



**K PAYMENT
GATEWAY**

คู่มือการเชื่อมต่อ

K-Payment Gateway

Merchant Integration Guideline

Version 4.3 (ฉบับภาษาไทย)



สารบัญ

1. ภาพรวมของระบบ K-Payment Gateway	12
2. การสมัครใช้บริการ K-Payment Gateway และข้อมูลที่ใช้ในการติดตั้งระบบ	12
2.1 คุณสมบัติของระบบ	12
2.2 การทดสอบบน Sandbox	13
2.3 การเปิดใช้งานจริง	13
3. รูปแบบการให้บริการ	14
3.1 รูปแบบการชำระเงิน	14
3.1.1 บัตรเครดิตและเดบิต	14
3.1.2 Thai QR code	15
3.1.3 E-Wallet ประเทศไทย	15
3.1.3.1 Alipay	15
3.1.3.2 WeChat Pay	15
3.1.4 Pay PLUS – การรับชำระเงินผ่านโทรศัพท์มือถือ	16
3.1.4.1 Pay PLUS (Notification)	16
3.1.4.2 Pay PLUS (App Switch)	16
3.2 รูปแบบการเชื่อมต่อระบบ	17
3.2.1 Embedded UI	17
3.2.2 Inline UI	17
3.2.3 Redirect	18
3.2.4 Direct API	19
3.2.5 Webview for Mobile Native App	19
4. วิธีการเชื่อมต่อเพื่อรับชำระเงิน	20
4.1 Embedded UI	20
4.1.1 การเชื่อมต่อเพื่อรับชำระเงินด้วยบัตร Visa, MasterCard และ JCB	20
4.1.1.1 การติดตั้งปุ่ม K-Pay	20
1. ชำระเต็มจำนวน (Full Payment) แบบ MCC	21
2. ชำระเต็มจำนวน (Full Payment) แบบ DCC	23
3. ผ่อนชำระ (SmartPay)	25



4.1.1.2 รับค่า Token จากระบบ K-Payment Gateway	31
1. Response Parameter: ชำระเต็มจำนวน (Full Payment) แบบ MCC.....	31
2. Response Parameter: ชำระเต็มจำนวน (Full Payment) แบบ DCC	32
3. Response Parameter: ผ่อนชำระ (SmartPay).....	32
4.1.1.3 ใช้ Token ที่ได้รับในการเรียก API เพื่อเรียกเก็บเงินให้เสร็จสมบูรณ์	34
4.1.1.4 วิธีการเชื่อมต่อ	35
1. วิธีการเชื่อมต่อแบบ 3D Secure.....	35
2. วิธีการเชื่อมต่อแบบ Non 3D Secure	39
4.1.2 การเชื่อมต่อเพื่อรับชำระเงินด้วย QR Code	41
4.1.2.1 การติดตั้งปุ่ม K-Pay	41
Attribute สำหรับชำระ QR Code.....	42
4.1.2.2 วิธีการเชื่อมต่อด้วย QR Code.....	43
4.2 Redirect	45
4.2.1 การเชื่อมต่อเพื่อรับชำระเงินด้วยบัตร UnionPay (TPN/UPI).....	45
4.2.2 การเชื่อมต่อเพื่อรับชำระเงินผ่าน Alipay	48
5. Feature สำหรับชำระเงินด้วยบัตร Visa, MasterCard และ JCB.....	50
5.1 จดจำเลขบัตร (Remember card)	50
5.1.1 การชำระเงินครั้งแรก	50
1. ชำระสินค้าครั้งแรกพร้อมจดจำหมายเลขบัตรไว้ใช้จ่ายครั้งต่อไป	50
2. Register 3D จดจำบัตรก่อนทำการซื้อสินค้า	51
5.1.2 การชำระเงินครั้งต่อไป.....	52
1. รูปแบบ Embedded UI	52
2. รูปแบบ Direct API	52
6. API Reference	53
6.1 Charge API	53
6.1.1. Create Charge	53
Request Parameters	53
Response Parameters.....	56
Sample Request	61



Sample Request for Card Payment – Visa, MasterCard & JCB	61
Sample Request for Card Payment – UnionPay (TPN/UPI)	61
Sample Request for Alipay	62
Sample Response.....	62
6.1.2. Inquiry Transaction	64
Path Parameters	64
Response Parameter	64
Sample Request	70
Sample Response.....	70
6.1.3. Void Transaction.....	72
Path Parameter.....	72
Request Parameters	72
Response Parameter	72
Sample Request	76
Sample Response.....	76
6.1.4. Settle Transaction	77
Path Parameter.....	77
Response Parameter	78
Sample Request	81
Sample Response.....	81
6.1.5. Refund Transaction	83
Path Parameter	83
Request Parameters	84
Response Parameters	84
Sample Request	87
Sample Response Success	88
Sample Response Error.....	89
6.2.Customer API.....	89
6.2.1. Create Customer	89



Request Parameter.....	89
Response Parameter	90
Sample Request	91
Sample Response.....	91
6.2.2. Inquiry Customer	92
Path Parameter.....	92
Response Parameter	92
Sample Request	93
Sample Response.....	93
6.2.3. Update Customer	94
Path Parameter.....	94
Request Parameter.....	94
Response Attributes	95
Sample Request	96
Sample Response.....	96
6.2.4 Delete Customer	97
Path Parameter.....	97
Response Parameter	97
Sample Request	97
Sample Response.....	98
6.2.5 Add New Card to Customer	98
Path Parameter.....	98
Response Parameter	99
Sample Request	100
Sample Response.....	100
6.2.6 Delete Card from Customer	101
Path Parameters.....	101
Response Parameter	102
Sample Request	102



Sample Response.....	102
6.3.Order API	103
6.3.1 Create Order	103
Request Parameter.....	103
Response Parameter	105
Sample Request	107
Sample Response.....	107
6.3.2 Inquiry Order.....	108
Path Parameter.....	108
Response Parameter	108
Sample Request	110
Sample Response.....	111
6.4.QR API.....	111
6.4.1. Inquiry QR Transaction	111
Path Parameter.....	111
Response Parameter	112
Sample Request	114
Sample Response.....	114
6.4.2. Void QR Transaction.....	115
Path Parameter.....	115
Response Attributes	115
Sample Request	117
Sample Response.....	117
6.4.3. Cancel QR Transaction.....	118
Path Parameter.....	118
Response Attributes	118
Sample Request	119
Sample Response.....	119
6.5.WebHook Notify API	120



6.5.1. How to calculate checksum	120
6.5.2. WebHook Notify API for Card Payment.....	122
Request Parameter.....	122
Response Parameters.....	125
Sample Request	126
Sample Response.....	126
6.5.3. WebHook Notify API for QR Payment	127
Request Parameter.....	127
Response Parameters.....	128
Sample Request	129
Sample Response.....	129
6.5.4. WebHook Notify API for Alipay	130
Request Parameter.....	130
Response Parameters.....	132
Sample Request	132
Sample Response.....	132
7. บัตรทดสอบใน Sandbox และ Simulator	133
7.1. การชำระเงินแบบ 3D Secure.....	133
สำหรับชำระเต็มจำนวนแบบ MCC และ ผ่อนชำระ (SmartPay)	133
สำหรับชำระเต็มจำนวนแบบ DCC	133
7.2. การชำระเงินแบบ Non 3D Secure.....	134
สำหรับชำระเต็มจำนวนแบบ MCC และ ผ่อนชำระ (SmartPay)	134
สำหรับชำระเต็มจำนวนแบบ DCC	134
7.3. บัตรทดสอบ UnionPay (TPN/UPI).....	135
7.4. ทดสอบการชำระเงินด้วย Thai QR Simulator ใน Sandbox	135
8. ตัวอย่าง Code	137
8.1. Code Java สำหรับการรับ Token	137
8.2. ตัวอย่าง Code Java สำหรับการเรียก Create Order API.....	137
8.3. ตัวอย่าง Code Java สำหรับการรับค่า Charge ID.....	138



8.4.ตัวอย่าง Code Java สำหรับการรับ WebHook Notify	138
9. Appendix	142
Transaction State and Status	142
Failure Code for Card Payment	142
Failure Code for QR Payment	148
API Error Code	150
ALL API.....	150
Charge API.....	150
Customer API.....	151
Order API	151
QR API	152
Refund API	152
10.คำถาวรที่พบบ่อย	154

Figures

Figure 1 ภาพรวมของระบบ K-Payment Gateway	12
Figure 2 บัตรเครดิตและเดบิต	14
Figure 3 Thai QR code	15
Figure 4 E-Wallet ประเทศไทย	15
Figure 5 Pay PLUS – การรับชำระเงินผ่านโทรศัพท์มือถือ	16
Figure 6 ตัวอย่างหน้าจากการชำระเงินแบบต่างๆ	17
Figure 7 ตัวอย่างหน้าจอ Inline UI	18
Figure 8 ตัวอย่างหน้าจอ Redirect	18
Figure 9 ตัวอย่างหน้าจอ Webview for Mobile Native App	19
Figure 10 หน้าจອการชำระเงินเต็มจำนวนแบบ MCC	22
Figure 11 หน้าจອการชำระเงินเต็มจำนวนแบบ DCC	24
Figure 12 ขั้นตอน set up ข้อมูล SmartPay flow	27
Figure 13 หน้าจອการชำระเงินแบบผ่อนชำระ (SmartPay)	31
Figure 14 Embedded UI 3D Secure Flow	36



Figure 15 Embedded UI Non 3D Secure Flow	39
Figure 16 QR Flow	43
Figure 17 Alipay Flow	48



Revisions/Change Control:

Version	Description of Changes	Date of Changes
1.0	- Initial Version	05/03/2019
1.1	<ul style="list-style-type: none"> - Update examples - Update descriptions for Embedded UI method - Update test cards - Update attributes for QR API (Create QR) - Add "How to Use QR Simulator Tools" - Update FAQ 	17/04/2019
1.11	<ul style="list-style-type: none"> - Update integration steps for card payment - Update FAQ 	18/04/2019
2.0	<ul style="list-style-type: none"> - Add new integration method for card payment that is "Inline UI". - Add new "WebHook Notify" feature to receive real-time payment result (applicable both card payment and QR payment) - Add WebHook Notify API - Add new attributes for Charge API - Update descriptions for implement card payment - Update descriptions for implement QR payment - Update FAQ 	29/05/2019
2.1	<ul style="list-style-type: none"> - Update attributes for SmartPay. - Add "Alipay" and "WeChat" as new payment methods. - Add UPI (UnionPay) for card payment. - Add "Redirect" as new integration method. - Add new test cards. - Update descriptions for Remember card. - Update APIs attributes. - Update FAQ 	20/08/2019
2.2	<ul style="list-style-type: none"> - Update transaction_state for TPN/UPI Integration. - Update attributes used for setup KBank UI. - Update integration flow for Card Payment, QR Code Payment and Alipay. - Remove Refund Transaction API. 	05/02/2020



3.0	<ul style="list-style-type: none"> - Add "Webview for Mobile Native App" as new card payment integration method. - Add "Pay PLUS" as new payment method. - Add attributes for BIN Campaign, Convenience Fee and MPI. - Update attribute used for setup KBank UI. - Update "Remember card" topic. - Update condition for Void QR Transaction API. - Update APIs and request/reponse message examples. - Remove Create QR API 	19/10/2020
4.0	<ul style="list-style-type: none"> - Add feature remember card mode register 3D - Add how to Set up SmartPay - Update integration flow Card Payment, QR Code Payment , UnionPay (TPN/UPI) and Alipay. - Update APIs Create Charge API add smartpay object - Update APIs Inquiry Transaction API 	12/01/2022
4.1	<ul style="list-style-type: none"> - Update Register 3D remembers the card before payment - Update Steps to Set up information SmartPay - Add smartpay sample for Create Charge API 	10/02/2022
4.2	<ul style="list-style-type: none"> - Update Return parameter post token - Update Inquiry Transaction API, Create Charge API,Customer API and Order API 	07/03/2022
4.3	<ul style="list-style-type: none"> - Update how to embedding K-Pay button for Installment (SmartPay) - Update how to receive Token from K-Payment Gateway for Installment (SmartPay) - Update the Inquiry Transaction API for Inquiry data more than 6 months - Add Refund API 	20/04/2022



1. ภาพรวมของระบบ K-Payment Gateway



Figure 1 ภาพรวมของระบบ K-Payment Gateway

ระบบ K-Payment Gateway ของธนาคารกสิกรไทย จำกัด (มหาชน) เป็นบริการรับชำระค่าสินค้าและบริการออนไลน์ด้วยช่องทางที่หลากหลาย ไม่ว่าจะเป็น รับชำระเงินด้วยบัตรเครดิตและบัตรเดบิตที่สามารถรับทุกสกุลเงินทั่วโลก หรือ รับชำระเงินด้วย QR Code ซึ่งรองรับทั้ง Thai QR Payment ผ่านโพรเจกต์ K-Payment Gateway EXCELLENCE ที่มีความปลอดภัยสูงสุดทางด้านข้อมูล เมื่อทำการชำระค่าสินค้าและบริการออนไลน์

ร้านค้าสามารถเชื่อมต่อกับ API ของระบบ K-Payment Gateway และทดสอบการจ่ายเงินผ่าน Sandbox ได้เอง เพื่อเตรียมความพร้อมก่อนเปิดใช้ระบบจริงได้อย่างรวดเร็ว ทำให้สามารถรับความต้องการทางธุรกิจที่หลากหลาย และมีประสิทธิภาพที่ดีมากยิ่งขึ้นกว่าเดิม

2. การสมัครใช้บริการ K-Payment Gateway และข้อมูลที่ใช้ในการติดตั้งระบบ

2.1 คุณสมบัติของระบบ

ระบบของร้านค้าจำเป็นต้องมี SSL Certificate ที่ออกโดย CA (Certificate Authority) เพื่อความปลอดภัยในการรับส่งข้อมูลระหว่างร้านค้าและธนาคาร โดยจะต้องมีคุณสมบัติดังนี้

- ✓ มีการเข้ารหัสแบบ 128-bit ขึ้นไป
- ✓ สนับสนุน TLS 1.2

ในกรณีที่ร้านค้าต้องการเชื่อมต่อแบบ Direct API ระบบของร้านค้าจะต้องได้รับการรับรองมาตรฐานด้านความมั่นคงปลอดภัยของการใช้บัตรชำระเงิน (PCI Data Security Standard หรือ PCI DSS) ซึ่งสามารถศึกษารายละเอียดได้จากเว็บไซต์ของ PCI DSS ดังนี้

https://www.pcisecuritystandards.org/document_library



2.2 การทดสอบบน Sandbox

ร้านค้าที่สนใจบริการ K-Payment Gateway สามารถกรอกใบสมัครบริการ K-Payment Gateway และส่งเอกสารประกอบการสมัครมา�ังธนาคารกสิกรไทย เพื่อรับ Username และ Password สำหรับเข้าสู่ Merchant Portal (Sandbox) ซึ่งร้านค้าสามารถดูข้อมูลที่จำเป็นในการทำการทดสอบ เช่น Public Key หรือ Secret Key ได้ด้วยตนเอง และต้องทำการทดสอบให้ผ่านครบถ้วนแบบบริการที่เปิด

URL สำหรับ Merchant Portal (Sandbox)

<https://dev-kpaymentgateway.kasikornbank.com/portal/v1/login>

2.3 การเปิดใช้งานจริง

หลังจากร้านค้าทดสอบระบบผ่านครบถ้วนแบบบริการที่เปิดและพร้อมใช้งานจริงแล้ว สามารถกรอกใบสมัครบริการ K-Payment Gateway และส่งเอกสารประกอบการสมัครมายังธนาคารกสิกรไทย เมื่อธนาคารตรวจสอบข้อมูลและดำเนินการเปิดใช้บริการให้ร้านค้าแล้ว ร้านค้าจะได้รับ Username และ Password สำหรับเข้าสู่ Merchant Portal และเริ่มรับชำระเงินได้ทันที (สำหรับรายละเอียดการใช้งาน Merchant Portal จะอยู่ในเอกสาร "คู่มือการใช้งาน Merchant Portal K-Payment Gateway (ฉบับภาษาไทย)"

URL สำหรับ Merchant Portal (Production)

<https://kpaymentgateway.kasikornbank.com/portal/v1/login>



3. รูปแบบการให้บริการ

3.1 รูปแบบการชำระเงิน

K-Payment Gateway รองรับการชำระเงินจากหลากหลายช่องทาง เพื่ออำนวยความสะดวกให้กับร้านค้า และเพิ่มตัวเลือกให้ลูกค้าในการชำระค่าสินค้าและบริการออนไลน์ มีทั้งหมด 5 รูปแบบการชำระเงิน ดังนี้

3.1.1 บัตรเครดิตและเดบิต

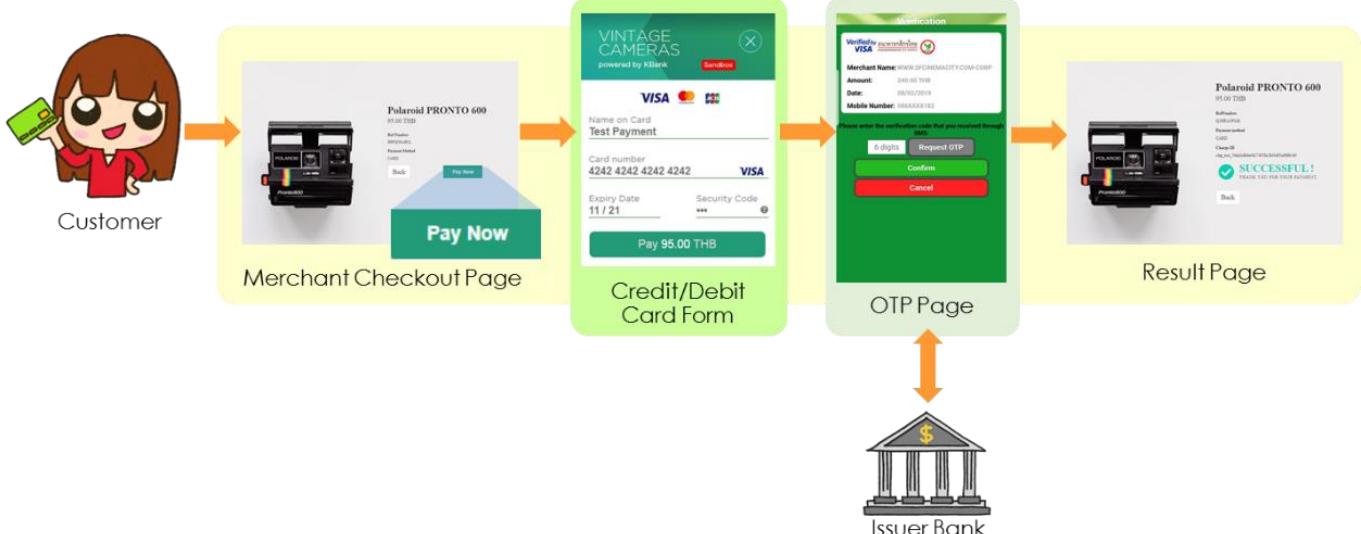


Figure 2 บัตรเครดิตและเดบิต

- ✓ รับบัตร Visa, MasterCard, JCB และ UnionPay (TPN/UPI)
- ✓ รองรับการทำรายการแบบ 3D Secure ผ่านระบบ Verified by VISA (VbV), MasterCard SecureCode และ J/Secure
- ✓ Full MCC (Multi Currency Conversion): รองรับ 36 สกุลเงินทั่วโลก
- ✓ Full DCC (Dynamic Currency Conversion): แปลงสกุลเงินได้อัตโนมัติสำหรับบัตรเครดิตต่างชาติ
- ✓ สำหรับบัตรเครดิตกสิกรไทย สามารถเลือกรับชำระเงินได้ทั้งแบบเต็มจำนวนและแบบผ่อนชำระ



3.1.2 Thai QR code



Figure 3 Thai QR code

ลูกค้าสามารถใช้ Mobile Banking Application ของธนาคารที่ได้รับอนุญาต เช่น ธนาคารกสิกรไทย, ธนาคารไทยพาณิชย์, ธนาคารกรุงไทย, ธนาคารกรุงเทพ, ธนาคารออมสิน, ธนาคารกรุงศรีอยุธยา, ธนาคารทหารไทย, ธนาคารธนชาต, ธนาคารเกียรตินาคิน หรือ Mobile Application ของผู้ให้บริการ E-Wallet ที่รองรับ เช่น AIS mPAY, TrueMoney เป็นต้น สแกน QR Code มาตรฐานเพื่อชำระเงินได้ทันที

3.1.3 E-Wallet ประเทศไทย



Figure 4 E-Wallet ประเทศไทย

3.1.3.1 Alipay

- เพิ่มช่องทางในการรับชำระเงินจากนักท่องเที่ยวจีน
- ร้านค้าไม่จำเป็นต้องมีบัญชีธนาคารของประเทศไทย ไม่ต้องตั้งค่าบัญชี Alipay

3.1.3.2 WeChat Pay

- ลูกค้าสามารถใช้ WeChat Application ในการสแกน QR Code เพื่อชำระเงินได้ทันที



3.1.4 Pay PLUS – การรับชำระเงินผ่านโทรศัพท์มือถือ



*Depend on the option when applying this service

Figure 5 Pay PLUS – การรับชำระเงินผ่านโทรศัพท์มือถือ

3.1.4.1 Pay PLUS (Notification)

- ลูกค้าสามารถชำระค่าสินค้า/บริการ โดยอิงจากเบอร์มือถือที่ผูกอยู่กับแอปพลิเคชัน และตัดเงินโดยตรงจากบัญชีได้อย่างสะดวกรวดเร็ว
- เรียกเก็บเงินไปยังเบอร์มือถือใช้ K PLUS
- เรียกเก็บเงินไปยังเบอร์มือถือที่ลงทะเบียนพร้อมเพรีย

3.1.4.2 Pay PLUS (App Switch)

เป็นที่รู้จักในชื่อ "Pay with K PLUS" ซึ่งจะช่วยให้ลูกค้าสามารถชำระเงินแบบไร้รอยต่อ (Seamless Payment Experience) ระหว่างแอปพลิเคชันของร้านค้ากับ K PLUS ได้อย่างสะดวก รวดเร็ว ไม่ต้องสลับหน้าจอกลับไปกลับมา



3.2 รูปแบบการเชื่อมต่อระบบ

ร้านค้าสามารถเชื่อมต่อกับ K-Payment Gateway ได้ 5 วิธี ดังนี้

3.2.1 Embedded UI

การเชื่อมต่อประเภทนี้คือการฝังปุ่ม

Pay Now

ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้า (แนะนำ)

ให้ฝังปุ่ม K-Pay (ไว้ที่หน้าชำระเงิน) เมื่อลูกค้าคลิกปุ่ม K-Pay จะปรากฏหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน หรือหน้าจอแสดง QR Code สำหรับชำระเงิน แล้วระบบ K-Payment Gateway จะดำเนินการเชื่อมต่อกับระบบที่เกี่ยวข้องและแจ้งผลการชำระเงินให้ร้านค้าทราบต่อไป

ตัวอย่างหน้าจອการชำระเงินแบบต่างๆ

Figure 6 ตัวอย่างหน้าจອการชำระเงินแบบต่างๆ

3.2.2 Inline UI

การเชื่อมต่อประเภทนี้คือการแทรก Tag HTML และปุ่มตามที่ทาง Kbank กำหนดเพื่อรองรับการชำระเงินด้วยบัตรไว้ในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน (โดยไม่ต้องเก็บข้อมูลบัตร) เมื่อลูกค้ากรอกรายละเอียดบัตรและคลิกปุ่มเพื่อยืนยันการชำระเงินแล้ว ระบบ K-Payment Gateway จะดำเนินการเชื่อมต่อกับระบบที่เกี่ยวข้องและแจ้งผลการชำระเงินให้ร้านค้าทราบต่อไป



ตัวอย่างหน้าจอ Inline UI

The figure displays two side-by-side screenshots of a payment form. On the left, the title is "Polaroid PRONTO 600" with a price of "100.00 THB". It features a placeholder image of a Polaroid Pronto 600 camera. Below the title, there is a field labeled "Ref Number" with the value "2OYLV7GBCO". A yellow dashed box highlights a payment input field containing "Name on Card" and "Test Payment", along with a card number "4242 4242 4242 4242" and a VISA logo. To the right of the card number are fields for "Expiry Date" (02 / 22) and "Security Code" (***). A checkbox below these fields reads: "Consent KASIKORN BANK PCL to collect and use this card information for next transactions". A green button at the bottom says "Pay 100.00 THB". On the right, a similar screenshot shows the same payment details and a "Pay 100.00 THB" button.

Figure 7 ตัวอย่างหน้าจอ Inline UI

3.2.3 Redirect

การเชื่อมต่อประเภทนี้คือการนำลูกค้าไปยัง URL ที่ปลอดภัยของระบบที่เกี่ยวข้อง เพื่อดำเนินการทำธุรกรรมให้เสร็จสิ้น หลังจากนั้นระบบ K-Payment Gateway จะแจ้งผลการทำธุรกรรมให้ร้านค้าทราบต่อไป ใช้สำหรับการรับชำระด้วยบัตร UnionPay และการรับชำระเงินด้วย Alipay

ตัวอย่างหน้าจอ Redirect

The figure shows two screenshots of a payment process. The left screenshot is titled "UnionPay Card Payment" and includes fields for "Card Number" (9100****0010), "Expiration Date" (03 MM 21 YY), "CVN2", "PIN", and "Verification Code" (brco). The right screenshot shows a QR code with the amount "0.25" and instructions to "Scan to Pay (元)". It also includes a small image of a computer monitor with the Alipay logo.

ชำระเงินด้วยบัตร UnionPay

ชำระเงินด้วย Alipay

Figure 8 ตัวอย่างหน้าจอ Redirect



3.2.4 Direct API

การเชื่อมต่อประเภทนี้เป็นการเรียก K-Payment Gateway API โดยตรง ซึ่งร้านค้าที่จะเชื่อมต่อแบบนี้ได้จะต้องได้รับการรับรองมาตรฐาน PCI DSS แล้วเท่านั้น เนื่องจาก API บางตัวกำหนดให้ร้านค้าส่งข้อมูลสำคัญเพื่อดำเนินการ เช่น หมายเลขบัตรเครดิตแบบเต็ม

ศึกษารายละเอียดได้จากเว็บไซต์ของ PCI DSS ดังนี้

https://www.pcisecuritystandards.org/document_library

3.2.5 Webview for Mobile Native App

การเชื่อมต่อประเภทนี้ใช้สำหรับร้านค้าที่พัฒนาแอปพลิเคชันสำหรับโทรศัพท์มือถือ โดยการเรียก Webview เพื่อแสดงหน้าจอให้ระบุรายละเอียดบัตรที่ต้องการใช้ในการชำระเงินหรือลงทะเบียนบัตร และระบบ K-Payment Gateway จะดำเนินการเชื่อมต่อกับระบบที่เกี่ยวข้องและแจ้งผลการชำระเงินหรือลงทะเบียนบัตรให้ร้านค้าทราบต่อไป

ตัวอย่างหน้าจอ Webview for Mobile Native App

Please fill your card.

Card number VISA Mastercard JCB

Name on card

Expiry date CVV ?

Consent KASIKORN BANK PCL to collect and use this card information for next transactions

Pay 1,200 THB

Secured By

Please fill your card.

Card number VISA Mastercard JCB

Name on card

Expiry date CVV ?

Consent KASIKORN BANK PCL to collect and use this card information for next transactions

Save

Secured By

หน้าจอชำระเงิน

หน้าจอลงทะเบียนบัตร

Figure 9 ตัวอย่างหน้าจอ Webview for Mobile Native App



4. วิธีการเชื่อมต่อเพื่อรับชำระเงิน

4.1 Embedded UI

Embedded UI คือ วิธีการเชื่อมต่อเพื่อรับชำระเงิน โดยการฝัง Code HTML ไว้ที่เว็บไซต์หรือแอปพลิเคชันมือถือของ ของร้านค้าที่หน้าชำระเงิน ซึ่งมี 2 รูปแบบการรับชำระเงินคือ ชำระเงินด้วยบัตร Visa, MasterCard, JCB และ ชำระเงิน ด้วย QR Code

4.1.1 การเชื่อมต่อเพื่อรับชำระเงินด้วยบัตร Visa, MasterCard และ JCB

4.1.1.1 การติดตั้งปุ่ม K-Pay

การติดตั้งปุ่ม K-Pay เพื่อเรียกหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงินร้านค้าทำการฝังปุ่ม

K-Pay Pay Now ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน เมื่อลูกค้าคลิกปุ่ม K-Pay จะ ปรากฏหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน

เพิ่ม Code ตามตัวอย่างด้านล่างนี้เข้าไปที่หน้าชำระเงิน แล้วแทนที่ด้วยข้อมูลของร้านค้า พร้อมทั้งระบุ URL สำหรับรับค่า Response กลับจากธนาคารใน action

```
<form method="POST" action="/checkout">
<script type="text/javascript"
src="https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js"
data-apikey="pkey_test_20923L43S4RABbtwVoM1zsopFCYzyZnyMwFsn"
data-amount="74.00"
data-currency="THB"
data-payment-methods="card"
data-name="Your Shop Name"
data-mid="401001001001001" >
</script>
</form>
```

ค่า src หรือ URL สำหรับเรียก JavaScript จะแตกต่างกันระหว่างการทำรายการทดสอบบน Sandbox และ การทำรายการจริง ดังนี้

- URL สำหรับเรียก JavaScript บน Sandbox

<https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js>

- URL สำหรับเรียก JavaScript เพื่อทำรายการจริง

<https://kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js>



!

Data Attribute Reference

M - Mandatory, C - Conditional, O - Optional

1. ชำระเต็มจำนวน (Full Payment) แบบ MCC

▪Attribute สำหรับชำระเต็มจำนวน (Full Payment) แบบ MCC

No	Field Name	Type	Mandatory	Description
1	data-apikey	Varchar (50)	M	Your public key Sample Data: data-apikey="pkey_prod_123456"
2	data-amount	Decimal (10,2)	M	Amount Sample Data: data-amount="1000.50"
3	data-payment-methods	Enum (card, qr, redirect)	M	Payment method Sample Data: data-payment-methods="card"
4	data-mid	Varchar (15)	M	Your merchant ID Sample Data: data-mid="444123456789001"
5	data-name	Varchar (15)	O	Your shop name Sample Data: data-name="Awesome Shop"
6	data-currency	Varchar (3)	O	Currency unit (Default THB) Sample Data: data-currency="USD"
7	data-customer-id	Varchar (50)	O	Customer ID to show saved card Sample Data: data-customer-id="cust_prod_12345"
8	data-campaign-id	Varchar (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion. Sample Data: data-campaign-id="00000001"



Sample Attribute

```
<form method="POST" action="/checkout">
<script type="text/javascript"
src="https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js"
data-apikey="pkey_test_20923L43S4RABbtwVoMlzopFCYzyZnyMwFsn"
data-amount="1.00"
data-currency="USD"
data-payment-methods="card"
data-name="Your Shop Name"
data-mid="402202106043001" >
</script>
</form>
```

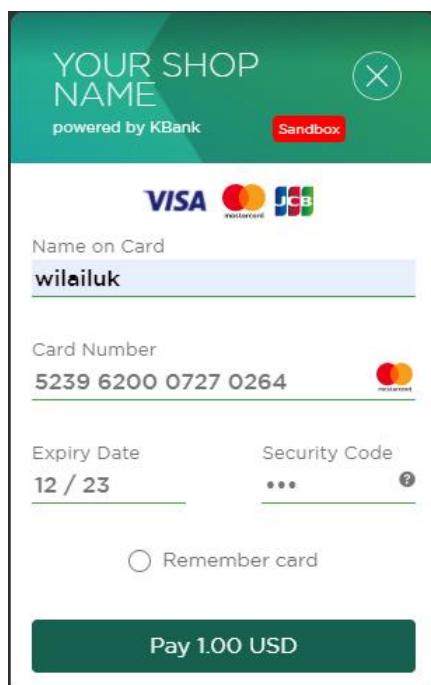


Figure 10 หน้าจอการชำระเงินเต็มจำนวนแบบ MCC



2. ชำระเต็มจำนวน (Full Payment) แบบ DCC

- Attribute สำหรับชำระเต็มจำนวน (Full Payment) แบบ DCC

!

รองรับเฉพาะประเภทบัตร Visa, MasterCard

No	Field Name	Type	Mandatory	Description
1	data-apikey	Varchar (50)	M	Your public key Sample Data: data-apikey="pkey_prod_123456789"
2	data-amount	Decimal (10,2)	M	Amount Sample Data: data-amount="1000.50"
3	data-payment-methods	Enum (card, qr, redirect)	M	Payment method Sample Data: data-payment-methods="card"
4	data-mid	Varchar (15)	M	Your merchant ID (Full DCC) Sample Data: data-mid="451123456789001"
5	data-name	Varchar (15)	O	Your shop name Sample Data: data-name="Awesome Shop"
6	data-customer-id	Varchar (50)	O	Customer ID to show saved card Sample Data: data-customer-id="cust_prod_1234"
7	data-campaign-id	Varchar (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion. Sample Data: data-campaign-id="00000001"



Sample Attribute

```
<form method="POST" action="/checkout">
<script type="text/javascript"
src="https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js"
data-apikey="pkey_test_20923L43S4RABbtwVoMlzopFCYzyZnyMwFsn"
data-amount="1.00"
data-payment-methods="card"
data-name="Your Shop Name"
data-mid="451111111111001" >
</script>
</form>
```

YOUR SHOP NAME
powered by KBank Sandbox

VISA **MasterCard** **JCB**

Name on Card
wilailuk

Card Number
5239 6200 0727 0264

Expiry Date **12 / 23** Security Code ******* ?

Remember card

Your credit card currency: USD(Dollar)
Exchange rate: 33.560 Baht/USD

0.00 USD

1.00 THB

I accept the payment currency and
exchange rate as stated above.
 Cardholder selected transaction
currency is final

Pay 1.00 THB **Pay USD**

Figure 11 หน้าจอการชำระเงินเต็มจำนวนแบบ DCC



3. ผ่อนชำระ (SmartPay)

เพื่อรับการผ่อนชำระประเภท ร้านค้ารับภาระดอกเบี้ย(The Merchant absorbs interest), ลูกค้ารับภาระดอกเบี้ย(The cardholder absorbs interest) และ ผู้ผลิตรับภาระดอกเบี้ย(Supplier absorbs interest)
ซึ่งมีวิธีการเชื่อมแบ่งออกเป็น 2 รูปแบบคือ SmartpayID และ Installment Plan

! วิธีการเชื่อมต่อแบบผ่อนชำระ (SmartPay)

1. SmartpayID จะรองรับการผ่อนชำระประเภท ร้านค้ารับภาระดอกเบี้ย และลูกค้ารับภาระดอกเบี้ย เท่านั้น
2. Installment Plan จะรองรับการผ่อนชำระประเภท ร้านค้ารับภาระดอกเบี้ย, ลูกค้ารับภาระดอกเบี้ย และ ผู้ผลิตรับภาระดอกเบี้ย

**ร้านค้าต้องเลือกส่ง Parameter อย่างใดอย่างหนึ่งเท่านั้น!

▪ ผ่อนชำระ (SmartPay) แบบ SmartpayID

ผ่อนชำระ (SmartPay) แบบ SmartpayID จะรองรับการผ่อนชำระประเภท ร้านค้ารับภาระดอกเบี้ย และ ลูกค้ารับภาระดอกเบี้ย เท่านั้น ให้ระบุ Attribute ตามตารางด้านล่าง

- Attribute สำหรับผ่อนชำระ (SmartPay)

No	Field Name	Type	Mandatory	Description
1	data-apikey	varchar (50)	M	Your public key Sample Data: data-apikey="pkey_prod_12345"
2	data-amount	decimal (10,2)	M	Amount Sample Data: data-amount="1000.50"
3	data-payment-methods	enum (card, qr, redirect)	M	Payment method Sample Data: data-payment-methods="card"
4	data-mid	varchar (15)	M	Your merchant ID Sample Data: data-mid="401123456789001"
5	data-smartpay-id	varchar (4)	M	Smart Pay ID Sample Data: data-smartpay-id="0001"
6	data-term	integer	O	Term Sample Data: data-term="10"
7	data-name	varchar (15)	O	Your shop name Sample Data: data-name="Awesome Shop"
8	data-currency	varchar (3)	O	Currency unit (Default THB) Sample Data: data-currency="USD"



No	Field Name	Type	Mandatory	Description
9	data-customer-id	varchar (50)	C	<p>Customer ID to show saved card</p> <p>Sample Data:</p> <p>data-customer-id="cust_prod_12345"</p>
10	data-campaign-id	varchar (8)	C	<p>BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion.</p> <p>Sample Data: data-campaign-id="00000001"</p>

- Sample Attribute

```

<form method="POST" action="/checkout">
<script type="text/javascript"
    src="https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js"
    data-apikey="pkey_test_20923L43S4RABbtwVoMlzsopFCYzyZnyMwFsn"
    data-amount="1.00"
    data-currency="THB"
    data-payment-methods="card"
    data-name="Your Shop Name"
    data-mid="401202106042001"
    data-smartpayid="0001"
    data-term="10" >
</script>
</form>

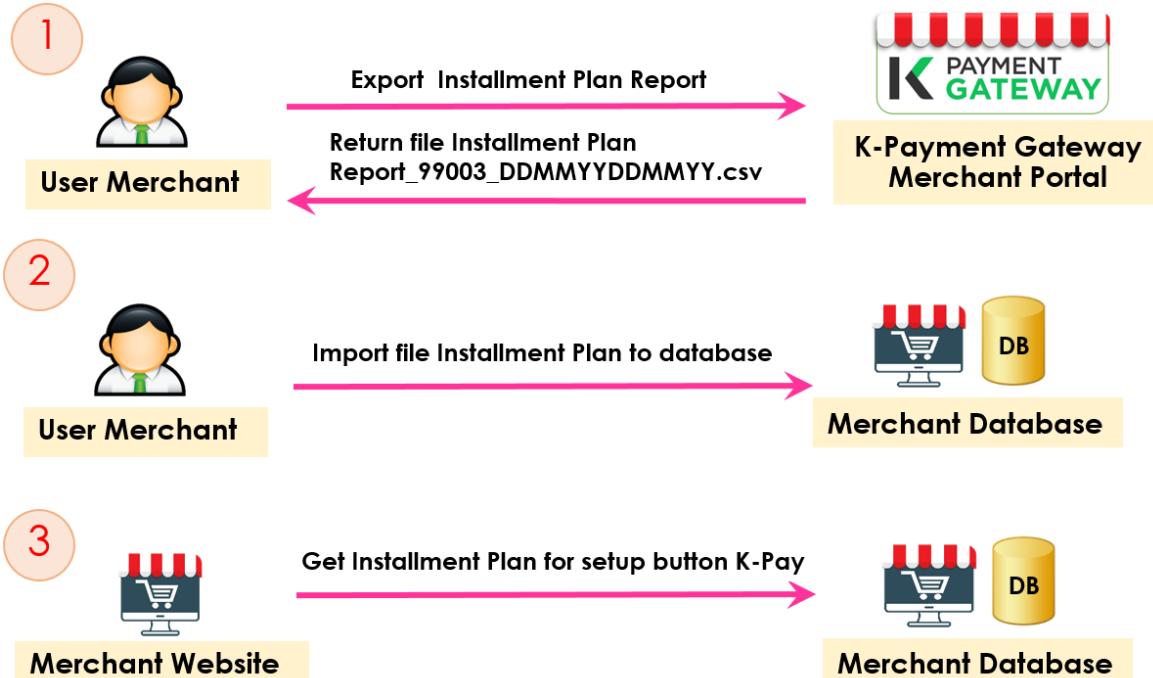
```



▪ ผ่อนชำระ (SmartPay) แบบ Installment Plan

ผ่อนชำระ (SmartPay) แบบ Installment Plan จะรองรับการผ่อนชำระประเภท ร้านค้ารับภาระดอกเบี้ย, ลูกค้ารับภาระดอกเบี้ย และ ผู้ผลิตรับภาระดอกเบี้ย โดย Set up ข้อมูล SmartPay และ ระบุ Attribute ได้ดังนี้

- ขั้นตอน Set up ข้อมูล SmartPay แบบ Installment Plan



1. ผู้ใช้งานของร้านค้าทำการลงชื่อเข้าใช้งาน Merchant Portal สำหรับบริการ K-Payment Gateway จากนั้น

ผู้ใช้งานเลือกเมนู "Report" ทำการ Create Report เลือก "Report Type" และเลือก "Installment Plan Report" ใส่คำค้นหาจากนั้นกดปุ่ม "Export" เพื่อส่งออกไฟล์ .CSV และ download ไฟล์ .CSV ชื่อไฟล์ **Installment Plan Report_SupplierNo_DDMMYYDDMMYY.csv**

▪ **Installment Plan Report Specification**

Field Name	Type	Attribute	Description
SeqNo.	Number	-	Sequence Number Sample Data: 1
PromotionName	Text (30)	-	Promotion Name
ValidFrom	Text (10)	-	Start date of promotion Format : DD/MM/YYYY (Year in Christian era) Sample Data: 25/09/2021



Field Name	Type	Attribute	Description
ValidTo	Text (10)	-	End date of promotion Format : DD/MM/YYYY (Year in Christian era) Sample Data: 31/10/2021
IPanMode	Text (2)	data-ipan-mode	IPan Mode (01, 02, 03, 04) Sample Data: 02
ProductName	Text (30)	-	Product Name Sample Data: ALL PRODUCT
SupplierNo.	Text (20)	data-supplier-no	Supplier Number Sample Data: 99902
ProductType	Number (3)	data-product-type	Product Type Sample Data: 3
ModelNo.	Text (10)	data-model-no	Model Number Sample Data: 999999
PlanID	Number (3)	data-plan-id	Plan ID Sample Data: 999
ModelName	Text (30)	-	Model Name Sample Data: Galaxy Z Flip3 5G
MinPrice	Number (9,2)	-	Minimum Price Sample Data: 100
MaxPrice	Number (9,2)	-	Maximum Price Sample Data: 999999999
PaymentTerm	Number (2)	data-term	Payment Term Sample Data: 2,3,6,12,...,72
CardType	Text (1)	-	Card Type (C,K) C: Credit Card K: K-Express Cash Sample Data: C

2. ผู้ใช้งานของร้านค้าทำการ Importไฟล์ Installment Plan Report ที่ Export มาจาก K-Payment Gateway Merchant Portal ลงใน Database ของร้านค้า



3. นำข้อมูลดังกล่าวไปใช้ในการติดตั้งปุ่ม K-Pay ตาม Attribute สำหรับผ่อนชำระ และใช้สำหรับส่งค่า Create Charge API ในส่วนของ smartpay object

- Attribute สำหรับผ่อนชำระ (SmartPay)

No	Field Name	Type	Mandatory	Description
1	data-apikey	varchar (50)	M	Your public key Sample Data: data-apikey="pkey_prod_12345"
2	data-amount	decimal (10,2)	M	Amount Sample Data: data-amount="1000.50"
3	data-payment-methods	enum (card, qr, redirect)	M	Payment method Sample Data: data-payment-methods="card"
4	data-mid	varchar (15)	M	Your merchant ID Sample Data: data-mid="401123456789001"
5	data-term	integer	O	Term Sample Data: data-term="10"
6	data-plan-id	string (3)	M	Plan ID Sample Data: data-plan-id="999"
7	data-ipan-mode	string (2)	M	IPan Mode Sample Data: data-ipan-mode="03"
8	data-product-type	string (3)	M	Product Type Sample Data: data-product-type="003"
9	data-supplier-no	string (5)	M	Supplier No Sample Data: data-supplier-no="99903"
10	data-model-no	string (6)	M	Model No Sample Data: data-model-no="123456"
11	data-name	varchar (15)	O	Your shop name Sample Data: data-name="Awesome Shop"
12	data-currency	varchar (3)	O	Currency unit (Default THB)



No	Field Name	Type	Mandatory	Description
				Sample Data: data-currency="USD"
13	data-customer-id	varchar (50)	C	Customer ID to show saved card Sample Data: data-customer-id="cust_prod_12345"
14	data-campaign-id	varchar (8)	C	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion. Sample Data: data-campaign-id="00000001"

- Sample Attribute

```

<form method="POST" action="/checkout">
<script type="text/javascript"
src="https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js"
data-apikey="pkey_test_20923L43S4RABbtwVoMlzsopFCYzyZnyMwFsn"
data-amount="1.00"
data-currency="THB"
data-payment-methods="card"
data-name="Your Shop Name"
data-mid="401202106042001"
data-plan-id="999"
data-ipan-mode="03"
data-product-type="003"
data-supplier-no="99903"
data-model-no="123456"
data-term="10" >
</script>
</form>

```



ร้านค้า/ผู้ผลิตรับการชำระโดยเบี้ย

ผู้ถือบัตรรับการชำระโดยเบี้ย

Figure 13 หน้าจอการชำระเงินแบบฟ่อนชำระ (SmartPay)

4.1.1.2 รับค่า Token จากระบบ K-Payment Gateway

ระบบ K-Payment Gateway จะรวบรวมข้อมูลการชำระเงินของลูกค้าผู้ถือบัตรอย่างปลอดภัย และส่ง Token ที่ใช้สำหรับการจ่ายเงินกลับไปให้ร้านค้า

เมื่อ K-Payment Gateway ทำการ Generate Token แล้ว ระบบจะส่งข้อมูลตอบกลับให้ร้านค้าเพื่อนำไปใช้งานในการส่ง Request ขั้นตอน Create Charge API ซึ่งมีข้อมูลดังนี้

1. Response Parameter: ชำระเต็มจำนวน (Full Payment) แบบ MCC

No	Field Name	Type	Mandatory	Description
1	token	string	M	Token ID generated from K-Payment Gateway Sample Data: token="tokn_prod_1234567890123456"
2	mid	string	M	Merchant ID Sample Data: mid="444123456789001"
3	paymentMethods	string	M	Payment method of payment Sample Data: paymentMethods="card"
4	savecard	string	C	Saved Card flag will return only when customer select "Remember card". Sample Data: savecard="true"



2. Response Parameter: ชำระเต็มจำนวน (Full Payment) แบบ DCC

No	Field Name	Type	Mandatory	Description
1	token	string	M	Token ID generated from K-Payment Gateway Sample Data: token="tokn_prod_1234567890123456"
2	dcc_currency	string	M	DCC currency that customer select to pay (for Full DCC option) Sample Data: dcc_currency="USD"
3	mid	string	M	Merchant ID Sample Data: mid="444123456789001"
4	paymentMethods	string	M	Payment method of payment Sample Data: paymentMethods="card"
5	savecard	string	C	Saved Card flag will return only when customer select "Remember card". Sample Data: savecard="true"

3. Response Parameter: ผ่อนชำระ (SmartPay)

Response Parameter แบบ SmartpayID

No	Field Name	Type	Mandatory	Description
1	token	string	M	Token ID generated from K-Payment Gateway Sample Data: token="tokn_prod_1234567890123456"
2	smartpayId	string	M	Smart Pay ID (for Installment option only) Sample Data: smartpayid="0001"
3	mid	string	M	Merchant ID Sample Data: mid="444123456789001"
4	term	string	M	Term of Payment Sample Data: term="10"
5	paymentMethods	string	M	Payment method of payment Sample Data: paymentMethods="card"



No	Field Name	Type	Mandatory	Description
6	savecard	string	C	<p>Saved Card flag will return only when customer select "Remember card".</p> <p>Sample Data: savecard="true"</p>

Response Parameter แบบ Installment Plan

No	Field Name	Type	Mandatory	Description
1	token	string	M	<p>Token ID generated from K-Payment Gateway</p> <p>Sample Data: token="tokn_prod_1234567890123456"</p>
3	mid	string	M	<p>Merchant ID</p> <p>Sample Data: mid="444123456789001"</p>
4	planid	string	C	<p>Plan ID</p> <p>Sample Data: planid ="999"</p>
5	ipanmode	string	C	<p>IPan Mode</p> <p>Sample Data: ipanmode="03"</p>
6	producttype	string	C	<p>Product Type</p> <p>Sample Data: producttype="003"</p>
7	supplierno	string	C	<p>Supplier No</p> <p>Sample Data: supplierno="99903"</p>
8	modelno	string	C	<p>Model No</p> <p>Sample Data: modelno="123456"</p>
9	term	string	M	<p>Term of Payment</p> <p>Sample Data: term="10"</p>
10	paymentMethods	string	M	<p>Payment method of payment</p> <p>Sample Data: paymentMethods="card"</p>
11	savecard	string	C	<p>Saved Card flag will return only when customer select "Remember card".</p> <p>Sample Data: savecard="true"</p>



4.1.1.3 ใช้ Token ที่ได้รับในการเรียก API เพื่อเรียกเก็บเงินให้เสร็จสมบูรณ์

!

ขั้นตอนนี้จะเกิดขึ้นหลังจากที่ร้านค้าได้รับ Token (แบบใช้ครั้งเดียว) เรียบร้อยแล้ว

การเชื่อมต่อเพื่อรับชำระเงินด้วยบัตรแบบ Full MCC, Full DCC และ SmartPay มีค่าที่จำเป็นต้องส่งตอนเรียก Create Charge API ต่างกันเล็กน้อย ดังนี้

Required Fields	Full MCC	Full DCC	SmartPay
token	✓	✓	✓
amount	✓	✓	✓
currency	✓	✓	✓
description	✓	✓	✓
source_type	✓ (ต้องระบุค่า source_type เป็น "card")	✓	✓
mode	✓ (ต้องระบุค่าใน mode เป็น "token")	✓	✓
reference_order	✓	✓	✓
dcc_data (object)	- (ต้องระบุค่าใน dcc_currency)	✓	-
additional_data (object)	✓ (ต้องระบุค่าใน mid)	✓ (ต้องระบุค่าใน mid)	✓ (ต้องระบุค่าใน mid , smartpay_id และ term*)
Smartpay (object)			✓ (*Applicable the Installment payment for Installment Plan only)

!

*Field Name "smartpay_id" and "term" : Applicable the Installment payment for SmartpayID only



4.1.1.4 วิธีการเชื่อมต่อ

การรับชำระเงินด้วยบัตรเครดิตและบัตรเดบิต (Visa, MasterCard และ JCB) โดยเชื่อมต่อแบบ Embedded UI แบ่งได้เป็น 2 รูปแบบ คือ 3D Secure และ Non 3D Secure

1. วิธีการเชื่อมต่อแบบ 3D Secure

!

ธนาคารแนะนำให้ร้านค้าสมัครใช้บริการและเชื่อมต่อด้วยวิธีนี้ เนื่องจากมีการยืนยันตัวตนผู้ถือบัตรก่อนเรียกชำระเงิน และร้านค้าจะได้รับการคุ้มครองกรณีเกิดปัญหาจากการชำระเงินออนไลน์

ร้านค้าที่เลือกสมัครแบบ Non 3D Secure จะต้องแจ้งความประสงค์และลงนามใน "หนังสือยอมรับการระความเสี่ยง บริการ K-Payment Gateway ประเภทร้านค้า Non-3D Secure"

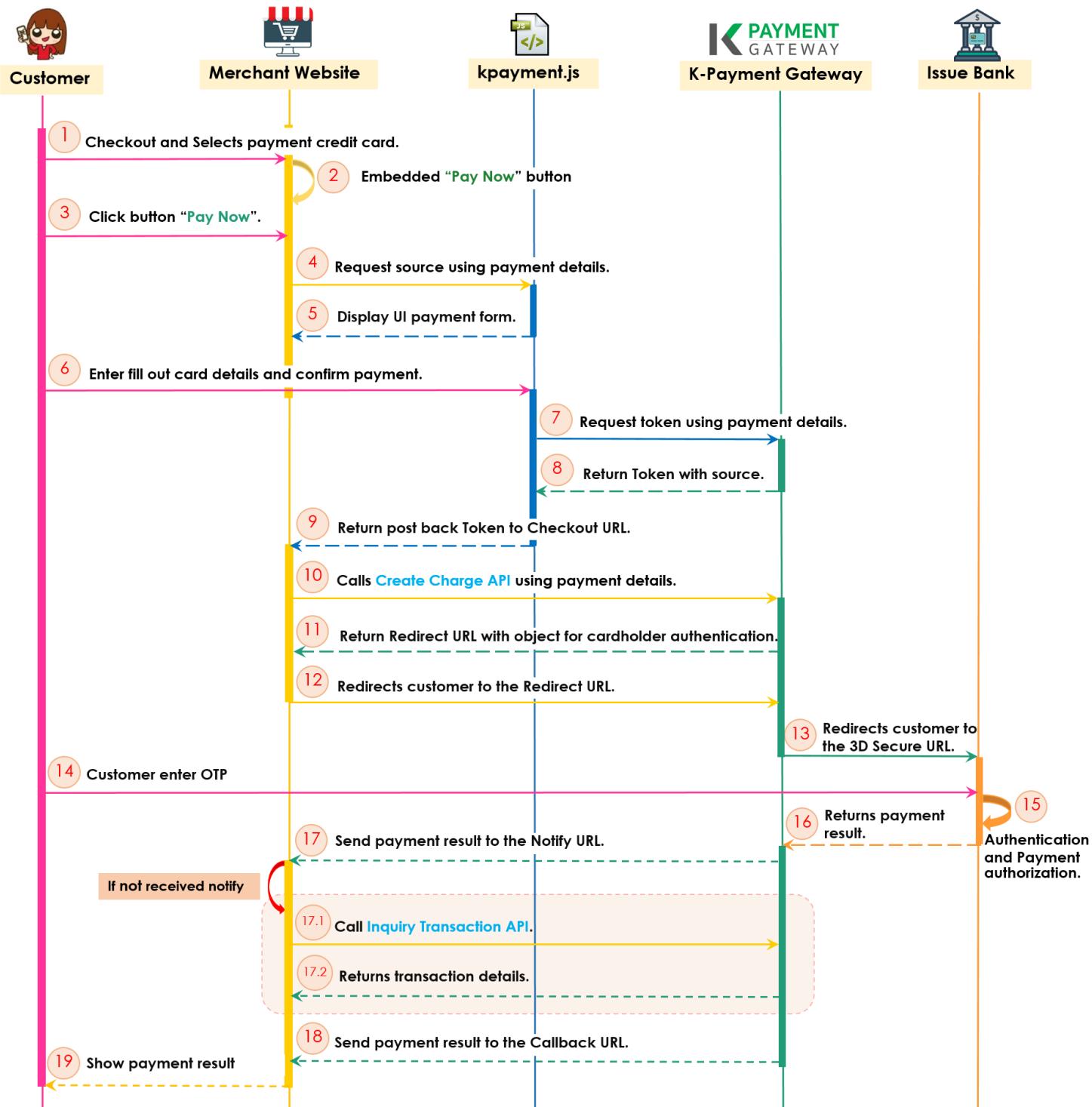


Figure 14 Embedded UI 3D Secure Flow

- ลูกค้าต้องการชำระเงินค่าและเลือกวิธีการชำระเงินประเภทบัตรเครดิต

- ร้านค้าทำการติดตั้งปุ่ม Pay Now (ดูรายละเอียด [การติดตั้งปุ่ม Pay Now](#)) ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน เพื่อเรียกหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน
- ลูกค้าคลิกปุ่ม Pay Now เพื่อทำการชำระเงิน



4. ปุ่ม Pay Now ทำการส่งคำสั่งไปยัง KPayment JavaScript File เพื่อทำการ Ganarate หน้าจอในการชำระเงิน
 5. KPayment JavaScript File แสดงหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงินเว็บไซต์หรือแอปพลิเคชัน มือถือของร้านค้า
 6. เมื่อลูกค้าใส่ข้อมูลรายละเอียดการชำระเงินเรียบร้อย และกดปุ่มยืนยันการชำระเงิน จากนั้นระบบ K-Payment Gateway จะรวบรวมข้อมูลการชำระเงินของลูกค้าอย่างปลอดภัย และทำการ Ganarate Token เพื่อใช้งานต่อไป
 7. KPayment JavaScript File ทำการร้องขอ Token จาก K-Payment Gateway
 8. K-Payment Gateway ทำการส่งข้อมูลกลับให้ KPayment javascript file พร้อมกับ Token
 9. KPayment JavaScript File ทำการส่งข้อมูลกลับให้ร้านค้าพร้อมกับ Token ที่ใช้สำหรับการชำระเงินกลับไปให้ร้านค้าใช้งานต่อไป (ดูรายละเอียดที่ [รับค่า Token จากระบบ K-Payment Gateway](#))
 10. หลังจากได้รับ Token ร้านค้าต้องเรียก Create Charge API (ดูรายละเอียดที่ [Create Charge API](#) ใน API Reference) โดยส่ง Token, รายละเอียดของรายการ และระบุตัวแปร source_type เป็น "card"
 11. K-Payment Gateway ส่งผลตอบรับจาก Charge API พร้อมกับค่า Dynamic URL ซึ่งเป็น URL สำหรับพาลูกค้าผู้ถือบัตรไปยืนยันตัวตน โดยส่งผ่านตัวแปรที่ชื่อ redirect_url
- !** ก่อนทำข้อต่อไป ร้านค้าจะต้องดำเนินการดังนี้

 - เก็บค่า Charge ID ที่ได้รับไว้ (ต้องใช้ตรวจสอบภายหลัง)
 - ตรวจสอบค่าในตัวแปร transaction_state ว่าต้องเป็น "Pre-Authorized"
 - ตรวจสอบค่าในตัวแปร status ว่าต้องเป็น "success" (รายละเอียด Transaction State และ Status อยู่ใน Appendix)
12. ร้านค้า Redirect ลูกค้าผู้ถือบัตรไปยัง Dynamic URL ที่ได้รับ
 13. K-Payment Gateway ทำการ Redirect ลูกค้าผู้ถือบัตรไปยังหน้าของธนาคารผู้ออกบัตรเพื่อยืนยันตัวตน เช่น กรอกรหัส OTP
 14. ลูกค้าผู้ถือบัตรทำการกรอก OTP
 15. ธนาคารผู้ออกบัตรทำการตรวจสอบตัวตนผู้ถือบัตรและอนุมัติการชำระเงิน
 16. ธนาคารผู้ออกบัตรส่งผลการทำการกลับมาให้ K-Payment Gateway



17. K-Payment Gateway ส่งผลการทำรายการแบบ Real-time ไปที่ Notify URL ของร้านค้า โดยร้านค้าจะต้องพัฒนาระบบ RESTful API เพื่อรองรับการเรียก WebHook Notify API for Card Payment ผ่านทาง HTTP Request แบบ POST (ดูรายละเอียดที่ [WebHook Notify API](#) ที่ API Reference)

17.1 ในกรณีที่ร้านค้าไม่ได้รับผลการทำรายการแบบ Real-time (Webhook Notify URL) ภายในระยะเวลา 10 นาที ของ Charge ID ข้างต้น ร้านค้าต้องเรียก Inquiry Transaction API เพื่อตรวจสอบรายละเอียดการทำเงินก่อนยืนยันผลการทำรายการ (ดูรายละเอียดที่ [Inquiry Transaction API](#) ใน API Reference)

17.2 K-Payment Gateway ส่งรายละเอียดการทำเงินกลับให้ร้านค้า

รายการสำเร็จ

- ค่าในตัวแปร transaction_state ต้องเป็น "Authorized"
- ค่าในตัวแปร status ต้องเป็น "success"

18. K-Payment Gateway ส่งข้อมูลรายการโดย Redirect กลับไปที่ Callback URL ของร้านค้า

ร้านค้าจะได้รับข้อมูลดังต่อไปนี้

Field Name	Data Type	Required	Description
objectId	Varchar(50)	M	Charge ID <i>Sample Data:</i> objectId="chrg_prod_12345678"
status	Varchar(15)	M	Status of this transaction (true=success, false=fail) <i>Sample Data:</i> status="true"
token	Varchar(15)	M	Token ID <i>Sample Data:</i> token="tokn_prod_123456"
saveCard	Varchar(15)	M	Flag to determine this transaction need to save card (true) or not (false) <i>Sample Data:</i> saveCard="true"



- ร้านค้าจะต้องเปรียบเทียบค่า Charge ID ที่ได้รับในขั้นตอนนี้ กับค่า Charge ID ที่เก็บไว้ (จากข้อ 2) เพื่อตรวจสอบความถูกต้องด้วย
- ผลที่ส่งให้นั้นเป็นเพียงการแจ้งเพื่อให้ร้านค้าดำเนินการขั้นตอนต่อไป ยังไม่สามารถอ้างอิงผลการตัดเงินได้

19. ร้านค้าแสดงผลการทำเงิน



2. วิธีการเข้ามต่อแบบ Non 3D Secure

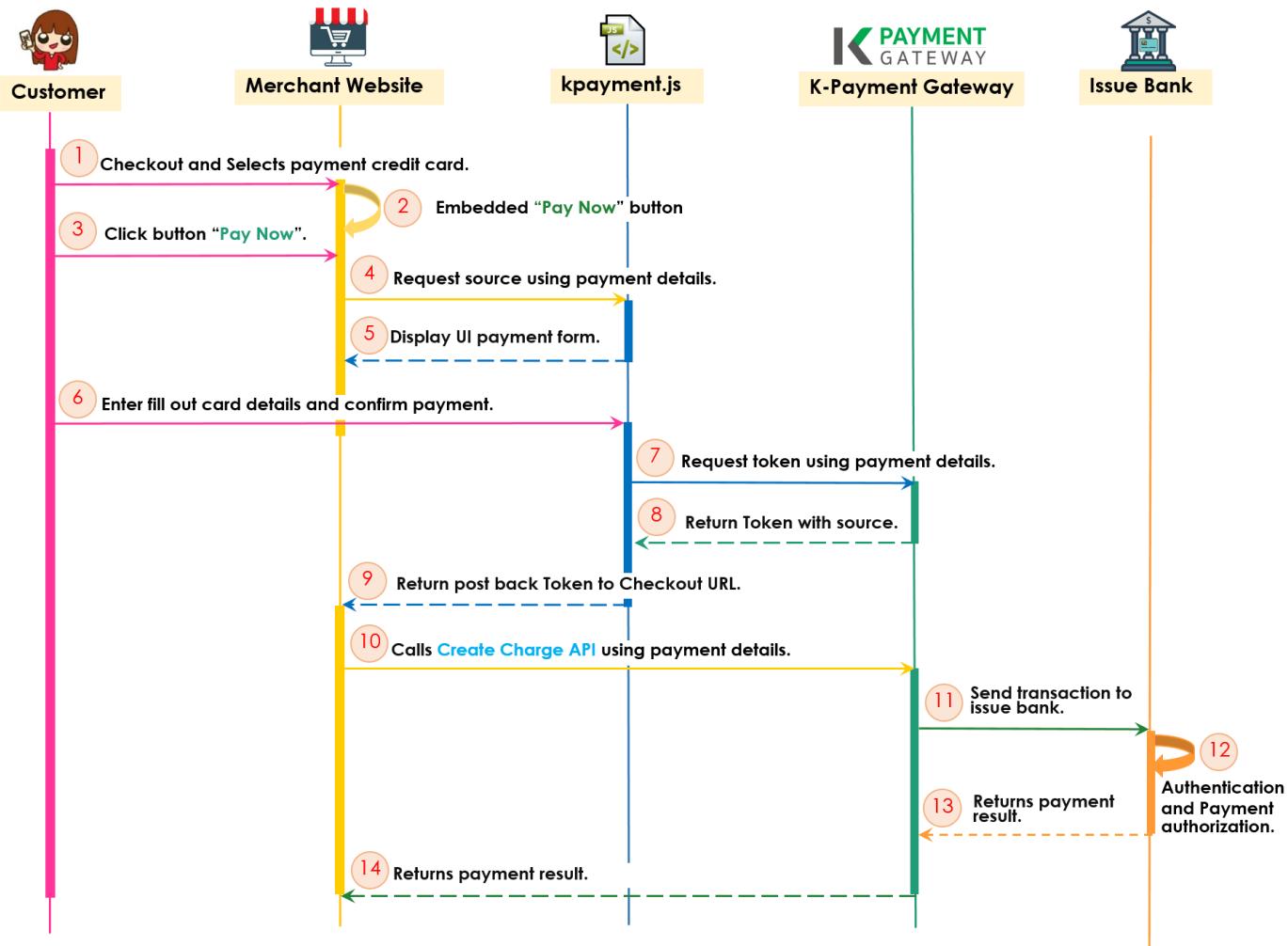


Figure 15 Embedded UI Non 3D Secure Flow

- ลูกค้าต้องการชำระสินค้าและเลือกวิธีการชำระเงินประเภทบัตรเครดิต
- ร้านค้าทำการติดตั้งปุ่ม Pay Now (ดูรายละเอียด [การติดตั้งปุ่ม Pay Now](#)) ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน เพื่อเรียกหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน
- ลูกค้าคลิกปุ่ม Pay Now เพื่อทำการชำระเงิน
- ปุ่ม Pay Now ทำการส่งคำสั่งไปยัง KPayment JavaScript File เพื่อทำการ Ganarate หน้าจอในการชำระเงิน
- KPayment javascript file แสดงหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงินในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้า



6. เมื่อลูกค้าใส่ข้อมูลรายละเอียดการชำระเงินเรียบร้อย และกดปุ่มยืนยันการชำระเงิน จากนั้นระบบ K-Payment Gateway จะรวบรวมข้อมูลการชำระเงินของลูกค้าอย่างปลอดภัย และทำการ Generate Token เพื่อใช้งานต่อไป
7. KPayment JavaScript File ทำการร้องขอ Token จาก K-Payment Gateway
8. K-Payment Gateway ทำการส่งข้อมูลกลับให้ KPayment JavaScript File พร้อมกับ Token
9. KPayment JavaScript File การส่งข้อมูลกลับให้ร้านค้าพร้อมกับ Token ที่ใช้สำหรับการจ่ายเงินกลับไปให้ร้านค้าใช้งานต่อไป
10. หลังจากได้รับ Token ร้านค้าต้องเรียก Create Charge API (ดูรายละเอียดที่ [Create Charge API](#) ใน API Reference) โดยส่ง Token, รายละเอียดของการ ชำระเงิน และระบุตัวแปร source_type เป็น "card"
11. K-Payment Gateway ส่งรายละเอียดรายการให้ธนาคารผู้ออกบัตร
12. ธนาคารผู้ออกบัตรทำการตรวจสอบรายการและอนุมัติการชำระเงิน
13. ธนาคารผู้ออกบัตรส่งผลการชำระการกลับมาให้ K-Payment Gateway
14. K-Payment Gateway ส่งผลตอบรับจาก Charge API ให้ร้านค้า

รายการสำเร็จ

- ค่าในตัวแปร transaction_state ต้องเป็น "Authorized"
- ค่าในตัวแปร status ต้องเป็น "success"



4.1.2 การเชื่อมต่อเพื่อรับชำระเงินด้วย QR Code

4.1.2.1 การติดตั้งปุ่ม K-Pay

การติดตั้งปุ่ม K-Pay เพื่อเรียกหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงินร้านค้าทำการฝังปุ่ม K-Pay

Pay Now

ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน เมื่อลูกค้าคลิกปุ่ม K-Pay จะปรากฏหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน

เพิ่ม Code ตามตัวอย่างด้านล่างนี้เข้าไปที่หน้าชำระเงิน แล้วแทนที่ด้วยข้อมูลของร้านค้า พร้อมทั้งระบุ URL สำหรับรับค่า Response กลับจากธนาคารใน action

```
<form method="POST" action="/checkout">
<script type="text/javascript"
src="https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js"
data-apikey="pkey_prod_75677dushd74774gdgd77d7dhsgfhfghfhdh"
data-amount ="13.00"
data-payment-methods="qr"
data-order-id="${order_id}" >
</script>
</form>
```

ค่า src หรือ URL สำหรับเรียก JavaScript จะแตกต่างกันระหว่างการทำรายการทดสอบบน Sandbox และการทำรายการจริง ดังนี้

- URL สำหรับเรียก JavaScript บน Sandbox
<https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js>
- URL สำหรับเรียก JavaScript เพื่อการทำรายการจริง
<https://kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js>



Attribute สำหรับชำระ QR Code

No	Field Name	Type	Mandatory	Description
1	data-apikey	Varchar (50)	M	Your public key Sample Data: data-apikey="pkey_prod_123456789"
2	data-amount	Decimal (10,2)	M	Amount Sample Data: data-amount="1000.50"
3	data-payment-methods	Enum (qr , wechat)	M	Payment method must be "qr" or "wechat" Sample Data: data-payment-methods="qr"
4	data-order-id	Varchar (50)	M	Order ID Sample Data: data-order-id="ordr_prod_12344321"
5	data-name	Varchar (15)	O	Your shop name Sample Data: data-name="Awesome Shop"
6	data-mid	Varchar (15)	O	Your merchant ID *Not require this value for Thai QR Payment Sample Data: data-mid="444123456789001"
7	data-ref-number	String (50)	O	Reference number generated by merchant site (must be unique) Sample Data: " data-ref-number =""INV000000005"



4.1.2.2 วิธีการเชื่อมต่อด้วย QR Code

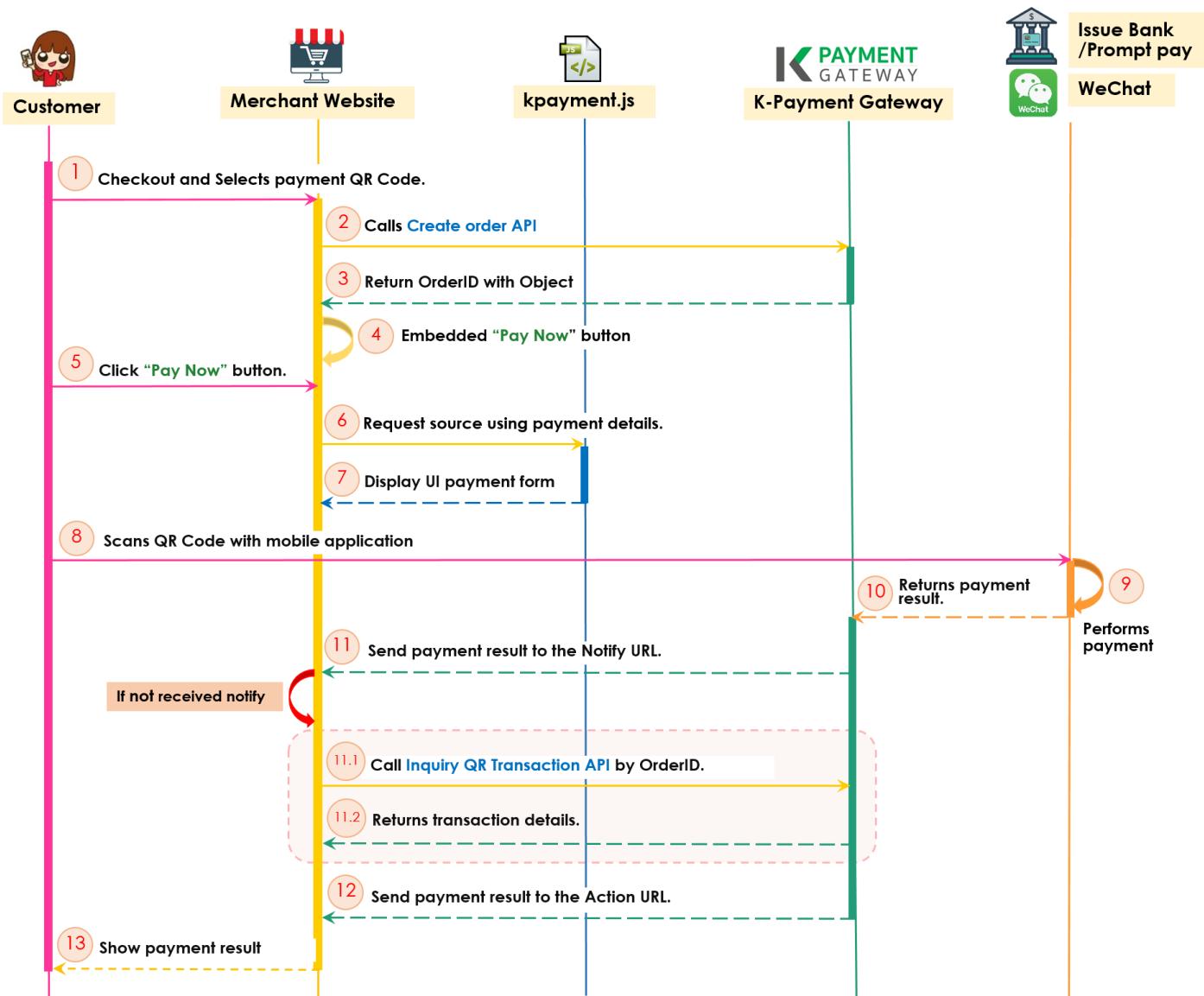


Figure 16 QR Flow

- ลูกค้าต้องการชำระเงินโดยใช้ QR Code
- ร้านค้าเรียก Create Order API (ดูรายละเอียดที่ [Order API](#) ใน API Reference) โดยระบุรายละเอียดของรายการ เช่น จำนวนเงินที่เรียกเก็บ, รายละเอียดของสินค้าหรือบริการ, เลขที่อ้างอิง เป็นต้น

Thai QR code

- ระบุตัวแปร source_type เป็น "qr".

WeChat Pay

- ระบุตัวแปร source_type เป็น "wechat".

- K-Payment Gateway ส่งผลตอบรับจาก Order API ให้ร้านค้าเก็บค่า Order ID ไว้ (ต้องใช้ในภายหลัง)



Pay Now

ເອົາໄວ້

4. ร้านค้า Redirect ลูกค้าไปยังหน้าที่ทำการ **ติดตั้งปุ่ม K-Pay**
5. ลูกค้าคลิกปุ่ม K-Pay เพื่อยืนยันการชำระเงิน
6. K-Payment Gateway จะแสดง QR Code บนหน้าจอการชำระเงิน
7. ลูกค้าใช้ Mobile Application สแกน QR Code เพื่อชำระเงิน
8. ธนาคารเจ้าของบัญชี/ผู้ให้บริการ WeChat Pay ตรวจสอบข้อมูลและอนุมัติการชำระเงิน
9. ธนาคารเจ้าของบัญชี/ผู้ให้บริการ WeChat Pay ส่งผลการทำรายการกลับมาให้ K-Payment Gateway
10. K-Payment Gateway ส่งผลการทำรายการแบบ Realtime ไปที่ Notify URL ของร้านค้า โดยร้านค้าจะต้องพัฒนาระบบ RESTful API เพื่อรับการเรียก WebHook Notify API for QR ผ่านทาง HTTP Request แบบ POST (ดูรายละเอียดที่ [WebHook Notify API](#) ใน API Reference)
 - 10.1 ในกรณีที่ร้านค้าไม่ได้รับผลการทำรายการแบบ Real-time (Webhook Notify URL) ของ Charge ID ซึ่งตั้ง ภายในระยะเวลา 10 นาที ร้านค้าต้องเรียก Inquiry QR Transaction API เพื่อตรวจสอบรายละเอียดการชำระเงินก่อนยืนยันผลการทำรายการ (ดูรายละเอียดที่ [Inquiry QR Transaction API](#) ใน API Reference)
 - 10.2 K-Payment Gateway ส่งรายละเอียดการชำระเงินกลับให้ร้านค้า

รายการสำเร็จ

 - ค่าในตัวแปร transaction_state ต้องเป็น "Authorized"
 - ค่าในตัวแปร status ต้องเป็น "success" เท่านั้น
11. K-Payment Gateway ส่งข้อมูลรายการกลับไปที่ร้านค้า (Action URL)
ร้านค้าจะได้รับข้อมูลดังต่อไปนี้

Field Name	Value
chargeId	Charge ID generated from K-Payment Gateway Sample Data: chargeId="chrg_prod_1234567890123456"

12. ร้านค้าแสดงผลการทำรายการ



4.2 Redirect

4.2.1 การเชื่อมต่อเพื่อรับชำระเงินด้วยบัตร UnionPay (TPN/UPI)

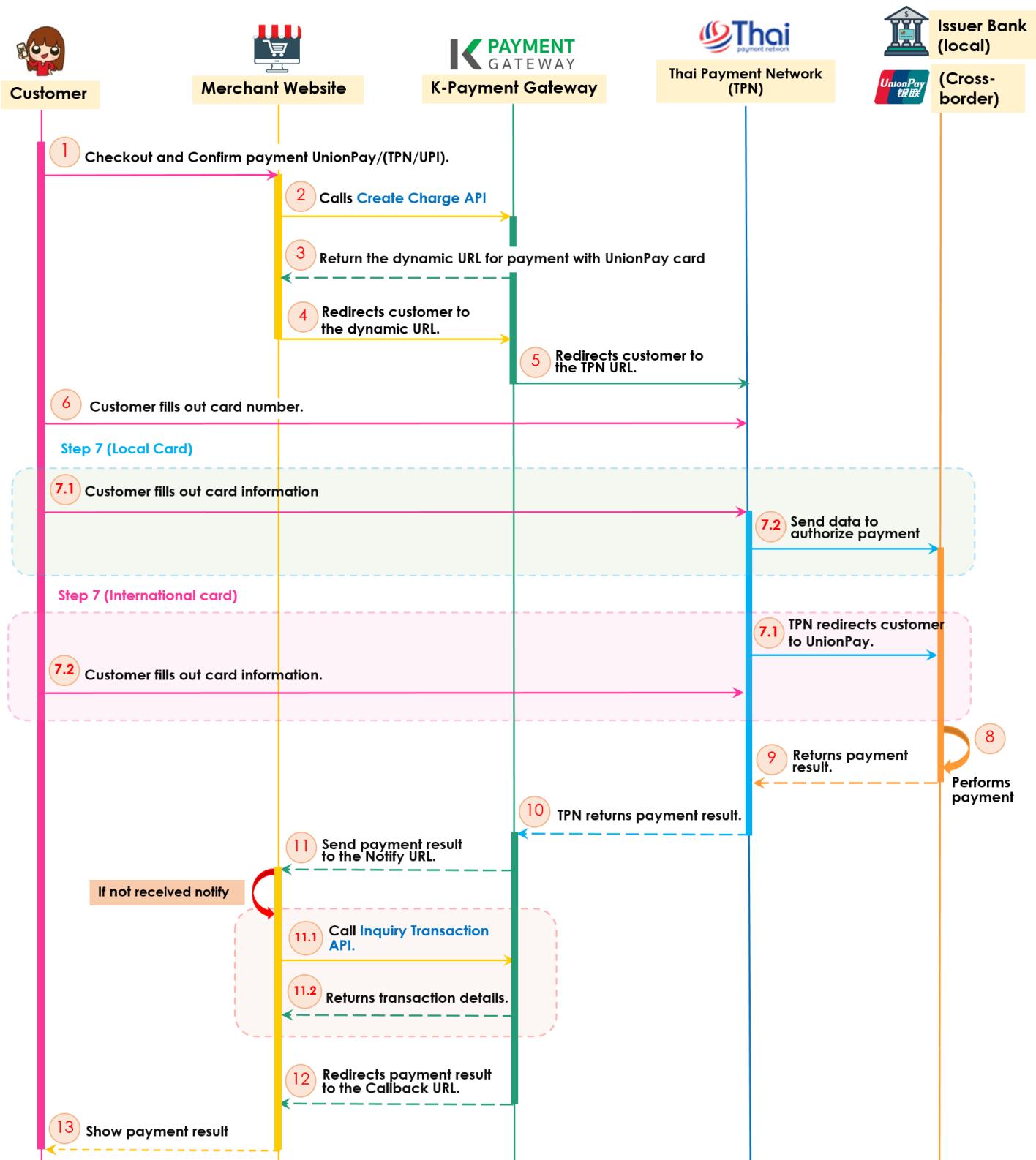


Figure 12 UnionPay (TPN/UPI) Flow



1. ลูกค้าเลือกสินค้า/บริการ และต้องการชำระเงินด้วยบัตร UnionPay
2. ร้านค้าเรียก **Create Charge API** (ดูรายละเอียดที่ [Create Charge API](#) ใน API Reference) โดยส่งรายละเอียดของรายการ และระบุตัวแปร **source_type** เป็น "unionpay"
3. K-Payment Gateway ส่งผลตอบรับจาก Charge API พร้อมกับค่า **Dynamic URL** ซึ่งเป็น URL สำหรับพากลุกค้าผู้ถือบัตรไปชำระเงิน โดยส่งผ่านตัวแปรที่ชื่อ **redirect_url**

!

ก่อนทำข้อต่อไป ร้านค้าจะต้องดำเนินการดังนี้

- เก็บค่า Charge ID ที่ได้รับไว้ (ต้องใช้ตรวจสอบภายหลัง)
- ตรวจสอบค่าในตัวแปร **transaction_state** ว่าต้องเป็น "Initialize"
- ตรวจสอบค่าในตัวแปร **status** ว่าต้องเป็น "success"

(รายละเอียด Transaction State และ Status อธิบายใน Appendix)

4. ร้านค้า Redirect ลูกค้าผู้ถือบัตรไปยัง Dynamic URL ที่ได้รับ
5. K-Payment Gateway ทำการ Redirect ลูกค้าผู้ถือบัตรไปยัง URL ของ TPN (Thai Payment Network)
6. ลูกค้าผู้ถือบัตรกรอกข้อมูลหมายเลขบัตร (บันหน้า TPN)
7. TPN จะทำการตรวจสอบหมายเลขบัตรก่อนดำเนินการต่อ โดยขั้นตอนสำหรับบัตรเดบิตภายนอกประเทศ (Local card) และบัตรเครดิต/บัตรเดบิตต่างประเทศ (International card) จะแตกต่างกัน ดังนี้

บัตรเดบิตภายนอกประเทศ

 - 7.1 ลูกค้ากรอกรายละเอียดบัตร (บันหน้า TPN)
 - 7.2 TPN ส่งรายละเอียดรายการให้ธนาคารผู้ออกบัตรตรวจสอบรายการและอนุมัติการชำระเงิน

บัตรเครดิตและบัตรเดบิตต่างประเทศ

 - 7.1 TPN Redirect ลูกค้าผู้ถือบัตรไปยัง URL ของ UnionPay
 - 7.2 ผู้ถือบัตรกรอกรายละเอียดบัตร เพื่อให้ UnionPay ตรวจสอบตัวตนและอนุมัติการชำระเงิน
8. ธนาคารผู้ออกบัตร/UnionPay ตรวจสอบรายการและอนุมัติ(หรือปฏิเสธ)การชำระเงิน
9. ธนาคารผู้ออกบัตร/UnionPay ส่งผลการทำรายการกลับไปให้ TPN
10. TPN ส่งผลการทำรายการกลับมาให้ K-Payment Gateway
11. K-Payment Gateway ส่งผลการทำรายการแบบ Real-time ไปที่ **Notify URL** ของร้านค้า โดยร้านค้าจะต้องพัฒนาระบบ RESTful API เพื่อรับการเรียก [WebHook Notify API](#) ผ่านทาง HTTP Request แบบ POST (ดูรายละเอียดที่ API Reference)
 - 11.1 ในการกรณีที่ร้านค้าไม่ได้รับผลการทำรายการแบบ Real-time (Webhook Notify URL) ของ Charge ID ข้างต้น ภายในระยะเวลา 10 นาที ร้านค้าต้องเรียก **Inquiry Transaction API** เพื่อตรวจสอบ



รายละเอียดการชำระเงินก่อนยืนยันผลการทำรายการ (ดูรายละเอียดที่ [Inquiry Transaction API](#) ใน API Reference)

11.2 K-Payment Gateway ส่งรายละเอียดการทำรายการชำระเงินกลับให้ร้านค้า

รายการสำเร็จ

- ค่าในตัวแปร transaction_state ต้องเป็น "Authorized"
- ค่าในตัวแปร status ต้องเป็น "success" เท่านั้น

12. K-Payment Gateway ส่งข้อมูลรายการโดย Redirect กลับไปที่ Callback URL ของร้านค้า

ร้านค้าจะได้รับข้อมูลดังต่อไปนี้

Field Name	Data Type	Required	Description
objectId	Varchar(50)	M	Charge ID Sample Data: objectId="chrg_prod_12345678"
status	Varchar(15)	M	Status of this transaction (true=success, false=fail) Sample Data: status="true"



ร้านค้าจะต้องเปรียบเทียบค่า Charge ID ที่ได้รับในขั้นตอนนี้ กับค่า Charge ID ที่เก็บไว้ (จากข้อ 2) เพื่อตรวจสอบความถูกต้องด้วย

ผลที่ส่งให้นั้นเป็นเพียงการแจ้งเพื่อให้ร้านค้าดำเนินการขั้นตอนต่อไป ยังไม่สามารถอ้างอิงผลการตัดเงินได้

13. ร้านค้าแสดงผลการทำรายการชำระเงิน



4.2.2 การเชื่อมต่อเพื่อรับชำระเงินผ่าน Alipay

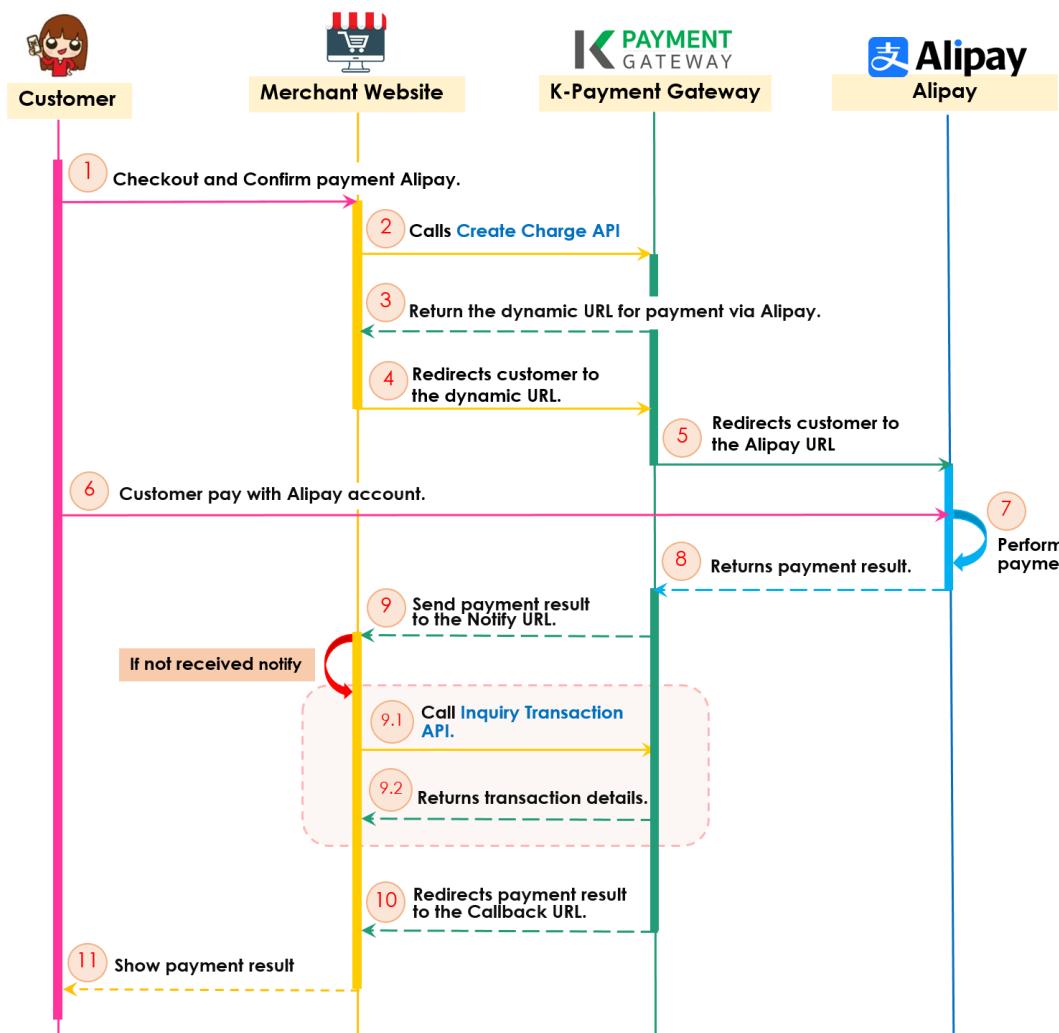


Figure 17 Alipay Flow

- ลูกค้าเลือกสินค้า/บริการ และต้องการชำระเงินด้วย Alipay
- ร้านค้าเรียก **Create Charge API** (ดูรายละเอียดที่ [Create Charge API](#) ใน API Reference) โดยส่งรายละเอียดของรายการ และระบุตัวแปร **source_type** เป็น "alipay"
- K-Payment Gateway ส่งผลตอบรับจาก Charge API พร้อมกับค่า **Dynamic URL** ซึ่งเป็น URL สำหรับพำนักผู้ถือบัตรไปชำระเงิน โดยส่งผ่านตัวแปรที่ชื่อ **redirect_url**

!

ก่อนทำข้อต่อไป ร้านค้าจะต้องดำเนินการดังนี้

- เก็บค่า Charge ID ที่ได้รับไว้ (ต้องใช้ตรวจสอบภายหลัง)
- ตรวจสอบค่าในตัวแปร **transaction_state** ว่าต้องเป็น "Initialize"
- ตรวจสอบค่าในตัวแปร **status** ว่าต้องเป็น "success"

(รายละเอียด Transaction State และ Status อุยใน Appendix)

- ร้านค้า Redirect ลูกค้าผู้ถือบัตรไปยัง Dynamic URL ที่ได้รับ



5. K-Payment Gateway ทำการ Redirect ลูกค้าผู้ถือบัตรไปยัง URL ของ Alipay
 6. ลูกค้าทำการชำระเงินด้วย Alipay E-Wallet
 7. ผู้ให้บริการ Alipay ตรวจสอบรายการและอนุมัติ(หรือปฏิเสธ)การชำระเงิน
 8. ผู้ให้บริการ Alipay ส่งผลการทำรายการกลับไปให้ K-Payment Gateway
 9. K-Payment Gateway ส่งผลการทำรายการแบบ Real-time ไปที่ Notify URL ของร้านค้า โดยร้านค้าจะต้องพัฒนาระบบ RESTful API เพื่อรับการเรียก [WebHook Notify API](#) ผ่านทาง HTTP Request แบบ POST (ดูรายละเอียดที่ API Reference)
 - 9.1 ในกรณีที่ร้านค้าไม่ได้รับผลการทำรายการแบบ Real-time (Webhook Notify URL) ของ Charge ID ข้างต้น ภายในระยะเวลา 10 นาที ร้านค้าต้องเรียก Inquiry Transaction API เพื่อตรวจสอบรายละเอียดการชำระเงินก่อนยืนยันผลการทำรายการ (ดูรายละเอียดที่ [Inquiry Transaction API](#) ใน API Reference)
 - 9.2 K-Payment Gateway ส่งรายละเอียดการทำรายการชำระเงินกลับให้ร้านค้า
- รายการสำเร็จ**
- ค่าในตัวแปร transaction_state ต้องเป็น "Authorized"
 - ค่าในตัวแปร status ต้องเป็น "success"
10. K-Payment Gateway ส่งข้อมูลรายการโดย Redirect กลับไปที่ Callback URL ของร้านค้า
ร้านค้าจะได้รับข้อมูลดังต่อไปนี้

Field Name	Data Type	Required	Description
objectId	Varchar(50)	M	Charge ID Sample Data: objectId="chrg_prod_12345678"
status	Varchar(15)	M	Status of this transaction (true=success, false=fail) Sample Data: status="true"



ร้านค้าจะต้องเบริยบเทียบค่า Charge ID ที่ได้รับในขั้นตอนนี้ กับค่า Charge ID ที่เก็บไว้ (จากข้อ 2) เพื่อตรวจสอบความถูกต้องด้วย

ผลที่ส่งให้นั้นเป็นเพียงการแจ้งเพื่อให้ร้านค้าดำเนินการขั้นตอนต่อไป ยังไม่สามารถอ้างอิงผลการตัดเงินได้

11. ร้านค้าแสดงผลการทำเงิน



5. Feature สำหรับชำระเงินด้วยบัตร Visa, MasterCard และ JCB

5.1 จดจำเลขบัตร (Remember card)

การจดจำเลขบัตร หรือ Remember card ใน K-Payment Gateway เป็น Function ที่ช่วยให้ร้านค้าอ่านวิถีความสะดวกให้ลูกค้าผู้ถือบัตรที่ต้องการจ่ายเงินด้วยบัตรใบเดิมได้โดยไม่ต้องกรอกรายละเอียดบัตรข้าหลายครั้ง ทำให้การใช้จ่ายง่ายและรวดเร็วมากยิ่งขึ้นสำหรับลูกค้าเดิม ซึ่งพัฒนาจนจดจำเลขบัตร มี 2 รูปแบบ คือ Tokenization ทำรายการชำระเงินสำเร็จก่อนบันทึกบัตร และ Register 3D จดจำหมายเลขบัตรก่อนทำการ

5.1.1 การชำระเงินครั้งแรก

1. ชำระสินค้าครั้งแรกพร้อมจดจำหมายเลขบัตรไว้ใช้จ่ายครั้งต่อไป

สิ่งที่ร้านค้าต้องทำเพิ่มก็คือ ใช้ Token ที่ได้รับในการเรียก API เพื่อเรียกเก็บเงินในการสร้าง Customer ให้ลูกค้าใหม่ที่เลือก Remember card ไว้ที่หน้าจอชำระเงินในครั้งแรก และการจ่ายเงินในครั้งต่อไปของลูกค้าเดิมที่ใช้ข้อมูล Customer ในการติดตั้งปุ่ม K-Pay และเรียก Charge API ได้เลย

The screenshot shows a test payment interface with the following details:

- Header: SHOP FOR TEST, powered by KBank, Sandbox
- Card Logos: VISA, MasterCard, JCB
- Text input: Name on Card (Test Payment)
- Text input: Card Number (4242 4242 4242 4242), VISA logo
- Text input: Expiry Date (05 / 25)
- Text input: Security Code (***)
- Checkboxes: Remember card (checked)
- Buttons: Pay Now, Pay 300.00 THB

- ลูกค้าต้องการชำระสินค้าและเลือกวิธีการชำระเงินประเภทบัตรเครดิต
- ร้านค้าทำการ ติดตั้งปุ่ม **Pay Now** (ดูรายละเอียด [การติดตั้งปุ่ม Pay Now](#)) ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน เพื่อเรียกหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน
- ลูกค้าคลิกปุ่ม Pay Now เพื่อทำการชำระเงิน
- ปุ่ม Pay Now ทำการส่งคำสั่งไปยัง KPayment JavaScript File เพื่อทำการ Ganarate หน้าจอในการชำระเงิน
- KPayment JavaScript File แสดงหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงินเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้า
- เมื่อลูกค้าใส่ข้อมูลรายละเอียดการชำระเงินเรียบร้อย และ เลือก "Remember Card" จากนั้นกดปุ่มยืนยันการชำระเงิน จากนั้นระบบ K-Payment Gateway จะรวบรวมข้อมูลการชำระเงินของลูกค้าอย่างปลอดภัย และทำการ Ganarate Token เพื่อใช้งานต่อไป

- ร้านค้าต้องทำการเชื่อมต่อ [รูปแบบการเชื่อมต่อ 3D Secure](#) ในขั้นตอนที่ 7-19 ให้เสร็จสมบูรณ์
- หลังจาก K-Payment Gateway ส่งรายละเอียดการชำระเงินกลับให้ร้านค้า เป็น [รายการสำเร็จ](#) เท่านั้น โดย
 - ค่าในตัวแปร transaction_state ต้องเป็น "Authorized"
 - ค่าในตัวแปร status ต้องเป็น "success"
- จากนั้นให้ร้านค้าดำเนินการเรียก ร้านค้าเรียก Create Customer API โดยส่ง Token และรายละเอียดลูกค้า ซึ่งได้แก่ email และ name และระบุตัวแปร mode เป็น "token" (ดูรายละเอียดที่ [Create Customer API](#) ใน API Reference)
- K-Payment Gateway จะส่งผลตอบกลับจากCustomer APIมาให้ร้านค้าเก็บค่าCustomer IDไว้สำหรับครั้งต่อไป



2. Register 3D จดจำบัตรก่อนทำการซื้อสินค้า

- ร้านค้าต้องการจดจำหมายเลขบัตรก่อนทำการซื้อสินค้า โดยใส่ Amount เท่ากับ 1
- ร้านค้าทำการ ติดตั้งปุ่ม Pay Now (ดูรายละเอียด [การติดตั้งปุ่ม Pay Now](#)) ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน เพื่อเรียกหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน
- ลูกค้าคลิกปุ่ม Pay Now เพื่อทำการชำระเงิน
- ปุ่ม Pay Now ทำการส่งคำสั่งไปยัง KPayment JavaScript File เพื่อทำการ Ganarate หน้าจอในการชำระเงิน
- KPayment JavaScript File แสดงหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงินเว็บไซต์หรือแอปพลิเคชัน

มือถือของร้านค้า

6. เมื่อลูกค้าใส่ข้อมูลรายละเอียดการชำระเงินเรียบร้อย และ **เลือก**

"Remember Card" จากนั้นกดปุ่มยืนยันการชำระเงิน จากนั้นระบบ K-

Payment Gateway จะรวบรวมข้อมูลการชำระเงินของลูกค้าอย่างปลอดภัย และทำการ Ganarate Token เพื่อใช้งานต่อไป

7. ร้านค้าต้องทำตามหัวข้อ **รูปแบบการเชื่อมต่อ 3D Secure** ในขั้นตอนที่ 7-19 ให้เสร็จสมบูรณ์

โดยขั้นตอนที่ 10 ร้านค้าต้องเรียก **Create Charge API** และระบุตัวแปร

"mode" = "register3d", "Amount" เท่ากับ 1 และหลังจาก ขั้นตอนที่ 16 หลังธนาคารผู้ออกบัตรส่งผลการทำการยกเลิกบัตรมาให้ K-Payment Gateway กรณี เป็น **รายการสำเร็จ ระบบ K-Payment Gateway จะทำการยกเลิกรายการ**

(Void) อัตโนมัติ



กรณีทำการผูกบัตรแบบ Register 3D จดจำบัตรก่อนทำการซื้อสินค้า จะไม่แสดงรายการใน Merchant Portal

- หลังจาก K-Payment Gateway ส่งรายละเอียดการชำระเงินกลับให้ร้านค้า เป็น **รายการสำเร็จ** เท่านั้น โดย

- ค่าในตัวแปร transaction_state ต้องเป็น "Voided"
- ค่าในตัวแปร status ต้องเป็น "success"

จากนั้นให้ร้านค้าดำเนินการเรียก ร้านค้าเรียก Create Customer API โดยส่ง Token และรายละเอียดลูกค้า ซึ่งได้แก่ email และ name และระบุตัวแปร mode เป็น "token" (ดูรายละเอียดที่ [Create Customer API](#) ใน API Reference)

- K-Payment Gateway จะส่งผลตอบกลับจาก Customer API มาให้ ร้านค้าเก็บค่า Customer ID ไว้สำหรับครั้งต่อไป



5.1.2 การชำระเงินครั้งต่อไป

1. รูปแบบ Embedded UI

The screenshot shows a payment interface with the following details:

- Header: SHOP FOR TEST, powered by KBank, Sandbox.
- Middle section: Please select an item.
- Card selection area:
 - VISA 4417 70** **** 5830
 - Mastercard 5404 88** **** 0836
 - Visa 3564 58** **** 6476
- Buttons: Use another card, Pay 3,000.00 THB.

- ร้านค้าทำการฝังปุ่ม **Pay Now** ลงในเว็บไซต์หรือแอปพลิเคชันมือถือ ของร้านค้าที่หน้าชำระเงิน โดยระบุค่าในตัวแปร `data-customer-id` ด้วย Customer ID ที่เก็บเอาไว้ เมื่อลูกค้าคลิกปุ่ม K-Pay จะปรากฏหน้าจอแสดงบัตรที่ลูกค้าเลือกจะนำเลขบัตรเอาไว้ (ตามภาพ)
- ระบบ K-Payment Gateway จะรวบรวมข้อมูลการชำระเงินของลูกค้าผู้ถือบัตรอย่างปลอดภัย และส่ง Token กลับไปให้ร้านค้า
- ร้านค้าใช้ Token ที่ได้รับในการเรียกเก็บเงินให้เสร็จสมบูรณ์
ขั้นตอนในการเรียก Create Charge API ต้องระบุตัวแปรดังนี้
 - ระบุตัวแปร `source_type` เป็น "card"
 - ระบุตัวแปร `mode` เป็น "token"
 - ระบุตัวแปร `token` ด้วย Token ที่ได้รับจากข้อ 2)
 - ใน Object "customer" ให้ระบุตัวแปร `customer_id` ด้วย Customer ID ที่เก็บเอาไว้
 - ส่งรายละเอียดของรายการ เช่น จำนวนเงินที่เรียกเก็บ, รายละเอียดของสินค้าหรือบริการ เป็นต้น

2. รูปแบบ Direct API

- ร้านค้าต้องเรียก Inquiry Customer API (ดูรายละเอียดที่ [Customer API](#) ใน API Reference) โดยระบุค่าในตัวแปร `customer_id` ด้วย Customer ID ที่เก็บเอาไว้ เพื่อดึงรายการบัตรที่ลูกค้าเลือกจะนำเลขบัตรเอาไว้มาแสดง
- เมื่อลูกค้าเลือกบัตรที่ต้องการและยืนยันการชำระเงิน ร้านค้าต้องเรียก Create Charge API (ดูรายละเอียดที่ [Create Charge API](#) ใน API Reference) โดยมีรายละเอียด ดังนี้
 - ระบุตัวแปร `source_type` เป็น "card"
 - ระบุตัวแปร `mode` เป็น "customer"
 - ใน Object "customer" ให้ระบุตัวแปร `customer_id` และ `card_id` ด้วย Customer ID ที่เก็บเอาไว้และ Card Object ID ที่ลูกค้าเลือก
 - ส่งรายละเอียดของรายการ เช่น จำนวนเงินที่เรียกเก็บ, รายละเอียดของสินค้าหรือบริการ เป็นต้น
- ผลการทำรายการ
 - ในกรณีที่ทำรายการ 3D Secure ร้านค้าต้องทำตามหัวข้อ [รูปแบบการเชื่อมต่อ 3D Secure](#) ในขั้นตอนที่ 11-19 ให้เสร็จสมบูรณ์ เพื่อทำการยืนยันตัวตนลูกค้าทุกครั้งก่อนตัดเงินจากบัตรเพื่อความปลอดภัย
 - ในกรณีที่ทำรายการ Non 3D Secure ร้านค้าจะได้รับผลตอบกลับจาก Charge API ทันที



6. API Reference

URL สำหรับเรียก API บนระบบจริง

<https://kpaymentgateway-services.kasikornbank.com>

URL สำหรับเรียก API เพื่อทำการทดสอบบน Sandbox

<https://dev-kpaymentgateway-services.kasikornbank.com>

!

Data Attribute Reference

M - Mandatory, C - Conditional, O - Optional

6.1. Charge API

6.1.1. Create Charge

URL	/card/v2/charge
HTTP Method	POST

Request Parameters

No	Name	Type	Mandatory	Description
1	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: 200.50
2	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
3	description	String (255)	M	Product description Sample Data: "Awesome Product"
4	source_type	Enum (card , alipay , unionpay)	M	Type to use Charge API Sample Data: "card"
5	mode	Enum (token , customer , register3D)	O	Mode to use Charge API for card payment Sample Data: "token"



6	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "2018053017560"
7	token	String (50)	O	Token ID Sample Data: "tokn_prod_12345678"
8	savecard	Boolean	C	Save new card flag for this transaction(required if savecard flag return from post token) Sample Data: "true"
9	order_id	String (100)	O	Order ID to charge Sample Data: "ordr_prod_12345678"
10	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
11	ref_2	String (128)	O	Reference 2 Sample Data: "reference_id234"
12	ref_3	String (128)	O	Reference 3 Sample Data: "reference_id345"
13	dcc_data	Object	C	DCC object data
13.1	dcc_currency	String (3)	M	DCC currency that customer select Sample Data: " USD "
Sample Data : "dcc_data": { "dcc_currency":"USD" }				
14	customer	Object	C	Customer object data
14.1	customer_id	String (256)	M	Customer ID Sample Data: "cust_prod_12345678"
14.2	card_id	String (50)	O	Card ID Sample Data: "card_prod_12345678"
Sample Data : "customer": { "customer_id": "cust_prod_12345678", "card_id": "card_prod_12345678" }				



15	additional_data	Object	C	Additional request object	
15.1	mid	String (20)	M	Merchant ID	
15.2	tid	String (20)	O	Terminal ID	
15.3	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion.	
15.4	smartpay_id	String (5)	C	Smartpay ID for Payment Installment	(*ต้องส่งค่านี้ กรณีที่ใช้ SmartpayID)
15.5	term	Integer	C	Term selected for Payment Installment	
Sample Data					
<pre>"additional_data": { "mid": "401123456789", "tid": "12345678", "smartpay_id": "0001", "term": "10", "campaign_id": "00000001" }</pre>					
16	callback_type	Enum (web , mobile)	O	<p>Callback type for 3D Secure and Redirect flow.</p> <ul style="list-style-type: none"> - For "register3D" mode, default value is "mobile". - Other modes, default value is "web". <p>Sample Data: "mobile"</p>	
17	smartpay	object	C	smartpay request object (*ต้องส่งค่านี้ กรณีที่ใช้ Installment Plan)	
17.1	plan_id	string (3)	M	<p>Plan id of installment transaction</p> <p>Sample Data: "999"</p>	
17.2	ipan_mode	string (2)	M	<p>Interest type paid by merchant or customer</p> <p>e.g. '01' paid by merchant , '02' paid by customer, '03' paid by supplier</p> <p>Sample Data: "02"</p>	



17.3	product_type	string (3)	M	The product type Sample Data: "2"
17.4	payment_term	string (2)	M	Installlment term Sample Data: "10"
17.5	supplier_no	string (5)	M	The supplier number Sample Data: "00003"
17.6	model_no	string (6)	M	The model number Sample Data: "001234"
Sample Data				
<pre>"smartpay": { "plan_id": "999", "ipan_mode": "01", "product_type": "4", "payment_term": "3", "supplier_no": "99903", "model_no": "999901" }</pre>				

Response Parameters

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "charge"
3	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"
4	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
5	transaction_state	String (50)	M	Charge transaction state <ul style="list-style-type: none"> - Initialize: Payment transaction initialized - Pre-Authorized: Payment need to do authentication 3D secure



No	Name	Type	Mandatory	Description
				<ul style="list-style-type: none"> - Authorized: Authorized success - Declined: Reject payment from host - Reversed: Payment failed from system reject <p>Sample Data: "Authorized"</p>
6	source	Object	M	The source object that was charged
6.1	id	String (50)	M	Object ID
6.2	object	String	M	Object type Sample Data: "card_prod_12345678"
6.3	brand	String (10)	M	Object brand Sample Data: "mastercard"
6.4	issuer_bank	String (56)	O	Issuer bank name Sample Data: "Kasikornbank Public Limited"
6.5	card_masking	String (16)	O	Masked card number Sample Data: "123443*****8765"
	Sample Data			
	<pre>"source": { "id": "card_prod_12345678", "object": "card", "brand": "mastercard", "issuer_bank": "Kasikornbank Public Limited", "card_masking": "424242*****4242" }</pre>			
7	created	String (17)	M	Creation date of the charge Format: YYYYMMDDHHmmSS Sample Data: "20180322121944000"
8	status	Enum (success , fail)	M	Whether the charge is authorized (success) or not (failed) Sample Data: "success"
9	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true





No	Name	Type	Mandatory	Description
15	redirect_url	String (256)	O	Dynamic URL returned for next process Sample Data: "http://kpgw.kasikornbank.com/3ds/dfesf3123123/"
16	settlement_info	Object	O	Settlement information object
17	refund_info	Object	O	Refund information object
18	approval_code	String (20)	O	Authorization's approval code from Issuer Sample Data: "123456"
19	multi_clearing	Object	O	Multi Clearing Object
20	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
21	ref_2	String (128)	O	Reference 2 Sample Data: "reference_id234"
22	ref_3	String (128)	O	Reference 3 Sample Data: "reference_id345"
23	dcc_data	Object	O	DCC Object
23.1	baht_amount	Decimal (14,4)	O	Thai baht amount
23.2	rate	Decimal (14,4)	O	Exchange rate
23.3	converted_amount	Decimal (14,4)	O	Amount base on base currency
23.4	converted_currency_code	String (3)	O	Converting currency of card number
23.5	converted_currency_name	String (128)	O	Converting currency name of card number
23.6	mid	String (22)	O	Merchant ID
	Sample Data			
	<pre>"dcc_data": { "baht_amount": 1000, "rate": 32.1, "converted_amount": 32.76, "converted_currency_code": "USD",</pre>			



No	Name	Type	Mandatory	Description
	"converted_currency_name": "USD (US Dollar)", "mid": "4529393939393" }			
24	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
25	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion. Sample Data: "00000001"
26	convfee	Object	O	Convenience Fee is an extra levied charged for the privilege of paying for a product or service which support to some specific merchant and applicable for Visa and MasterCard card payment only. *Condition: Refund will be processed by bank operation team for convenience fee transaction and only full refund is allowed.
26.1	convfee_amount	Decimal (14,4)	O	Convenience fee amount Sample Data: 2.5
26.2	convfee_vat_amount	Decimal (14,4)	O	Convenience fee vat amount Sample Data: 0.18
26.3	net_amount	Decimal (14,4)	O	Total amount include Convenience fee and vat amount Sample Data: 12.68
	Sample Data			
	"convfee": { "convfee_amount": 2.5, "convfee_vat_amount": 0.18, "net_amount": 12.68 }			



Sample Request

Sample Request for Card Payment – Visa, MasterCard & JCB

```
curl -X POST https://dev-kpaymentgateway-services.kasikornbank.com/card/v2/charge \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d'{
    "amount": 200.50,
    "currency": "THB",
    "description": "TEST PRODUCT",
    "source_type": "card",
    "mode": "token",
    "token": "tokn_prod_1fadecb16520513a076c2d97df3b0841f9",
    "reference_order": "20200530175600",
    "ref_1": "ref1",
    "ref_2": "123456",
    "additional_data": {
        "mid": "40188888888001"
    } }'
```

Sample Request for Card Payment – UnionPay (TPN/UPI)

```
curl -X POST https://dev-kpaymentgateway-services.kasikornbank.com/card/v2/charge \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d'{
    "amount": 15.00,
    "currency": "THB",
    "description": "TEST UNIONPAY",
    "source_type": "unionpay",
    "reference_order" : "20200530175601",
    "additional_data": {
        "mid": "40188888888001"
    } }'
```



Sample Request for Alipay

```
curl -X POST https://dev-kpaymentgateway-services.kasikornbank.com/card/v2/charge \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d'{
  "amount": 15.00,
  "currency": "THB",
  "description": "TEST ALIPAY",
  "source_type": "alipay",
  "reference_order" : "20200530175602"
}'
```

Sample Response

```
{
  "id": "chrg_prod_6024360e68682f247f082b7fdcdac803c2f",
  "object": "charge",
  "amount": 305,
  "currency": "THB",
  "transaction_state": "Authorized",
  "source": {
    "id": "card_prod_60248630ce8860422e9f7226644eb2772b8",
    "object": "card",
    "brand": "visa",
    "card_masking": "441770*****1827",
    "issuer_bank": "Kasikornbank Public Company Limited"
  },
  "created": "20220307212451000",
  "status": "success",
  "livemode": true,
  "metadata": {},
  "failure_code": null,
  "failure_message": null,
```



```
"description": "Testmerchant",
"mpi": {
  "eci": "05",
  "xid": "3030303030303030303030303738373936353739",
  "cavv": "0000010338075900000001053607590000000000",
  "kbank_mpi": true,
  "version": 2,
  "dtransid": "094abc81-1aed-40bf-9f9c-65e39770ae32"
},
"redirect_url": null,
"settlement_info": null,
"refund_info": null,
"approval_code": "000941",
"multi_clearing": null,
"ref_1": null,
"ref_2": null,
"ref_3": null,
"dcc_data": null,
"reference_order": "AP220307314388053",
"campaign_id": null
}
```



6.1.2. Inquiry Transaction

URL	/card/v2/charge/{charge_id reference_order order_id }/{transaction_date}
HTTP Method	GET

Path Parameters

No	Name	Type	Mandatory	Description
1	charge_id	String (100)	O	Charge ID request to inquiry information Sample Data: "chrg_prod_12345678"
2	reference_order	String (50)	O	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
3	order_id	String (100)	O	Order ID request to inquiry information Sample Data: "ordr_prod_12345678"
4	transaction_date	String (8)	O	Creation date of the charge. System will be mandatory when a merchant requests to refund more than 180 days and not more than 366. Sample Data: "20180322"

Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "charge"
3	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"
4	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
5	transaction_state	String (50)	M	Charge transaction state



No	Name	Type	Mandatory	Description
				<ul style="list-style-type: none"> - Initialize: Payment transaction initialized - Pre-Authorized: Payment need to do authentication 3D secure - Authorized: Authorized success - Declined: Reject payment from host - Reversed: Payment failed from system reject - Voided: Void transaction complete - Captured: Settle transaction success (wait for complete in next day) - Settled: Payment transaction settled - Refund Sent: Successfully refund (wait for completion the next day) - Refunded: Payment transaction refunded - Partial Refund: Some of amount refunded <p>Sample Data: "Authorized"</p>
6	source	Object	M	The source object that was charged
6.1	id	String (50)	M	Object ID
6.2	object	String	M	Object type Sample Data: "card_prod_12345678"
6.3	brand	String (10)	M	Object brand Sample Data: "mastercard"
6.4	issuer_bank	String (56)	O	Issuer bank name Sample Data: "Kasikornbank Public Limited"
6.5	card_masking	String (16)	O	Masked card number Sample Data: "123443*****8765"
	Sample Data <pre>"source": { "id": "card_prod_12345678", "object": "card", "brand": "mastercard", }</pre>			







No	Name	Type	Mandatory	Description
17.3	exchange_rate	Decimal (14,4)	O	Refund Exchange rate
17.4	orig_amount	Decimal (14,4)	O	Refund amount
17.5	orig_mdr	Decimal (14,4)	o	Refund MDR
17.6	orig_vat	Decimal (14,4)	O	Refund VAT
17.7	orig_net_amount	Decimal (14,4)	O	Refund Net Amount
17.8	refund_id	String (100)	O	Refund ID
17.9	error_code	String (20)	O	Refund error code Sample Data: "OBVAL"
17.10	error_message	String (256)	O	Refund error message Sample Data: "Insufficient Balance"
17.11	refund_date	String (17)	O	Refund date Sample Data: "20180322121944000"
	Sample Data			
	<pre>"refund_info": [{"refund_amount": 1.0, "baht_amount": 1.0, "exchange_rate": 1.0, "orig_amount": 1.0, "orig_mdr": 0.0, "orig_vat": 0.0, "orig_net_amount": 1.0, "refund_id": "refd_prod_362747fb9b55ab4e37a8017a2f02e4440", "error_code": "OBAVL", "error_message": "Insufficient Balance", "refund_date": "20180322121944000"}]</pre>			
18	approval_code	String (20)	O	Authorization's approval code from Issuer Sample Data: "123456"
19	multi_clearing	Object	O	Multi Clearing Object
20	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
21	ref_2	String (128)	O	Reference 2 Sample Data: "reference_id234"



No	Name	Type	Mandatory	Description
22	ref_3	String (128)	O	Reference 3 Sample Data: "reference_id345"
23	dcc_data	Object	O	DCC Object
23.1	baht_amount	Decimal (14,4)	O	Thai baht amount
23.2	rate	Decimal (14,4)	O	Exchange rate
23.3	converted_amount	Decimal (14,4)	O	Amount base on base currency
23.4	converted_currency_code	String (3)	O	Converting currency of card number
23.5	converted_currency_name	String (128)	O	Converting currency name of card number
23.6	mid	String (22)	O	Merchant ID
	Sample Data			
	"dcc_data": { "baht_amount": 1000, "rate": 32.1, "converted_amount": 32.76, "converted_currency_code": "USD", "converted_currency_name": "USD (US Dollar)", "mid": "4529393939393" }			
24	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
25	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion. Sample Data: "00000001"
26	convfee	Object	O	Convenience Fee is an extra levied charged for the privilege of paying for a product or service



No	Name	Type	Mandatory	Description
				which support to some specific merchant and applicable for Visa and MasterCard card payment only. *Condition: Refund will be processed by bank operation team for convenience fee transaction and only full refund is allowed.
26.1	convfee_amount	Decimal (14,4)	O	Convenience fee amount
26.2	convfee_vat_amount	Decimal (14,4)	O	Convenience fee vat amount
26.3	net_amount	Decimal (14,4)	O	Total amount include Convenience fee and vat amount
	Sample Data			
	<pre>"convfee": { "convfee_amount": 2.5, "convfee_vat_amount": 0.18, "net_amount": 12.68 }</pre>			

Sample Request

```
curl -X GET https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/charge/chrg_prod_47b66904ca59846c6be83cf444870a2f2 \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR"
```

Sample Response

```
{
  "id": "chrg_prod_208adcd781192334b1eab9c626a287cc18e",
  "object": "charge",
  "amount": 11,
  "currency": "THB",
  "transaction_state": "Refunded",
  "source": {
    "id": "card_prod_20868ec30f8460d763d6a3bc70dd35f477f",
    "object": "card",
    "brand": "mastercard",
```



```
"card_masking": "540488*****0808",
"issuer_bank": "KASIKORN BANK PUBLIC COMPANY LIMITED"
},
"created": "20190604161947000",
"status": "success",
"reference_order": "JUASU95Q4F",
"description": "Test Product",
"redirect_url": "https://uat-kpaymentgateway.com/KPGW-Auth3d-
Webapi/Authorize/auth_prod_206f7b0d74d90420b932632704b1046d1",
"settlement_info": {
    "settle_amount": 11,
    "baht_amount": 11,
    "exchange_rate": 1,
    "orig_amount": 11,
    "orig_mdr": 0.28,
    "orig_vat": 0.02,
    "orig_net_amount": 10.7
},
"refund_info": [
    {
        "settle_amount": 11,
        "baht_amount": 11,
        "exchange_rate": 1,
        "orig_amount": 11,
        "orig_mdr": 0.28,
        "orig_vat": 0.02,
        "orig_net_amount": 10.7,
        "refund_id": "refd_prod_00000012345678901234567890123456789"
    }
],
"approval_code": "123456",
"ref_1": "Ref123",
"ref_2": "123456",
"ref_3": null,
"dcc_data": null,
"livemode": true,
"metadata": {},
"failure_code": null,
"failure_message": null,
```

6.1.3. Void Transaction

URL	/card/v2/charge/{charge_id}/void
HTTP Method	POST

Path Parameter

No	Name	Type	Mandatory	Description
1	charge_id	String (100)	M	Charge ID request to void transaction Sample Data: "chrg_prod_12345678"

Request Parameters

No	Name	Type	Mandatory	Description
1	reason	String (256)	M	Reason to void transaction Sample Data: "Cancel transaction"

Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "charge"
3	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"



No	Name	Type	Mandatory	Description
4	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
5	transaction_state	String (50)	M	Charge transaction state - Authorized : Void transaction fail - Voided : Void transaction complete Sample Data: "Voided"
6	source	Object	M	The source object that was charged
6.1	id	String (50)	M	Object ID
6.2	object	String	M	Object type
6.3	brand	String (10)	M	Object brand
6.4	issuer_bank	String (56)	O	Issuer bank name
6.5	card_masking	String (16)	O	Masked card number
Sample Data				
	<pre>"source": { "id": "card_prod_12345678", "object": "card", "brand": "mastercard", "issuer_bank": "Kasikornbank Public Limited", "card_masking": "424242*****4242" }</pre>			
7	created	String (17)	M	Creation date of the charge Format: YYYYMMDDHHmmSS Sample Data: "20180322121944000"
8	status	Enum (success , fail)	M	Whether the charge is authorized or not Sample Data: "success"
9	livemode	Boolean	M	Whether this is live environment (true) or not (false) Data Type: Boolean Sample Data: "livemode": true
10	metadata	Hash	O	Metadata JSON Hash (Reserved field)



No	Name	Type	Mandatory	Description
11	failure_code	String (256)	O	Failure code returned when there is an error Sample Data: "token_invalid"
12	failure_message	String (256)	O	Failure description returned when there is an error Sample Data: "Token was expired"
13	description	String (255)	M	Product description Sample Data: "Awesome Product"
14	redirect_url	String (256)	O	Dynamic URL returned for next process
15	settlement_info	Object	O	Settlement information object
16	refund_info	Object	O	Refund information object
17	approval_code	String (20)	O	Authorization's approval code from Issuer Sample Data: "123456"
18	multi_clearing	Object	O	Multi Clearing Object
19	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
20	ref_2	String (128)	O	Reference 2 Sample Data: "reference_id234"
21	ref_3	String (128)	O	Reference 3 Sample Data: "reference_id345"
22	dcc_data	Object	O	DCC Object
22.1	baht_amount	Decimal (14,4)	O	Thai baht amount
22.2	rate	Decimal (14,4)	O	Exchange rate
22.3	converted_amount	Decimal (14,4)	O	Amount base on base currency
22.4	converted_currency_code	String (3)	O	Converting currency of card number
22.5	converted_currency_name	String (128)	O	Converting currency name of card number
22.6	mid	String (22)	O	Merchant ID
	Sample Data			
	"dcc_data": {			



No	Name	Type	Mandatory	Description
	"baht_amount": 1000, "rate": 32.1, "converted_amount": 32.76, "converted_currency_code": "USD", "converted_currency_name": "USD (US Dollar)", "mid": "4529393939393" }			
23	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
24	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion. Sample Data: "00000001"
25	convfee	Object	O	Convenience Fee is an extra levied charged for the privilege of paying for a product or service which support to some specific merchant and applicable for Visa and MasterCard card payment only. *Condition: Refund will be processed by bank operation team for convenience fee transaction and only full refund is allowed.
25.1	convfee_amount	Decimal (14,4)	O	Convenience fee amount
25.2	convfee_vat_amount	Decimal (14,4)	O	Convenience fee vat amount
25.3	net_amount	Decimal (14,4)	O	Total amount include Convenience fee and vat amount
	Sample Data			
	"convfee": { "convfee_amount": 2.5,			



No	Name	Type	Mandatory	Description
		"convfee_vat_amount": 0.18, "net_amount": 12.68 }		

Sample Request

```
curl -X POST https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/charge/chrg_prod_47b66904ca59846c6be83cf444870a2f2/void \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXgghLpRro5oAtR" \
-d'{
  "reason": "TEST VOID"
}'
```

Sample Response

```
{
  "id": "chrg_prod_208e7eb1773b5f84587bf6689644027520f",
  "object": "charge",
  "amount": 11.5,
  "currency": "THB",
  "transaction_state": "Voided",
  "source": {
    "id": "card_prod_208452304f5b6621f19b45e682f57fd17ee",
    "object": "card",
    "brand": "visa",
    "card_masking": "492141*****4242",
    "issuer_bank": "Kasikornbank Public Company Limited"
  },
  "created": "20190606134355773",
  "status": "success",
  "reference_order": "INV1234567",
```



```

"description": "TestDescription",
"redirect_url": "",
"approval_code": "123456",
"ref_1": "1",
"ref_2": "2",
"ref_3": "3",
"dcc_data": null,
"livemode": true,
"failure_code": null,
"failure_message": null,
"campaign_id": "00000001"
}

```

6.1.4. Settle Transaction

URL	/card/v2/charge/{charge_id}/settle
HTTP Method	POST

Path Parameter

No	Name	Type	Mandatory	Description
1	charge_id	String (100)	M	Charge ID request for settlement <i>Sample Data:</i> "chrg_prod_12345678"

Request Parameter

No	Name	Type	Mandatory	Description
1	amount	Decimal (12,2)	M	The settlement amount request (must be equal or less than charged amount) <i>Sample Data:</i> 200.50
2	type	Enum (full , partial)	M	Settlement type <i>Sample Data:</i> "full"



Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "charge"
3	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"
4	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
5	transaction_state	String (50)	M	Charge transaction state - Authorized : Settle transaction fail - Captured : Settle transaction success (wait for complete in next day) Sample Data: "Captured"
6	source	Object	M	The source object that was charged
6.1	id	String (50)	M	Object ID
6.2	object	String	M	Object type
6.3	brand	String (10)	M	Object brand
6.4	issuer_bank	String (56)	O	Issuer bank name
6.5	card_masking	String (16)	O	Masked card number
	Sample Data			
	<pre>"source": { "id": "card_prod_12345678", "object": "card", "brand": "mastercard", "issuer_bank": "Kasikornbank Public Limited", "card_masking": "424242*****4242" }</pre>			
7	created	String (17)	M	Creation date of the charge



No	Name	Type	Mandatory	Description
				Format: YYYYMMDDHHmmSS Sample Data: "20180322121944000"
8	status	Enum (success , fail)	M	Whether the charge is authorized (success) or not (failed) Sample Data: "success"
9	livemode	Boolean	M	Whether this is live environment (true) or not (false) Data Type: Boolean Sample Data: "livemode": true
10	metadata	Hash	O	Metadata JSON Hash (Reserved field)
11	failure_code	String (256)	O	Failure code returned when there is an error Sample Data: "token_invalid"
12	failure_message	String (256)	O	Failure description returned when there is an error Sample Data: "Token was expired"
13	description	String (255)	M	Product description Sample Data: "Awesome Product"
14	redirect_url	String (256)	O	Dynamic URL returned for next process
15	settlement_info	Object	O	Settlement information object
16	refund_info	Object	O	Refund information object
17	approval_code	String (20)	O	Authorization's approval code from Issuer Sample Data: "123456"
18	multi_clearing	Object	O	Multi Clearing Object
19	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
20	ref_2	String (128)	O	Reference 2 Sample Data: "reference_id234"
21	ref_3	String (128)	O	Reference 3 Sample Data: "reference_id345"
22	dcc_data	Object	O	DCC Object
22.1	baht_amount	Decimal (14,4)	O	Thai baht amount
22.2	rate	Decimal (14,4)	O	Exchange rate



No	Name	Type	Mandatory	Description
22.3	converted_amount	Decimal (14,4)	O	Amount base on base currency
22.4	converted_currency_code	String (3)	O	Converting currency of card number
22.5	converted_currency_name	String (128)	O	Converting currency name of card number
22.6	mid	String (22)	O	Merchant ID
	Sample Data			
	<pre>"dcc_data": { "baht_amount": 1000, "rate": 32.1, "converted_amount": 32.76, "converted_currency_code": "USD", "converted_currency_name": "USD (US Dollar)", "mid": "4529393939393" }</pre>			
23	reference_order	String (50)	M	<p>Reference number generated by merchant site (must be unique)</p> <p>Sample Data: "20180530175600"</p>
24	campaign_id	String (8)	O	<p>BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion.</p> <p>Sample Data: "00000001"</p>
25	convfee	Object	O	<p>Convenience Fee is an extra levied charged for the privilege of paying for a product or service which support to some specific merchant and applicable for Visa and MasterCard card payment only.</p> <p>*Condition: Refund will be processed by bank operation team for convenience fee transaction and only full refund is allowed.</p>
25.1	convfee_amount	Decimal (14,4)	O	Convenience fee amount



No	Name	Type	Mandatory	Description
25.2	convfee_vat_amount	Decimal (14,4)	O	Convenience fee vat amount
25.3	net_amount	Decimal (14,4)	O	Total amount include Convenience fee and vat amount
Sample Data				
<pre>"convfee": { "convfee_amount": 2.5, "convfee_vat_amount": 0.18, "net_amount": 12.68 }</pre>				

Sample Request

```
curl -X POST https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/charge/chrg_prod_47b66904ca59846c6be83cf444870a2f2/settle \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d'{
    "amount": 100.00,
    "type": "full"
}'
```

Sample Response

```
{
    "id": "chrg_prod_29f1cf15af3e554e00950ced9bdcd4b69b",
    "object": "charge",
    "amount": 5,
    "currency": "THB",
    "transaction_state": "Captured",
    "source": {
```

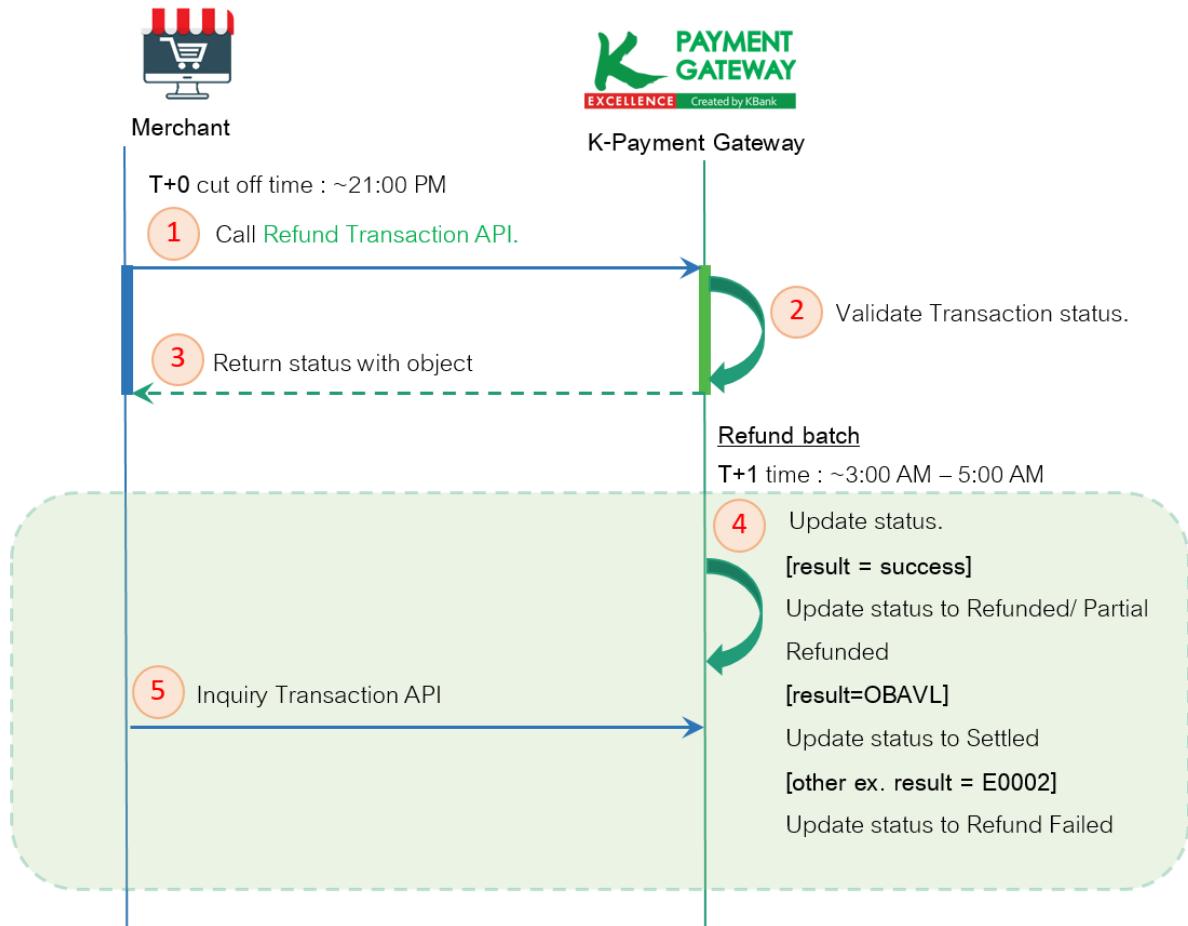


```
"id": "card_prod_29ac5ac5a548fa18e47275b29e3235437e",
  "object": "card",
  "brand": "visa",
  "card_masking": "492141*****4242",
  "issuer_bank": "Kasikornbank Public Company Limited"
},
"created": "20190606140512000",
"status": "success",
"reference_order": "INV1234567",
"description": "TestDescription",
"redirect_url": null,
"approval_code": "123456",
"ref_1": "1",
"ref_2": "2",
"ref_3": "3",
"dcc_data": null,
"livemode": true,
"failure_code": null,
"failure_message": null,
"campaign_id": "00000001"
}
```



6.1.5. Refund Transaction

The Refund API is used to generate or retrieve refund entries. The refund is related to the payment transaction. This can only be on items that charge by **credit card (Full MCC, Full DCC, Smart-Pay)** and This can only be done on items that meet the requirements.



URL	/card/v2/charge/{charge_id}/refund
HTTP Method	POST

Path Parameter

No	Name	Type	Mandatory	Description
1	charge_id	String (100)	M	Charge ID request for refund Sample Data: charge_id="chrg_prod_1234567"



Request Parameters

No	Name	Type	Mandatory	Description
1	reason	String (256)	M	Reason to refund transaction <i>Sample Data:</i> "Reject order":
2	amount	Decimal (12,2)	M	Refund amount request (must be equal or less than Settled amount) <i>Sample Data:</i> 200.50
3	transaction_date	String (8)	C	Creation date of the charge. <i>System will be mandatory when a merchant requests to refund more than 180 days and not more than 366.</i> <i>Format:</i> YYYYMMDD <i>Sample Data:</i> "20180322"

Response Parameters

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID <i>Sample Data:</i> "chrg_prod_12345678"
2	object	String	M	Object type <i>Sample Data:</i> "charge"
3	amount	Decimal (12,2)	M	The amount charged in currency unit <i>Sample Data:</i> 200.5
4	currency	String (3)	M	3 Letter ISO Currency code in upper case <i>Sample Data:</i> "THB"
5	transaction_state	String (50)	M	Charge transaction state - Settled : Refund transaction fail - Refund Sent : Refund transaction success wait for complete in next day <i>Sample Data:</i> "Refund Sent"
6	created	String (17)	M	Creation date of the charge "20180322121944000"



No	Name	Type	Mandatory	Description
7	status	Enum (success , fail)	M	Whether the charge is authorised (success) or not (failed) Sample Data: "success"
8	failure_code	String (256)	O	Failure code returned when there is an error Sample Data: "token_invalid"
9	failure_message	String (256)	O	Failure description returned when there is an error Sample Data: "Token was expired or cannot be found"
10	metadata	Hash	O	Metadata JSON Hash (Reserved field)
11	refund_id	String (100)	M	Refund ID (Unique Id for used reconcile in refund report) Sample Data: "refd_prod_208636636631881"
12	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
13	source	Object	M	The source object that was charged
13.1	id	String (50)	M	Card object ID Sample Data: "card_prod_12345678"
13.2	object	String	M	Object type Sample Data: "card"
13.3	brand	String (10)	M	Card brand Sample Data: "mastercard"
13.4	issuer_bank	String (56)	O	Card issuer bank name Sample Data: "Kasikornbank Public Limited"
13.5	card_masking	String (16)	O	Masked card number Sample Data: "123443*****8765"
14	description	String (255)	M	Product description Sample Data: "Polaroid Pronto"
15	redirect_url	String (256)	O	Authentication URL returned for 3D Secure process
16	settlement_info	Object	M	Settlement information object



No	Name	Type	Mandatory	Description
16.1	settlement_amount	Decimal (14,4)	M	Settlement amount requested
16.2	baht_amount	Decimal (14,4)	M	Settlement amount in Thai Baht unit
16.3	exchange_rate	Decimal (14,4)	M	Settlement Exchange rate
16.4	orig_amount	Decimal (14,4)	M	Settlement amount
16.5	orig_mdr	Decimal (14,4)	M	Settlement MDR
16.6	orig_vat	Decimal (14,4)	M	Settlement VAT
16.7	orig_net_amount	Decimal (14,4)	M	Settlement Net Amount
17	refund_info	Object List	O	Refund information array
17.1	refund_amount	Decimal (14,4)	O	Refund amount requested Sample Data: 100
17.2	baht_amount	Decimal (14,4)	O	Refund amount in Thai Baht unit Sample Data: 100
17.3	exchange_rate	Decimal (14,4)	O	Refund Exchange rate
17.4	orig_amount	Decimal (14,4)	O	Refund amount
17.5	orig_mdr	Decimal (14,4)	O	Refund MDR
17.6	orig_vat	Decimal (14,4)	O	Refund VAT
17.7	orig_net_amount	Decimal (14,4)	O	Refund Net Amount
17.8	error_code	String (20)	O	Refund error code Sample Data: "OBVAL"
17.9	error_message	String (256)	O	Refund error message Sample Data: "จำนวนเงินคงเหลือในบัญชีร้านค้าไม่เพียงพอต่อการคืนเงิน กรุณาตรวจสอบยอดเงินในบัญชีหรือติดต่อ K-Biz Contact Center 028888822"
17.1	refund_date	String (17)	O	Refund date Sample Data: "20180322121944000"
18	approval_code	String (20)	O	Authorization's approval code from Issuer Sample Data: "123456"
19	multi_clearing	Object	O	Multi Clearing Object
20	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
21	campaign_id	String (8)	O	Campaign ID Sample Data: "00000001"



No	Name	Type	Mandatory	Description
22	ref_2	String (128)	O	Reference 2 Sample Data: "reference_id234"
23	ref_3	String (128)	O	Reference 3 Sample Data: "reference_id345"
24	reference_order	String (50)	M	Reference number generated by merchant site Sample Data: "20180530175600"
25	dcc_data	Object	O	DCC Object
25.1	baht_amount	Decimal (14,4)	O	Thai baht amount Sample Data: 1000
25.2	rate	Decimal (14,4)	O	Exchange rate Sample Data: 32.1
25.3	converted_amount	Decimal (14,4)	O	Amount base on base currency Sample Data: 32.76
25.4	converted_currency_code	String (3)	O	Converting currency of credit card number Sample Data: "USD"
25.5	converted_currency_name	String (128)	O	Converting currency name of credit card number Sample Data: "USD (US Dollar)"
25.6	mid	String (22)	O	Merchant ID Sample Data: "453939393939393"

Sample Request

```
curl -X POST https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/charge/chrg_prod_208adcd781192334b1eab9c626a28/refund \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d'{
    "amount": 11,
    "reason": "wrong product spec",
    "transaction_date": "20180322"
}'
```



Sample Response Success

```
{  
  "id": "chrg_prod_2082d5d99e4bb90422fa3fcfe2ba3714234",  
  "object": "charge",  
  "amount": 60.0,  
  "currency": "THB",  
  "transaction_state": "Refund Sent",  
  "created": "20210920175324000",  
  "status": "success",  
  "reason": null,  
  "failure_code": null,  
  "failure_message": null,  
  "metadata": {},  
  "refund_id": "refd_prod_2081713a23c14b5033adb074b5464fb6c66",  
  "livemode": true,  
  "source": {  
    "id": "card_prod_2084f1317b8f8b1f57d3607621e00f79368",  
    "object": "card",  
    "brand": "visa",  
    "card_masking": "441770*****0915",  
    "issuer_bank": "Kasikornbank Public Company Limited"  
  },  
  "description": "SmartpayTest",  
  "redirect_url": null,  
  "settlement_info": {  
    "settle_amount": 92.2,  
    "baht_amount": 92.2,  
    "exchange_rate": 1.0,  
    "orig_amount": 0.0,  
    "orig_mdr": 0.0,  
    "orig_vat": 0.0,  
    "orig_net_amount": 0.0  
  },  
  "refund_info": null,  
  "approval_code": null,  
  "multi_clearing": null,  
  "ref_1": null,  
  "campaign_id": null,  
  "ref_2": null,  
  "ref_3": null,  
  "dcc_data": null,  
}
```



```
"reference_order": "SmartpayTest202109004" }
```

Sample Response Error

```
{
  "object": "error",
  "code": "amount_exceed",
  "message": "Refund amount cannot exceed the available limit"
}
```

6.2. Customer API

6.2.1. Create Customer

URL	/card/v2/customer
HTTP Method	POST

Request Parameter

No	Name	Type	Mandatory	Description
1	email	String (50)	M	Customer's email (used as key reference) Sample Data: "test@kpgw.com"
2	name	String (256)	M	Customer's name Data Type: String (256) Sample Data: "Mr.Somchai Jaidee"
3	mode	Enum (token)	M	Mode to use Create Customer API Sample Data: "token"
4	token	String (50)	O	Token ID that request to attach to new customer object Sample Data: "tokn_prod_12345678"



Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Customer ID Sample Data: "cust_prod_12345678"
2	object	String	M	Object type Sample Data: "object": "customer"
3	email	String (50)	M	Customer's email Sample Data: "testcard@gmail.com"
4	name	String (256)	M	Customer's name Sample Data: "Mr.Somchai Jaidee"
5	created	String (17)	M	Creation date of the customer Sample Data: "20180322121944000"
6	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
7	default_card	String (50)	M	Default Card ID of this customer Sample Data: "card_prod_12345678"
8	cards	List Object	M	The card object that attached with this customer
8.1	object	String	M	Object type
8.2	id	String (50)	M	Card object ID
8.3	brand	String (10)	M	Card brand
8.4	name	String (100)	M	The card holder name
8.5	expired	String (17)	M	Expiry time to store card for this customer
8.6	card_masking	String (16)	M	Masked card number
	Sample Data			
	<pre>"cards": [{" "id": "card_prod_12345678", "object": "card", "name": "SOMRAK T", "brand": "MASTERCARD",</pre>			



No	Name	Type	Mandatory	Description
	"card_masking": "123443*****8765", "expired": "20180905123652678" }]			

Sample Request

```
curl -X POST https://dev-kpaymentgateway-services.kasikornbank.com/card/v2/customer \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d '{
  "email" : "somrak@gmail.com",
  "name" : "Somrak Raktrakoon",
  "mode" : "token",
  "token" : "token_prod_537f4427e6fa69c3cdb9dd6545b380196a"
}'
```

Sample Response

```
{
  "id": "cust_prod_8630ebd57bc243e2aa696729fc540bc",
  "object": "customer",
  "email": "somrak@gmail.com",
  "name": "Somrak Raktrakoon",
  "created": "20180322141833391",
  "livemode": true,
  "default_card": "card_prod_0e72e4e4495a45b68f94a2f298f525fc",
  "cards": [
    {
      "id": "card_prod_0e72e4e4495a45b68f94a2f298f525fc",
      "object": "card",
      "brand": "MASTERCARD",
      "name": " Somrak Raktrakoon",
```



```

"card_masking": "514950*****9015",
"expired": "20220421000000000"
}
]}
]
}

```

6.2.2. Inquiry Customer

URL	/card/v2/customer/{customer_id}
HTTP Method	GET

Path Parameter

No	Name	Type	Mandatory	Description
1	customer_id	String (100)	M	Customer ID request to delete card Sample Data: "cust_prod_12345678"

Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Customer ID Sample Data: "cust_prod_12345678"
2	object	String	M	Object type Sample Data: "customer"
3	email	String (50)	M	Customer's email Sample Data: "testcard@gmail.com"
4	name	String (256)	M	Customer's name Sample Data: "Mr.Somchai Jaidee"
5	default_card	String (50)	M	Default Card ID of this customer Sample Data: "card_prod_12345678"
6	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
7	cards	List Object	M	The card object that attached with this customer
7.1	id	String (50)	M	Card object ID

No	Name	Type	Mandatory	Description
7.2	brand	String (10)	M	Card brand
7.3	name	String (100)	M	Card name
7.4	number	String (16)	M	Masked card number
7.5	expired	String (17)	M	Expiry time to store card for this customer
	Sample Data			
	<pre>"cards": ["id": "card_prod_12345678", "name": "Somchai J.", "brand": "MASTERCARD", "number": "123443*****8765", "expired": "20190626000000"]</pre>			

Sample Request

```
curl -X GET https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/customer/cust_prod_12345678 \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR"
```

Sample Response

```
{
  "id": "cust_prod_1234568",
  "object": "customer",
  "email": "somrak@gmail.com",
  "name": "Somrak Rakdee",
  "created": "20180322142601775",
  "livemode": true,
  "default_card": "card_prod_12345678",
  "cards": [
    {
      "id": "card_prod_12345678",
      "name": "Somchai J.",
      "brand": "MASTERCARD",
      "number": "123443*****8765",
      "expired": "20190626000000"
    }
  ]
}
```



```

"id": "card_prod_12345678",
"brand": "MASTERCARD",
"name": "SOMRAK T.",
"number": "514950*****9015",
"expired": "202204210000000000"
},
{
"id": "card_prod_87654321",
"brand": "VISA",
"name": "SOMRAK TASANA",
"number": "441770*****2327",
"expired": "202204210000000000"
}
]
}

```

6.2.3. Update Customer

URL	/card/v2/customer/{customer_id}
HTTP Method	POST

Path Parameter

No	Name	Type	Mandatory	Description
1	customer_id	String (100)	M	Customer ID request to update Sample Data: "cust_prod_12345678"

Request Parameter

No	Name	Type	Mandatory	Description
1	email	String (50)	M	Customer's email (used as key reference) Sample Data: "testcard@gmail.com"



No	Name	Type	Mandatory	Description
2	name	String (256)	O	Customer's name Sample Data: "Mr.Somchai Jaidee"
3	default_card	String (50)	O	Default Card ID of this customer Sample Data: "card_prod_12345678"

Response Attributes

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Customer ID Sample Data: "cust_prod_12345678"
2	object	String	M	Object type Sample Data: "customer"
3	email	String (50)	M	Customer's email Sample Data: "testcard@gmail.com"
4	name	String (256)	M	Customer's name Sample Data: "Mr.Somchai Jaidee"
5	default_card	String (50)	M	Default Card ID of this customer Sample Data: "card_prod_12345678"
6	livemode	Boolean	M	Whether this is live environment or not Data Type: Boolean Sample Data: true
7	cards	List Object	M	The card object that attached with this customer
7.1	id	String (50)	M	Card object ID
7.2	brand	String (10)	M	Card brand
7.3	name	String (100)	M	Card name
7.4	number	String (16)	M	Masked card number
7.5	expired	String (17)	M	Expiry time to store card for this customer
	Sample Data			
	"cards": [{"			



No	Name	Type	Mandatory	Description
	"id": "card_prod_12345678", "brand": "MASTERCARD", "name": "Somchai J.", "number": "123443*****8765", "expired": "20220626000000" }]			

Sample Request

```
curl -X PATCH https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/customer/cust_prod_8630ebd57bc243e2aa696729fcd540bc \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d '{
    "email" : "somrak@gmail.com",
    "name" : "somrak rakdee"
}'
```

Sample Response

```
{
  "id": "cust_prod_8630ebd57bc243e2aa696729fcd540bc",
  "object": "customer",
  "email": "somrak@gmail.com",
  "name": "Somrak Raktrakoon",
  "created": "20180322141833391",
  "livemode": true,
  "default_card": "card_prod_0e72e4e4495a45b68f94a2f298f525fc",
  "cards": [
    {
      "id": "card_prod_32b568415c6a42a9b0ff952a9193d465",
      "brand": "VISA",
      "name": "TEST VISA",
      "number": "441770*****0915",
    }
  ]
}
```



```
"expired": "20200831000000000" }]
```

```
}
```

6.2.4 Delete Customer

URL	/card/v2/customer/{customer_id}
HTTP Method	DELETE

Path Parameter

No	Name	Type	Mandatory	Description
1	customer_id	String (100)	M	Customer ID request to update Sample Data: "cust_prod_12345678"

Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Customer ID Sample Data: "cust_prod_12345678"
2	object	String	M	Object type Sample Data: "customer"
3	deleted	Boolean	M	Whether this customer is deleted (true) or not (false) Sample Data: true
4	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true

Sample Request

```
curl -X DELETE https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/customer/cust_prod_8630ebd57bc243e2aa696729fc540bc \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR"
```



Sample Response

```
{
  "id": "cust_prod_8630ebd57bc243e2aa696729fc540bc",
  "object": "customer",
  "deleted": true,
  "livemode": true
}
```

6.2.5 Add New Card to Customer

URL	/card/v2/customer/{customer_id}/card
HTTP Method	POST

Path Parameter

No	Name	Type	Mandatory	Description
1	customer_id	String (100)	M	Customer ID request to update Sample Data: "cust_prod_12345678"

Request Attributes

No	Name	Type	Mandatory	Description
1	mode	Enum	M	Mode to use Add New Card to Customer API Enum ('token') Sample Data: "token"
2	token	String (50)	O	Token ID that request to attach to new customer object Sample Data: "tokn_prod_12345678"



Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Customer ID Data Type: String (100) Sample Data: "id": "cust_prod_12345678"
2	object	String	M	Object type Data Type: String Sample Data: "object": "customer"
3	email	String (50)	M	Customer's email Data Type: String (50) Sample Data: "email": "test@kpgw.com"
4	name	String (256)	M	Customer's name Data Type: String (256) Sample Data: "name": "Mr.Somchai Jaidee"
5	default_card	String (50)	M	Default Card ID of this customer Data Type: String (50) Sample Data: "default_card": "card_prod_12345678"
6	livemode	Boolean	M	Whether this is live environment or not Data Type: Boolean Sample Data: "livemode": true
7	cards	List Object	M	The card object that attached with this customer
7.1	id	String (50)	M	Card object ID
7.2	name	String (100)	M	Card name
7.3	brand	String (10)	M	Card brand
7.4	number	String (16)	M	Masked card number
7.5	expired	String (17)	M	Expiry time to store card for this customer
	Sample Data "cards": [{			



No	Name	Type	Mandatory	Description
	<pre>"id": "card_prod_12345678", "name": "Somchai J.", "brand": "MASTERCARD", "number": "123443*****8765", "expired": "20220626000000" }</pre>			

Sample Request

```
curl -X POST https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/customer/cust_prod_8630ebd57bc243e2aa696729fcd540bc/card \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d '{
    "mode" : "token",
    "token" : "token_prod_537f4427e6fa69c3cdb9dd6545b380496b"
}'
```

Sample Response

```
{
  "id": "cust_prod_8630ebd57bc243e2aa696729fcd540bc",
  "object": "customer",
  "email": "somrak@gmail.com",
  "name": "somrak rakdee",
  "created": "20180322142601775",
  "livemode": true,
  "default_card": "card_prod_0e72e4e4495a45b68f94a2f298f525fc",
  "cards": [
    {
      "id": "card_prod_12345678",
      "name": "Somchai J.",
      "brand": "MASTERCARD",
      "number": "123443*****8765",
      "expired": "20220626000000"
    }
  ]
}
```



```

"id": "card_prod_0e72e4e4495a45b68f94a2f298f525fc",
"brand": "MASTERCARD",
"name": "SOMRAK T.",
"number": "514950*****9015",
"expired": "202204210000000000"
},
{
"id": "card_prod_0e72e4e4495a45b68f94a2f298f5245fg",
"brand": "VISA",
"name": "SOMRAK TASANA",
"number": "441770*****2327",
"expired": "202104210000000000"
}
]
}

```

6.2.6 Delete Card from Customer

URL	/card/v2/customer/{customer_id}/card/{card_id}
HTTP Method	DELETE

Path Parameters

No	Name	Type	Mandatory	Description
1	customer_id	String (256)	M	Customer ID request to delete card <i>Sample Data:</i> "cust_prod_12345678"
2	card_id	String (50)	M	Card ID request to be deleted <i>Sample Data:</i> "card_prod_12345678"



Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (256)	M	Customer ID Sample Data: "cust_prod_12345678"
2	object	String	M	(Mandatory) Object type Data Type: String Sample Data: "customer"
3	deleted	Boolean	M	(Mandatory) Whether this customer is deleted (true) or not (false) Sample Data: true
4	livemode	Boolean	M	(Mandatory) Whether this is live environment (true) or not (false) Sample Data: true

Sample Request

```
curl -X DELETE https://dev-kpaymentgateway-
services.kasikornbank.com/card/v2/customer/cust_prod_8630ebd57bc243e2aa696729fc540bc
/card/card_prod_0e72e4e4495a45b68f94a2f298f525fc \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR"
```

Sample Response

```
{
  "id": "card_prod_285ed8e40e474efcb6803d82cadd6f96",
  "object": "card",
  "livemode": true,
  "deleted": true
}
```



6.3. Order API

6.3.1 Create Order

URL	/qr/v2/order
HTTP Method	POST

Request Parameter

No	Name	Type	Mandatory	Description
1	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: 200.50
2	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
3	description	String (255)	M	Product description Sample Data: "Awesome Product"
4	source_type	Enum (card , qr , redirect , wechat)	M	The string "qr" or "wechat" Sample Data: "qr"
5	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "INV000000005"
6	additional_data	Object	O	Additional request object
6.1	mid	String (20)	O	Merchant ID
6.2	tid	String (20)	O	Terminal ID
6.3	smartpay_id	String (5)	O	Smartpay ID for SmartPay mode
6.4	term	Integer	O	Term selected for SmartPay mode
6.5	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion.
Sample Data				
"additional_data": {				



No	Name	Type	Mandatory	Description
	"mid": "401123456789", "tid": "7063773", "smartpay_id": "888888", "term": "10", "campaign_id": "00000001" }			
7	customer	Object	O	Customer object data
7.1	customer_id	String (256)	M	Customer ID
Sample Data				
	"customer": { "customer_id": "cust_prod_12345678" }			
8	metadata	List Object (255)	O	Additional request object
8.1	item	String (255)	O	Product item
8.2	qty	Integer	O	Product item quantity
8.3	amount	Decimal	O	The amount charged in currency unit
Sample Data				
	"metadata": ["item": "paper", "qty": "6", "amount": "10"]			
9	ref_1	String (128)	O	Reference 1 generated by merchant site Sample Data: "0888888888"
10	ref_2	String (128)	O	Reference 2 generated by merchant site Sample Data: "202001010001"
11	ref_3	String (128)	O	Reference 3 generated by merchant site Sample Data: "000000000001"



No	Name	Type	Mandatory	Description
12	info1	String (128)	O	Information 1 Sample Data: "Product Info1"
13	info2	String (128)	O	Information 2 Sample Data: "Product Info2"
14	info3	String (128)	O	Information 3 Sample Data: "Product Info3"
15	shop_image_url	String (128)	O	Shop Image Url for show shop picture in KPLUS Mobile App

Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Order ID Sample Data: "order_prod_12345678"
2	object	String	M	Object type Sample Data: "order"
3	created	String (17)	M	Creation date of the order Sample Data: "20180322121944000"
4	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
5	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: 200.50
6	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
7	customer	Object	O	Customer object data
7.1	customer_id	String (256)	O	Customer ID
	Sample Data			
	"customer": { "customer_id": "cust_prod_12345678" }			



No	Name	Type	Mandatory	Description
8	description	String (255)	M	Product description Sample Data: "Awesome Product"
9	metadata	List Object (255)	O	Additional request object
9.1	item	String (255)	O	Product item
9.2	qty	Integer	O	Product item quantity
9.3	amount	Decimal	O	The amount charged in currency unit
	Sample Data			
	<pre>"metadata": ["item": "paper", "qty": "6", "amount": "10"]</pre>			
10	status	Enum (success , fail)	M	Whether the order is created (success) or not (failed) Sample Data: "success"
11	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "INV000000005"
12	source_type	Enum (card , qr)	M	The string "card" or "qr" Sample Data: "qr"
13	additional_data	Object	O	Additional request object
13.1	term	Integer	O	Term selected for SmartPay mode
13.2	mid	String (20)	O	Merchant ID
13.3	tid	String (20)	O	Terminal ID
13.4	smartpay_id	String (5)	O	Smartpay ID for SmartPay mode
13.5	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion.
	Sample Data			



No	Name	Type	Mandatory	Description
	"additional_data": { "term": null, "mid": null, "tid": null, "smartpay_id": null, "campaign_id": null }			
14	failure_code	String (256)	O	Failure code returned when there is an error
15	failure_message	String (256)	O	Failure description returned when there is an error
16	expire_time_seconds	Integer	M	Order expire time Sample Data: " 3600"

Sample Request

```
curl -X POST https://dev-kpaymentgateway-services.kasikornbank.com/qr/v2/order \
-H "Content-Type: application/json" \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR" \
-d'{  
    "amount": 200,  
    "currency": "THB",  
    "description": "TESTPRODUCT",  
    "source_type": "qr",  
    "reference_order": "WT1234567" }'
```

Sample Response

```
{  
    "id": "order_prod_12345678",  
    "object": "order",  
    "amount": 200,
```



```

"currency": "THB",
"reference_order": "WT1234567",
"source_type": "card",
"description": "TESTPRODUCT",
"status": "success",
"created": "20180625121550200",
"livemode": true,
"failure_code": null,
"failure_message": null,
"expire_time_seconds": 3600
}

```

6.3.2 Inquiry Order

URL	/qr/v2/order/{order_id}
HTTP Method	GET

Path Parameter

No	Name	Type	Mandatory	Description
1	order_id	String (100)	M	Order ID request to inquiry information Sample Data: "ordr_prod_12345678"

Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Order ID Sample Data: "order_prod_12345678"
2	object	String	M	Object type Sample Data: "order"
3	created	String (17)	M	Creation date of the order Sample Data: "20180322121944000"



No	Name	Type	Mandatory	Description
4	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
5	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"
6	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
7	description	String (255)	M	Product description Sample Data: "Awesome Product"
8	customer	Object	O	Customer object data
8.1	customer_id	String (256)	O	Customer ID
Sample Data				
"customer": { "customer_id": "cust_prod_12345678" }				
9	metadata	List Object (255)	O	Additional request object
9.1	item	String (255)	O	Product item
9.2	qty	Integer	O	Product item quantity
9.3	amount	Decimal	O	The amount charged in currency unit
Sample Data				
"metadata": [{ "item": "paper", "qty": "6", "amount": "10" }]				
10	status	Enum (success , fail)	M	Whether the order is created (success) or not (failed) Sample Data: "success"
11	source_type	Enum (card , qr)	M	The string "card" or "qr" Sample Data: "qr"
12	additional_data	Object	O	Additional request object
12.1	mid	String (20)	O	Merchant ID



No	Name	Type	Mandatory	Description
12.2	tid	String (20)	O	Terminal ID
12.3	smartpay_id	String (5)	O	Smartpay ID for SmartPay mode
12.4	term	Integer	O	Term selected for SmartPay mode
12.5	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion.
Sample Data				
<pre>"additional_data": { "mid": "401123456789", "tid": "7063773", "smartpay_id": "888888", "term": "10", "campaign_id": "00000001" }</pre>				
13	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "INV00000005"
14	failure_code	String (256)	O	Failure code returned when there is an error
15	failure_message	String (256)	O	Failure description returned when there is an error
16	expire_time_seconds	Integer	M	Order expire time Sample Data: 3600

Sample Request

```
curl -X GET https://dev-kpaymentgateway-services.kasikornbank.com/qr/v2/order/order_prod_12345678 \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR"
```

Sample Response

```
{
  "id": "order_prod_12345678",
  "object": "order",
  "amount": 200,
  "currency": "THB",
  "reference_order": "WT1234567",
  "source_type": "card",
  "description": "TESTPRODUCT",
  "status": "success",
  "created": "20180625121550200",
  "livemode": true,
  "failure_code": null,
  "failure_message": null,
  "expire_time_seconds": 3600
}
```

6.4. QR API

6.4.1. Inquiry QR Transaction

URL	/qr/v2/qr/{charge_id order_id}
HTTP Method	GET

Path Parameter

No	Name	Type	Mandatory	Description
1	charge_id	String (100)	O	Charge ID request to inquiry information Sample Data: "chrg_prod_12345678"
2	order_id	String (100)	O	Order ID request to inquiry information Sample Data: "ordr_prod_12345678"



Response Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "charge"
3	created	String (17)	M	Creation date of the charge Format: YYYYMMDDHHmmSS Sample Data: "20180322121944000"
4	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
5	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "amount": "200.50"
6	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "currency": "THB"
7	description	String (255)	M	Product description Sample Data: "Awesome Product"
8	metadata	List Object (255)	O	Additional request object
8.1	item	String (255)	O	Product item
8.2	qty	Integer	O	Product item quantity
8.3	amount	Decimal	O	The amount charged in currency unit
	Sample Data			
	<pre>"metadata": ["item": "paper", "qty": "6", "amount": "10"]</pre>			
9	source	Object	M	The source object that was charged
9.1	id	String (50)	M	Object ID
9.2	object	String	M	Object type
9.3	brand	String (10)	M	Object brand



No	Name	Type	Mandatory	Description
	Sample Data			
	<pre>"source": { "id": "qr_prod_12345678", "object": "qr", "brand": "ThaiQR" }</pre>			
10	status	Enum (success , fail)	M	Whether the charge is authorized (success) or not (failed) Sample Data: "success"
11	order_id	String (100)	M	Order ID Sample Data: "order_prod_12345678"
12	transaction_state	String (50)	M	Charge transaction state - Authorized : Authorized success - Voided : Void transaction complete - Settled : Payment transaction settled - Refunded : Payment transaction refunded Sample Data: "Authorized"
13	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
14	failure_code	String (256)	O	Failure code returned when there is an error Sample Data: "token_invalid"
15	failure_message	String (256)	O	Failure description returned when there is an error Sample Data: "Token was expired or cannot be found"



Sample Request

```
curl -X GET https://dev-kpaymentgateway-services.kasikornbank.com/qr/v2/qr/chrg_prod_12345678 \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXgghLpRro5oAtR"
```

Sample Response

```
{
  "id": "chrg_test_192f3dac0c01a3048a19678d13c180e2d21",
  "order_id": "order_test_1927d35fb9b9cf943a19963e5387678db60",
  "object": "charge",
  "amount": 1.07,
  "currency": "THB",
  "transaction_state": "Authorized",
  "source": {
    "id": "qr_test_192840aaed9a01a402893fc99026cee5021",
    "object": "qr",
    "brand": "ThaiQR"
  },
  "created": "20190610220533000",
  "status": "success",
  "reference_order": "c2676e3f99283143a7d9",
  "description": "TestDescription",
  "livemode": false,
  "metadata": null,
  "failure_code": null,
  "failure_message": null
}
```



6.4.2. Void QR Transaction

!

การยกเลิกรายการ Thai QR Code ทำได้เฉพาะรายการที่ชำระเงินด้วยแอปพลิเคชัน K PLUS เท่านั้น

URL	/qr/v2/qr/{charge_id}/void
HTTP Method	POST

Path Parameter

No	Name	Type	Mandatory	Description
1	charge_id	String (100)	M	Charge ID request to void transaction Sample Data: charge_id="chrg_prod_12345678"

Response Attributes

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "charge"
3	created	String (17)	M	Creation date of the charge Format: YYYYMMDDHHmmSS Sample Data: "20180322121944000"
4	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
5	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"
6	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"



No	Name	Type	Mandatory	Description
7	description	String (255)	M	Product description Sample Data: "Awesome Product"
8	metadata	List Object (255)	O	Additional request object
8.1	item	String (255)	O	Product item
8.2	qty	Integer	O	Product item quantity
8.3	amount	Decimal	O	The amount charged in currency unit
Sample Data <pre>"metadata": [{ "item": "paper", "qty": "6", "amount": "10" }]</pre>				
9	source	Object	M	The source object that was charged
9.1	id	String (50)	M	Object ID
9.2	object	String	M	Object type
9.3	brand	String (10)	M	Object brand
Sample Data <pre>"source": { "id": "qr_prod_12345678", "object": "qr", "brand": "ThaiQR" }</pre>				
10	status	Enum (success , fail)	M	Whether the charge is authorized (success) or not (failed) Sample Data: "success"
11	order_id	String (100)	M	Order ID Sample Data: "order_prod_12345678"
12	transaction_state	String (50)	M	Charge transaction state - Authorized : Void transaction fail



No	Name	Type	Mandatory	Description
				<p>- Voided : Void transaction complete Sample Data: "Voided"</p>
13	reference_order	String (50)	M	<p>Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"</p>
14	failure_code	String (256)	O	<p>Failure code returned when there is an error Sample Data: "no_transaction_found"</p>
15	failure_message	String (256)	O	<p>Failure description returned when there is an error Sample Data: Transaction does not exist in the system"</p>

Sample Request

```
curl -X POST https://dev-kpaymentgateway-
services.kasikornbank.com/qr/v2/qr/chrg_prod_12345678/void \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR"
```

Sample Response

```
{
  "id": "chrg_test_2092372cd2a07d3e14414b697288904f84df8",
  "object": "charge",
  "created": "20220311163558675",
  "livemode": false,
  "amount": 200.0,
  "currency": "THB",
  "description": "TESTPRODUCT",
  "metadata": null,
  "source": {
    "id": "qr_test_20923e21726b5c17445b3b12aa4c639f15816",
    "object": "charge",
    "brand": "ThaiQR",
    "card_masking": null,
  }
}
```



```

    "issuer_bank": null
},
"status": "success",
"order_id": "order_test_2092319d5fa92414646c89ef1aed547274e3d",
"transaction_state": "VOID",
"reference_order": null,
"failure_code": null,
"failure_message": null
}

```

6.4.3. Cancel QR Transaction

URL	/qr/v2/qr/{qr_id}/cancel
HTTP Method	POST

Path Parameter

No	Name	Type	Mandatory	Description
1	qr_id	String (100)	M	QR ID request to cancel Sample Data: "qr_prod_12345678"

Response Attributes

No	Name	Type	Mandatory	Description
1	id	String (100)	M	QR ID Sample Data: "qr_prod_12345678"
2	object	String	M	Object type Sample Data: "qr"
3	created	String (17)	M	Creation date of the qr Sample Data: "20180322121944000"
4	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true



No	Name	Type	Mandatory	Description
5	status	Enum (success , fail)	M	Whether the qr is cancel (success) or not (fail) Sample Data: "success"
6	failure_code	String (256)	O	Failure code returned when there is an error
7	failure_message	String (256)	O	Failure description returned when there is an error

Sample Request

```
curl -X POST https://dev-kpaymentgateway-
services.kasikornbank.com/qr/v2/qr/qr_prod_12345678/cancel \
-H "x-api-key : skey_prod_41Bbw6At8dJjVyKV3ZaXghhLpRro5oAtR"
```

Sample Response

```
{
  "id": "qr_test_20923824b98882b5f473ba3fd0efa5388854e",
  "object": "qr",
  "created": "20220311165317200",
  "livemode": false,
  "status": "success",
  "failure_code": null,
  "failure_message": null
}
```



6.5. WebHook Notify API

!
เพื่อความปลอดภัยในการรับ-ส่งข้อมูล Real-time Response ระหว่างร้านค้ากับระบบ K-Payment Gateway ร้านค้าจะต้องมีการตรวจสอบค่า Checksum ทุกครั้ง โดยนำข้อมูลที่ได้รับใน JSON request body มา Generate ค่าตามที่กำหนด (ตามรายละเอียดด้านล่าง) เพื่อเปรียบเทียบว่าตรงกับค่าในตัวแปร checksum ที่ได้รับหรือไม่ หากค่าไม่ตรงกัน ร้านค้าจะต้องตรวจสอบรายการอีกครั้งก่อนดำเนินการต่อ

6.5.1. How to calculate checksum

เมื่อได้รับ WebHook Notify ระบบของร้านค้าจะต้องตอบกลับให้ K-Payment Gateway ทราบด้วย Response HTTP status code **200 OK** โดยไม่จำเป็นต้องมี JSON response body

สำหรับการ Generate ค่า Checksum ทำได้โดยนำข้อมูลที่ได้รับใน JSON request body มาเรียงต่อกันตามลำดับที่กำหนดไว้ แล้วต่อท้ายด้วย Secret Key ของร้านค้า (ที่ได้รับจาก KBank) จากนั้นนำไปเข้ารหัสแบบ SHA-256

ลำดับการเรียงข้อมูลสำหรับนำไป Generate ค่า Checksum เป็นดังนี้

!
id (Charge ID) + amount (convert to string format to 4 decimal places) + currency + status + transaction_state + salt (salt is merchant secret key)

!
ร้านค้าจะต้องเก็บข้อมูล Secret Key ไว้เป็นความลับระหว่างร้านค้าและธนาคารเท่านั้น
ห้ามเปิดเผยข้อมูลแก่บุคคลอื่นที่ไม่มีหน้าที่เกี่ยวข้องเป็นอันขาด



Sample

```
{
  "id": "chrg_prod_12345678",
  "order_id": "order_prod_12345678",
  "object": "qr",
  "amount": 15,
  "currency": "THB",
  "transaction_state": "Authorized",
  "source": {
    "id": "qr_prod_12345678",
    "object": "qr",
    "brand": "ThaiQR"
  },
  "created": "20180322121944000",
  "status": "success",
  "reference_order": "WT123456",
  "description": "TestDescription",
  "livemode": true,
  "failure_code": "",
  "failure_message": "",
  "checksum": "cc1152742bc4a1169341f8fee95baf2d5886510d577fa95e751cdc6c29162a3a"
}
```

Id+ amount+ currency+ status+
transaction_state+ salt



chrg_prod_1234567815.0000THBsucce
ssAuthorizedskey_prod_41Bbw6At8dJJ
VyKV3ZaXgħħLpRro5oAtR



using SHA-256 encrypted

cc1152742bc4a1169341f8fee95baf2d5
886510d577fa95e751cdc6c29162a3a

ร้านค้าจะต้องเปรียบเทียบค่า Checksum ที่ Generate ได้กับค่าที่ได้รับจากตัวแปร checksum ใน JSON request body ว่าค่าตรงกันหรือไม่ ถ้าค่าทั้งสองตรงกัน แสดงว่าข้อมูลที่ได้รับถูกต้อง และสามารถนำข้อมูลการชำระเงินของรายการนี้ไป Update ในระบบของร้านค้าได้



6.5.2. WebHook Notify API for Card Payment

URL	The Notify URL that provided by merchant
HTTP Method	POST

Request Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "object": "charge"
3	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"
4	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
5	transaction_state	String (50)	M	Charge transaction state - Pre-Authorized: Payment need to do authentication 3D secure - Authorized: Authorized success - Declined: Reject payment from host - Reversed: Payment failed from system reject Sample Data: "Authorized"
6	source	Object	M	The source object that was charged
6.1	id	String (50)	M	Object ID
6.2	object	String	M	Object type
6.3	brand	String (10)	M	Object brand
6.4	card_masking	String (16)	M	Masked card number
6.5	issuer_bank	String (56)	O	Card issuer bank name
	Sample Data			



No	Name	Type	Mandatory	Description
	"source": { "id": "card_prod_12345678", "object": "card", "brand": "mastercard", "card_masking": "123443*****8765", "issuer_bank": "Kasikornbank Public Limited" }			
7	created	String (17)	M	Creation date of the charge Format: YYYYMMDDHHmmSS Sample Data: "20180322121944000"
8	status	Enum (success , fail)	M	Whether the charge is authorized (success) or not (failed) Enum ('success','fail') Sample Data: "success"
9	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
10	description	String (255)	M	Product description Sample Data: "Awesome Product"
11	redirect_url	String (256)	O	Dynamic URL returned for next process Sample Data: "redirect_url": "http://kpgw.kasikornbank.com/3ds/dfesf31231 23/"
12	approval_code	String (20)	O	Authorization's approval code from Issuer Sample Data: "123456"
13	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
14	ref_2	String (128)	O	Reference 2 Sample Data: "ref_2"="reference_id234"
15	ref_3	String (128)	O	Reference 3 Sample Data: "ref_3"="reference_id345"



No	Name	Type	Mandatory	Description
16	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
17	failure_code	String (256)	O	Failure code returned when there is an error Sample Data: "token_invalid"
18	failure_message	String (256)	O	Failure description returned when there is an error Sample Data: "Token was expired or cannot be found"
19	campaign_id	String (8)	O	BIN - Campaign is feature to support when merchant set promotion by merchant able to specify date/time and participating card promotion. Sample Data: "00000001"
20	convfee	Object	O	Convenience Fee is an extra levied charged for the privilege of paying for a product or service which support to some specific merchant and applicable for Visa and MasterCard card payment only. *Condition: Refund will be processed by bank operation team for convenience fee transaction and only full refund is allowed.
20.1	convfee_amount	Decimal (14,4)	O	Convenience fee amount
20.2	convfee_vat_amount	Decimal (14,4)	O	Convenience fee vat amount
20.3	net_amount	Decimal (14,4)	O	Total amount include Convenience fee and vat amount
	Sample Data			
	<pre>"convfee": { "convfee_amount": 2.5, "convfee_vat_amount": 0.18,</pre>			



No	Name	Type	Mandatory	Description
	"net_amount": 12.68 }			
21	mpi	Object	O	MPI Object
21.1	eci	String (2)	O	Electronic Commerce Indicator (ECI)
21.2	xid	String (40)	O	The transaction ID from the 3D Secure provider
21.3	cavv	String (40)	O	Cardholder Authentication Verification Value (CAVV)
21.4	kbank_mpi	Boolean	O	Flag for mark current charge is KBank MPI or not
	Sample Data			
	<pre>"mpi": { "eci": "05", "xid": "30", "cavv": "000001087033360000000008633333000000000000", "kbank_mpi": true}</pre>			
22	checksum	String (1024)	M	<p>Checksum value generated by K-Payment Gateway</p> <p>Sample Data:</p> <p>"5f60d113a45fa44055ce2359c51bda57aaf42172 81979db824bc2ecb771b736f"</p>

Response Parameters

Field Name	Type	Mandatory	Description
status code 200	string	M	The HTTP 200 OK success status response code indicates that the request has succeeded.

Sample Request

Sample Response

The HTTP 200 OK



6.5.3. WebHook Notify API for QR Payment

URL	The Notify URL that provided by merchant
HTTP Method	POST

Request Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "chrg_prod_12345678"
2	order_id	String (100)	M	Order ID Sample Data: "order_prod_12345678"
3	object	String	M	Object type Sample Data: "charge"
4	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "200.50"
5	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
6	transaction_state	String (50)	M	Charge transaction state - Authorized : Authorized success Sample Data: "Authorized"
7	source	Object	M	The source object that was charged
7.1	id	String (50)	M	Object ID
7.2	object	String	M	Object type
7.3	brand	String (10)	M	Object brand
	Sample Data			
	"source": { "id": "qr_prod_12345678", "object": "qr", "brand": "ThaiQR"}			
8	created	String (17)	M	Creation date of the charge



No	Name	Type	Mandatory	Description
				Format: YYYYMMDDHHmmSS Data Type: String (17) Sample Data: "20180322121944000"
9	status	Enum (success , fail)	M	Whether the charge is authorized (success) or not (failed) Sample Data: "success"
10	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
11	description	String (255)	M	Product description Sample Data: "Awesome Product"
12	livemode	Boolean	M	Whether this is live environment (true) or not (false) Sample Data: true
13	failure_code	String (256)	O	Failure code returned when there is an error Sample Data: "token_invalid"
14	failure_message	String (256)	O	Failure description returned when there is an error Sample Data: "Token was expired or cannot be found"
15	checksum	String (1024)	M	Checksum value generated by K-Payment Gateway Sample Data: "5f60d113a45fa44055ce2359c51bda57aaf42172 81979db824bc2ecb771b736f"

Response Parameters

Field Name	Type	Mandatory	Description
status code 200	string	M	The HTTP 200 OK success status response code indicates that the request has succeeded.



Sample Request

```
{  
    "id": "chrg_prod_12345678",  
    "order_id": "order_prod_12345678",  
    "object": "qr",  
    "amount": 15,  
    "currency": "THB",  
    "transaction_state": "Authorized",  
    "source": {  
        "id": "qr_prod_12345678",  
        "object": "qr",  
        "brand": "ThaiQR"  
    },  
    "created": "20180322121944000",  
    "status": "success",  
    "reference_order": "WT123456",  
    "description": "TestDescription",  
    "livemode": true,  
    "failure_code": "",  
    "failure_message": "",  
    "checksum": "5f60d113a45fa44055ce2359c51bda57aaf4217281979db824bc2ecb771b736f"  
}
```

Sample Response

```
The HTTP 200 OK
```



6.5.4. WebHook Notify API for Alipay

URL	The Notify URL that provided by merchant
HTTP Method	POST

Request Parameter

No	Name	Type	Mandatory	Description
1	id	String (100)	M	Charge ID Sample Data: "id": "chrg_prod_12345678"
2	object	String	M	Object type Sample Data: "object": "charge"
3	amount	Decimal (12,2)	M	The amount charged in currency unit Sample Data: "amount": "200.50"
4	currency	String (3)	M	3 Letter ISO Currency code in upper case Sample Data: "THB"
5	transaction_state	String (50)	M	Charge transaction state <ul style="list-style-type: none"> - Initialize: Payment transaction initialized - Authorized: Authorized success - Declined: Reject payment from host - Reversed: Payment failed from system reject Sample Data: "Authorized"
6	source	Object	M	The source object that was charged
6.1	id	String (50)	M	Object ID
6.2	object	String	M	Object type
6.3	brand	String (10)	M	Object brand
	Sample Data			
	<pre>"source": { "id": "redr_prod_12345678", "object": "redirect", "brand": "Alipay" }</pre>			
7	created	String (17)	M	Creation date of the charge



No	Name	Type	Mandatory	Description
				Format: YYYYMMDDHHmmSS Sample Data: "20180322121944000"
8	status	Enum (success , fail)	M	Whether the charge is authorized (success) or not (failed) Sample Data: "success"
9	reference_order	String (50)	M	Reference number generated by merchant site (must be unique) Sample Data: "20180530175600"
10	description	String (255)	M	Product description Sample Data: "Awesome Product"
11	redirect_url	String (256)	O	Dynamic URL returned for next process Sample Data: "http://kpgw.kasikornbank.com/alipay/dfesf31223/"
12	ref_1	String (128)	O	Reference 1 Sample Data: "reference_id123"
13	ref_2	String (128)	O	Reference 2 Sample Data: "reference_id234"
14	ref_3	String (128)	O	Reference 3 Sample Data: "reference_id345"
15	livemode	Boolean	M	Whether this is live environment or not Sample Data: true
16	failure_code	String (256)	O	Failure code returned when there is an error Sample Data: "token_invalid"
17	failure_message	String (256)	O	Failure description returned when there is an error Sample Data: "Token was expired or cannot be found"
18	checksum	String (1024)	M	Checksum value generated by K-Payment Gateway Sample Data:



No	Name	Type	Mandatory	Description
				"5f60d113a45fa44055ce2359c51bda57aaf42172 81979db824bc2ecb771b736f"

Response Parameters

Field Name	Type	Mandatory	Description
status code 200	string	M	The HTTP 200 OK success status response code indicates that the request has succeeded.

Sample Request

```
{
  "id": "chrg_test_1796dc2004414f8297c3825d0982cde3",
  "object": "charge",
  "amount": 1.0,
  "currency": "THB",
  "transaction_state": "Authorized",
  "source": {
    "id": "redr_prod_42f00571ac396ad600ce8e72b0e58def1",
    "object": "redirect",
    "brand": "Alipay"
  },
  "created": "20180930212842000",
  "status": "success",
  "reference_order": "QT5663363",
  "description" : "TestDescription",
  "redirect_url": null,
  "ref_1" : "reference_1",
  "ref_2" : "reference_2",
  "ref_3" : "reference_3",
  "livemode": false,
  "failure_code":null,
  "failure_message": "",
  "checksum": "cebe1f2637a644b3535bd40feb38f270fdf377b874eb6bce375e8d1207377e37"
}
```

Sample Response

The HTTP 200 OK



7. บัตรทดสอบใน Sandbox และ Simulator

7.1. การชำระเงินแบบ 3D Secure

!

ขั้นตอนการยืนยันตัวตนของลูกค้าผู้ถือบัตรเป็นเพียงระบบจำลองเท่านั้น

กรุณาดูปุ่ม "Request OTP" และระบุค่า OTP เป็น "123456"

สำหรับชำระเต็มจำนวนแบบ MCC และ ผ่อนชำระ (SmartPay)

Card Number	Card Brand	CVV	Expiry (MM/YY)	Response Code	Description
5239620007270264	MASTERCARD	Any	Any	00	Approval
5404888000072727	MASTERCARD	Any	Any	00	Approval
5404888005380836	MASTERCARD	Any	Any	00	Approval
5411768003140872	MASTERCARD	Any	Any	00	Approval
5411768003141953	MASTERCARD	Any	Any	00	Approval
2221000604647064	MASTERCARD	Any	Any	01	Declined – Refer to card issuer
5596886582263266	MASTERCARD	Any	Any	05	Declined – Do not honor
5149504001969007	MASTERCARD	Any	Any	51	Declined – Insufficient Funds
5265434283867885	MASTERCARD	Any	Any	51	Declined – Insufficient Funds
4417706600005830	VISA	Any	Any	00	Approval
3564580000056476	JCB	Any	Any	00	Approval

สำหรับชำระเต็มจำนวนแบบ DCC

Card Number	Card Brand	CVV	Expiry (MM/YY)	Response Code	Description
5239620007270264	MASTERCARD	Any	Any	00	Approval (USD)
4417706600005830	VISA	Any	Any	00	Approval (JPY)



7.2. การชำระเงินแบบ Non 3D Secure

สำหรับชำระเต็มจำนวนแบบ MCC และ ผ่อนชำระ (SmartPay)

Card Number	Card Brand	CVV	Expiry (MM/YY)	Response Code	Description
5431289719925031	MASTERCARD	Any	Any	00	Approval
5565694828631115	MASTERCARD	Any	Any	00	Approval
2221000604647064	MASTERCARD	Any	Any	01	Declined – Refer to card issuer
5596886582263266	MASTERCARD	Any	Any	05	Declined – Do not honor
5265434283867885	MASTERCARD	Any	Any	51	Declined – Insufficient Funds
5363314260912204	MASTERCARD	Any	Any	500	Declined – Timeout
4024007134786226	VISA	Any	Any	00	Approval
4485169016314923	VISA	Any	Any	00	Approval
4877300371267473	VISA	Any	Any	01	Declined – Refer to card issuer
4532741632025415	VISA	Any	Any	05	Declined – Do not honor
4147602662715373	VISA	Any	Any	51	Declined – Insufficient Funds
4024007121398514	VISA	Any	Any	500	Declined – Timeout
3536353014086177	JCB	Any	Any	00	Approval
3540000981507917	JCB	Any	Any	01	Declined – Refer to card issuer
3539074482135968	JCB	Any	Any	05	Declined – Do not honor
3528730936342392	JCB	Any	Any	51	Declined – Insufficient Funds
3532598050794027	JCB	Any	Any	500	Declined – Timeout

สำหรับชำระเต็มจำนวนแบบ DCC

Card Number	Card Brand	CVV	Expiry (MM/YY)	Response Code	Description
5372074248113841	MASTERCARD	Any	Any	00	Approval (USD)
4205203648166408	VISA	Any	Any	00	Approval (JPY)



7.3. บัตรทดสอบ UnionPay (TPN/UPI)

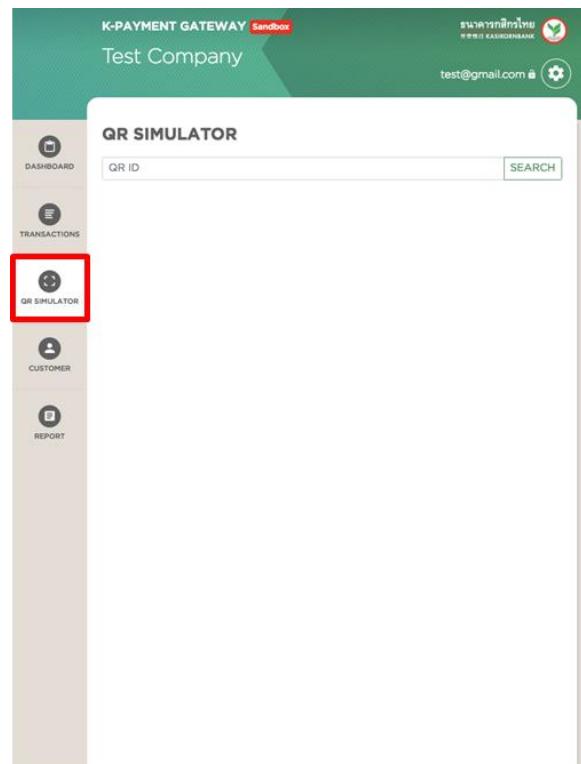
Card Number	Card Brand	CVN2	Expiry (MM/YY)	SMS Code	Pin
6210947764000027	TPN	456	10/30	-	111111
9100009764000010	TPN	123	03/25	-	111111
9100008764000025	TPN	123	03/25	-	111111
9100009764000015	TPN	123	03/25	-	111111
6250947000000014	UPI	123	12/33	111111 (PC) or	111111
6250946000000016	UPI	123	12/33	123456 (Mobile)	111111

7.4. ทดสอบการชำระเงินด้วย Thai QR Simulator ใน Sandbox

(1) บนหน้าจอการชำระเงินด้วย QR Code จะแสดง
ข้อมูล QR ID แสดงอยู่ด้านล่าง



(2) ร้านค้าเข้าสู่ Merchant Portal (Sandbox) และ^ลคลิกที่ไอคอนการตั้งค่า จากนั้นไปที่เมนู "QR Simulator"





(3) ค้นหา QR ID ที่แสดงบนหน้าจอชำระเงิน

(5) กดปุ่ม "Pay" จำลองการสแกน QR Code เพื่อชำระเงิน

(4) ระบบจะแสดงรายละเอียดรายการชำระเงินของ QR Code นั้น

(6) ที่หน้าจอชำระเงินจะเปลี่ยนเป็นแสดงผลการชำระเงินสำเร็จดังภาพ



8. ตัวอย่าง Code

8.1. Code Java สำหรับการรับ Token

```
// Set your secret key: remember to change this to your live secret key in production
String apiKey = "sk_test_BQokikJOvBil2HlWgH4olfQ2";
// Token is created using Checkout or Elements!
// Get the payment token ID submitted by the form:
String token = request.getParameter("token");
Client client=null;
// Charge the user's card:
Endpoint ep = configService.getEndpoint();
client = new Client(ep, apiKey);
Builder chargeRequest = new
Charge.Builder().mode("token").amount(100).currency("THB").source_type("card").reference_o
rder("INV00001").description(description).token(token);
Charge charge = Charge.create(chargeRequest);
// ** please check payment success condition from charge response object with transaction_state =
"Authorized" and status = "success"
if (ChargeStatus.success.equals(charge.getStatus())) {
    // show payment success to customer
} else {
    // show fail payment to customer
}
```

8.2. ตัวอย่าง Code Java สำหรับการเรียก Create Order API

```
// Set your secret key: remember to change this to your live secret key in production
String apiKey = "sk_test_BQokikJOvBil2HlWgH4olfQ2";
Client client=null;
// Create order:
Endpoint ep = configService.getEndpoint();
client = new Client(ep, apiKey);
```



```
Builder chargeModel = new Order.Builder();
orderModel = new Order.Builder().amount(amount).currency(currency).description(description).refNumber(ref_number).sourceType("qr");

Order order = client.order().create(orderModel);
// keep order id in session for used in KPAY UI SDK
setOrder_id(order.getId());
```

8.3. ตัวอย่าง Code Java สำหรับการรับค่า Charge ID

```
// Set your secret key: remember to change this to your live secret key in production
// Set your secret key: remember to change this to your live secret key in production

String apiKey = "sk_test_BQokikJOvBil2HlWgH4olfQ2";
// Get the payment charge ID submitted by the form:
String chargeld = request.getParameter("chargeld");
Client client=null;
// Inquiry QR payment status by chargeld
Endpoint ep = configService.getEndpoint();
client = new Client(ep, apiKey);
Charge charge = client.qr().inquiry(chargeld);
// Verify QR payment status for present result to customer's view
if (ChargeStatus.success.equals(charge.getStatus())) {
    setIsSuccess(true);
} else {
    setErrorMessage(charge.getFailureCode() + " " + charge.getFailureMessage());
}
```

8.4. ตัวอย่าง Code Java สำหรับการรับ WebHook Notify

- ร้านค้าต้องพัฒนาระบบ RESTful API เพื่อรับ HTTP POST WebHook Notify จาก K-Payment Gateway ซึ่งข้อมูลต่างๆจะถูกส่งให้ใน JSON request body

```
@RequestMapping(value = "/charge-webhook/callback", method = RequestMethod.POST)
@ResponseBody
```



```

public ResponseEntity<tech.kbtg.ngw.sandbox.web.model.charge3d.Charge3dResp>
postWebHook(@RequestBody String stringRequestEntity) {
    // Set your secret key: remember to change this to your live secret key in production
    String apiKey = "sk_test_BQokikJOvBil2HlWgH4olfQ2";
    Charge3dResp resp = new Charge3dResp();
    Charge3dCallbackReq charge3dCallbackReq;
    try {
        ObjectMapper objectMapper = new ObjectMapper();
        charge3dCallbackReq = objectMapper.readValue(stringRequestEntity,
        Charge3dCallbackReq.class);
    } catch (IOException e) {
        log.error("error phasing charge request", e);
        resp.setResponseCode("ERROR");
        resp.setResponseMSG("JSON parsing error");
        return ResponseEntity.ok(resp);
    }
}

```

2. สร้าง Function เพื่อ Generate ค่า Checksum โดยนำข้อมูลที่ได้รับใน JSON request body มาเรียงต่อกันตามลำดับที่กำหนดไว้ และต่อท้ายด้วย Secret Key ของร้านค้า(ที่ได้รับจาก KBank) จากนั้นนำไปเข้ารหัสแบบ SHA-256

ลำดับการเรียงข้อมูลสำหรับนำไป Generate ค่า Checksum เป็นดังนี้

id (Charge ID) + **amount** (convert to string format to 4 decimal places) + **currency** + **status** + **transaction_state** + **salt** (salt is merchant secret key)

ตัวอย่าง Java Function

```

protected String generateChecksum(Charge3dCallbackReq charge, String secretKey) {
    String amount = Optional.ofNullable(charge.getAmount()).map(d ->
    String.format("%.4f", d)).orElse("null");
    String initString = String.join("", Arrays.asList(charge.getId(),
    amount,
    charge.getCurrency(),
    charge.getStatus(),
    
```



```

        charge.getTransactionState(),
        charge.getFailureCode(),
        secretKey));
String hash = hashSha256(initString);
return hash;
}

public static String hashSha256(String value) {
try {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    md.update(value.getBytes());
    byte byteData[] = md.digest();
    // convert the byte to hex
    StringBuffer sb = new StringBuffer();
    for (int i = 0; i < byteData.length; i++) {
        sb.append(Integer.toString((byteData[i] & 0xff) + 0x100,
16).substring(1));
    }
    return sb.toString();
} catch (NoSuchAlgorithmException e) {
    throw new IllegalStateException();
}
}
}

```

3. เมื่อร้านค้าได้รับ WebHook Notify จาก K-Payment Gateway (ด้วย HTTP POST) ก็เรียก Function generateChecksum โดยใช้ค่าที่ได้รับจาก JSON request body และ Secret Key

```

// generate check sum from charge response
String checkSum = generateChecksum(charge3dCallbackReq, secretKey);

```

4. นำค่า Checksum ที่ได้จากการเรียก Function generateChecksum มาเปรียบเทียบกับค่าที่ได้รับจากตัวแปร checksum ใน JSON request body

ถ้าค่าทั้งสองตรงกัน แสดงว่าข้อมูลที่ได้รับถูกต้อง และสามารถนำข้อมูลการชำระเงินของรายการนี้ไป Update ในระบบของร้านค้าได้



```
// compare check sum with check value in web hook json data to confirm the data not be  
change by another  
  
if(!checkSum.equals(charge3dCallbackReq.getChecksum())) {  
    resp.setResponseCode("ERROR");  
    resp.setResponseMSG(" Invalid - checksum calculated = " + checkSum + ",  
received=" + charge3dCallbackReq.getChecksum());  
    return ResponseEntity.ok(resp);  
}
```

จากนั้น ระบบของร้านค้าจะต้องตอบกลับ K-Payment Gateway ด้วย Response HTTP status code "200 OK"



9. Appendix

Transaction State and Status

1) รายการ 3D Secure

Description	Transaction State	Status
รายการอนุมัติ	Authorized	Success
ผู้ถือบัตรยืนยันตัวตนผ่านแล้ว แต่รายการไม่ได้รับการอนุมัติ	Declined	Failed
ผู้ถือบัตรยืนยันตัวตนไม่ผ่าน	Pre-Authorized	Failed
ผู้ถือบัตรยังไม่ได้ยืนยันตัวตน (ยังไม่ได้กรอก OTP)	Pre-Authorized	Pending
K-Payment ส่ง Dynamic URL สำหรับการยืนยันตัวตน (OTP) ให้ร้านค้าแล้ว รอร้านค้าดำเนินการต่อ	Pre-Authorized	Success

2) รายการ Non 3D Secure

Description	Transaction State	Status
รายการอนุมัติ	Authorized	Success
รายการไม่ได้รับการอนุมัติ	Declined	Failed

Failure Code for Card Payment

Failure Code	Failure Message	Explanation
01	Refer to Card Issuer	The customers issuing bank has indicated there is a problem with the credit card number. The customer should contact their bank. The customer should use an alternate credit card.
02	Refer to Issuer's Special Conditions	The customers issuing bank has indicated there is a problem with the credit card number. The customer should contact their bank. The customer should use an alternate credit card.
03	Invalid Merchant ID	The Merchant ID is invalid, you should contact your Bank and ensure you have provided the correct Merchant Account Number.
04	Pick Up Card	The customers issuing bank has declined the transaction and requested that your customer's credit card be retained. (card reported lost or stolen). The customer should use an alternate credit card.



Failure Code	Failure Message	Explanation
05	Do Not Honor	The customers issuing bank has declined the transaction as the credit card number has failed a security check, or the funds have been frozen or depleted. The customer should use an alternate credit card.
06	Error	The customers issuing bank has declined the transaction as there is a problem with the credit card number. The customer should contact their bank. The customer should use an alternate credit card.
07	Pick Up Card, Special Conditions	The customer's bank (Card Issuer) has declined the transaction and requested that your customer's credit card be retained. (card reported lost or stolen) The customer should use an alternate credit card.
08	Honor with ID	The transaction was successful. Some Banks use this code instead of '00'.
09	Request in Progress	The customer's bank (Card Issuer) has indicated there is a problem with the credit card number. The customer should contact their bank. The customer should use an alternate credit card.
10	Partial Amount Approved	The transaction was successful.
11	Approved VIP	The transaction was successful. (This code is not used in Australia.)
12	Invalid Transaction	The customer's bank (Card Issuer) has declined the transaction because of an invalid format or field. Check the transaction information and try processing the transaction again.
13	Invalid Amount	An invalid character (e.g. a dollar sign or a space) may be being passed to the gateway. Check your website's code.
14	Invalid Card Number	The customer's bank (Card Issuer) has declined the transaction as the Credit Card number does not exist. Check the credit card information and try processing the transaction again.
15	No Sun Issuer	The customer's bank (Card Issuer) does not exist. Check the credit card information and try processing the transaction again.
16	Approved, Update Track 3	The transaction was successful.
17	Customer Cancellation	This indicates that the transaction was authorised and subsequently voided. Voided transactions do not appear on the customer's statement or form part of your settlement total.
18	Customer Dispute	



Failure Code	Failure Message	Explanation
19	Re-enter Transaction	The transaction has not been processed and the customer should attempt to process the transaction again.
20	Invalid Response	The customers issuing bank has declined the transaction as the credit card number has failed a security check, or the funds have been frozen or depleted. The customer should use an alternate credit card.
21	No Action Taken	The customer's bank (Card Issuer) has indicated there is a problem with the credit card number. The customer should contact their bank. The customer should use an alternate credit card.
22	Suspected Malfunction	The customer's bank (Card Issuer) cannot be contacted during the transaction. The customer should check the credit card information and try processing the transaction again.
23	Unacceptable Transaction Fee	An unspecified error has occurred.
24	File Update not Supported by Receiver	
25	Unable to Locate Record on File	The customer's bank (Card Issuer) does not recognise the credit card details. The customer should check the credit card information and try processing the transaction again.
26	Duplicate File Update Record	
27	File Update Field Edit Error	
28	File Update File Locked Out	
29	File Update not Successful	
30	Format Error	The customer's bank (Card Issuer) does not recognise the transaction details. The customer should check the transaction information and try processing the transaction again. Usually due to an incorrect TID.
31	Bank not Supported by Switch	The customer's bank (Card Issuer) has declined the transaction as it does not allow transactions originating through mail /telephone, fax, email or Internet orders. This error is associated customers attempting to use a Discover Card. The customer should use an alternate credit card.



Failure Code	Failure Message	Explanation
32	Completed Partially	
33	Expired Card - Pick Up	The customer's bank (Card Issuer) has declined the transaction as Credit Card has expired or the date is incorrect. Check the expiry date in the transaction and try processing the transaction again.
34	Suspected Fraud - Pick Up	The customer's bank (Card Issuer) has declined the transaction as there is a suspected fraud on this Credit Card number.
35	Contact Acquirer - Pick Up	The customer's bank (Card Issuer) has declined the transaction and requested that the customer's credit card be retained (card reported lost or stolen). The customer should use an alternate credit card.
36	Restricted Card - Pick Up	The customer's bank (Card Issuer) has declined the transaction and requested that the customer's credit card be retained. (card reported lost or stolen) The customer should use an alternate credit card.
37	Call Acquirer Security - Pick Up	The customer's bank (Card Issuer) has declined the transaction and requested that your customer's credit card be retained. (card reported lost or stolen) The customer should use an alternate credit card.
38	Allowable PIN Tries Exceeded	The customer's bank (Card Issuer) has declined the transaction as the customer has entered the incorrect PIN three times. The customer's bank (Card Issuer) has requested you retain the credit card. The customer should use an alternate credit card and contact their bank.
39	No Credit Account	The customer's bank has declined the transaction as the Credit Card number used is not a credit account. The customer should use an alternate credit card.
40	Requested Function not Supported	The customer's bank (Card Issuer) has declined the transaction as it does not allow this type of transaction. The customer should use an alternate credit card.
41	Lost Card - Pick Up	The customer's bank (Card Issuer) has declined the transaction as the card has been reported lost. The customer should use an alternate credit card.
42	No Universal Amount	The customer's bank (Card Issuer) has declined the transaction as the account type selected is not valid for this credit card number. The customer should use an alternate credit card.
43	Stolen Card - Pick Up	The customer's bank (Card Issuer) has declined the transaction as the card has been reported stolen. The customer should use an alternate credit card.



Failure Code	Failure Message	Explanation
44	No Investment Account	The customer's bank (Card Issuer) has declined the transaction as the account type selected is not valid for this credit card number. The customer should use an alternate credit card.
50	Failed Authentication	
51	Insufficient Funds	The customers issuing bank has declined the transaction as the credit card does not have sufficient funds. The customer should use an alternate credit card.
52	No Cheque Account	The customer's bank (Card Issuer) has declined the transaction as the credit card number is associated to a cheque account that does not exist. The customer should use an alternate credit card.
53	No Savings Account	The customer's bank (Card Issuer) has declined the transaction as the credit card number is associated to a savings account that does not exist. The customer should use an alternate credit card.
54	Expired Card	The customer's bank (Card Issuer) has declined the transaction as the credit card appears to have expired. The customer should check the expiry date entered and try again, or use an alternate credit card.
55	Incorrect PIN	The customer's bank (Card Issuer) has declined the transaction as the customer has entered an incorrect PIN. The customer should re-enter their PIN, otherwise use an alternate credit card.
56	No Card Record	The Customer's bank has declined the transaction as the credit card number does not exist. The customer should use an alternate credit card.
57	Trans. not Permitted to Cardholder	The Customer's bank has declined the transaction as this credit card cannot be used for this type of transaction. The customer should use an alternate credit card.
58	Transaction not Permitted to Terminal	The Customer's bank has declined the transaction as this credit card cannot be used for this type of transaction. This may be associated with a test credit card number. The customer should use an alternate credit card.
59	Suspected Fraud	The customer's bank has declined this transaction as the credit card appears to be fraudulent.
60	Card Acceptor Contact Acquirer	The customer's bank (card issuer) has declined the transaction. The customer should contact their bank and retry the transaction.
61	Exceeds Withdrawal Amount Limits	The customer's bank has declined the transaction as it will exceed the customer's card limit. The customer should use an alternate credit card.



Failure Code	Failure Message	Explanation
62	Restricted Card	The customer's bank has declined the transaction as the credit card has some restrictions. The customer should use an alternate credit card.
63	Security Violation	The customer's bank has declined the transaction. The customer should use an alternate credit card.
64	Original Amount Incorrect	The customer's bank has declined the transaction due to the amount attempting to be processed. The customer should check the transaction amount and try again.
65	Exceeds Withdrawal Frequency Limit	The customer's bank has declined the transaction as the customer has exceeded the withdrawal frequency limit. The customer should use an alternate credit card.
66	Card Acceptor Call Acquirer Security	The customer's bank has declined the transaction and request the Merchant to contact the bank. The customer should use an alternate credit card.
67	Hard Capture - Pick Up Card at ATM	The customer's bank has declined the transaction as the card is suspected to be counterfeit. The customer's bank (Card Issuer) has requested that your customer's credit card be retained. The customer should use an alternate credit card.
68	Response Received Too Late	
75	Allowable PIN Tries Exceeded	The customer's bank has declined the transaction as the customer has entered the incorrect PIN more than three times. The customer should use an alternate credit card.
86	ATM Malfunction	
87	No Envelope Inserted	
88	Unable to Dispense	
89	Administration Error	
90	Cut-off in Progress	The customer's bank is temporarily not able to process this customer's credit card. The customer should attempt to process this transaction again.
91	Issuer or Switch is Inoperative	The customer's bank is unable to be contacted to authorise the transaction. The customer should attempt to process this transaction again.
92	Financial Institution not Found	The customer's bank cannot be found for routing. This response code is often returned when the customer is using a test credit card number. The customer should attempt to process this transaction again.



Failure Code	Failure Message	Explanation
93	Trans Cannot be Completed	The customer's bank has declined the transaction and request the customer to contact their bank. The customer should use an alternate credit card.
94	Duplicate Transmission	The customer's bank has declined the transaction as this transaction appears to be a duplicate transmission. No action required.
95	Reconcile Error	
96	System Malfunction	The customer's bank was not able to process the transaction. The customer should attempt to process this transaction again.
97	Reconciliation Totals Reset	
98	MAC Error	
99	Reserved	
N7	Invalid CVV2	
NC	Invalid CVV2 with Approval Code	
500	Timeout	
600	Reverse Transaction	

Failure Code for QR Payment

Failure Code	Failure Message
account_error	Cannot credit to account. (EMACF)
account_error	Account is closed. (EMAC)
authentication_error	Wrong partner keys. (EMA)
authentication_error	Miss match partner id and mid. (EMPM)
bad_request	Invalid JSON. (EMBR)
duplicate_request	This request is duplicated. (EMDR)
internal_error	Cannot connect to database. (EMIDB)
internal_error	System unavailable. (EMIU)
internal_error	EAI system unavailable. (EMIEU)
internal_error	EAI system timeout. (EMIETO)
internal_error	Cannot open wallet account. (EMIGTM)
invalid_account	Account Number does not exist. (EMRAD)



Failure Code	Failure Message
invalid_currency_code	Currency code XXX is not allowed. (EMICC)
invalid_origPartnerTxnUid	origPartnerTxnUid does not exist. (EMITDNE)
invalid_txn_amount	Invalid Amount. (EMITA)
invalid_txn_amount	Amount must be an integer with two digits precision. (EMITF)
invalid_qr_type	QR Type X is not allowed. (EMIQR)
invalid_redis	Invalid Redis. (EMIRD)
invalid_reference1	Reference1 must not be empty. (EMIR)
invalid_shop_id	Shop ID is a duplicate. (EMISD)
invalid_shop_id	Shop ID does not exist. (EMISDNE)
invalid_document_id	Document ID invalid. (EMID)
invalid_account	Invalid account and/or card status. (EMACST)
invalid_settlement_type	Invalid settlement type. (EMIST)
merchant_closed	Merchant is already closed. (EMMC)
duplicate_merchant	Partner ID and Partner Shop ID already registered. (EMRD)
create_merchant	Merchant has not been created. (EMCR)
invalid_merchant	Merchant ID does not exist. (EMDNE)
invalid_merchant	Wallet is suspended. (EMMS)
invalid_merchant	Wallet is suspended or closed (EMMSOC)
invalid_merchant	Merchant balance does not exist. (EMBDNE)
qr_cancelled	QR is cancelled. (EMQRC)
qr_paid	QR already paid. (EMQRP)
qr_void	QR status is not paid. (EMQSNP)
qr_void	QR already voided. (EMQRV)
qr_void	Cannot void between settlement process. (EMQRCVS)
qr_void	Cannot void other channel. (EMQRVOTCH)
qr_void	Cannot void over the day. (EMQRVOD)
qr_expired	QR already expired. (EMQRAE)
qr_expired	QR has expired. (EMQRE)
gtm_error	Cannot open wallet account. (EMGTMA)
gtm_error	Cannot reverse. (EMGTMCR)
gtm_error	Upsert accum error. (EMGTMEAE)
gtm_error	Cannot connect to GTM service. (EMGTMSE)



Failure Code	Failure Message
gtm_error	Cannot receive message from GTM. (EMGTMRE)
gtm_error	Cannot voided. (EMGTMKF10)
settlement_error	Transaction is already settled (EMSET)
settlement_error	Manual settlement is not allow between 22:00 - 02:30 (EMSMSG)
settlement_error	Current balance mismatch. (EMSBM)
settlement_error	Settlement is not allowed. (EMSNA)
settlement_error	Settlement has been failed. (EMSFA)
callback_error	Callback has been failed. (EMCBFA)
callback_error	Save callback logging has been failed. (EMCBSL)
register_error	Settlement account number/name is required. (EMRSR)
register_error	Parent's Merchant is required. (EMHQID)
register_error	Owner email is required. (EMHQID)
update_error	Cannot update merchant (EMUDM)

API Error Code

ALL API

Error Code	Error Message
authentication_failure	Failure to properly authenticate yourself in the ALL request
mandatory_is_empty	Mandatory parameter is missing

Charge API

Error Code	Error Message
amount_exceed	Amount exceeds the available limit which can be CHARGE
amount_invalid	Amount must be positive
card_error	Card type is not supported
charge_id_not_exist	Charge ID does not exist or invalid
currency_invalid	Currency is currently not supported
decrypt_card_error	An error occurred while processing the card decryption
exceed_amount_per_day	exceed maximum charge amount limit per day
exceed_amount_per_txn	exceed maximum charge amount limit per transaction
exceed_count_charge_per_day	exceed maximum number of charge transaction limit per day



float_not_support	Decimal places are not supported with small CHARGE currencies
foreign_card_not_supported	Not support foreign card for register with 3D mode
ip_not_allow	IP address is not allowed
mid_and_tid_not_exist	MID and/or TID do not exist or are not mapped to this currency
not_found_settlement_detail	Settlement detail of this transaction is invalid
payment_type_not_support	Refund is not allowed for Smartpay transaction
pending_state	Transaction status is pending please try again in 15 minutes
processing_card_error	An error occurred while processing the card
refund_state_invalid	Transaction status is not eligible for refund
register_3d_mode_not_allow	Not allow register 3d mode for this merchant
settle_state_invalid	Transaction status is not eligible for settlement
token_conflict	Token has been changed
token_invalid	Token was expired or cannot be found
used_token	Token was already used
void_state_invalid	Transaction status is not eligible for void
campaign_card_invalid	Sorry, your card does not support for this campaign.
campaign_id_invalid	Campaign id does not exist or invalid.

Customer API

Error Code	Error Message
card_invalid	Card ID does not exist or invalid
card_not_verified	Customer ID cannot be generated due to unsuccessful charge
customer_authen_fail	Customer ID does not belong to your company
customer_invalid	Customer ID is invalid or does not exist
decrypt_card_error	An error occurred while processing the card decryption

Order API

Error Code	Error Message
amount_invalid	Amount must be positive
currency_invalid	Currency is currently not supported
exceed_amount_per_day	exceed maximum charge amount limit per day



exceed_amount_per_tx	exceed maximum charge amount limit per transaction
exceed_count_charge_per_day	exceed maximum number of charge transaction limit per day
mid_currency_conflict	MID conflict with currency code
smartpay_invalid	smartpay information cannot found in system
sof_is_empty	don't have any source of fund support for this merchant

QR API

Error Code	Error Message
cancel_state_invalid	Transaction status is not eligible for cancel
charge_id_not_exist	Charge ID does not exist or invalid
exceed_amount_per_day	exceed maximum charge amount limit per day
exceed_amount_per_tx	exceed maximum charge amount limit per transaction
exceed_count_charge_per_day	exceed maximum number of charge transaction limit per day
invalid_order_id	Order ID cannot found in system
no_transaction_found	Transaction status is not eligible for settlement
order_id_is_already_used	Order ID was already used
void_state_invalid	Transaction status is not eligible for void

Refund API

Code	Message	Condition
charge_id_not_exist	Charge ID does not exist or invalid	Validate Charge ID is exist.
authentication_failure	Failure to properly authenticate yourself in the request	Validate your secret key
refund_not_allowed	Refund is not allowed	Validate transaction not ever received Response = OBVAL
refund_transaction_is_not_owed	Invalid refund date range	Exceeding the specified number of days. -In case not specify transaction date and more than 180 days. -In case specify transaction date and more than 366 days.



currency_not_support_deci mals	This currency does not support decimals	Validate small currency in configuration of KPGW
refund_state_invalid	Transaction status is not valid to refund	Validate Transaction status = Settle, Partial Refund, Refund sent
amount_exceed	Refund amount cannot exceed the available limit	Validate Refund Amount not exceed Remaining Amount
partial_refund_is_not_allowed	Invalid refund request	Refund Installment Case Partial Refund a second refund.
internal_permission_denied	Permission denied.	Refund Installment Case Partial Refund M% - IPAN = Mode 01
internal_permission_denied	Permission denied.	Refund Installment Case Full Refund M%-IPAN Mode 01 and C% - IPAN Mode 02
Pending_state	Transaction status is pending please try again in 15 minutes	Validate Transaction status “Refund Fail”
convenience_fee_not_supported	Convenience fee transaction not support	Validate Transaction Convenience fee
invalid_parameter	Invalid Charge ID or Transaction date value	In case Transaction date and chargeid does not match.
transaction_date_invalid	Invalid Transaction date format	In case Transaction date format incorrectly.



10. คำตามที่พบบ่อย

■ การเชื่อมต่อกับ K-Payment Gateway มีวิธี แตกต่างกันอย่างไร

ตอบ ร้านค้าสามารถเชื่อมต่อกับ K-Payment Gateway ได้ 5 วิธี ดังนี้

- 1) Embedded UI คือการฝังปุ่ม **Pay Now** ลงในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้า (ซึ่งแนะนำให้ฝังปุ่ม K-Pay ไว้ที่หน้าชำระเงิน) เมื่อลูกค้าคลิกปุ่ม K-Pay จะปรากฏหน้าจอให้ระบุรายละเอียดบัตรที่ใช้ในการชำระเงิน หรือหน้าจอแสดง QR Code สำหรับชำระเงิน แล้วระบบ K-Payment Gateway จะดำเนินการเชื่อมต่อกับระบบที่เกี่ยวข้องและแจ้งผลการชำระเงินให้ร้านค้าทราบต่อไป
การเชื่อมต่อวิธีนี้รองรับได้ทั้งการชำระเงินด้วยบัตรและการชำระเงินด้วย QR Code
- 2) Inline UI คือการแทรก Tag HTML และปุ่มตามที่ทาง KBank กำหนดไว้ในเว็บไซต์หรือแอปพลิเคชันมือถือของร้านค้าที่หน้าชำระเงิน (โดยไม่ต้องเก็บข้อมูลบัตร) เมื่อลูกค้ากรอรายละเอียดบัตรและคลิกปุ่มเพื่อยืนยันการชำระเงินแล้ว ระบบ K-Payment Gateway จะดำเนินการเชื่อมต่อกับระบบที่เกี่ยวข้องและแจ้งผลการชำระเงินให้ร้านค้าทราบต่อไป
การเชื่อมต่อวิธีนี้รองรับเฉพาะการชำระเงินด้วยบัตรแบบ Full MCC เท่านั้น
- 3) Redirect คือการนำลูกค้าไปยัง URL ที่ปลอดภัยของผู้ออกบัตรหรือระบบที่เกี่ยวข้อง เพื่อดำเนินการชำระเงินให้เสร็จสิ้น หลังจากนั้นระบบ K-Payment Gateway จะแจ้งผลการชำระเงินให้ร้านค้าทราบต่อไป
การเชื่อมต่อวิธีนี้ใช้สำหรับการรับชำระด้วยบัตร UnionPay และการรับชำระเงินด้วย Alipay
- 4) Direct API เป็นการเรียก K-Payment Gateway API โดยตรง ซึ่งร้านค้าที่จะเชื่อมต่อแบบนี้ได้จะต้องได้รับการรับรองมาตรฐาน PCI DSS แล้วเท่านั้น (ศึกษารายละเอียดได้จากเว็บไซต์ของ PCI DSS ดังนี้ https://www.pcisecuritystandards.org/document_library)
การเชื่อมต่อวิธีนี้รองรับได้ทั้งการชำระเงินด้วยบัตรและการชำระเงินด้วย QR Code
- 5) Webview for Mobile Native App ใช้สำหรับร้านค้าที่พัฒนาแอปพลิเคชันสำหรับโทรศัพท์มือถือ โดยการเรียก Webview เพื่อแสดงหน้าจอให้ระบุรายละเอียดบัตรที่ต้องการใช้ในการชำระเงินหรือลงทะเบียนบัตร แล้วระบบ K-Payment Gateway จะดำเนินการเชื่อมต่อกับระบบที่เกี่ยวข้องและแจ้งผลการชำระเงินหรือลงทะเบียนบัตรให้ร้านค้าทราบต่อไป
การเชื่อมต่อวิธีนี้รองรับเฉพาะการชำระเงินด้วยบัตรแบบ Full MCC เท่านั้น

■ Token คืออะไร

ตอบ Token คือชุดข้อมูลเสมือนที่ได้จากการสุ่มโดยระบบเพื่อนำมาใช้แทนหมายเลขบัตรในการชำระเงิน เนื่องจาก การชำระค่าสินค้า/บริการออนไลน์จะต้องมีการกรอกข้อมูลส่วนตัวรวมถึงข้อมูลบัตรซึ่งเป็นข้อมูลที่เป็นความลับ ดังนั้นเพื่อความปลอดภัยของการทำธุรกรรม จึงต้องมีกระบวนการเปลี่ยนข้อมูลทั้งหมดของผู้ถือบัตรให้เป็นรูปแบบ Token ก่อนส่งต่อไปยังระบบการชำระค่าสินค้า/บริการ โดยที่ร้านค้าไม่ต้องเก็บข้อมูลใดๆ ลูกค้าเลย ซึ่งการถอด Token ให้กลับไปเป็นข้อมูลทำได้ยาก จึงปลอดภัยสูงทั้งต่อผู้ถือบัตรและร้านค้า



- ถ้าต้องการเก็บข้อมูลบัตรสำหรับให้ลูกค้าเลือก โดยไม่ต้องใส่เลขบัตรใหม่ทุกครั้ง ต้องทำอย่างไร
 - ตอบ** K-Payment Gateway มี Function "จดจำเลขบัตร" (Remember card) สำหรับร้านค้าที่มีระบบสมาชิกเพื่อช่วยอำนวยความสะดวกให้ลูกค้าผู้ถือบัตรที่ต้องการจ่ายเงินด้วยบัตรใบเดิมโดยไม่ต้องกรอกรายละเอียดบัตรซ้ำ ที่ร้านค้าต้องทำ ก็คือ เมื่อมีลูกค้าเลือก Remember Card พร้อมกับชำระเงินในครั้งแรก หลังจากที่ใช้ Token เรียก API เพื่อเก็บเงินแล้วก็ต้องสร้าง Customer Object เก็บเอาไว้ และครั้งต่อไปก็ใช้ข้อมูล Customer ในการติดตั้งปุ่ม K-Pay และเรียก Charge API ได้เลย ซึ่งรายละเอียดทั้งหมดอยู่ในหัวข้อ "จดจำเลขบัตร (Remember card)"

■ Notify URL คืออะไร

ตอบ Notify URL คือ URL ที่ร้านค้าจัดทำสำหรับรับผลการทำรายการแบบ Realtime โดยร้านค้าจะต้องพัฒนาระบบ RESTful API เพื่อรับการเรียก WebHook Notify API จาก K-Payment Gateway ผ่านทาง HTTP Request แบบ POST (ดูรายละเอียดที่ API Reference)

ร้านค้าจะต้องแจ้งข้อมูล Notify URL ให้ธนาคารทราบตอนสมัครบริการ K-Payment Gateway จึงจะสามารถรับผลการทำเงินแบบ Realtime ได้

■ Callback URL คืออะไร

ตอบ Callback URL คือ URL ที่ร้านค้าต้องจัดทำขึ้นเพื่อรับการทำรายการชำระเงินด้วยบัตรแบบ 3D Secure โดยทำหน้าที่ในการรับผลการทำรายการ หลังจากที่ธนาคารผู้ออกบัตรทำการตรวจสอบตัวตนผู้ถือบัตรและอนุมัติการทำเงินแล้ว ร้านค้าจะต้องแจ้งข้อมูล Callback URL ให้ธนาคารทราบตอนสมัครบริการ K-Payment Gateway จึงจะสามารถรับการทำเงินด้วยบัตรแบบ 3D Secure ได้ ในกรณีที่ต้องการเปลี่ยนแปลงข้อมูล Callback URL ร้านค้าสามารถทำได้ด้วยตนเองผ่าน Merchant Portal (User ที่มีสิทธิ์ในการแก้ไขข้อมูลนี้ จะต้องเป็น Level: Admin เท่านั้น)

■ Dynamic URL หรือ Redirect URL ที่ได้รับเมื่อรับการทำรายการชำระเงินด้วยบัตรแบบ 3D Secure คืออะไร

ตอบ การการทำรายการชำระเงินด้วยบัตรแบบ 3D Secure จะต้องมีขั้นตอนของการยืนยันตัวตนของลูกค้า ปัจจุบันก็คือหน้าให้ใส่ OTP ซึ่งการจะพาลูกค้าผู้ถือบัตรไปที่หน้ายืนยันตัวตนของธนาคารผู้ออกบัตรนั้น K-Payment Gateway จะสร้าง Dynamic URL ขึ้นมาหนึ่งชุดเพื่อความปลอดภัยแล้วส่งให้ร้านค้า ร้านค้าต้องซ่อม Redirect ลูกค้าไปที่ URL ดังกล่าว และเมื่อผ่านกระบวนการยืนยันตัวตนและชำระเงินแล้ว K-Payment Gateway จะส่งค่ากลับไปบอกให้ร้านค้าทราบผ่าน Callbak URL เพื่อให้ตรวจสอบรายละเอียดการทำรายการชำระเงินก่อนยืนยันผลการทำรายการให้ลูกค้าทราบ จึงจะถือว่าการทำรายการชำระเงินเสร็จสิ้น

■ จะทราบได้อย่างไรว่ารายการสำเร็จแล้ว

ตอบ การจะยืนยันผลการทำรายการจะต้องดูข้อมูลประกอบกัน 2 ส่วนคือค่า Transaction State และ Status ในกรณีที่รายการสำเร็จ ค่าใน transaction_state ต้องเป็น "Authorized" และค่าใน status ต้องเป็น "success" เท่านั้น

■ ในกรณีที่เข้มต้องแบบ Embedded UI จะสามารถ Customize ปุ่ม K-Pay ได้อย่างไร

ตอบ ร้านค้าสามารถสร้างปุ่มได้ด้วยตนเอง โดยมีขั้นตอนดังนี้

- 1) เพิ่ม Attribute `data-show-button="false"` ตอนผู้งปุ่ม K-Pay
- 2) สร้างปุ่มตามที่ต้องการ โดยเรียก Function `KPayment.show()` ที่ Event Onclick ของปุ่ม



ตัวอย่าง

```
<form method="POST" action="/checkout">
  <script type="text/javascript"
    src="https://dev-kpaymentgateway.kasikornbank.com/ui/v2/kpayment.min.js"
    data-apikey="pkey_prod_75677dushd74774ggd77d7dhsgfhfghfhgdf"
    data-amount="74.00"
    data-currency="THB"
    data-payment-methods="card"
    data-name="Your Shop Name"
    data-show-button="false" >
  </script>
  <input type="button" value="Pay Now" onclick="KPayment.show()">
</form>
```

- ในกรณีที่ใช้รูปแบบ Embedded UI จะทราบได้อย่างไรว่าลูกค้าปิดฟอร์มการชำระเงินของ K-Payment Gateway ก่อนทำการเสร็จสมบูรณ์

ตอบ ร้านค้าต้องเพิ่ม JavaScript ตามด้านล่างนี้เข้าไปพร้อมกับการฝังปุ่ม K-Pay เอาไว้

```
<script type="text/javascript">
  var merchantFunction = function(){
    // Add notify action.
    console.log("Close popup!");
  };
  KPayment.onClose(merchantFunction);
</script>
```

- ต้องการทดสอบระบบ ต้องทำอย่างไรบ้าง และใช้ข้อมูลอะไรในการทดสอบ

ตอบ สำหรับการทดสอบระบบก่อนใช้งานจริง ร้านค้าจะได้รับ Username และ Password สำหรับเข้าสู่ Merchant Portal (Sandbox) ซึ่งร้านค้าสามารถดูข้อมูลที่จำเป็นในการทำการทดสอบ เช่น Public Key หรือ Secret Key ได้ด้วยตนเอง และตรวจสอบรายการการทดสอบได้ทันที และเมื่อร้านค้าทดสอบระบบและพร้อมใช้งานจริงแล้ว สามารถกรอกใบสมัครบริการ K-Payment Gateway และส่งเอกสารประกอบการสมัครมายังธนาคารกสิกรไทย เมื่อธนาคารตรวจสอบข้อมูลและดำเนินการเปิดให้บริการให้ร้านค้าแล้ว ร้านค้าจะได้รับ Username และ Password สำหรับเข้าสู่ Merchant Portal และเริ่มรับชำระเงินได้ทันที