Description and prerequisites for Apple Pay

Implementation guide

Document version 1.0

Contents

1. PRESENTATION

Apple Pay is a wallet that allows you to pay securely with several types of cards (Visa, MasterCard, etc.)
Available on Apple devices and from the Safari browser, Apple Pay works in countries and regions around the world.

- **Quick**: a payment in a few seconds thanks to *Face ID* or *Touch ID*.
- **Limited risk of fraud**: During your integration, you will not have to deal with the 3D Secure authentication process during payments.Apple PaySince it is already capable of handling payments and has built-in biometric authentication, it is fully compatible with strong customer authentication. The buyer's identity is automatically verified by Apple, which greatly reduces the fraud risk.
- **Operating for your customers:**
  Making a payment via the Apple Pay wallet is very simple and takes 2 steps:
  1. The buyer chooses Apple Pay To pay. The application Apple Pay opens.
  2. The buyer accesses their registered cards and chooses the payment card to proceed with the payment.
- **Operating for you**: several integration solutions depending on your subscription.
  This document describes the various options and/or steps for:
  - Setting up your Merchant ID.
  - Choose the possible integration mode(s).
  - Manage your transactions.

Supported currencies	• Currencies of your domain
Supported countries	• Worldwide* **Contact us for more information*
Details information	• Payment possible from a Safari browser on an Apple device. See *Apple’s documentation on compatibility*. • Operations available (cancellation, refund, etc.) depending on the acceptance network of the payment method registered in the wallet.

2. PREREQUISITES

Merchant:

- Subscribe to offer including Apple Pay
- Set up your Apple Pay merchant ID.
- Have a CB and/or AMEX contract associated with the store that offers Apple Pay.
- Choose an integration solution that's compatible with your contract mode. See: *Technical information* on page 5.

Buyer:

- Have an Apple Pay account and a compatible device.
- Have one or more payment methods associated with your Apple Pay account.

> In production mode, Apple Pay is only available from the Safari browser on Apple-branded devices (Mac, iPhone, iPad, etc.).
>
> In test mode, a payment simulator is available. It lets you pay from a Safari browser on Apple-branded devices (Mac, iPhone, iPad, etc.), but also from non-compatible browsers and devices (e.g. Chrome, Firefox, etc.).

3. TECHNICAL INFORMATION

Technical code	Supported currencies	Supported countries	Authorization validity period	Payment process
APPLE_PAY	Multiple	World Wide	according to payment card	Deferred capture

Which integration solution to choose in the contract?			
Integration solution	ModeMerchant	ModeWrapper *	ModeGateway
Embedded mode (smartForm)	✓	✓	✓
Embedded mode (cards)	✗	✗	✗
Redirection mode	✗	✗	✓
Native button in your application	✓	✗	✗
Mobile SDK	✓	✗	✗

* Unavailable

Type of integration	
Embedded mode (smartForm)	✓
Embedded mode (cards)	✗
Redirection mode	✓
Native button in your application	✓
Mobile SDK	✓
Data collection form	✓
Back Office	✗
Iframe	✗

Payment type	
Immediate payment	✓
Deferred payment	✓
Payment in installments	✗
Payment by subscription	✗
Payment by file (token or batch)	✗
One-click payment	✗
Payment by wallet	✗

Operations with transactions	
Cancellation	✓
Refund	✓
Modification	✓

Operations with transactions	
Validation	✓
Duplication	✗
Manual reconciliation	✓

Sales channel	
e-commerce	✓
m-commerce	✓
MOTO payment	✗
Interactive Voice Server	✗
Payment order by e-mail/SMS	✓

Miscellaneous	
Reporting	✓
Transaction settlement	✗
Chargeback settlement	✗
Extra payment attempts	✗

4. SETTING UP YOUR MERCHANT ID

Which integration solution to choose in the contract?			
Integration solution	**ModeMerchant**	**ModeWrapper***	**ModeGateway**
Embedded mode (smartForm)	✓	✓	✓
Embedded mode (cards)	✗	✗	✗
Redirection mode	✗	✗	✓
Native button in your application	✓	✗	✗
Mobile SDK	✓	✗	✗

* Unavailable

Follow the procedure adapted to your integration to define your Apple Pay contract.

- **Define a contract in Gateway mode**

  The payment platform takes care of creating the Apple Pay ID, which serves as the contract number. You do not have to register or create a merchant ID with Apple. After signing up for Apple Pay on the payment gateway, you must read and accept the *Apple Pay terms of service* via your Merchant Back Office.

- **Define a contract in Merchant mode**

  You must create your Apple Pay merchant ID directly withApple Developeror use an existing account. Then, you will follow the different activation steps specified by Apple and those specified by the payment gateway to set up your contract and finalize your subscription.

- **Defining a contract in Wrapper mode**

  You are part of a group while having your own legal entity and your shop URL depends on the domain of this group.

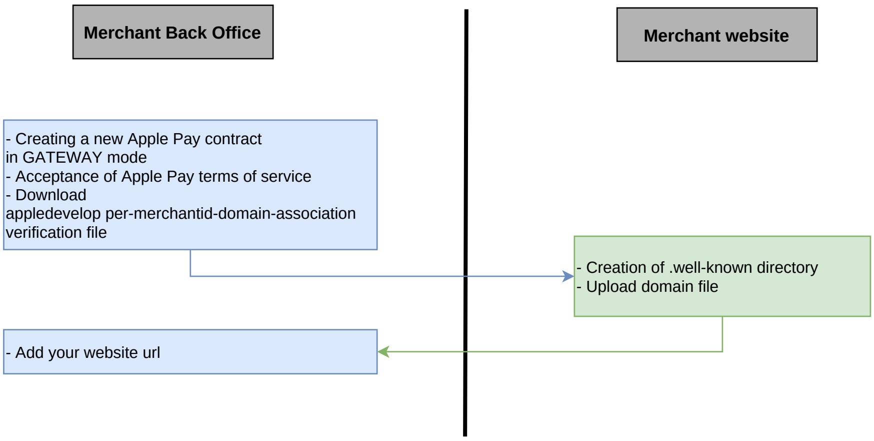
  With this mode, you'll have a single merchant ID on the Apple Developer account in the name of the master domain for the whole community.

  This Apple Pay merchant ID from the *master* domain must be associated with your company when Apple Pay option on the payment platform.

! **This integration solution is not available on Scellius.**

4.1. Defining a contract in Gateway mode

The payment gateway automatically creates a merchant ID for Apple Pay and manages the relationship with Apple. You don't need an Apple account. After subscribing to Apple Pay, you must you must read and accept the *Apple Pay terms of service* from the Merchant Back Office.


Merchant Back Office
Merchant website
- Creating a new Apple Pay contract in GATEWAY mode
- Acceptance of Apple Pay terms of service
- Download appledevelop per-merchantid-domain-association verification file
- Creation of .well-known directory
- Upload domain file
- Add your website url

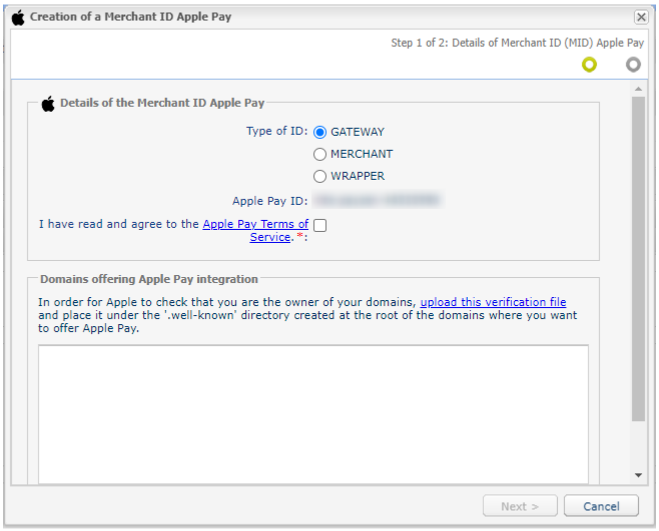
4.1.1. Creating a merchant ID for Apple Pay in the Merchant Back Office

1. Sign in to the *Back Office.*

2. Go to **Settings** > **Company** > **Merchant IDs**.

3. Click **Create a Merchant ID (MID)**.
   The **Selection of Merchant ID type** dialog box appears. Merchant ID types are displayed as icons.

4. Select **Apple Pay**.
   The merchant ID creation assistant opens.



Creation of a Merchant ID Apple Pay

Step 1 of 2: Details of Merchant ID (MID) Apple Pay

Details of the Merchant ID Apple Pay

Type of ID: ● GATEWAY
○ MERCHANT
○ WRAPPER

Apple Pay ID: [redacted]

I have read and agree to the Apple Pay Terms of Service. *: □

Domains offering Apple Pay integration

In order for Apple to check that you are the owner of your domains, upload this verification file and place it under the '.well-known' directory created at the root of the domains where you want to offer Apple Pay.

5. Select the GATEWAY ID type.

> There are other types of Apple Pay ID.
>
> If you already have an Apple Developer merchant account and/or are interested in a particular type of ID, please refer to the relevant chapters.
>
> Your purchaser does not offer WRAPPER mode.

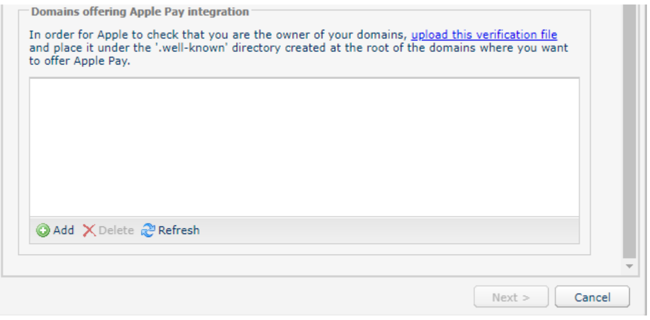
6. The payment platform generates an Apple Pay ID as a merchant contract number.

7. Read and accept the *Apple Pay terms of service* by checking the box provided for that purpose.

   This action is required for finalizing the creation of a Merchant ID.

8. Apple provides a file to verify the owner of your domain. You must follow the process below to allow this verification.

   a. Click on **download this verification file**.

   Domains offering Apple Pay integration

   In order for Apple to check that you are the owner of your domains, upload this verification file and place it under the '.well-known' directory created at the root of the domains where you want to offer Apple Pay.

   b. Retrieve the downloaded apple-developer-merchantid-domain-association file and create a directory named **.well-known** at the root of your domain for uploading the file.

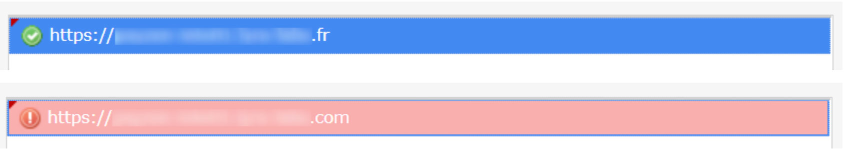
   In the end, you should have a url like this: https://your-domain.com/.well-known/apple-developer-merchantid-domain-association

   > ! This step is very important. Apple uses this file to check that you are the owner of the domain.
   >
   > If you have several domains, you will have to create this directory at the root of each domain and upload the same file there.

   c. Once the file is placed in the **.well-known** directory at the root of your domain, click on **Add** and enter your URL.

   The payment gateway checks the presence of the file in real time. If the file is present in the **.well-known** directory, this type of ✓ icon will precede the added URL. Otherwise, the URL will be preceded by this type of icon ! .

   Examples

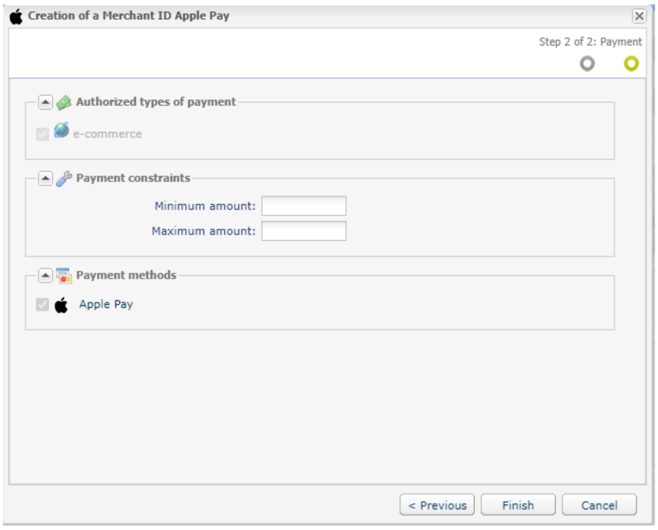
   https://[redacted].fr

   https://[redacted].com

   It is possible to delete a domain by selecting it and clicking **Delete**.

9. Click **Next** to continue the process.

   The rest of the setting is displayed.



Authorized types of payment

☑ e-commerce

Payment constraints

Minimum amount:

Maximum amount:

Payment methods

☑ Apple Pay

The **e-commerce** box is enabled by default since it is the only type of payment allowed.

10. Enter the **Minimum amount** and **Maximum amount** authorized during a payment if you want to define thresholds and/or payment caps.

11. The **Apple Pay** payment method is selected by default.

12. Click **Finish**.

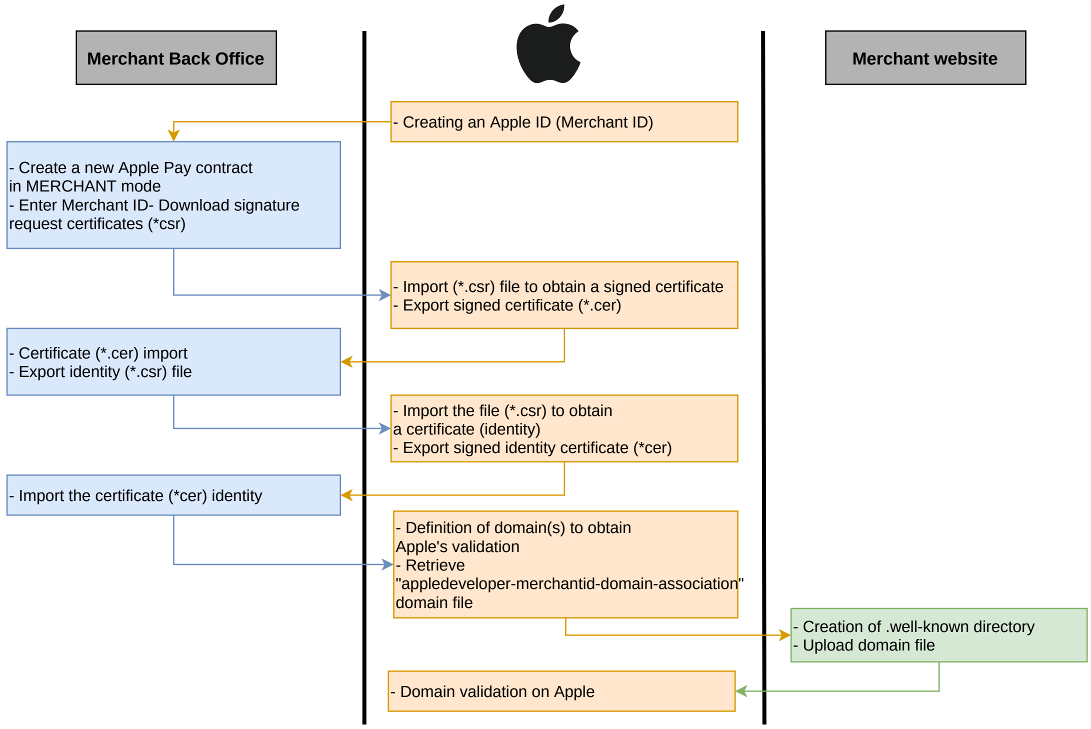
    When the **Association of the Merchant ID (MID)** window appears, check the shop(s) to associate with this contract.

> If you are unable to associate the contract with your shop, contact the E-Banking Merchant Support Service for checking your offer.

4.2. Defining a contract in Merchant mode

You must first open an Apple Developer merchant account. If you already have an Apple Developer account, you can use it.

Declare your merchant ID in the Merchant Back Office and follow the activation steps.



Merchant Back Office

Merchant website

- Creating an Apple ID (Merchant ID)

- Create a new Apple Pay contract in MERCHANT mode
- Enter Merchant ID- Download signature request certificates (*csr)

- Import (*.csr) file to obtain a signed certificate
- Export signed certificate (*.cer)

- Certificate (*.cer) import
- Export identity (*.csr) file

- Import the file (*.csr) to obtain a certificate (identity)
- Export signed identity certificate (*cer)

- Import the certificate (*cer) identity

- Definition of domain(s) to obtain Apple's validation
- Retrieve "appledeveloper-merchantid-domain-association" domain file

- Creation of .well-known directory
- Upload domain file

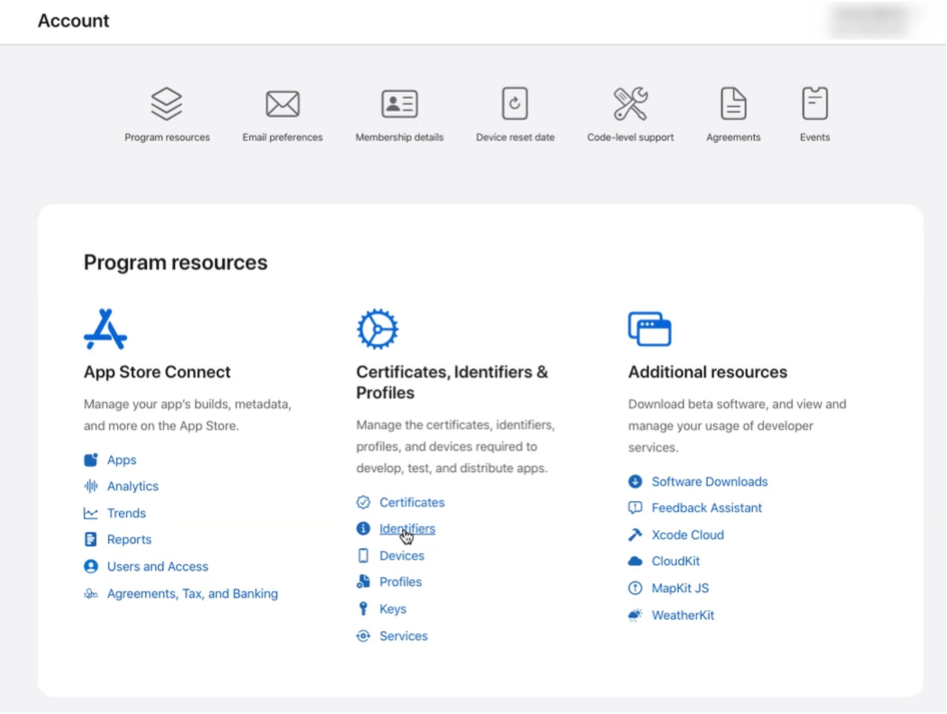
- Domain validation on Apple

4.2.1. Creating an Apple Developer merchant ID.

> ! You must have an account with *Apple Developer* to create a merchant ID.

1. Login to *Apple Developer*.

2. Go to *Certificates, Identifiers & Profiles*.



Account [redacted]

Program resources | Email preferences | Membership details | Device reset date | Code-level support | Agreements | Events

Program resources

App Store Connect

Manage your app's builds, metadata, and more on the App Store.

- Apps
- Analytics
- Trends
- Reports
- Users and Access
- Agreements, Tax, and Banking

Certificates, Identifiers & Profiles

Manage the certificates, identifiers, profiles, and devices required to develop, test, and distribute apps.

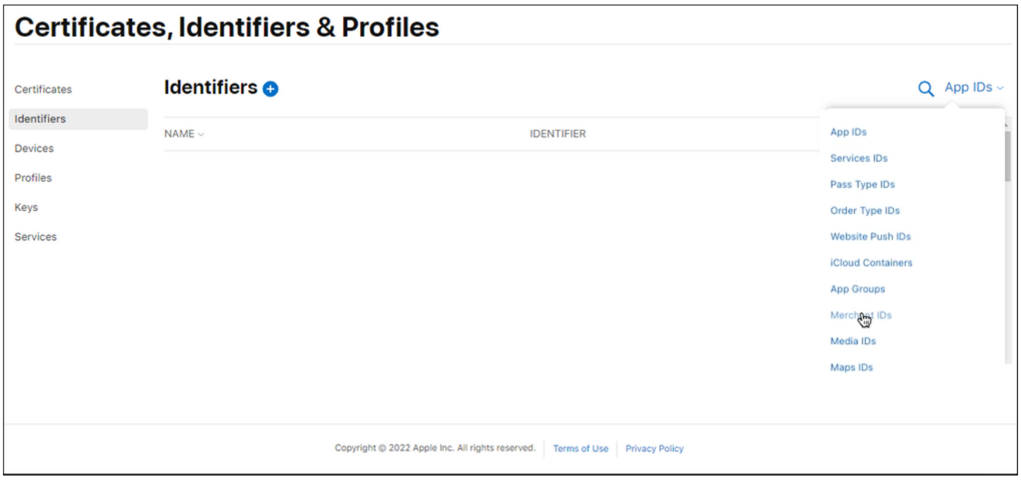
- Certificates
- Identifiers
- Devices
- Profiles
- Keys
- Services

Additional resources

Download beta software, and view and manage your usage of developer services.

- Software Downloads
- Feedback Assistant
- Xcode Cloud
- CloudKit
- MapKit JS
- WeatherKit

3. Click on the **Identifiers** tab.
   The identifier management page opens.



Certificates, Identifiers & Profiles

- Certificates
- Identifiers
- Devices
- Profiles
- Keys
- Services

Identifiers ⊕

NAME ⌄	IDENTIFIER

App IDs ⌄

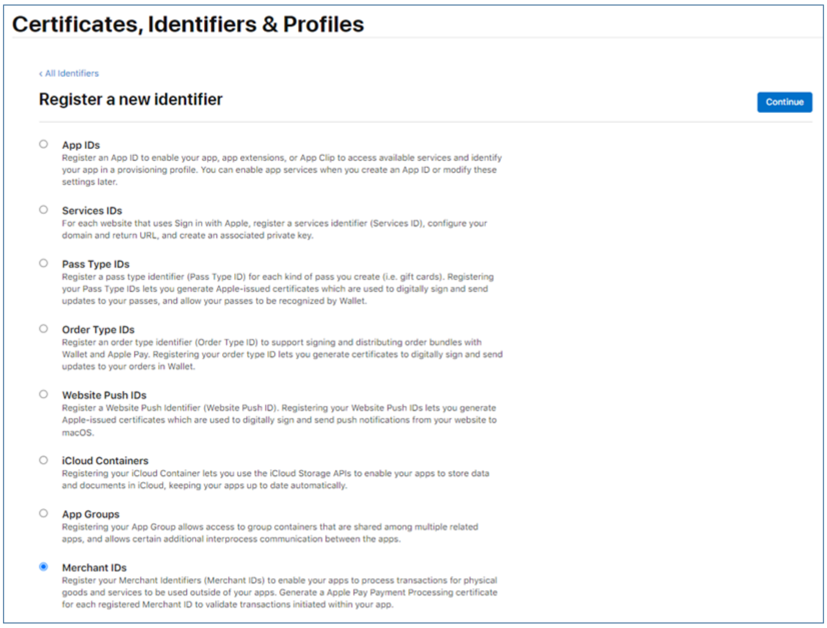
- App IDs
- Services IDs
- Pass Type IDs
- Order Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs
- Media IDs
- Maps IDs

Copyright © 2022 Apple Inc. All rights reserved. | Terms of Use | Privacy Policy

4. Click **Merchant IDs** in the ***App IDs*** list.

5. Then click on the **+** icon to the right of **Identifiers**

   The merchant identifier registration page opens.



Certificates, Identifiers & Profiles

Register a new identifier

○ **App IDs**
Register an App ID to enable your app, app extensions, or App Clip to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

○ **Services IDs**
For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

○ **Pass Type IDs**
Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

○ **Order Type IDs**
Register an order type identifier (Order Type ID) to support signing and distributing order bundles with Wallet and Apple Pay. Registering your order type ID lets you generate certificates to digitally sign and send updates to your orders in Wallet.

○ **Website Push IDs**
Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

○ **iCloud Containers**
Registering your iCloud Container lets you use the iCloud Storage APIs to enable your apps to store data and documents in iCloud, keeping your apps up to date automatically.

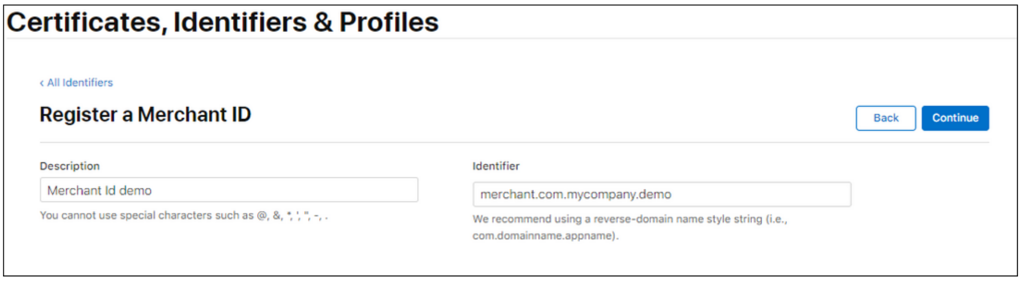
○ **App Groups**
Registering your App Group allows access to group containers that are shared among multiple related apps, and allows certain additional interprocess communication between the apps.

● **Merchant IDs**
Register your Merchant Identifiers (Merchant IDs) to enable your apps to process transactions for physical goods and services to be used outside of your apps. Generate a Apple Pay Payment Processing certificate for each registered Merchant ID to validate transactions initiated within your app.

6. Check the **Merchant IDs** radio button then click on **Continue**.

   The merchant identifier entry page opens.

> We recommend that you create two separate merchant IDs, one for your test environment and one for production.



Certificates, Identifiers & Profiles

Register a Merchant ID

Description: Merchant Id demo

You cannot use special characters such as @, &, *, ', ", -, .

Identifier: merchant.com.mycompany.demo

We recommend using a reverse-domain name style string (i.e., com.domainname.appname).

7. Enter your merchant ID in the **Identifier** field.

   You must choose your own ID according to Apple's recommendations.

   Example: *merchant.com.mycompany.demo* if it is your test account.

8. Fill in the **Description** field if you wish.

   Example: *merchant id demo* if it is your test account.

9. Click **Continue** to complete.

   Your identifier is created. Contact the E-Banking Merchant Support Service to receive your merchant ID.

4.2.2. Declaring a merchant ID for Apple Pay in the Merchant Back Office

1. Sign in to the *Back Office*

2. Go to **Settings** > **Company** > **Merchant IDs**

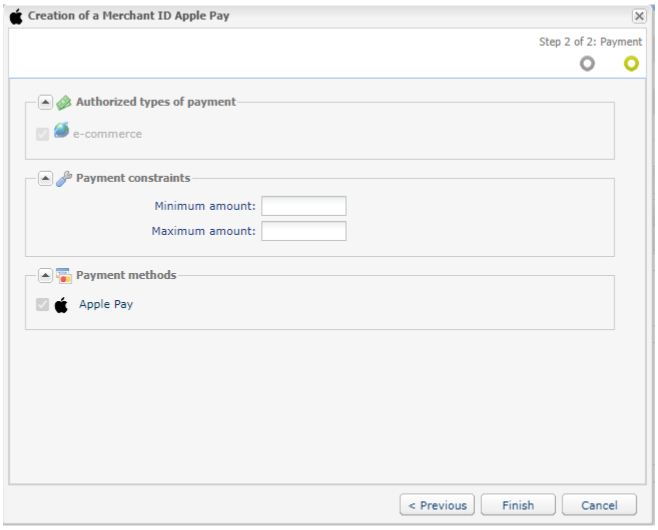
3. Click **Create a Merchant ID (MID)**.
   Merchant ID types are displayed as icons.

4. Select **Apple Pay**.
   The merchant ID creation assistant opens.

5. Select the MERCHANT ID type.

6. Enter your **Apple Pay merchant ID** that you obtained when you registered for Apple Developer.

7. Click **Next** to continue the process.
   The rest of the setting is displayed.



Creation of a Merchant ID Apple Pay

Step 2 of 2: Payment

Authorized types of payment

☑ e-commerce

Payment constraints

Minimum amount:

Maximum amount:

Payment methods

☑ Apple Pay

The **e-commerce** box is enabled by default since it is the only type of payment allowed.

8. If you wish to set payment thresholds and/or ceilings, enter the **Minimum amount** and **Maximum amount** authorized for a payment.

9. **Apple Pay** payment method is selected by default.

10. Click **Finish**.

    When the **Association of the Merchant ID (MID)** window appears, check the shop(s) to associate with this contract.

    If you are unable to associate the contract with your shop, contact the E-Banking Merchant Support Service for checking your offer.

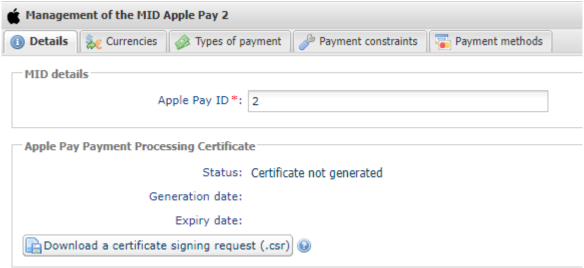
Follow the steps below to finalize the contract definition.

4.2.3. Setting up your Apple Pay Merchant ID in the Merchant Back Office

In the menu **Settings** > **Company** > **Merchant IDs** of your Merchant Back Office:

1. Select your **Apple Pay** MID created in the **Merchant IDs** tab.

   The contract details appear.


Management of the MID Apple Pay 2
Details
Currencies
Types of payment
Payment constraints
Payment methods
Download a certificate signing request (.csr)

MID details

Apple Pay ID *: 2

Apple Pay Payment Processing Certificate

Status: Certificate not generated

Generation date:

Expiry date:

2. In the *Apple Pay Payment Processing Certificate* section, click on **Download a certificate signature request (*. csr)**.

   The platform requires a signed certificate for payment processing. This certificate is mandatory.

3. Connect to *Apple Developer* to import the downloaded file. Follow these steps to obtain a signed certificate:

   a. Go to *Certificates, Identifiers & Profiles*.

   b. In the *App IDs* list, select *Merchant IDs*.

   c. From page *Merchant ID*, go to section ***Apple Pay Payment Processing Certificate***.

   d. Click *Create Certificate*.

      The certificate creation page appears.

   e. Click *Choose file* then load the file downloaded from your Merchant Back Office.

      Example: processing.bdupont-1234-apple.csr

   f. Click *Continue* then *Download* on the page that appears.

      The signed certificate is generated.

   g. Register the downloaded certificate.

4. In the *Apple Pay Payment Processing Certificate* section, click on **Import the certificate signed by Apple (.cer)**.

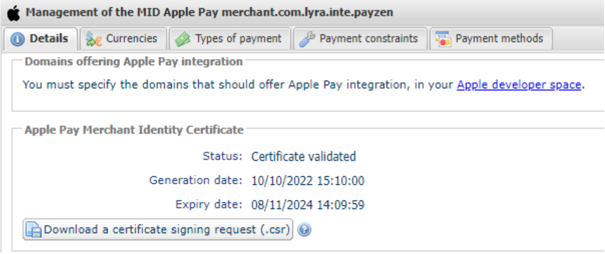
   This button appears after the certificate signing request has been downloaded.

5. Import the downloaded certificate.

   Example: processing.bdupont-1234-apple.csr

   The Status of the field changes to **Certificate enabled**.

6. In the *Apple Pay Merchant Identity Certificate* section, click on **Download a certificate signature request (.csr)**.


Management of the MID Apple Pay merchant.com.lyra.inte.payzen
Details
Currencies
Types of payment
Payment constraints
Payment methods
Download a certificate signing request (.csr)

Domains offering Apple Pay integration

You must specify the domains that should offer Apple Pay integration, in your Apple developer space.

Apple Pay Merchant Identity Certificate

Status: Certificate validated

Generation date: 10/10/2022 15:10:00

Expiry date: 08/11/2024 14:09:59

The platform needs a signed certificate for merchant identity. This certificate is mandatory.

ℹ The two files each have a different prefix for easy identification.

Examples:

processing.bdupont-1234-apple.csr

identity.bdupont-1234-apple.csr

7. From your Apple accountApple Developer:
   a. Go to *Certificates, Identifiers & Profiles*.
   b. In the *App IDs* list, select *Merchant IDs*.
   c. From *Merchant ID* page, go to ***Apple Pay Merchant Identity Certificate*** section.
   d. Click *Create Certificate*.

      The certificate creation page appears.
   e. Click *Choose file* then load the file downloaded from your Merchant Back Office.

      Example: identity.bdupont-1234-apple.csr
   f. Click *Continue* then *Download* on the page that appears.

      The signed certificate is generated.
   g. Register the downloaded certificate.
8. In the *Apple Pay Merchant Identity Certificate* section, click on **Import the certificate signed by Apple (.cer)**.

   This button appears after the certificate signing request has been downloaded.
9. Import the downloaded certificate.

   Example: identity.bdupont-1234-apple.csr

   The Status of the field changes to **Certificate enabled**.

10. Declare domains.

    You must declare all the URLs where Apple Pay will be used, and these URLs must be in https.
    a. Sign in to *your Apple Developer account*.
    b. In the *App IDs* list, select *Merchant IDs*.
    c. Select your *Merchant ID* (see: *Creating an Apple Developer merchant ID.* on page 11).
    d. Under the *Merchant Domain* section, select ***Add Domain***.
    e. Fill in your domain name and click save.
    f. Select ***Download*** then save the apple-developer-merchantid-domain-association file obtained.
    g. Create a directory called **.well-known** at the root of your domain.
    h. Retrieve the registered file apple-developer-merchantid-domain-association and drop it into the directory.

       You should obtain a url like this: https://your-domain.com/.well-known/apple-developer-merchantid-domain-association
    i. Return to your *Merchant Domain* home then click on ***Verify***.

    The Apple Pay contract declaration is complete.

ℹ You will receive an e-mail from Apple when the certificate renewal period approaches.

5. SETTING UP NOTIFICATIONS

The Merchant Back Office provides several types of notifications:

- Instant Payment Notification URL call
- E-mail sent to the merchant
- E-mail sent to the buyer
- SMS sent to the merchant
- SMS sent to the buyer

The notification is sent according to the event that takes place (payment accepted, abandonment by the buyer, cancellation by the merchant, etc.).

> ! Instant Payment Notification URL call-type notifications are the most important. They are the only reliable way for the merchant website to obtain the payment result.

Instant Payment Notification URL failed

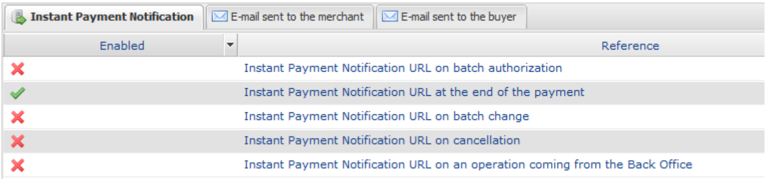
If the platform is unable to reach your URL, it sends an e-mail to the specified address.

This e-mail contains:

- the HTTP code of the encountered;
- parts of analysis depending on the error;
- the consequences of error;
- instructions via the Merchant Back Office to resend the request to the previously defined URL.

Access to notification rules management

Go to the following menu **Settings** > **Notification rules**.



Enabled	Reference
✗	Instant Payment Notification URL on batch authorization
✓	Instant Payment Notification URL at the end of the payment
✗	Instant Payment Notification URL on batch change
✗	Instant Payment Notification URL on cancellation
✗	Instant Payment Notification URL on an operation coming from the Back Office

5.1. Setting up the Instant Payment Notification

The payment gateway notifies on the merchant website in the following cases:

- Payment accepted
- Payment refused
- Payment waiting for authorization (file analysis is in progress)

This notification is required to communicate the result of the payment request.

It will inform the merchant website of the payment result even if your client has not clicked the “Return to the shop” button.

1. Right-click **Instant Payment Notification URL at the end of the payment**.

2. Select **Manage the rule**.

3. Enter the **E-mail address(es) to notify in case of failure** field in the "General settings” section. To specify several e-mail addresses, separate them with a semi colon (;).

4. Check the box **Automatic retry in case of failure** if you wish to authorize the gateway to automatically resend the notification in case of a failure (can be done up to 4 times).

5. If you wish to receive notifications in API Form format, enter the URL of your page in **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** field in the "Instant Payment Notification URL of the API form V1, V2” section.

6. Save the changes.

5.2. Setting up notifications in case of abandoned or canceled payments

The payment gateway notifies on the merchant website in the following cases:

- Abandon/cancel, if the buyer clicks on **Cancel and return to shop**.
- The buyer did not complete their payment before the expiration of the payment session.

  The length of a payment session is 10 minutes.

This rule is **disabled by default**.

1. Right-click **Instant Payment Notification URL on cancellation**.
2. Select **Manage the rule**.
3. Enter the **E-mail address(es) to notify in case of failure** field in the "General settings" section. To specify several e-mail addresses, separate them with a semi colon (;).
4. To allow the platform to automatically resend the notification in the event of failure, check the **Automatic retry in case of failure** box. This mechanism allows up to 4 attempts to be made.
5. If you wish to receive notifications in API Form format, enter the URL of your page in **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** field in the "Instant Payment Notification URL of the API form V1, V2" section.
6. If you're using the clientJavaScript, specify the URL of your page in the fields **Target URL of the IPN to notify in TEST mode** and **Target URL of the IPN to notify in PRODUCTION mode** in the "REST API Instant Payment Notification URL" section.
7. Save the changes.
8. Right-click the **Instant Payment Notification URL on cancellation** and select **Enable the rule**.