Apple Pay

About Apple Pay

What is Apple Pay

Logo	
Info	Apple Pay is a digital wallet for storing payment details, providing an easy and secure way to pay in iOS applications, websites running on Safari browser and contactless POS terminals. In applications and on websites, users can quickly and securely provide their payment, shipping, and contact information to check out with just one touch using Apple's Touch ID. Apple Pay's simplicity increases conversion rates and new user adoption that come with it.
Type	Payments by eWallet

Computop's Apple Pay solution supports following scenarios:

- In-application payments (also part of the Computop SDK)
- Web payments

Notice: Regarding the liability shift in case of the card processing, please contact your Acquirer directly. Because Computop is not a direct contract partner, your Acquirer should specify the liability shift rules for each card brand.

Notice: For further information about the button solution for Apple Pay, please see https://support.computop.com/hc/en-gb/articles/4559076653970-Apple-Pay-und-Google-Pay-button-solution-PaySSL-aspx?source=search.

Web payment

Apple Pay on the Web enables purchases while using Safari web browser. For shopping on iPhone or iPad, after selecting checkout with Apple Pay, a payment sheet appears, prompting the customer to confirm payment via Touch ID. For shopping on Mac, customers need to have an iOS device in close range, and they'll be prompted on that device to authorize the payment, which will then synchronize to the browser. The latest Mac Book Pro allows customers to authorize payment directly on the Mac Book.

Apple Pay requirements:

- iOS 10. Apple Pay Web payments are supported on all iOS devices with a Secure Element.
- macOS 10.12. Apple Pay Web payments are supported in Safari.
  - The user must have an iPhone or Apple Watch that can authorize the payment.
  - On the latest Mac Book Pro payments can be authorized directly via Touch ID
- All pages that incorporate Apple Pay must be served over HTTPS.
- Your website must comply with the Apple Pay guidelines. For more information, see Apple Pay on the Web Acceptable Use Guidelines
- Server requirements

In-app payment

Apple Pay can be used for payments within the iOS applications. The main advantage of Apple Pay is that it is frictionless process with no need to re-type credit card data, shipping or billing address, which can be cumbersome on the smaller iPhone screens. Customer just choose to checkout with Apple Pay and confirms the payment with Touch ID.

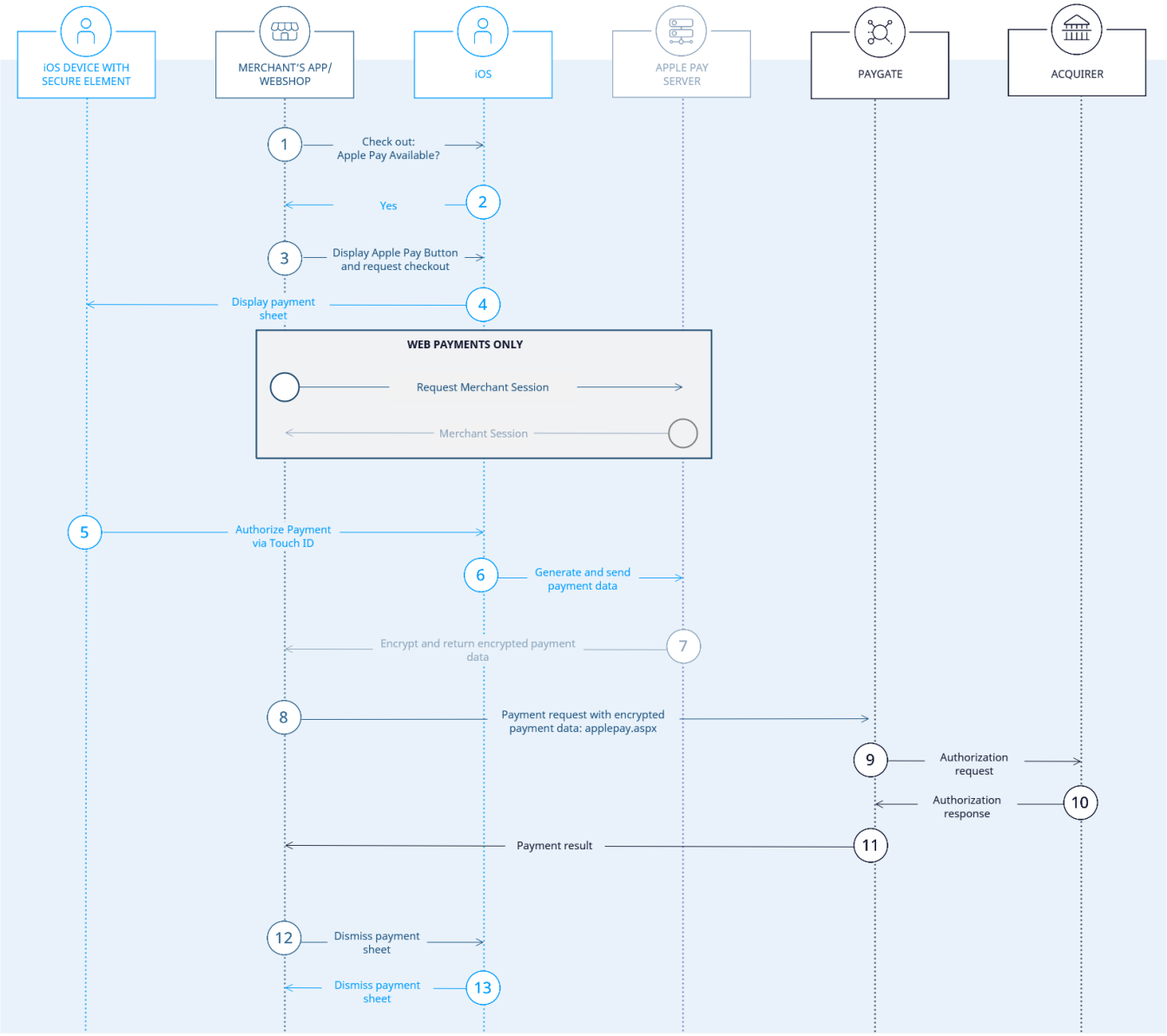
Merchants can also use Computop's SDK for facilitating the in-app payment integration, making it easier and faster.

Apple Pay requirements:

- Apple Pay in-app payments are supported on all iOS devices with a Secure Element.

On this page

Process flow chart



Apple Pay process flow

Presenting the Apple Pay button

Within apps, PassKit provides the API's that your app will use to determine if it is running on a device with a Secure Element and if the device has been provisioned with payment cards that you support. On websites, WebKit provides APIs that allow your website to check if the individual has an Apple Pay capable device and if it is set up.

If the device is Apple Pay enabled you should present the Buy with Apple Pay button using APIs supplied within PassKit within apps and Webkit within Safari.

Presenting the Payment Sheet

When your user selects goods or services to buy, and selects Apple Pay as the payment method, you create a payment request and ask PassKit in apps or Webkit on your website to present the payment sheet to the user. The payment sheet must immediately follow the user tapping the Apple Pay button, without any interim screens or pop-ups except to prompt for necessary product details, such as size or quantity.

Your app specifies the contents of the payment sheet but it does not control the user's interaction with the sheet. You must decide if it makes sense to present shipping and billing information, shipping method, and other line items to the user. You should only request the information necessary to process the transaction.

Processing Payments

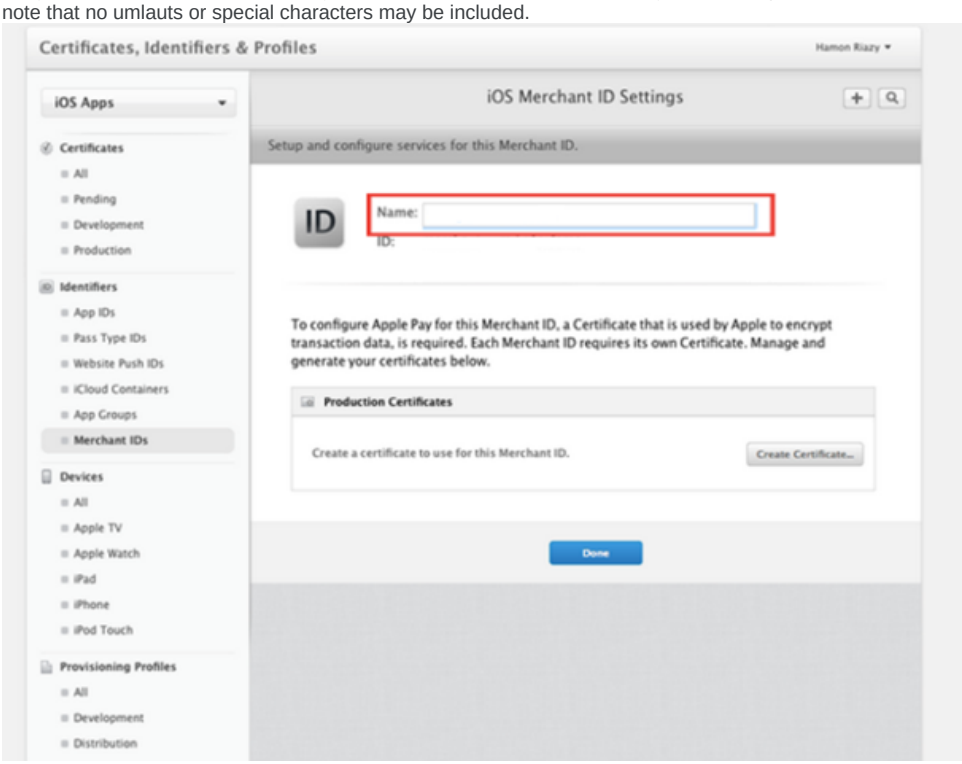
Once authorized by the user with Touch ID (and bank PIN code in China), your app receives a payment token from PassKit. The payment token encapsulates the information needed to complete a payment transaction, including the device-specific account number, the amount, and a unique, one-time-use cryptogram. The encrypted payment bundle will be decrypted by Computop, which then handles the whole processing of the transaction.

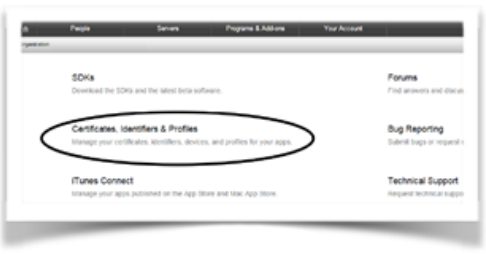
Step by Step Set-Up Guide for Apple Pay

Once you have decided to support Apple Pay, there is a set of steps that need to be completed:

1. Validate if the Acquirer supports Apple Pay
    a. Not all Acquirers are supporting Apple Pay for in-app and web payments
    b. Computop can provide information about supporting Acquirers

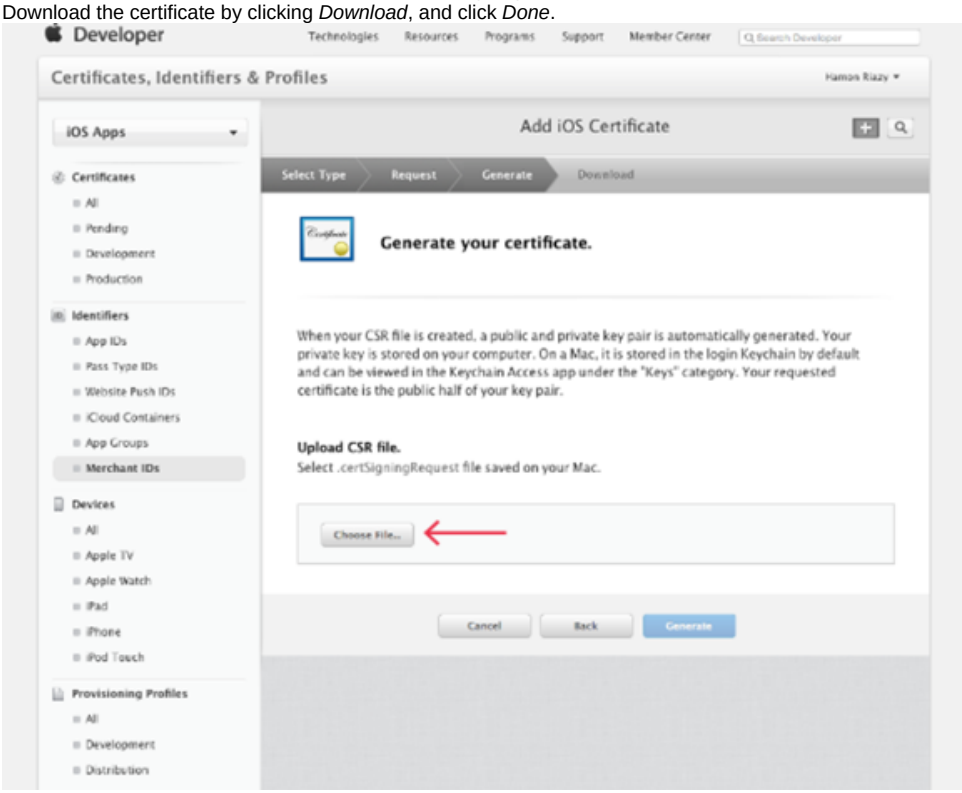
2. Create **Merchant ID**, unique identifier that represents a merchant for Apple Pay
    a. Access your Apple Developer Account and visit the *Certificates, Identifiers & Profiles* section of the Apple member center. You can then navigate to *Merchant IDs* and click *add*.
    b. Set your Merchant ID. The identifier you enter should be a reverse DNS style identifier prefixed with the word "merchant". Please note that no umlauts or special characters may be included.

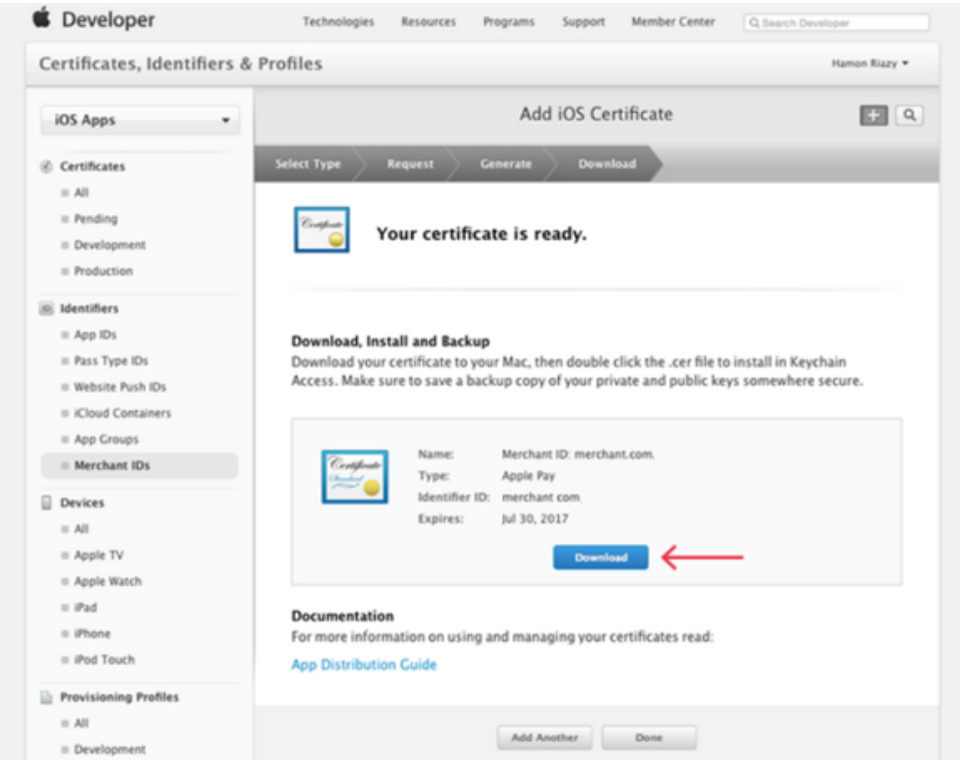




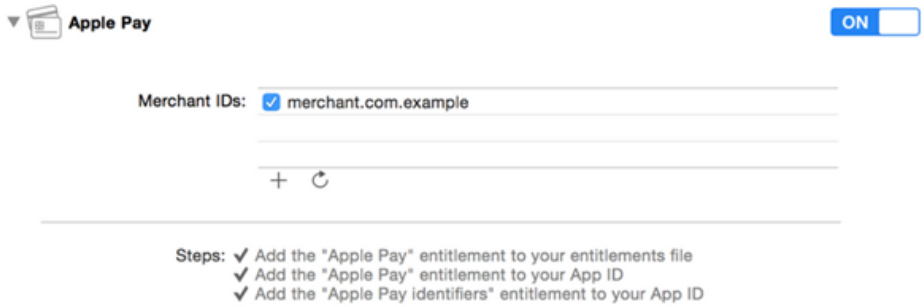
    c. Send your Apple Merchant ID to Computop.

3. Create **Payment Processing Certificate**, used to securely transfer payment data. Apple Pay servers use the payment processing certificate' s public key to encrypt the payment data.
    a. Computop creates Certificate Signing Request (CSR).
        i. With your Apple Merchant ID Computop Helpdesk creates the Apple Pay CSR and provides you with this information.
        ii. CSR will be needed by Apple to generate actual certificate for signing Payment Token
    b. Upload CSR at Apple
        i. In Member Center, select *Certificates, Identifiers & Profiles*. Under *Identifiers*, select *Merchant IDs*. Select the merchant ID from the list, and click *Edit*.
        ii. Click *Choose File*, select the CSR received from Computop and click *Generate*.
        iii. Download the certificate by clicking *Download*, and click *Done*.





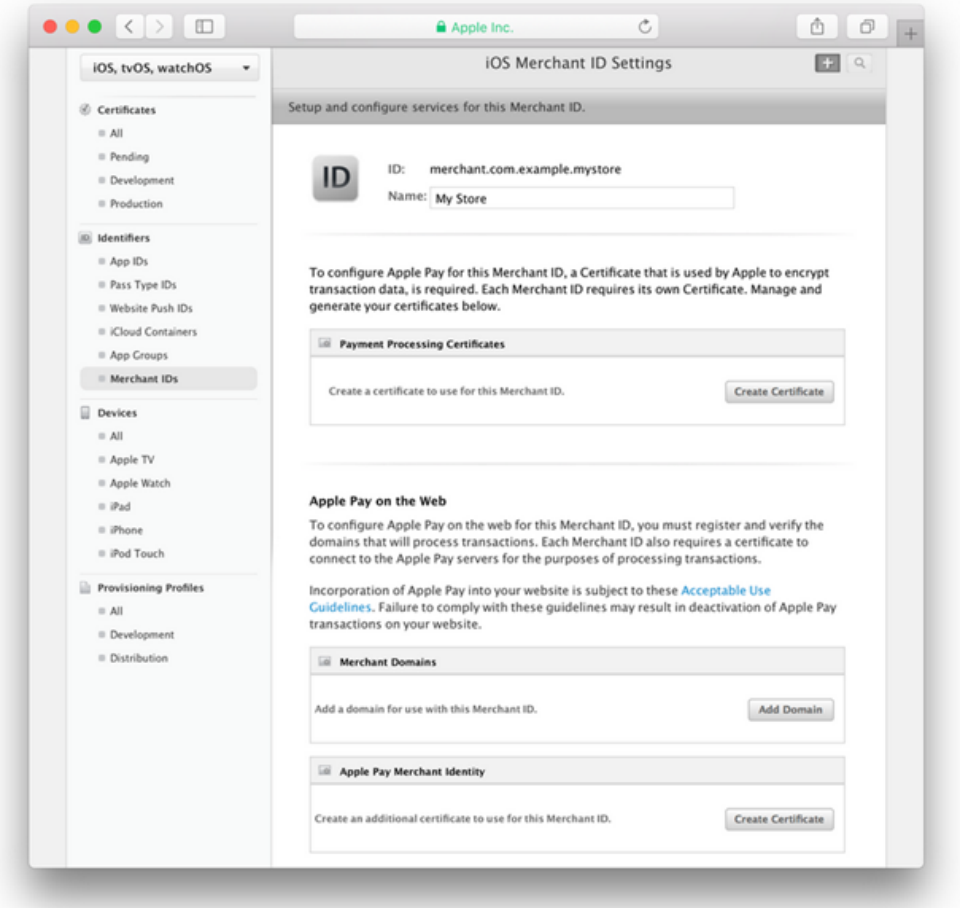
4. Enable Apple Pay for your app in **Xcode**
    a. Open *Capabilities* pane
    b. Select the switch in the *Apple Pay* row, and then select the merchant IDs you want the app to use.



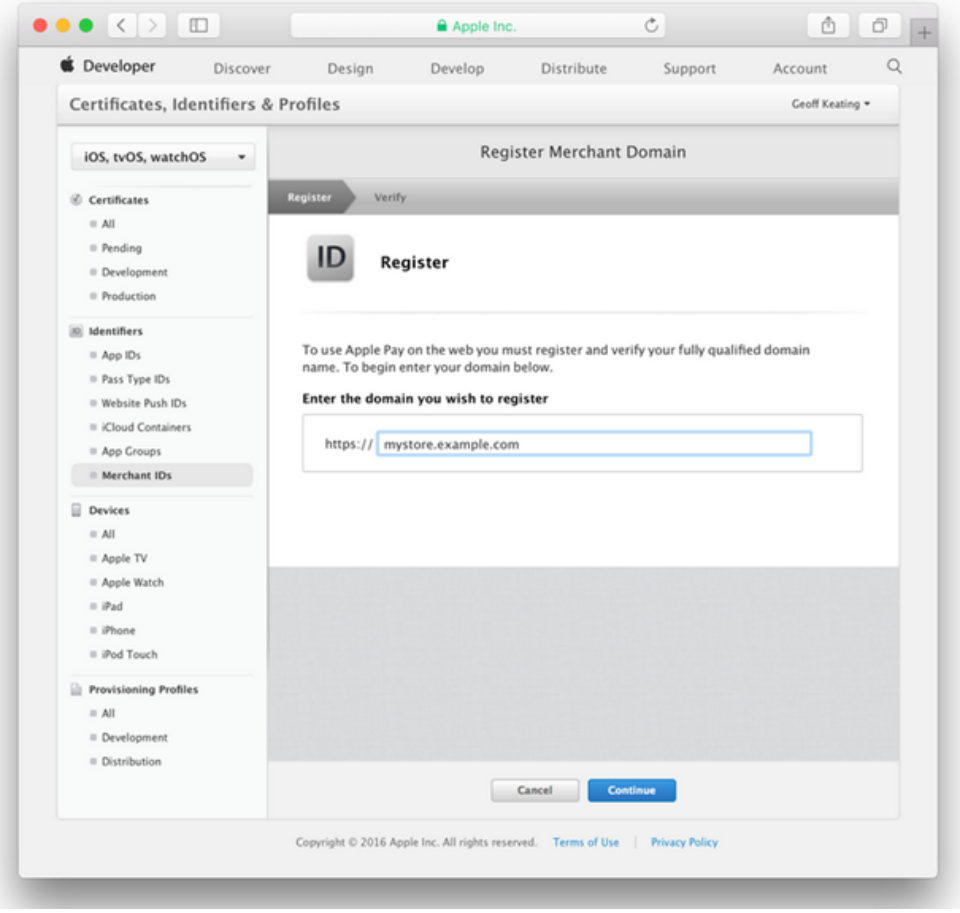
5. (**WEB ONLY**) Create **Merchant Identity Certificate**. A Transport Layer Security (TLS) certificate used to authenticate your merchant sessions with the Apple Pay servers.

    To register and verify your domain, and create your Merchant Identity Certificate:

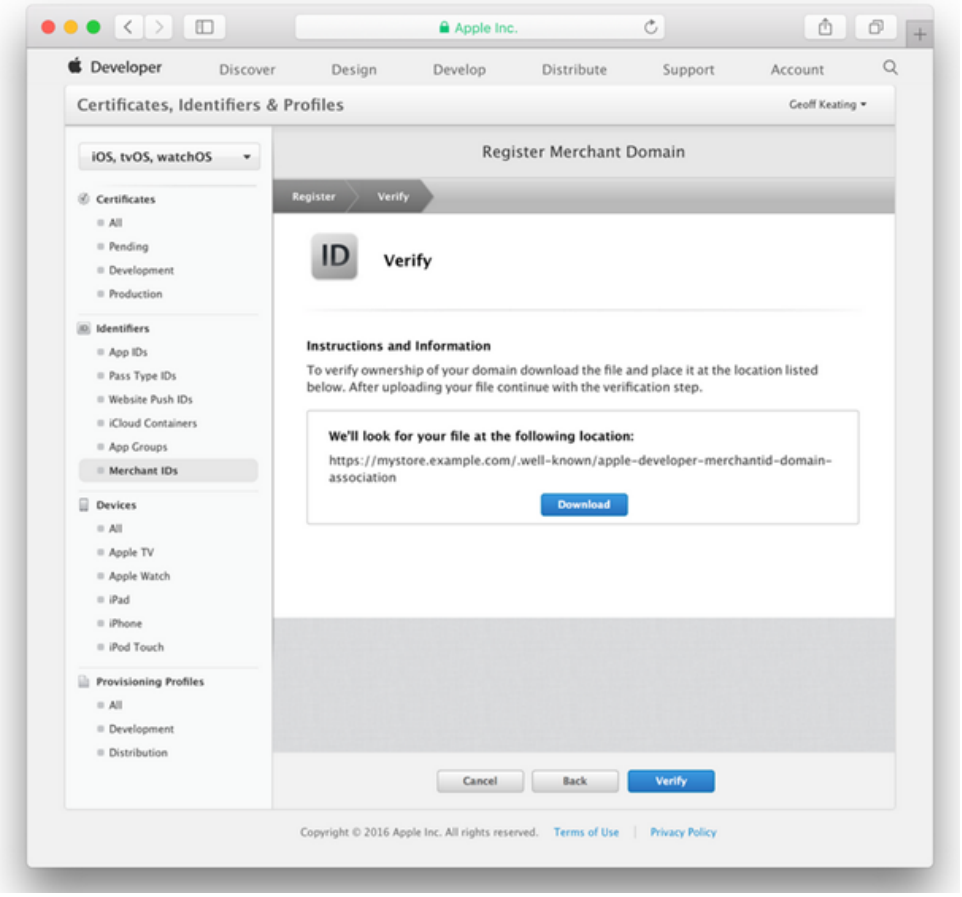
    a. In Member Center, select *Certificates, Identifiers and Profiles*.
    b. Under *Identifiers*, select *Merchant IDs*.
    c. Select the merchant ID from the list, and click *Edit*.
    d. In the *Apple Pay on the Web* section, click the *Add Domain*



e. Enter your fully qualified domain name, and click *Continue*.



f. The site creates a file and makes it available to download. Download this file and host it at the provided location.



g. As soon as the file is available on your server, click the Verify button. If the verification succeeds, the site returns to the iOS Merchant ID Settings page and shows a green *Verified* status label.
h. In the *Apple Pay on the Web* section, click the *Create Certificate* Follow the instructions to create and download your Merchant Identity Certificate. Use this certificate when requesting a merchant session during payment validation. For more information, see Merchant Validation in ApplePaySession.

6. Integrate Apple Pay at your application or web shop
   a. Follow guidelines: https://developer.apple.com/apple-pay
   b. **Important**: When creating PKPaymentRequest, request.merchantCapabilities = PKMerchantCapability3DS. Computop only supports 3DS types.

Paygate interface

Definitions

Data formats

Format	Description
a	alphabetical
as	alphabetical with special characters
n	numeric
an	alphanumeric
ans	alphanumeric with special characters
ns	numeric with special characters
bool	boolean expression (true or false)

3	fixed length with 3 digits/characters
..3	variable length with maximum 3 digits/characters
enum	enumeration of allowed values
dttm	ISODateTime (YYYY-MM-DDThh:mm:ss)

Abbreviations

Abbreviation	Description	Comment
CND	condition	
M	mandatory	If a parameter is mandatory, then it must be present
O	optional	If a parameter is optional, then it can be present, but it is not required
C	conditional	If a parameter is conditional, then there is a conditional rule which specifies whether it is mandatory or optional

Notice: Please note that the names of parameters can be returned in upper or lower case.

Calling the interface

Two transactions are created when making a credit card payment via Apple Pay. In the Apple Pay transaction the required credit card data are determined first and the actual credit card transaction is then carried out automatically. This takes place via a server-to-server connection and supports all usual credit card transaction options. The difference here, however, is that you don't transmit the credit card information, which you don't know. Instead, you transmit the PKPaymentToken generated by Apple Pay, which contains the required credit card data in encrypted form.

In order to make a credit card payment with Apple Pay, please use the following URL:

https://www.computop-paygate.com/applepay.aspx

Notice: For security reasons, Computop Paygate rejects all payment requests with formatting errors. Therefore, please use the correct data type for each parameter.

The following table describes the encrypted payment request parameters:

The table describes just basic parameters. It is also possible to include all credit card payment parameters (without credit card data). More information about this you can find within Card processing. If you are processing Recurring or stored Credential on File transactions, please make sure to review all the necessary information in "COF Mandate Integration Guide". If there are any specific parameters that are mandatory for credit card payment, these parameters are also mandatory for Apple Pay.

Key	Format	CND	Description
MerchantID	ans..30	M	MerchantID, assigned by Computop. Additionally this parameter has to be passed in plain language too.
TransID	ans..64	M	TransactionID provided by you which should be unique for each payment
RefNr	ns..30	C	Unique reference number. Mandatory in case that this parameter is mandatory for merchant's card processing.
Amount	n..10	M	Amount in the smallest currency unit (e.g. EUR Cent). Please contact the Computop Helpdesk, if you want to capture amounts <100 (smallest currency unit).
Currency	a3	M	Currency, three letters DIN / ISO 4217, e.g. EUR, USD, GBP. Please find an overview here: A1 Currency table
MAC	an64	M	Hash Message Authentication Code (HMAC) with SHA-256 algorithm. Details can be found here: • HMAC Authentication (Request) • HMAC Authentication (Notify)
UserData	ans..1024	O	If specified at request, Paygate forwards the parameter with the payment result to the shop.
RTF	a1	O	Establishment of Credential on File Agreement (Cardholder agrees that his card data can be stored for further transactions).

<table>
<tr><td>Capture</td><td>an..6</td><td>OM</td><td>Determines the type and time of capture.

<table><tr><th>Capture Mode</th><th>Description</th></tr><tr><td>AUTO</td><td>Capturing immediately after authorisation (default value).</td></tr><tr><td>MANUAL</td><td>Capturing made by the merchant. Capture is normally initiated at time of delivery.</td></tr><tr><td><Number></td><td>Delay in hours until the capture (whole number; 1 to 696).</td></tr></table></td></tr>
<tr><td>Order Desc</td><td>ans..64</td><td>M</td><td>Description of purchased goods, unit prices etc.</td></tr>
<tr><td>URLNotify</td><td>ans..256</td><td>M</td><td>Complete URL which Paygate calls up in order to notify the shop about the payment result. The URL may be called up only via port 443. It may not contain parameters: Use the UserData parameter instead.

Common notes:
Before follow-up actions (capture / credit / reversal) are carried out on an existing transaction, the first Notify must have been answered by the shop.
Fraudster may just copy the encrypted DATA-element which are sent to URLFailure and send the DATA to URLSuccess /URLNotify. Therefore ensure to check the "code"-value which indicates success/failure of the action. Only a result of "code=00000000" should be considered successful.</td></tr>
<tr><td>MerchantIDExt</td><td>ans..128</td><td>C</td><td>Merchant Identifier of Public Key

Mandatory if more then one Merchant Identifiers are stored with Computop</td></tr>
<tr><td>TokenExt</td><td>ans..6000</td><td>M</td><td>Apple Pay PKPaymentToken as JSON string in the Base64 format.

TokenExt is only valid for 2 minutes after retrieving from the shop and must also be sent within 2 minutes.

Example for TokenExt

<pre>{
 "paymentData": {
  "data":
"GiZiyzsI6r6lnPYUeceR6itk2PDyBozl2Xy77c5u2X8Ze7l5EasyyH4Q6BoAevrvBfe0FnUNARBEXRySLwq
qnpUHO6Du
/amZEECRXxlrH91wFqH4oXry2CTDRu7TaIlmnR+s3ien5JI8iWo9hoEW7hyJOE7QGaS6rfR1CtQ4DWJEUq
/tFnW98tj3kwKU6iOAAvE467boopMDGBS1fK5HzGXs4hH
/6r+LPRfSOKBi1L5VWAexs9Bzw3ByyG69i52doRuFb1xOcMOJbmPg40hap13IjBW6dnj1phbsqP2i
/JxvWPV3EcuqpuIoVZr5w53w
//pPsl54kmeXNddIjVD5dIhhOKZ8AznD4eL2dbzkp6bic8xScBf3G8hrKXTRTL7V+KT2S+TQliHN0SNXrFu6
B6o=",
  "signature":
"MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABgkqhkiG9w0BBwEAAKCAMIID4zCCA4i
gAwIBAgIITDBBSVGdVDYwCgYIKoZIzj0EAwIwejEuMCwGA1UEAwwlQXBwbGUgQXBwbGljYXRpb24gSW50ZWd
yYXRpb24gQ0EgLSBHMzEmMCQGA1UECwwdQXBwbGUgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxEzARBgNVBAo
MCkFwcGxlIEluYy4xCzAJBgNVBAYTAlVTMB4XDTE5MDUxODAxMzI1N1oXDTI0MDUxNjAxMzI1N1owXzElMCM
GA1UEAwwcZWNjLXNtcC1icm9rZXItc2lnbl9VQzQtUFJPRDEUMBIGA1UECwwLaU9TIFN5c3RlbXMxEzARBgN
VBAoMCkFwcGxlIEluYy4xCzAJBgNVBAYTAlVTMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEwhV37evWx7I
hj2jdcJChIY3HsL1vLCg9hGCV2Ur0pUEbg0IO2BHzQH6DMx8cVMP36zIg1rrV1O
/0komJPnwPE6OCAhEwggINMAwGA1UdEwEB/wQCMAAwHwYDVR0jBBgwFoAUI/JJxE+T5O8n5sT2KGw
/orv9LkswRQYIKwYBBQUHAQEEOTA3MDUGCCsGAQUFBzABhilodHRwOi8vb2NzcC5hcHBsZS5jb20vb2NzcDA
0LWFwcGxlYWljYTMwMjCCAR0GA1UdIASCARQwggEQMIIBDAYJKoZIhvdjZAUBMIH+MIHDBggrBgEFBQcCAjC
BtgyBs1JlbGlhbmNlIG9uIHRoaXMgY2VydGlmaWNhdGUgYnkgYW55IHBhcnR5IGFzc3VtZXMgYWNjZXB0YW5
jZSBvZiB0aGUgdGhlbiBhcHBsaWNhYmxlIHN0YW5kYXJkIHRlcm1zIGFuZCBjb25kaXRpb25zIG9mIHVzZSw
gY2VydGlmaWNhdGUgcG9saWN5IGFuZCBjZXJ0aWZpY2F0aW9uIHByYWN0aWNlIHN0YXRlbWVudHMuMDYGCCs
GAQUFBwIBFipodHRwOi8vd3d3LmFwcGxlLmNvbS9jZXJ0aWZpY2F0ZWF1dGhvcml0eS8wNAYDVR0fBC0wKzA
poCegJYYjaHR0cDovL2NybC5hcHBsZS5jb20vYXBwbGVhaWNhMy5jcmwwHQYDVR0OBBYEFJRX22
/VdIGGiYl2L35XhQfnm1gkMA4GA1UdDwEB
/wQEAwIHgDAPBgkqhkiG92NkBh0EAgUAMAoGCCqGSM49BAMCA0kAMEYCIQC+CVcf5x4ec1tV5a+stMcv60Rf
MBhSIsclEAK2Hr1vVQIhANGLNQpd1t1usXRgNbEess6Hz6Pmr2y9g4CJDcgs3apjMIIC7jCCAnWgAwIBAgII
SW0vvzqY2pcwCgYIKoZIzj0EAwIwZzEbMBkGA1UEAwwSQXBwbGUgUm9vdCBDQSAtIEczMSYwJAYDVQQLDB1B
cHBsZSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTETMBEGA1UECgwKQXBwbGUgSW5jLjELMAkGA1UEBhMCVVMw
HhcNMTQwNTA2MjM0NjMwWhcNMjkwNTA2MjM0NjMwWjB6MS4wLAYDVQQDDCVBcHBsZSBBcHBsaWNhdGlvbiBJ
bnRlZ3JhdGlvbiBDQSAtIEczMSYwJAYDVQQLDB1BcHBsZSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTETMBEG
A1UECgwKQXBwbGUgSW5jLjELMAkGA1UEBhMCVVMwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAATwFxGEGddk
hdUaXiWBB3bogKLv3nuuTeCN/EuT4TNW1WZbNa4i0Jd2DSJOe7oI/XYXzojLdrtmcL7I6CmE
/1RFo4H3MIH0MEYGCCsGAQUFBwEBBDowODA2BggrBgEFBQcwAYYqaHR0cDovL29jc3AuYXBwbGUuY29tL29j
c3AwNC1hcHBsZXJvb3RjYWczMB0GA1UdDgQWBBQj8knET5Pk7yfmxPYobD+iu
/0uSzAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFLuw3qFYM4iapIqZ3r6966
/ayySrMDcGA1UdHwQwMC4wLKAqoCiGJmh0dHA6Ly9jcmwuYXBwbGUuY29tL2FwcGxlcm9vdGNhZzMuY3JsMA
4GA1UdDwEB
/wQEAwIBBjAQBgoqhkiG92NkBgIOBAIFADAKBggqhkjOPQQDAgNnADBkAjA6z3KDURaZsYb7NcNWymK
/9Bft2Q91TaKOvvGcgV5Ct4n4mPebWZ+Y1UENj53pwv4CMDIt1UQhsKMFd2xd8zg7kGf9F3wsIW2WT8ZyaYI
Sb1T4en0dbmcubCYkhYQaZDwmSHQAAMYIBizCCAYcCAQEwgYYwejEuMCwGA1UEAwwlQXBwbGUgQXBwbGljYX
Rpb24gSW50ZWdyYXRpb24gQ0EgLSBHMzEmMCQGA1UECwwdQXBwbGUgQ2VydGlmaWNhdGlvbiBBdXRob3JpdH</pre></td></tr>
</table>

			kxEzARBgNVBAoMCkFwcGxlIEluYy4xCzAJBgNVBAYTAlVTAghMMEFJUZ1UNjANBglghkgBZQMEAgEFAKCBlTAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMjAyMjMxMDMyMzFaMCoGCSqGSIb3DQEJNDEdMBswDQYJYIZIAWUDBAIBBQChCgYIKoZIzj0EAwIwLwYJKoZIhvcNAQkEMSIEIKELTeQBJkyBdJ9Ge0BlmVOTIqU4sV75S
			/aC6sJMIHxbMAoGCCqGSM49BAMCBEYwRAIgC0iKpRgZQE2vMCSczjMRe+4b0aqiO79D2d0+9CKMmA8CICnC+e7RBgIPVbA32ZsKOV8e3iTdvm1OaH/ABCDEFGHIJKL",
			"header": {
			"publicKeyHash": "OgiD2qBTWYf/a+LDshFeQcPq6tOmePu0epHpP4ZkNicc=",
			"ephemeralPublicKey": "MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEc
			/HxA3lJZrC+B0ITom0Iji+gFdn7ivGtpI+fl2u8n8XByPgBaVK2b44qUvsGigoNd0OFLNXo0Q07R2B54eIdS3A==",
			"transactionId":
			"156632b2aadf355d4958d9051a42bf62e07aea5716e72083aa64247944f6e3e14d"
			},
			"version": "EC_v1"
			},
			"paymentMethod": {
			"displayName": "MasterCard 0063",
			"network": "MasterCard",
			"type": "debit"
			},
			"transactionIdentifier":
			"156632B2AAD12F355D4958D9051A42BF62E07AE5716E720AA6424794F6E3E14567D"
			}

Channel	a..10	C	Channel over which the order is processed. Allowed values are WEBSITE and MOBILE_APP. The channel parameter is mandatory for RedSys. Please provide it if your processor is RedSys. For other processors, it is not obligatory to provide this information.

Parameters for Apple Pay

The following table describes the result parameters with which the Computop Paygate responds to your system

pls. be prepared to receive additional parameters at any time and do not check the order of parameters

the key (e.g. mid, RefNr) should not be checked case-sentive

Key	Format	CND	Description
mid	ans..30	M	MerchantID, assigned by Computop
PayID	an32	M	ID assigned by Paygate for the payment, e.g. for referencing in batch files This ID refers to the credit card transaction unless an error occurs already at the Apple Pay transaction.
XID	an32	M	ID for all single transactions (authorisation, capture, credit note) for one payment assigned by Paygate
TransID	ans..64	M	TransactionID provided by you which should be unique for each payment
refnr	ans..30	O	Merchant's unique reference number
UserData	ans..1024	O	If specified at request, Paygate forwards the parameter with the payment result to the shop.
Status	a..50	M	OK (URLSuccess) or FAILED (URLFailure)
Description	ans..1024	M	Further details in the event that payment is rejected. Please **do not** use the **Description but** the Code parameter for the transaction status analysis!
Code	an8	M	Error code according to Paygate Response Codes (A4 Error codes)
MAC	an64	M	Hash Message Authentication Code (HMAC) with SHA-256 algorithm. Details can be found here: • HMAC Authentication (Request) • HMAC Authentication (Notify)
schemeReferenceID	ans..64	C	Card scheme specific transaction ID required for subsequent credential-on-file payments, delayed authorizations and resubmssions.

Result parameters for Apple Pay

Capture / Credit / Reversal

Captures, credits and reversals do not refer to the Apple Pay transaction but directly to the credit card transaction. More information about this you can find within Card processing.

Batch processing via the interface

Captures, credits and reversals via batch do not refer to the Apple Pay transaction but directly to the credit card transaction. More information about this you can find within the document Card processing.