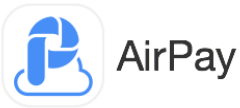
Payment API Specification & Integration Guide For Partners



AirPay (Thailand) Co., Ltd.

Version History

Version	Changes	Author	Date
1.0.0	+ Initial creation	Lei Lei	2014-09-29
1.0.1	+ Added get_landing_url API. - Removed API signature. . Changed all API calls to POST.	Lei Lei	2014-10-03
1.0.2	+ Added "from" parameter to get_landing_url API + Added "app" type of the "from" parameter of get_landing_url API. + Added procedure of calling get_landing_url in Mobile App integration workflow. + Added one more order_status: value 101.	Lei Lei	2014-10-07
1.0.3	. Renamed "from" to "source" in get_landing_url API.	Jon	2014-10-09
1.0.4	- Removed app-supplied parameters to return_url for security purpose. + Added instruction of QR Code generation.	Lei Lei	2014-10-10
1.0.5	+ Added order_status 102.	Lei Lei	2014-12-18
1.0.6	- Removed Sandbox order IDs. + Added Sandbox AppID and AppKey.	Lei Lei	2015-01-07
1.0.7	+ Added "msg" field to notify_order_status api.	Lei Lei	2015-03-11
1.0.8	+ Added optional "payment_id" field to get_order_status and notify_order_status api.	Jon	2015-06-26
1.0.9	+ Added order_status 501.	Jon	2015-08-24
1.1.0	+ Added "extra_data" field to order structure	Kwann	2015-10-13
1.2.0	+ Update SSL key	Kwann	2015-10-28
1.3.0	+ Migrate documents to new location, added examples	Kwann	2015-11-24
1.3.1	+ Add QRCode image API	Kwann	2016-01-15
1.3.2	+ Add Refund API	Kwann	2016-03-09

1.3.3	+ Add Update Order API	Kwann	2016-05-04

Table of Contents

1. Introduction
2. Platforms and Workflows
    - 2.1 PC Web
        - Usage Scenario
        - QR Code
        - Example
        - Workflow
    - 2.2 Mobile Web
        - Usage Scenario
        - Landing URL
        - Workflow
    - 2.3 Mobile App
        - Usage Scenario
        - Landing URL
        - Mobile SDK
        - Workflow
    - 2.4 Payment Procedure
3. Values and Data Structures
    - 3.1 order_id
    - 3.2 app_id
    - 3.3 app_key
    - 3.4 order_validity
    - 3.5 order_status
    - 3.6 order
    - 3.7 payment_api_domain
    - 3.8 payment_id
    - 3.9 result_code
4. API Definitions
    - 4.1 get_landing_url (Payment API)
        - Request Query Parameters
        - Response JSON Fields
        - Example
    - 4.2 get_order_status (Payment API)
        - Request Query Parameters
        - Response JSON Fields
        - Example
    - 4.3 get_order_details (Merchant Callback)
        - Request Body Parameters
        - Response JSON Fields
        - Example
    - 4.4 validate_order (Merchant Callback)
        - Request Body Parameters
        - Response JSON Fields
        - Example
    - 4.5 notify_order_status (Merchant Callback)

Request Body Parameters
Response JSON Fields
Example
4.6 refund_order (Payment API)
Request Query Parameters
Response JSON Fields
Example
4.7 update_order (Payment API)
Request Query Parameters
Response JSON Fields
Example
5. API Security
5.1 Encryption
5.2 API Secret Key
6. Sandbox Environment
6.1 Purpose
6.2 Accessing Method

1. Introduction

AirPay is a secure and convenient mobile payment service that allows smartphone users to make direct payment through their bank accounts, credit cards, or counter services.

The Payment API provides a way for any online merchants to collect payment without the hassle of integration with various banks and payment gateways. It serves as the one-stop payment hub for businesses without the resources of developing their own payment gateways.

AirPay serves as one of the payment channels of the merchant. It appears in the "Payment Method Selection" page when the customer checks out from the merchant.

Note: This document only contains the Server-to-Server APIs for PC Web and Mobile Web integrations. If you are integrating with the payment API from your Mobile App, please also refer to the Android App and iOS App API documents.

2. Platforms and Workflows

The payment API provides integration to the following platforms.

- PC Web
- Mobile Web (iOS and Android)
- Mobile App (iOS and Android)

2.1 PC Web

Usage Scenario

The merchant may opt to display a QR code on the Payment Method Selection page on the merchant's website. The QR code can be scanned using either the AirPay mobile app (recommended) or via an alternative scanner. The QR code will direct users to the LandingURL from which users would be prompted to complete their purchase.

QR Code

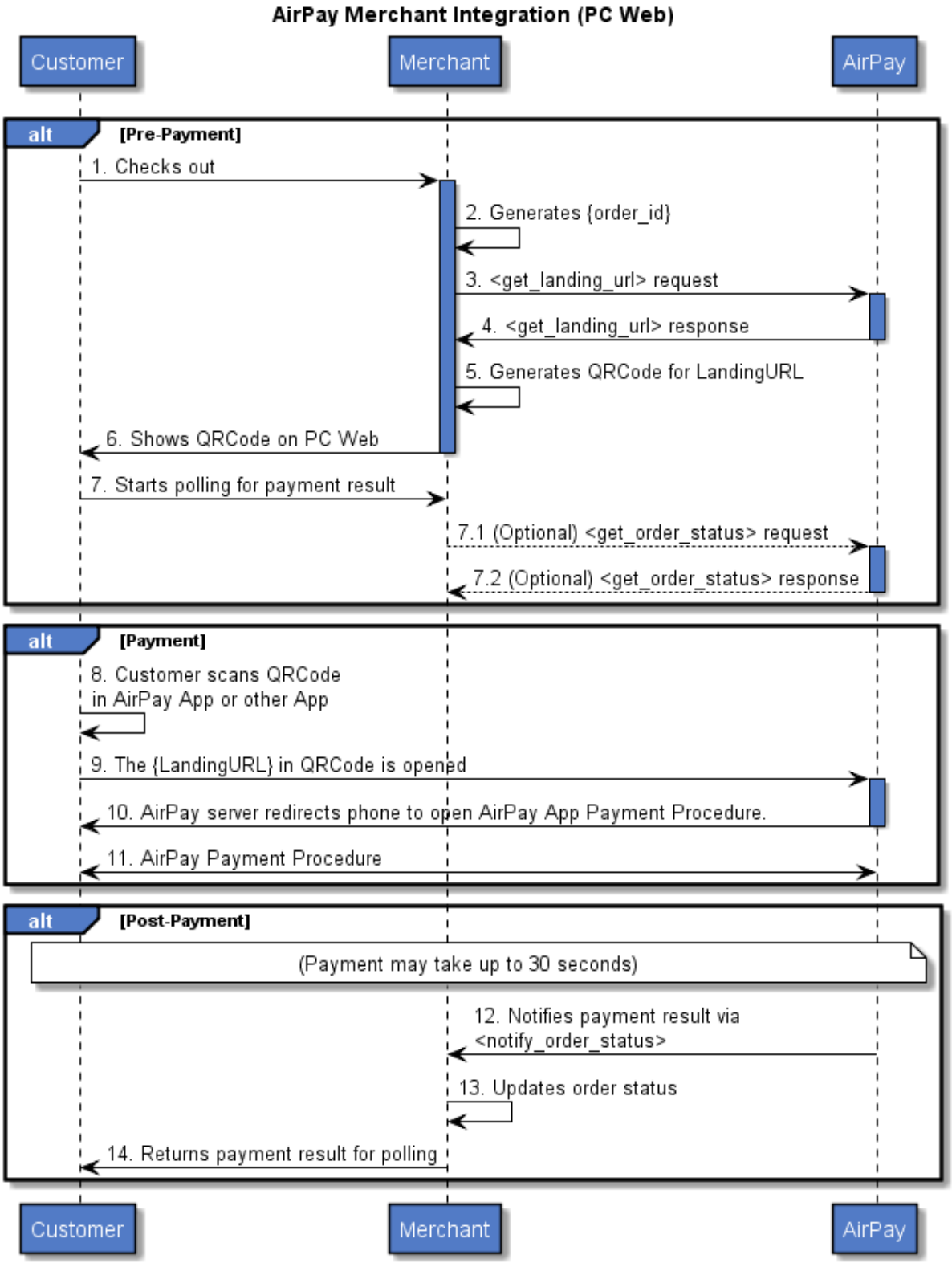
The QR Code is to be generated by the Merchant's website. The free Google Charts API could be used to generate a QR Code.

Example

Landing URL (provided by Payment API)
https://pay.airpay.in.th/pay?order_id=SBIPS_1448261830_311964342&source=qr&app_id=100001&key=nrqL56b9b2e1a9c15b88

QR Code Image URL
https://chart.googleapis.com/chart?chs=200x200&cht=qr&choe=UTF-8&chld=H&chl=https%3A%2F%2Fpay.airpay.in.th%2Fpay%3Forder_id%3DSBIPS_1448261830_311964342%26source%3Dqr%26app_id%3D100001%26key%3DnrqL56b9b2e1a9c15b88

Workflow


AirPay Merchant Integration (PC Web)
Customer
Merchant
AirPay
alt
[Pre-Payment]
1. Checks out
2. Generates {order_id}
3. <get_landing_url> request
4. <get_landing_url> response
5. Generates QRCode for LandingURL
6. Shows QRCode on PC Web
7. Starts polling for payment result
7.1 (Optional) <get_order_status> request
7.2 (Optional) <get_order_status> response
alt
[Payment]
8. Customer scans QRCode
in AirPay App or other App
9. The {LandingURL} in QRCode is opened
10. AirPay server redirects phone to open AirPay App Payment Procedure.
11. AirPay Payment Procedure
alt
[Post-Payment]
(Payment may take up to 30 seconds)
12. Notifies payment result via
<notify_order_status>
13. Updates order status
14. Returns payment result for polling
Customer
Merchant
AirPay

2.2 Mobile Web

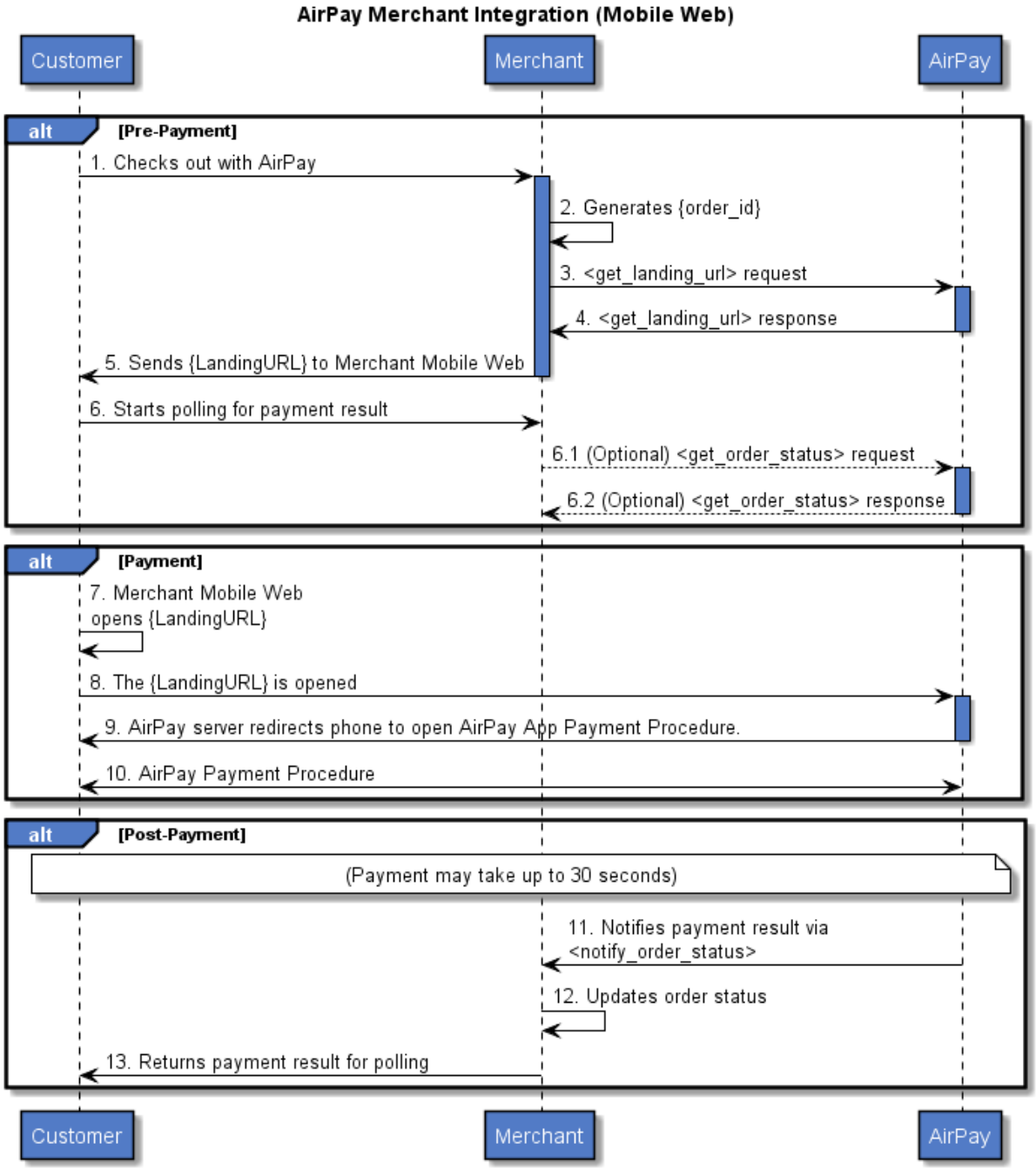
Usage Scenario

At the Payment Method Selection page on the merchant's web site (in a mobile web browser), after the customer has selected AirPay as the payment method, the merchant's web page should redirect the customer to the LandingURL. The landing page would prompt the user to open the AirPay App to complete the payment.

Landing URL

The Merchant's web site needs to open the LandingURL to launch the AirPay App.

Workflow


AirPay Merchant Integration (Mobile Web)
Customer
Merchant
AirPay
alt
[Pre-Payment]
1. Checks out with AirPay
2. Generates {order_id}
3. <get_landing_url> request
4. <get_landing_url> response
5. Sends {LandingURL} to Merchant Mobile Web
6. Starts polling for payment result
6.1 (Optional) <get_order_status> request
6.2 (Optional) <get_order_status> response
alt
[Payment]
7. Merchant Mobile Web opens {LandingURL}
8. The {LandingURL} is opened
9. AirPay server redirects phone to open AirPay App Payment Procedure.
10. AirPay Payment Procedure
alt
[Post-Payment]
(Payment may take up to 30 seconds)
11. Notifies payment result via <notify_order_status>
12. Updates order status
13. Returns payment result for polling
Customer
Merchant
AirPay

2.3 Mobile App

Usage Scenario

At the Payment Method Selection screen of the merchant's mobile app, after the customer has selected AirPay as the payment method, the merchant's mobile app opens the AirPay mobile app to make the payment, which will then redirect the customer back to the merchant's app after the payment procedure.

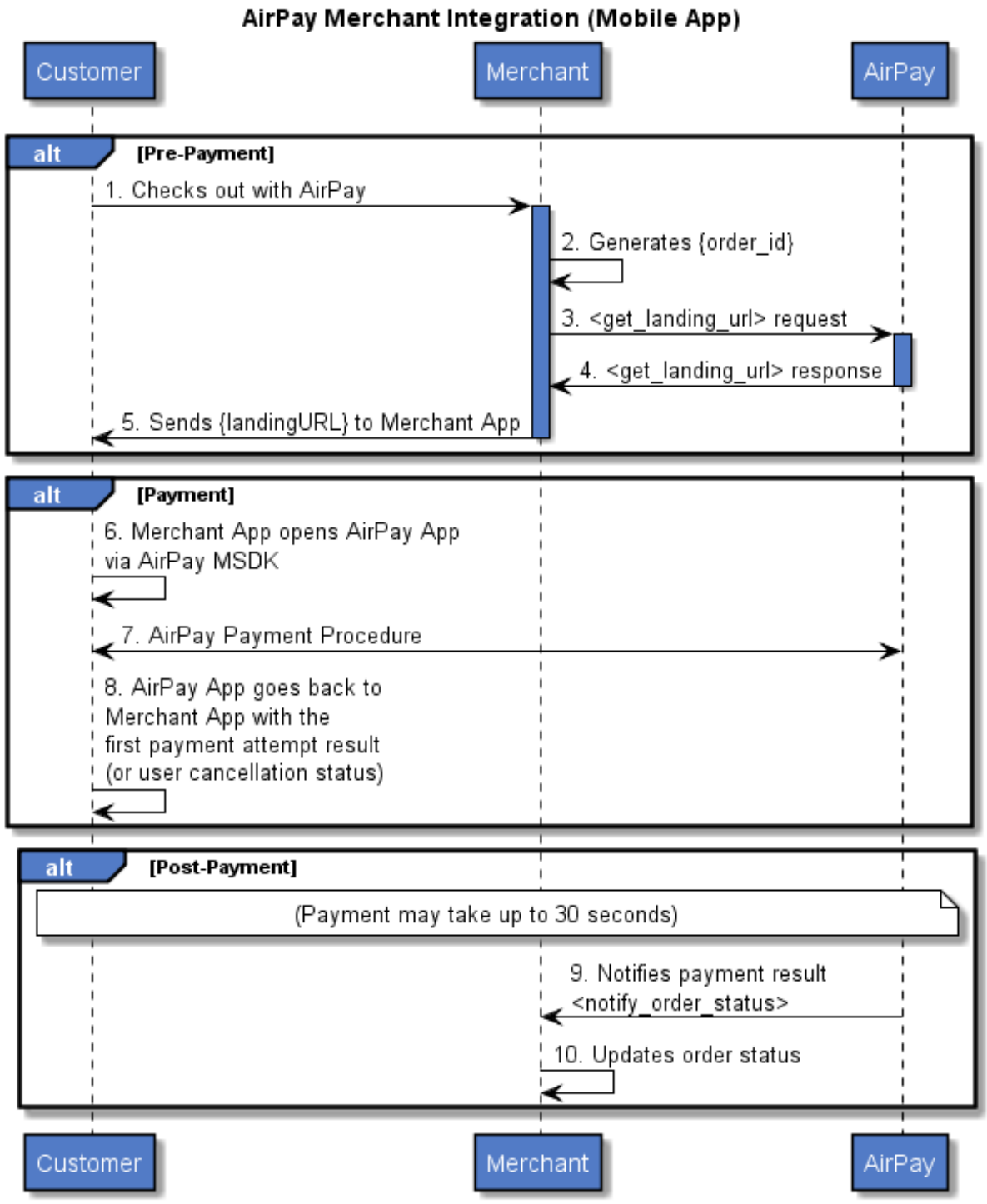
Landing URL

The Merchant's App needs to pass the LandingURL to the Mobile SDK to start the payment.

Mobile SDK

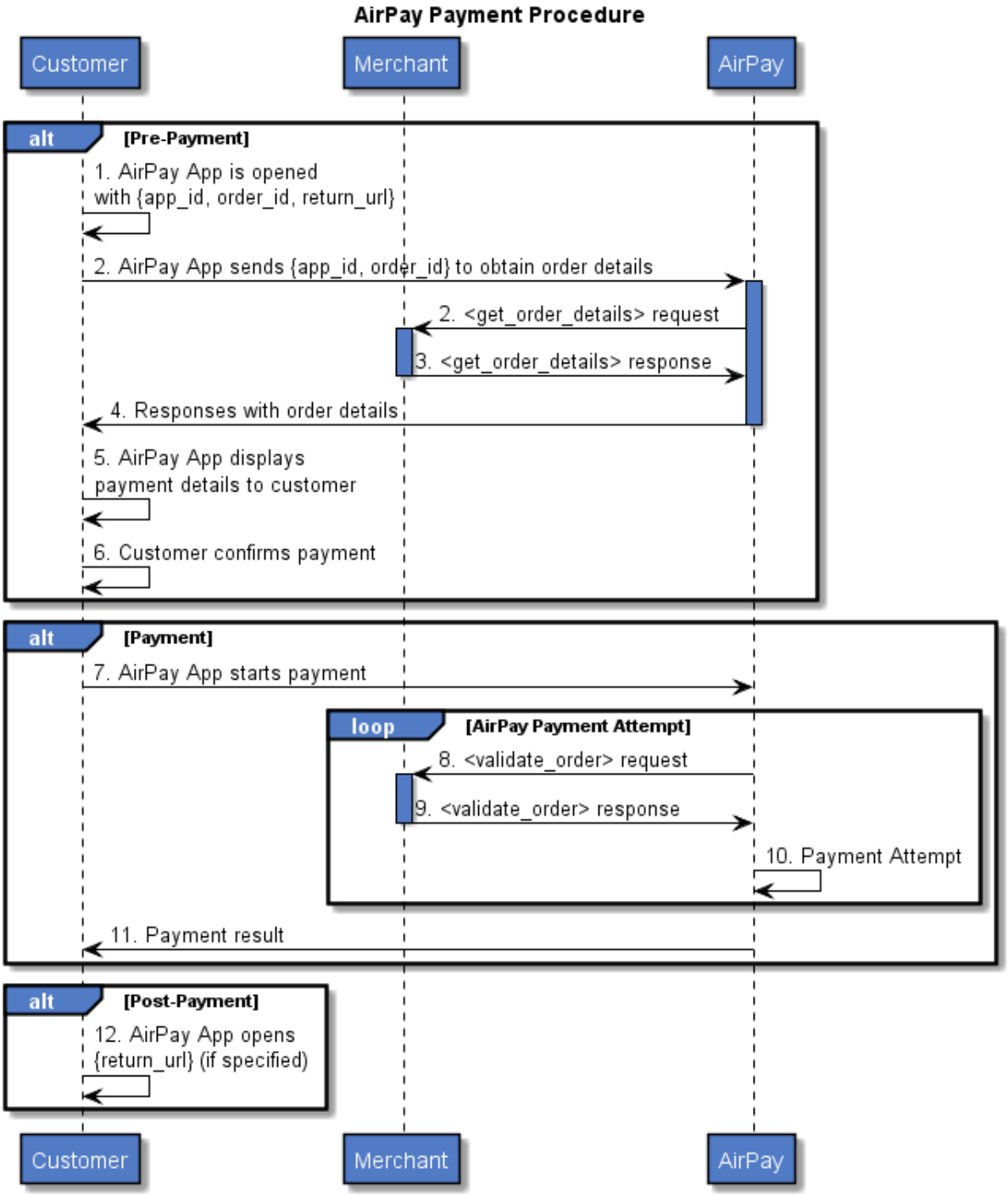
The Merchant App will need to integrate with the AirPay Mobile SDK to handle the inter-app communication.

Workflow


AirPay Merchant Integration (Mobile App)
Customer
Merchant
AirPay
alt
[Pre-Payment]
1. Checks out with AirPay
2. Generates {order_id}
3. <get_landing_url> request
4. <get_landing_url> response
5. Sends {landingURL} to Merchant App
alt
[Payment]
6. Merchant App opens AirPay App via AirPay MSDK
7. AirPay Payment Procedure
8. AirPay App goes back to Merchant App with the first payment attempt result (or user cancellation status)
alt
[Post-Payment]
(Payment may take up to 30 seconds)
9. Notifies payment result <notify_order_status>
10. Updates order status
Customer
Merchant
AirPay

2.4 Payment Procedure

This is a "zoomed-in" look of the "Payment Procedure" step of the above workflows.



3. Values and Data Structures

The following values and data structures are used by APIs.

- order_id
- app_id
- app_key
- order_validity
- order_status
- order
- payment_api_domain
- payment_id

3.1 order_id

Name	order_id
Type	string
Length	Up to 32 characters max
Description	Merchant-generated ID to identify an order. Must be unique in Merchant's system. Every payment in AirPay must be bond to an order in the Merchant's system.
Note	There are a set of special order_id prefixes used specifically for the Sandbox environment.

3.2 app_id

Name	app_id
Type	uint32
Description	ID to identify a Merchant. Two for each Merchant (one for Sandbox, one for Production).
Note	Please contact your AirPay account manager to obtain this ID.

3.3 app_key

Name	app_key
Type	string
Description	API key for a Merchant. Two for each Merchant (one for Sandbox, one for Production). Refer to Chapter 5 for details.
Note	Please contact your AirPay account manager to obtain this Key.

3.4 order_validity

Name	order_validity
Type	int32
Description	Validity of an order in the Merchant system.
Note	Values are in the table below. Note that once an order is created, it can only return the following status values.

Value	Description
0	<Reserved>. Do NOT return this value.
200	Order is valid
400	Order does not exist
401	Order has expired

3.5 order_status

Name	order_status
Type	int32
Description	Status of an order in the AirPay system.
Note	Values are in the table below.

Value	Description
0	<Reserved>. Do NOT return this value.
100	Payment in progress. Payment attempt has started, but not finished yet.
101	Order exists, but payment has not started yet (customer has not yet selected to pay.)
102	Order landing URL has been generated, but not yet opened in AirPay app by customer.
200	Money collected.
201	Money collected, but Merchant API (notify_order_status) not accessible. AirPay will retry calling Merchant again. (Note that this value is only used by get_order_status. It's never sent by notify_order_status.)
202	Customer has selected a "Counter Service" as the payment method. Note that this kind of payment methods requires the customer to personally go to the payment counter, therefore it may take minutes to hours before the money can be collected.
300	Order has been canceled by customer. No money was collected.
400	Order does not exist. No money was collected.
401	Order had expired while attempting to collect money. No money was collected.
500	Failed to collect money.
501	Money collected, but refunded, due to merchant cancellation.

3.6 order

Name	order
As HTTP Parameters	Expands to HTTP Query Parameters. Each field is one HTTP Query Parameter.
As JSON	An JSON Object. Each field is one JSON field.
Description	The complex data structure with details of an order.
Note	This data structure can be in different forms as stated in the "As" rows above. The actual form depends on the API definition.

Field Name	Field Type	Description
app_id	app_id	The Merchant ID given by AirPay to the Merchant.
order_id	order_id	The Order ID generated by the Merchant.
currency	string	Currency type. AirPay should use "THB"
payable_amount	uint64	Payment amount in the smallest currency unit. AirPay should use Thai Satang (1 THB = 100) Example: 123.50 THB = 12350
expiry_time	uint32	Expiry time for this order. It is the Unix timestamp in seconds (number of seconds from 1970/01/01 00:00:00 GMT).
item_name	string	Product's display name. This can be a brief product description to be displayed to use in AirPay App.
item_image	string	(Optional) Product image URL. Only PNG and JPEG are supported. This image will not be shown in large size, so a smaller image (not larger than 150x150) is preferred.
extra_data	string	(Optional) Additional information to tag with the order, JSON serialized.

3.7 payment_api_domain

Country	Domain
Thailand	airpay.in.th

3.8 payment_id

Name	payment_id
Type	string
Length	Up to 32 characters max
Description	ID generated by Payment API to identify an AirPay payment. Is unique in AirPay system. Every payment in AirPay must be bond to an order in the Merchant's system.
Note	-

3.9 result_code

Name	result_code
Type	int32
Description	Result of an operation
Note	Values are in the table below.

Value	Description
0	Success
1	API Key or ID is not valid
2	Order does not exist
3	Order was not refunded because it is outside of refund window
4	Order does not meet requirements for this operation
5	An unknown error occurred
6	Order is already refunded
7	An error occurred while updating the order
8	Unable to connect to merchant callback or merchant callback returned negative response

9	API is disabled for this merchant

4. API Definitions

4.1 get_landing_url (Payment API)

Caller	Merchant Server
Callee	AirPay Server
URL	https://api.<payment_api_domain>/pay/get_landing_url
Method	POST
Req Body	application/x-www-form-urlencoded
Resp Body	JSON
Description	The Merchant Server calls this API to get the landing URL for a specific order.

Request Query Parameters

Name	Type	Description
app_id	app_id	The Merchant ID provided by AirPay.
app_key	app_key	The Merchant Key provided by AirPay.
order_id	order_id	Order ID generated by Merchant.
source	string	'qr', 'qrcode', 'web', or 'app'. • qr: Returns URL that should be encoded into a QR code. Intended for merchants to generate their own QR codes. • qrcode: Returns URL pointing to a QR code image • web: Returns URL that should be opened by the user in the web browser • app: Returns URL that should be opened within the mobile app
return_url	string	Optional. The URL of Merchant's mobile webpage that will be opened by AirPay App after payment.

Response JSON Fields

Field Name	Field Type	Description
url	string	Landing URL to be used by Mobile Web and PC Web.

Example

Merchant Request (Mobile Web)
curl --data "app_id=[app_id]&app_key=[app_key]&order_id=[merchant_order_id]&source=web" "http://api.airpay.in.th/pay/get_landing_url"

4.2 get_order_status (Payment API)

Caller	Merchant Server
Callee	AirPay Server
URL	https://api.<payment_api_domain>/pay/get_order_status
Method	POST
Req Body	application/x-www-form-urlencoded
Resp Body	JSON
Description	The Merchant Server calls this API to get the status of a payment.

Request Query Parameters

Parameter Name	Parameter Type	Description
app_id	app_id	The Merchant ID given by AirPay to the Merchant.
app_key	app_key	The Merchant Key given by AirPay.
order_id	order_id	Original order ID generated by Merchant.

Response JSON Fields

Field Name	Field Type	Description
order_status	order_status	Status of the order in AirPay system.
payment_id	payment_id	The payment ID in AirPay system, if payment has been initiated.

Example

Merchant Request
curl --data "app_id=[app_id]&app_key=[app_key]&order_id=[merchant_order_id]" "http://api.airpay.in.th/pay/get_order_status"

4.3 get_order_details (Merchant Callback)

Caller	AirPay Server
Callee	Merchant Server
URL	https://<merchant_url_path>/get_order_details
Method	POST
Req Body	application/x-www-form-urlencoded
Resp Body	JSON
Description	The AirPay Server calls this API to obtain the details of an order.

Request Body Parameters

Parameter Name	Parameter Type	Description
app_key	app_key	The Merchant Key given by AirPay.
order_id	order_id	Original order ID generated by Merchant.

Response JSON Fields

Field Name	Field Type	Description
order_validity	order_validity	Validity of the order in Merchant's system.
order	order (as JSON)	Order details. If the order_id in the request does not exist in the Merchant's system, this object should be empty but the JSON key MUST exist.

Example

Merchant Callback Response
{"order_validity":200,"order":{"app_id":100001,"order_id":1,"currency":"THB","payable_amount":1000,"expiry_time":1448879325,"item_name":"Demo Item","item_image":"","extra_data":"{\"data\":\"additional data\"}"}}

4.4 validate_order (Merchant Callback)

Caller	AirPay Server
Callee	Merchant Server
URL	https://<merchant_url_path>/validate_order
HTTP Method	POST
Req Body	application/x-www-form-urlencoded
Resp Body	JSON
Description	The AirPay Server calls this API to validate an order before carrying out each payment attempt.

Request Body Parameters

Name	Type	Description
app_key	app_key	The Merchant Key given by AirPay.
order_id	order_id	Original order ID generated by Merchant.

Response JSON Fields

Name	Type	Description
order_validity	order_validity	Validity of the order in Merchant's system.

Example

Merchant Callback Response
{"order_validity":200}

4.5 notify_order_status (Merchant Callback)

Caller	AirPay Server
Callee	Merchant Server
URL	https://<merchant_url_path>/notify_order_status
HTTP Method	POST
Req Body	application/x-www-form-urlencoded
Resp Body	JSON
Description	The AirPay Server calls this API to update the final payment status to the Merchant Server. Notes about the final state: If a payment attempt fails and AirPay Server decides to try again, then this "fail" state will not be sent to the Merchant server. The AirPay Server only sends the state after it decides to no longer attempt to collect payment.

Request Body Parameters

Name	Type	Description
app_key	app_key	The Merchant Key provided by AirPay.
order_id	order_id	Order ID provided by merchant.
order_status	order_status	Status of the payment.
payment_id	payment_id	The payment ID in AirPay system, if payment has been initiated.

Response JSON Fields

Field Name	Field Type	Description
order_id	order_id	Original order_id in the request. This response value is only used to verify that the Merchant Server is working properly. If this response is not returned, or if the HTTP Response Code is not 200, then the AirPay Server will keep retrying this API until it succeeds.

msg	String	A text message to be shown to the user in the Order Receipt screen in AirPay app. The merchant can use this field to include deliverables such as PIN code. Note: Please do NOT send deliverables if order_status is not successful.

Example

Merchant Callback Response
{"order_id":100001,"msg":"Test Order"}

4.6 refund_order (Payment API)

Caller	Merchant Server
Callee	AirPay Server
URL	https://api.<payment_api_domain>/pay/refund_order
Method	POST
Req Body	application/x-www-form-urlencoded
Resp Body	JSON
Description	Merchant can request a refund on the completed order. A completed order can be refunded if the refund is requested within 30 days of the order creation. **Warning**: Refunds cannot be reversed.

Request Query Parameters

Parameter Name	Parameter Type	Description
app_id	app_id	The Merchant ID given by AirPay to the Merchant.
app_key	app_key	The Merchant Key given by AirPay.
order_id	order_id	Original order ID generated by Merchant.

Response JSON Fields

Field Name	Field Type	Description
code	result_code	A number representing the result of the operation.

Example

Merchant Request
curl --data "app_id=[app_id]&app_key=[app_key]&order_id=[merchant_order_id]" "http://api.airpay.in.th/pay/refund_order"

4.7 update_order (Payment API)

Caller	Merchant Server
Callee	AirPay Server
URL	https://api.<payment_api_domain>/pay/update_order
Method	POST
Req Body	application/x-www-form-urlencoded
Resp Body	JSON
Description	Used for updating item_name and extra_data information on an order. Order information would be obtained by AirPay via calling the get_order_details merchant callback. **Note**: This API is not required for regular payment procedure. Please consult AirPay before using this API

Request Query Parameters

Parameter Name	Parameter Type	Description
app_id	app_id	The Merchant ID given by AirPay to the Merchant.
app_key	app_key	The Merchant Key given by AirPay.
order_id	order_id	Original order ID generated by Merchant.

Response JSON Fields

Field Name	Field Type	Description
code	result_code	A number representing the result of the operation.

Example

Merchant Request
curl --data "app_id=[app_id]&app_key=[app_key]&order_id=[merchant_order_id]" "http://api.airpay.in.th/pay/update_order"

5. API Security

Security of the Payment API is ensured by both encryption and authentication procedures as described in this section.

5.1 Encryption

The Payment API uses industry-standard HTTPS protocol for server identity verification and data encryption.

Save the content of the following table to a file and use in your server code.
Note: Please use the appropriate certificate for the relevant service.

AirPay
-----BEGIN CERTIFICATE----- MIIDVDCCAjygAwIBAgIDAjRWMA0GCSqGSIb3DQEBBQUAMEIxCzAJBgNVBAYTAlVT MRYwFAYDVQQKEw1HZW9UcnVzdCBJbmMuMRswGQYDVQQDExJHZW9UcnVzdC BHbG9i YWwgQ0EwHhcNMDIwNTIxMDQwMDAwWhcNMjIwNTIxMDQwMDAwWjBCMQswCQY DVQQG EwJVUzEWMBQGA1UEChMNR2VvVHJ1c3QgSW5jLjEbMBkGA1UEAxMSR2VvVHJ1c3 Qg R2xvYmFsIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2swYYzD9 9BcjGlZ+W988bDjkcbd4kdS8odhM+KhDtgPpTSEHCIjaWC9mOSm9BXiLnTjoBbdq fnGk5sRgprDvgOSJKA+eJdbtg/OtppHHmMlCGDUUna2YRpIuT8rxh0PBFpVXLVDv iS2Aelet8u5fa9IAjbkU+BQVNdnARqN7csiRv8lVK83Qlz6cJmTM386DGXHKTubU 1XupGc1V3sjs0l44U+VcT4wt/lAjNvxm5suOpDkZALeVAjmRCw7+OC7RHQWa9k0+ bw8HHa8sHo9gOeL6NlMTOdReJivbPagUvTLrGAMoUgRx5aszPeE4uwc2hGKceeoW MPRfwCvocWvk+QIDAQABo1MwUTAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBTA ephojYn7qwVkDBF9qn1luMrMTjAfBgNVHSMEGDAWgBTAephojYn7qwVkDBF9qn1l uMrMTjANBgkqhkiG9w0BAQUFAAOCAQEANeMpauUvXVSOKVCUn5kaFOSPeCpilKIn Z57QzxpeR+nBsqTP3UEaBU6bS+5Kb1VSsyShNwrrZHYqLizz/Tt1kL/6cdjHPTfS tQWVYrmm3ok9Nns4d0iXrKYgjy6myQzCsplFAMfOEVEiIuCl6rYVSAlk6l5PdPcF PseKUgzbFbS9bZvlxrFUaKnjaZC2mqUPuLk/IH2uSrW4nOQdtqvmlKXBx4Ot2/Un hw4EbNX/3aBd7YdStysVAq45pmp06drE57xNNB6pXE0zX5IJL4hmXXeXxx12E6nV 5fEWCRE11azbJHFwLJhWC9kXtNHjUStedejV0NxPNO3CBWaAocvmMw== -----END CERTIFICATE----- -----BEGIN CERTIFICATE----- MIIEJTCCAw2gAwIBAgIDAjp3MA0GCSqGSIb3DQEBCwUAMEIxCzAJBgNVBAYTAlVT MRYwFAYDVQQKEw1HZW9UcnVzdCBJbmMuMRswGQYDVQQDExJHZW9UcnVzdC BHbG9i YWwgQ0EwHhcNMTQwODI5MjEzOTMyWhcNMjIwNTIwMjEzOTMyWjBHMQswCQYDV QQG EwJVUzEWMBQGA1UEChMNR2VvVHJ1c3QgSW5jLjEgMB4GA1UEAxMXUmFwaWRT

U0wg
U0hBMjU2IENBIC0gRzMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC
v
VJvZWF0eLFbG1eh/9H0WA//Qi1rkjqfdVC7UBMBdmJyNkA+8EGVf2prWRHzAn7Xp
SowLBkMEu/SW4ib2YQGRZjEiwzQ0Xz8/kS9EX9zHFLYDn4ZLDqP/oIACg8PTH2lS
1p1kD8mD5xvEcKyU58Okaiy9uJ5p2L4KjxZjWmhxgHsw3hUEv8zTvz5IBVV6s9cQ
DAP8m/0Ip4yM26eO8R5j3LMBL3+vV8M8SKeDaCGnL+enP/C1DPz1hNFTvA5yT2AM
QriYrRmIV9cE7Ie/fodOoyH5U/02mEiN1vi7SPIpyGTRzFRIU4uvt2UevykzKdkp
YEj4/5G8V1jlNS67abZZAgMBAAGjggEdMIIBGTAfBgNVHSMEGDAWgBTAephojYn7
qwVkDBF9qn1luMrMTjAdBgNVHQ4EFgQUw5zz/NNGCDS7zkZ/oHxb8+Ily1kwEgYD
VR0TAQH/BAgwBgEB/wIBADAOBgNVHQ8BAf8EBAMCAQYwNQYDVR0fBC4wLDAqo
Cig
JoYkaHR0cDovL2cuc3ltY2luY29tL2NybHMvZ3RnbG9iYWwuY3JsMC4GCCsGAQUF
BwEBBCIwIDAeBggrBgEFBQcwAYYSaHR0cDovL2cuc3ltY2QuY29tMEwGA1UdIARF
MEMwQQYKYIZIAYb4RQEHNjAzMDEGCCsGAQUFBwIBFiVodHRwOi8vd3d3Lmdlb3R
y
dXN0LmNvbS9yZXNvdXJjZXMvY3BzMA0GCSqGSIb3DQEBCwUAA4IBAQCjWB7GQzK
s
rC+TeLfqrlRARy1+eI1Q9vhmrNZPc9ZE768LzFvB9E+aj0I+YK/CJ8cW8fuTgZCp
fO9vfm5FlBaEvexJ8cQO9K8EWYOHDyw7l8NaEpt7BDV7o5UzCHuTcSJCs6nZb0+B
kwwHtnm8hEqddwnxxYny8LScVKoSew26T++TGezvfU5ho452nFnPjJSxhJf3GrkH
uLLGTxN5279PURt/aQ1RKsHWFf83UTRlUfQevjhq7A6rvz17OQV79PP7GqHQyH5O
Zl3NjGFVkP46yl0lD/gdo0p0Vk8aVUBwdSWmMy66S6VdU5oNMOGNX2Esr8zvsJmh
gP8L8mJMcCaY
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIE1DCCA7ygAwIBAgISESF/duG7H12FLtBzxjKhMj3wMA0GCSqGSIb3DQEBCwUA
MEwxCzAJBgNVBAYTAkJFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMSIwIAY
D
VQQDExlBbHBoYVNTTCBDQSAtIFNIQTI1NiAtIEcyMB4XDTE0MTAwMTA0MTQ1NloX
DTE5MTAwMTA0MTQ1NlowPDEhMB8GA1UECxMYRG9tYWluIENvbnRyb2wgVmFsaWRh
dGVkMRcwFQYDVQQDDA4qLmFpcnBheS5pbi50aDCCASIwDQYJKoZIhvcNAQEBBQAD
ggEPADCCAQoCggEBAMDqltr+f3kOHXCPJCPRCxoRMCvE1JnT/pfuaZ2o+5p079ID
GHl2B5vr8gom2NWOYO6E/3IQXS8+/mrsHs0BI0bmVW6+wKvrv7hF2qNkGCZ1dMBx
qkqmfmzXrLDhn7OlC0VdD3Spv8er5kharoah41a04yC5r5c+gSfwqUYxbwEY5iKt
zQ8p2LEszL/AXkV8ImoWHmTZJhkNVqrNm/PySHjygBXgJVYwflkms7xitExNZ/ia
GQQn0x6NliGHvlELn6FqIJ1FDLTGITSOeV8nT+Ov3ll0UUEW2jeILC41NkX7gfgL
qYO1FBar0taNI837KrWwv1oQ4k3R9PO/Jchnb7UCAwEAAaOCAb4wggG6MA4GA1Ud
DwEB/wQEAwIFoDBJBgNVHSAEQjBAMD4GBmeBDAECATA0MDIGCCsGAQUFBwIB
FiZo
dHRwczovL3d3dy5nbG9iYWxzaWduLmNvbS9yZXBvc2l0b3J5LzAnBgNVHREEIDAe
gg4qLmFpcnBheS5pbi50aIIMYWlycGF5LmluLnRoMAkGA1UdEwQCMAAwHQYDVR0l
BBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMD4GA1UdHwQ3MDUwM6AxoC+GL
Wh0dHA6
Ly9jcmwyLmFscGhhc3NsLmNvbS9ncy9nc2FscGhhc2hhMmcyLmNybDCBiQYIKwYB
BQUHAQEEfTB7MEIGCCsGAQUFBzAChjZodHRwOi8vc2VjdXJlMi5hbHBoYXNzbC5j

b20vY2FjZXJ0L2dzYWxwaGFzaGEyZzJyMS5jcnQwNQYIKwYBBQUHMAGGKWh0dHA6
Ly9vY3NwMi5nbG9iYWxzaWduLmNvbS9nc2FscGhhc2hhMmcyMB0GA1UdDgQWBBT
x
aoofFIetTTjwvWWz0eiL53ORrzAfBgNVHSMEGDAWgBT1zdU8CFD5ak86t5faVoPm
adJo9zANBgkqhkiG9w0BAQsFAAOCAQEAJlp12Ms9xllU5GIzlVcP6rTFgNu3o9ZW
jdTHP8lygLniI8D7zOpWrqFIdsUiz2zl35XOUsEU4Nwgj5gyx49rUuzO2d7+cAbF
VkwrhX9Sh0PKii0D9jAnEQEOGV9iJ2+NBm66gLZ6kJo6AXLgoJHS6GeFvZiFdAZA
5dYvQHSW4fgOTKjlUQETWncxq/opL0PgsJsdqk8YKiUEM4NELbsYoh0lOTvkmZNW
W6tDAwQtU2oRYMJHALCKdJ8bisvc0BtIlriAE83kxiKQovOtPvhNS8UYLFzI6ctJ
7XHEgBWPo2IwtmXhD7ba6mZ1b2uE79Vu0zzBMZcfJeQ+nY+I6MLQOA==
-----END CERTIFICATE-----

5.2 API Secret Key

Two Secret Keys (one for Sandbox, one for Production) are created for each merchant. Both AirPay and the Merchant knows this Secret Key for this specific Merchant. This is the "shared secret". For each API call, the caller (both AirPay or Merchant) appends this secret key in the POST parameters (app_key), and the callee validates the legitimacy of the caller by validating this key.

The merchant can contact AirPay customer service to obtain the Secret Key.

6. Sandbox Environment

6.1 Purpose

To help merchants to develop and test the integration, AirPay has set up a Sandbox environment with the exact same procedure as the production environment but without actually deducting money from the associated testing accounts. The merchant should integrate and test with the Sandbox first, before releasing the service to actual customers.

6.2 Accessing Method

The merchant uses the Sandbox app_id + app_key to access the Sandbox. The orders in Sandbox will NOT deduct money from the AirPay account.