PaymentCreditToken ซึ่งเป็น Token ที่เก็บข้อมูลบัตรและแนบ Token นี้เป็น Input เพิ่มใน Form เพื่อส่งไปยัง Server ของร้านค้า เพื่อให้ Server ร้านค้าสามารถยิง API เพื่อทำการจ่ายเงินต่อไปได้
- หาก Script ไม่อยู่ใน Form อาจทำให้ไม่สามารถใช้ระบบ Auto Create PaymentCreditToken นี้ได้ และต้อง Manage Event เอง [ดูรายละเอียดเพิ่มเติมที่ ข้อ 1.2.2 การจัดการ Event ของระบบ Credit Card Inline และ ข้อ 1.2.3 การสร้าง PaymentCreditToken แบบ Manual.](#)

ค่า src ของ script จะเปลี่ยนไปตาม environment ที่ต้องการเชื่อมต่อ โดยมีดังนี้

Sandbox : <https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js>

Production : <https://cdn.chill.credit/js/ccdpayment.js>

ตัวอย่าง หน้า HTML การ Payment ง่ายๆ (file name: payment.html)

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Payment Page</title>
</head>
<body>
<div id="CreditCardData">
    <br>
    <div style="font-size:30px;vertical-align:center;">Payment Page</div>
    <br>
    <div>
        <label>Cardholder Name</label>
        <div id="ccdinline-card-name"></div>
        <label>Card Number</label>
        <div id="ccdinline-card-number"></div>
        <label>Expiry Date</label>
        <div id="ccdinline-card-expiry"></div>
        <label>Security Code</label>
        <div id="ccdinline-card-cvv" ></div>
        <label>Remember Card</label>
        <div id="ccdinline-card-remember"></div>
    <br>
    <br>
    <form id="payment-form" method="POST" action="payment_post.php">
        <script
            src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
            data-merchant-code="XXX-MERCHANTCODE-XXX"
            data-api-key="XXXXXXXX-API-KEY-XXXXXXXX">
        </script>
        <label>Payment Details</label>
        <div><span> Order No : </span><input id="OrderNo" name="OrderNo"
type="text" value="my order" maxlength="20" ></div>
        <div><span> Customer Id : </span><input id="CustomerId" name="CustomerId"
type="text" value="customer id" maxlength="100" ></div>
        <div><span> Amount : </span><input id="Amount" name="Amount"
type="number" step="0.01" value="1500.00" ></div>
    </form>
    </div>
</div>

```

```

        <button>Submit</button>
    </form>
</div>
</div>
</body>
</html>
    
```

หาก script ทำงานอย่างถูกต้อง เมื่อมีการกด submit form ตัว script จะทำการแนบ input ไปกับ form ด้วยทั้งหมด 3 ตัว ดังนี้

- PaymentCreditToken : เป็น Token ตัวแทนข้อมูลบัตรเครดิตที่ลูกค้ากรอกเข้ามา จะถูกใช้ในการยิง API Payment ที่ร้านค้าจะยิงไปทำรายการที่ระบบ ChillPay ต่อไป
- RememberCard : มีค่าเป็น Y กับ N โดยเป็นค่าจากการที่ลูกค้ากดจำบัตรจาก UI ระบบ Credit Card Inline จะถูกใช้ใน การยิง API Payment เช่นกัน
- CreditToken : เป็นค่า Token การจำบัตรที่ใช้ในระบบ ChillPay ซึ่งค่านี้ระบบร้านค้าจะได้รับเมื่อลูกค้ามีการกดจำ บัตรเอาไว้ ร้านค้าสามารถเก็บค่า CreditToken นี้เอาไว้ เพื่อเป็นทางเลือกให้ลูกค้าสามารถเรียกใช้ กรณีลูกค้าต้องการ ใช้ข้อมูลบัตรที่ตัวเองเคยกดจำไว้

การเรียกใช้งาน Script ร้านค้าสามารถเช็ค Attribute ของ UI ที่ระบบ ChillPay สร้างขึ้นมาได้ในระดับหนึ่ง โดยจะมี Attribute ที่สามารถเช็คได้ ดังตาราง 1.1

ตาราง 1.1: Attribute ที่สามารถเช็คให้กับ Script ของระบบ Credit Card Inline

No.	Attribute Name	Mandatory /Optional	Data Type	Default	Description
1	src	M	string, url		เป็นค่า URL ของ script ของ ChillPay ที่ ต้องการใช้งาน (ค่าเปลี่ยนแปลงไปตาม Environment ที่เชื่อมต่อ)
2	data-merchant-code	M	string		MerchantCode ที่ระบบ ChillPay ออกให้ ทางร้านค้า
3	data-api-key	M	string		ApiKey ระบบ ChillPay ของทางร้านค้า
4	data-lang	O	string, lang	th	เช็คการแสดงผลภาษา ที่มีการแสดงใน UI ของ ระบบ Credit Card Inline
5	data-callback-event-receiver	O	string		ใส่ชื่อ function (JavaScript) ที่ต้องการรับ event จากระบบ Credit Card Inline *ดูรายละเอียดเพิ่มเติมที่ ข้อ 1.2.2 การจัดการ event ของระบบ Credit Card Inline
6	data-auto-adjust-height	O	string, boolean	true	เช็คให้ Script ของระบบ Credit Card Inline ทำการ Auto Adjust ค่า height ของ iframe ที่สร้างขึ้นมาให้มีขนาดเท่ากับความสูงจริงของ UI ภายใน iframe นั้นๆ พอดี (ทำครั้งเดียว ตอน Script สร้าง UI ขึ้นมา)
7	data-show-error-	O	string, boolean	true	เช็คให้ Script แสดง default error message

	message				ของแต่ละ input
8	data-auto-create-payment-credit-token-on-submit	O	string, boolean	true	<p>เช็คให้ Script ทำการแทรกการทำตอน Submit form โดยให้ทำการ Request ไปทาง ChillPay เพื่อสร้าง PaymentCreditToken และแนบ ข้อมูลลงไป ใน form อัตโนมัติแล้ว ค่อย Submit หากไม่ต้องการใช้ Feature นี้ สามารถ Manual จัดการการสร้าง PaymentCreditToken เองได้</p> <p>*ดูรายละเอียดเพิ่มเติมที่ ข้อ 1.2.3 การสร้าง PaymentCreditToken แบบ manual</p>
9	data-frame-bg-color	O	string, color		เช็ค Attribute ที่ iframe ที่ Script สร้าง ขึ้นมา (Apply กับทุก iframe)
10	data-frame-width	O	string, element size	100%	เช็ค Attribute ที่ iframe ที่ Script สร้าง ขึ้นมา (Apply กับทุก iframe)
11	data-frame-height	O	string, element size	150px	เช็ค Attribute ที่ iframe ที่ Script สร้าง ขึ้นมา (Apply กับทุก iframe)
12	data-body-bg-color	O	string, color		เช็ค Attribute ที่ body ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
13	data-body-border-style	O	string, border style		เช็ค Attribute ที่ body ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
14	data-body-border-width	O	string, dimension size		เช็ค Attribute ที่ body ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
15	data-body-border-color	O	string, color		เช็ค Attribute ที่ body ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
16	data-input-bg-color	O	string, color		เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
17	data-input-border-style	O	string, border style		เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
18	data-input-border-width	O	string, dimension size		เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
19	data-input-border-color	O	string, color		เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
20	data-input-padding	O	string, dimension size		เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
21	data-input-margin	O	string, dimension size		เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
22	data-input-font-family	O	string, font family		เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")

23	data-input-font-style	O	string, font style	เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
24	data-input-font-weight	O	string, font weight	เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
25	data-input-font-size	O	string, font size	เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
26	data-input-font-color	O	string, color	เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
27	data-input-text-align	O	string, text align	เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
28	data-input-width	O	string, element size	เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
29	data-input-height	O	string, element size	เช็ค Attribute ที่ input ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
30	data-error-bg-color	O	string, color	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
31	data-error-border-style	O	string, border style	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
32	data-error-border-width	O	string, dimension size	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
33	data-error-border-color	O	string, color	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
34	data-error-padding	O	string, dimension size	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
35	data-error-margin	O	string, dimension size	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
36	data-error-font-family	O	string, font family	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
37	data-error-font-style	O	string, font style	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
38	data-error-font-weight	O	string, font weight	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
39	data-error-font-size	O	string, font size	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
40	data-error-font-color	O	string, color	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")
41	data-error-text-align	O	string, text align	เช็ค Attribute ที่ div error ของ HTML ที่อยู่ใน iframe (ยกเว้น "card-remember")

42	data-card-name-frame-width	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-name" ที่ Script สร้างขึ้นมา
43	data-card-name-frame-height	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-name" ที่ Script สร้างขึ้นมา
44	data-card-name-input-width	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-name"
45	data-card-name-input-height	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-name"
46	data-card-number-frame-width	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-number" ที่ Script สร้างขึ้นมา
47	data-card-number-frame-height	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-number" ที่ Script สร้างขึ้นมา
48	data-card-number-input-width	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-number"
49	data-card-number-input-height	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-number"
50	data-card-expiry-frame-width	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-expiry" ที่ Script สร้างขึ้นมา
51	data-card-expiry-frame-height	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-expiry" ที่ Script สร้างขึ้นมา
52	data-card-expiry-input-width	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-expiry"
53	data-card-expiry-input-height	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-expiry"
54	data-card-cvv-frame-width	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-cvv" ที่ Script สร้างขึ้นมา
55	data-card-cvv-frame-height	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-cvv" ที่ Script สร้างขึ้นมา
56	data-card-cvv-input-width	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-cvv"
57	data-card-cvv-input-height	○	string, element size		เชื่อมต่อ Attribute ที่ input ของ HTML ที่อยู่ใน iframe "card-cvv"
58	data-card-remember-frame-width	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-remember" ที่ Script สร้างขึ้นมา
59	data-card-remember-frame-height	○	string, element size		เชื่อมต่อ Attribute ที่ iframe "card-remember" ที่ Script สร้างขึ้นมา
60	data-card-remember-checkbox-style	○	string, checkbox style	rect-1	เชื่อมต่อรูปแบบของ checkbox ที่จะแสดงของ "card-remember"

61	data-card-consent-font-family	O	string, font family		เช็ค Attribute ที่ div consent ของ HTML ที่อยู่ใน iframe "card-remember"
62	data-card-consent-font-style	O	string, font style		เช็ค Attribute ที่ div consent ของ HTML ที่อยู่ใน iframe "card-remember"
63	data-card-consent-font-weight	O	string, font weight		เช็ค Attribute ที่ div consent ของ HTML ที่อยู่ใน iframe "card-remember"
64	data-card-consent-font-size	O	string, font size		เช็ค Attribute ที่ div consent ของ HTML ที่อยู่ใน iframe "card-remember"
65	data-card-consent-font-color	O	string, color		เช็ค Attribute ที่ div consent ของ HTML ที่อยู่ใน iframe "card-remember"
66	data-card-consent-text-align	O	string, text align		เช็ค Attribute ที่ div consent ของ HTML ที่อยู่ใน iframe "card-remember"
67	data-card-select-frame-width	O	string, element size	100%	เช็ค attribute ที่ iframe "card-select" ที่ script สร้างขึ้นมา
68	data-card-select-frame-height	O	string, element size	150px	เช็ค attribute ที่ iframe "card-select" ที่ script สร้างขึ้นมา
69	data-card-select-show-card-type	O	string, boolean	true	เช็คการแสดงข้อมูลของแต่ละ box ข้อมูล (ข้อมูลบัตรที่ให้เลือกได้) ว่าให้แสดงข้อมูล Card Type หรือไม่
70	data-card-select-show-card-number	O	string, boolean	true	เช็คการแสดงข้อมูลของแต่ละ box ข้อมูล (ข้อมูลบัตรที่ให้เลือกได้) ว่าให้แสดงข้อมูล Card Number หรือไม่
71	data-card-select-show-card-name	O	string, boolean	false	เช็คการแสดงข้อมูลของแต่ละ box ข้อมูล (ข้อมูลบัตรที่ให้เลือกได้) ว่าให้แสดงข้อมูล Card Name หรือไม่
72	data-card-select-show-card-expiry	O	string, boolean	true	เช็คการแสดงข้อมูลของแต่ละ box ข้อมูล (ข้อมูลบัตรที่ให้เลือกได้) ว่าให้แสดงข้อมูล Card Expiry หรือไม่
73	data-card-select-container-display	O	string, display type		เช็ค attribute ที่ div container ของ html ที่อยู่ใน iframe "card-select"
74	data-card-select-container-flex-direction	O	string, flex direction		เช็ค attribute ที่ div container ของ html ที่อยู่ใน iframe "card-select"
75	data-card-select-container-flex-wrap	O	string, flex wrap		เช็ค attribute ที่ div container ของ html ที่อยู่ใน iframe "card-select"
76	data-card-select-container-padding	O	string, dimension size		เช็ค attribute ที่ div container ของ html ที่อยู่ใน iframe "card-select"
77	data-card-select-container-margin	O	string, dimension size		เช็ค attribute ที่ div container ของ html ที่อยู่ใน iframe "card-select"

78	data-card-select-container-width	O	string, element size		เชื่อมต่อ attribute ที่ div container ของ html ที่อยู่ใน iframe "card-select"
79	data-card-select-container-height	O	string, element size		เชื่อมต่อ attribute ที่ div container ของ html ที่อยู่ใน iframe "card-select"
80	data-card-select-box-bg-color	O	string, color		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
81	data-card-select-box-padding	O	string, dimension size	8px 16px	เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
82	data-card-select-box-margin	O	string, dimension size		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
83	data-card-select-box-width	O	string, element size		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
84	data-card-select-box-height	O	string, element size		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
85	data-card-select-box-font-family	O	string, font family		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
86	data-card-select-box-font-style	O	string, font style		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
87	data-card-select-box-font-weight	O	string, font weight		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
88	data-card-select-box-font-size	O	string, font size		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
89	data-card-select-box-font-color	O	string, color		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
90	data-card-select-box-text-align	O	string, text align		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select"
91	data-card-select-box-hover-bg-color	O	string, color	rgba(160, 160, 160, 0.3)	เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event hover
92	data-card-select-box-hover-font-style	O	string, font style		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event hover
93	data-card-select-box-hover-font-weight	O	string, font weight		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event hover
94	data-card-select-box-hover-font-size	O	string, font size		เชื่อมต่อ attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event hover

95	data-card-select-box-hover-font-color	○	string, color		เซ็ต attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event hover
96	data-card-select-box-selected-bg-color	○	string, color	rgba(102, 153, 255, 0.4)	เซ็ต attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event selected
97	data-card-select-box-selected-font-style	○	string, font style		เซ็ต attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event selected
98	data-card-select-box-selected-font-weight	○	string, font weight		เซ็ต attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event selected
99	data-card-select-box-selected-font-size	○	string, font size		เซ็ต attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event selected
100	data-card-select-box-selected-font-color	○	string, color	white	เซ็ต attribute ที่ div ข้อมูลบัตร(box) ของ html ที่อยู่ใน iframe "card-select" สำหรับ event selected

รายละเอียดของ Data Type มีดังนี้

- **string** : value ที่ใส่ลง script attribute จะอยู่ใน "" เสมอ
- **url** : value เป็น url
- **langcode** : ค่าที่ใช้ได้คือ en, th
- **boolean** : ค่าที่ใช้ได้คือ true, false
- **color** : ค่าที่ใช้ได้คือ
 - **13 basic colors** : ได้แก่ transparent, black, blue, brown, gray, green, orange, pink, purple, red, violet, white, yellow
 - **รหัสสีแบบ hex format** : ขึ้นต้นด้วย # และตามด้วยเลข hex 6 หลัก ตัวอย่าง "#0F0F0F"
 - **รหัสสีแบบ rgb format** : อยู่ใน format rgb(<N>, <N>, <N>) โดย <N> เป็นตัวเลข 0 - 255 ตัวอย่าง "rgb(255, 255, 255)"
 - **รหัสสีแบบ rgba format** : อยู่ใน format rgba(<N>, <N>, <N>, <D>) โดย <N> เป็นตัวเลข 0 - 255, <D> เป็นเลขทศนิยม ค่าตั้งแต่ 0 - 1 โดยมีทศนิยมได้สูงสุด 6 หลัก ตัวอย่าง "rgba(255, 100, 100, 0.5)"
 - **รหัสสีแบบ hsl format** : อยู่ใน format hsl(<H>, <P>%, <P>%) โดย <H> เป็นตัวเลข 0 - 359, <P> เป็นตัวเลข 0 - 100 ตัวอย่าง "hsl(9, 100%, 64%)"
 - **รหัสสีแบบ hsla format** : อยู่ใน format hsla(<H>, <P>%, <P>%, <D>) โดย <H> เป็นตัวเลข 0 - 359, <P> เป็นตัวเลข 0 - 100, <D> เป็นเลขทศนิยม ค่าตั้งแต่ 0 - 1 โดยมีทศนิยมได้สูงสุด 6 หลัก ตัวอย่าง "hsla(9, 100%, 64%, 0.5)"
- **element size** : สามารถใส่ค่าได้ 2 แบบ คือ เป็น pixel(px) หรือ percentage(%) โดย ตัวเลขของ px จะเป็นจำนวนเต็มเสมอ ส่วนตัวเลขของ % สามารถใส่เป็นทศนิยมได้ โดยมีทศนิยมได้สูงสุด 6 หลัก ตัวอย่าง "300px", "50%",

"60.02%"

- **border style** : ค่าที่ใช้ได้คือ dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden.
- **dimension size** : เป็นการใส่ค่าขนาดของกรอบ จะรับเฉพาะเป็น pixel(px) เท่านั้น โดยสามารถขีดได้ 4 แบบคือ
 - ขีดค่าเดียว จะเป็นการขีดขนาดของทั้ง 4 ด้าน (top, bottom, right, left) เป็นค่าเดียวกัน ตัวอย่าง "4px"
 - ขีด 2 ค่า โดยค่าแรกจะเป็นของ top และ bottom, ค่าที่สองเป็นของ right และ left ตัวอย่าง "2px 4px"
 - ขีด 3 ค่า โดยค่าแรกจะเป็นของ top, ค่าที่สองเป็นของ right และ left, ค่าที่สามเป็นของ bottom ตัวอย่าง "2px 4px 2px"
 - ขีด 4 ค่า โดยค่าแรกจะเป็นของ top, ค่าที่สองเป็นของ right, ค่าที่สามเป็นของ bottom, ค่าที่สี่เป็นของ left ตัวอย่าง "2px 4px 2px 4px"
- **font family** : ค่าที่ใช้ได้คือ "times new roman", "georgia", "garamond", "arial", "verdana", "courier new", "lucida console", "monaco", "brush script mt", "lucida handwriting", "copperplate", "papyrus".
- **font style** : ค่าที่ใช้ได้คือ normal, italic
- **font weight** : ค่าที่ใช้ได้คือ normal, bold
- **font size** : สามารถใส่ค่าได้ 2 แบบ คือ เป็น pixel(px) หรือ em โดยตัวเลขที่ใส่ของ em จะสามารถใส่ทศนิยมได้สูงสุด 6 หลัก ตัวอย่าง "22px", "1.12345em"
- **text align** : ค่าที่ใช้ได้คือ center, left, right
- **checkbox style** : ค่าที่ใช้ได้คือ rect-1, rect-2, rect-3, circle-1, circle-2, circle-3, slide-1, slide-2, slide-3, slide-4, slide-5
- **display type** : ค่าที่ใช้ได้คือ block, inline-block, flex, inline-flex
- **flex direction** : ค่าที่ใช้ได้คือ row, row-reverse, column, column-reverse
- **flex wrap** : ค่าที่ใช้ได้คือ nowrap, wrap, wrap-reverse

1.2.1.3 Server ของร้านค้าจะทำการยิง api เพื่อทำรายการชำระเงินกับระบบ ChillPay ผ่าน API เฉพาะสำหรับระบบ Credit Card Inline หลังจากที่ได้ข้อมูล "PaymentCreditToken" และข้อมูลอื่นๆ มาจาก Script ของระบบ Credit Card Inline แล้วทางร้านค้าสามารถใช้ข้อมูลนั้นเพื่อเรียก API Payment ของทาง ChillPay เพื่อทำการจ่ายเงิน โดยการเรียกมายัง URL ดังต่อไปนี้

Sandbox : <https://sandbox-api-directcredit.chillpay.co/api/v1/payment>

Production : <https://api-directcredit.chillpay.co/api/v1/payment>

การเรียก API Payment จะมี header ตามตาราง 1.2, parameter ตามรายละเอียดในตาราง 1.3 และจะมี response ตามรายละเอียดในตาราง 1.4 โดยการส่ง request parameter นั้นให้ส่งในรูปแบบ JSON format ส่วน response ที่จะได้รับกลับก็จะอยู่ในรูปแบบ JSON format เช่นกัน

การเรียก Api หาก Api ทำงานสำเร็จ จะ response กลับด้วย HTTP Code 200(OK) แต่หากเกิด error ขึ้นจะ response กลับด้วย HTTP Code 400(Bad Request)

ตาราง 1.2 : Header สำหรับการเรียก API Payment ระบบ Credit Card Inline ของ ChillPay

No.	Header Name	Data Type	Mandatory/Optional	Description
1	CHILLPAY-MerchantCode	String	M	MerchantCode ที่ระบบ ChillPay ออกให้ทางร้านค้า
2	CHILLPAY-ApiKey	String	M	ApiKey ระบบ ChillPay ของทางร้านค้า

ตาราง 1.3 : Request Parameter สำหรับการเรียก Api Payment ระบบ Credit Card Inline ของ ChillPay

No.	Request Parameter	Data Type	Length	Mandatory/ Optional	Description	Remark
1	OrderNo	String	20	M	เลขที่ Order หรือรหัสอ้างอิงของรายการ	เป็นค่าที่ออกโดยร้านค้า ต้องเป็นค่าเฉพาะที่ไม่ซ้ำกัน และต้องเป็นตัวเลขอย่างเดียว หรือตัวเลขผสมรวมกับตัวอักษรภาษาอังกฤษเท่านั้น ห้ามมีตัวอักษรอักขระพิเศษ เช่น +*-/#\$_ หรืออื่น ๆ
2	CustomerId	String	100	M	รหัสอ้างอิงของลูกค้า (End User) หรือชื่อลูกค้า	เช่น CUS000001หรือ นายแบงก์ มีมาก สามารถเว้นวรรคตรงกลางได้เท่านั้น ไม่รองรับการเว้นวรรคตำแหน่งแรกหรือตำแหน่งสุดท้าย และห้ามมีตัวอักษรอักขระพิเศษ เช่น +*-/#\$_ หรืออื่น ๆ
3	Amount	Number	12	M	จำนวนเงินที่ชำระของรายการ (2 หลักสุดท้ายจะเป็นค่าทศนิยม) *สกุลเงิน JPY และ KRW จะไม่มีหน่วยทศนิยม	ตัวอย่างเช่น ส่งค่าเข้ามาเป็น 55025 ระบบจะอ่านค่าเป็น 550.25 ส่วนสกุลเงิน JPY และ KRW ระบบจะอ่านค่าเป็น 55025.00
4	PhoneNumber	Number	10	O	เบอร์โทรศัพท์มือถือของลูกค้า (End User)	
5	Description	String	100	O	ข้อมูลรายละเอียดในการชำระเงิน	เช่น “สินค้าไอโฟน X 64GB ColorBlack”
6	ChannelCode	String	30	M	รหัสอ้างอิงแทนธนาคารที่รับชำระ	เช่น creditcard ปัจจุบันระบบ Credit Card Inline รองรับแค่ creditcard, installment_kbank เท่านั้น
7	Currency	String	3	M	ข้อมูลรหัสสกุลเงินให้กำหนดค่าเป็นเลขรหัส currency number	เช่น 764 ดูรายละเอียดในภาคผนวกง.
8	LangCode	String	2	O	รหัสภาษาที่จะแสดงบนเว็บไซต์ชำระเงินของทางธนาคาร	TH: ภาษาไทย EN: ภาษาอังกฤษ
9	RouteNo	Number	2	M	หมายเลขช่องทาง เป็นค่าที่จะออกให้โดยระบบ ChillPay	เช่น 1 หากร้านค้าต้องการใช้งานมากกว่า 1 หมายเลข สามารถติดต่อได้ที่ customersupport@chillpay.co

10	IPAddress	String	20	M	ข้อมูล IP Address ของลูกค้า	เช่น 203.255.255.155
11	TokenType	String	2	M	รูปแบบการใช้ token ที่จะใช้ในการชำระเงิน	DT: การชำระเงินแบบปกติ (ใช้ PaymentCreditToken) CT: การชำระเงินแบบใช้ token การจำบัตร (ใช้ PaymentCreditToken + CreditToken)
12	CreditToken	String	255	O	ข้อมูล token การจำบัตร	เป็น token ที่ออกให้โดยระบบ ChillPay เมื่อลูกค้ามีการเลือกจำบัตร หาก TokenType เป็น CT จะต้องมียกค่านี้ด้วย
13	DirectCreditToken	String	255	M	ข้อมูล token ที่เป็นตัวแทนข้อมูลบัตร	เป็น token ที่ออกให้โดยระบบ ChillPay เมื่อลูกค้าทำการจ่ายเงินผ่านช่องทาง Credit Card Inline
14	CreditMonth	Number	2	O	จำนวนเดือนที่ผ่อนชำระสำหรับการทำรายการผ่อนชำระผ่าน Credit Card	เช่น 3 , 8 , 10 หากเป็นการจ่ายช่องทาง installment จะต้องมีการระบุค่านี้นี้
15	ShopID	String	2	O	ค่า Shop ID สำหรับการทำรายการผ่อนชำระผ่าน Credit Card	01: สำหรับร้านค้าเป็นคนรับภาระดอกเบี้ย 02: สำหรับลูกค้า (End User) เป็นคนรับภาระดอกเบี้ย หากเป็นการจ่ายช่องทาง installment จะต้องมีการระบุค่านี้นี้
16	CustEmail	String	255	O	email ของลูกค้า(End User)	
17	SaveCard	String	1	O	ข้อมูลระบุว่าจะมีการจำบัตรจากการจ่ายเงินครั้งนี้หรือไม่	N: ไม่มีการจำบัตร Y: มีการจำบัตร กรณีไม่ระบุจะถือว่าเป็นค่า default คือค่า N: ไม่มีการจำบัตร
18	Checksum	String	32	M	ผลลัพธ์ที่ได้จากการนำค่าทั้งหมดไปเข้ารหัส MD5	ใช้เพื่อยืนยันความถูกต้องของรายการ

วิธีการคำนวณหา CheckSum

- นำค่าข้อมูลจาก Parameter จากข้อ 1-17 จากตาราง 1.3 มาเรียงต่อกันตามลำดับ(Parameter ที่ไม่มีการระบุค่าจะถือว่าเป็นค่าว่าง)
- เพิ่มค่าสุดท้ายด้วยค่า SecretKey ที่ได้จากทาง ChillPay (ค่า SecretKey เป็นค่าที่ห้ามเปิดเผยให้ที่ไม่มีส่วนเกี่ยวข้องกับเด็ดขาด)

3. ทำการ Hash ข้อมูลที่ได้ด้วยวิธีการ MD5 ดูรายละเอียดได้จาก <https://en.wikipedia.org/wiki/MD5> เมื่อ Hash แล้วจะได้ค่า MD5 Hash นำมาใช้เป็นค่า Parameter สำหรับตัวแปร CheckSum (ตามข้อ 18 ในตารางที่ 1.3)

ตัวอย่าง การคำนวณ CheckSum

ตัวอย่างข้อมูล Parameter ที่ใช้เรียก API Payment

```
{
  "OrderNo" : "Ord000001",
  "CustomerId" : "Cust_Test",
  "Amount" : 1500000,
  "PhoneNumber" : "0911111111",
  "Description" : "Desc Test",
  "ChannelCode" : "creditcard",
  "Currency" : "764",
  "LangCode" : "TH",
  "RouteNo" : 1,
  "IPAddress" : "127.0.0.1",
  "TokenType" : "DT",
  "CreditToken" : null,
  "DirectCreditToken" : "x1mPrAKM2kiZoQdrN7bwTbOBxvtvs3kw",
  "CreditMonth" : null,
  "ShopID" : null,
  "CustEmail" : "customer_01@a@b.com",
  "SaveCard" : "N"
}
```

ตัวอย่างข้อมูลอื่นๆ ที่จำเป็นของร้านค้า

```
Merchant SecretKey: "XXXXXXSECRETXXXXXX"
```

จากวิธีคำนวณข้อ 1, หาค่าข้อมูลที่มาต่อกัน

```
ConcatData = OrderNo + CustomerId + Amount + PhoneNumber + Description + ChannelCode + Currency + LangCode
+ RouteNo + IPAddress + TokenType + CreditToken + DirectCreditToken + CreditMonth + ShopID + CustEmail +
SaveCard
```

เนื่องจาก CreditToken, CreditMonth, ShopID มีค่าเป็น null(ไม่มีค่า) ดังนั้นแทนด้วยค่าว่าง จะได้ผลลัพธ์ดังนี้

```
ConcatData = "Ord000001Cust_Test15000000911111111Desc
Testcreditcard764TH1127.0.0.1DTx1mPrAKM2kiZoQdrN7bwTbOBxvtvs3kwcustomer_01@a@b.comN"
```

จากวิธีคำนวณข้อ 2, ต่อท้ายด้วย Merchant SecretKey จะได้ผลลัพธ์ดังนี้

```
ConcatData = "Ord000001Cust_Test15000000911111111Desc
Testcreditcard764TH1127.0.0.1DTx1mPrAKM2kiZoQdrN7bwTbOBxvtvs3kwcustomer_01@a@b.comNXXXXXXSECRETXXXXXX"
```

จากวิธีคำนวณข้อ 3, นำไปเข้า hash MD5 จะได้ผลลัพธ์ดังนี้

```
Checksum = "2b332382ff893b8c4c18960d87499e4b"
```

ตาราง 1.4 : Response Parameter ที่ส่งกลับจากการเรียก API Payment ระบบ Credit Card Inline ของ ChillPay

No.	Response Parameter	Data Type	Length	Mandatory/ Optional	Description	Remark
1	status	int		M	รหัสสถานะผลของการทำรายการ (การเรียก API)	ดูรายละเอียดในภาคผนวก ก.
2	message	string	255	M	คำอธิบายผลของการทำรายการ (การเรียก API)	
3	data	object		O	ข้อมูลผลของการทำรายการ (การเรียก API)	หากการทำรายการไม่สำเร็จ จะไม่มีข้อมูล
3.1	data.paymentStatus	string	30	M	สถานะการจ่ายเงิน	ดูรายละเอียดในภาคผนวก ค.
3.2	data.amount	long		M	จำนวนเงินที่ชำระของรายการ (2 หลัก สุดท้ายจะเป็นค่าทศนิยม) *สกุลเงิน JPY และ KRW จะไม่มีหน่วยทศนิยม	ตัวอย่างเช่น ส่งค่าเข้ามาเป็น 55025 ระบบจะอ่านค่าเป็น 550.25 ส่วนสกุลเงิน JPY และ KRW ระบบจะอ่านค่าเป็น 55025.00
3.3	data.orderNo	string	20	M	เลขที่ Order หรือรหัสอ้างอิงของรายการ	ค่าที่ได้รับมาจากร้านค้า
3.4	data.customerId	string	100	M	รหัสอ้างอิงของลูกค้า(End User) หรือชื่อลูกค้า	ค่าที่ได้รับมาจากร้านค้า
3.5	data.returnUrl	string	255	M	URL ที่จะแสดงเมื่อลูกค้าทำรายการสำเร็จ	เป็นหน้าเว็บไซต์ของทางร้านค้า (Result URL)
3.6	data.paymentUrl	string	255	O	URL ที่จะทำการ Redirect ไปหน้ายืนยันรหัส OTP ของธนาคาร	ทางร้านค้าจะต้องเขียนโปรแกรมให้ Redirect ไปที่ URL นี้
3.7	data.ipAddress	string	20	M	IP Address ของลูกค้าที่ทำรายการ	เช่น 20.25.55.20
3.8	data.token	string	60	M	รหัส Token สำหรับอ้างอิงรายการ	ใช้สำหรับ ChillPay
3.9	data.transactionId	long		M	เลขที่อ้างอิงรายการสำหรับร้านค้า	เพื่อใช้สำหรับตรวจสอบรายการอ้างอิงระหว่างร้านค้ากับ ChillPay เช่น 10071

3.10	data.channelCode	string	20	M	รหัสอ้างอิงแทนธนาคารที่รับชำระ	เช่น creditcard ปัจจุบันระบบ Credit Card Inline รองรับแค่ creditcard, installment_kbank เท่านั้น
3.11	data.createdDate	string	20	M	วันที่ทำรายการ	รูปแบบ: YYYYMMDDHHMMSS เช่น 20180712173122
3.12	data.expiredDate	string	20	M	วันที่รายการหมดอายุ	รูปแบบ: YYYYMMDDHHMMSS เช่น 20180712173122

ตัวอย่างการเรียก Api Payment (Code ภาษา PHP) ([file name: payment_post.php](#))

```
<?php

// Utilities Class/Function
class DirectPaymentPostData {
    public $OrderNo;
    public $CustomerId;
    public $Amount;
    public $PhoneNumber;
    public $Description;
    public $ChannelCode;
    public $Currency;
    public $LangCode;
    public $RouteNo;
    public $IPAddress;
    public $TokenType;
    public $CreditToken;
    public $DirectCreditToken;
    public $CreditMonth;
    public $ShopID;
    public $CustEmail;
    public $SaveCard;
    public $Checksum;

    public function getConcatString() {
        return $this->OrderNo . $this->CustomerId . $this->Amount . $this->PhoneNumber .
            $this->Description . $this->ChannelCode . $this->Currency . $this->LangCode .
```

```

        $this->RouteNo . $this->IPAddress . $this->TokenType . $this->CreditToken .
        $this->DirectCreditToken . $this->CreditMonth . $this->ShopID . $this->CustEmail .
        $this->SaveCard;
    }
    public function calculateChecksum($secretKey) {
        $this->Checksum = md5($this->getConcatString() . $secretKey);
    }
}
function IsNullOrEmptyString($str){
    return (!isset($str) || $str === null || trim($str) === "");
}

// CONFIG
$paymentUrl = "https://sandbox-api-directcredit.chillpay.co/api/v1/payment";
$merchantCode = "XXX-MERCHANTCODE-XXX";
$apiKey = "XXXXXXXX-API-KEY-XXXXXXXX";
$secretKey = "XXXXXXXX-SECRET-KEY-XXXXXXXX";

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    echo "Inline Payment Example: <br>";

    // from POST [Script Inline]
    $paymentCreditToken = isset($_POST['PaymentCreditToken']) ? $_POST['PaymentCreditToken'] : null;
    $creditToken = isset($_POST['CreditToken']) ? $_POST['CreditToken'] : null;
    $rememberCard = isset($_POST['RememberCard']) ? $_POST['RememberCard'] : null;

    echo "Script POST Data: <br>";
    echo " PaymentCreditToken = " . $paymentCreditToken . "<br>";
    echo " CreditToken = " . $creditToken . "<br>";
    echo " RememberCard = " . $rememberCard . "<br>";

    // from POST [Merchant's page]
    $orderNo = $_POST['OrderNo'];
    $customerId = $_POST['CustomerId'];
    $amount = intval(floatval($_POST['Amount']) * 100);

    echo "Merchant POST Data: <br>";
    echo " OrderNo = " . $orderNo . "<br>";

```

```
echo "" CustomerId = "" . $customerId . ""<br>";
echo "" Amount = "" . $amount . ""<br>";

// find tokenType from creditToken
$tokenType = IsNullOrEmptyString($creditToken) ? ""DT"" : ""CT"";

// post data
$postData = new DirectPaymentPostData();
$postData->OrderNo = $orderNo;
$postData->CustomerId = $customerId;
$postData->Amount = $amount;
$postData->PhoneNumber = "0911111111";
$postData->Description = "Description";
$postData->ChannelCode = "creditcard";
$postData->Currency = "764";
$postData->LangCode = "TH";
$postData->RouteNo = 1;
$postData->IPAddress = "127.0.0.1";
$postData->TokenType = $tokenType;
$postData->CreditToken = $creditToken;
$postData->DirectCreditToken = $paymentCreditToken;
$postData->CreditMonth = null;
$postData->ShopID = null;
$postData->CustEmail = "customer@a.com";
$postData->SaveCard = $rememberCard;

echo "ConcatData = " . $postData->getConcatString() . "<br><br>";

$postData->calculateChecksum($secretKey);
echo "Checksum = " . $postData->Checksum . "<br><br>";

$postDataJson = json_encode($postData);
echo "PostData = " . $postDataJson . "<br><br>";

// call direct payment api
$curl = curl_init();
curl_setopt_array($curl, array(
    CURLOPT_URL => $paymentUrl,
```

```

CURLOPT_RETURNTRANSFER => true,
CURLOPT_ENCODING => "",
CURLOPT_MAXREDIRS => 10,
CURLOPT_TIMEOUT => 30,
CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => "POST",
CURLOPT_POSTFIELDS => $postDataJson,
CURLOPT_HTTPHEADER => array(
    "Cache-Control: no-cache",
    "Content-Type: application/json",
    "CHILLPAY-MerchantCode: " . $merchantCode,
    "CHILLPAY-ApiKey: " . $apiKey
),
);
$response = curl_exec($curl);
$error = curl_error($curl);
curl_close($curl);

if ($error) {
    echo "cURL Error #:" . $error;
} else {
    echo "response #:" . $response . "<br><br>";

    $responseObject = json_decode($response, true);

    if ($responseObject['status'] != 200) {
        echo "API FAILED status: " . $responseObject['status'] . ", message: " . $responseObject['message'] . "<br><br>";
    } else {
        if ($responseObject['data']['paymentStatus'] == "Paid") {
            echo "PAID SUCCESS <br><br>";
        } else if ($responseObject['data']['paymentStatus'] == "WaitAuthorize") {
            $redirectUrl = $responseObject['data']['paymentUrl'];

            echo "wait 10 sec to redirect to url #:" . $redirectUrl . "<br><br>";
            header("refresh:10; url=" . $redirectUrl . "");
        } else {
            echo "PAYMENT STATUS ERROR paymentStatus: " . $responseObject['data']['paymentStatus'] . "<br><br>";
        }
    }
}

```

```

    }
  }
}
?>

```

ตัวอย่าง ข้อมูล Response(JSON) ที่ได้รับจากการเรียก API เคสทำรายการสำเร็จ

```

{
  "status" : 200,
  "message" : "Success",
  "data" : {
    "paymentStatus" : "WaitAuthorize",
    "amount" : 1500000,
    "orderNo" : "Ord000001",
    "customerId" : "Cust_Test",
    "returnUrl" : "https://merchant.com/result",
    "paymentUrl" : "https://bank.com/confirmOtp",
    "ipAddress" : "127.0.0.1",
    "token" : "xxxxxxx",
    "transactionId" : 123456,
    "channelCode" : "creditcard ",
    "createdDate" : "20240408133145",
    "expiredDate" : "20240409133145"
  }
}

```

ตัวอย่าง ข้อมูล Response(JSON) ที่ได้รับจากการเรียก API เคสทำรายการไม่สำเร็จ

```

{
  "status" : 3001,
  "message" : "Payment failed",
  "data" : null
}

```

1.2.2 การจัดการ Script event ของระบบ Credit Card Inline

ในระบบ script ของ Credit Card Inline ทางร้านค้าสามารถเขียน JavaScript เพิ่มเติม เพื่อจัดการ event ต่างๆ ของระบบ Inline Script ได้ โดยจะมีรายละเอียด event ทั้งหมด ดังตาราง 1.5

ตาราง 1.5 : Event ของ Credit Card Inline Script

No.	Event Id	Event SubId	Data Type	Data Structure	Remark
1	Initialized		string		เป็น event ที่จะเกิดขึ้นเมื่อการ Initialize Inline UI เสร็จสิ้น สามารถใช้ร่วมกับการเรียกใช้งาน ฟังก์ชันของระบบ Inline Script ได้ เนื่องจากฟังก์ชันของระบบ Inline Script จะถูกเรียกใช้งานได้ ก็ต่อเมื่อ UI ทำการ Initialize เสร็จสิ้นแล้วเท่านั้น
2	CreateTokenSucceed	value มีค่าเป็น "Success"	object	{ PaymentCreditToken: "XXXXXX", CreditToken: "YYYYYY", RememberCard: "N" }	เป็น event เมื่อการสร้าง PaymentCreditToken สำเร็จ มีรายละเอียดดังนี้ PaymentCreditToken: token ที่สร้างสำเร็จ CreditToken: credit token ที่เชื่อมต่อระบบ script Inline RememberCard: ค่าจาก checkbox มีค่าที่เป็นไปได้คือ "", "N", "Y" โดยค่าเริ่มต้นจะเป็น "" ซึ่งเทียบเท่ากับค่า "N" หากมีการกด checkbox ถ้า checkbox ถูกเลือกอยู่ค่าจะเป็น "Y" ถ้าไม่ถูกเลือกจะเป็น "N"
3	CreateTokenFailed	value มีค่าเป็นได้ดังนี้ 0: Success 1: Fail 2: Error 3: SystemError	object	{ Code: 3001, Message: "System Error", }	เป็น event เมื่อการสร้าง PaymentCreditToken ไม่สำเร็จ โดยเกิด error จากฝั่ง server Code: เป็น error code ที่ฝั่ง server จากการสร้าง PaymentCreditToken ไม่สำเร็จ Message: เป็น error message
4	CreateTokenError	value มีค่าเป็นได้ดังนี้ -1: Invalid Input -2: Invalid MerchantCode or MerchantApiKey ค่าอื่นๆ: เป็นค่า status ที่	string	จะเป็น message โดยค่าจะเปลี่ยนไปตาม Event SubId ได้แก่ "Invalid Input", "Invalid MerchantCode or MerchantApiKey" หรือ เป็นค่า statustext ที่ได้จาก	เป็น event เมื่อการสร้าง PaymentCreditToken ไม่สำเร็จ โดยเกิด error ที่ฝั่ง script เช่น validate parameter ไม่ผ่าน หรือเกิด error ขณะเรียก api เช่น เชื่อมต่อ api ไม่ได้

		ได้จาก xhr(XMLHttpRequest) เมื่อเกิด error		xhr(XMLHttpRequest) เมื่อ เกิด error	
5	InputState	value เป็นค่าที่บอกว่า เป็น input ไหน โดยมีค่า เป็นได้ดังนี้ "card-name": input ชื่อ เจ้าของบัตร "card-number": input หมายเลขบัตร "card-expiry": input วัน หมดอายุบัตร "card-cvv": input cvv	boolean	true	เป็น event เมื่อมีการ เปลี่ยนแปลงข้อมูลของ input ภายใน script โดย Data จะเป็น boolean ที่บอกว่า input ณ ขณะนั้น valid หรือไม่ <i>*event นี้สามารถนำไปใช้ร่วมกับ script attribute "data-show- error-message" ได้ โดยการเซต attribute เป็น false เพื่อปิด default error message ของ ระบบ script แล้วดัก event นี้ เพื่อให้ฝั่งร้านค้าสามารถแสดง error message ของตัวเองแทน ได้</i>
6	AddCardSelectSucceed	value มีค่าเป็น "Success"	object	{ CreditToken: "YYYYYY", CardType: "VISA" }	เป็น event เมื่อการ Add ข้อมูล การ์ด (จาก CreditToken) เข้าสู่ Card Select UI สำเร็จ มี รายละเอียดดังนี้ CreditToken: Token ที่ทำการ Add CardType: ชนิดของบัตร เช่น VISA, MASTERCARD, JCB, UnionPay
7	AddCardSelectFailed	Value มีค่าเป็นได้ดังนี้ 0: Success 1: Fail 2: Error 3: SystemError	object	{ Code: 3001, Message: "System Error", }	เป็น event เมื่อการ Add ข้อมูล การ์ด(จาก CreditToken) เข้าสู่ Card Select UI ไม่สำเร็จ โดย เกิด error จากฝั่ง server Code: เป็น Error code ที่ฝั่ง Server จากการหา CreditToken ไม่สำเร็จ Message: เป็น Error message
8	AddCardSelectError	Value มีค่าเป็นได้ดังนี้ -2: Invalid MerchantCode or	string	จะเป็น message โดยค่าจะ เปลี่ยนไปตาม Event SubId ได้แก่ "Invalid"	เป็น event เมื่อการadd ข้อมูล การ์ด(จาก CreditToken) เข้าสู่ Card Select UI ไม่สำเร็จ โดย

		MerchantApiKey ค่าอื่นๆ: เป็นค่า status ที่ได้จาก xhr(XMLHttpRequest) เมื่อเกิด error		MerchantCode or MerchantApiKey" หรือ เป็น ค่า statustext ที่ได้จาก xhr(XMLHttpRequest) เมื่อ เกิด error	เกิด error ที่ฝั่ง script เช่น validate parameter ไม่ผ่าน หรือเกิด error ขณะเรียก api เช่น เชื่อมต่อ api ไม่ได้
9	RemoveCardSelectSucceed	Value มีค่าเป็น "Success"	object	{ CreditToken: "YYYYYY" }	เป็น event เมื่อการ remove ข้อมูลการ์ด(จาก CreditToken) ออกจาก Card Select UI สำเร็จ มีรายละเอียดดังนี้ CreditToken: Token ที่ทำการ remove
10	UseCreditTokenSucceed	Value มีค่าเป็น "Success"	object	{ CreditToken: "YYYYYY", CardType: "VISA" }	เป็น event เมื่อการ use CreditToken (ระบุให้ Inline Script ใช้งาน CreditToken ที่ ต้องการในการจ่ายเงิน) สำเร็จ มี รายละเอียดดังนี้ CreditToken: Token ที่ทำการ use CardType: ชนิดของบัตร เช่น VISA, MASTERCARD, JCB, UnionPay
11	UseCreditTokenFailed	Value มีค่าเป็นได้ดังนี้ 0: Success 1: Fail 2: Error 3: SystemError	object	{ Code: 3001, Message: "System Error", }	เป็น event เมื่อการ use CreditToken (ระบุให้ Inline Script ใช้งาน CreditToken ที่ ต้องการในการจ่ายเงิน) ไม่สำเร็จ โดยเกิด Error จากฝั่ง Server Code: เป็น Error code ที่ฝั่ง Server จากการหา CreditToken ไม่สำเร็จ Message: เป็น Error message
12	UseCreditTokenError	Value มีค่าเป็นได้ดังนี้ -2: Invalid MerchantCode or MerchantApiKey ค่าอื่นๆ: เป็นค่า status ที่ได้จาก	string	จะเป็น Message โดยค่าจะ เปลี่ยนไปตาม Event SubId ได้แก่ "Invalid MerchantCode or MerchantApiKey" หรือ เป็น ค่า statustext ที่ได้จาก	เป็น event เมื่อการ use CreditToken (ระบุให้ Inline Script ใช้งาน CreditToken ที่ ต้องการในการจ่ายเงิน) ไม่สำเร็จ โดยเกิด Error ที่ฝั่ง Script เช่น Validate parameter ไม่ผ่าน

		xhr(XMLHttpRequest) เมื่อเกิด error		xhr(XMLHttpRequest) เมื่อเกิด Error	หรือเกิด Error ขณะเรียก API เช่น เชื่อมต่อ API ไม่ได้
13	UnuseCreditTokenSucceed	Value มีค่าเป็น "Success"	string		เป็น event เมื่อการ unuse CreditToken (ยกเลิกการเลือก CreditToken ที่เคยเลือกให้ Inline Script ใช้งาน) สำเร็จ
14	CheckCreditTokenSucceed	Value มีค่าเป็น "Success"	object	{ CreditToken: "YYYYYY", CardType: "VISA" }	เป็น event เมื่อมีการเรียก function check CreditToken (เช็คข้อมูล CreditToken ที่ฝั่ง server ว่า CreditToken อยู่ในสถานะที่ใช้งานได้หรือไม่) สำเร็จ มีรายละเอียดดังนี้ CreditToken: Token ที่ทำการ check CardType: ชนิดของบัตร เช่น VISA, MASTERCARD, JCB, UnionPay
15	CheckCreditTokenFailed	Value มีค่าเป็นได้ดังนี้ 0: Success 1: Fail 2: Error 3: SystemError	object	{ Code: 3001, Message: "System Error", }	เป็น event เมื่อมีการเรียก function check CreditToken (เช็คข้อมูล CreditToken ที่ฝั่ง server ว่า CreditToken อยู่ในสถานะที่ใช้งานได้หรือไม่) ไม่สำเร็จ โดยเกิด Error จากฝั่ง Server Code: เป็น Error code ที่ฝั่ง Server จากการหา CreditToken ไม่สำเร็จ Message: เป็น Error message
16	CheckCreditTokenError	Value มีค่าเป็นได้ดังนี้ -2: Invalid MerchantCode or MerchantApiKey ค่าอื่นๆ: เป็นค่า Status ที่ได้จาก xhr(XMLHttpRequest) เมื่อเกิด Error	string	จะเป็น Message โดยค่าจะเปลี่ยนไปตาม Event SubId ได้แก่ "Invalid MerchantCode or MerchantApiKey" หรือ เป็นค่า statustext ที่ได้จาก xhr(XMLHttpRequest) เมื่อเกิด Error	เป็น event เมื่อมีการเรียก function check CreditToken (เช็คข้อมูล CreditToken ที่ฝั่ง server ว่า CreditToken อยู่ในสถานะที่ใช้งานได้หรือไม่) ไม่สำเร็จ โดยเกิด Error ที่ฝั่ง Script เช่น Validate parameter ไม่ผ่าน หรือเกิด Error ขณะเรียก API เช่น เชื่อมต่อ API ไม่ได้

17	DeleteCreditTokenSucceed	Value มีค่าเป็น "Success"	object	{ CreditToken: "YYYYYY" }	เป็น event เมื่อมีการเรียก function delete CreditToken (ลบข้อมูล CreditToken ที่ฝั่ง server) สำเร็จ มีรายละเอียดดังนี้ CreditToken: Token ที่ทำการ delete
18	DeleteCreditTokenFailed	Value มีค่าเป็นได้ดังนี้ 0: Success 1: Fail 2: Error 3: SystemError	object	{ Code: 3001, Message: "System Error", }	เป็น event เมื่อมีการเรียก function delete CreditToken (ลบข้อมูล CreditToken ที่ฝั่ง server) ไม่สำเร็จ โดยเกิด error จากฝั่ง Server Code: เป็น Error code ที่ฝั่ง Server จากการหา CreditToken ไม่สำเร็จ Message: เป็น Error message
19	DeleteCreditTokenError	Value มีค่าเป็นได้ดังนี้ -2: Invalid MerchantCode or MerchantApiKey ค่าอื่นๆ: เป็นค่า status ที่ได้จาก xhr(XMLHttpRequest) เมื่อเกิด Error	string	จะเป็น Message โดยค่าจะ เปลี่ยนไปตาม Event SubId ได้แก่ "Invalid MerchantCode or MerchantApiKey" หรือ เป็น ค่า statutext ที่ได้จาก xhr(XMLHttpRequest) เมื่อ เกิด Error	เป็น event เมื่อมีการเรียก function delete CreditToken (ลบข้อมูล CreditToken ที่ฝั่ง server) ไม่สำเร็จ โดยเกิด Error ที่ฝั่ง Script เช่น Validate parameter ไม่ผ่าน หรือเกิด Error ขณะเรียก API เช่น เชื่อมต่อ API ไม่ได้

ขั้นตอนการแก้หน้า html เพื่อจัดการ script event เอง

- 1) เพิ่ม JavaScript Function ในหน้า HTML เพื่อเป็น Function ที่รับ Callback จากระบบ Script Inline

ตัวอย่าง Script function callback

```
<script>
function CCDCallbackEvent(eventId, subEventId, data) {
    if (eventId === "Initialized") {
        console.log("CCDCallbackEvent: Initialized:");
    }
    else if (eventId === "CreateTokenSucceed") {
        console.log("CCDCallbackEvent: CreateTokenSucceed: PaymentCreditToken=" +
data.PaymentCreditToken + ", CreditToken=" + data.CreditToken + ", RememberCard=" + data.RememberCard);
    }
    else if (eventId === "CreateTokenFailed") {
        console.log("CCDCallbackEvent: CreateTokenFailed: status=" + subEventId + ", code=" + data.Code + ",
```

```
message=" + data.Message);
    }
    else if (eventId === "CreateTokenError") {
        console.log("CCDCallbackEvent: CreateTokenError: status=" + subEventId + ", message=" + data);
    }
    else if (eventId === "InputState") {
        console.log("CCDCallbackEvent: InputState: input=" + subEventId + ", IsValid=" + data);
    }
    else if (eventId === "AddCardSelectSucceed") {
        console.log("CCDCallbackEvent: AddCardSelectSucceed: CreditToken=" + data.CreditToken + ",
CardType=" + data.CardType);
    }
    else if (eventId === "AddCardSelectFailed") {
        console.log("CCDCallbackEvent: AddCardSelectFailed: status=" + subEventId + ", code=" + data.Code +
", message=" + data.Message);
    }
    else if (eventId === "AddCardSelectError") {
        console.log("CCDCallbackEvent: AddCardSelectError: status=" + subEventId + ", message=" + data);
    }
    else if (eventId === "RemoveCardSelectSucceed") {
        console.log("CCDCallbackEvent: RemoveCardSelectSucceed: CreditToken=" + data.CreditToken);
    }
    else if (eventId === "UseCreditTokenSucceed") {
        console.log("CCDCallbackEvent: UseCreditTokenSucceed: CreditToken=" + data.CreditToken + ",
CardType=" + data.CardType);
    }
    else if (eventId === "UseCreditTokenFailed") {
        console.log("CCDCallbackEvent: UseCreditTokenFailed: status=" + subEventId + ", code=" + data.Code +
", message=" + data.Message);
    }
    else if (eventId === "UseCreditTokenError") {
        console.log("CCDCallbackEvent: UseCreditTokenError: status=" + subEventId + ", message=" + data);
    }
    else if (eventId === "UnuseCreditTokenSucceed") {
        console.log("CCDCallbackEvent: UnuseCreditTokenSucceed:");
    }
    else if (eventId === "CheckCreditTokenSucceed") {
        console.log("CCDCallbackEvent: CheckCreditTokenSucceed: CreditToken=" + data.CreditToken + ",
```

```

CardType=" + data.CardType);
    }
    else if (eventId === "CheckCreditTokenFailed") {
        console.log("CCDCallbackEvent: CheckCreditTokenFailed: status=" + subEventId + ", code=" + data.Code
+ ", message=" + data.Message);
    }
    else if (eventId === "CheckCreditTokenError") {
        console.log("CCDCallbackEvent: CheckCreditTokenError: status=" + subEventId + ", message=" + data);
    }
    else if (eventId === "DeleteCreditTokenSucceed") {
        console.log("CCDCallbackEvent: DeleteCreditTokenSucceed: CreditToken=" + data.CreditToken);
    }
    else if (eventId === "DeleteCreditTokenFailed") {
        console.log("CCDCallbackEvent: DeleteCreditTokenFailed: status=" + subEventId + ", code=" +
data.Code + ", message=" + data.Message);
    }
    else if (eventId === "DeleteCreditTokenError") {
        console.log("CCDCallbackEvent: DeleteCreditTokenError: status=" + subEventId + ", message=" + data);
    }
}
</script>

```

2) เพิ่ม Attribute เพื่อเชื่อมต่อ Callback Function ให้ Script

ตัวอย่าง การเรียก Script ที่มีการเพิ่ม Attribute เพื่อใช้งาน Callback Function

```

<script
src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
data-merchant-code="XXX-MERCHANTCODE-XXX"
data-api-key="XXXXXXXX-API-KEY-XXXXXXXX"
data-callback-event-receiver="CCDCallbackEvent" >
</script>

```

ตัวอย่าง การเรียก Script ที่มีการเพิ่ม Attribute เพื่อใช้งาน Callback Function

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Payment Page</title>

```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
    var cardNameValid = true;
    var cardNumberValid = true;
    var cardExpiryValid = true;
    var cardCvvValid = true;
    function RefreshErrorMessage() {
        if (!cardNameValid) {
            $('#my-custom-error-msg').html("Card name is invalid!!");
        }
        else if (!cardNumberValid) {
            $('#my-custom-error-msg').html("Card number is invalid!!");
        }
        else if (!cardExpiryValid) {
            $('#my-custom-error-msg').html("Card expiry is invalid!!");
        }
        else if (!cardCvvValid) {
            $('#my-custom-error-msg').html("Card cvv is invalid!!");
        }
        else {
            $('#my-custom-error-msg').html("");
        }
    }
    function CCDCallbackEvent(eventId, subEventId, data) {
        if (eventId === "InputState") {
            if (subEventId === "card-name") {
                cardNameValid = data;
                RefreshErrorMessage();
            }
            else if (subEventId === "card-number") {
                cardNumberValid = data;
                RefreshErrorMessage();
            }
            else if (subEventId === "card-expiry") {
                cardExpiryValid = data;
                RefreshErrorMessage();
            }
            else if (subEventId === "card-cvv") {
```

```

        cardCvvValid = data;
        RefreshErrorMessage();
    }
}
}
</script>
</head>
<body>
    <div id="CreditCardData">
        <br>
        <div style="font-size:30px;vertical-align:center;" >Payment Page</div>
        <br>
        <div>
            <label>Cardholder Name</label>
            <div id="ccdinline-card-name"></div>
            <label>Card Number</label>
            <div id="ccdinline-card-number"></div>
            <label>Expiry Date</label>
            <div id="ccdinline-card-expiry"></div>
            <label>Security Code</label>
            <div id="ccdinline-card-cvv" ></div>
            <label>Remember Card</label>
            <div id="ccdinline-card-remember"></div>
        <br>
        <div id="my-custom-error-msg" style="color:red; font-size: 22px; "></div>
        <br>
        <form id="payment-form" method="POST" action="payment_post.php">
            <script
                src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
                data-merchant-code="XXX-MERCHANTCODE-XXX"
                data-api-key="XXXXXXXX-API-KEY-XXXXXXXX"
                data-callback-event-receiver="CCDCallbackEvent"
                data-show-error-message="false" >
            </script>

            <label>Payment Details</label>
            <div><span> Order No : </span><input id="OrderNo" name="OrderNo" type="text" value="my
order" maxlength="20" ></div>

```

```

        <div><span> Customer Id : </span><input id="CustomerId" name="CustomerId" type="text"
value="customer id" maxlength="100" ></div>
        <div><span> Amount : </span><input id="Amount" name="Amount" type="number" step="0.01"
value="1500.00" ></div>
        <button>Submit</button>
        </form>
    </div>
</div>
</body>
</html>
    
```

1.2.3 การปรับการใช้งาน Script เพื่อสร้าง PaymentCreditToken แบบ manual

ในระบบ Script ของ Credit Card Inline โดยปกติจะมีการแทรกการทำงาน สร้าง PaymentCreditToken ตอนที่มีการ Submit Form และแนบข้อมูล Token นี้(พร้อม CreditToken และ RememberCard) ไปกับ Form ด้วย แต่ในบางกรณีหน้าเว็บของร้านค้าอาจจะไม่รองรับการทำงานส่วนนี้ เช่น การ POST Form การจ่ายเงินอยู่คนละหน้ากันกับหน้าที่กรอกบัตร หรือ ไม่มี Form (ตอน Submit Form ค่อยสร้าง Form ณ ตอนนั้น หรือเป็นการยิง AJAX หรือต้องการเปลี่ยน Input name เป็นชื่ออื่น) ซึ่งในกรณีนี้ทางร้านค้าจำเป็นต้องจัดการการสร้าง PaymentCreditToken เอง สามารถทำได้ดังนี้

- 1) เพิ่ม Attribute เพื่อเปิด Option การ Auto Create PaymentCreditToken on Submit ทิ้ง (Attribute "data-auto-create-payment-credit-token-on-submit") และเพิ่ม Callback Receiver เพื่อรับและจัดการ event เกี่ยวกับการสร้าง PaymentCreditToken

ตัวอย่าง การเรียก Script ที่มีการเพิ่ม Attribute เพื่อใช้งาน Callback function

```

<script
    src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
    data-merchant-code="XXX-MERCHANTCODE-XXX"
    data-api-key="XXXXXXXX-API-KEY-XXXXXXXX"
    data-callback-event-receiver="CCDCallbackEvent"
    data-auto-create-payment-credit-token-on-submit="false" >
</script>
    
```

- 2) เพิ่ม script การทำงานที่จะสร้าง PaymentCreditToken เอง อยู่ใน logic การทำงานของหน้าเว็บของร้านค้าเอง โดยให้เรียก function(Javascript) ดังนี้

```

ccdinline.CreatePaymentCreditToken();
    
```

- 3) เขียนโค้ด JavaScript เพื่อจัดการ event ผลลัพธ์การสร้าง PaymentCreditToken โดยมี event ที่ต้องจัดการทั้งหมด 3 ตัวคือ
 1. CreateTokenSucceed : เป็น event ที่จะเกิดเมื่อการสร้าง PaymentCreditToken สำเร็จ โดยจะได้รับข้อมูล 3 ตัวคือ PaymentCreditToken, CreditToken และ RememberCard ซึ่งข้อมูลทั้ง 3 ตัวเป็นข้อมูลจำเป็นที่จะเอาไปใช้ตอนยิง api Payment
 2. CreateTokenFailed : เป็น event ที่จะเกิดเมื่อการสร้าง PaymentCreditToken ไม่สำเร็จ โดยเกิด error ที่ฝั่ง server ของ ChillPay จะมีการส่งข้อมูล Error Code และ Error Message มาให้
 3. CreateTokenError : เป็น event ที่จะเกิดเมื่อการสร้าง PaymentCreditToken ไม่สำเร็จ โดยเกิด error ที่ฝั่ง script Inline

หรือ เกิด error ระหว่างการเชื่อมต่อกับ api ของระบบ Credit Card Inline จะมีการส่งข้อมูล Status และ Status Message มาให้

- 4) ตัวอย่าง หน้า HTML การ Payment แบบปิด auto-create-payment-credit-token-on-submit ของ Script Inline แล้วสร้าง PaymentCreditToken เอง

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Payment Page</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script>
    var isOnCreate = false;
    var currentPaymentCreditToken = null;
    var timeoutId;
    var isSubmit = false;

    function SetCreatPaymentCreditToken() {
      if (timeoutId) {
        clearTimeout(timeoutId);
      }

      timeoutId = setTimeout(() => {
        currentPaymentCreditToken = null;
      }, 120000);
    }

    $(document).ready(function() {
      $('#payment-btn').on("click", function() {
        if (isOnCreate) {
          return false;
        }
        if (currentPaymentCreditToken === null) {
          return false;
        }
        if (isSubmit) {
          return false;
        }
      }
    }
  )
</script>
```

```

        isSubmit = true;
        $('#payment-form').submit();
    });
    $('#crate-payment-credit-token-btn').on("click", function() {
        if (isOnCreate) {
            return false;
        }

        isOnCreate = true;
        ccdinline.CreatePaymentCreditToken();
    });
});
function CCDCallbackEvent(eventId, subEventId, data) {
    if (eventId === "CreateTokenSucceed") {
        $('#PaymentCreditToken').val(data.PaymentCreditToken);
        $('#CreditToken').val(data.CreditToken);
        $('#RememberCard').val(data.RememberCard);

        currentPaymentCreditToken = data.PaymentCreditToken;
        SetCreatPaymentCreditToken();

        isOnCreate = false;
    }
    else if (eventId === "CreateTokenFailed") {
        console.error("CCDCallbackEvent: CreateTokenFailed: status=" + subEventId + ",
code=" + data.Code + ", message=" + data.Message);
        isOnCreate = false;
    }
    else if (eventId === "CreateTokenError") {
        console.error("CCDCallbackEvent: CreateTokenError: status=" + subEventId + ",
message=" + data);
        isOnCreate = false;
    }
    }
}
</script>
</head>
<body>
    <div id="CreditCardData">

```

```

<br>
<div style="font-size:30px;vertical-align:center;">Payment Page</div>
<br>
<div>
  <label>Cardholder Name</label>
  <div id="ccdinline-card-name"></div>
  <label>Card Number</label>
  <div id="ccdinline-card-number"></div>
  <label>Expiry Date</label>
  <div id="ccdinline-card-expiry"></div>
  <label>Security Code</label>
  <div id="ccdinline-card-cvv" ></div>
  <label>Remember Card</label>
  <div id="ccdinline-card-remember"></div>
  <br>
  <br>
  <form id="payment-form" method="POST" action="payment_post.php">
    <script
      src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
      data-merchant-code="XXX-MERCHANTCODE-XXX"
      data-api-key="XXXXXXXX-API-KEY-XXXXXXXX"
      data-callback-event-receiver="CCDCallbackEvent"
      data-auto-create-payment-credit-token-on-submit="false" >
    </script>

    <label>Payment Details</label>
    <div><span> Order No : </span><input id="OrderNo" name="OrderNo"
type="text" value="my order" maxlength="20" ></div>
    <div><span> Customer Id : </span><input id="CustomerId" name="CustomerId"
type="text" value="customer id" maxlength="100" ></div>
    <div><span> Amount : </span><input id="Amount" name="Amount"
type="number" step="0.01" value="1500.00" ></div>
    <input id="PaymentCreditToken" name="PaymentCreditToken" type="hidden"
value="" >
    <input id="CreditToken" name="CreditToken" type="hidden" value="" >
    <input id="RememberCard" name="RememberCard" type="hidden" value="" >
    <button id="payment-btn">Submit</button>
  </form>

```

```

        <button id="crate-payment-credit-token-btn">Create PaymentCreditToken</button>
    </div>
</div>
</body>
</html>
    
```

*โดย default แล้ว PaymentCreditToken จะมีอายุ 120 seconds(2 mins) หากไม่ถูกใช้งานในช่วงเวลานี้ จะหมดอายุแล้วต้องทำการสร้างใหม่ ดังนั้นทางร้านค้าต้องคำนึงถึงจุดนี้ด้วยในการจัดการหรือสร้าง PaymentCreditToken เอง

1.2.4 การใช้งาน Card Select UI

การทำงานของระบบ Inline Script ที่เกี่ยวข้องกับการใช้งาน CreditToken จะมี 2 โหมดด้วยกันนั่นคือ การใช้งาน Card Select UI ซึ่งเป็น Built-in UI ของระบบ Inline Script เอง หรืออีกโหมดหนึ่งคือการที่ร้านค้าจัดการเรื่องการเลือก CreditToken ของลูกค้าเอง(อาจจะทำ UI ให้ลูกค้าเลือกเอง หรือร้านค้าบังคับเลือกให้เอง) วิธีนี้เพื่อให้ระบบ Inline Script สามารถรับรู้ได้ว่า CreditToken อันไหนถูกเลือกใช้ เว็บของร้านค้าจะต้องทำการเรียกฟังก์ชันของระบบ script เพื่อทำการเรียกใช้ CreditToken แบบ manual เอง(ดูรายละเอียดในหัวข้อ 2.4.6 การใช้งาน CreditToken แบบ manual) ส่วนในหัวข้อนี้จะเน้นโหมดแรก ที่เป็นการใช้ Card Select UI

การที่จะระบุว่าต้องการใช้ Card Select UI หรือไม่สามารถระบุได้จากการสร้าง div id "ccdinline-card-select" (รายละเอียดในหัวข้อ 1.2.1.1) โดยหากมี div นี้จะหมายความว่าเลือกโหมดใช้งาน Card Select UI แต่ถ้าไม่มี div นี้จะหมายความว่า เลือกโหมดการใช้ CreditToken แบบ manual

การใช้งานโหมดนี้จะมีฟังก์ชันให้ใช้งานดังนี้

1. ฟังก์ชันสำหรับการเพิ่มข้อมูลบัตรที่จะให้เลือกใน Card Select UI

```
ccdinline.AddCreditTokenToCardSelectUI(requestExpireDate, creditToken, securityCheck);
```

ในส่วนของ Parameter ที่จำเป็นในการเรียกฟังก์ชัน จะมีรายละเอียดดังนี้

- **requestExpireDate:** ระบุวันหมดอายุของ request นี้ (เป็นค่าที่ใช้เพื่อเพิ่ม security ในการเข้าถึงข้อมูล และจำกัดการใช้งานข้อมูล parameter ชุดนี้ให้อยู่ในช่วงเวลาหนึ่ง) โดยค่าจะเป็นค่าเวลาที่อยู่ในโซน UTC และอยู่ใน format "yyyyMMddHHmss"
ตัวอย่าง "20240408134550" จะเท่ากับวันที่ 8 เดือนเมษา 2024 เวลา 13:45:50 น. (เวลาเป็น UTC หากต้องการรู้เวลาไทย ต้อง + 7 ชม.) โดยหากเลยเวลาที่ระบุตามค่านี้ไปแล้ว Parameter ทั้งชุดนี้จะใช้งานไม่ได้อีก ต้องสร้างใหม่เท่านั้น(นั่นคือระบุ requestExpireDate ค่าใหม่ และค่านวน merchantSecurityCheck ใหม่)
- **creditToken:** เป็น Token ตัวแทนข้อมูลบัตรที่ลูกค้าเคยจ่ายแล้วส่งให้จำเอาไว้ ซึ่งร้านค้าจะได้ Token นี้เมื่อลูกค้าจ่ายเงินสำเร็จผ่านทาง Background Url ที่ร้านค้าแจ้งไว้กับระบบ ChillPay ตอนสมัครใช้บริการ ทางร้านค้าจะต้องเก็บค่านี้ไว้ และบริหารจัดการเอง โดยจะต้อง Mapping กับ ข้อมูลลูกค้าของร้านค้าเอง เพื่อให้สามารถเลือก CreditToken ของลูกค้ามาใช้ได้ถูกต้อง
- **merchantSecurityCheck:** เป็นค่า Hash MD5 ตาม format ที่กำหนด การหา MerchantSecurityCheck มีวิธีการดังนี้
 - 1.) นำข้อมูลมาต่อกันตามลำดับโดยมีข้อมูลดังนี้ MerchantCode + MerchantApiKey + RequestExpireDate + CreditToken + MerchantSecretKey

```
ConcatData = "XXX-MERCHANTCODE-XXXXXXXXXX-API-KEY-XXXXXXXX20240408134550MYCREDITTOKENXXXXXXXX-SECRET-KEY-XXXXXXXX"
```

- 2.) นำข้อมูลที่ได้จากข้อ 1.) มาหา Hash MD5 ก็จะได้เป็นค่า MerchantSecurityCheck

```
MerchantSecurityCheck = "573b38a80d803cd7559f4c347a08d1dc"
```

*การหา MerchantSecurityCheck จำเป็นต้องใช้ MerchantSecretKey ดังนั้นลอจิกส่วนนี้ควรจะอยู่ที่ฝั่ง Server ของร้านค้า ซึ่งทาง

ร้านค้าสามารถ Implement ตรงส่วนนี้ได้ 2 แบบ

แบบแรก ทำการคำนวณและหาค่าไว้ตั้งแต่ตอนที่ลูกค้าเข้าหน้า payment เพื่อให้มีข้อมูลพร้อมตั้งแต่แรก วิธีนี้อาจจะมีปัญหาเรื่องข้อมูล parameter หดอายุก่อนถูกเรียกใช้งานได้ ร้านค้าต้อง manage ส่วนนี้เอง

แบบที่สอง ทำ api ภายในของร้านค้าเอง เพื่อสร้าง MerchantSecurityCheck เมื่อต้องการเรียก function ของระบบ Inline Script ซึ่งวิธีนี้จะยืดหยุ่นกว่าแต่อาจจะยุ่งยากในการ implement และต้องคำนึงถึง security ด้วย เพื่อไม่ให้ผู้อื่นมาเรียกใช้งานได้(จำกัดให้เฉพาะเว็บร้านค้าเองเรียกใช้เท่านั้น)

เมื่อเรียกแล้วผลลัพธ์การเรียกข้อมูลจะถูกส่งเป็น event โดยจะมี eventId ที่ต้องจัดการ 3 ตัวดังนี้

- **AddCardSelectSucceed** : เป็น event ที่จะเกิดเมื่อการเพิ่มข้อมูลบัตรลงใน Card Select UI สำเร็จ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5

- **AddCardSelectFailed** : เป็น event ที่จะเกิดเมื่อการเพิ่มข้อมูลบัตรลงใน Card Select UI ไม่สำเร็จ โดยเกิด error ที่ฝั่ง server ของ ChillPay สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5

- **AddCardSelectError** : เป็น event ที่จะเกิดเมื่อการเพิ่มข้อมูลบัตรลงใน Card Select UI ไม่สำเร็จโดยเกิด error ที่ฝั่ง script Inline หรือเกิด error ระหว่างการเชื่อมต่อกับ api ของระบบ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5

2. ฟังก์ชันสำหรับการลบข้อมูลบัตรออกจาก Card Select UI

ในส่วนของ parameter ที่จำเป็นในการเรียกฟังก์ชัน จะมีรายละเอียดดังนี้

- **creditToken**: เป็น Token ตัวแทนข้อมูลบัตรที่ลูกค้าเคยจ่ายแล้วส่งให้จำเอาไว้ ซึ่งร้านค้าจะได้ Token นี้เมื่อลูกค้าจ่ายเงินสำเร็จผ่านทาง Background Url ที่ร้านค้าแจ้งไว้กับระบบ ChillPay ตอนสมัครใช้บริการ ทางร้านค้าจะต้องเก็บค่านี้ไว้ และบริหารจัดการเอง โดยจะต้อง Mapping กับ ข้อมูลลูกค้าของร้านค้าเอง เพื่อให้สามารถเลือก CreditToken ของลูกค้ามาใช้ได้ถูกต้อง

เมื่อเรียกแล้วผลลัพธ์การเรียกข้อมูลจะถูกส่งเป็น event โดยจะมี eventId ที่ต้องจัดการ 1 ตัวดังนี้

- **RemoveCardSelectSucceed**: เป็น event ที่จะเกิดเมื่อการลบข้อมูลบัตรออกจาก Card Select UI สำเร็จ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5

*การลบข้อมูลจะไม่มีการเกิด Error นั่นคือหาก creditToken ที่ส่งมา ไม่มีใน UI ก็จะไม่เกิดการลบแต่ผลลัพธ์ยังเป็น Success อยู่

*การเรียกใช้งานฟังก์ชันที่มีในระบบ Inline Script นี้จำเป็นต้องรอให้ระบบ Inline Script ทำการ Initialize เสร็จสิ้นแล้วเท่านั้น จึงจะสามารถเรียกใช้ฟังก์ชันได้

ตัวอย่าง หน้า HTML การ Payment แบบมีการใช้งาน Card Select UI

ตัวอย่างนี้เป็นการ ใช้งาน Card Select UI และมีการเพิ่มข้อมูลบัตรลงใน UI 3 บัตรโดยมีข้อมูล creditToken อยู่แล้ว

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Payment Page</title>
  <script>
    // sample data for add to Card Select UI
    var cardSelectList = [
      { RequestExpireDate: "20240515133045", CreditToken: "XXX-CreditToken-1-XXX",
MerchantSecurityCheck: "1234567890abcdef1234567890abcdef" },
```

```

        { RequestExpireDate: "20240515133045", CreditToken: "XXX-CreditToken-2-XXX",
MerchantSecurityCheck: "1234567890abcdef1234567890abcdef" },
        { RequestExpireDate: "20240515133045", CreditToken: "XXX-CreditToken-3-XXX",
MerchantSecurityCheck: "1234567890abcdef1234567890abcdef" }
    ];

    function CCDCallbackEvent(eventId, subEventId, data) {
        if (eventId === "Initialized") {
            console.log("CCDCallbackEvent: Initialized:");

            for (let i = 0; i < cardSelectList.length; i++) {

                ccdinline.AddCreditTokenToCardSelectUI(cardSelectList[i].RequestExpireDate,
cardSelectList[i].CreditToken, cardSelectList[i].MerchantSecurityCheck);
            }
        }
        else if (eventId === "AddCardSelectSucceed") {
            console.log("CCDCallbackEvent: AddCardSelectSucceed: CreditToken=" +
data.CreditToken + ", CardType=" + data.CardType);
        }
        else if (eventId === "AddCardSelectFailed") {
            console.log("CCDCallbackEvent: AddCardSelectFailed: status=" + subEventId + ",
code=" + data.Code + ", message=" + data.Message);
        }
        else if (eventId === "AddCardSelectError") {
            console.log("CCDCallbackEvent: AddCardSelectError: status=" + subEventId + ",
message=" + data);
        }
        else if (eventId === "RemoveCardSelectSucceed") {
            console.log("CCDCallbackEvent: RemoveCardSelectSucceed: CreditToken=" +
data.CreditToken);
        }
    }
}
</script>
</head>
<body>
    <div id="CreditCardData">
        <br>

```

```

<div style="font-size:30px;vertical-align:center;">Payment Page</div>
<br>
<div>
    <label>Cardholder Name</label>
    <div id="ccdinline-card-name"></div>
    <label>Card Number</label>
    <div id="ccdinline-card-number"></div>
    <label>Expiry Date</label>
    <div id="ccdinline-card-expiry"></div>
    <label>Security Code</label>
    <div id="ccdinline-card-cvv" ></div>
    <label>Remember Card</label>
    <div id="ccdinline-card-remember"></div>
    <label>Select Card</label>
    <div id="ccdinline-card-select"></div>
<br>
<br>
<form id="payment-form" method="POST" action="payment_post.php">
    <script
        src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
        data-merchant-code="XXX-MERCHANTCODE-XXX"
        data-api-key="XXXXXXXX-API-KEY-XXXXXXXX"
        data-callback-event-receiver="CCDCallbackEvent" >
    </script>

    <label>Payment Details</label>
    <div><span> Order No : </span><input id="OrderNo" name="OrderNo"
type="text" value="my order" maxlength="20" ></div>
    <div><span> Customer Id : </span><input id="CustomerId" name="CustomerId"
type="text" value="customer id" maxlength="100" ></div>
    <div><span> Amount : </span><input id="Amount" name="Amount"
type="number" step="0.01" value="1500.00" ></div>
    <button>Submit</button>
</form>
</div>
</div>
</body>
</html>

```

1.2.5 การใช้ CreditToken แบบ Manual

หากทางร้านค้าต้องการจัดการเกี่ยวกับการเลือก CreditToken ให้ลูกค้าเอง (ทำ UI การเลือกเอง หรือต้องการระบุการใช้งาน CreditToken ให้ลูกค้าเองโดยตรง) สามารถทำได้ โดยจะต้องไม่ใช้งาน Card Select UI ของระบบ Inline Script (ไม่มีการสร้าง div id "ccdinline-card-select") จึงจะสามารถใช้งานฟังก์ชันที่บังคับให้ระบบ Inline Script ใช้งาน CreditToken เองนี้ การใช้งาน CreditToken แบบ manual จะมีฟังก์ชันให้ใช้งานดังนี้

1. ฟังก์ชันสำหรับการบังคับให้ระบบ Inline Script ใช้งาน CreditToken ที่ต้องการ

```
ccdinline.UseCreditToken(requestExpireDate, creditToken, securityCheck);
```

โดยรายละเอียดเกี่ยวกับ parameter จะเหมือนกับฟังก์ชัน "ccdinline.AddCreditTokenToCardSelectUI" ที่อยู่ในหัวข้อ 1.2.4 เมื่อเรียกแล้วผลลัพธ์การเรียกข้อมูลจะถูกส่งเป็น event โดยจะมี eventId ที่ต้องจัดการ 3 ตัวดังนี้

- **UseCreditTokenSucceed** : เป็น event ที่จะเกิดเมื่อบังคับให้ระบบ Inline Script ใช้งาน CreditToken ที่ต้องการ สำเร็จ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5
- **UseCreditTokenFailed** : เป็น event ที่จะเกิดเมื่อบังคับให้ระบบ Inline Script ใช้งาน CreditToken ที่ต้องการ ไม่สำเร็จ โดยเกิด error ที่ฝั่ง server ของ ChillPay สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5
- **UseCreditTokenError** : เป็น event ที่จะเกิดเมื่อบังคับให้ระบบ Inline Script ใช้งาน CreditToken ที่ต้องการ ไม่สำเร็จ โดยเกิด error ที่ฝั่ง script Inline หรือ เกิด error ระหว่างการเชื่อมต่อกับ api ของระบบ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5

2. ฟังก์ชันสำหรับการบังคับให้ระบบ Inline Script ยกเลิกการใช้งาน CreditToken ที่ใช้อยู่ ณ ขณะนั้น

```
ccdinline.UnuseCreditToken();
```

เมื่อเรียกแล้วผลลัพธ์การเรียกข้อมูลจะถูกส่งเป็น event โดยจะมี eventId ที่ต้องจัดการ 1 ตัวดังนี้

- **UnuseCreditTokenSucceed**: เป็น event ที่จะเกิดเมื่อบังคับให้ระบบ Inline Script ยกเลิกการใช้งาน CreditToken ที่ใช้อยู่ ณ ขณะนั้น สำเร็จ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5

*การยกเลิกการใช้งาน CreditToken จะไม่มีการเกิด error นั่นคือแม้ว่าตอนนั้นระบบ Inline Script จะไม่ได้มีการเลือกใช้ CreditToken ตัวไหนอยู่ ก็สามารถเรียกคำสั่งนี้ได้โดยไม่เกิด error และผลลัพธ์ก็จะเป็น Success อยู่

การเรียกใช้งานฟังก์ชันที่มีในระบบ Inline Script นี้จำเป็นต้องรอให้ระบบ Inline Script ทำการ Initialize เสร็จสิ้นแล้วเท่านั้น จึงจะสามารถเรียกใช้ฟังก์ชันได้

ตัวอย่าง หน้า HTML การ Payment แบบบังคับใช้งาน CreditToken ที่เลือกไว้แล้ว

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Payment Page</title>
  <script>
    // sample data for UseCreditToken
    var useCreditToken = { RequestExpireDate: "20240515133045", CreditToken: "XXX-CreditToken-1-XXX",
MerchantSecurityCheck: "1234567890abcdef1234567890abcdef" };

    function CCDCallbackEvent(eventId, subEventId, data) {
```

```

        if (eventId === "Initialized") {
            console.log("CCDCallbackEvent: Initialized:");
            ccdinline.UseCreditToken(useCreditToken.RequestExpireDate, useCreditToken.CreditToken,
useCreditToken.MerchantSecurityCheck);
        }
        else if (eventId === "UseCreditTokenSucceed") {
            console.log("CCDCallbackEvent: UseCreditTokenSucceed: CreditToken=" +
data.CreditToken + ", CardType=" + data.CardType);
        }
        else if (eventId === "UseCreditTokenFailed") {
            console.log("CCDCallbackEvent: UseCreditTokenFailed: status=" + subEventId + ", code=" +
data.Code + ", message=" + data.Message);
        }
        else if (eventId === "UseCreditTokenError") {
            console.log("CCDCallbackEvent: UseCreditTokenError: status=" + subEventId + ", message="
+ data);
        }
        else if (eventId === "UnuseCreditTokenSucceed") {
            console.log("CCDCallbackEvent: UnuseCreditTokenSucceed:");
        }
    }
</script>
</head>

<body>
    <div id="CreditCardData">
        <br>
        <div style="font-size:30px;vertical-align:center;">Payment Page</div>
        <br>
        <div>
            <label>Cardholder Name</label>
            <div id="ccdinline-card-name"></div>
            <label>Card Number</label>
            <div id="ccdinline-card-number"></div>
            <label>Expiry Date</label>
            <div id="ccdinline-card-expiry"></div>
            <label>Security Code</label>
            <div id="ccdinline-card-cvv" ></div>
        </div>
    </div>

```

```

<label>Remember Card</label>
<div id="ccdinline-card-remember"></div>
<br>
<br>
<form id="payment-form" method="POST" action="payment_post.php">
  <script
    src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
    data-merchant-code="XXX-MERCHANTCODE-XXX"
    data-api-key="XXXXXXXX-API-KEY-XXXXXXXX"
    data-callback-event-receiver="CCDCallbackEvent" >
  </script>

  <label>Payment Details</label>
  <div><span> Order No : </span><input id="OrderNo" name="OrderNo" type="text"
value="my order" maxlength="20" ></div>
  <div><span> Customer Id : </span><input id="CustomerId" name="CustomerId"
type="text" value="customer id" maxlength="100" ></div>
  <div><span> Amount : </span><input id="Amount" name="Amount" type="number"
step="0.01" value="1500.00" ></div>
  <button>Submit</button>
</form>
</div>
</div>
</body>
</html>

```

1.2.6 การใช้งาน function CheckCreditToken และ DeleteCreditToken

ฟังก์ชัน 2 ตัวนี้ เป็น utility function ที่ระบบ Inline Script จัดเตรียมไว้เพื่อให้ร้านค้าใช้งานในการจัดการ CreditToken ได้ โดยจะมี

ดังนี้

1. ฟังก์ชันสำหรับการเช็ค CreditToken ว่ายังพร้อมใช้งานอยู่หรือไม่

```
ccdinline.CheckCreditToken(requestExpireDate, creditToken, securityCheck);
```

โดยรายละเอียดเกี่ยวกับ parameter จะเหมือนกับฟังก์ชัน "ccdinline.AddCreditTokenToCardSelectUI" ที่อยู่ในหัวข้อ 1.2.4 เมื่อเรียกแล้วผลลัพธ์การเรียกข้อมูลจะถูกส่งเป็น event โดยจะมี eventId ที่ต้องจัดการ 3 ตัวดังนี้

- **CheckCreditTokenSucceed** : เป็น event ที่จะเกิดเมื่อฝั่ง ChillPay เช็ค CreditToken แล้ว token อยู่ในสถานะพร้อมใช้งาน สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5
- **CheckCreditTokenFailed** : เป็น event ที่จะเกิดเมื่อฝั่ง ChillPay เช็ค CreditToken แล้ว ไม่พบ token หรือ token อยู่ในสถานะที่ไม่พร้อมใช้งาน สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5
- **CheckCreditTokenError** : เป็น event ที่จะเกิดเมื่อทำการเช็ค CreditToken ที่ต้องการ ไม่สำเร็จ โดยเกิด error ที่ฝั่ง script Inline หรือ เกิด error ระหว่างการเชื่อมต่อกับ API ของระบบ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้

จากตาราง 1.5

2. ฟังก์ชันสำหรับการลบ CreditToken ออกจากระบบ

```
ccdinline.DeleteCreditToken(requestExpireDate, creditToken, securityCheck);
```

โดยรายละเอียดเกี่ยวกับ parameter จะเหมือนกับฟังก์ชัน "ccdinline.AddCreditTokenToCardSelectUI" ที่อยู่ในหัวข้อ 1.2.4 เมื่อเรียกแล้วผลลัพธ์การเรียกข้อมูลจะถูกส่งเป็น event โดยจะมี eventId ที่ต้องจัดการ 3 ตัวดังนี้

- **DeleteCreditTokenSucceed** : เป็น event ที่จะเกิดเมื่อฝั่ง ChillPay ทำการลบ CreditToken ออกจากระบบ สำเร็จ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5
- **DeleteCreditTokenFailed** : เป็น event ที่จะเกิดเมื่อฝั่ง ChillPay ทำการลบ CreditToken ออกจากระบบ ไม่สำเร็จ โดยเกิด error ที่ฝั่ง server ของ ChillPay สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5
- **DeleteCreditTokenError** : เป็น event ที่จะเกิดเมื่อฝั่ง ChillPay ทำการลบ CreditToken ออกจากระบบ ไม่สำเร็จ โดยเกิด error ที่ฝั่ง script Inline หรือ เกิด error ระหว่างการเชื่อมต่อกับ api ของระบบ สามารถดูรายละเอียดข้อมูลที่จะได้รับจาก event นี้ได้จากตาราง 1.5

การเรียกใช้งานฟังก์ชันที่มีในระบบ Inline Script นี้จำเป็นต้องรอให้ระบบ Inline Script ทำการ Initialize เสร็จสิ้นแล้วเท่านั้น จึงจะสามารถเรียกใช้ฟังก์ชันได้

ตัวอย่าง หน้า html การ payment ที่มีการเช็ค CreditToken ที่เลือกไว้แล้ว

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Payment Page</title>
  <script>
    // sample data for CheckCreditToken
    var checkCreditToken = { RequestExpireDate: "20240515133045", CreditToken: "XXX-CreditToken-1-XXX", MerchantSecurityCheck: "1234567890abcdef1234567890abcdef" };

    function CCDCallbackEvent(eventId, subEventId, data) {
      if (eventId === "Initialized") {
        console.log("CCDCallbackEvent: Initialized");

        ccdinline.CheckCreditToken(checkCreditToken.RequestExpireDate,
checkCreditToken.CreditToken, checkCreditToken.MerchantSecurityCheck);
      }
      else if (eventId === "CheckCreditTokenSucceed") {
        console.log("CCDCallbackEvent: CheckCreditTokenSucceed: CreditToken=" +
data.CreditToken + ", CardType=" + data.CardType);
      }
      else if (eventId === "CheckCreditTokenFailed") {
```

```

        console.log("CCDCallbackEvent: CheckCreditTokenFailed: status=" + subEventId
+ ", code=" + data.Code + ", message=" + data.Message);
    }
    else if (eventId === "CheckCreditTokenError") {
        console.log("CCDCallbackEvent: CheckCreditTokenError: status=" + subEventId +
", message=" + data);
    }
    else if (eventId === "DeleteCreditTokenSucceed") {
        console.log("CCDCallbackEvent: DeleteCreditTokenSucceed: CreditToken=" +
data.CreditToken);
    }
    else if (eventId === "DeleteCreditTokenFailed") {
        console.log("CCDCallbackEvent: DeleteCreditTokenFailed: status=" + subEventId
+ ", code=" + data.Code + ", message=" + data.Message);
    }
    else if (eventId === "DeleteCreditTokenError") {
        console.log("CCDCallbackEvent: DeleteCreditTokenError: status=" + subEventId
+ ", message=" + data);
    }
}
</script>
</head>
<body>
    <div id="CreditCardData">
        <br>
        <div style=""font-size:30px;vertical-align:center;">Payment Page</div>
        <br>
        <div>
            <label>Cardholder Name</label>
            <div id="ccdinline-card-name"></div>
            <label>Card Number</label>
            <div id="ccdinline-card-number"></div>
            <label>Expiry Date</label>
            <div id="ccdinline-card-expiry"></div>
            <label>Security Code</label>
            <div id="ccdinline-card-cvv" ></div>
            <label>Remember Card</label>
        </div>
    </div>

```

```

<div id="ccdinline-card-remember"></div>
<br>
<br>
<form id="payment-form" method="POST" action="payment_post.php">
  <script
    src="https://sandbox-bankdemo3.chillpay.co/js/ccdpayment.js"
    data-merchant-code="XXX-MERCHANTCODE-XXX"
    data-api-key="XXXXXXXX-API-KEY-XXXXXXXX"
    data-callback-event-receiver="CCDCallbackEvent" >
  </script>

  <label>Payment Details</label>
  <div><span> Order No : </span><input id="OrderNo" name="OrderNo"
type="text" value="my order" maxlength="20" ></div>
  <div><span> Customer Id : </span><input id="CustomerId" name="CustomerId"
type="text" value="customer id" maxlength="100" ></div>
  <div><span> Amount : </span><input id="Amount" name="Amount"
type="number" step="0.01" value="1500.00" ></div>
  <button>Submit</button>
</form>
</div>
</div>
</body>
</html>

```

2. การเชื่อมต่อบริการ ChillCredit Public Api

2.1 API สำหรับ CreatePaymentCreditToken

การเรียก API CreatePaymentCreditToken จะมี header ตามตาราง 2.1, parameter ตามรายละเอียดในตาราง 2.2 และจะมี response ตามรายละเอียดในตาราง 2.3 โดยการส่ง request parameter นั้นให้ส่งในรูปแบบ JSON format ส่วน response ที่จะได้รับกลับก็จะอยู่ในรูปแบบ JSON format เช่นกัน

- URL สำหรับ POST ข้อมูลบน Sandbox : <https://sandbox-bankdemo3.chillpay.co/ChillCredit/public/CreatePaymentCreditToken>
- URL สำหรับ POST ข้อมูลบน Production : <https://api.chill.credit/api/v1/public/PaymentCreditToken/Create>

ตาราง 2.1 : Header สำหรับการเรียก API Payment ระบบ Credit Card Inline ของ ChillPay

No.	Header Name	Data Type	Mandatory/Optional	Default
1	Content-Type	String	M	application/json

ตาราง 2.2 : Request Parameter สำหรับการเรียก Api CreatePaymentCreditToken

No.	Request Parameter	Data Type	Length	Mandatory /Optional	Description	Remark
1	Portal	String	50	M	เป็นค่าที่จะออกให้โดย ChillPay	เช่น ChillPay
2	Merchant	String	50	M	เป็นค่าที่จะออกให้โดย ChillPay	เช่น M000001
3	MerchantAPIKey	String	1000	M	รหัส API Key ที่ใช้ในการเชื่อมต่อระบบ	เป็นค่าที่จะออกให้โดย ChillPay
4	CardNumber	String	30	M	หมายเลขบัตรเครดิต	
5	ExpireMonth	String	2	M	ข้อมูลเดือนหมดอายุของบัตรเครดิต	เช่น 01
6	ExpireYear	String	4	M	ข้อมูลปีหมดอายุของบัตรเครดิต	เช่น 2026
7	NameOnCard	String	255	M	ชื่อผู้ถือบัตรเป็นภาษาอังกฤษ	
8	CWV	String	3	M	รหัสหลังบัตรเครดิต	เช่น 123
9	PortalCreditToken	String	500	M	ข้อมูล token การจำบัตร	เป็น token ที่ออกให้โดยระบบ ChillPay เมื่อลูกค้ามีการเลือกจำบัตร หากต้องการใช้บัตรที่เคยจำไว้จะต้องมีค่านี้อยู่ด้วย
10	MerchantCheckSum	String	32	M	ผลลัพธ์ที่ได้จากการนำค่าทั้งหมดไปเข้ารหัส MD5	ใช้เพื่อยืนยันความถูกต้องของรายการ

วิธีการคำนวณหา CheckSum

- นำค่าข้อมูลจาก Parameter จากข้อ 1-10 จากตาราง 2.2 มาเรียงต่อกันตามลำดับ (Parameter ที่ไม่มีการระบุค่าจะถือว่าเป็นค่าว่าง)
- เพิ่มค่าสุดท้ายด้วยค่า SecretKey ที่ได้จากทาง ChillPay (ค่า SecretKey เป็นค่าที่ห้ามเปิดเผยให้ที่ไม่มีส่วนเกี่ยวข้องเด็ดขาด)
- ทำการ Hash ข้อมูลที่ได้ด้วยวิธีการ MD5 ดูรายละเอียดได้จาก <https://en.wikipedia.org/wiki/MD5> เมื่อ Hash แล้วจะได้ค่า MD5 Hash นำมาใช้เป็นค่า Parameter สำหรับตัวแปร CheckSum (ตามข้อ 10 ในตารางที่ 1.3)

ตัวอย่าง การคำนวณ CheckSum

```
{
  "Portal": "ChillPay",
  "Merchant": "M000001",
  "MerchantAPIKey": "xxx",
  "CardNumber": "123456789",
  "ExpireMonth": "01",
  "ExpireYear": "2026",
  "NameOnCard": "testname",
  "CVV": "123",
  "PortalCreditToken": "123456789"
}
```

ตัวอย่างข้อมูลอื่นๆ ที่จำเป็นของร้านค้า

Merchant SecretKey: "yyy"

หาค่าข้อมูลที่มาต่อกัน

ConcatData = Portal + Merchant + MerchantAPIKey + CardNumber + ExpireMonth + ExpireYear + NameOnCard + CVV + PortalCreditToken + Merchant SecretKey

จะได้ผลลัพธ์ดังนี้

ConcatData = " ChillPayM000001xxx123456789012026testname123123456789yyy"

นำไปเข้า hash MD5 จะได้ผลลัพธ์ดังนี้

MerchantCheckSum = "2b332382ff893b8c4c18960d87499e4b"

ตาราง 2.3 : Response Parameter ที่ส่งกลับจากการเรียก API CreatePaymentCreditToken

No.	Response Parameter	Data Type	Length	Mandatory /Optional	Description	Remark
1	status	int		M	รหัสสถานะผลของการทำรายการ (การเรียก API)	ดูรายละเอียดในภาคผนวก ก.
2	code	int		M	รหัสสถานะผลของการทำรายการ (ลำดับที่ 2)	ดูรายละเอียดในภาคผนวก ข.
3	message	string	255	M	คำอธิบายผลของการทำรายการ (การเรียก API)	
4	totalRecord	int		O	จำนวนข้อมูล	

5	data	object		O	ข้อมูลผลของการทำรายการ (การเรียก API)	หากการทำรายการไม่สำเร็จ จะไม่มีข้อมูล
5.1	data.portal	string	50	M	เป็นค่าที่จะออกให้โดย ChillPay	เช่น ChillPay
5.2	data.merchant	string	50	M	เป็นค่าที่จะออกให้โดย ChillPay	เช่น M000001
5.3	data.paymentCreditToken	string	32	M	ข้อมูล token สำหรับการทำรายการชำระเงิน	เป็น token ที่ออกให้โดยระบบ ChillCredit เพื่อใช้ในการทำรายการชำระเงินในขั้นต่อไป

ตัวอย่างการเรียก Api CreatePaymentCreditToken (Code ภาษา PHP)

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
    CURLOPT_URL => 'https://sandbox-bankdemo3.chillpay.co/ChillCredit/public/CreatePaymentCreditToken',
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => 'POST',
    CURLOPT_POSTFIELDS => '{
        "Portal": "portal01",
        "Merchant": "M000001",
        "MerchantAPIKey": "xxx",
        "CardNumber": "123456789",
        "ExpireMonth": "01",
        "ExpireYear": "2026",
        "NameOnCard": "testname",
        "CVV": "123",
        "PortalCreditToken": "123456789",
        "MerchantCheckSum": "2b332382ff893b8c4c18960d87499e4b"
    }',
    CURLOPT_HTTPHEADER => array(
        'Content-Type: application/json'
    ),
));
$response = curl_exec($curl);
```

```
curl_close($curl);  
?>
```

ตัวอย่าง ข้อมูล Response(JSON) ที่ได้รับจากการเรียก API เคสทำรายการสำเร็จ

```
{  
  "status": 0,  
  "code": 200,  
  "message": "Success",  
  "totalRecord": 1,  
  "data": {  
    "portal": "ChillPay",  
    "merchant": "M000001",  
    "paymentCreditToken": "-pUkAKDF3EhbN6lQ3m5rTafnrXbGDCfe"  
  }  
}
```

ตัวอย่าง ข้อมูล Response(JSON) ที่ได้รับจากการเรียก API เคสทำรายการไม่สำเร็จ

```
{  
  "status": 1,  
  "code": 1001,  
  "message": "Invalid CVV",  
  "totalRecord": 0,  
  "data": null  
}
```

B. ข้อมูลการทดสอบบน Sandbox

ข้อมูลหมายเลขบัตรเครดิต สำหรับทดสอบบน Sandbox

หมายเลขบัตรเครดิต - สำหรับทดสอบ Credit Card, Pay with points, Installment (ยกเว้นช่องทาง Krungsri Consumer และ Krungsri First Choice)				
Card Number	Card Scheme	Expiry (MM/YY)	CVV	OTP Pass
4141 0000 0000 1414	VISA	Any	Any	123456
5353 0000 0000 3535	MASTER Card	Any	Any	123456
3535 0000 0000 5353	JCB	Any	Any	123456
6262 0000 0000 2626	UnionPay	Any	Any	123456

C. ภาคผนวก

ภาคผนวก ก.

รหัสผลของการทำรายการ ลำดับที่ 1

Response Code	Description	Remark
0	ทำรายการสำเร็จ	Success
1	ทำรายการไม่สำเร็จ	Fail
2	รายการเกิดข้อผิดพลาด	Error
3	ระบบเกิดข้อผิดพลาด ไม่สามารถทำงานได้	System Error

ภาคผนวก ข.

รหัสผลของการทำรายการ ลำดับที่ 2

Response Code	Description	Remark
200	ทำรายการสำเร็จ	Success
1001	ข้อมูลพารามิเตอร์ไม่ถูกต้อง	Invalid Parameter
1002	รหัสร้านค้าไม่ถูกต้อง	Invalid Merchant Code
1003	API Key ไม่ถูกต้อง	Invalid API Key
1004	Bank Channel Code ไม่ถูกต้อง	Invalid Channel ID
1005	Route No. ไม่ถูกต้อง	Invalid Route No
1006	Order Number ไม่ถูกต้อง	Invalid Order No
1007	จำนวนเงินไม่ถูกต้อง	Invalid Amount
1008	Customer ID/Customer Name ไม่ถูกต้อง	Invalid Customer ID
1009	เบอร์โทรศัพท์มือถือไม่ถูกต้อง	Invalid Mobile Phone Number
1010	รหัสภาษาไม่ถูกต้อง	Invalid Language Code
1011	รหัส Checksum ไม่ถูกต้อง	Invalid Checksum
1012	เลขที่อ้างอิงรายการ(Transaction ID) ไม่ถูกต้อง	Invalid Transaction ID
1013	รายละเอียดการชำระเงินไม่ถูกต้อง	Invalid Description
1014	ข้อมูล Token Flag ไม่ถูกต้อง	Invalid Token Flag
1015	ข้อมูลรหัสสกุลเงินไม่ถูกต้อง	Invalid Currency Code
1016	ข้อมูล Credit Card Token ไม่ถูกต้อง	Invalid Credit Card Token
1017	ข้อมูลเดือนที่ผ่อนชำระไม่ถูกต้อง	Invalid Installment Month
1018	ข้อมูล IP Address ไม่ถูกต้อง	Invalid IP Address
1019	ข้อมูล Shop ID ไม่ถูกต้อง	Invalid Shop ID
1020	ข้อมูล URL ของรูปภาพสินค้าไม่ถูกต้อง	Invalid Product Image URL

1021	ข้อมูล Card Type ไม่ถูกต้อง	Invalid Card Type
1035	พารามิเตอร์ Token Type ไม่ถูกต้อง	Invalid Token Type
1036	พารามิเตอร์ Save Card ไม่ถูกต้อง	Invalid Save Card
2001	Account ของร้านค้ายังไม่ได้รับการอนุมัติ	Account Unauthorized
2002	API Key ร้านค้ายังไม่ได้รับการอนุมัติ	Invalid Merchant API Key
2003	ข้อมูล Checksum ไม่ถูกต้อง	Invalid Checksum Data
2004	ไม่พบข้อมูลช่องทางการชำระเงินสำหรับร้านค้า	Invalid Merchant Routing
2005	จำนวนเงินที่ชำระน้อยกว่าจำนวนเงินขั้นต่ำสุดที่กำหนดไว้	Payment minimum price {0} less than {1} เช่น Payment minimum price 15.00 less than 20.00
2006	จำนวนเงินที่ชำระมากกว่าจำนวนเงินขั้นสูงสุดที่กำหนดไว้	Payment maximum price {0} is over than {1} เช่น Payment maximum price 55000.00 is over than 50000.00
2007	ไม่สามารถสร้างรายการรับชำระเงินได้	Request Transaction Error
2008	ไม่สามารถสร้างรายการ Token รับชำระเงินได้	Request Transaction Token Error
2009	ไม่พบข้อมูลรายการที่ต้องการ	Transaction Not Found
2010	ไม่พบข้อมูล Credit Card Token ที่ใช้ชำระเงิน	Credit Card Token Not Found
2011	ไม่พบข้อมูลรหัสสกุลเงินที่ใช้ชำระเงิน	Currency Code Not Supported
2012	ไม่พบข้อมูลเดือนที่ใช้ผ่อนชำระ	Installment Month Not Found
2013	เบอร์โทรศัพท์มือถือที่ลงทะเบียนไว้กับ K PLUS ไม่ถูกต้อง	Mobile Phone No. not registered with K PLUS Mobile Application.
2014	ไม่พบ Email หรือรูปแบบ Email ไม่ถูกต้อง	Invalid/Incorrect email address.
3001	ระบบเกิดข้อผิดพลาดไม่สามารถทำรายการได้	System Error
3002	ระบบไม่สามารถให้บริการได้ในช่วงเวลา xx:xx:xx – xx:xx:xx กรุณาทำรายการเข้ามาใหม่อีกครั้ง	System is unavailable at xx:xx:xx – xx:xx:xx o'clock. Please try again later.
9001	KPLUS อยู่ระหว่างการปิดปรับปรุง	KPLUS System Maintenance Time

ภาคผนวก ค.

รหัสผลของการชำระเงินของลูกค้า (ข้อมูลการชำระเงินจากทางธนาคาร)

Response Code	Description	Remark
0	ทำรายการชำระเงินสำเร็จ	Success
1	ทำรายการชำระเงินไม่สำเร็จ	Fail
2	ลูกค้ายกเลิกทำรายการชำระเงิน	Cancel
3	เกิดข้อผิดพลาดระหว่างทำรายการชำระเงิน	Error
9	รอการชำระค่าสินค้า/บริการ	Transaction Pending
20	ทำรายการ Void สำเร็จ	Void Success
21	ทำรายการ Refund สำเร็จ	Refund Success
22	ทำรายการเพื่อขอ Refund	Request to Refund
23	ทำรายการโอนเงินให้ร้านค้าสำเร็จ	Settlement Success
24	ทำรายการ Void ไม่สำเร็จ	Void Fail
25	ทำรายการ Refund ไม่สำเร็จ	Refund Fail

ภาคผนวก ง.

รหัสสกุลเงิน (Currency Code)

No.	Code (ISO Code)	Currency
1	764	THB - บาทไทย
2	840	USD - ดอลลาร์สหรัฐ
3	978	EUR - ยูโร
4	392	JPY - เยนญี่ปุ่น
5	826	GBP - ปอนด์สเตอร์ลิง (สหราชอาณาจักร)
6	036	AUD - ดอลลาร์ออสเตรเลีย
7	554	NZD - ดอลลาร์นิวซีแลนด์
8	344	HKD - ดอลลาร์ฮ่องกง
9	702	SGD - ดอลลาร์สิงคโปร์
10	756	CHF - ฟรังก์สวิส
11	458	MYR - ริงกิตมาเลเซีย
12	156	CNY - หยวน

ภาคผนวก จ.

สถานะการชำระเงินของระบบ Credit Card Inline

Payment Status	Description	Remark
Paid	การจ่ายเงินสำเร็จ	การจ่ายเงินสำเร็จแล้ว สามารถแสดงหน้าผลการชำระได้เลย
WaitAuthorize	พารามิเตอร์ไม่ถูกต้อง	การจ่ายเงินยังไม่สำเร็จ รอการยืนยันการชำระเงิน (Confirm OTP) อยู่ ร้านค้าต้อง Redirect ให้ลูกค้าไปหน้ายืนยัน OTP ของธนาคาร เพื่อทำการยืนยัน OTP ก่อน