



COLLECTING SOLUTION

Integrating the Apple Pay API for your mobile payments

Document version 1.0

Contents

1. HISTORY OF THE DOCUMENT.....	3
2. PRESENTATION.....	4
3. PREREQUISITES.....	5
4. CONFIGURING THE CONTRACTS.....	6
4.1. Contract definition in the Gateway mode.....	7
4.1.1. Creating an Apple Pay merchant ID in the Expert Back Office.....	7
4.2. Contract definition in the Merchant mode.....	10
4.2.1. Creating a merchant ID in Apple Pay Developer.....	10
4.2.2. Declaring an Apple Pay merchant ID in the Expert Back Office.....	12
4.2.3. Setting up your Apple Pay Merchant ID in the Expert Back Office.....	14
4.3. Contract definition in the merchant mode “Wrapper”.....	17
4.3.1. Declaring an Apple Pay merchant ID in “Wrapper” mode.....	17
5. EXCHANGE PROCESS WITHIN THE MOBILE SDK.....	20
6. ENABLING PAYMENTS VIA APPLE PAY IN THE SDK.....	21
6.1. Prerequisites.....	21
6.2. Enabling Apple Pay capability in your application.....	21
6.3. Initializing the SDK.....	21
7. VIEWING THE EXAMPLES OF INTEGRATION CODES.....	23
8. MANAGING TRANSACTIONS VIA THE LYRA COLLECT BACK OFFICE.....	24
8.1. Viewing transaction details via Apple Pay.....	24
8.2. Canceling a transaction.....	26
8.3. Modifying a transaction.....	27
8.4. Duplicating a transaction.....	27
8.5. Making a refund.....	28
8.6. Validating a transaction.....	29
9. OBTAINING HELP.....	30

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
1.0	Lyra Collect	10/7/2022	Initial version

This document and its contents are confidential. It is not legally binding. Any reproduction and / or distribution of all or part of this document or its content to a third party is strictly prohibited or subject to prior written authorization from Lyra Collect. All rights reserved.

2. 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.

- **FAST:** a payment in a few seconds thanks to Face ID or Touch ID
- **LIMITED FRAUD RISK :** During your integration, you will not have to deal with the 3D Secure authentication process during payments. Since Apple Pay is already capable of handling payments that have 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 for making the payment. 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 procedure to follow for a successful integration.



Supported currencies

- Currencies of your domain



Supported countries

- Worldwide*

**Contact us for more information*



Additional information

- Apple Pay can be used via Safari on an Apple device.
For more information, see [Apple's documentation on compatibility](#).
- Only the payment methods compatible with Apple Pay will be presented to the buyer on their account at checkout.
- The operations available in the Merchant Back Office (cancellation, refund, etc.) once the Apple Pay transaction has been carried out depend on the acceptance network of the payment method registered in the wallet.

This document describes mobile app integration using payment credentials sent by the Apple Pay API.

3. PREREQUISITES

For the merchant:

- Subscribe to our 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.

For the buyer:

- Have an Apple Pay account.
- Have one or more payment methods associated with your Apple Pay account.
- Agree to the Apple Pay terms and conditions.
- Be able to make your purchase from a Safari browser or an Apple Pay device (iPhone, iPad, etc.).

For more information on devices that support Apple Pay, see [Apple's documentation on compatibility](#).

4. CONFIGURING THE CONTRACTS

The table below defines the scope of use of Apple Pay depending on your integration solution and the way the contract is defined.

Integration solution	Contract definition	
	ModeMerchant/ ModeWrapper	ModeGateway
Embedded form / SmartForm	YES	YES
Native button in the apps / REST WS	YES (recommended)	YES
Mobile SDK	YES (recommended)	YES

Depending on your integration solution, you can follow the relevant steps to define your contract for Apple Pay:

- [Contract definition in the modeGateway](#)

The payment gateway creates the Apple Pay ID for you that serves as your 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 Expert Back Office.

- [Contract definition in the modeMerchant](#)

In this mode, you must create your Apple Pay merchant ID directly with Apple Pay Developer by signing the Apple Pay terms of service or using 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.

- [Contract definition in the “Wrapper” merchant 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 method of contract definition, you will have a single merchant ID on the **Apple Pay Developer** account on behalf of the master domain and for the entire group.

You must associate this master domain Apple Pay merchant ID with your company when you sign up for the Apple Pay option on the payment gateway.

4.1. Contract definition in the Gateway mode

In this contract definition mode, the payment gateway automatically creates a merchant ID for Apple Pay and manages the relationship with Apple. The merchant does not have to register or create an account with Apple. However, after subscribing to the Apple Pay option, the merchant must read and accept the [Apple Pay terms of service](#) via his or her Expert Back Office.

4.1.1. Creating an Apple Pay merchant ID in the Expert Back Office

1. Sign in to the [Back Office](#)

2. Click **Other actions** to access your Expert Back Office.

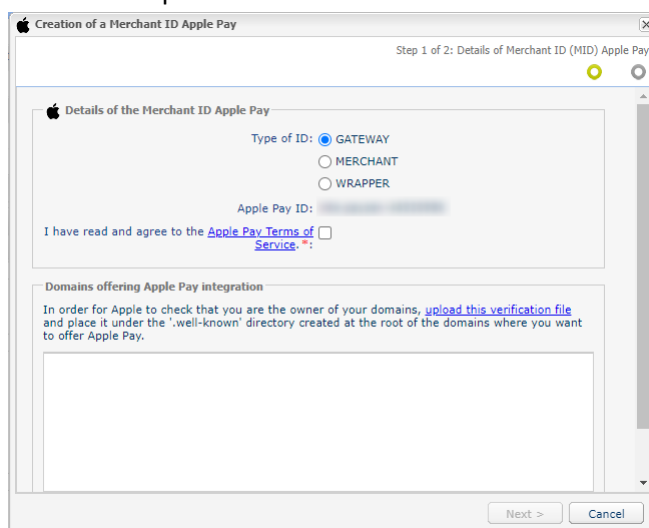
3. Go to **Settings > Company > Merchant IDs**

4. Click **Create a Merchant ID (MID)**.

The **Creation of an Apple Pay Merchant ID** dialog box appears. Merchant ID types are displayed as icons.

5. Select **Apple Pay**.

The merchant ID creation assistant opens.



6. Select the **GATEWAY** ID type.



For your information, there are other types of Apple Pay IDs called **MERCHANT** and **WRAPPER**.

If you already have a merchant account on Apple Pay Developer and/or if you are interested by these contract definition modes, we invite you to see the relevant chapters.

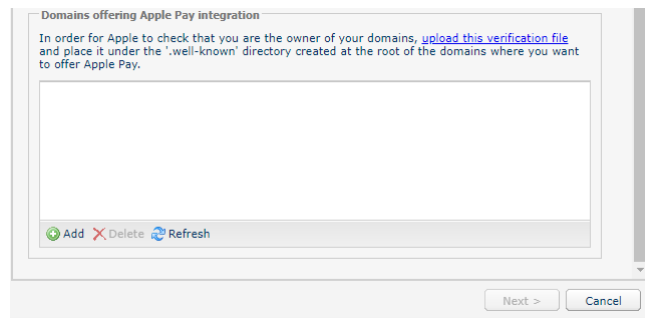
7. By default, the payment gateway generates an Apple Pay ID as a Merchant ID.

8. 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.

9. Apple provides a file for verifying that you are the owner of your domain. As part of your integration, you must follow the process below to allow this verification.

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



- 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.txt>



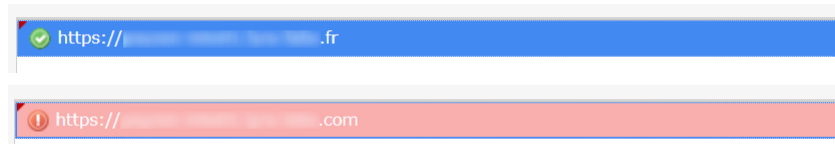
This step is very important. Apple uses this file to check that you are the owner of the domain.

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 .

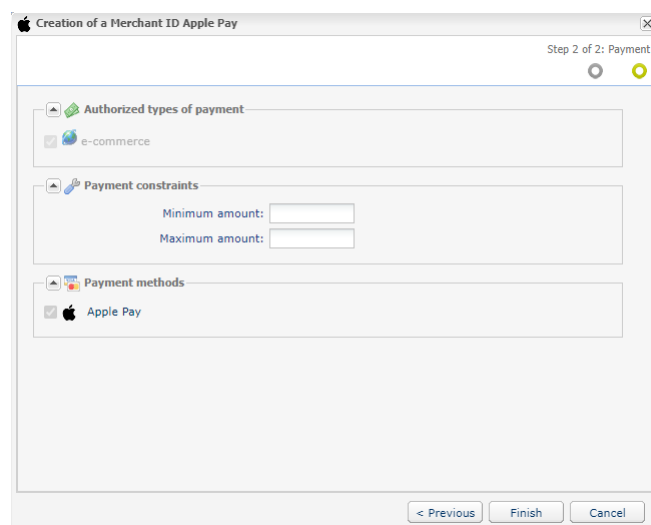
Examples



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

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

The rest of the setting is displayed.



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

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

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

13. 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 adv@lyra-collect.com for checking your offer.

4.2. Contract definition in the Merchant mode

In this contract definition mode, the merchant first opens their account directly with Apple Pay Developer by subscribing to the Apple Pay terms of service. The merchant receives their Apple Pay merchant ID. Naturally, if the merchant already has an Apple Pay Developer account, he or she can use it.

The merchant declares this identifier in the Expert Back Office and follows the activation steps specified by Apple and the ones specified by the payment gateway.



A contract defined in the modeMerchant will not work as an integration via a form with redirect. The Apple Pay button will not be suggested on the payment page.

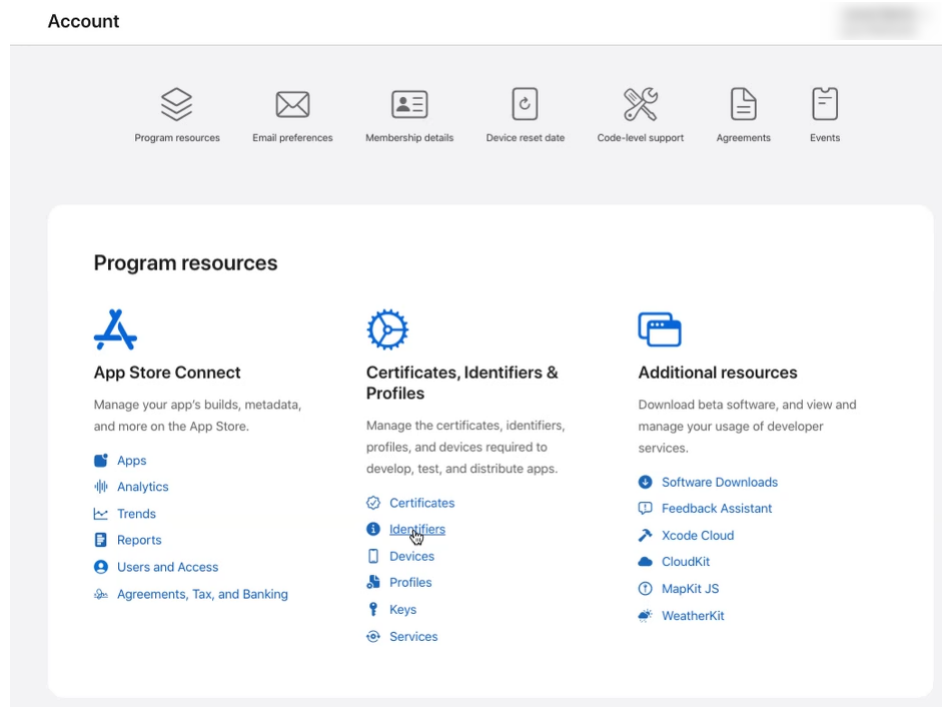
4.2.1. Creating a merchant ID in Apple Pay Developer



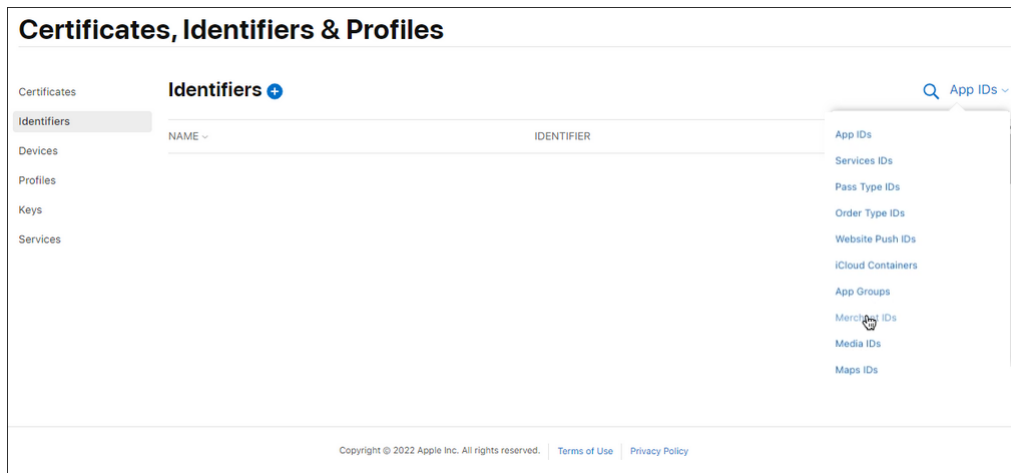
First, you must **Create your Apple ID** in [Apple Pay Developer](#) to be able to log in and create a merchant ID.

If you already have an account, you can use it.

1. Connect to [Apple Pay Developer](#).
2. Go to the section *Certificates, Identifiers & Profiles*.



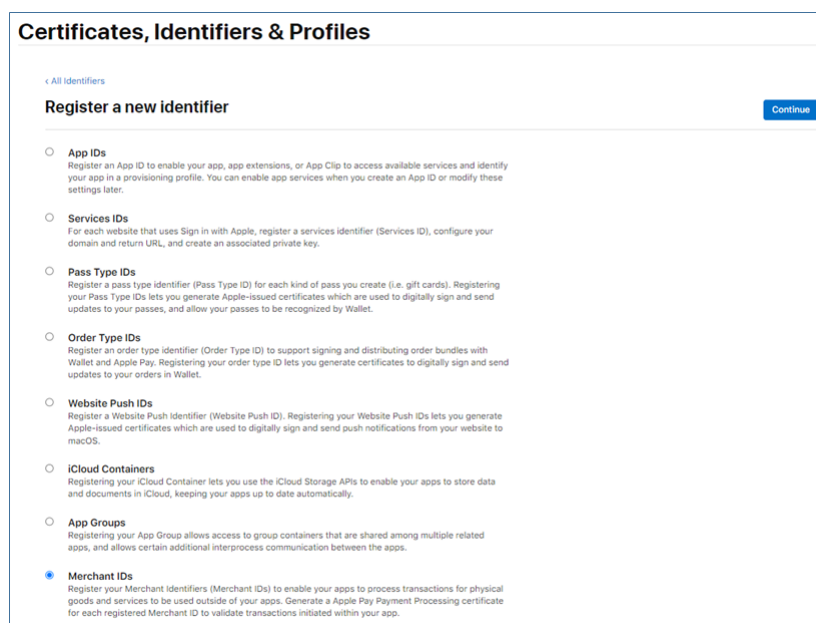
3. Click on the **Identifiers** tab.
The identifier management page opens.



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

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

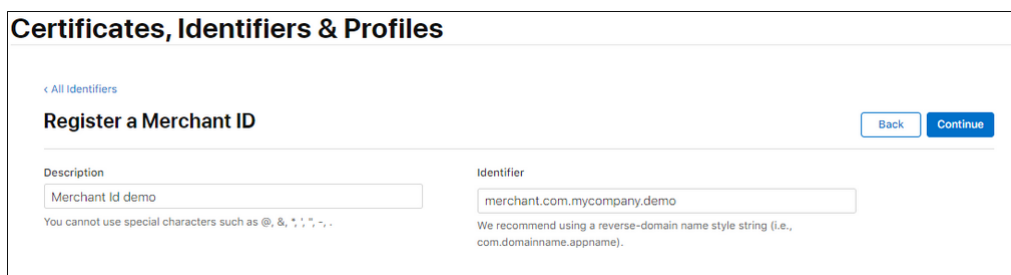
The merchant identifier registration page opens.



6. Select **Merchant IDs** and click **Continue**.

The merchant identifier entry page opens.

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



7. Enter your merchant ID in the **login** 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 on **Continue** to finish.

Your identifier is created. All you have to do is contact adv@lyra-collect.com to communicate your merchant ID.

4.2.2. Declaring an Apple Pay merchant ID in the Expert Back Office

1. Sign in to the [Back Office](#)

2. Click **Other actions** to access your Expert Back Office.

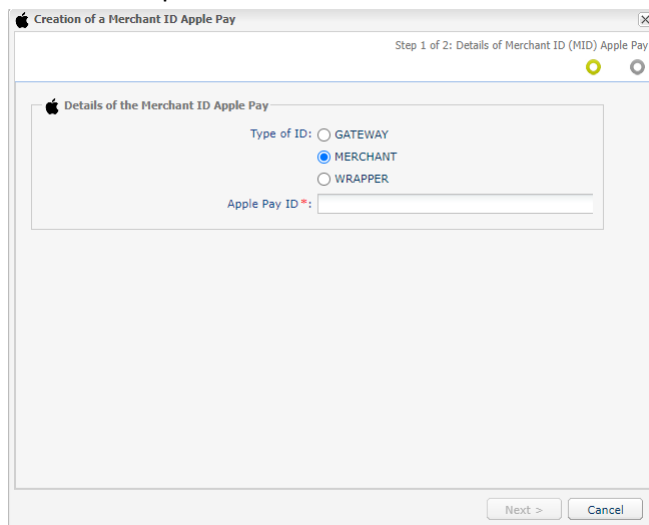
3. Go to **Settings > Company > Merchant IDs**

4. Click **Create a Merchant ID (MID)**.

The **Creation of an Apple Pay Merchant ID** dialog box appears. Merchant ID types are displayed as icons.

5. Select **Apple Pay**.

The merchant ID creation assistant opens.

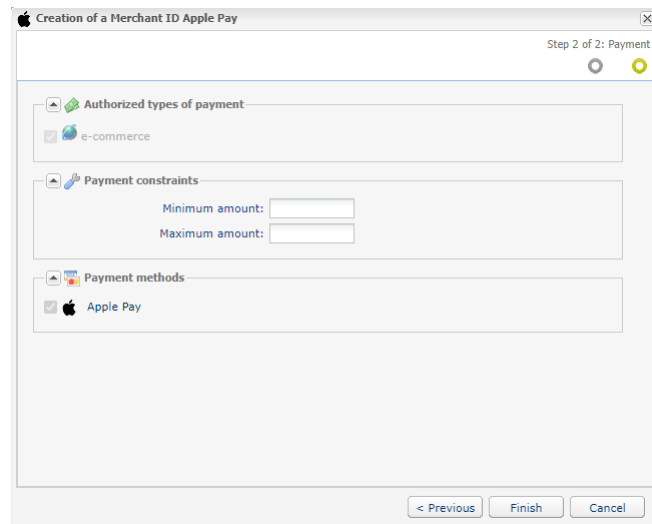


6. Select the **MERCHANT** ID type.

7. Enter your **Apple Pay merchant ID** that you obtained when signing up for Apple Pay Developer.

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

The rest of the setting is displayed.



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

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

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

11. 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 adv@lyra-collect.com for checking your offer.

All that remains is for you to follow the following procedure to finalize the definition of your contract.

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



Once you have declared your Apple Pay merchant ID, you still have a few actions to perform in three different areas to finalize the contract creation:

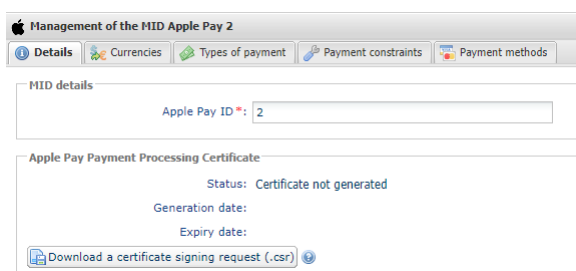
- Your Expert Back Office.
- Your Apple Developer account.
- Your domain server.

To facilitate your setup, make sure you can access the different areas simultaneously.

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

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

The contract details appear.



2. Click on **Download a certificate signature request (.csr)** in the section *Apple Pay Payment Processing Certificate*.

The gateway provides a CSR file to request the certification of **payment processing**. This file is downloaded to a local folder on your computer.

3. Log into your [Apple Pay Developer](#) account to import the downloaded file and then follow these steps to obtain a signed certificate:

- a. Go to *Certificates, Identifiers & Profiles*.
- b. Select *Merchant IDs* in the *App IDs* list.
- c. From the *Merchant ID* page, go to **Apple Pay Payment Processing Certificate**.
- d. Click on *Create Certificate*.

The certificate creation page appears.

- e. Click on *Choose file* then load the downloaded file from your Expert Back Office.

Example: processing.bdupont-1234-apple.csr

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

The signed certificate is generated.

- g. Save the downloaded certificate locally on your computer.

This certificate is mandatory for processing payments between Apple and the payment gateway.

4. Click on **Import the certificate signed by Apple (.cer)** in the section *Apple Pay Payment Processing Certificate*.

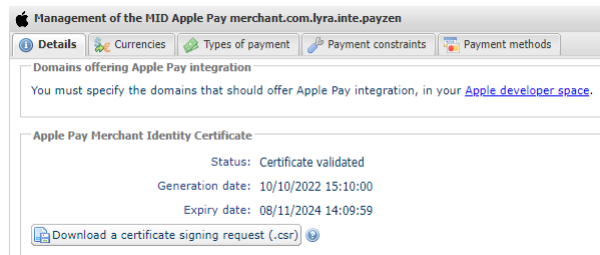
This import button appears as soon as you click on **Download a certificate signature request (.csr)**.

5. Upload the downloaded certificate file to perform the import.

Example: processing.bdupont-1234-apple.csr

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

6. Then click on **Download a certificate signing request (.csr)** in the section *Apple Pay Merchant Identity Certificate*.



The gateway provides a CSR file to request the certification of **the merchant's identity**.



The two uploaded files each have a different prefix followed by the merchant ID to help identify the role of each file.

Examples:

`processing.bdupont-1234-apple.csr`

`identity.bdupont-1234-apple.csr`

7. From your Apple Developer account:
 - a. Go to *Certificates, Identifiers & Profiles*.
 - b. Select *Merchant IDs* in the *App IDs* list.
 - c. From the *Merchant ID* page, go to **Apple Pay Merchant Identity Certificate**.
 - d. Click on *Create Certificate*.

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

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

The signed certificate is generated.
 - g. Save the downloaded certificate locally on your computer.
8. Click on **Import the certificate signed by Apple (.cer)** in the section *Apple Pay Merchant Identity Certificate*.

This import button appears as soon as you click on **Download a certificate signing request (.csr)**.
9. Upload the downloaded certificate file to perform the import.

Example: `identity.bdupont-1234-apple.csr`

The Status of the field changes to **Certificate enabled**.
10. To finalize the declaration of the contract, you still need to fill in the domain(s) offering Apple Pay integration to get their validation from Apple.

You must declare all URLs where Apple Pay will be used and these URLs must be `https`.

 - a. Sign in to *your Apple Developer account*.
 - b. Select *Merchant IDs* in the *App IDs* list.
 - c. Select your *Merchant ID* (see chapter *Creating a merchant ID in Apple Pay Developer* on page 10)
 - d. Select **Add Domain** under the *Merchant Domain* section.
 - e. Fill in your domain name and click save.
 - f. Select **Download** and save the file `apple-developer-merchantid-domain-association.txt` obtained.
 - g. Retrieve the downloaded `apple-developer-merchantid-domain-association.txt` 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.txt>

- h.** Once the file is placed in the **.well-known** directory at the root of your domain, return to your *Merchant Domain* space and click on **Verify**.

The Apple Pay contract declaration is now complete.



Apple will send you an e-mail reminder when the renewal period for a certificate is approaching.

4.3. Contract definition in the merchant mode “Wrapper”

This mode is recommended for example in the context of a group composed of several legal entities that share a single domain name that could be called “master domain”.

A master domain is when the main URL is shared between several clients. For example, all of the group’s clients share a master domain (e.g.: <https://www.domain-name.net>). Each client has their own URL that dissociates it from the other one (e.g.: <https://www.domain-name.net/city1>, <https://www.domain-name.net/city2>, etc.)

The group’s customers who want to set up an Apple Pay contract using the modeMerchantand whose domain depends on the master domain must therefore use this contract definition mode called “**Wrapper**”.

A first contract should be created in theMerchant. This contract may be fictitious or reserved for the group’s first customer.

Thus, only one merchant identifier will be created on the **Apple Pay Developer** account and the URL of the master domain will be declared there (e.g.: <https://www.domain-name.net>).

Then, this merchant ID will be associated with each Apple Pay contract created by a customer with a URL dependent on the master domain.

To associate them, each customer will need the Apple Pay merchant ID created for that master domain and their company ID (SIRET (Système d'Identification du Répertoire des Etablissements)).

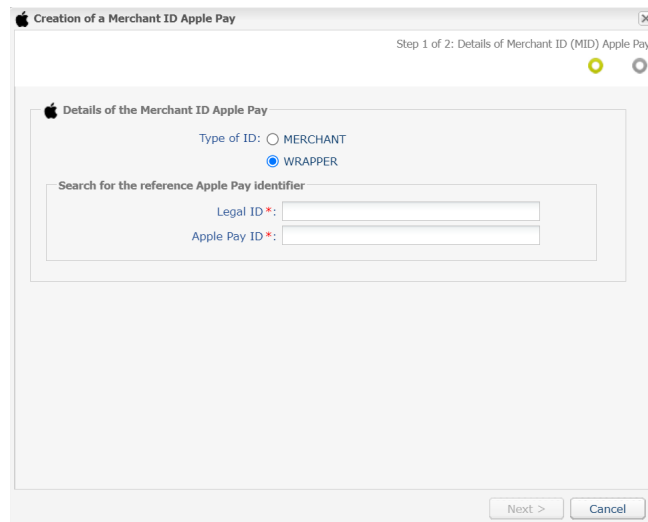
4.3.1. Declaring an Apple Pay merchant ID in “Wrapper” mode



Before you get started, make sure you have the Apple Pay merchant ID number created on Lyra Collect when you signed up for Apple Pay. This Apple Pay merchant ID number was normally created using the Apple Pay merchant ID obtained when the master domain was registered on the Apple Pay Developer website (see [Creating a merchant ID in Apple Pay Developer](#) on page 10).

Also make sure you have your company ID (SIRET (Système d'Identification du Répertoire des Etablissements)). You can copy it from the **Settings** menu > **Company** > tab **Company details**.

1. Sign in to the [Back Office](#)
2. Click **Other actions** to access your Expert Back Office.
3. Go to **Settings** > **Company** > **Merchant IDs**
4. Click **Create a Merchant ID (MID)**.
The **Creation of an Apple Pay Merchant ID** dialog box appears. Merchant ID types are displayed as icons.
5. Select **Apple Pay**.
The merchant ID creation assistant opens.



6. Select the **WRAPPER** ID type.

7. Enter your **Legal ID**.

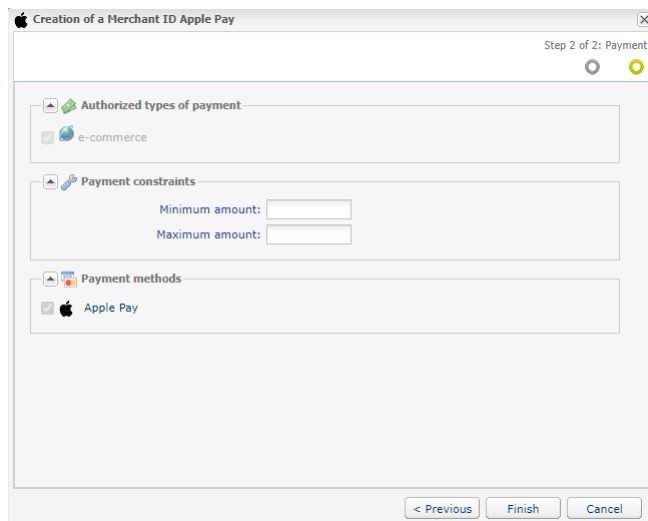
Refers to your Company ID (SIRET (Système d'Identification du Répertoire des Etablissements)).

8. Enter the **Apple Pay ID** associated with the Apple Pay contract of the parent company to which your shop company belongs.

If you do not know this ID, you can contact adv@lyra-collect.com.

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

The rest of the setting is displayed.



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.

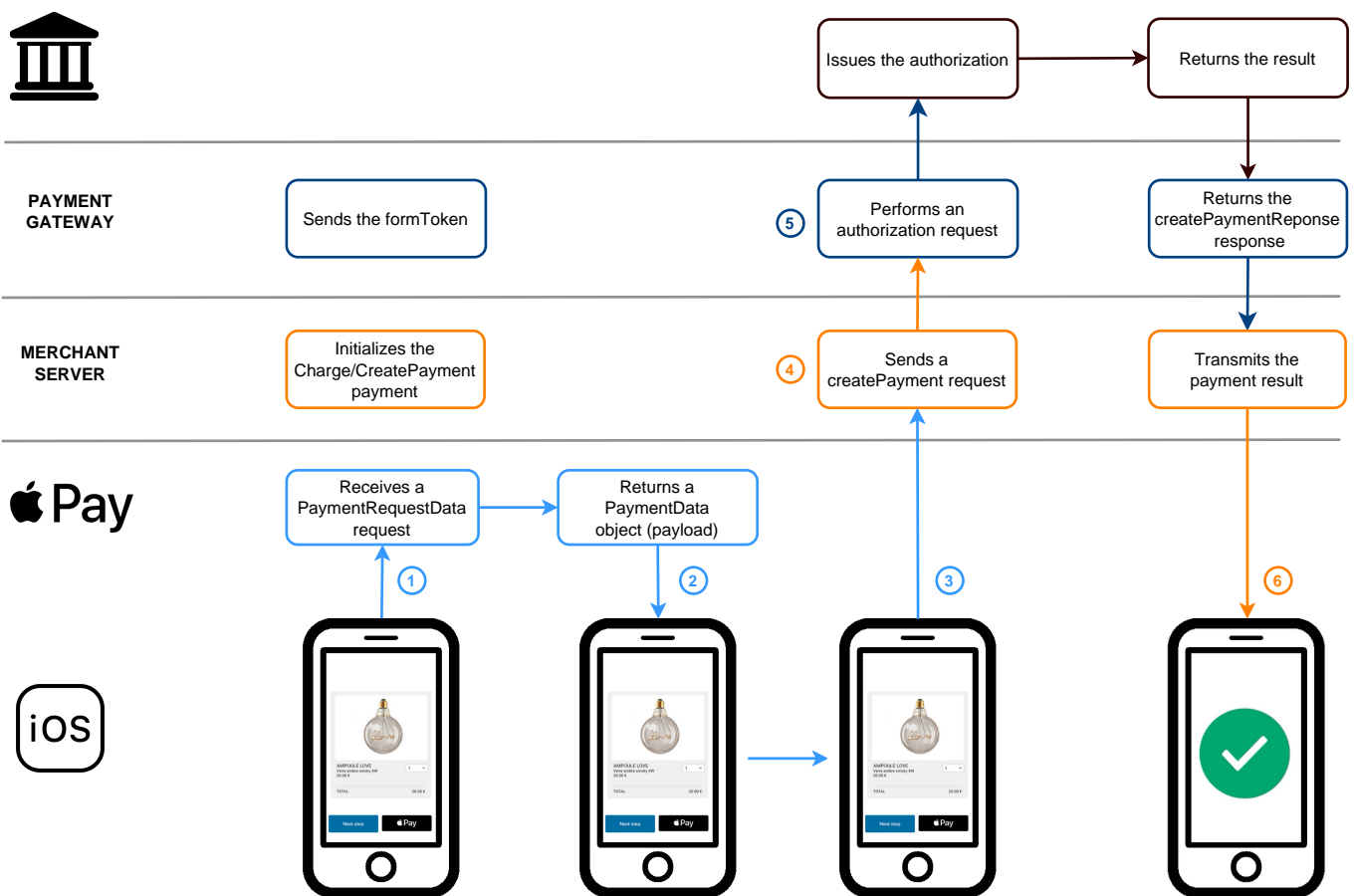
12. Click **Finish**.

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 adv@lyra-collect.com for checking your offer.

5. EXCHANGE PROCESS WITHIN THE MOBILE SDK



1. The mobile application submits a `paymentDataRequest` to Apple Pay API to collect the buyer's card details.
2. The API Apple Pay returns a `PaymentData` object containing the encrypted data (also called "payload").
3. The mobile application transmits the payload to the merchant server.
4. The merchant server creates and submits a `createPayment` request using the payload to populate the `walletPayload` property of the `cardRequest` object. The property `scheme` is valued at `APPLE_PAY`.
The other properties of the `cardRequest` object must not be populated.
5. The payment gateway analyzes the request, decrypts the card data and proceeds to the authorization request. It then transmits the payment result to the merchant server.
6. The merchant server receives a `createPaymentResponse` and analyzes the payment result. It transmits the information to the app that proceeds to process it.

6. ENABLING PAYMENTS VIA APPLE PAY IN THE SDK



Our SDK is compatible starting from **iOS 11.0**

6.1. Prerequisites

Before starting the activation process, you should [consult the Mobile SDK integration guide](#). You can find:

- Prerequisites to follow before starting the SDK integration
- An explanation of how the SDK solution works
- How to add the payment SDK to your application

This guide also contains information on how to make a payment, check the transaction status, customize the SDK.

6.2. Enabling Apple Pay capability in your application

Prior to initializing the SDK, you must enable Apple Pay capability in your application and associate it with the Apple Pay merchant ID

1. Choose the application target in Xcode.
2. Click on the **Signing & Capabilities** tab in the project editor.
3. In the toolbar, click the **Library (+)** button to open the **Capabilities** library.
4. Select the **Apple Pay** capability.
5. In Apple Pay, click the refresh button to sync your merchant credentials from the [Apple Pay Developer](#) website.
6. Once the synchronization is done, select the **Apple Merchant Identifier** number to use with your application.

6.3. Initializing the SDK

It is necessary to initialize the SDK at the launch of your application in the `didFinishLaunchingWithOptions` method of your `AppDelegate`. This allows you to configure the SDK with your data.

1. Import the `LyraPaymentSDK` into your application
2. Call the `initialize` method with the following parameters:

Parameter	Format	Description
<code>publicKey</code>	string	Fill in your public key (available in the Settings menu > Shop > REST API keys your Expert Back Office)
<code>options</code>	[String: Any]	Dictionary that allows you to configure the behavior of the SDK.

The possible keys in this **dictionary** are:

Keys	Format	Description	Required
<code>apiServerName</code>	string	Expected value: REST API server name (available in the Settings menu > Shop > REST API keys of your Expert Back Office)	Required

Keys	Format	Description	Required
cardScanningEnabled	Bool	Enables/Disables the card scan feature (see the iOS integration guide)	Optional
applePayMerchantId	string	Expected value: <code>APPLE_MERCHANT_ID</code> See Configuring the contracts on page 6 chapter to learn more about your <code>APPLE_MERCHANT_ID</code> which is actually your Apple Pay contract number.	Required

Example of a call

```
<div class="doc-component-wrapper"> <doc-code-block> <div class="doc-code" data-
languages="swift, objective_c"> <pre data-language="swift"> //Active Apple Pay
functionality configurationOptions[Lyra.applePayMerchantId] = APPLE_MERCHANT_ID </pre>
<pre data-language="objective_c"> //Active Apple Pay functionality [configurationOptions
setValue:APPLE_MERCHANT_ID forKey:Lyra.applePayMerchantId]; </pre> </div> </doc-code-block>
</div>
```

For more details on the examples according to the calling method (swift,objective_c) see the [iOS integration guide](#).



Optionally, you can configure the text to be displayed as a summary in the payment request. All you need to do is send the text to be displayed as value of the **applePaymentSummaryDescription** key in the option configuration dictionary when calling the `processPayment`.

7. VIEWING THE EXAMPLES OF INTEGRATION CODES

For more guidance, you can refer to additional examples of implementation. See link below:

<https://github.com/lyra/android-ios-sdk-integration-examples/tree/master/ios>

8. MANAGING TRANSACTIONS VIA THE LYRA COLLECT BACK OFFICE

Operations related to transactions must be made via the Merchant Back Office available at this address: <https://secure.lyra.com/portal/>.

However, if you have opted for the MarketPlace offer with LyraCollect, you must use the [MarketPlace API](#).

8.1. Viewing transaction details via Apple Pay

Transactions can be viewed in the Merchant Back Office via the **Transactions** menu.

To view the details of a transaction:

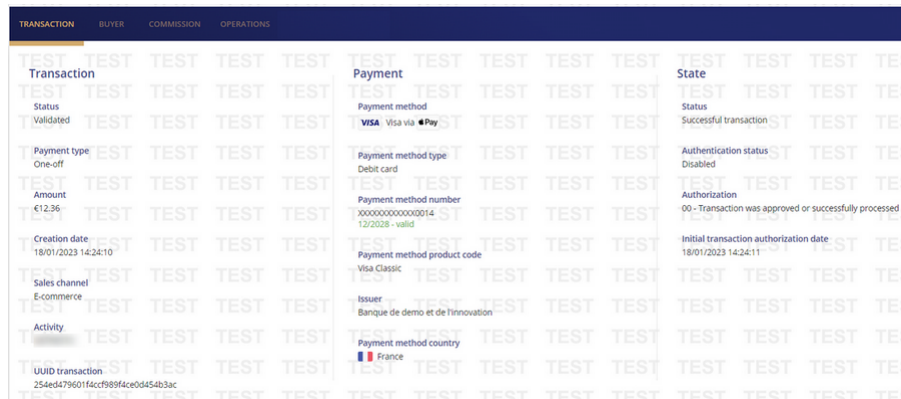
1. Search for the corresponding Apple Pay transaction.

You can refine the search by:

- Using an extended filter for the payment method.
- Specifying the payment date or the order number.

2. Click on the link in the **Order ID** column.

The **Transaction** tab is displayed by default.



The details include:

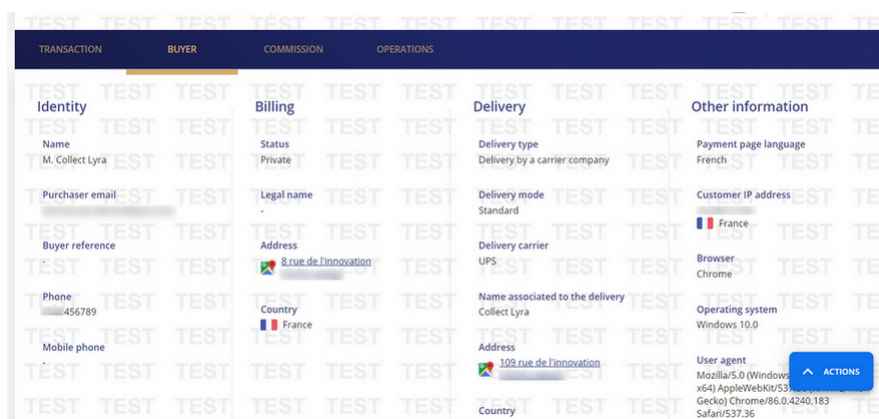
- **The payment status:**

Status	Description
Canceled	The payment has been canceled by the merchant.
Expired	The payment has not been validated by the merchant within the set deadlines.
Declined	The payment has been refused.
Manual review	The payment is waiting for validation by the merchant. Without merchant validation, the payment expires once the capture date arrives.
Validated	The payment has been accepted.

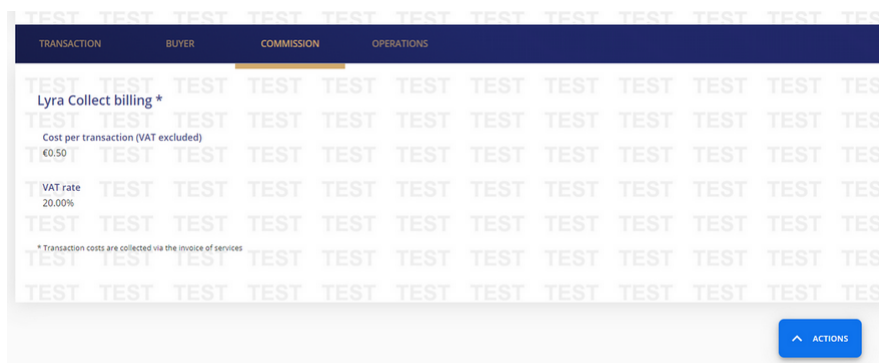
- **The payment type:**
 - **One-off** in case of a payment.

- **Installment** in case of payment in installments.
- **Multi-cards** when several payment methods have been used to pay for the order.
- **Refund** in case of a refund.
- **Subscription** In case of a recurring payment.
- **Chargeback** In case of a chargeback.
- **The transaction amount.**
- **The Lyra Collect transaction number.**
- **The payment method** with the clarification that the payment was made via the **Apple Pay** wallet.

The **Buyer** tab contains data related to the buyer (billing and shipping address) transmitted in your payment request.



The **Commission** tab indicates the amount of the commission as well as the applied VAT rates.



The **Operations** tab presents the different payment types associated with the initial transaction (Refund, Chargeback, Installment, Multi-cards, etc.)

TRANSACTION	BUYER	COMMISSION	OPERATIONS																
<div style="display: flex; justify-content: space-between; align-items: center;"> ↓ COLUMNS </div> <table border="1"> <thead> <tr> <th>Creation date</th> <th>Payment type</th> <th>Amount</th> <th>Payment method</th> <th>Sequence number</th> <th>Status</th> <th>Capture date</th> <th>Acquirer</th> </tr> </thead> <tbody> <tr> <td>10/05/2021 17:14:06</td> <td>One-off</td> <td>€150.00</td> <td></td> <td>1</td> <td>Validated</td> <td>10/05/2021</td> <td>Lyra Colle</td> </tr> </tbody> </table>				Creation date	Payment type	Amount	Payment method	Sequence number	Status	Capture date	Acquirer	10/05/2021 17:14:06	One-off	€150.00		1	Validated	10/05/2021	Lyra Colle
Creation date	Payment type	Amount	Payment method	Sequence number	Status	Capture date	Acquirer												
10/05/2021 17:14:06	One-off	€150.00		1	Validated	10/05/2021	Lyra Colle												

8.2. Canceling a transaction

If you have opted for the MarketPlace offer with LyraCollect, you must use the [MarketPlace API](#) to cancel your transactions.


The **Cancel** operation is only available for the transactions that have not been captured.

In order to cancel a transaction:

1. Display the **Transactions** tab.
2. Select the transaction.

Click on the **Filters** icon () to search for the relevant transaction if it is not on the transactions list.

3. Display the **contextual menu**.

Click on the  icon.

4. Click **Cancel**.
5. Click on **OK** to save your request.

After the cancellation, the payment status becomes **Canceled**

8.3. Modifying a transaction

If you have opted for the MarketPlace offer with LyraCollect, you must use the [MarketPlace API](#) to edit your transactions.

The **Modify** option is only available for payments with the status:

- Manual review
- Manual review, pending authorization
- Waiting for authorisation
- Validated, as long as the capture date has not arrived.


To modify a transaction:

1. Display the **Transactions** tab.

2. Select the transaction.

Click on the **Filters** icon () to search for the relevant transaction if it is not on the transactions list.

3. Display the **contextual menu**.

Click on the  icon.

4. Click **Modify**.

5. Enter a new amount.

The new amount must be lower than the initial amount.

6. Enter the new capture date.

7. Click on **OK** to save your request.

8.4. Duplicating a transaction



The action “**Duplicate a transaction**” is not yet available for Apple Pay.

8.5. Making a refund

If you have opted for the MarketPlace offer with LyraCollect, you must use the [MarketPlace API](#) to refund your transactions.


To initiate a refund request :

1. Display the **Transactions** tab.

2. Select the transaction.

Click on the **Filters** icon () to search for the relevant transaction if it is not on the transactions list.

3. Display the **contextual menu**.

Click on the  icon.

4. Click **Refund**. The dialog box **Request a refund for this transaction** appears.

5. Enter the amount to be refunded. The amount to refund cannot be greater than the transaction initial amount.

6. Click on **OK** to save your request.

8.6. Validating a transaction

If you have opted for the MarketPlace offer with LyraCollect, you must use the [MarketPlace API](#) to validate your transactions.

The validation mode is automatic for immediate payments.

As for deferred payments, the merchant has the possibility to choose between manual and automatic validation.

If the merchant has opted for manual validation, they will have to validate the payment in their Merchant Back Office.

In this case, the transaction status will be:

- **Manual review** If the initial capture delay is shorter than the authorization validation period.
- **Manual review, pending authorization** If the initial capture time is longer than the authorization validation period.

To validate a transaction:

1. Display the **Manual review** tab.

2. Select the transaction.

Click on the **Filters** icon () to search for the relevant transaction if it is not on the transactions list.

3. Display the **contextual menu**.

Click on the  icon.

4. Click **Apply**.

5. Click on **OK** to save your request.

After validation,

- the status of a payment with a capture delay that is longer than the authorization validation period is **Waiting for authorisation**.
- the status of a payment with a capture delay that is shorter or equal to the authorization validation period is **Validated**.

9. OBTAINING HELP

Looking for help? Check our FAQ on our website

<https://docs.lyra.com/en/collect/faq/sitemap.html>

For any technical inquiries or if you need any help, contact *technical support*.

To help us process your demands, you will be asked to communicate your customer code (e.g.: **CLXXXXX**, **MKXXXXX** or **AGXXXXX**).

This information is available in the Merchant Back Office (top of menu).