



IATA Pay Airlines Integration Guide

v2.64



Table of Contents

Change History	5
1 Background Information	8
1.1 IATA	8
1.1.1 History	8
1.1.2 Today	8
1.1.3 Size and Locations	9
1.2 IATA Pay	9
1.2.1 Why IATA Pay?	9
2 Purpose of Document	11
2.1 Sections	11
3 Payment Journey	12
4 Basic Concepts	13
4.1 Actors	13
4.2 Transactions	14
4.3 Transaction Statuses	15
4.3.1 Payment Transaction flow	15
4.3.1.1 Unexpected Settled payment	16
4.3.2 Refund Transaction Flow	18
4.4 Failure Codes	18
4.5 Timeouts	19
5 Payment Transactions	20
5.1 Actors	20
5.2 Prerequisites / Rules	20
5.3 Steps	20
5.3.1 Creation	21
5.3.2 Checkout	21
5.3.3 Authorization	21
5.3.4 Confirmation	21
5.3.5 Reconciliation	21
5.4 Checkout and Authorization Methods	22
5.4.1 Redirect Way	29
5.4.2 Light Box Way	30
5.5 Payment Workflow example	33
5.6 API Summary	36
5.6.1 Request - Data from the Airline to IATA Pay	36
5.6.2 Response - Data from IATA Pay to the Airline	38
5.7 Dummy values for testing	39

5.8	Finish Notification	40
5.9	Languages	42
6	Refund Transactions	43
6.1	Actors	43
6.2	Prerequisites / Rules	43
6.3	Refund Workflow	44
6.4	Refund Batches	45
6.5	API Summary	45
6.5.1	Request - Data from the Airline to IATA Pay	45
6.5.2	Response - Data from IATA Pay to the Airline	46
6.6	Finish Notification	47
7	Searches	48
7.1	Payments	48
7.2	Refunds	48
7.3	Refund Batches	49
7.4	Finished Transactions	49
7.4.1	API Summary	49
7.4.1.1	Request - Data from the airline to IATA Pay	49
7.4.1.2	Response - Data from IATA Pay to the airline	50
8	API Specification	51
8.1	Environments	51
8.2	Authentication	51
8.2.1	Client Credentials	51
8.2.1.1	Tokens	52
8.2.1.2	Steps to get an access token	52
8.2.1.3	Using an access token	53
8.3	Protocol	54
8.4	Content Type	54
8.5	Swagger Specification	54
8.6	Notification Specification	55
8.6.1	Notification signature spec	55
8.6.2	Notification retries	56
8.7	Basic flows testing	57
8.7.1	Client login	57
8.7.2	Payment initiation	57
8.7.3	Payment created	59
8.7.4	Checkout (with redirect)	59
8.7.5	Payment status notification	60
8.7.6	Validation Endpoint	60

8.7.7	Refund initiation	65
8.7.8	Refund created	66
8.7.9	Refund status finish notification	66
8.7.10	Ticket Purchase enabled	67
8.7.11	Error / Failure codes simulation	74
9	IATA PAY in Call Centers	78
9.1	Payment creation via API according to the swagger specs	78
10	IATA PAY Testing Scenarios	85
10.1	Functional Testing Scenarios	85
10.2	Testing Scenario IPTS-AUT-01. Obtain Access Token	86
10.3	Testing Scenario IPTS-PCR-01. Payment created successfully	87
10.4	Testing Scenario IPTS-PCK-01. User redirected successfully to authorize payment on IATA PAY	89
10.5	Testing Scenario IPTS-PCK-02. Airline successfully show IATA PAY lightbox to authorize payment	91
10.6	Testing Scenario IPTS-PAU-01. User successfully authorize payment	93
10.7	Testing Scenario IPTS-PAU-02. User cancel authorization	96
10.8	Testing Scenario IPTS-PAU-03. User close browser	99
10.9	Testing Scenario IPTS-PAU-04. User do not authorize payment	101
10.10	Testing Scenario IPTS-PAU-05. Unexpected Error	103
10.11	Testing Scenario IPTS-REF-01. Refund created/processed successfully	106
10.12	Testing Scenario IPTS-REF-02. Error on refund creation (amount exceeds original payment).	108
10.13	Testing Scenario IPTS-ADM-01. User can login successfully on IATA PAY Admin Portal	109
10.14	Testing Scenario IPTS-ADM-02. User can review payments processed	113
10.15	Testing Scenario IPTS-ADM-03. User can review refunds	115
10.16	Testing Scenario IPTS-ADM-04. User can download CSV report for payments/refunds	117

Change History

Issue number	Dated	Reason for revision
V2.46	2024/01/15	<ul style="list-style-type: none"> Updated “notificationUrl” field as mandatory (Updated by Paloma López Fernández and Naiara Rodríguez Gutiérrez)
V2.47	2024/02/12	<ul style="list-style-type: none"> Updated failure codes and failure codes simulation sections Updated 8.7.6 VPA validation endpoint (Updated by Paloma López Fernández and Naiara Rodríguez Gutiérrez)
V2.48	2024/03/06	<ul style="list-style-type: none"> New Chapter IATAPAY Test Scenarios for Airline Integration in Sandbox environment Updated 8.7.6 Validation endpoint Updated 5.7 Dummy values for testing Updated 8.1 Environments (Updated by Paloma López Fernández and Naiara Rodríguez Gutiérrez)
V2.49	2024/04/18	<ul style="list-style-type: none"> Updated 8.2.1.2 Section Updated 5.4 Section Updated 5.6.1 Section Updated 8.7.6 Section Updated 9.1 Section (Updated by Paloma López Fernández and Naiara Rodríguez Gutiérrez)
V2.50	2024/05/21	<ul style="list-style-type: none"> Updated 6.2 Prerequisites/Rules section Updated preferredCheckoutMethod field (Updated by Paloma López Fernández and Naiara Rodríguez Gutiérrez)
V2.51	2024/06/10	<ul style="list-style-type: none"> Reviewed all sections of the document (Updated by Paloma López Fernández and Naiara Rodríguez Gutiérrez)
V2.52	2024/06/25	<ul style="list-style-type: none"> Updated Dummy Values for testing (Updated by Paloma López Fernández and Naiara Rodríguez Gutiérrez)

V2.53	2024/06/30	<ul style="list-style-type: none"> Reviewed all sections of the document (Updated by Teresa Ruiz-Rico)
V2.54	2024/07/30	<ul style="list-style-type: none"> Updated all the screenshots related to the new Admin Portal version Updated 5.6 API Summary Updated 9 IATA PAY in call centres (Updated by Naiara Rodríguez Gutiérrez)
V2.55	2024/08/20	<ul style="list-style-type: none"> Reviewed all sections of the document (Updated by Naiara Rodríguez Gutiérrez)
V2.56	2024/09/10	<ul style="list-style-type: none"> Reviewed all sections of the document (Updated by Naiara Rodríguez Gutiérrez)
V2.57	2024/10/04	<ul style="list-style-type: none"> Reviewed all sections of the document (Updated by Naiara Rodríguez Gutiérrez)
V2.58	2024/11/06	<ul style="list-style-type: none"> Reviewed all sections of the document Updated 4.3.2 Refund Transaction Flow section (Updated by Naiara Rodríguez Gutiérrez)
V2.59	2024/12/10	<ul style="list-style-type: none"> Updated all the screenshots related to filters from sections 10.15 and 10.16 (Updated by Naiara Rodríguez Gutiérrez)
V2.60	2025/01/08	<ul style="list-style-type: none"> Reviewed all sections of the document (Updated by Naiara Rodríguez Gutiérrez)
V2.61	2025/02/04	<ul style="list-style-type: none"> Updated values from API Summary chapter Updated 10.6 bizum typo error Updated Payment details screenshots (Updated by Naiara Rodríguez Gutiérrez)
V2.62	2025/03/04	<ul style="list-style-type: none"> Reviewed all the sections of the guide (Updated by Naiara Rodríguez Gutiérrez)

V2.63	2025/04/11	<ul style="list-style-type: none">• Updated graph from 4.3.1 section• Updated 5.6 section <p>(Updated by Naiara Rodríguez Gutiérrez)</p>
V2.64	2025/05/07	<ul style="list-style-type: none">• Reviewed all the sections of the guide <p>(Updated by Naiara Rodríguez Gutiérrez)</p>

1 Background Information

1.1 IATA

1.1.1 History

The International Air Transport Association (IATA) is the global association of the world's airlines. It is the successor to the International Air Traffic Association which was founded in The Hague in 1919 - the year of the world's first international scheduled services.

IATA is a not-for-profit organization and was founded in Havana, Cuba, in April 1945. It is the prime vehicle for inter-airline cooperation in promoting safe, reliable, secure, and economical air services - for the benefit of the world's consumers. At its founding, IATA had fifty-seven (57) Members from thirty-one (31) nations, mostly in Europe and North America.

1.1.2 Today

The international scheduled air transport industry is now more than one hundred times larger than it was in 1945. Few industries can match the dynamism of that growth, which would have been much less spectacular without the standards, practices and procedures developed within IATA.

IATA's 250+ member airlines comprise some 84% of scheduled traffic. The Headquarters are located in Montreal, Canada and our Executive Office is in Geneva, Switzerland. Among IATA's other sixty (60) offices worldwide, it has major regional offices in Singapore, Beijing, Madrid, Amman, Miami and Johannesburg.

At IATA, we work with our members to make our industry safer, more profitable and efficient. Areas of involvement include the following:

- Aircraft Operations
- Airport Development and Infrastructure
- Cargo
- Finance
- Industry Initiatives
- Passenger
- Regulatory and Public Policies
- Safety
- Security and Facilitation
- Simplifying the Business

IATA's wide range of products and services can support the ongoing efforts of many functions within our members' organisation: finance, security, operations, strategic management, safety and much more. Our activities cross all the travel industry's stakeholders.



For more detailed information on our organisation, including a description of our key projects and an organisation chart, please visit our website: www.iata.org.

1.1.3 Size and Locations

IATA has over 1,600+ staff spread across many locations worldwide with the current distribution being:

- Head office in Montreal.
- Executive Offices in Geneva.
- Five (5) Hubs in Amman, Beijing, Madrid, Miami, and Singapore.
- More than fifty (50+) local offices.

1.2 IATA Pay

Open Banking regulations that create opportunities for new payment methods are being introduced in many countries. At the same time, new mobile and peer-to-peer instant payment options are very popular, particularly with millennials. IATA Pay is an industry-supported initiative that responds to these twin developments by offering a new payment method for consumers when purchasing a ticket from an airline website.

IATA Pay responds to rapid changes in the payments landscape.

- Consumers needs are evolving, for instance with millennials' expectations for mobile and peer-to-peer instant payment.
- New banking regulations in many countries encourage the use of bank account-to-bank account direct payments. Examples of these regulations are the EU's Payment Services Directive 2 (PSD2) and the UK's Open Banking regulations.

1.2.1 Why IATA Pay?

Payment costs in the airline industry are significant; credit card interchange fees alone are averaging around US \$8 billion per year (Source: Phocuswright) and this is not including charge backs and fraud-related costs. These costs are forecast to increase significantly in the coming years.

The current range of payment methods also has a significant impact on the airlines' working capital due to the time it takes to settle a payment after it is made.

IATA Pay addresses these issues and aims to create an alternative solution that is cost-effective, highly secure, and improves working capital.

For airlines, the advantages of IATA Pay are:

- Cheaper payment option compared to other alternatives.
- Highly secure.



- Faster cash flow to the merchant.
- Simpler payment process resulting in fewer lost sales.
- Optional, white-labelled payment method.

For consumers the benefits include:

- Highly secure.
- Enhanced consumer experience.
- Seamless user experience.
- Real time refunds.

2 Purpose of Document

The objectives of this document are:

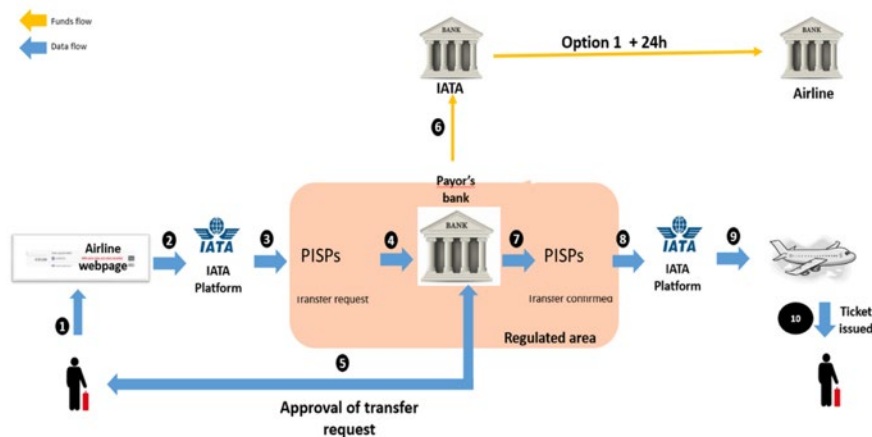
- To be the primary source for the definition of the rules and obligations of IATA Pay Platform.
- To provide authoritative information to airlines, TPPs and other relevant parties about how the Scheme works.
- To provide involved parties with relevant information to support development and operational activities.

2.1 Sections

The document is mainly divided into the following sections:

- Basic concepts. It lists which are the actors involved when IATA Pay is used, and which are the supported money transactions –payments and refunds of these payments– and their statuses.
- Payment transactions. It details the whole workflow of a payment, from the decision of the airlines' users to pay with IATA Pay until the redirection to the confirmation page of the airline. This section also explains how the airline needs to be integrated with IATA Pay and the different ways to do it.
- Refund transactions. It describes how to execute refunds of previous payments, either individually or in a batch, and the associated workflow.
- Searches. Once a payment or a refund is created, it can be searched to recover its info.
- API specification. This section is highly technical. It includes links to a Swagger UI to browse the complete API of IATA Pay and it explains how to be authenticated to be able to send requests.

3 Payment Journey



1. Users select IATA Pay as the payment method.
2. The airline transfers to IATA platform (IATA Pay) the payment details. IATA platform performs some validations and identify the PISP (Payment Initiation Service Providers) that will process the transaction.

Regulated area

3. The PISP receives the payment initiation request. It performs some checks as per regulations and send the transfer initiation request to the payer's bank.
4. Payer's bank receives the request from the PISP. As per the regulation, it checks in the PISP directory if this request has been sent from a provider which has a valid licence to do this action.
5. The bank contacts its customer (the payer for that sale transaction) to authenticate/approve the payment transfer initiation.
6. Once the approval of the payment initiation is received from the payer, the bank will initiate an instant payment transfer to IATA's bank account. IATA will reconcile and transfer to the airline 24h later leveraging in the current banking infrastructure utilized for ISS.
7. Payer's bank confirms the approval of the payment request and the initiation of the instant bank transfer to the PISP.

End of regulated area

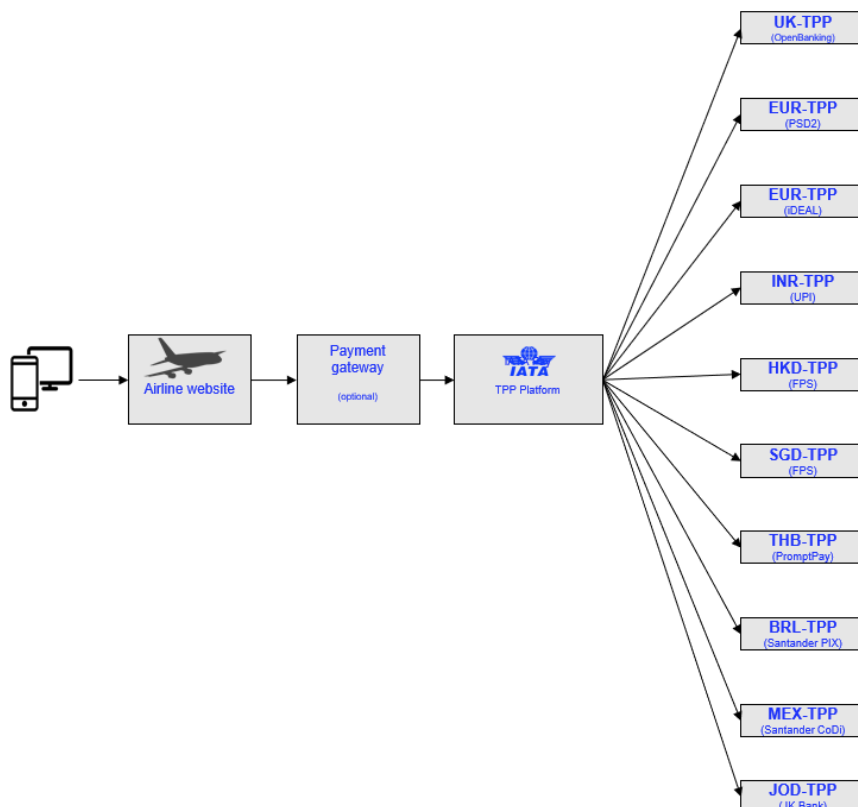
8. The confirmation of the payment is told to IATA Pay.
9. The confirmation of the payment is forwarded to the airline.
10. The airline issues the ticket.

4 Basic Concepts

4.1 Actors

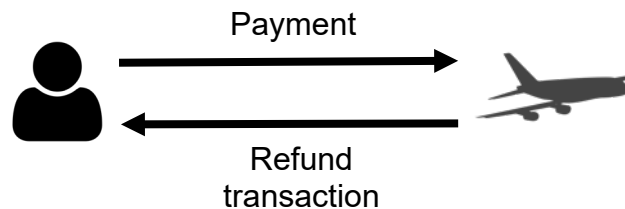
The execution of an IATA Pay operation involves the following actors:

- **User, passenger and payer.** The user is the one who wants to buy a ticket or other airline's services. In the case of buying a ticket, the user might not be the passenger. Also, the user might not be the payer, that is, the one who pays for the ticket/service. However, the user, the passenger and the payer use to be the same person.
- **Airline website.** Airline point of sales in Internet for direct sales transactions.
- **Payment gateway.** Piece of software that communicates the airline payment page with the acquirers (internal or external to airline). Some airlines use these gateways, while others don't. Since this actor is not always present, in this document we will consider that the airlines are the ones that communicate with IATA Pay, although it is done by a gateway.
- **IATA Pay system.** Acquirer platform which performs account to account operations.
- **TPP/PISP.** Regulated entity that processes the IATA Pay's operations.
- **Payer's bank.** Bank of the person who pays.
- **IATA's bank.** IATA's bank account where the funds will arrive after the operation is settled (payment request) and the origin account from where the funds are provided to perform a refund operation.

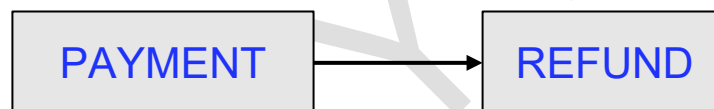


4.2 Transactions

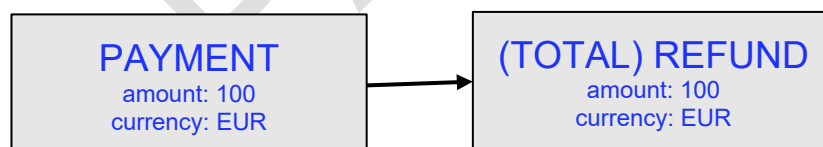
A **transaction** is an instance of money transfer between two parts. It can be a “payment” (user as payer and airline as recipient) or a “refund” (airline as payer and user as recipient):



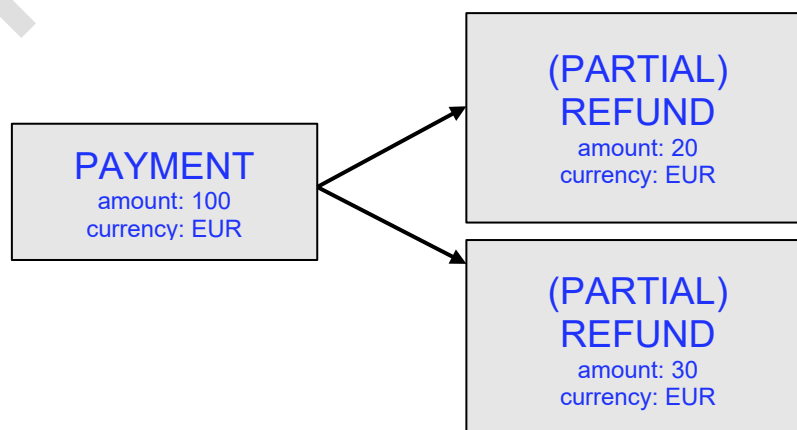
- **Payment transaction.** Money transaction **from the payer to the airline**. It is usually generated when the user buys a ticket or an airline’s service. A payment may have zero, one or more “refund” transactions.
- **Refund transaction.** Money transaction **from the airline to the payer**. It may be a total refund equal to the payment total amount or a partial refund of an amount lower than the payment total amount, on the condition that the total amount of all not failed refunds must be lower or equal to the payment total amount (the failed refunds don’t count).



Relation between payments and refund transactions.



Example of a payment with a refund of the total amount.



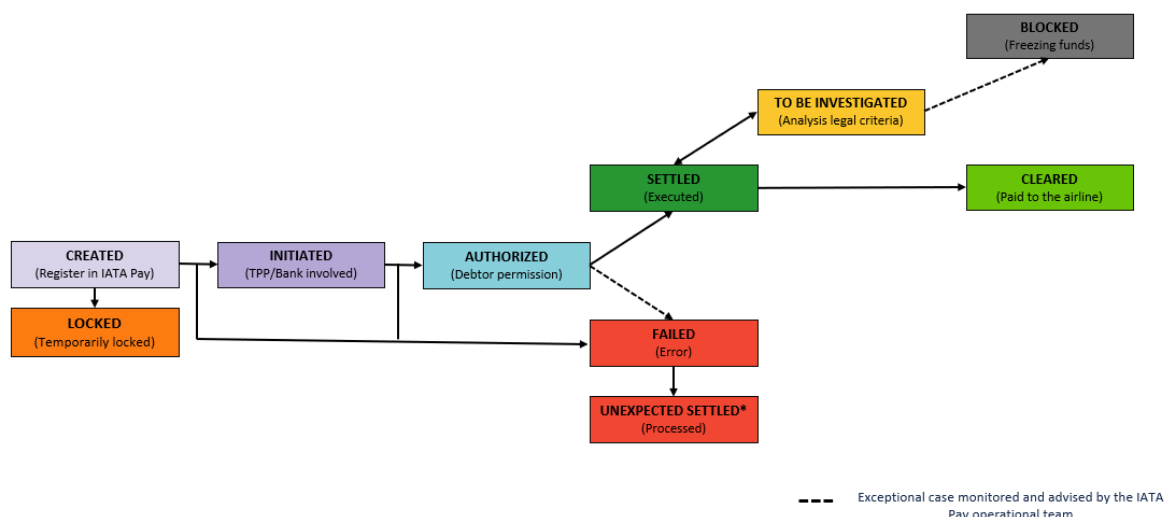
Example of payment with partial refunds.

4.3 Transaction Statuses

The status of a transaction (payment or refund) may be one of the following ones:

- **CREATED.** The transaction has been created but not approved by the debtor.
- **LOCKED.** Only for refunds. The transaction has been temporarily locked.
- **INITIATED.** The transaction has been requested to the TPP/bank.
- **AUTHORIZED.** The transaction has been authorized by the debtor. The paid service could be provided (for example the ticket can be issued).
- **SETTLED.** The transaction has been executed and the funds have arrived at IATA account
- **FAILED.** The transaction process has failed.
- **CLEARED.** The transaction has been paid to the airline.
- **UNEXPECTED SETTLED.** Transaction with **FAILED** status force its status to **UNEXPECTED SETTLED**
- **TOBEINVESTIGATED.** Transaction with **SETTLED** status whose user, passenger or payer must be investigated for possibly not meeting legal criteria.
- **BLOCKED.** Transaction with previous status **TOBEINVESTIGATED**, which has not been able to pass the legal compliance process. Freezing fund until legal resolution

4.3.1 Payment Transaction flow



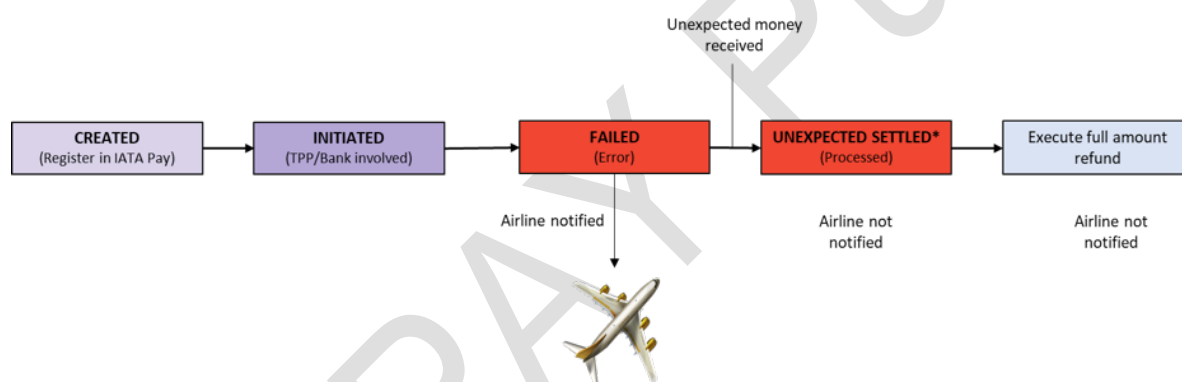
In addition, a transaction is considered as **FINISHED** when it has *SETTLED*, *UNEXPECTED SETTLED*, *BLOCKED*, *FAILED* or *CLEARED* status.

A **call-back notification** will be sent to the airline when the transaction changes to the statuses *AUTHORIZED*, *SETTLED*, *CLEARED* or *FAILED* (this will be notified to the airline but first the field `notificationUrl` which is mandatory must be completed by the airline)

4.3.1.1 Unexpected Settled payment

In rare cases a payment that is already in failed status could be processed successfully, therefore money from that payment is received by IATA Pay system.

In that case the payment will adopt the proper successful status (*AUTHORIZED* or *UNEXPECTEDSETTLED* status) and automatically perform a refund of all the amount the payment has. Like any refund, it will be executed only when the payment is in *UNEXPECTEDSETTLED* status. These status changes and refund initiation will not be notified to the airline, in order to avoid confusion, as it was previously notified of the failure of the payment and no funds will be received by the airline from that payment, so it has to take no action on it.



These status changes will be visible on the Admin Portal, on the payment events section, logging the status change as an event and registering the automatic refund execution in the refunds section so an operator can check it if needed.



Unexpected settled payment status example

Single Refund Multiple Refund

Export refunds to CSV Refresh

17

Refund details

Details

Events

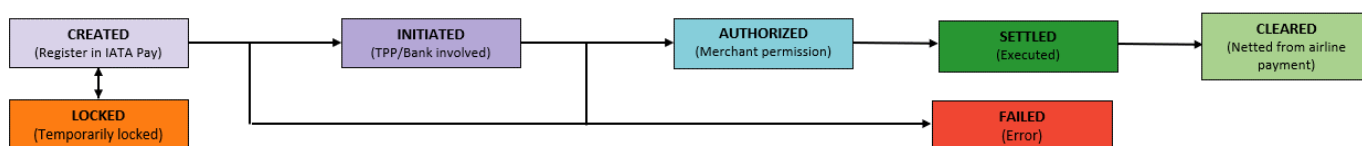
Cancel refund

Merchant	XX0000000
Merchant Refund Id	Not defined
IATA Refund Id	R9QCOARQS2IJZ
Settlement Id	
Amount	A\$1.01
Country	AU
Merchant Payment Id	IDBV589WOTK1L6K67QBE
IATA Payment Id	P5MTVJUH58BTN
Amount	A\$1.01
Description	Refund Lambda Airlines
Status	<div><div></div>UNEXPECTEDSETTLED</div> Cause
Creation date	05/22/2024, 3:22 PM
Finish date	05/22/2024, 5:01 PM
Last update date	05/22/2024, 5:01 PM
Clearance date	-

Automatic Refund detail for Unexpected Settled payment status example

These transactions are included in Airline reconciliation/transactional reports for transparency and potential need in case of end customer queries. Impact in net amount is Zero as there should be always a Payment and its refund for the same amount. Both Payment and Refund should have Unexpected Settled status.

4.3.2 Refund Transaction Flow



In addition, a refund transaction is considered as **FINISHED** when it is **SETTLED** or **FAILED**.

Refund will be marked as **CLEARED** with the same process as for Payments (T+1 working days) by netting these against the positive sales reported as **CLEARED**

4.4 Failure Codes

Every FAILED transaction will have one of the following “failure codes”:

- **CANCEL**. The user has cancelled the transaction.

- *TIMEOUT_CREATED*. The transaction has exceeded the maximum waiting time (10 minutes) with the status *CREATED*.
- *TIMEOUT_INITIATED*. The transaction has exceeded the maximum waiting time (18 minutes) with the status *INITIATED*.
- *TIMEOUT_AUTHORIZED*. The transaction has exceeded the maximum waiting time (the time depends on the transaction channel) with the status *AUTHORIZED*. This is an unexpected error very unusual that requires a cancelation of the purchased service if this has been already issued.
- *TIMEOUT_LOCKED*. Error produced only on refunds when has exceeded the maximum waiting time with the status *LOCKED*. This can be set according to the merchant's preferences.
- *TPP ERROR*. Unexpected error produced usually on transaction initiation on TPP / PISP side.
- *UNEXPECTED*. Unknown error. Unexpected error produced usually on transaction initiation on TPP / PISP provider or in IATA PAY.
- *PAYMENT AUTHORIZATION FAILED*. The TPP reject the authorization of the payment.
- *PAYMENT URL RELOADED*. The user refreshes the purchase page.
- *INVALID TOKEN ID*. The token id introduced by the airline is incorrect.

4.5 Timeouts

Some transaction status has a maximum time for their execution.

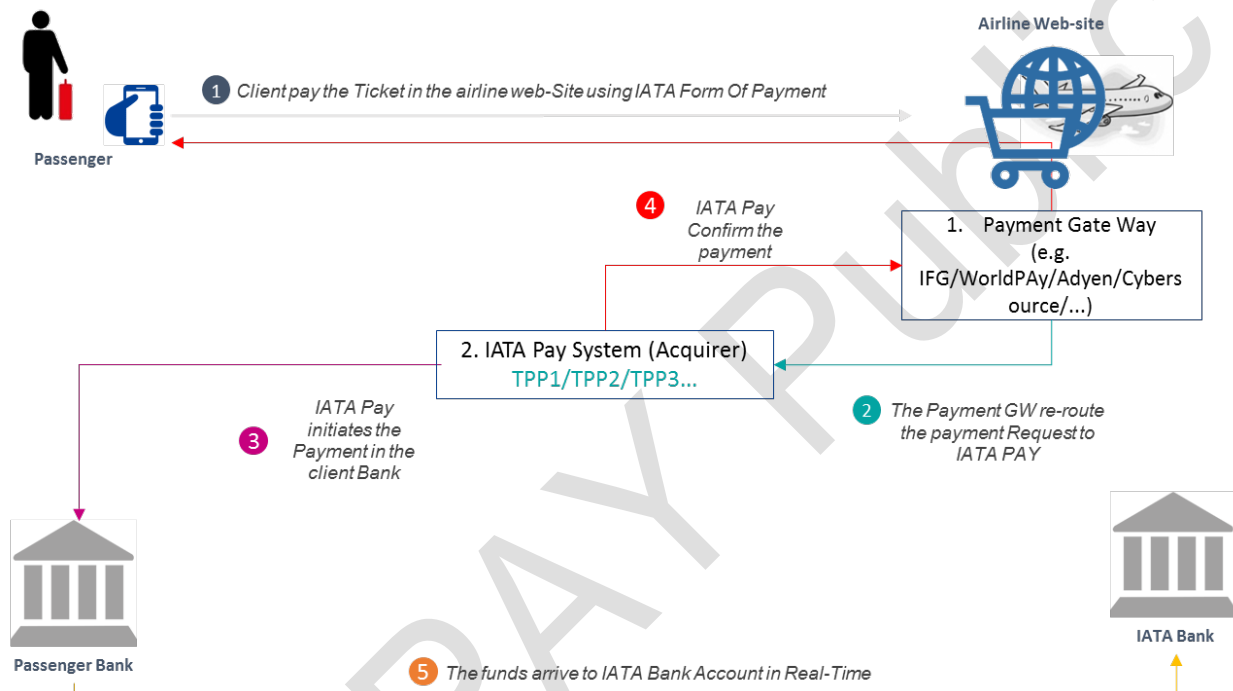
For payments:

- *CREATED*. The transaction exceeds the maximum waiting time if it stays longer than 10 minutes on this status. If the payment was created with a "retriable until date" then the minimum timeout is the given date.
- *INITIATED*. The transaction exceeds the maximum waiting time if it stays longer than 18 minutes on this status.
- *AUTHORIZED*. The transaction exceeds the maximum waiting time if it stays longer than X time on this status, depending on the transaction channel:

5 Payment Transactions

5.1 Actors

The following diagram illustrates the roles of the actors identified in the payment transaction:



5.2 Prerequisites / Rules

- The payer's bank account must be reachable by the IATA Pay Platform. IATA Pay is able to route the payment request to a regulated TPP in the territory where the licence is valid.
- IATA Pay is integrated with the payment gateway that the airline uses or directly with the airline.

5.3 Steps

The payment flow has several steps that conform the complete flow of calls made internally by the system, that means that every step doesn't have to have a screen, user interaction or any other form of visualization associated, as some steps will be done in a different manner depending on the regulation. The steps are:





5.3.1 Creation

This is the first step and the most important one for the airline, as it is the one that has to do the payment creation process calling POST [/api/v1/payments/](#) endpoint, documented inside IATA Pay API. To do the call and get to the next step, the airline has to be authenticated in the API with his client credentials. The fundamental part of this step is obtaining the IATA payment id, as that indicates that the payment is created and will be the id that can be used, alongside the merchant id (if indicated in the payment endpoint call), to follow the steps and check the payment.

5.3.2 Checkout

In this step the user fills the required fields to initiate the payment authorization. The fields required to the user will depend on the user account country regulation, there might be some cases where the user doesn't have to fill anything, and this step will just redirect to the next one.

5.3.3 Authorization

This step is solely focused on the payment authorization by the user. The usual flow will be redirecting the user to his bank, where the user will login (including MFA if the bank has it) and confirm the payment. In some occasions, this step will be shown into an iframe. In some other cases, the flow will be decoupled, where the user could just scan a QR code with a device and confirm the payment that way.

5.3.4 Confirmation

The user is redirected to the airline desired confirmation web page (independently of the checkout and authorization method), that was indicated in the payment API call (fields "successUrl" and "failureUrl") done in the creation step. This confirmation page could be the successUrl or failureUrl depending on the outcome of the payment. Also, as indicated in the payment API call (field "notificationUrl" which is mandatory), the airline will receive a webhook request informing about the finish status of payment, so the airline doesn't have to call repeatedly to check the status of the payment.

5.3.5 Reconciliation

Daily, IATA will pay to the airline a reconciliation amount. That amount will consist of the summatory of all successful payment transactions subtracting all refund transactions. The transactions to be involved must be confirmed (money reached IATA's bank account) and verified by IATA platform.

IATA Pay provides access in real time to the status of the transactions. Also, via different ways, Airline can make queries or receive transactional reports useful for different Airline departments such as Revenue Accounting, Account Receivables and Treasury.

5.4 Checkout and Authorization Methods

When the user browses the airline's payment webpage and selects IATA Pay to buy tickets or the airline's services, there are two alternatives to ask them for their required information to initiate the payment.

Checkout methods:

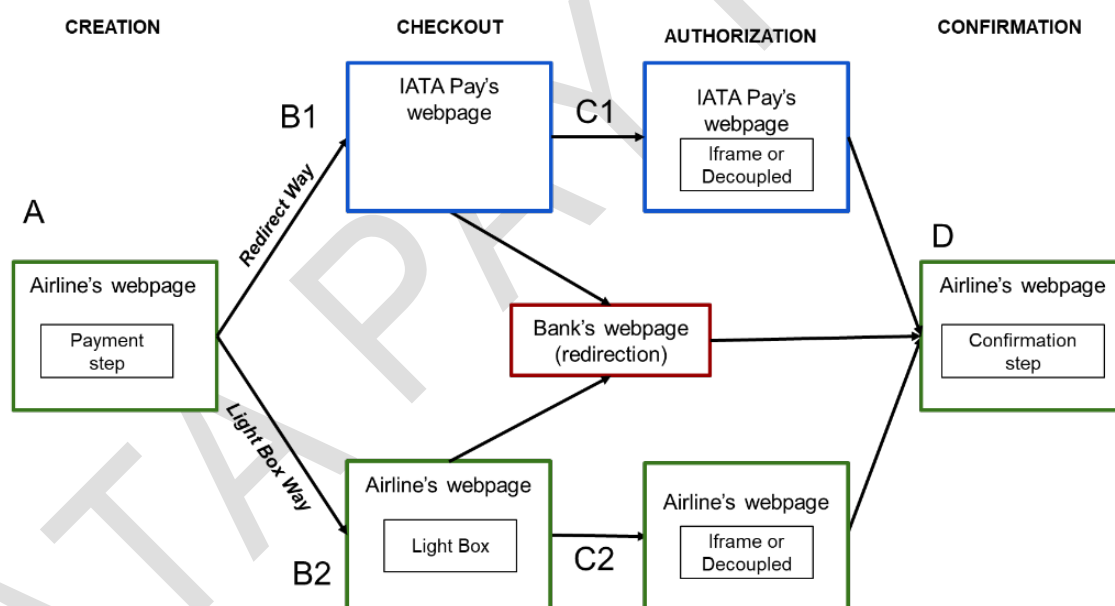
- **Redirect way:** the user is redirected to IATA Pay's webpage where is asked for the appropriate information, depending on the regulation the user is in, using a form. When all the information is gathered, the user is redirected to proceed with the authorization phase.
- **Light Box way:** the payer's information is gathered in a lightbox inside the airline's payment webpage. When all the information is gathered, the user is redirected to proceed with the authorization phase. This form is not developed by the airline, but dynamically generated by an IATA Pay's JavaScript loaded on the website allowing you to customize the styles.

After the payer's info is collected, the payer will be shown the procedure that their bank needs to finalize the payment. He or she authorises the payment and is redirected to the airline's webpage where can see the confirmation of his or her purchase. Depending on the country, sometimes he or she will authorize the payment with an iframe or a decoupled method. It is invisible to the user in any checkout integration way.

Authorization methods:

- **EUR – Austria, Belgium, Estonia Finland, France, Germany, Ireland, Italy, Latvia, Lithuania, Luxembourg, Netherlands, Portugal, Spain (AT, BE, EE, FI, FR, DE, IE, IT, LV, LT, LU, NL, PT, ES)** iframe with a TPP's authorization webpage. This will execute an "[Instant Payment](#)" using a "TPP powered by [Deutsche Bank](#)" and connecting with the bank's "[Berlin Group PSD2 spec](#)".
- **EUR – Netherlands (iDEAL)** iframe with a TPP's authorization webpage. This will execute an "[Instant Payment](#)" using a "TPP powered by [Deutsche Bank](#)" and connecting with the bank's "[iDEAL group](#)".
- **GBP - United Kingdom (GB):** redirection to the user's bank authorization webpage. This will execute "[Faster Payment](#)" using a "TPP powered by [Open Banking](#)".
- **INR – India (IN):** iframe with a TPP's authorization webpage. This will execute an "[UPI \(Unified Payment Interface\)](#)" using a "TPP powered by [Standard Chartered Bank](#)".
- **HKD – Hong Kong SAR, China (HK):** iframe with a TPP's authorization webpage. This will execute a "[FPS \(Faster Payment System\)](#)" using a "TPP powered by [DBS](#)".
- **SGD – Singapore (SG):** iframe with a TPP's authorization webpage. This will execute "[FPS \(Faster Payment System\)](#)" using a "TPP powered by [DBS](#)".
- **THB – Thailand (TH):** iframe with a TPP's authorization webpage. This will execute a "[PromptPay \(Bangkok Bank PromptPay\)](#)" using a "TPP powered by [CITI Bank](#)".
- **BRL – Brazil (BR):** iframe with a TPP's authorization webpage. This will execute a "[Santander PIX](#)" using a "TPP powered by [Banco Santander](#)".
- **MXN – Mexico (MX):** iframe with a TPP's authorization webpage. This will execute a "[CoDi Santander](#)" using a "TPP powered by [Banco Santander](#)".
- **GHS – Ghana (GH):** iframe with a TPP's authorization webpage. This will execute a "[GhanaQR](#)" using a "TPP powered by [Standard Chartered Bank](#)".

- **VND – Vietnam (VN):** iframe with a TPP's authorization webpage. This will execute a "[VietQr](#)" using a "TPP powered by [Standard Chartered Bank](#)"
- **MYR – Malaysia (MY):** iframe with a TPP's authorization webpage. This will execute a [DuitnowQr](#) using a "TPP powered by [Standard Chartered Bank](#)"
- **PHP – Philippines (PH):** iframe with a TPP's authorization webpage. This will execute a [QrPh](#) using a "TPP powered by [Standard Chartered Bank](#)"
- **JOD – Jordan (JO):** iframe with a TPP's authorization webpage. This will execute a [Cliq](#) using a "TPP powered by [JK Bank](#)"
- **AUD – Australia (AU):** iframe with a TPP's authorization webpage. This will execute a [PayTo](#) payment using a TPP powered by [ANZ bank](#) (Australia and New Zealand Banking Group Limited)
- **COP – Colombia (CO):** Redirect to TPP's authorization webpage. This will execute a payment in [PSE \(ACH Coolombia\)](#) using a "TPP powered by [BBVA bank](#)."



Webpages visited by the user according to the alternative selected by the airline to integrate IATA Pay.

A) Mockup of *Airline's* webpage - *Payment* step:

LAMBDAairlines

LocationAboutContact

1. Select flight

2. Passenger Data

3. Payment information

4. Purchase confirmation

Payment methods

Lambda Pay

Credit or Debit Cards

Paypal

Price breakdown

Price

TOTAL1,01 EUR

Flight details

DEPARTURE ✕Wed 2nd Jan 2019
MAD 13:10 → SXF 17:30

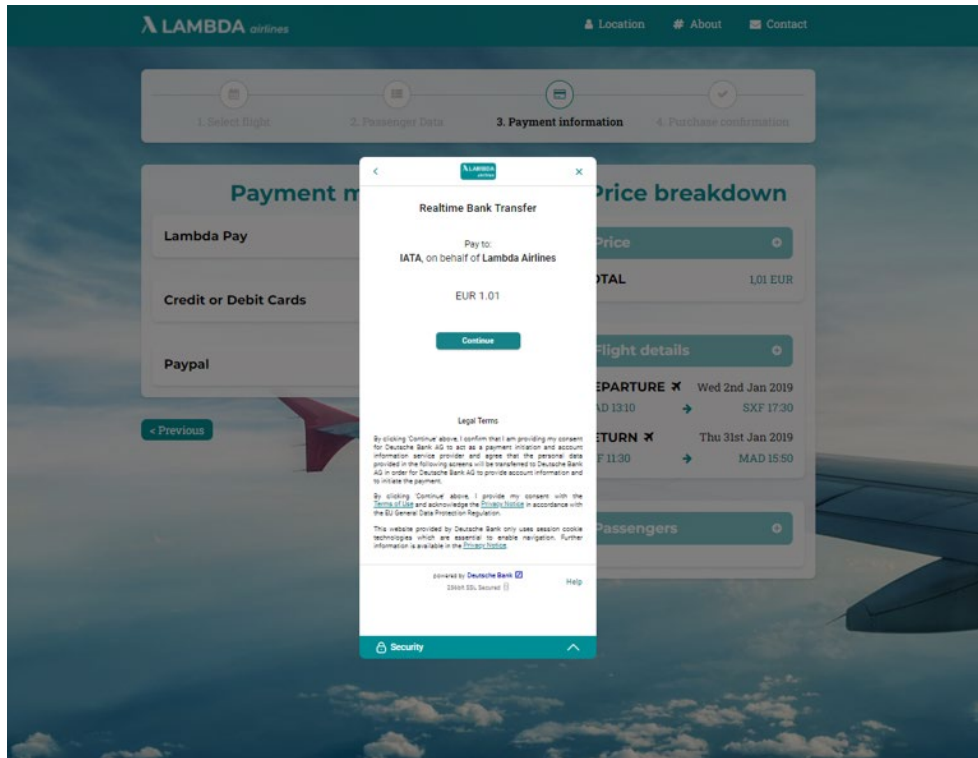
RETURN ✕Thu 31st Jan 2019
SXF 11:30 → MAD 15:50

Passengers

B1) Mockup of *IATA Pay's* webpage - Redirect way:

[illegible]

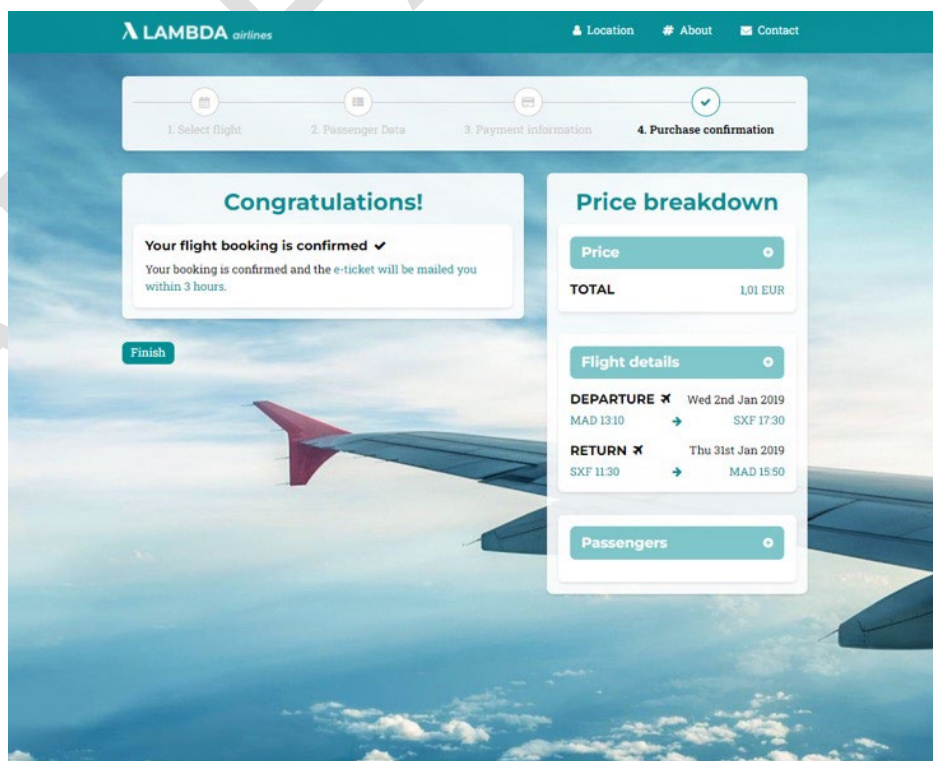
B2) Mockup of Airline's webpage - Lightbox way:



C) To execute the payment authorization depending on the country the user can be:

- User authorizes the payment into an iframe or with a decoupled authorization method ([C1/C2 cases in diagram](#)).

D) Mockup of Airline's webpage - Confirmation step:

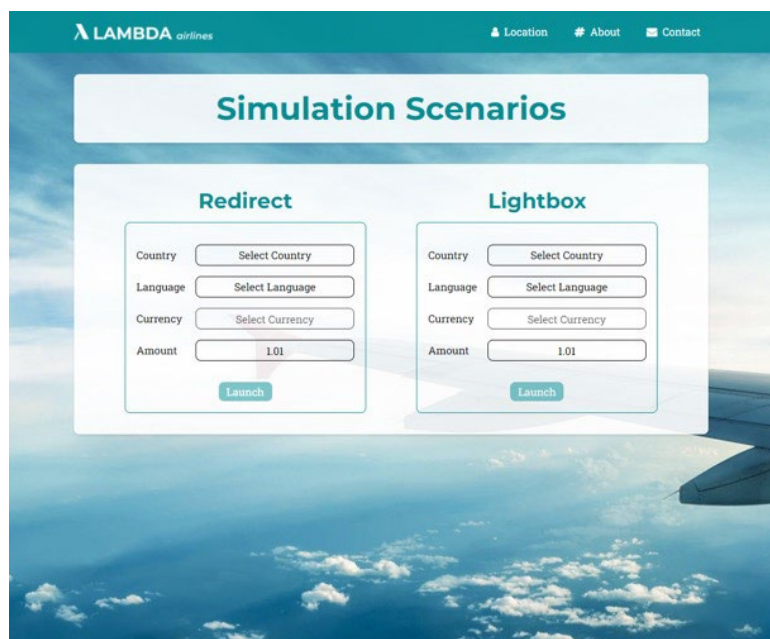


Example/Simulation

IATA Pay platform has got an emulated airline called “Lambda Airlines”. It is possible to try out the different integration methods in <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and get an idea of the UX.

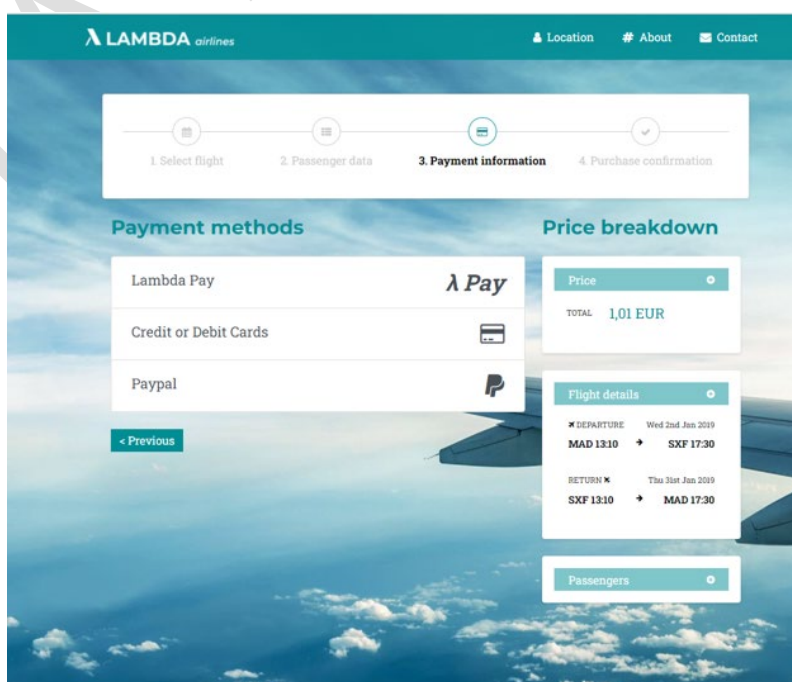
This simulation includes:

1. Front page to select the scenario to be simulated, “Redirect Way” or “Light Box way”, and the country.

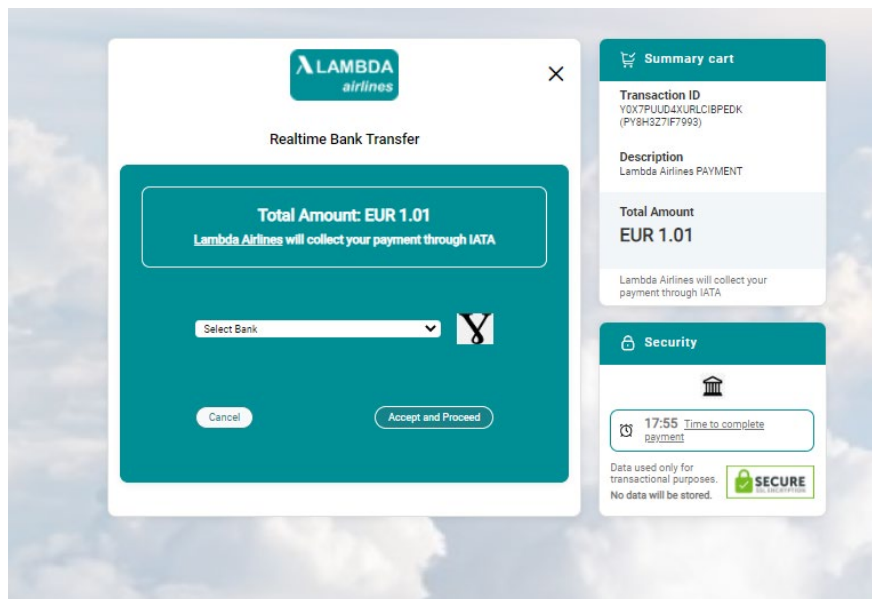


2. “Lambda Airlines” payment page, independently from the one selected the main page will be the same, with two checkout methods:

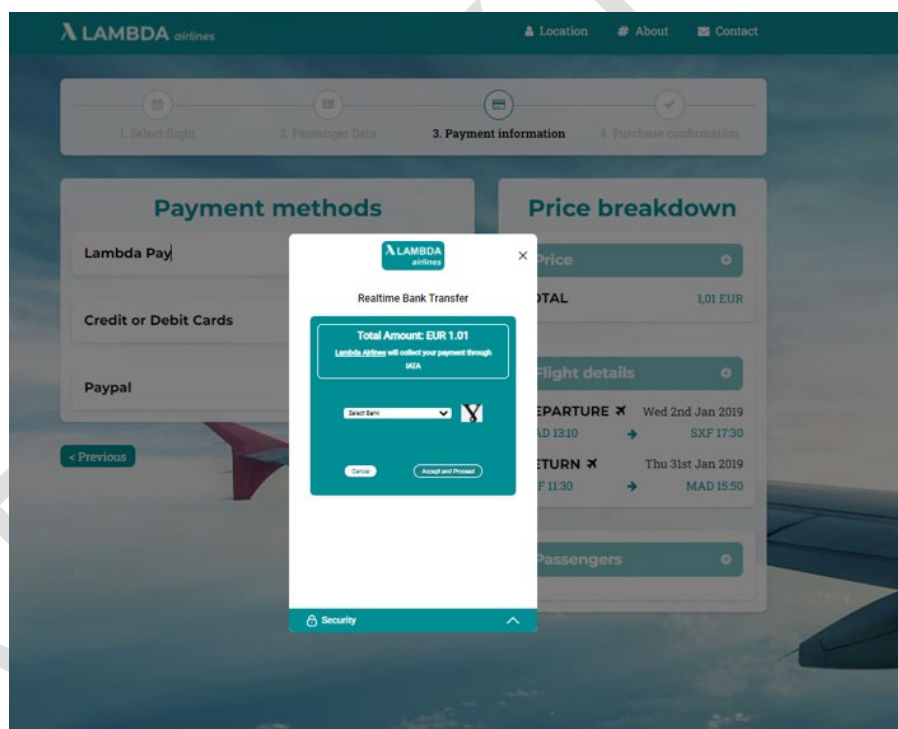
Select “Gamma Bank” to use the simulated Bank for testing purposes or Deutsche Bank using **DE43000000005686751168** to test IBAN flow)



- a. “Redirect way” going through “IATA Pay” checkout page.



- b. “Light Box” with an embedded checkout light box.

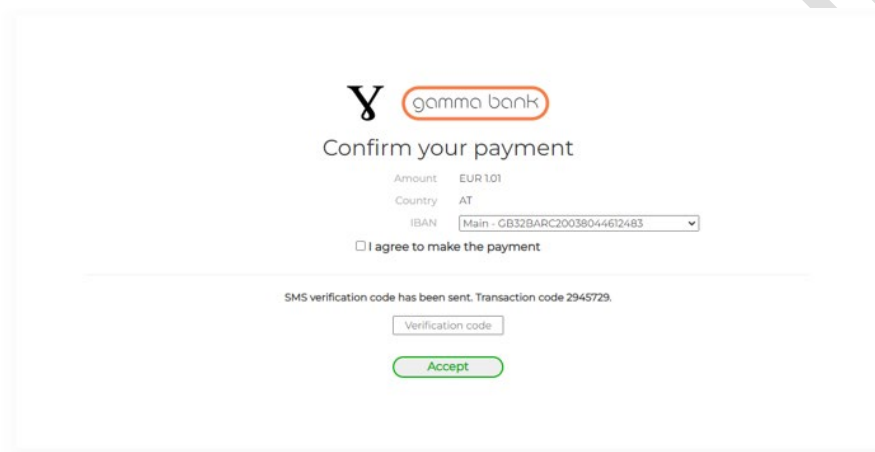


3. “Mock TPP” that shows a transition page to “Gamma Bank”.
4. “Gamma Bank” payment authorization page. This has two steps:
 - a. Login: any username or password is accepted.



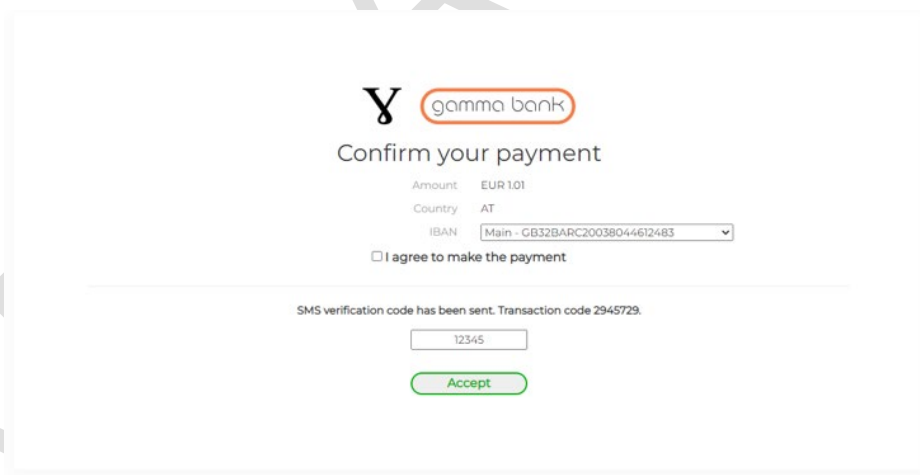
The login form for Gamma Bank features the bank's logo at the top. Below the logo are two input fields: 'Username' and 'Password'. A blue 'Sign in' button is positioned at the bottom of the form.

- b. Confirmation page: select or confirm your account and payment.



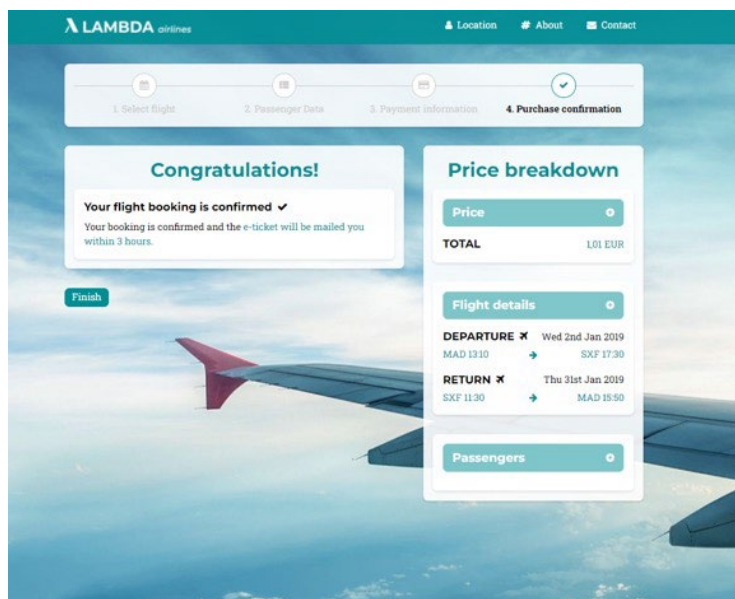
The confirmation page displays the Gamma Bank logo and the heading 'Confirm your payment'. It shows transaction details: Amount (EUR 1.01), Country (AT), and IBAN (Main - GB32BARC20038044612483). A checkbox for 'I agree to make the payment' is present. Below this, a message states 'SMS verification code has been sent. Transaction code 2945729.' followed by a 'Verification code' input field and an 'Accept' button.

On login, If you write "error" in the email input, the confirmation page will allow you to emulate an error.



This version of the confirmation page is identical to the previous one, but the 'Verification code' input field contains the text '12345'.

5. “Lambda Airlines” success/failure confirmation page.



5.4.1 Redirect Way

This is the basic and default behaviour.

How to integrate:

1. [Prerequisite] The IATA Pay’s webpage may be customised to look like the airline webpage, so the airline must give its logo and its brand main colour.
2. The airline sends a POST to /api/v1/payments to initiate a new payment in the platform. The response contains a field called “checkoutMethods.redirect” which holds a property called “redirectUrl”. The value of this property is the URL to the IATA Pay’s webpage where the user must be redirected.
3. The airline redirects the user to IATA Pay’s webpage (the “redirectUrl”) where the payer is asked for the required bank information.
4. The payer fills the required information and is redirected to the TTP/bank’s page.

In section “[API Specification / Basic flows testing](#)” there is an example about how to test this checkout method with your airline client credentials.

Pros:

- The airline doesn’t have to develop its own form to gather the payer’s bank information.
- The airline doesn’t have to modify the form in case of a future change of the needed payer’s data.

Cons:

- It is IATA Pay webpage who asks the payer for his or her info instead of the airline’s webpage, what may imply that the payer doesn’t feel comfortable because thinks that it might be a scam.

Examples with “Lambda Airlines”:

- Country GB and GBP: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select United Kingdom in country combo box.
- Country AT, BE, DE, EE, ES, FI, FR, IE, IT, LU, LV, LT, NL, PT and EUR: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select one of the Euro countries or Netherlands (SEPA) in the country combo box.
- Country NL and EUR <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Netherlands (IDEAL).
- Country GB and GBP: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select United Kingdom in the country combo box.
- Country HK and HKD: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Hong Kong SAR China in country combo box.
- Country SG and SGD: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Singapore countries in the country combo box.
- Country IN and INR: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select India in country combo box.
- Country TH and THB: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Thailand countries in the country combo box.
- Country BR and: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Brazil in the country combo box.
- Country MX and: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Mexico in the country combo box.
- Country GH and: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Ghana in the country combo box.
- Country VN and <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Vietnam in the country combo box.
- Country MY and: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Malaysia in the country combo box.
- Country PH and: <https://lambda.sandbox.iata-pay.iata.org/web/#/dashboard> and select Philippines in the country combo box.

5.4.2 Light Box Way

Prerequisites:

The airline must include a `<script>` tag in the head of its payment webpage. For example, in sandbox environment tag to include will be:

```
<script type="text/javascript" id="iatapay-lightbox" src="https://sandbox.iata-pay.iata.org/iatapay-lightbox/iatapay-lightbox.js">
```

How to integrate:

1. The airline sends a POST to `/api/v1/payments` to initiate a new payment in the platform.
2. When the payment creation response is received the airline must call the script function **checkout** (included in `iatapay-lightbox` script). The airline must pass the response as parameter to the function. Here we can see an example of invoking checkout function:

```
function xxxxxxxxxx() {
    Object response = /* code to create a payment */
    ...
    checkout(response); /* Show the lightbox */
    ...
}
```

3. The Airline page which calls 'checkout' must implement a JavaScript function with the name as 'iatapayNotify'. This function will be called to pass control back to airline page. 'iatapayNotify' will receive an object with the name 'paymentstatus' as parameter. This object will contain the following data:

Key	M/O/C	Type & Length	Description
status	M	X(35)	Current payment status
iataPaymentId	M	X(13)	Unique identifier generated by IATA Pay for this payment
merchantPaymentId	M	X(100)	Unique identifier assigned by the airline for this payment.
dateTime	M	X(14)	Datetime stamp of status notification Format: DDMMYYYYHH(24)MMSS (UTC)

Example of fail notification:

```
{ status: "FAILED", iataPaymentId:"PAK33CD7BB9A8", merchantPaymentId:"Reference1234",
dateTime:"20122019125932" }
```

Example of success notification:

```
{ status: "AUTHORIZED", iataPaymentId:"PAK33CD7BB9A8", merchantPaymentId:"Reference1234",
dateTime:"20122019125932" }
```

Note that: In the lightbox integration, **Success URL** and **Failure URL** are also mandatory to be provided by Airline. This is needed in case user's bank requires that the customer is redirected to the banks page, upon return, user will be needed to be addressed back to Airline success or failure page.

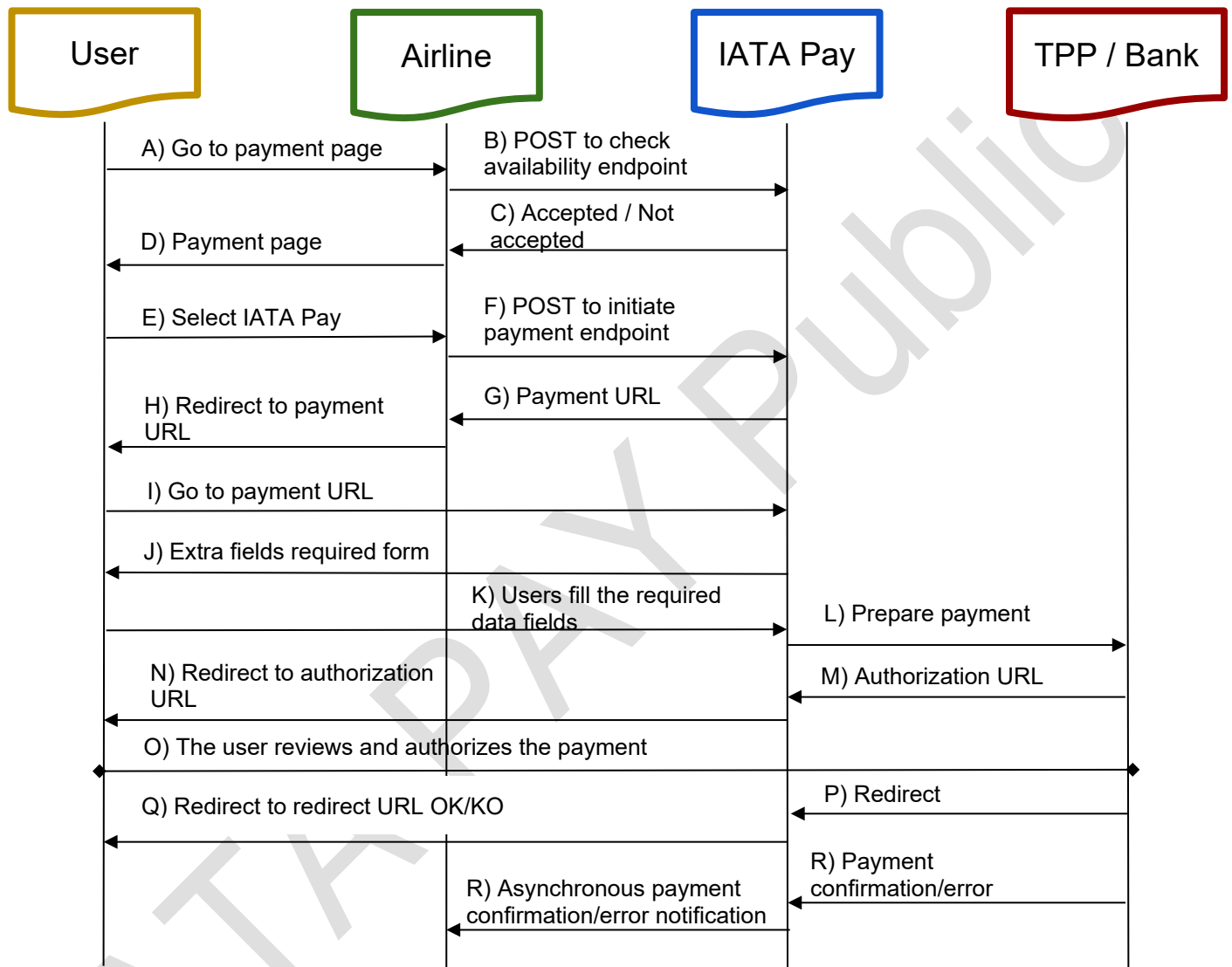
Pros:

- The payer is asked for his or her bank information within the airline's webpage, so the payment step is not hosted in a foreign webpage so that the user feels comfortable.

- The airline doesn't have to modify the light box in case of a future change of the needed payer's data, since the fields of the light box are given by the loaded JavaScript and, therefore, dynamically generated.

5.5 Payment Workflow example

Apart from how the payer's bank information is collected, the workflow is the same for every alternative. It is described with the **redirect way**:



- A. The user goes to the payment page of the airline website.
- B. The airline checks if IATA Pay payment method is available for the country/currency where the user is purchasing.

Request POST /api/v1/payments/allow

This request can be cached if it doesn't contain any user-specific fields.

- C. IATA Pay responds with "Accepted" or "Not accepted" response code. If the response is "Accepted", this flow continues; else, the user cannot pay with IATA Pay.

- D. The airline displays the payment page with IATA Pay payment method available.
- E. The user selects IATA Pay as the payment method in the check-out page in the airline website.
- F. The airline initiates the payment in IATA Pay.

Request POST /api/v1/payments

The request must include the final user redirect URLs (OK and KO), the amount, the country and the currency.

The POST request may include a field called “merchantPaymentId”, which could be used by the airline to identify the sale of the ticket/service in the administration dashboard or reporting system.

- G. In *redirect way*, the response contains the URL to redirect the user to finish the payment transaction. In “*Light Box way*”, a JavaScript is called to show pop up panel to finish the payment transaction.
- H. In *redirect way*, the user is directly redirected to the payment URL since IATA Pay will act as the acquirer to ask for the needed data to initiate the payment. In “*Lightbox Way*”, the payment details are gathered in the airline’s webpage.
- I. The user is redirected to IATA Pay’s webpage. If extra data wasn’t needed, the user would be redirected to the TPP/Bank page directly.
- J. The user fills the required data fields.
- K. IATA Pay will perform some interactions with the TPP and the payer’s bank before the payment instruction can be delivered to the payer’s bank:
 - 1. Decide the regulated TPP to be used – Request Routing.
 - 2. Prepare the authorization process depending on the TPP selected.

After these steps, IATA Pay will redirect the payment request to the chosen TPP. The TPP will have to perform, at least, the following actions in the payment request:

- Acquire the authorization from the user to issue the instruction in the payer’s bank if it has not already been obtained.
 - Send the payment request to the payer’s bank.
- L. The payer’s bank receives the payment initiation request and returns the required redirect URL to IATA Pay.
- M. IATA Pay redirects the user to the authorization page.
- N. The payer’s bank will check and will initiate the payment request authorization process. This interaction may happen under different authentication models. (E.g., PSD2 define three alternatives – Embedded, Decoupled, Redirected).

The payer authorizes the payment request following the schema that their bank proposes them.

- O. The user is redirected to the redirect URL OK/KO.
- P. IATA Pay will redirect the user to the airline’s website, either to “*redirectUrls.successUrl*” or to “*redirectUrls.failureUrl*”, depending on the success/failure.



IATA Pay receives a payment authorization. IATA Pay will forward the **AUTHORIZED** notification to the airline's website with a standard format. When the airline receives the AUTHORIZED notification, the paid services can be provided (for example the ticket can be issued).

If an error occurred after or before the AUTHORIZATION, then IATA Pay will notify it to the airline's website with other notification.

IATA Pay will notify as well when receiving the funds with **SETTLED** status. In addition, if the payment is suspected of not meeting any legal criteria, IATA Pay will notify as well when receiving the funds with **TOBEINVESTIGATED** status.

IATA Pay will notify as well when sending the funds to the airline with **CLEARED** status.

Both notifications are **asynchronous**: after or before the user redirection. See more info in the section ["Status Notification"](#).

5.6 API Summary

The purpose of this subsection is showing an overview of the info that IATA Pay needs to create payments. You should keep in mind that not every API parameter is described here – that is the goal of the [API specification](#).

Note: * means that the parameter is mandatory.

5.6.1 Request - Data from the Airline to IATA Pay

The following table describes the data to be gathered in the airline's webpage in order to initialize a payment request in the IATA Pay platform. The examples that are in the table are illustrative.

Name	Description
<i>merchantId</i> *	Unique identifier of the merchant in IATA Pay.
<i>merchantPaymentId</i>	Unique identifier assigned by the merchant for this payment.
<i>amount</i> *	Total amount of the payment. Decimals are separated with a point, and this is the only symbol.
<i>country</i> *	Code of the country/market. Compliant with ISO 3166 with two characters.
<i>locale</i> *	Language to be used on user messages. Compliant with IETF language tag and RFC 5646.
<i>currency</i> *	Currency of the payment. Compliant with ISO 4217 with three characters.
<i>departureDate</i> *	The departure date is the day of the first flight in the purchase cart. Its format is "YYYY-MM-DD" (e.g., 2022-12-23).
<i>successUrl</i> *	Merchant's URL to redirect the user when the payment finishes successfully.
<i>failureUrl</i> *	Merchant's URL to redirect the user when the payment finishes with error.
<i>notificationUrl</i> *	Merchant's callback URL to notify the payment's changes (when the status changes to authorized, settled, failed or cleared). It must be HTTPS. This field must content between 10 and 250 characters
<i>Authorization Timeout (per payment)</i>	<p>Time range (min) allowed to authorize the payment. It should be an integer number between 1 and 60 minutes. E.g., if merchant wants the user to have 10 minutes to complete the payment it should be provided in payment request with that value: 'Authorization Timeout: 10'.</p> <p>This field is optional, so, if it is not provided, it will work as follows:</p> <ol style="list-style-type: none"> 1. Firstly, if the merchant has completed the time within the merchant settings in the advanced options screen (Merchant Authorization Timeout), it will use that time. 2. Secondly, if the above is not completed by the merchant, the default timeout of 19 minutes shall be used for this case.
<i>bankTransferDescription</i>	<p>Info to be placed in the description of the bank transfer. By default the description in their bank statement will be "AirlineName PAYMENT AirlinePaymentId" (i.e. "LAMBDA AIRLINES PAYMENT L3CDS20190130").</p> <p>Also Airline could have the possibility to customize the information they want to</p>

	show to the customer in the "bankTransferDescription" field. Any non-alphanumeric character could be deleted by the bank. Potentially, other banking scheme references/info may be added along the process (Customer's Bank). IATA works on customising as much as possible those on IATA's hands.																		
loyalUser	Info about the loyal user logged in the merchant. For example, frequent flyer.																		
PNR	Passenger Name Record (up to 200 alpha - numeric characters, including underscore, dot and dash). If the merchant is interested in including it.																		
retriableUntil	Maximum date-time to retry the payment when it fails. No retries allowed when this field is empty. With ISO date-time format and compliant with UTC time zone.																		
tokenId	It contains information regarding the payment method selected by the payor (these data will not be saved in the system for confidentiality reasons). (e.g., "+3470000000" can be used in mobile phone for test purposes in the integration with bizum), (e.g., a set of fictitious VPA ID can be used for test purposes in the integration with VPA: -Scenario VPA ID is invalid: "invalidVPA@iata" -Valid VPA ID but timeout, so fail scenario: "failedtest@iata" -Valid VPA ID resulting in Success: "successtest@iata" (e.g a set of fictitious values that can be used for test purposes in the integration with JK Bank via RTP: - Mobile: "00962791212121" - Alias: "FUADCBJ" - Alias: "TEST09")																		
preferredCheckoutMethod	<p>In the case of India, if VPA is marked, it overrides any other even if it was not the first to be marked. For all other cases, the first one ticked from the list in the merchant configuration is launched.</p> <p>If the value arrives empty, VPA is taken if it is ticked by default. If it is not, the first one ticked from the list in the merchant configuration.</p> <p>If they send a preferred payment method that the merchant does not have configured, it fails in the initiation of the payment.</p> <p>If the value received is not one of the available ones, it will fail at payment creation.</p> <table><thead><tr><th>COUNTRY</th><th>VALUE</th><th>NOTES</th></tr></thead><tbody><tr><td>IN</td><td>Netbanking</td><td>India Netbanking</td></tr><tr><td>IN</td><td>QR</td><td>UPI India QR</td></tr><tr><td>IN</td><td>VPA</td><td>UPI India VPA</td></tr><tr><td>JO</td><td>QR</td><td>CliQ Jordan QR</td></tr><tr><td>JO</td><td>RTP</td><td>CliQ Jordan RTP</td></tr></tbody></table>	COUNTRY	VALUE	NOTES	IN	Netbanking	India Netbanking	IN	QR	UPI India QR	IN	VPA	UPI India VPA	JO	QR	CliQ Jordan QR	JO	RTP	CliQ Jordan RTP
COUNTRY	VALUE	NOTES																	
IN	Netbanking	India Netbanking																	
IN	QR	UPI India QR																	
IN	VPA	UPI India VPA																	
JO	QR	CliQ Jordan QR																	
JO	RTP	CliQ Jordan RTP																	
tokentype	For payments made in Jordan with JK Bank RTP option, it is required to complete the tokentype choosing one of the following options: "mobile" or "alias".																		

Payment Method	Payer Input	Format
Pay To	Pay ID	Maximum of 256 characters in lower case.

		Regex patter: <code>^[!-@\[\~][-@\[\~]{0,254}![!-@\[\~]\$</code>
	BSB	6 character number Regex patter: <code>^\d{6}\$</code>
	Account	Number of 5 to 9 characters. Regex patter: <code>^\d{5,9}\$</code>
VPA	VPA	Email with a minimun of 5 and maximun 99 characters. Regex patter : <code>^[a-zA-Z0-9.-]{2,49}@[a-zA-Z]{2,49}\$</code>
CLIQ	ALIAS	Please find below the Alias format and regex patter: <ul style="list-style-type: none"> • Alphanumeric • Minimum of 3 characters • Maximum of 10 characters • Contains at least 1 letter Regex: <code>^(?=.*[A-Za-z])[A-Za-z0-9]{3,10}\$</code>
ANZ	BSB	<code>^(0[1-9] [1-9]\d)\d{4}\$</code>
CLIQ	Mobile	00962798014928 (14 digits)
BIZUM	Mobile number	9 character number Regex patter: <code>^\d{9}\$</code>

Furthermore, airlines have the option to include details of the several items of a purchase by using this information:

<i>item – description</i>	Description of the item (e.g., <i>MAD-LGW 2019-12-31 12:30</i>)
<i>item – sku</i>	SKU of the item. Pattern: {1,100} (e.g., <i>X0123456789</i>).
<i>item – price</i>	Price of the item (e.g., <i>4567.89</i>). Decimals are separated with a point, and this is the only symbol.
<i>item – currency</i>	Currency of the item. Pattern: {1,100} (e.g., EUR) Compliant with ISO 4217 with three characters.
<i>item - quantity</i>	Quantity of the item (e.g., 1). Decimals are separated with a point, and this is the only symbol.

5.6.2 Response - Data from IATA Pay to the Airline

This is some of the most relevant data that IATA Pay returns to the airline after the payment has been initiated:

Name	Description
<i>iataPaymentId</i>	Unique identifier generated by IATA Pay.
<i>authorizationDateTime</i>	Date-time of the authorization of the payment. It won't be set until the payment is authorized.
<i>creationDateTime</i>	Date-time of the creation of the payment in the platform.
<i>failureCode</i>	Code of the error if the payment's status is FAILED.
<i>failureDetails</i>	Transaction error details when the status is FAILED. This field gives more information on why the payment failed
<i>status</i>	Payment's status.

Note: many of the data values passed from the airline to IATA Pay are also returned in the response.

5.7 Dummy values for testing

For the creation of a new payment to test in Sandbox environment the following values can be used depending on the market and the corresponding payment method selected.

BIZUM (Spain)

- Phone number: +34700000000

<5 € Payment failure

>5€ Payment success

VPA (India)

-Scenario VPA ID is invalid: "invalidVPA@iata"

-Valid VPA ID but timeout, so fail scenario: "failedtest@iata"

-Valid VPA ID resulting in Success: "successtest@iata"

CliQ (Jordan)

- Mobile: "00962791212121"

- Alias: "FUADCBJ"

- Alias: "TEST09"

PayTo Australia

Identifier	Value	Operation Result	Description
<i>BSB</i>	<i>any</i>	<i>OK</i>	<i>Operation successfully completed</i>

Identifier	Value	Operation Result	Description
PayId	any	OK	Successful operation. If random strings are entered, strings containing capital letters will not be valid. E.g.: Jaime - invalid; jaime - valid.
PayId	declined	KO	Failed operation (USER_CANCEL). The user denies the authorisation of the mandate.
PayId	expired	KO	Failed operation (INVALID_TOKEN_ID). Simulates erroneous token id in the creation of a mandate. Forces a retry of the payment, returning the user to the ANZ form to enter the token.
PayId	account_not_found	KO	Biller receives Mandate action outcome to advise mandate received unsuccessfully. Creditor Account not found
PayId	account_closed	KO	Biller receives Mandate action outcome to advise mandate received unsuccessfully. Creditor Account closed
PayId	account_non_npp	KO	Biller receives Mandate action outcome to advise mandate received unsuccessfully. Creditor account is not eligible for NPP

PSE Colombia

It is not necessary to use specific values, just fill in all fields with data.

5.8 Finish Notification

The user is redirected to the OK URL or the KO URL, depending on the success or failure of the payment operation. However, **redirecting a user to the OK URL doesn't imply that the success of the payment is confirmed**. It means only that the user has completed the payment operation correctly.

When the payment is *authorized* by the TPP/bank, IATA Pay will notify **asynchronously** to the airline. IATA Pay will also notify to the airline when the payment is *settled* or *cleared* using the same method.

This notification will be sent to the endpoint specified by the parameter *notificationUrl* set when the payment was created by the airline. This field is mandatory.

The content of this request will be the same as the response of the request to get data of a payment (*GET /api/v1/payments/{iataPaymentId}*).

The airline will have to check the field *status*, whose value can be AUTHORIZED SETTLED, CLEARED or FAILED.

A notification with status **FAILED** will be sent when the payment fails as the settled notification. In this case the request body contains the fields "failureCode" and "failureDetails" filled.



Therefore, the airline will be able confirm to end user the purchase of the paid services if the status value is AUTHORIZED, SETTLED (immediate status when payment journey ends) or CLEARED (this status is available after IATA processes the payment of the funds to the airline).

The **http response status** for this webhook notification should be **OK (200)** to confirm the reception of this notification. Otherwise, the notification will be sent again as explained below.

This notification webhook request contains a signature header in order to verify the origin of this. More details about the technical spec of this and other webhooks in the section "[Notification Specification](#)".

5.9 Languages

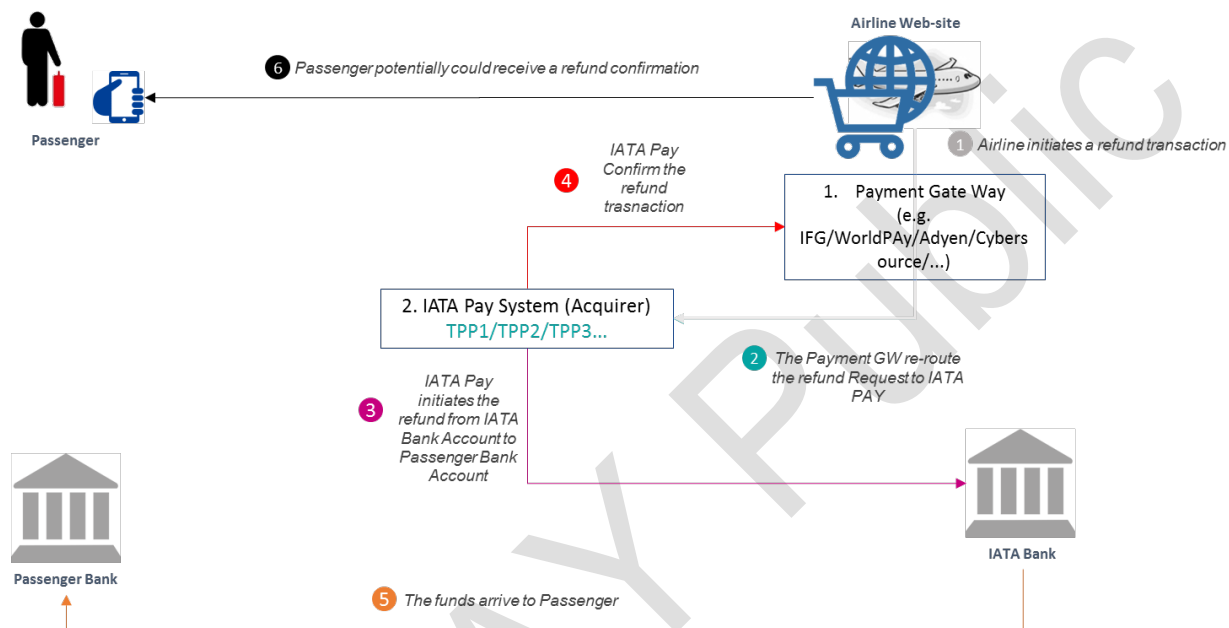
IATA Pay offers the possibility to choose the language of the interfaces that will be shown to the final users using the “locale” field inside payment creation API endpoint. This field contains the language and the country, compliant with IETF language tag and RFC 5646. For example, “en-GB” is the code for British English. If the country code of a language is not available, the system will fall back to the same language obviating the country code. If a selected language is not available, the system will show English as the default language. Currently the supported languages are:

- Arabic (ar)
- Chinese Traditional (zh-HK)
- Chinese Simplified (zh-CN)
- Dutch (nl)
- English (en)
- Estonian (et)
- Finnish (fi)
- French (fr)
- German (de)
- Italian (it)
- Indonesian (id)
- Korean (ko)
- Latvian (lv)
- Lithuanian (lt)
- Malay (ms)
- Philippine (tl)
- Portuguese (pt)
- Spanish (es)
- Thai (th)
- Turkish (tr)
- Russian (ru)
- Vietnamese (vi)

6 Refund Transactions

6.1 Actors

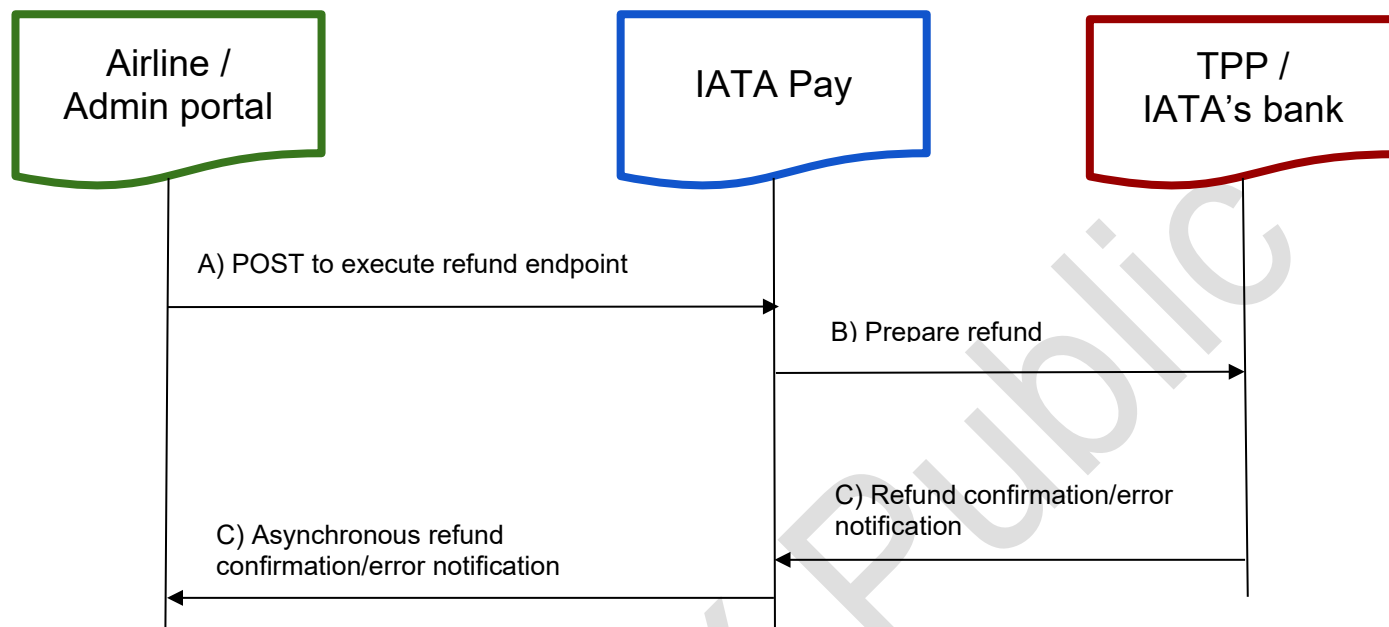
The following diagram illustrates the role of actors identified in the refund transaction.



6.2 Prerequisites / Rules

- The payer's bank account must be reachable by IATA Pay. IATA Pay is able to route the refund payment request to a regulated TPP in the territory where the licence is valid.
- The origin bank account in the refund payment instruction will belong to IATA.
- Only previous payment transactions processed using IATA Pay Platform can be refunded through IATA Pay.
- The original payment transaction should be in SETTLED, CLEARED or AUTHORIZED status.
- The amount to be refunded (including all the refund requests processed) in the refund request must be equal or lower than the original payment.

6.3 Refund Workflow



Every step of this workflow is executed synchronously.

- A. The refund process can be originated by an airline directly using the API or by a user in the IATA Pay administration portal. In any case, the request is sent to IATA Pay through one of these endpoints:

Request POST /api/v1/payments/{iataPaymentId}/refund

Request POST /api/v1/merchants/{merchantId}/payments/{merchantPaymentId}/refund

The former is used to create refunds of payments given their *iataPaymentIds*, whereas the latter is used when their *merchantPaymentIds* are given.

There can be several refund requests for an original payment. The sum of the refund amounts linked to an original payment must be lower or equal to the total amount of the original payment. Therefore, the amount of a refund mustn't be greater than the remaining amount that hasn't been refunded yet.

If the amount is not provided, then IATA Pay will take the remaining amount that hasn't been refunded yet.

- B. IATA Pay asks the proper regulated TPP to perform the refund request. The chosen TPP will be in charge of asking IATA's bank to refund the given amount to the payer's account – the account of the person who paid for the ticket or the airline's services.
- C. IATA Pay receives a refund confirmation. After making an internal checking:
- IATA's bank will settle the payment.
 - Payer's bank will receive the refund.
 - IATA Pay will notify the confirmation/error to the airline's website. This notification is **asynchronous**.



See more info in the section ["Status Notification"](#).

Users of the admin portal will be able to see the states of the refunds.

6.4 Refund Batches

It is possible to execute several refunds in batch, instead of doing it one by one. The previous workflow remains, and the only difference is the set of endpoints:

Request POST /api/v1/refunds/batches

Request POST /api/v1/merchant/{merchantId}/refunds/batches

There are two endpoints because of the same reasons there are two endpoints to create refunds individually: the former is used to create refunds of payments given the *iataPaymentId* of each payment, whereas the latter is used given the *merchantPaymentId* of each payment.

Both endpoints require that every payment belongs to the same merchant.

6.5 API Summary

6.5.1 Request - Data from the Airline to IATA Pay

This is the data that the merchant must/may send to IATA Pay to execute a refund, either individually or in batch:

Name	Description
<i>iataPaymentId</i> (1)	Unique identifier generated by IATA Pay for the original payment to be refunded.
<i>merchantPaymentId</i> (1)	Unique identifier assigned by the merchant for this payment.
<i>merchantId</i> (2)	Unique identifier of the merchant in IATA Pay.
<i>merchantRefundId</i>	Identifier assigned by the merchant for this refund.
<i>amount</i>	<p>Amount of the refund with the same currency of the payment.</p> <p>This value mustn't be greater than the remaining amount that hasn't been refunded yet. E.g., the payment was 800 so the first refund mustn't be greater than 800; if the first refund is 200, then the next refund cannot be greater than 600, and so on.</p> <p>If no value is specified, then the remaining amount that hasn't been refunded yet will be refunded.</p>
<i>bankTransferDescription</i>	<p>Info to be placed in the description of the bank transfer. By default, the description in their bank statement will be "RefundId <Refund bankTransferDescription> REFUND OF PAYMENT PaymentId" (i.e., "R3CDS20190130 Lambda Airlines REFUND OF PAYMENT P388S2019ABCD"). Else, the airline does not complete this field, it will be populated with "RefundId < AirlinePublicName > ..."</p> <p>Also, Airline could have the possibility to customize the information they want to show to the customer in the "bankTransferDescription" field. Any non-alphanumeric</p>

	character could be deleted by the bank. Potentially, other banking scheme references/info may be added along the process (Customer's Bank). IATA works on customising as much as possible those on IATA's hands.
--	---

(1) *iataPaymentId* is required by the endpoints that allow creating refunds of payments given their *iataPaymentIds*, whereas *merchantPaymentId* is required by the endpoints that allow creating refunds of payments given their *merchantPaymentIds*.

(2) *merchantId* is required to create refunds in batch but, to create refunds individually, it is only required by the endpoint that allows creating refunds of payments given their *merchantPaymentIds*.

The requests of the endpoints used to create refunds in batch may also contain the following value:

Name	Description
<i>merchantRefundBatchId</i>	Unique identifier assigned by the merchant for this batch.

6.5.2 Response - Data from IATA Pay to the Airline

IATA Pay responds with the data of the request, with all the information of the payment related to this refund and with the following values:

Name	Description
<i>iataRefundId</i>	Unique identifier generated by IATA Pay.
<i>creationDateTime</i>	Date-time of the creation of the refund in the platform.

The response of the endpoints used to create refunds in batch also contains the following value:

Name	Description
<i>iataRefundBatchId</i>	Unique identifier generated by IATA Pay.

6.6 Finish Notification

In the same way that IATA Pay notifies a new payment status, it will notify when a refund is **AUTHORIZED**, **SETTLED**, **CLEARED** or **FAILED**.

This notification will be sent to the endpoint specified by the parameter *notificationUrl* set when the refund was initiated by the airline. This field is mandatory.

IATA Pay will notify **asynchronously** to the airline in using the *notificationUrl*. The content of this request will be the same as the response of the request to get data of a refund (*GET /api/v1/refunds/{iataRefundId}*).

The airline will have to check the field *status*, whose value can be **AUTHORIZED**, **SETTLED**, **CLEARED** or **FAILED**. A notification with status **FAILED** will be sent when the refund fails. In this case the request body contains the fields "failureCode" and "failureDetails" filled.

Therefore, **the airline will be able to confirm the refund if the status value is AUTHORIZED, SETTLED or FAILED.**

The **http response status** for this webhook notification should be **OK (200)** to confirm the reception of this notification.

In the same way that notifications of finished payments, this notification webhook request contains a signature header in order to verify the origin of this. More details about the technical spec of this and other webhooks in the section "[Notification Specification](#)".

7 Searches

7.1 Payments

When a payment is created, airlines can provide an ID called “merchantPaymentId” in the request, a value used by the airline to identify this payment. This ID is optional and IATA Pay will always assign another ID which is returned in the response, called “iataPaymentId”. Therefore, airlines may have two IDs to identify a payment.

Once it was said, it makes sense that a specific payment can be found in two ways:

- Using “iataPaymentId”:

Request GET /api/v1/payments/{iataPaymentId}

- Using “merchantPaymentId” and “merchantId”, an ID assigned to each merchant when is registered in IATA Pay platform:

Request GET /api/v1/merchants/{merchantId}/payments/{merchantPaymentId}

The response for both endpoints is the same as the one returned when a payment is created.

We can also get all payments:

Request GET /api/v1/payments

This endpoint provides several filter options, such as pagination or date times. Its response is a JSON array of payments, each one with the same structure as the one returned when a payment is created.

7.2 Refunds

The IDs set when a refund is created are similar to those given to payments: airlines can provide an ID called “merchantRefundId” and IATA Pay will always assign another ID called “iataRefundId”. Therefore, a specific refund can be found in two ways:

- Using “iataRefundId”:

Request GET /api/v1/refunds/{iataRefundId}

- Using “merchantRefundId” and “merchantId”:

Request GET /api/v1/merchants/{merchantId}/refunds/{merchantRefundId}

The response of both endpoints is the same as the one returned when a payment is created.

IATA Pay doesn't provide any endpoint to GET all refunds.

7.3 Refund Batches

The creation of refund batches share behaviour with the creation of payments or refunds: airlines can provide an ID called “merchantRefundBatchId” and IATA Pay will always assign another ID called “iataRefundBatchId”. Therefore, a specific refund batch can be found in two ways:

- Using “iataRefundBatchId”:

Request GET /api/v1/refunds/batches/{iataRefundBatchId}

- Using “merchantRefundBatchId” and “merchantId”:

Request GET /api/v1/merchants/{merchantId}/refunds/batches/{merchantRefundBatchId}

The response of both endpoints is a collection of refunds, each one with the same structure as the one returned when a refund is created, either individually or in batch.

7.4 Finished Transactions

The following endpoints allow to consult any type of finished transactions (you can filter by “UNEXPECTED SETTLED”, “SETTLED”, “CLEARED” or “FAILED” status), either payments or refunds:

- As JSON:

Request GET /api/v1/transactions/finished/

- As CSV:

Request GET /api/v1/transactions/finished/csv

7.4.1 API Summary

7.4.1.1 Request - Data from the airline to IATA Pay

There are some filters that must/may be used to get the finished transactions:

Name	Description
<i>finishDateTimeFrom</i> *	Date time to limit the first finished transaction. Format required: YYYY-MM-DDThh:mm:ss. (e.g., 2018-08-04T13:32:54.123456)
<i>finishDateTimeTo</i> *	Date time to limit the last finished transaction. Format required: YYYY-MM-DDThh:mm:ss. (e.g., 2018-08-04T13:32:54.123456)
<i>merchantIds</i>	Collection of unique identifiers of the merchants in IATA Pay.
<i>selectedStatuses</i>	Available options to filter by status.

7.4.1.2 Response - Data from IATA Pay to the airline

The response is a collection of transactions. Some of the values of each one are the following:

Name	Description
<i>iataTransactionId</i>	Unique identifier generated by IATA Pay. It matches the <i>iataPaymentId</i> or the <i>iataRefundId</i> .
<i>merchantId</i>	Unique identifier of the merchant in IATA Pay.
<i>merchantTransactionId</i>	ID assigned by the merchant (airline) for each transaction (Payment or Refund). <i>merchantPaymentId</i> : It is optional and unique <i>merchantRefundId</i> : it is optional and not unique It could be empty if the airline does not provide it at the payment or refund creation.
<i>creationDateTime</i>	Date and Time of the creation of the transaction.
<i>finishDateTime</i>	Date and Time of the completion of the transaction.
<i>amount</i>	Total amount of the transaction. Decimals are separated with a point and this is the only symbol.
<i>currency</i>	Currency of the payment. Compliant with ISO 4217 with three characters.
<i>status</i>	Current transaction status.
<i>failureCode</i>	Transaction error code if the status is FAILED.
<i>failureDetails</i>	Transaction error details when the status is FAILED. This field gives more information on why the payment failed
<i>type</i>	Transaction type.
<i>pnr</i>	Passenger Name Record (up to 200 alpha-numeric characters, including underscore, dot and dash).

8 API Specification

8.1 Environments

There are two environments for the airlines integration: sandbox and production.

Below is specified the “Endpoints base URL” in order to execute any API call, and the “Public output IPs” in order to allow the asynchronous notification webhooks from this IPs (with a whitelist).

- **Sandbox.** Environment for testing.
 - Endpoints base url: <https://sandbox.iata-pay.iata.org/>
 - Public Output IPs: 63.3.251.33, 145.46.137.70, 27.52.212.31, 192
- **Production.** Environment for real payments.
 - Endpoints base url: <https://iata-pay.iata.org/>
 - Public Output IPs: 34.251.243.231, 63.32.18.13

8.2 Authentication

The main endpoints of the API are secured with OAuth 2.0 authentication. According to the roles defined by OAuth 2.0 (see <https://tools.ietf.org/html/rfc6749#section-1.1>), the airline is the *client*.

When an airline is registered as a client of IATA Pay, it receives an ID (*client_id*) and a secret key (*client_secret*):

- *client_id*. The authorization server issues the registered client a client identifier, which is a unique string representing the registration information provided by the client. The client identifier is not secret, since it is exposed to the resource owner and is not used alone for client authentication. The client identifier is unique to the authorization server.
- *client_secret*. It is a secret known only by the application and the authorization server.

An airline may be registered in more than one environment, but its credentials will be different in each environment.

IATA Pay exposes the following endpoint for authentication: */oauth/token*.

OAuth2 defines several *grant types* for different use cases (see <https://tools.ietf.org/html/rfc6749#section-4>). *Client credentials* is the type that airlines need to interact with the API.

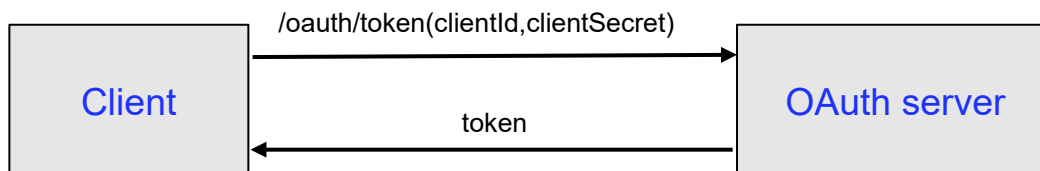
8.2.1 Client Credentials

This grant type is used when applications request an access token to access their own resources, not on users' behalf.

8.2.1.1 Tokens

An access token is a credential that can be used by an application to access an API. They inform the API that the bearer of the token has been authorized to access the API and perform actions specified by the scope that has been granted.

Access tokens are requested to the auth server using the client credentials:



IATA Pay uses tokens in JWT (JSON Web Token) format and doesn't store them.

8.2.1.2 Steps to get an access token

If an airline needs to get an access token, it should make a POST `/api/v1/oauth/token` request to the authentication endpoint, whose header and body must be the following:

- Header. Parameters:
 - "Authorization". Its value is "Basic `base64Encode(client_id + ':' + client_secret)`", that is, the string "Basic " followed by the result of Base64 encoding a string composed of the `client_id`, a colon and the `client_secret`. For example:

Key	Value
Authorization	Basic aG9sYWZzZGFzZHNhYXNkYXNkYXM=

- Body. Its format is `application/x-www-form-urlencoded` and its parameters the following ones:
 - "grant_type". Its value must be "client_credentials".
 - "scope". Its value must be "payment".

Key	Value
grant_type	client_credentials
scope	payment

The authentication server validates the credentials and returns an error if they are invalid. Else, it returns a success response that contains:

- access_token**. It contains the access token in JWT (JSON Web Token) format.
- token_type**. It tells the type of the access token. Nowadays, the auth server always return "bearer" tokens.
- expires_in**. The access token will expire in the number of seconds given in this attribute. If the access token expires, the client will have to follow these steps to get a new access token.
- scope**. It lists the scopes of information that the token contains. It may be different to the requested scopes.



- *jti*. It provides a unique identifier for the JWT.

This whole communication must be hosted to host for security reasons, since the client credentials mustn't be exposed. In fact, as we already said, the access tokens are not stored by IATA Pay and they expire soon.

Depending on the environment, the endpoint to make the POST call to get the token is:

Sandbox: <https://sandbox.iata-pay.iata.org/api/v1/oauth/token>

Production: <https://iata-pay.iata.org/api/v1/oauth/token>

8.2.1.3 Using an access token

All API requests have to contain the “*access_token*” in the header. The header parameter has an “Authorization” key and the value is “Bearer *access_token*”, that is, the string “Bearer” followed by the value of the field “*access_token*” from the response of the token login request (token is in JWT format).

For example:

Key	Value
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c

8.3 Protocol

The protocol for the communications is HTTPS. **Hypertext Transfer Protocol Secure (HTTPS)** is an extension of the Hypertext Transfer Protocol (HTTP) for secure communication over a computer network, and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS), or, formerly, its predecessor, Secure Sockets Layer (SSL). The protocol is therefore also often referred to as HTTP over TLS, or HTTP over SSL.

The security of HTTPS is that of the underlying TLS, which typically uses long-term public and private keys to generate a short-term session key, which is then used to encrypt the data flow between client and server. X.509 certificates are used to authenticate the server (and sometimes the client as well). As a consequence, certificate authorities and public key certificates are necessary to verify the relation between the certificate and its owner, as well as to generate, sign, and administer the validity of certificates. While this can be more beneficial than verifying the identities via a web of trust, the 2013 mass surveillance disclosures drew attention to certificate authorities as a potential weak point allowing man-in-the-middle attacks. An important property in this context is forward secrecy, which ensures that encrypted communications recorded in the past cannot be retrieved and decrypted should long-term secret keys or passwords be compromised in the future. Not all web servers provide forward secrecy.

The API has been developed with REST. **Representational State Transfer (REST)** is a software architectural style that defines a set of constraints to be used for creating web services. Web services that conform to the REST architectural style, termed RESTful web services, provide interoperability between computer systems on the Internet. RESTful web services allow the requesting systems to access and manipulate textual representations of web resources by using a uniform and predefined set of stateless operations. Other kinds of web services, such as SOAP web services, expose their own arbitrary sets of operations.

8.4 Content Type

The main content type for all API request is JSON. **JSON (JavaScript Object Notation)** is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

8.5 Swagger Specification

The API's format is OpenAPI Specification (formerly Swagger Specification). You can find more information in <https://swagger.io/docs/specification/about/>

The tool *Swagger Codegen* can be used to generate clients of our API in different languages: <https://swagger.io/tools/swagger-codegen/>

The API can be explored and tested with a friendly UI (Swagger UI) available in <https://sandbox.iata-pay.iata.org/swagger-ui.html>

8.6 Notification Specification

The platform will always send an asynchronous notification for all transactions to notify finished payments or refunds.

The **http response status** for any webhook notification should be **OK (200)** to confirm the reception of this notification. If there is the case of receiving a notification from IATA Pay, either for a payment or a refund, and the system receiving it is not being able to get it due to whatever reason, that system will have to use the corresponding GET endpoint to get the information about the payment or refund where the notification was missed, as those endpoints are always available for them to query the information, solving the problem of a bad notification. **IATA Pay will not retry the sending of a notification.** Reasons why below.

All the notification webhook requests contain a **signature header** in order to verify the origin of this.

8.6.1 Notification signature spec

This body is signed with a unique and secret key (*notification key*, which is provided to each airline when it is registered in the IATA Pay platform) only known by the airline and IATA Pay, so we achieve two goals when the airline receives a signed request in its endpoint notificationUrl:

- The airline can confirm that the request has been sent by IATA Pay – if it hadn't, the signature wouldn't be correct because the attacker wouldn't have the secret key and, therefore, wouldn't know how to sign the requests.
- The airline can confirm that the body hasn't been altered in transit – if it had, the signature wouldn't be correct because it was generated with the original body and, if the body had changed, the signature would be different (different bodies produce different signatures).

The body is signed using HS256 (HMAC with SHA-256, see [RFC 7518 - JSON Web Algorithms \(JWA\) - HMAC with SHA-2 Functions](#)) and its result (the generated MAC) is encoded in Base64 and placed in the header "Authorization", after a prefix "IATA Pay-HMAC-SHA256 ". Example: if HS256 generates an output "my_hash" and the Base64 coding of this hash generates "my_encoded_hash", then the value of the header "Authorization" would be "IATA Pay-HMAC-SHA256 my_encoded_hash".

The airline need to check if the requests to notificationUrl are correctly signed, they should do it to avoid attacks. So, how do the airlines check that the request signature is correct? They must do the same as IATA Pay does: sign the body. They must take the body as is, sign it and compare the result with the signature that is placed in the header "Authorization": its value without the prefix "IATA Pay-HMAC-SHA256 ". If they are equal, then the request should be trusted. Else, the request must be rejected.

How do we generate the signature of the body?

1. We receive the body as a String.
2. We use HMAC with SHA-256 to create a MAC, a hash in this case, from this String using the *notification key* as the MAC key.
3. We encode the result (the hash) in Base64.

MAC, HMAC (see [RFC 2104 - HMAC: Keyed-Hashing for Message Authentication](#)), SHA-2 functions like SHA-256 (see [RFC 6234 - US Secure Hash Algorithms](#)) and Base64 (see [RFC 4648 - The Base16, Base32, and Base64 Data Encodings](#)) are all standards used by many web applications, so there are many libraries in different languages that provide utils about them.

For instance, we can use Apache Commons and the JDK to generate the signature of the body with the *notification key*:

```
import org.apache.commons.codec.binary.Base64;
import org.apache.commons.codec.digest.HmacUtils;
import javax.crypto.Mac;
import static java.nio.charset.StandardCharsets.UTF_8;
import static org.apache.commons.codec.digest.HmacAlgorithms.HMAC_SHA_256;

public static String getHmacSha256InBase64(String stringToEncode, String key) {
    Mac sha256Hmac = HmacUtils.getInitializedMac(HMAC_SHA_256, key.getBytes(UTF_8));
    byte[] hash = sha256Hmac.doFinal(stringToEncode.getBytes(UTF_8));
    return Base64.encodeBase64String(hash);
}
```

8.6.2 Notification retries

Transaction status notification it is sent only once. **IATA Pay will not retry the sending of a notification.**

We have evaluated the risks and rewards of implementing a retry system in case an Airline can't get a notification and discarded for the following concerns:

- If the retry is fixed to a particular time, airline system could be down when the retry occurs, so the only way to manage notifications to solve the system down problem would be to receive an acknowledgement to stop the retries and ensure the notification was received.
- An acknowledgement endpoint would have to be implemented so that the airline would have to call it for each notification they receive so that they can confirm they have received it for IATA Pay to stop the retry process.
- Retrying notifications would cause more particular case scenarios where new problems would arise. For example, the call to the acknowledge endpoint could be lost or not sent. In this case, IATA Pay would continue to send notifications of the same transaction that the airline captures. The airline could avoid that by implementing a system check to check they don't store a repeated notification of a transaction, but it would be too complex to do it real time as would require database checks and other possible checks.
- If by any case the airline stores a repeated notification as a new transaction it would be impossible to do the settlement and reconciliation process and it would break the whole system, this example alone is the reason to not retry notifications, as it is a risk too big to allow to happen.
- Another example case is that the acknowledgement is late enough so that the IATA Pay system sends a retry of a transaction they were already notified so the airline systems take it as another transaction.
- Even with the ones listed, there are more disadvantages to do the retry system both involving IATA Pay and the airline doing the integration, so there will be no retry of notifications. The airline can call when they need the GET endpoint to get the appropriate information as it contains the same information as the notification.

To avoid problems, the integration should be capable of receiving more than one notification of the same transaction.

8.7 Basic flows testing

Hereby there is a fully detailed and in-depth example of how to go through all IATA Pay flow to show how the integration has to be done.

8.7.1 Client login


The common step between all ways of integration is creating a payment, these are the steps:

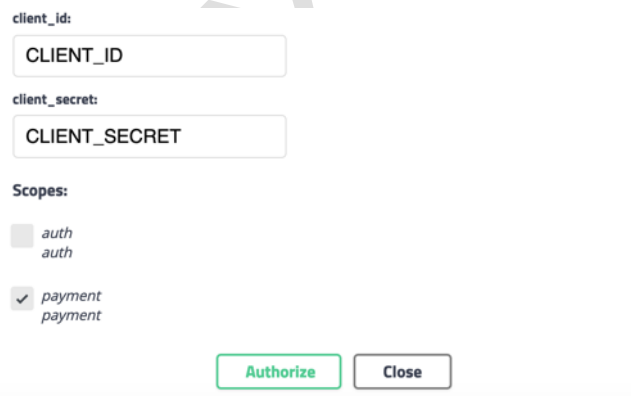
1. IATA Pay Admin user will create the client credentials and get your merchant's id from the IATA Pay Admin Portal.

Detailed info about how to create the credentials in section "3.5.5. Credentials" inside: <https://sandbox.iata-pay.iata.org/docs/admin-portal-guide>

2. Enter in the sandbox environment Swagger spec website:

<https://sandbox.iata-pay.iata.org/swagger-ui.html#/>

3. Log in with your user credentials that we have previously provided. Click on  , and in the screen, substitute CLIENT_ID and CLIENT_SECRET with yours, select "payment" scope as in the image and click authorize:



client_id:
CLIENT_ID

client_secret:
CLIENT_SECRET

Scopes:

☐ auth
auth

☒ payment
payment

Authorize Close

8.7.2 Payment initiation

You must be logged in as explained in [Client login](#) to initiate a payment

1. Create a payment: Open payments section and open "**POST /api/v1/payments**", click on "Try it out"

Checkout method - API
Manage the API checkout.

Payments
Manage payments.

GET
/api/v1/merchants/{merchantId}/payments/{merchantPaymentId}
Get the info of a payment.

GET
/api/v1/payments
Get all payments with their info.

POST
/api/v1/payments/
Initiate a new payment in the platform.

Parameters

Try it out

Name	Description
paymentRequestDto * required (body)	paymentRequestDto Example Value Model

The screen has to be like this:

POST
/api/v1/payments/
Initiate a new payment in the platform.

Parameters

No parameters

Request body required

application/json

```

{
  "merchantId": "XX00000000",
  "merchantPaymentId": "HR3DC2-20190321.125515.123456",
  "amount": 4567.89,
  "currency": "EUR",
  "country": "DE",
  "locale": "en-GB",
  "bankTransferDescription": "MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "loyalUser": {
    "userId": "miking",
  },
  "payerInfo": {
    "tokenId": "+34700000000",
  },
  "userId": "19.117.63.126",
  "pin": "HR3DC2",
  "authorizationTimeout": 18,
  "items": [
    {
      "description": "MAD LGW 2019-12-31 12:30"
    }
  ]
}

```

Execute

The selected parts in the image below are fundamental to avoid errors when executing the payment:

- Substitute (or delete if you don't want to use it) **merchantPaymentId** with your own id. This must be unique for each transaction.
- Change **merchantId** with yours (in the example the id is lambda), use the one we have previously provided.

POST
/api/v1/payments/
Initiate a new payment in the platform.

Parameters

No parameters

Request body required

application/json

```

{
  "merchantId": "XX00000000",
  "merchantPaymentId": "HR3DC2-20190321.125515.123456",
  "amount": 4567.89,
  "currency": "EUR",
  "country": "DE",
  "locale": "en-GB",
  "bankTransferDescription": "MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "loyalUser": {
    "userId": "miking",
  },
  "payerInfo": {
    "tokenId": "+34700000000",
  },
  "userId": "19.117.63.126",
  "pin": "HR3DC2",
  "authorizationTimeout": 18,
  "items": [
    {
      "description": "MAD LGW 2019-12-31 12:30"
    }
  ]
}

```

Execute

2. Before executing the payment, you may want to change:

- a. **notificationUrl**: where all notifications related to changes in the payment will be asynchronously received. We recommend using a tool like "<https://webhook.site>" to test the notification webhooks. This parameter is mandatory.
 - b. **redirectUrls**: URL where the user will be redirected when the payment is finished, either successfully (**successUrl**) or with a failure (**failureUrl**)
3. You could also change the country of the payment initiation from Germany to United Kingdom, just change fields with these values **country**: "**GB**" and **currency**: "**GBP**".

4. Execute the payment initiation clicking on

Execute

8.7.3 Payment created

Once the payment is created, you will get a response like this:

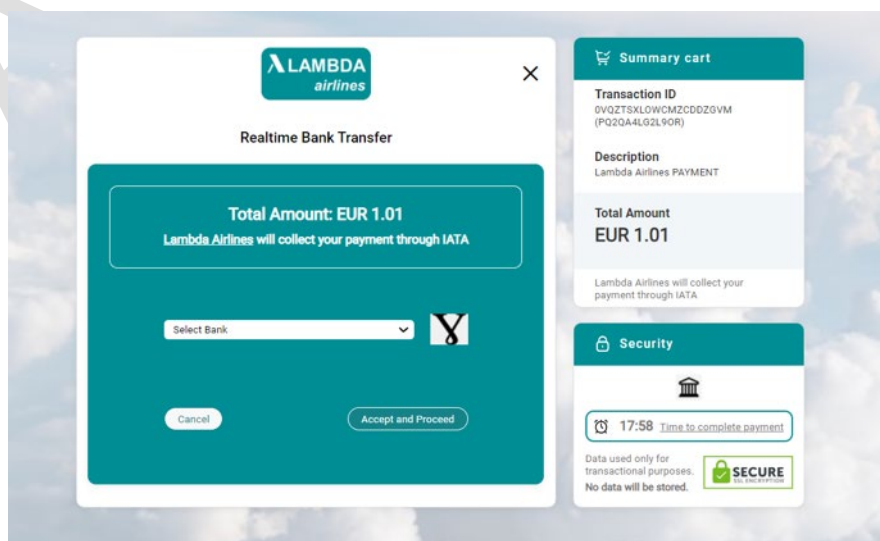
```
{
  "creationDateTime": "2023-03-31T12:35:05.027Z",
  "finishDateTime": "2023-03-31T12:35:05.027Z",
  "updateDateTime": "2023-03-31T12:35:05.027Z",
  "clearanceDateTime": "2023-03-31T12:35:05.027Z"
},
{
  "checkoutMethods": {
    "redirect": {
      "redirectUrl": "http://pay.iata.org/api/v1/payments/P23E5RS8TNT2S/checkout/redirect/initiate"
    }
  },
  "tpp": {
    "tppId": "MOCKTPP",
    "label": "Mock TPP",
    "tppLogoUrl": "string",
    "checkoutMethodType": "IFRAME",
    "steps": "2018-08-04T13:36:00.123456",
    "schemaLogoUrl": "string",
    "payerInfoRules": {

```

Depending on the chosen checkout method you will use for example redirect URL.

8.7.4 Checkout (with redirect)

Just copy the url inside redirectUrl field and redirect the user to it, the user will have to continue the flow.



If you want to use "Lightbox" as checkout method, you can view the specification of this in the "Checkout Methods" sections:

- [Light box Way](#)

8.7.5 Payment status notification

Because you have set a “notificationUrl” in the payment URL you will receive a webhook request when the payment is AUTHORIZED, SETTLED, CLEARED or FAILED.

An example of the notification content is:

```
[
  {
    "amount": 4567.89,
    "bankTransferDescription": "MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
    "checkoutMethods": {
      "form": {
        "jsUrl": "http://pay.iata.org/api/v1/payments/P23E5RS8TNT2S/checkout/form/script.js?divId=[divId]&buttonId=[buttonId]"
      },
      "redirect": {
        "paymentUrl": "http://pay.iata.org/api/v1/payments/P23E5RS8TNT2S/checkout/redirect/initiate"
      }
    },
    "country": "DE",
    "creationDateTime": "2018-08-04T13:32:54.123456",
    "currency": "EUR",
    "failureCode": "AC01",
    "finishDateTime": "2018-08-04T13:36:00.123456",
    "iataPaymentId": "P23E5RS8TNT2S",
    "locale": "en-GB",
    "merchantId": "XX00000000",
    "merchantPaymentId": "HR3DC2-20190321.125515.123456",
    "refundedAmount": 1567.89,
    "refunds": [
      {
        "amount": 1567.89,
        "bankTransferDescription": "Refund of MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
        "creationDateTime": "2018-08-04T13:32:54.123456",
        "currency": "EUR",
        "failureCode": "AC01",
        "finishDateTime": "2018-08-04T13:36:00.123456",
        "iataRefundId": "R53E5RS8TGT2V",
        "merchantRefundId": "HR3DC2-20190321.125515.123456",
        "status": "CREATED"
      }
    ],
    "status": "CREATED",
    "unrefundedAmount": 3000
  }
]
```

This body is signed with a unique and secret key. You can view the notification spec in the section [“Status Notification”](#).

This feature is available for refunds too.

8.7.6 Validation Endpoint

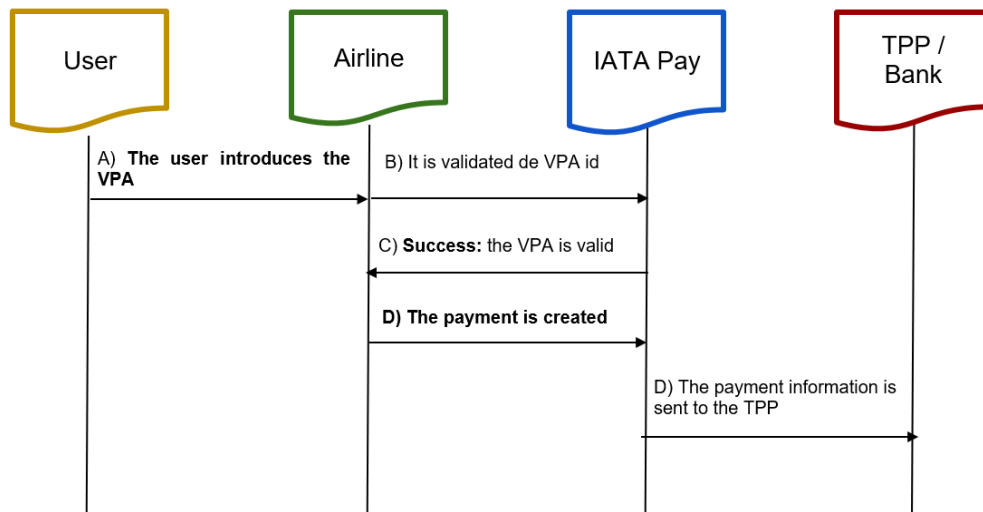
A) VPA Validation Endpoint

You must be logged in as explained in [Client login](#).

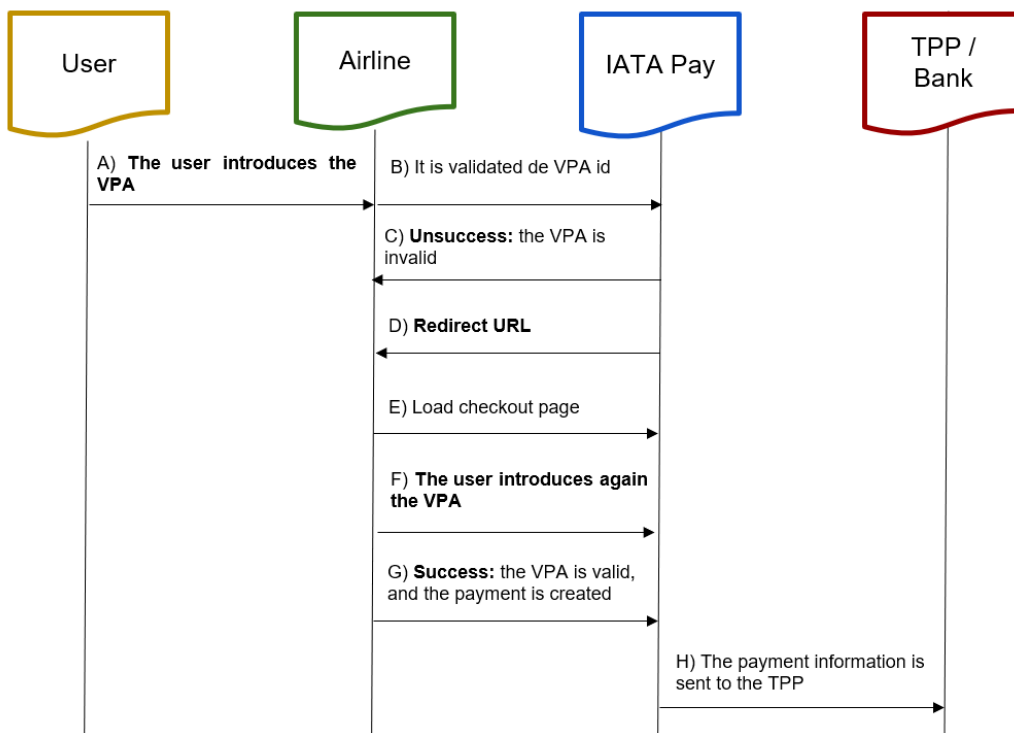
This is an opened endpoint where the airlines have the option to provide the VPA data previously introduced by the user. Then IATA Pay will check if the “tokenId” is valid. If the airline doesn’t provide the VPA or it is invalid, IATA takes it from the checkout page and will make its validation.

“GET/api/v1/token/validate”

In the following diagram you can see the process in case the VPA introduced by the user is valid:



However, if the VPA is invalid, the process will be as follow:



Payments Validate payment tokenId.

POST /api/v1/payments/allow Check whether a new payment can be initiated in the platform.

POST /api/v1/payments/ Initiate a new payment in the platform.

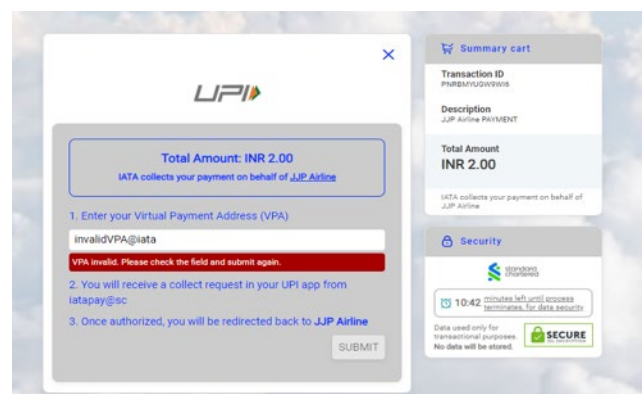
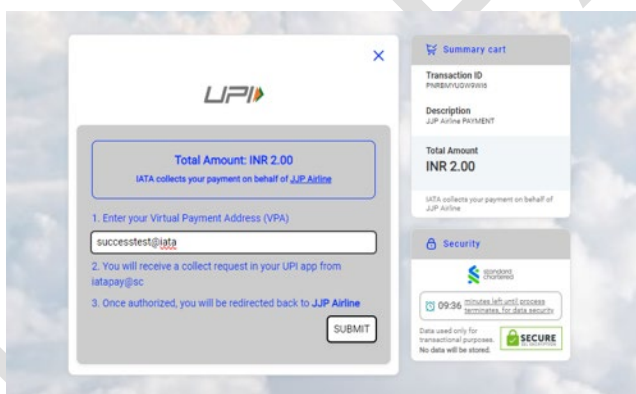
GET /api/v1/token/validate Validate tokenId.

Parameters Try it out

Name	Description
validationType * required string (query)	Token ID Validation Type Available values : VPA, CLIQ, BIZUM Example : VPA <input type="text" value="VPA"/>
tokenId * required string (query)	Payer's token id. Example : test@sc <input type="text" value="tokenId"/>
tokenType string (query)	Payer's token type Available values : MOBILE, ALIAS, PAY_ID, BSB Example : MOBILE or ALIAS <input type="text" value="--"/>

In order to fill out the required fields correctly, you must select the validation type from the available options and then introduce a valid “tokenId”

IATA will check this field even if the airline hasn’t provide the “tokenId”. If it is valid the process continues as expected, if it is not, a message appears alerting of this event. You can see below screenshots of both options if the VPA provided is valid or invalid.



If the VPA introduced by the user is invalid, the “submit” button will be disabled, in order to prevent any failure due to invalid VPA.

Once the validation has been carried out, two responses can be obtained, “success” in the case that the field is valid or “invalid” in the case that it is not, as you can see below:

Example Value | Schema

```

TPpTokenIdValidationResponseDto {
  status string
  Enum:
    [ SUCCESS, INVALID ]
}

```

It should be noted that even with successful validation of the VPA, there may be failures in the initiation of payment at the provider (TPP) unrelated to this validation.

B) BIZUM Validation Endpoint

This validation endpoint is also used to validate the BIZUM number provided via “tokenId”. With the objective of avoiding problems with invalid phone numbers, this endpoint checks the phone number that the user has entered on the airline's website prior to creating the payment in IATA PAY.

The following parameters must be completed to make the verification:

GET /api/v1/token/validate Validate tokenId.

Parameters

Name	Description
validationType * required string (query)	Token ID Validation Type BIZUM
tokenId * required string (query)	Payer's token id. 655059875
tokenType string (query)	Payer's token type MOBILE

Execute

The status field in the answer indicates whether it is correct or not.

Example of a correct answer:

Responses

Code	Description	Links
200	TokenId validated	No links

Media type: */*

Controls Accept header.


Example Value | Schema

```

{
  "status": "SUCCESS",
  "customerInfo": {
    "date": "2024-02-19T14:48:38.367Z",
    "account": "string",
    "name": "string"
  }
}

```

Example of invalid phone number:

Code	Details
200	<p>Response body</p> <pre>{ "status": "INVALID", "customerInfo": null }</pre> <div>  Download </div>

C) Cliq RTP Validation Endpoint

This validation endpoint is also used to validate the number or alias provided via “tokenId”. With the objective of avoiding problems with invalid data, this endpoint checks the phone number and the alias that the user has entered on the airline's website prior to creating the payment in IATA PAY. The following parameters must be completed to make the verification:

GET

/api/v1/token/validate Validate tokenId.

Parameters

Cancel

Name	Description
validationType * required string (query)	Token ID Validation Type <div>CLIQ</div>
tokenId * required string (query)	Payer's token id. <div>00962791212121</div>
tokenType string (query)	Payer's token type <div>MOBILE</div>

Execute

Clear

GET

/api/v1/token/validate Validate tokenId.

Parameters

Cancel

Name	Description
validationType * required string (query)	Token ID Validation Type <div>CLIQ</div>
tokenId * required string (query)	Payer's token id. <div>TEST09</div>
tokenType string (query)	Payer's token type <div>ALIAS</div>

Execute

Clear

The status field in the answer indicates whether it is correct or not.
Example of a correct answer:

Code	Details
200	<p>TokenId validated</p> <p>Media type</p> <div> <div>application/json</div> </div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "status": "SUCCESS", "customerInfo": { "date": "2024-04-18T15:15:20.054Z", "account": "string", "name": "string" } }</pre>

Example of invalid phone number or alias:

400

Bad request

No links

Media type

/

Example Value | Schema

```
{
  "status": 0,
  "error": "string",
  "message": "string",
  "details": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
}
```

8.7.7 Refund initiation

You must be logged in as explained in [Client login](#) to initiate a payment

1. Create a refund with iataPaymentId: Open Refunds - Individual section and open “**POST /api/v1/payments/{iataPaymentId}/refund**”, click on “Try it out”

POST /api/v1/payments/{iataPaymentId}/refund Initiate a new refund transaction for this payment.

Try it out

Parameters

Name	Description
iataPaymentId * required string (path)	Unique identifier generated by IATA Pay. Example : P23E5RS8TNT2S

P23E5RS8TNT2S

The screen has to be like this:

POST /api/v1/payments/{iataPaymentId}/refund Initiate a new refund transaction for this payment.

Cancel

Parameters

Name	Description
iataPaymentId * required string (path)	Unique identifier generated by IATA Pay.

P23E5RS8TNT2S

Request body required

application/json

```
{
  "merchantId": "XXXXXXXXXXXX",
  "merchantRefundId": "SK3DC2-20190321.125515.123456",
  "amount": 1567.89,
  "currency": "EUR",
  "bankTransferDescription": "Refund of MAD-IGH 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "notificationUrl": "https://www.lambda-airlines.com/api/refund/notification",
  "notificationEmails": [
    "SD_IATAPAY_ServiceDesk@kpmg.es"
  ]
}
```

The selected parts in the image below are fundamental to avoid errors when executing the refund:

- Substitute (or delete if you don't want to use it) **merchantRefundId** with your own id



- Change **merchantId** with yours (in the example the id is lambda), use the one we have previously provided.
- Set **currency** with the same value as it had in the payment, the refunds must be done in the same currency as the payment

```
{
  "merchantId": "XX0000000",
  "merchantRefundId": "SK3DC2-20190321.125515.123456",
  "amount": 1567.89,
  "currency": "EUR",
  "bankTransferDescription": "Refund of MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "notificationUrl": "https://www.lambda-airlines.com/api/refund/notification",
  "notificationEmails": [
    "SD_IATAPAY_ServiceDesk@kpmg.es"
  ]
}
```

2. Before executing the refund, you may want to change:

- amount:** the amount of the payment to be refunded, it can be any amount as long as it doesn't surpass the payment unrefunded amount. If you put the wrong amount the API call will return an error.
- notificationUrl:** where all notifications related to changes in the payment will be asynchronously received. We recommend using a tool like "<https://webhook.site>" to test the notification webhooks. This parameter is mandatory.

Execute

3. Execute the refund initiation clicking on

The same exact individual refund flow can be done with your own merchantPaymentId (if indicated in the payment initiation) calling `/api/v1/merchants/{merchantId}/payments/{merchantPaymentId}/refund` substituting the values in braces by the correspondent, the rest of the flow is the same.

8.7.8 Refund created

Once the payment is created, you will get a response like this:

```
{
  "iataRefundId": "R53E5RS8TG72V",
  "merchantRefundId": "SK3DC2-20190321.125515.123456",
  "amount": 1567.89,
  "currency": "EUR",
  "bankTransferDescription": "Refund of MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "status": "CREATED",
  "failureCode": "TIMEOUT_INITIATED",
  "failureDetails": "The Refund was rejected",
  "lockReason": "WAITING_APPROVAL",
  "creationDateTime": "2023-01-26T10:19:48.760Z",
  "finishDateTime": "2023-01-26T10:19:48.760Z",
  "updateDateTime": "2023-01-26T10:19:48.760Z",
  "clearanceDateTime": "2023-01-26T10:19:48.760Z"
}
```

8.7.9 Refund status finish notification

Because you have set a "notificationUrl" in the payment URL you will receive a webhook request when the refund is AUTHORIZED, SETTLED, CLEARED, BLOCKED or FAILED.

An example of this notification content body is:



```
{
  "iataRefundId": "R53E5RS8GT2V",
  "merchantRefundId": "SK3DC2-20190321.125515.123456",
  "amount": 1567.89,
  "currency": "EUR",
  "bankTransferDescription": "Refund of MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "status": "CREATED",
  "failureCode": "TIMEOUT_INITIATED",
  "failureDetails": "The refund was rejected",
  "lockReason": "WAITING_APPROVAL",
  "creationDateTime": "2023-01-26T10:19:48.760Z",
  "finishDateTime": "2023-01-26T10:19:48.760Z",
  "updateDateTime": "2023-01-26T10:19:48.760Z",
  "clearanceDateTime": "2023-01-26T10:19:48.760Z"
}
```

This body is signed with a unique and secret key. You can view all the notification spec in the section [“Notification Specification”](#).

8.7.10 Ticket Purchase enabled

The Ticket purchase endpoint helps to confirm and close the gap of the purchase activity. It allows to inform IATA of the result of the transaction purchase, i.e: acknowledgement that a successful collection is linked to a successful ticket issuance, or a failed collection corresponds to failed ticket issuance.

If you set “Create refund if no ticket purchase” in the IATA Pay Admin Portal, you will have the possibility to enable this feature, and its operation depending on whether the ticket is issued by the airline.

Once, you have login with your CLIENT_ID and CLIENT_SECRET credentials in swagger, you will find in the “IATA Pay Payment API definition” – “Ticket purchase” section the following option. For testing purpose will be needed a payment already created through the IATA Pay payment method:

Tickets purchase <small>Notify ticket purchase.</small>		^
POST	/api/v1/payments/{iataPaymentId}/purchase <small>Notify ticket purchase result related with the payment in airline platform.</small>	✓ 🔒
POST	/api/v1/merchants/{merchantId}/payments/{merchantPaymentId}/purchase <small>Notify ticket purchase result related with the merchant in airline platform.</small>	✓ 🔒

There are two options for implementing it:

1. With the first option, you can test through the definition of an already created **"iataPaymentId"**:
 - Open **“POST /api/v1/payments/{iataPaymentId}/purchase”**, and then click on “Try it out”
 - To do the test, substitute **"iataPaymentId"** with your own id. This must be unique for each transaction.

Tickets purchase Ticket purchase confirmation.

POST /api/v1/payments/{iataPaymentId}/purchase Notify ticket purchase result related with the payment in airline platform.

Parameters

Name	Description
iataPaymentId * required string (path)	Unique identifier generated by IATA Pay. <i>Example</i> : P23E5RS8TNT2S

Request body * required

application/json

Example Value | Schema

```
{
  "purchased": true
}
```

Responses

Code	Description	Links
200	OK	No links
400	Bad request	No links
401	You are not authorized to view the resource	No links
403	Accessing the resource you were trying to reach is forbidden	No links
404	Payment not found	No links
500	Internal server error	No links

2. With the second option, you can test through the definition of an already created **"merchantPaymentId"**:

- Open **"POST/api/v1/payments/{merchantId}/payments/{merchantPaymentId}/purchase"**, and then click on **"Try it out"**.
- To do the test, substitute **"merchantPaymentId"** with your own id. This must be unique for each transaction.
- To continue, change the **"merchantId"** with yours, use the one we have previously provided.

POST /api/v1/merchants/{merchantId}/payments/{merchantPaymentId}/purchase Notify ticket purchase result related with the merchant in airline platform.

Parameters

Name	Description
merchantId * required string (path)	Unique identifier of the merchant in IATA Pay. <i>Example</i> : XX0000000
merchantPaymentId * required string (path)	Unique identifier assigned by the merchant for this payment. <i>Example</i> : HR3DC2-20190321.125515.123456

Request body * required

application/json

Example Value | Schema

```
{
  "purchased": true
}
```

Responses		
Code	Description	Links
200	OK	No links
400	Bad request	No links
401	You are not authorized to view the resource	No links
403	Accessing the resource you were trying to reach is forbidden	No links
404	Payment not found	No links
500	Internal server error	No links

There are two possible values to be recognized for the parameter “purchased”:

- **“true”**: The flight ticket has been issued by the airline to the relevant user.
- **“false”**: The flight ticket has not been issued by the airline to the relevant user.

The scenarios we would face and the power of this functionality if it is enabled:

- **Scenario 0**: The payment has been **AUTHORISED / SETTLED** successfully, and the ticket has been issued by the airline (ticket purchase = true). This is the normal scenario for a success collection and no action is triggered.
- **Scenario 1**: The payment has been **FAILED** and the ticket has not been issued by the airline (ticket purchase = false). This is the normal scenario for a failed transaction and no action is triggered.
- **Scenario 2**: The payment has been **AUTHORISED / SETTLED** through the IATA Pay Platform, but the airline **did not issue** the flight ticket to the user.

In this scenario, the Airline can configure in the Merchant configuration that in these cases an **automatic refund** will be triggered in order to return the payment with no purchase associated. This functionality can be enabled in Merchant Configuration Advanced.

Create refund if no ticket purchase ?

Yes

You can see an example below:

Step 1. The payment has been authorized or settled by the user and will appear in the IATA Pay Admin Portal with the corresponding success statuses:

Payment details ×

[Details](#)
[Refunds](#)
[Events](#)

+ New refund

Merchant	XX0000000		
TPP Transaction Id	25000294155		
Merchant Payment Id	Not defined		
IATA Payment Id	PLW19JJNT50VX		
Settlement Id			
Amount	₹13.00		
Country	IN	Locale	en-GB
Description	Lambda Airlines		
PNR	Not defined		
Departure Date	Not defined		
Status	● SETTLED		
Compliance Done	☑		
Creation date	02/04/2025, 9:48 AM		
Finish date	02/04/2025, 9:48 AM		
Last update date	02/04/2025, 9:50 AM		
Clearance date	-		

Step 2. The Airline will have to fill the parameter “purchased” with the value = “**false**”.

To do the test, substitute with your “iataPaymentId” and click on “Execute”.

Tickets purchase Ticket purchase confirmation. ⬆

POST

/api/v1/payments/{iataPaymentId}/purchase

Notify ticket purchase result related with the payment in airline platform.

⬆

🔒

Parameters

Cancel

Reset

Name	Description
iataPaymentId * required	Unique identifier generated by IATA Pay.
string (path)	
	P5WR4ETCTYDOP

Request body required

application/json

```
{
  "purchased": false
}
```

Execute

The outcome of this scenario **will automatically perform a refund** of all the amount the flight ticket has, from IATAPAY to the relevant user. Like any refund, it will be executed only when the payment is in SETTLED status. So, the airline doesn’t have to take any action on it.

If the payment status is still in AUTHORISED, the refund will be on hold until the payment changes to SETTLED.

In accordance with the refund, all the rest of business rules will apply.

>	●	P6E4OGJ00VGTB Merchant payment id is not defined	€12.00 - €12.00	ES	XX00000004	07/18/2024, 12:00 PM	07/18/2024, 12:00 PM
---	---	---	--------------------	----	------------	----------------------	----------------------

These transactions are included in the Airline reconciliation/transactional reports for transparency and potential need in case of end customer queries. The impact on the net amount is zero since there should always be a payment and its return for the same amount. Both Payment and Refund should have finally Settled status.

You can test the same functionality with the second option substituting “merchantId” and “merchantPaymentId”.

- **Scenario 3:** the payment has **FAILED** through the IATA Pay Platform, but the airline **issued** the flight ticket to the user.

There is extremely low likelihood of this case and could be due to any unexpected glitch, however it has been implemented this control (including alerts to IATA Pay support team). In these cases, immediately action should be taken. It is strongly advisable that the Airline fulfills reliable information in the payment creation parameter “departureDate” so that the severity of the issue can be better managed. The “departureDate” parameter will be of great importance for exception handling and would simplify the managing process.

You can see an example below:

Step 1. The payment has failed and will appear in the IATA Pay Admin Portal with this status:

Payment details				×
<div> Details Refunds Events </div>				
Merchant	XX00000000			
TPP Transaction Id	Not defined			
Merchant Payment Id	Not defined			
IATA Payment Id	PX87KHGLW0KJD			
Settlement Id				
Amount	₹5.00			
Country	IN	Locale	en-GB	
Description	Lambda Airlines			
Payment Link:	https://integration.iata-pay.iata.org/api/v1/payments/PX87KHGLW0KJD/checkout/redirect			
	Send this link to the customer to proceed with payment authorization.			
PNR	Not defined			
Departure Date	Not defined			
Status	<div> <div>●</div> <div>FAILED</div> <div>Cause</div> <div>Tpp error</div> </div>			
Failure details	Transaction Failed			
Compliance Done				
Creation date	02/04/2025, 9:49 AM			
Finish date	02/04/2025, 9:49 AM			
Last update date	02/04/2025, 9:49 AM			
Clearance date	-			

Step 2. The Airline will have to fill the parameter “purchased” with the value = “true”

Tickets purchase
Ticket purchase confirmation.

POST
/api/v1/payments/{iataPaymentId}/purchase
Notify ticket purchase result related with the payment in airline platform.

Parameters

NameDescription

iataPaymentId * required
string
(path)
Unique identifier generated by IATA Pay.

PLJRT07E0QRJ9

Request body

required

application/json

```
{
  "purchased": true
}
```

Execute

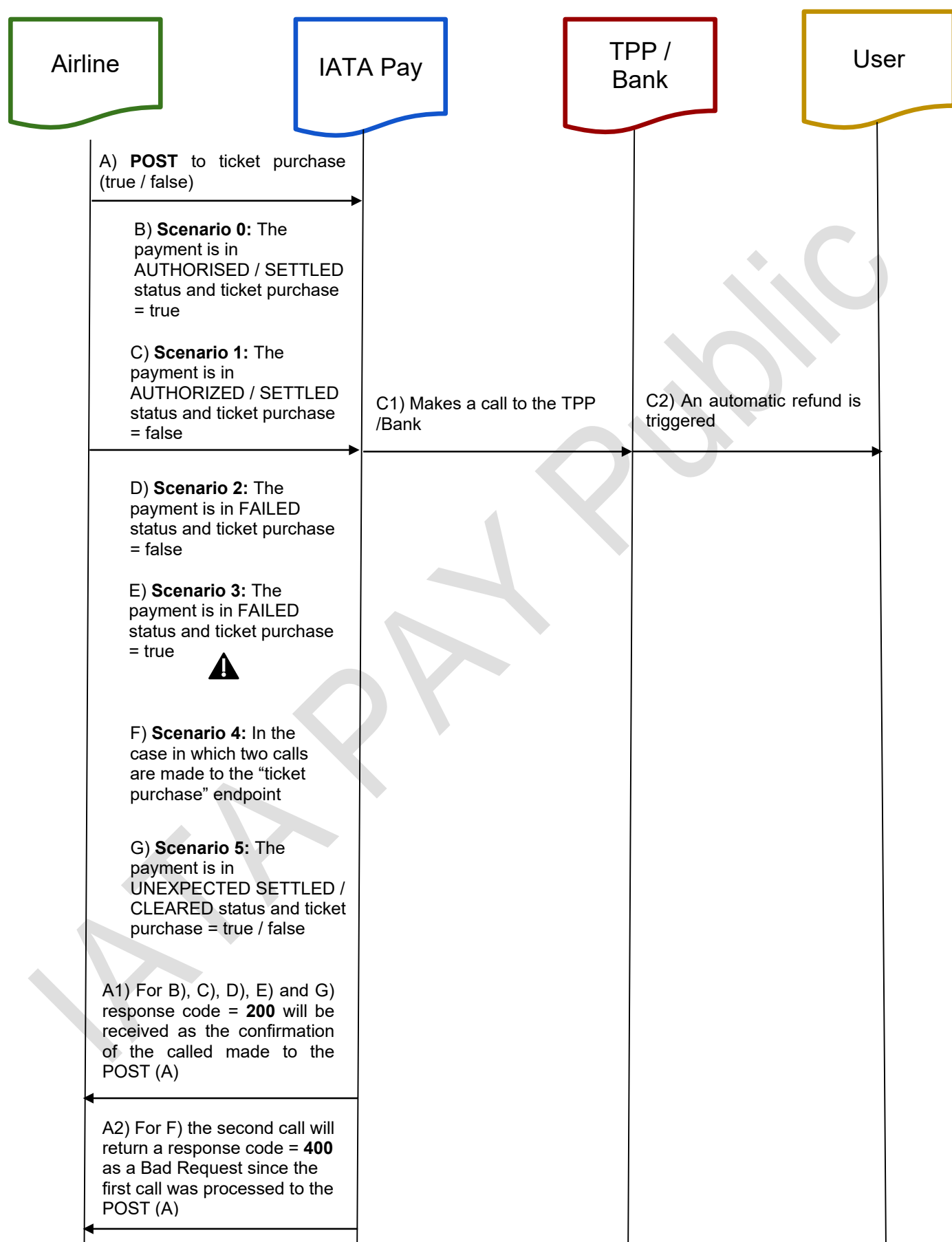
To do the test, substitute with your “iataPaymentId” and click on “Execute”

The outcome of this scenario **will be to send an alert notification to the IATA Pay Business Team and IATA Pay Support Team** who will talk to the airline to handle these cases. It will remain as Failed status in the IATA Pay Admin Portal.

●	PLUG1H4JXTPT5 Merchant payment id is not defined	€1.00	NL	RR1230000	06/24/2024, 6:13 PM	06/24/2024, 6:34 PM
---	---	-------	----	-----------	---------------------	---------------------

You can test the same functionality with the second option substituting “merchantId” and “merchantPaymentId”.

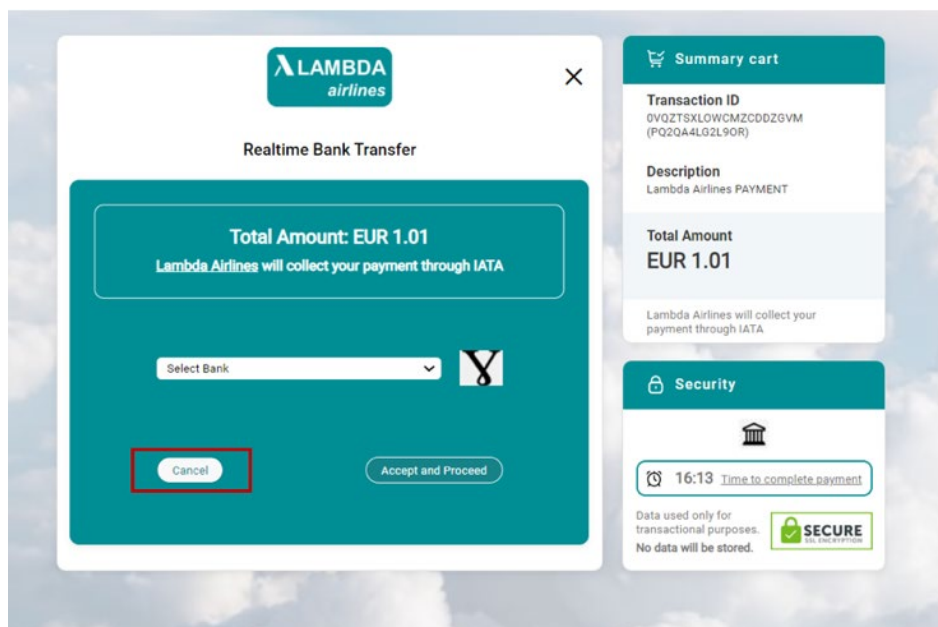
- **Scenario 4:** In the case in which two calls are made to the “ticket purchase” endpoint, the second one will return an error response = BAD REQUEST, since the first call was processed to request the ticket issuance status (true / false).
- **Scenario 5:** The payment has been **UNEXPECTED SETTLED** or **CLEARED**, the ticket issuance status and date time stamped is saved but no action is triggered.



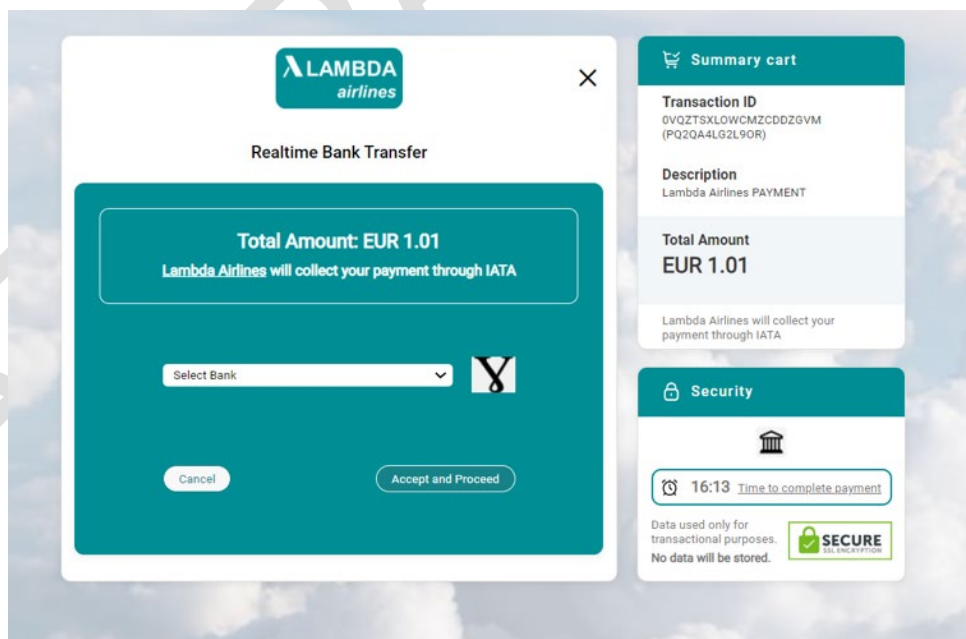
8.7.11 Error / Failure codes simulation

There are different ways to force error in order to test all the possible flows in a payment / refund:

- The CANCEL error can be simulated by cancelling the payment in the checkout webpage. This can be done by pressing the “Cancel” button.



- The TIMEOUT_CREATED error can be simulated waiting 10 minutes at the checkout webpage.



- The TIMEOUT_INITIATED error can be simulated waiting 10 minutes at the bank (Gamma Bank) webpage.



Confirm your payment

Amount £1.02

Country GB

IBAN Main - GB32BARC20038044612483

☐ I agree to make the payment

SMS verification code has been sent. Transaction code 2945729.

Verification code

Accept

- The `TIMEOUT_LOCKED` (Only for refunds) error can be simulated by enabling this feature in the Merchant Advanced configuration and waiting the max time for "unlock" refunds (configured in your merchant advanced settings).

Lambda Airlines (Net Banking)

×

① General ② Credentials ③ Styles ④ TPPs **⑤ Advanced** ⑥ Report ⑦ Sepa Threshold

This section defines other advanced properties related to this merchant

Payment authorization ②

Authorization Timeout (minutes 1 - 60)

19 minutes

Refund approvals ②

Refunds require approval ☒

Approval Timeout (minutes)

36,000 minutes

Only manual refunds require approval ☐

Refund limits ②

Hong Kong Dollar (HK\$) Margin Amount

Margin Amount

Mexican Peso (MX\$) Margin Amount

Margin Amount

Euro (€) Margin Amount

Margin Amount

Colombian Peso (COP) Margin Amount

Margin Amount

Cancel

✓ Save

- The `SUCCESS_WHEN_FAILED` error is an unexpected and very unusual error that will be internally managed by IATA Pay (not notified to the airline as the error of the payment was already notified and the airline has to take no action) when a notification (that implies money is coming to IATA Pay

system) of a payment that is in failed status is received, thus changing the status of the payment from failed to UNEXPECTED SETTLED.

- The **TIMEOUT_AUTHORIZED** error is an unexpected and very unusual error that requires a cancelation of the provided service, if this has been already issued. If the system is notified with this error code, an internal alert can be generated in order to handle this case either manually or automatically.
- The **UNEXPECTED** error is any other error unknown by IATA Pay. This is the more usual error in which the transaction fails for any cause before the authorization/settlement.
- The **TPP ERROR** will get an unexpected error produced usually on transaction initiation on TPP / PISP side. Therefore, it will be necessary to contact to the corresponding TPP/PISP.
- **CANCEL**. The user has cancelled the transaction.
- **PAYMENT AUTHORIZATION FAILED**. The TPP reject the authorization of the payment.
- **PAYMENT URL RELOADED**. The user refreshes the purchase page.
- **INVALID TOKEN ID**. The token id introduced by the airline is incorrect.

SUCCESS_WHEN_FAILED, **TIMEOUT_AUTHORIZED** and **UNKNOWN** errors can be simulated through Gamma's Bank webpage. At the login page, set "error" as username (the password field is irrelevant).



In the next page, a "Force Error" select field allows you to choose which of these errors to simulate. This field is only present when accessing with username "error".



Confirm your payment

Force error	<div>▼</div>
Amount	None
Country	UNKNOWN
IBAN	TIMEOUT_AUTHORIZED
	SUCCESS_WHEN_FAILED
	Main - GB32BARC20038044612483

☐ I agree to make the payment

SMS verification code has been sent. Transaction code 2945729.

Verification code

Accept

9 IATA PAY in Call Centers

IATA Pay will also support the sale of tickets through the Call Center framework. This organizational structure involves the dynamic relationship between an inbound or outbound type of call center services.

In order to successfully complete the booking and sale of the flight ticket, through this method, the airline will be able to do so through the method explained in the following section:

9.1 Payment creation via API according to the swagger specs

This feature allows to create a payment from the back-office side, via API and its corresponding development. An airline operator can create a "Payment request" and get a link to send to the customer so it can perform the payment. Once the payment is completed, the operator is notified in order to continue with the ticket emission process. Take into account that the VPA Validation can take some minutes for processing the data.

Bear in mind that you can use the *'retriableUntil'* field, as maximum date-time to retry the payment when it fails. No retries allowed when this field is empty. With ISO date-time format and compliant with UTC time zone.

From airline's back-office side, you will need to proceed with the payment by completing the following steps:

1. From airline side, you must be logged in as explained in [Client login](#) to initiate a payment
2. The airline checks if IATA Pay payment method is available for the country/currency where the user is purchasing.

*Request **POST** /api/v1/payments/allow*

IATA Pay responds with "Accepted" or "Not accepted" response code. If the response is "Accepted", this flow continues; else, the customer cannot pay with IATA Pay.

3. From airline side, you should initiate the payment in IATA Pay sending a POST to /api/v1/payments to initiate a new payment in the platform. The response contains a field called **"checkoutMethods.redirect"** which holds a property called **"redirectUrl"**. The value of this property is the URL that the operator will send to the customer on which the payer will be redirected to progress with payment authorisation.

*Request **POST** /api/v1/payments*

Hereby, you will have to fill the data with the mandatory fields to be gathered in the airline's webpage, in order to initialize a payment request in the IATA Pay Platform. The examples that are in the table are illustrative.

Name	Description
<i>merchantId *</i>	Unique identifier of the merchant in IATA Pay.
<i>merchantPaymentId</i>	Unique identifier assigned by the merchant for this payment.
<i>amount *</i>	Total amount of the payment. Decimals are separated with a point and this is the only symbol.
<i>country *</i>	Code of the country/market. Compliant with ISO 3166 with two characters.
<i>locale *</i>	Language to be used on user messages. Compliant with IETF language tag and RFC 5646.
<i>currency *</i>	Currency of the payment. Compliant with ISO 4217 with three characters.
<i>departureDate *</i>	The departure date is the day of the first flight in the purchase cart. Its format is "YYYY-MM-DD" (e.g., 2022-12-23)
<i>successUrl *</i>	Merchant's URL to redirect the user when the payment finishes successfully.
<i>failureUrl *</i>	Merchant's URL to redirect the user when the payment finishes with error.
<i>Authorization Timeout (per payment)</i>	Time range (min) allowed to authorize the payment. It should be an integer number between 1 and 60 minutes. E.g., if merchant wants the user to have 10 minutes to complete the payment it should be provided in payment request with that value: 'Authorization Timeout: 10'. This field is optional, so, if it is not provided, it will work as follows: Firstly, if the merchant has completed the time within the merchant settings in the advanced options screen (Merchant Authorization Timeout), it will use that time. Secondly, if the above is not completed by the merchant, the default timeout of 19 minutes shall be used for this case.
<i>bankTransferDescription</i>	Info to be placed in the description of the bank transfer. By default, the description in their bank statement will be "AirlineName PAYMENT AirlinePaymentId" (i.e. "LAMBDA AIRLINES PAYMENT L3CDS20190130"). Also, Airline could have the possibility to customize the information they want to show to the customer in the "bankTransferDescription" field. Any non-alphanumeric character could be deleted by the bank. Potentially, other banking scheme references/info may be added along the process (Customer's Bank). IATA works on customising as much as possible those on IATA's hands.
<i>loyalUser</i>	Info about the loyal user logged in the merchant. For example, frequent flyer.
<i>PNR</i>	Passenger Name Record (up to 200 alpha - numeric characters, including underscore, dot and dash). If the merchant is interested in including it.
<i>notificationEmails</i>	Merchant's email/s to notify the payment's authorization or failure.
<i>retriableUntil</i>	Maximum date-time to retry the payment when it fails. No retries allowed when this field is empty. With ISO date-time format and compliant with UTC time zone.
<i>tokenId</i>	It contains information regarding the payment method selected by the payor (these data will not be saved in the system for confidentiality reasons).

	<p>(e.g., "+3470000000" can be used in mobile phone for test purposes in the integration with bizum), (e.g., a set of fictitious VPA ID can be used for test purposes in the integration with VPA: -Scenario VPA ID is invalid: "invalidVPA@iata" -Valid VPA ID but timeout, so fail scenario: "failedtest@iata" -Valid VPA ID resulting in Success: "successtest@iata" (e.g a set of fictitious values that can be used for test purposes in the integration with JK Bank via RTP: - Mobile: "00962791212121" - Alias: "FUADCBJ" - Alias: "TEST09")</p>																		
preferredCheckoutMethod	<p>In the case of India, if VPA is marked, it overrides any other even if it was not the first to be marked. For all other cases, the first one ticked from the list in the merchant configuration is launched. If the value arrives empty, VPA is taken if it is ticked by default. If it is not, the first one ticked from the list in the merchant configuration. If they send a preferred payment method that the merchant does not have configured, it fails in the initiation of the payment. If the value received is not one of the available ones, it will fail at payment creation.</p> <table><tr><th>COUNTRY</th><th>VALUE</th><th>NOTES</th></tr><tr><td>IN</td><td>Netbanking</td><td>India Netbanking</td></tr><tr><td>IN</td><td>QR</td><td>UPI India QR</td></tr><tr><td>IN</td><td>VPA</td><td>UPI India VPA</td></tr><tr><td>JO</td><td>QR</td><td>CliQ Jordan QR</td></tr><tr><td>JO</td><td>RTP</td><td>CliQ Jordan RTP</td></tr></table>	COUNTRY	VALUE	NOTES	IN	Netbanking	India Netbanking	IN	QR	UPI India QR	IN	VPA	UPI India VPA	JO	QR	CliQ Jordan QR	JO	RTP	CliQ Jordan RTP
COUNTRY	VALUE	NOTES																	
IN	Netbanking	India Netbanking																	
IN	QR	UPI India QR																	
IN	VPA	UPI India VPA																	
JO	QR	CliQ Jordan QR																	
JO	RTP	CliQ Jordan RTP																	
tokentype	<p>For payments made in Jordan with JK Bank RTP option, it is required to complete the tokentype choosing one of the following options: "mobile" or "alias".</p>																		

Payment Method	Payer Input	Format
Pay To	Pay ID	Maximum of 256 characters in lower case. Regex patter: <code>^[!-@\[\~]]-@[\~]{0,254}![-@\[\~]]\$</code>
	BSB	6 character number Regex patter: <code>^\d{6}\$</code>
	Account	Number of 5 to 9 characters. Regex patter: <code>^\d{5,9}\$</code>
VPA	VPA	Email with a minimun of 5 and maximun 99 characters. Regex patter : <code>^[a-zA-Z0-9.-]{2,49}@[a-zA-Z]{2,49}\$</code>
CliQ	ALIAS	Please find below the Alias format and regex patter:

		<ul style="list-style-type: none"> • Alphanumeric • Minimum of 3 characters • Maximum of 10 characters • Contains at least 1 letter <p>Regex: <code>^(?=.*[A-Za-z])[A-Za-z0-9]{3,10}\$</code></p>
ANZ	BSB	<code>^(0[1-9] [1-9]d)d{4}\$</code>
CLIQ	Mobile	00962798014928 (14 digits)
BIZUM	Mobile number	9 character number Regex patter: <code>^\d{9}\$</code>

Furthermore, airlines have the option to include details of the several items of a purchase by using this information:

<i>item – description</i>	Description of the item (e.g., <i>MAD-LGW 2019-12-31 12:30</i>)
<i>item – sku</i>	SKU of the item. Pattern: {1,100} (e.g., <i>X0123456789</i>).
<i>item – price</i>	Price of the item (e.g., <i>4567.89</i>). Decimals are separated with a point, and this is the only symbol.
<i>item – currency</i>	Currency of the item. Pattern: {1,100} (e.g., <i>EUR</i>) Compliant with ISO 4217 with three characters.
<i>item - quantity</i>	Quantity of the item (e.g., <i>1</i>). Decimals are separated with a point, and this is the only symbol.

Below is an example of this POST request to *api/v1/payments*:

```
{
  "merchantId": "XX0000000",
  "merchantPaymentId": "HR3DC2-20190321.125515.123456",
  "amount": 4567.89,
  "currency": "EUR",
  "country": "DE",
  "locale": "en-GB",
  "bankTransferDescription": "MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "loyalUser": {
    "userId": "mlking",
  },
  "payerInfo": {
    "tokenId": "+347000000000"
  },
  "pnr": "HR3DC2",
  "authorizationTimeout": 18,
  "items": [
```

```
{
  "description": "MAD-LGW 2019-12-31 12:30",
  "sku": "X0123456789",
  "price": 4567.89,
  "currency": "EUR",
  "quantity": 1
},
"redirectUrls": {
  "successUrl": "https://lambda.sandbox.iata-pay.iata.org/web/#/confirm",
  "failureUrl": "https://lambda.sandbox.iata-pay.iata.org/web/#/error"
},
"notificationUrl": "https://lambda.sandbox.iata-pay.iata.org/api/v1/lambda/payments/iatapay/notification",
"notificationEmails": [
  "sales@lambda-airlines.com"
],
"retriableUntil": "2023-03-30T07:17:26.343Z",
"departureDate": "2022-12-23",
"preferredCheckoutMethod": "QR"
}
```

Note that 'retriableUntil' field should be always a future date.

- The response to this request contains a field called "checkoutMethods.redirect" which holds a property called "redirectUrl". The value of this property is the URL to send to the customer in order to complete the payment process on a browser

This link is not sent by IATA Pay to the customer automatically. Emails to the customer are the responsibility of the airline.

Below is an example of the response to the previous POST to api/v1/payments:

```
{
  "iataPaymentId": "PT333G378YHJL",
  "merchantId": "XX0000000",
  "merchantPaymentId": "123420230214",
  "amount": 1.89,
  "unrefundedAmount": 1.89,
  "refundedAmount": 0,
  "currency": "EUR",
  "retries": 0,
  "isRetrievable": false,
  "status": "CREATED",
  "failureCode": null,
  "lockReason": null,
  "failureDetails": null,
  "creationDateTime": "2023-02-14T15:49:00.422307",
  "finishDateTime": null,
  "updateDateTime": "2023-02-14T15:49:00.422307",
  "clearanceDateTime": null,
  "retriableUntil": "2023-02-16T15:46:35.875",
  "country": "DE",
}
```

```

"accountCountry": "DE",
"locale": "en-GB",
"bankTransferDescription": "MAD-LGW 2019-12-31 12:30 http://www.lambda-
airlines.com/PNR/HR3DC2",
"pnr": "test",
"refunds": [],
"checkoutMethods": {
  "redirect": {
    "redirectUrl": "https://sandbox.iata-
pay.iata.org/api/v1/payments/PT333G378YHJL/checkout/redirect"
  },
  "tpp": null
},
"complianceCheckDone": false
}

```

If you are interested in other relevant data that IATA Pay returns to the airline after the payment has been initiated, see chapter 5, [Response – Data from IATA Pay to the Airline](#)

5. When customer clicks on the URL, the airline redirects the user to IATA Pay's webpage (the "redirectUrl") where the customer is asked for the required bank information, therefore redirected to the TPP/bank's page.
6. Once customer fills the required information, IATA Pay will redirect the user to the airline's website, either to "redirectUrls.successUrl" or to "redirectUrls.failureUrl" (included in step 3) depending on the success/failure.

In section "[API Specification / Basic flows testing](#)" there is an example about how to test this with your airline client credentials.

IATA Pay receives a payment authorization. IATA Pay will forward the **AUTHORIZED** notification to the airline's website with a standard format. When the airline receives the AUTHORIZED notification, the paid services can be provided (for example the ticket can be issued).

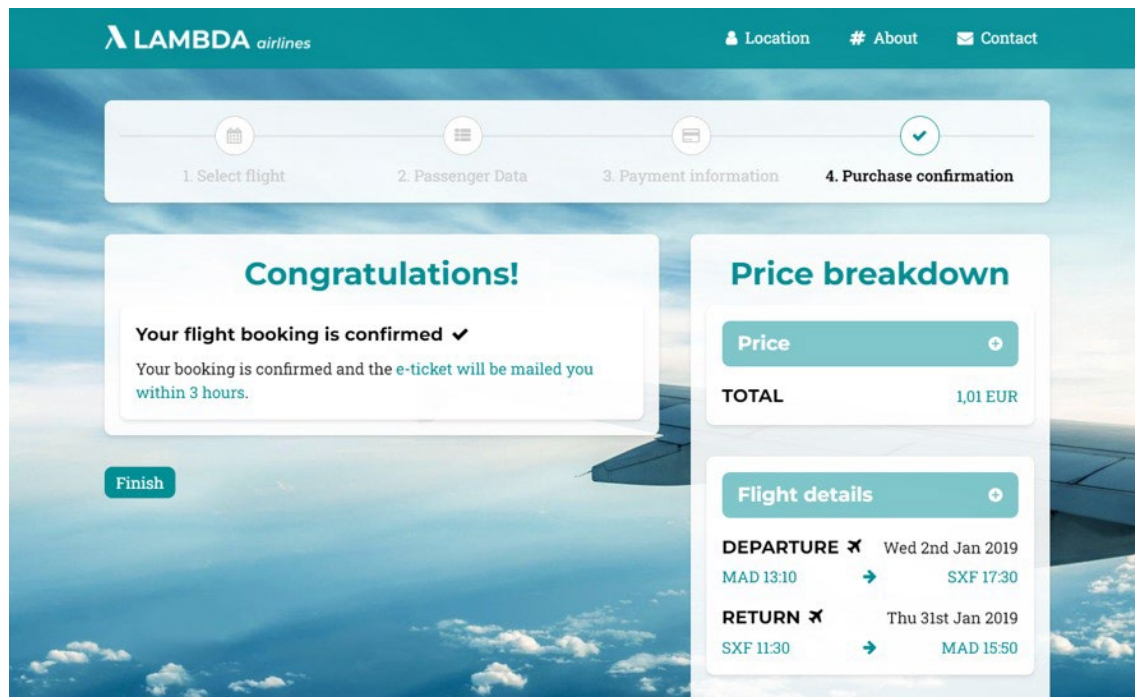
If an error occurred after or before the AUTHORIZATION, then IATA Pay will notify it to the airline's website with other notification.

IATA Pay will notify as well when receiving the funds with **SETTLED** status. In addition, if the payment is suspected of not meeting any legal criteria, IATA Pay will notify as well when receiving the funds with **TOBEINVESTIGATED** status.

IATA Pay will notify as well when sending the funds to the airline with **CLEARED** status.

Both notifications are **asynchronous**: after or before the user redirection. See more info in the section "[Status Notification](#)".

Once the customer finishes the payment process, it will be redirected to a web page like this:



10 IATA PAY Testing Scenarios

This section outlines various testing scenarios to be executed during the testing phase of integration of any airline with IATA PAY. Testing scenarios are designed to verify the functionality of the system under different conditions.

10.1 Functional Testing Scenarios

There are different test scenarios that should be covered during testing phase of airline integration with IATA PAY. We have grouped cases based on functionality:

- Authentication
 - IPTS-AUT-01. Obtain Access Token.
- Payment Creation
 - IPTS-PCR-01. Payment created successfully.
- Payment Checkout
 - IPTS-PCK-01. User redirected successfully to authorize payment on IATA PAY.
 - IPTS-PCK-02. Airline successfully show IATA PAY lightbox to authorize payment.
- Payment Authorization
 - IPTS-PAU-01. User successfully authorize payment.
 - IPTS-PAU-02. User cancel authorization.
 - IPTS-PAU-03. User close browser.
 - IPTS-PAU-04. User do not authorize payment.
 - IPTS-PAU-05. Unexpected Error.
- Refund
 - IPTS-REF-01. Refund created/processed successfully.
 - IPTS-REF-02. Error on refund creation (amount).
- Admin Portal
 - IPTS-ADM-01. User can login successfully on IATA PAY Admin Portal.
 - IPTS-ADM-02. User can review payments processed.
 - IPTS-ADM-03. User can review refunds.
 - IPTS-ADM-04. User can download CSV report for payments/refunds.



Verify that airline is able to access to the main endpoints of IATA PAY API that are secured with OAuth 2.0 authentication. This test will obtain an access token that will be used on different integration flows.

Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:

- Merchant Id
- ClientId / Client Secret

Client Id / Client Secret provided by IATA PAY team.

Step 1: Prepare header authentication according to section 8.2 Authentication

Step 2: Invoke IATA PAY API endpoint to obtain a token:

- POST <https://sandbox.iata-pay.iata.org/oauth/token>

Airline can extract bearer access token from API response.

- Pass: Response status code 200 ok.
- Fail: Response status code 401 Unauthorized

Example of successful response:

```
"access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.  
    eyJhdWQiOiJYWDAwMDAwMDAtY2xpZW50Iiwic2NvcGuiOiJsICGF5bWVudCjDdLCJpc3MiOiJodHRwciovL3NhbmRib3guaWF0YS1wYXkuaWF0YS5vcmcvYXBpL3YxL  
    29hdXRoiwiZWhwTjoxNzASnJU2Nzk4LClhdXRob3JpdGlscyI6WyJSfGRlcXQyOTFUbKlQU5UX0FMtFYWDAwMDAwMDAiSwianRpIjoiuZ4leSOUFOWtZRrNNVMZ3  
    lneWlOVUtSVkc0IiwiaWY2xpZW50ZXNjaioiWiFGwMDAwMDAwLnswVudCJ9.  
    flIB_i5LSuiclhmMXQRt21xHj8cV2KUUYLU0UIoZstPKAQmZquEbAqCRmBS2giZr1M3RsYJni1tlil72BF5IHd-ybQQ41P1T_3fpHyWvMp1BkYL0xOnGFhgTo-  
    HBkv6lcRcURkJXUID6PJn-PIL27qGDZ_yrNGIbBsCJoYxQ",  
  
"token_type": "bearer",  
"expires_in": 299,  
"scope": "payment",  
"iti": "Sn5xNNP V0406uLgygvyitUKRVG4"
```

Example of error response (not authorized):

```
"timestamp": "2024-03-05T16:36:46.262+00:00",
"status": 401,
"error": "Unauthorized",
"path": "/api/v1/oauth/token"
```


10.3 Testing Scenario IPTS-PCR-01. Payment created successfully

Objective:

Verify that airline can create payments on IATA PAY

Preconditions:

- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)

Test Data:

Prepare mandatory test data to create a payment on IATA PAY. Please review section [Request - Data from the Airline to IATA Pay](#) to prepare required test data for payment.

Steps to Reproduce:

Step 1: Prepare payment request creation.

Step 2: Invoke IATA PAY API endpoint to create a payment:

- POST <https://sandbox.iata-pay.iata.org/api/v1/payments>

Expected Result:

Airline can extract information about payment created in IATA PAY from API response (payment identifier, status, redirection url, etc.)

Pass/Fail Criteria:

- Pass: Response status code 201.
- Fail: Response status code different from 201.

Notes:

Example of payment creation request:

```
1 {
2   "merchantId": "XX0000000",
3   "merchantPaymentId": "202403051740",
4   "amount": 1.89,
5   "bankTransferDescription": "MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
6   "country": "DE",
7   "currency": "EUR",
8   "locale": "en-EN",
9   "notificationUrl": "https://lambda.integration.iata-pay.iata.org/api/v1/lambda/payments/iatapay/notification",
10  "redirectUrls": {
11    "failureUrl": "https://lambda.integration.iata-pay.iata.org/web/#/error?id=1",
12    "successUrl": "https://lambda.integration.iata-pay.iata.org/web/#/confirm?id=1"
13  }
14 }
```

Example of payment successfully created response:

```
{
  "iataPaymentId": "PJWYF5VUCYW2J",
  "merchantId": "XX0000000",
  "merchantPaymentId": "202403051740",
  "amount": 1.89,
  "unrefundedAmount": 1.89,
  "refundedAmount": 0,
  "currency": "EUR",
  "retries": 0,
  "isRetrievable": false,
  "status": "CREATED",
  "failureCode": null,
  "lockReason": null,
  "failureDetails": null,
  "creationDateTime": "2024-03-05T16:52:34.645693",
  "finishDateTime": null,
  "updateDateTime": "2024-03-05T16:52:34.645693",
  "clearanceDateTime": null,
  "retryableUntil": null,
  "departureDate": null,
  "country": "DE",
  "accountCountry": "DE",
  "locale": "en",
  "bankTransferDescription": "MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
  "pnr": null,
  "refunds": [],
  "checkoutMethods": {
    "redirect": {
      "redirectUrl": "https://sandbox.iata-pay.iata.org/api/v1/payments/PJWYF5VUCYW2J/checkout/redirect"
    },
    "tpp": null
  },
  "complianceCheckDone": false,
  "settlementId": null,
  "purchased": null,
  "purchasedDateTime": null
}
```

Example of error on payment creation:

```
{
  "status": 400,
  "error": "BAD_REQUEST",
  "message": "There is already a payment with merchantPaymentId \"202403051740\" for merchant with ID \"XX0000000\"",
  "details": null
}
```

10.4 Testing Scenario IPTS-PCK-01. User redirected successfully to authorize payment on IATA PAY

Objective:

Verify that airline can redirect user to complete checkout process of payment created on IATA PAY. This test scenario is prepared for Airlines who has integrated using redirect checkout method.

Preconditions:

- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
- Airline has completed checkout integration following steps of section [5.4.1. Redirect Way](#)
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Payment has been successfully created on IATA PAY on testing scenario IPTS- PCR-01.

Test Data:

Redirect URL obtained from IATA PAY API payment creation response (redirectUrl)

Steps to Reproduce:

Step 1: Use redirection url in browser to redirect user to IATA PAY to authorize the payment.

Expected Result:

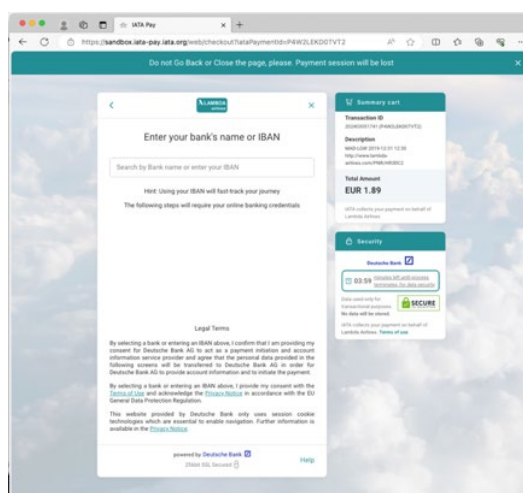
IATA PAY checkout page will appear to allow users to complete payment. This page will have different designs depending on the different markets/payment methods airline has included in the integration.

Pass/Fail Criteria:

- Pass: IATA PAY checkout page is shown for market/payment method.
- Fail: User is redirected to IATA PAY feedback page or to failureUrl provided on payment creation data.

Notes:

Example of IATA PAY checkout page for EUR Germany market with SEPA payment method:





Example of IATA PAY feedback page:

A screenshot of a web browser showing a feedback survey. The browser's address bar displays the URL: <https://integration.iata-pay.iata.org/web/survey?iataPaymentId=PjKLZLF23YNQH>. The survey is titled "Survey" and has a red header that reads "PAYMENT SESSION INTERRUPTED" with the subtext "We are redirecting you back to Lambda Airlines". Below this, the survey asks for feedback with the heading "Help us improve with your feedback". There are five radio button options: "I wanted to see if there were some additional perks.", "I don't want to pay directly with my bank account.", "I did not know how to complete the payment.", "I don't feel secure about providing bank details.", and "I couldn't find my bank.". To the right of these options is an icon of a clipboard with a checklist. Below the options is a text input field labeled "Description (optional)" with the placeholder text "Tell us more about it.". A green "Send survey" button is located at the bottom right of the survey form. The background of the page is a light blue sky with white clouds.



10.5 Testing Scenario IPTS-PCK-02. Airline successfully show IATA PAY lightbox to authorize payment

Objective:

Verify that airline can show lightbox popup panel to complete checkout process of payment created on IATA PAY. This test scenario is prepared for Airlines who has integrated using lightbox checkout method.

Preconditions:

- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
- Airline has completed checkout integration following steps of section [5.4.2. Light Box Way](#)
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Payment has been successfully created on IATA PAY on testing scenario IPTS- PCR-01.

Test Data:

Response data obtained from IATA PAY API payment creation response.

Steps to Reproduce:

Step 1: Use response from payment creation to invoke js function checkout to show IATA PAY lightbox to authorize the payment.

Expected Result:

IATA PAY lightbox popup panel will appear to allow users to complete payment. This page will have different designs depending on the different markets/payment methods airline has included in the integration.

Pass/Fail Criteria:

- Pass: IATA PAY checkout page is shown for market/payment method.
- Fail: IATA PAY lightbox panel is not shown.

Notes:



10.6 Testing Scenario IPTS-PAU-01. User successfully authorize payment

Objective:

Verify that airline can continue processing purchase once user has completed checkout process successfully on IATA PAY.

Preconditions:

- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
 - Notification secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Payment has been successfully created on IATA PAY on testing scenario IPTS- PCR-01.
- Airline has initiated checkout process on IATA PAY using redirection or lightbox methods

Test Data:

Use fake data on sandbox environment on IATA PAY

Steps to Reproduce:

Step 1: Complete authorization process in IATA PAY according to the markets:

- EUR (All countries) - SEPA:
 - Choose DEMO bank related to country.
 - Use fake/random data for rest of screens related with authorization.
- EUR (Spain) – BIZUM:
 - Use phone number: +3470000000 and payment amount > 5 €
- INR (India) - VPA:
 - Use successtest@iata
- INR (India) – Lightbox option for QR and VPA:
 - Wait 20s to simulate a success processing.
- JOD (Jordan) – CliQ:
 - Mobile: “00962791212121”
 - Alias: “FUADCBJ”
 - Alias: “TEST09”
- Rest of markets: Click on simulate button.

Expected Result:

- Redirect Checkout Method Integration option:
 - IATA PAY will redirect back to airline using returnUrl provided by on payment creation data.
- Lightbox Checkout Method Integration option:
 - IATA PAY will close lightbox popup panel.
 - Call iatapayNotify function to return navigation control to airline.
- Notification (conditional - airline has provided notificationUrl on payment creation data):
 - Review section [5.8. Finish Notification](#) and [8.6. Notification Specification](#)
 - IATA PAY send webhook notification with payment status:
 - AUTHORIZED

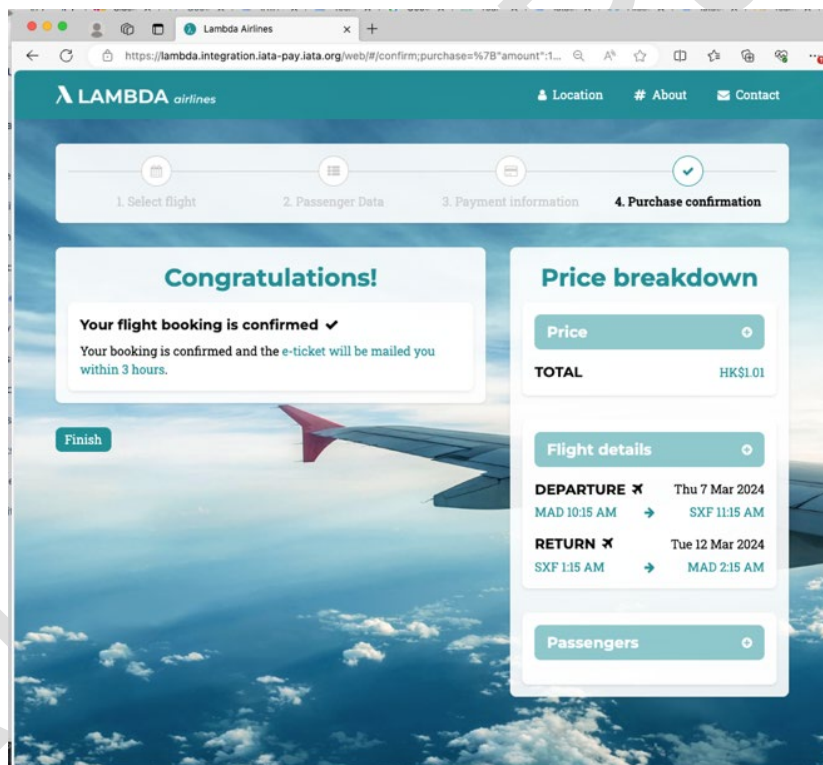
- SETTLED (depending on markets).
 - Airline can update payment status on his system.
- If Airline do not process IATA PAY notifications:
 - Review section [5.8. Finish Notification](#)
 - Airline should check payment status when navigation is back to airline website using API endpoint GET /api/v1/payments/{iataPaymentId}

Pass/Fail Criteria:

- Pass:
 - Airline successUrl page is shown.
 - Optional: Airline receives, and process AUTHORIZED and SETTLED notifications.
 - Optional: Airline checks payment status.
- Fail:
 - Airline successUrl page is not shown.
 - Error on processing notification or checking payment status.

Notes:

Example success url page of Lambda Airline:



Example of AUTHORIZED notification sent:

```

1  {
2    "iataPaymentId": "PD6DHWPF10DA",
3    "merchantId": "XX0000000",
4    "merchantPaymentId": "5V8NJWE4CP60KR3Z1REW",
5    "amount": 1.01,
6    "unrefundedAmount": 1.01,
7    "refundedAmount": 0,
8    "currency": "HKD",
9    "retries": 0,
10   "isRetrievable": false,
11   "status": "AUTHORIZED",
12   "failureCode": null,
13   "lockReason": null,
14   "failureDetails": null,
15   "creationDateTime": "2024-03-06T08:43:51",
16   "finishDateTime": "2024-03-06T08:44:08",
17   "updateDateTime": "2024-03-06T08:46:20",
18   "clearanceDateTime": null,
19   "retryableUntil": null,
20   "departureDate": null,
21   "country": "HK",
22   "accountCountry": "HK",
23   "locale": "en-GB",
24   "bankTransferDescription": "Lambda Airlines PAYMENT",
25   "pnr": null,
26   "refunds": [],
27   "checkoutMethods": null,
28   "complianceCheckDone": true,
29   "settlementId": null,
30   "purchased": null,
31   "purchasedDateTime": null
32 }

```

10.7 Testing Scenario IPTS-PAU-02. User cancel authorization

Objective:

Verify that airline can manage user cancels purchasing process during IATA PAY checkout.

Preconditions:

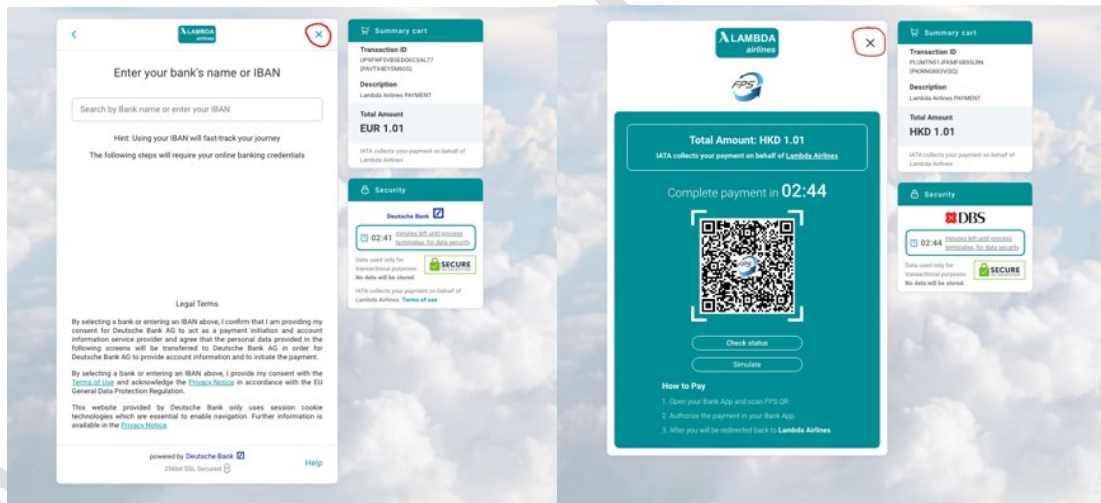
- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
 - Notification secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Payment has been successfully created on IATA PAY on testing scenario IPTS- PCR-01.
- Airline has initiated checkout process on IATA PAY using redirection or lightbox methods.

Test Data:

N/A

Steps to Reproduce:

Step 1: User cancel payment processing on checkout page (some examples below)



Expected Result:

- Redirect Checkout Method Integration option:
 - IATA PAY show feedback page to request user comments.
 - IATA PAY will redirect back to airline using failureUrl provided by on payment creation data.
- Lightbox Checkout Method Integration option:
 - IATA PAY show feedback page to request user comments.
 - IATA PAY will close lightbox popup panel.
 - Call iatapayNotify function to return navigation control to airline.
- Notification (conditional - airline has provided notificationUrl on payment creation data):
 - Review section [5.8. Finish Notification](#) and [8.6. Notification Specification](#)
 - IATA PAY send webhook notification with payment status FAILED.
 - Airline can update payment status on his system.

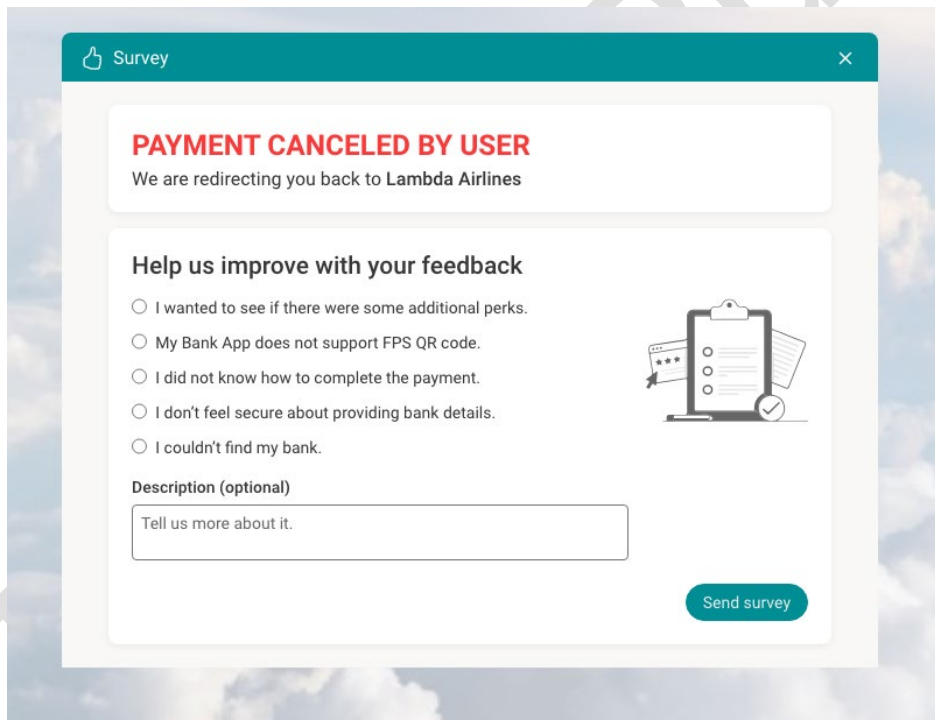
- If Airline do not process IATA PAY notifications:
 - Review section [5.8. Finish Notification](#).
 - Airline should check payment status FAILED when navigation is back to airline website using API endpoint GET /api/v1/payments/{iataPaymentId}.

Pass/Fail Criteria:

- Pass:
 - IATA PAY feedback page is shown.
 - Airline failureUrl page is shown.
 - Optional: Airline receive and process FAILED notification.
 - Optional: Airline checks payment status and verifies that it is FAILED.
- Fail:
 - Airline failureUrl page is not shown.
 - Error on processing notification or checking payment status.

Notes:

Example of feedback page on IATA PAY:



Example of FAILED notification sent:

```

1  {
2    "iataPaymentId": "P8XFFURQXJPUW",
3    "merchantId": "XX00000000",
4    "merchantPaymentId": "93AZT4PJ6VSR2BJFMU2B",
5    "amount": 1.01,
6    "unrefundedAmount": 1.01,
7    "refundedAmount": 0,
8    "currency": "EUR",
9    "retries": 0,
10   "isRetrievable": false,
11   "status": "FAILED",
12   "failureCode": "CANCEL",
13   "lockReason": null,
14   "failureDetails": "The payment has been canceled by the user in DB UI",
15   "creationDateTime": "2024-03-06T09:47:24",
16   "finishDateTime": "2024-03-06T09:47:38",
17   "updateDateTime": "2024-03-06T09:47:38",
18   "clearanceDateTime": null,
19   "retryableUntil": null,
20   "departureDate": null,
21   "country": "DE",
22   "accountCountry": "DE",
23   "locale": "en-GB",
24   "bankTransferDescription": "Lambda Airlines PAYMENT",
25   "pnr": null,
26   "refunds": [],
27   "checkoutMethods": null,
28   "complianceCheckDone": false,
29   "settlementId": null,
30   "purchased": null,
31   "purchasedDateTime": null
32 }

```

10.8 Testing Scenario IPTS-PAU-03. User close browser

Objective:

Verify that airline can manage user has close browser and do not complete payment authorization on IATA PAY checkout.

Preconditions:

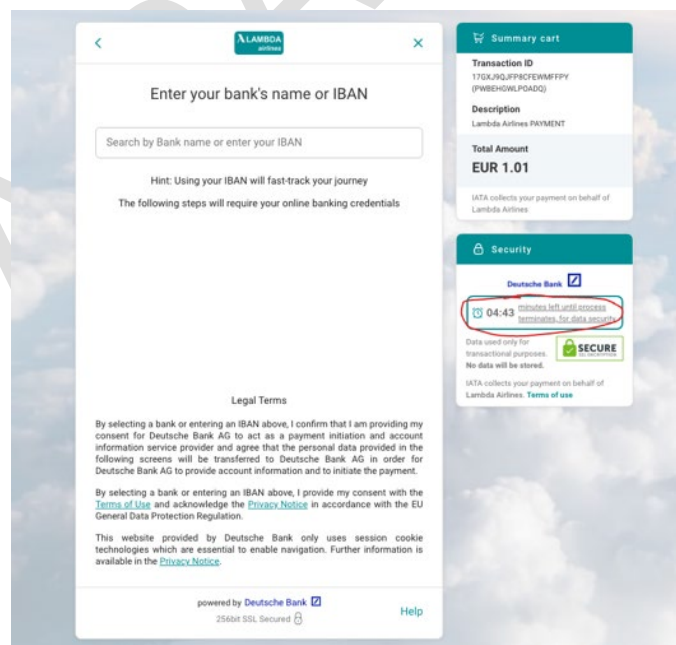
- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
 - Notification secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Payment has been successfully created on IATA PAY on testing scenario IPTS- PCR-01.
- Airline has initiated checkout process on IATA PAY using redirection or lightbox methods.
- Payment Authorization Timeout is configured at merchant level or used by default (18 min)

Test Data:

N/A

Steps to Reproduce:

Step 1: On checkout screen it is shown timeout time for user to complete payment authorization. User close browser on IATA PAY checkout.



Expected Result:

- Redirect Checkout Method Integration option:
 - IATA PAY show feedback page to request user comments.
 - IATA PAY will redirect back to airline using failureUrl provided by on payment creation data.
- Lightbox Checkout Method Integration option:



- IATA PAY show feedback page to request user comments.
- IATA PAY will close lightbox popup panel.
- Call iatapayNotify function to return navigation control to airline.
- Notification (conditional - airline has provided notificationUrl on payment creation data):
 - Review section [5.8. Finish Notification](#) and [8.6. Notification Specification](#).
 - When Authorization Timeout is reached, IATA PAY send webhook notification with payment status FAILED.
 - Airline can update payment status on his system.
- If Airline do not process IATA PAY notifications:
 - Review section [5.8. Finish Notification](#)
 - At the end of Authorization Timeout, airline should check payment status FAILED when navigation is back to airline website using API endpoint GET.
/api/v1/payments/{iataPaymentId}
- There is not navigation back to airline due to user has closed browser.

Pass/Fail Criteria:

- Pass:
 - Optional: Airline receive and process FAILED notification
 - Optional: Airline checks payment status and verifies that it is FAILED
- Fail:
 - Error on processing notification or checking payment status.

Notes:

Example of FAILED notification sent:

```
1 {
2   "iataPaymentId": "PWBEHGWLP0ADQ",
3   "merchantId": "XX0000000",
4   "merchantPaymentId": "17GXJ9QJFP8CFEWMFFPY",
5   "amount": 1.01,
6   "unrefundedAmount": 1.01,
7   "refundedAmount": 0,
8   "currency": "EUR",
9   "retries": 0,
10  "isRetrievable": false,
11  "status": "FAILED",
12  "failureCode": "TIMEOUT_INITIATED",
13  "lockReason": null,
14  "failureDetails": "Payment timeout: Payment was not authorised within 5 minute(s).",
15  "creationDateTime": "2024-03-06T09:53:45",
16  "finishDateTime": "2024-03-06T09:58:50",
17  "updateDateTime": "2024-03-06T09:58:50",
18  "clearanceDateTime": null,
19  "retryableUntil": null,
20  "departureDate": null,
21  "country": "DE",
22  "accountCountry": "DE",
23  "locale": "en-GB",
24  "bankTransferDescription": "Lambda Airlines PAYMENT",
25  "pnr": null,
26  "refunds": [],
27  "checkoutMethods": null,
28  "complianceCheckDone": false,
29  "settlementId": null,
30  "purchased": null,
31  "purchasedDateTime": null
32 }
```

10.9 Testing Scenario IPTS-PAU-04. User do not authorize payment

Objective:

Verify that airline can manage user do not complete payment authorization on IATA PAY checkout.

Preconditions:

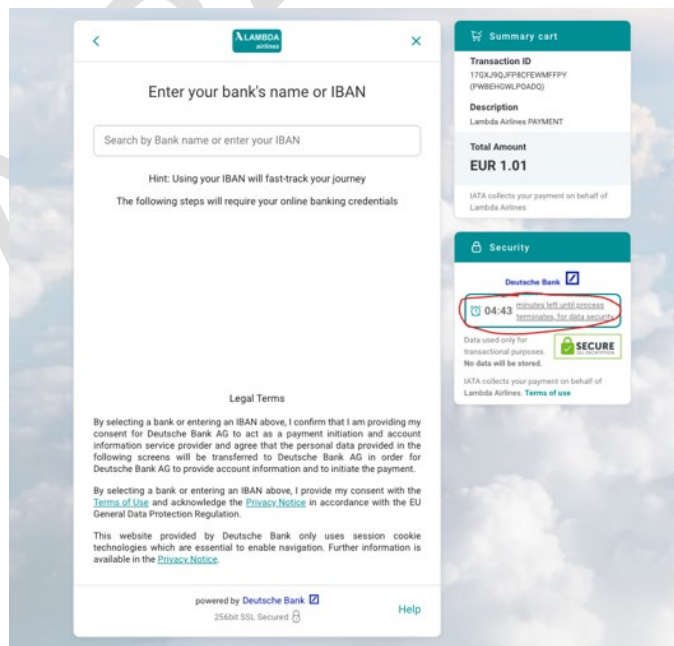
- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
 - Notification secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Payment has been successfully created on IATA PAY on testing scenario IPTS- PCR-01.
- Airline has initiated checkout process on IATA PAY using redirection or lightbox methods.
- Payment Authorization Timeout is configured at merchant level or used by default (18 min).

Test Data:

N/A

Steps to Reproduce:

Step 1: On checkout screen it is shown timeout time for user to complete payment authorization. Wait for authorization timeout without doing any action.



Expected Result:

- Redirect Checkout Method Integration option:
 - When Authorization Timeout is reached, IATA PAY show feedback page to request user comments.



- IATA PAY will redirect back to airline using failureUrl provided by on payment creation data.
- Lightbox Checkout Method Integration option:
 - When Authorization Timeout is reached, IATA PAY show feedback page to request user comments.
 - IATA PAY will close lightbox popup panel.
 - Call iatapayNotify function to return navigation control to airline.
- Notification (conditional - airline has provided notificationUrl on payment creation data):
 - Review section [5.8. Finish Notification](#) and [8.6. Notification Specification](#)
 - When Authorization Timeout is reached, IATA PAY send webhook notification with payment status FAILED.
 - Airline can update payment status on his system.
- If Airline do not process IATA PAY notifications:
 - Review section [5.8. Finish Notification](#)
 - At the end of Authorization Timeout, airline should check payment status FAILED when navigation is back to airline website using API endpoint GET /api/v1/payments/{iataPaymentId}
- There is not navigation back to airline due to user has closed browser.

Pass/Fail Criteria:

- Pass:
 - IATA PAY feedback page is shown.
 - Airline failureUrl page is shown.
 - Optional: Airline receive and process FAILED notification.
 - Optional: Airline checks payment status and verifies that it is FAILED.
- Fail:
 - Airline failureUrl page is not shown.
 - Error on processing notification or checking payment status.

Notes:

Example of FAILED notification sent:

```
1 {
2   "iataPaymentId": "PIRPGG8H27HT7",
3   "merchantId": "XX0000000",
4   "merchantPaymentId": "75W9GU980QP3M7CQXE4C",
5   "amount": 1.01,
6   "unrefundedAmount": 1.01,
7   "refundedAmount": 0,
8   "currency": "EUR",
9   "retries": 0,
10  "isRetrievable": false,
11  "status": "FAILED",
12  "failureCode": "TIMEOUT_INITIATED",
13  "lockReason": null,
14  "failureDetails": "Timeout on initiated status",
15  "creationDateTime": "2024-03-06T10:16:07",
16  "finishDateTime": "2024-03-06T10:21:10",
17  "updateDateTime": "2024-03-06T10:21:10",
18  "clearanceDateTime": null,
19  "retryableUntil": null,
20  "departureDate": null,
21  "country": "DE",
22  "accountCountry": "DE",
23  "locale": "en-GB",
24  "bankTransferDescription": "Lambda Airlines PAYMENT",
25  "pnx": null,
26  "refunds": [],
27  "checkoutMethods": null,
28  "complianceCheckDone": false,
29  "settlementId": null,
30  "purchased": null,
31  "purchasedDateTime": null
32 }
```


10.10 Testing Scenario IPTS-PAU-05. Unexpected Error

Objective:

Verify that airline can manage technical error during payment authorization on IATA PAY checkout.

Preconditions:

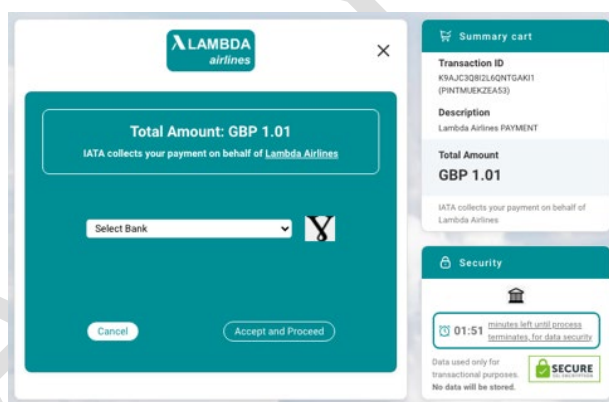
- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
 - Notification secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Payment has been successfully created on IATA PAY on testing scenario IPTS- PCR-01.
- Airline has initiated checkout process on IATA PAY using redirection or lightbox methods.
- Request IATA PAY team to assign MockTPP to country selected to do this test.

Test Data:

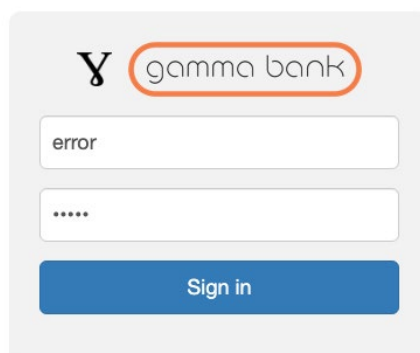
Follow instructions to generate unexpected error described in section [8.7.11. Error / Failure codes simulation](#)

Steps to Reproduce:

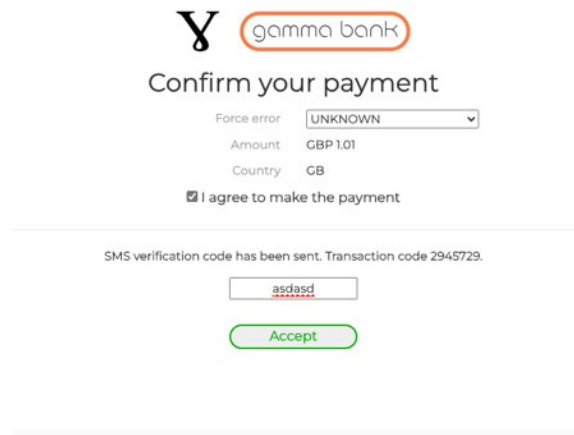
Step 1: Select Gamma Bank and click on button Accept and Proceed.



Step 2: Enter error and fake data on password.



Step 3: Select UNKNOWN on Force error field and fill rest of fields with fake data. Click on button Accept.



Expected Result:

- Redirect Checkout Method Integration option:
 - When Authorization Timeout is reached, IATA PAY show feedback page to request user comments.
 - IATA PAY will redirect back to airline using failureUrl provided by on payment creation data
- Lightbox Checkout Method Integration option:
 - When Authorization Timeout is reached, IATA PAY show feedback page to request user comments.
 - IATA PAY will close lightbox popup panel.
 - Call iatapayNotify function to return navigation control to airline.
- Notification (conditional - airline has provided notificationUrl on payment creation data):
 - Review section [5.8. Finish Notification](#) and [8.6. Notification Specification](#)
 - When Authorization Timeout is reached, IATA PAY send webhook notification with payment status FAILED.
 - Airline can update payment status on his system.
- If Airline do not process IATA PAY notifications:
 - Review section [5.8. Finish Notification](#)
 - At the end of Authorization Timeout, airline should check payment status FAILED when navigation is back to airline website using API endpoint GET `/api/v1/payments/{iataPaymentId}`
- There is not navigation back to airline due to user has closed browser.

Pass/Fail Criteria:

- Pass:
 - IATA PAY feedback page is shown.
 - Airline failureUrl page is shown.
 - Optional: Airline receive and process FAILED notification.
 - Optional: Airline checks payment status and verifies that it is FAILED.
- Fail:
 - Airline failureUrl page is not shown.
 - Error on processing notification or checking payment status.

Notes:

Example of FAILED notification sent:

```
1 {
2   "iataPaymentId": "PILZCI0ZEJN87",
3   "merchantId": "XX00000001",
4   "merchantPaymentId": "U02M3CMCETBCB5JR3KU0",
5   "amount": 1.01,
6   "unrefundedAmount": 1.01,
7   "refundedAmount": 0,
8   "currency": "GBP",
9   "retries": 0,
10  "isRetrievable": false,
11  "status": "FAILED",
12  "failureCode": "UNKNOWN",
13  "lockReason": null,
14  "failureDetails": "Simulated error: UNKNOWN",
15  "creationDateTime": "2024-03-06T10:31:50",
16  "finishDateTime": "2024-03-06T10:32:11",
17  "updateDateTime": "2024-03-06T10:32:11",
18  "clearanceDateTime": null,
19  "retryableUntil": null,
20  "departureDate": null,
21  "country": "GB",
22  "accountCountry": "GB",
23  "locale": "en-GB",
24  "bankTransferDescription": "Lambda Airlines PAYMENT",
25  "pnr": null,
26  "refunds": [],
27  "checkoutMethods": null,
28  "complianceCheckDone": false,
29  "settlementId": null,
30  "purchased": null,
31  "purchasedDateTime": null
32 }
```

10.11 Testing Scenario IPTS-REF-01. Refund created/processed successfully

Objective:

Verify that airline create a refund and it is processed successfully on IATA PAY.

Preconditions:

- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
 - Notification secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#).
- Airline has successfully processed a payment (same day of refund testing to avoid any block condition).

Test Data:

Prepare partial / full amount related to original payment. Data needed:

- IATA PAY paymentId
- Currency of payment

Steps to Reproduce:

Step 1: Prepare refund request creation.

Step 2: Invoke IATA PAY API endpoint to create a refund:

- POST <https://sandbox.iata-pay.iata.org/api/v1/payments/{iataPaymentId}/refund> or
- POST <https://sandbox.iata-pay.iata.org/api/v1/merchants/{merchantId}/payments/{merchantPaymentId}/refund>

Expected Result:

- Refund creation response status code is 200 OK.
- Notification (conditional - airline has provided notificationURL on refund creation data):
 - Review section [5.8. Finish Notification](#) and [8.6. Notification Specification](#)
 - IATA PAY send webhook notification with refund status:
 - AUTHORIZED
 - SETTLED
 - Airline can update refund status on his system.
- If Airline do not process IATA PAY notifications:
 - Review section [5.8. Finish Notification](#)
 - Airline should check refund periodically refund status to get result of refund processing. Airline can use: API endpoint GET [/api/v1/refunds/{iataRefundId}](#)

Pass/Fail Criteria:

- Pass:
 - Response status code on refund creation request is 200 OK.



- Optional: After some minutes, airline receive, and process SETTLED notification.
- Optional: Airline checks refund status until receive SETTLED status.
- Fail:
 - Response status code different from 200 ok.
 - Error on processing notification or checking payment status.

Notes:

Example refund creation response for refund of 1 HKD related to payment PD6DHWP1FIODA:

```
1 {
2   "iataRefundId": "R67NP8710UKI5",
3   "merchantRefundId": null,
4   "amount": 1.00,
5   "currency": "HKD",
6   "bankTransferDescription": "REFUND MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
7   "status": "CREATED",
8   "failureCode": null,
9   "failureDetails": null,
10  "lockReason": null,
11  "creationDateTime": "2024-03-06T11:15:16.347124",
12  "finishDateTime": null,
13  "updateDateTime": "2024-03-06T11:15:19.74971",
14  "clearanceDateTime": null,
15  "settlementId": null
16 }
```

Example of refund status in case of use of notificationUrl on refund creation request or response from get refund status API endpoint:

```
1 {
2   "iataRefundId": "R67NP8710UKI5",
3   "merchantRefundId": null,
4   "amount": 1.00,
5   "currency": "HKD",
6   "bankTransferDescription": "REFUND MAD-LGW 2019-12-31 12:30 http://www.lambda-airlines.com/PNR/HR3DC2",
7   "status": "SETTLED",
8   "failureCode": null,
9   "failureDetails": null,
10  "lockReason": null,
11  "creationDateTime": "2024-03-06T11:15:16",
12  "finishDateTime": "2024-03-06T11:17:00",
13  "updateDateTime": "2024-03-06T11:17:00",
14  "clearanceDateTime": null,
15  "settlementId": null,
16  "iataPaymentId": "PD6DHWP1FIODA",
17  "merchantPaymentId": "5V8NJWE4CP60KR3Z1REW",
18  "paymentAmount": 1.01,
19  "paymentStatus": "SETTLED",
20  "paymentPurchased": null,
21  "merchantId": "XX0000000",
22  "accountCountry": "HK"
23 }
```



10.12 Testing Scenario IPTS-REF-02. Error on refund creation (amount exceeds original payment).

Objective:

Verify that airline can manage error on refunds creation request on IATA PAY.

Preconditions:

- Merchants involved in testing for different currencies has been created in IATA PAY and merchants ids together with credentials has been provided to airline. For each merchant created IATA PAY team will provide to the airline following data:
 - Merchant Id
 - ClientId / Client Secret
 - Notification secret
- Merchant has obtained an access token following instructions of section [8.2 Authentication](#)
- Airline has successfully processed a payment (same day of refund testing to avoid any block condition).

Test Data:

Prepare refund related to original payment but provide an amount that it is higher than amount of related payment.

Steps to Reproduce:

Step 1: Prepare refund request creation.

Step 2: Invoke IATA PAY API endpoint to create a refund:

- POST <https://sandbox.iata-pay.iata.org/api/v1/payments/{iataPaymentId}/refund> or
- POST <https://sandbox.iata-pay.iata.org/api/v1/merchants/{merchantId}/payments/{merchantPaymentId}/refund>

Expected Result:

- Refund creation response status code is 400 (Bad request)
- Refund is not created on IATA PAY

Pass/Fail Criteria:

- Pass: Response status code is 400 Bad Request including message related to error.
- Fail: Response status other different from 400.

Notes:

Example refund creation response:

```
1  {
2    "status": 400,
3    "error": "BAD_REQUEST",
4    "message": "The amount to be refunded (1.00) is greater than the unrefunded amount (0.01): the amount of the payment is 1.01 and the refunded amount is 1.00",
5    "details": null
6  }
```

10.13 Testing Scenario IPTS-ADM-01. User can login successfully on IATA PAY Admin Portal

Objective:

Verify that the user can log into the IATA PAY Administration Portal successfully.

Preconditions:

Merchant user(s) involved in testing and who should have access to the IATAPAY Administration portal should ensure that their profile has been created in the following instances:

- The user(s) have been duly registered on the IATA Customer Portal (Salesforce) with success.
- After completing the prior step, the IATAPAY Administrators properly create the user within the IATAPAY Admin Portal, assigning an appropriate role.
- At last, user(s) receive login credentials (email and password) for the IATA PAY Admin Portal.

Test Data:

N/A

Steps to Reproduce:

Step 1: Confirm user registration on the IATA Customer Portal (Salesforce). Otherwise, the user can register at the following link:

- Link to [IATA Customer Portal](#).

Step 2: Confirm the user registration for IATAPAY service. Otherwise, the user can register at the following link:

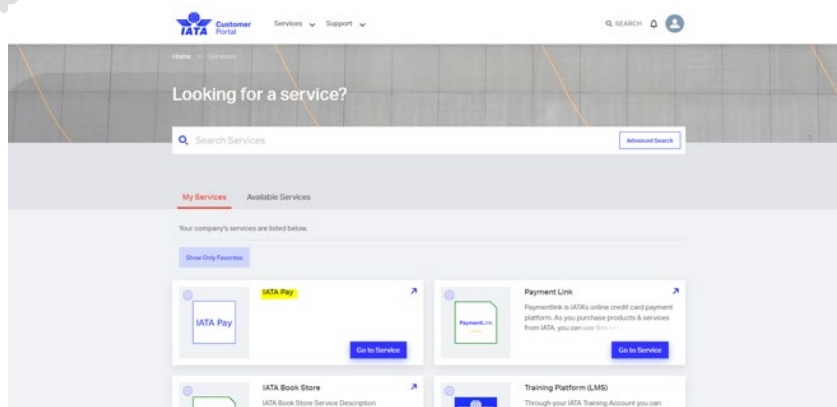
- Link to [IATA PAY Registration](#).

Step 3: Finally, the user logs in to the IATAPAY Admin Portal with IATAPAY credentials.

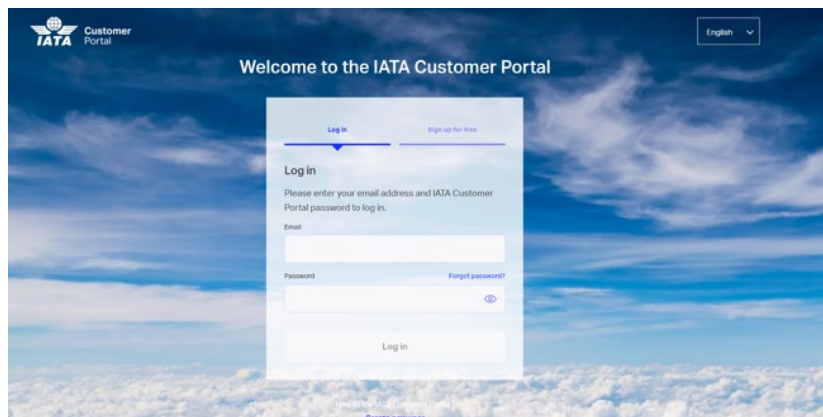
- Link to [IATA PAY Admin Portal](#).

Expected Result:

The user can access the IATAPAY Admin Portal through their IATA Customer Portal account:

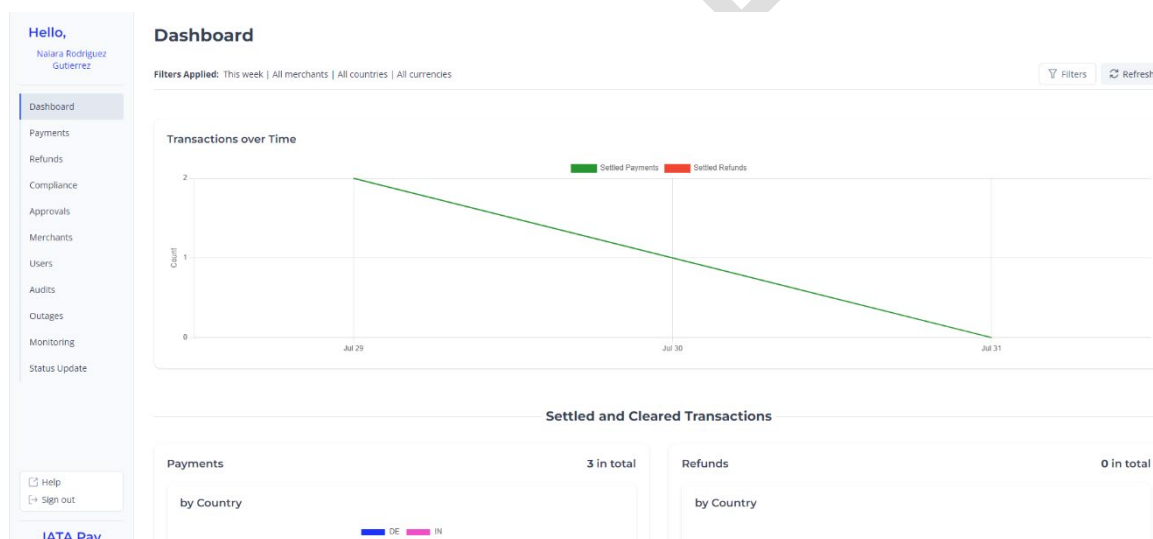


The user inputs the credentials:



Pass/Fail Criteria:

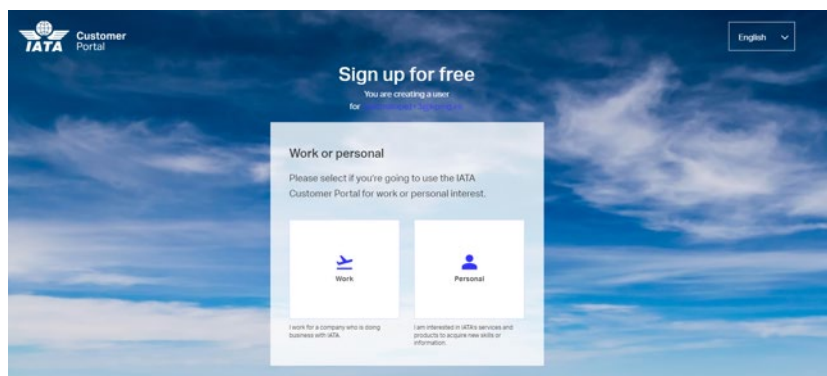
When user logs in, will see a screen resembling the image below. This would imply that the login process has been successfully completed.



Notes:

If it is necessary to register the user in the IATAPAY Admin Portal, the following screens may be encountered:

- a. User will be queried if the account is to be used for work or for personal interest:




IATA Customer Portal English

Sign up for free

You are creating a user for [IATA Customer Portal](#)


Work or personal

Please select if you're going to use the IATA Customer Portal for work or personal interest.



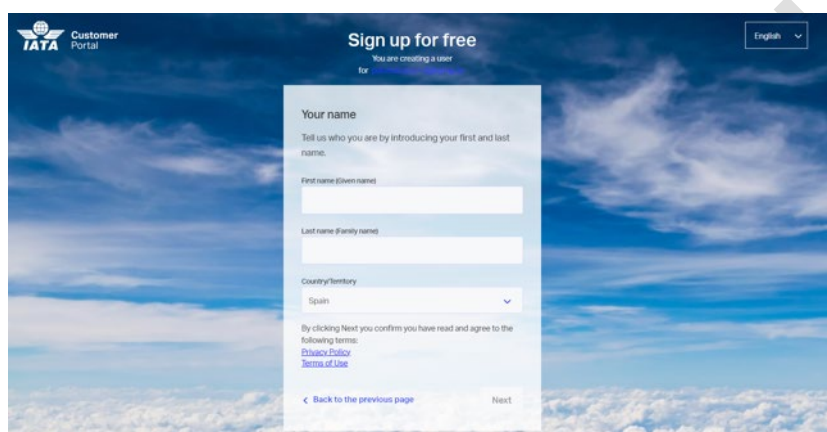
Work

I work for a company who is doing business with IATA.



Personal

I am interested in IATA's services and products to acquire new skills or information.



IATA Customer Portal English

Sign up for free

You are creating a user for [IATA Customer Portal](#)

Your name

Tell us who you are by introducing your first and last name.

First name (given name)

Last name (family name)

Country/Territory

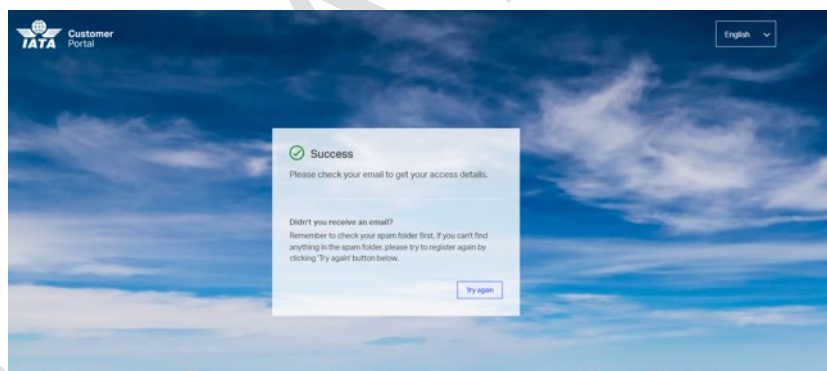
Spain

By clicking Next you confirm you have read and agree to the following terms:

[Privacy Policy](#) [Terms of Use](#)

[Back to the previous page](#) [Next](#)

b. Success screen notification of the registration:



IATA Customer Portal English

Success

Please check your email to get your access details.

Didn't you receive an email?
Remember to check your spam folder first. If you can't find anything in the spam folder please try to register again by clicking "Try again" button below.

[Try again](#)

c. Email confirming the registration:



We are happy to have you with us in our IATA Customer Portal!

The IATA Customer Portal is packed with useful services, product information, useful tips and tricks specially tailored for you. You can use it to contact our Customer Services Team easily, including live chat or directly purchase from our online store. It's your gate to IATA!


[Take me to the Portal!](#)

To enhance your IATA Customer Portal experience even more, you can complete your account information whenever you want to. You can easily do so by clicking [here](#).

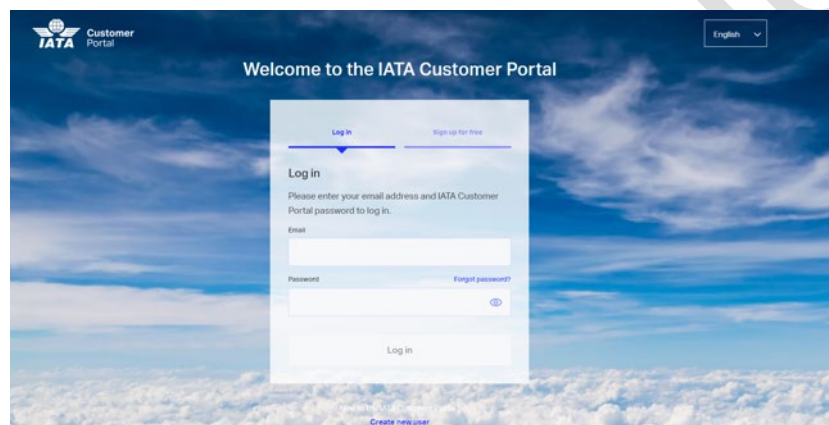
We hope you enjoy our IATA Customer Portal!

Thanks for being part of IATA.

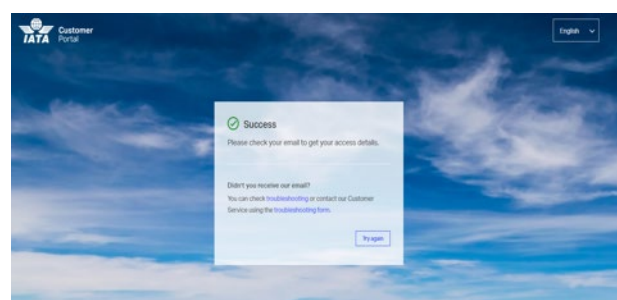
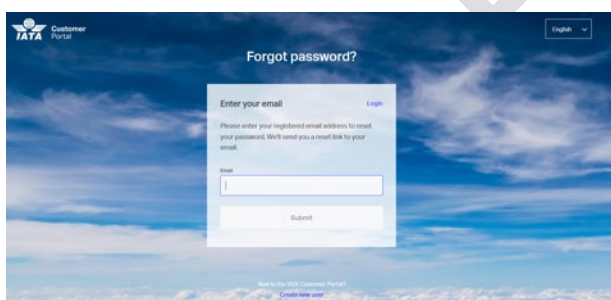
IATA

 Our mission is to represent, lead and serve the airline industry

d. Login screen:



e. In case of Forgotten Password, user(s) can reset the password:



We have received your request to reset your IATA Customer Portal password.

Please click [here](#) to set a new password.

[Set Password](#)

For security reasons, this link remains valid for 8 hours and can be used only once. After this time limits has expired, you can use the [forgot password](#) option to reset your access.

Thanks for being part of IATA.

IATA

 Our mission is to represent, lead and serve the airline industry

10.14 Testing Scenario IPTS-ADM-02. User can review payments processed

Objective:

Verify that the Airline users can view the payments processed in IATAPAY Admin Portal.

Preconditions:

Merchant user(s) involved in testing, and requiring access to view payments created in IATAPAY, should ensure that their profiles include the following attributes:

- The user(s) have been duly registered in the IATAPAY Admin Portal.
- The user(s) have been assigned the role of "Read Transactions" by the authorised merchants in the IATAPAY Admin Portal.

Test Data:

N/A

Steps to Reproduce:

Step 1: Log in to IATAPAY Admin Portal.

- Link to [IATA PAY Admin Portal](#).

Step 2: User can view "Payments" tab.

Step 3: User can click on "Payments" tab.

Expected Result:

The airline user can see "Payments" tab in the IATAPAY Admin Portal and search for payments processed on IATA PAY using different filtering criteria (iata payment id, status, etc.)





Pass/Fail Criteria:

When airline user clicks on “Payments” tab, will see a screen resembling the image below.

Hello,

Naiara Rodriguez
Gutierrez

Dashboard

Payments

Refunds

Compliance

Approvals

Merchants

Users

Audits

Outages

Monitoring

Status Update

Help

+ Sign out

IATA Pay

Payments

+ New

Multiple Refund

Export payments to CSV

Export transactions to CSV

Refresh

Status	ID	Amount	Country	Merchant	Creation Date	Finish Date	Clearance Date	Filters
<div></div>	P0BPP938DY09 Merchant payment id is not defined	₹1.00	IN	6E3120000	07/30/2024, 12:25 PM	07/31/2024, 12:00 AM		
<div></div>	PYRA6P0SGGKG Merchant payment id is not defined	₹1.00	IN	XX0000006	07/30/2024, 2:29 PM	07/30/2024, 2:29 PM		
<div></div>	P1YNURV85OG5ZL Merchant payment id is not defined	€1.00	ES	YY1280000	07/30/2024, 1:49 PM	07/30/2024, 2:09 PM		
<div></div>	PW5EVUN14N7L1 Merchant payment id is not defined	€1.00	ES	YY1280000	07/30/2024, 1:43 PM	07/30/2024, 2:02 PM		
<div></div>	PA4IF55JVMFH Merchant payment id is not defined	€1.00	ES	YY1280000	07/30/2024, 1:41 PM	07/30/2024, 1:41 PM		
<div></div>	P56AC075TAZ36 Merchant payment id is not defined	€1.00	ES	YY1280000	07/30/2024, 1:37 PM	07/30/2024, 1:37 PM		
<div></div>	P0730ZNSIPAX Merchant payment id is not defined	€1.00	ES	YY1280000	07/30/2024, 10:21 AM	07/30/2024, 10:21 AM		
<div></div>	P1A9CCLWUN7C Merchant payment id is not defined	€11.00	DE	AB1230002	07/29/2024, 10:27 AM	07/29/2024, 10:27 AM		
<div></div>	P443JB647TYIN JOWHYOMETOG8EEC3INO PNR is not defined	€1.01	DE	XX0000000	07/29/2024, 10:28 AM	07/29/2024, 10:29 AM		
<div></div>	PAUSGMJCEGNSQ Merchant payment id is not defined	JOD12:120	JO	JJ0000010	10/23/2023, 5:44 PM	10/23/2023, 5:45 PM		
<div></div>	PXKXWAB47GRCS Merchant payment id is not defined	JOD12:120	JO	JJ0000010	10/24/2023, 10:17 AM	10/24/2023, 10:22 AM		
<div></div>	PQ8MGOROWQNY1							

User will be able to check payment information details and events related to them.

Payment details

Details

Refunds

Events

Merchant

TPP Transaction Id

Merchant Payment Id

IATA Payment Id

Settlement Id

Amount

Country

Description

Payment Link

PNR

Departure Date

Status

Failure details

Compliance Done

Creation date

Finish date

Last update date

Clearance date

XX0000000

Not defined

Not defined

PX87KHGLW0KJD

₹5.00

IN

Locale

en-GB

Lambda Airlines

https://integration.iata-pay.iata.org/api/v1/payments/PX87KHGLW0KJD/checkout/redirect

Send this link to the customer to proceed with payment authorization.

Not defined

Not defined

●

 FAILED Cause

Tpp error

Transaction Failed

02/04/2025, 9:49 AM

02/04/2025, 9:49 AM

02/04/2025, 9:49 AM

-

Notes:

N/A

10.15 Testing Scenario IPTS-ADM-03. User can review refunds

Objective:

Verify that the user can view the refunds created in IATAPAY from the authorized merchants.

Preconditions:

Merchant user(s) involved in testing, and requiring access to view refunds created in IATAPAY, should ensure that their profiles include the following attributes:

- The user(s) have been duly registered in the IATAPAY Admin Portal.
- The user(s) have been assigned the role of "Read Transactions" by the authorised merchants in the IATAPAY Admin Portal.

Test Data:

N/A

Steps to Reproduce:

Step 1: Log in to IATAPAY Admin Portal.

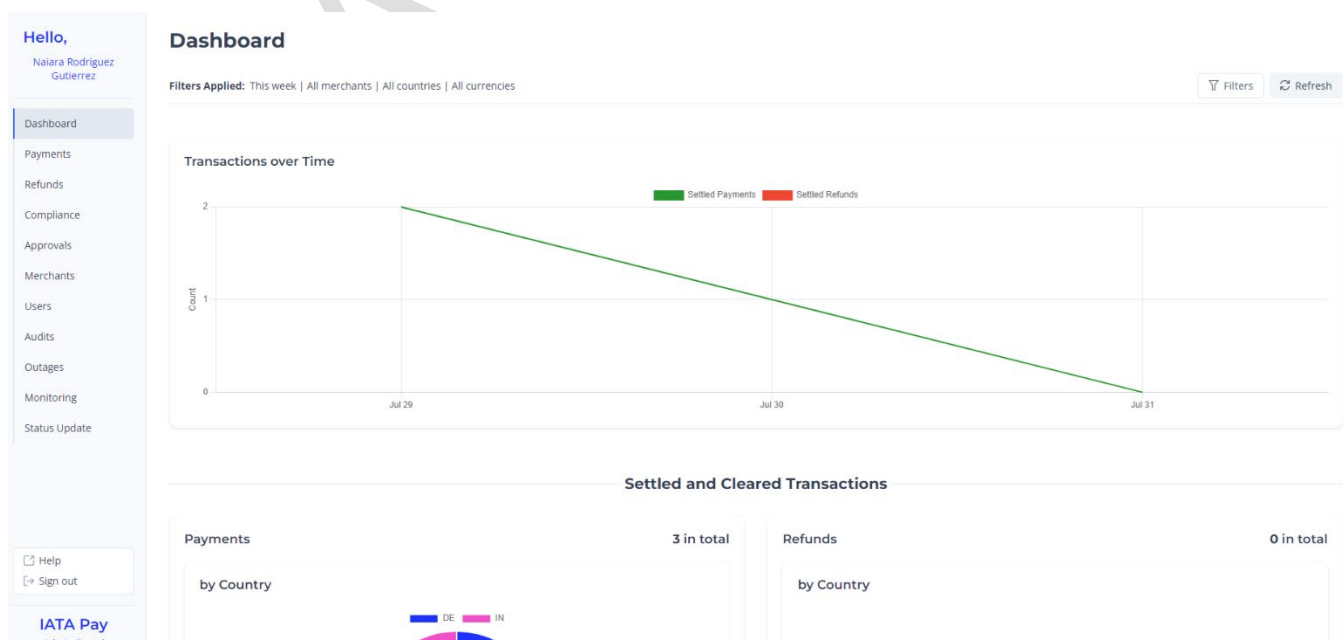
- Link to [IATA PAY Admin Portal](#).

Step 2: User can view "Refunds" tab.

Step 3: User can click on "Refunds" tab.

Expected Result:

The airline user can see "Refunds" tab in the IATAPAY Admin Portal and search for refunds processed on IATA PAY using different filtering criteria (iata refund id, status, etc.)





Pass/Fail Criteria:

When airline user click on “Refunds” tab, will see a screen resembling the image below.

Refunds

[Single Refund](#)[Multiple Refund](#)

Single

[Export refunds to CSV](#)[Refresh](#)

Status	Refund ID	Payment Status / ID	Amount	Country	Merchant	Creation Date ↑↓	Finish Date ↑↓	Clearance Date ↑↓	Filters
●	RYD4CTDMHMUZW Merchant refund id is not defined	● PJM2HN486GZ2R Merchant payment id is not defined	€2.00 €2.00	AT	RR1230000	12/03/2024, 12:28 PM	12/06/2024, 12:37 AM		⋮
●	R9PMDG9MLLKEX Merchant refund id is not defined	● P7ACSP099NXSB Merchant payment id is not defined	€6.25 €6.25	ES	XX0000004	12/04/2024, 9:07 AM			⋮
●	RVUKALIQ4SZGW Merchant refund id is not defined	● PR6JPJFRCT5HN Merchant payment id is not defined	€2.00 €2.00	AT	RR1230000	12/03/2024, 12:26 PM	12/03/2024, 12:36 PM		⋮
●	RTQ7PBT7JE592 Merchant refund id is not defined	● P47PD2I5Y4JBI AS1LZFHFWNDG2LT02LRP	€1.01 €1.01	ES	XX0000004	11/08/2024, 11:54 AM	12/05/2024, 1:18 PM		⋮
●	RMB72UJ2CWYNB Merchant refund id is not defined	● PX2GUFZQFUEPD Merchant payment id is not defined	THB2.00 THB4.00	TH	JJ0000000	11/04/2024, 11:33 AM	11/29/2024, 11:35 AM		⋮
●	RQ1B8VXMAI0ST Merchant refund id is not defined	● PK4ATL2Y31607 Merchant payment id is not defined	HK\$1.00 HK\$1.00	HK	AB1230000	11/13/2024, 10:38 AM	12/04/2024, 7:31 PM		⋮
●	RCPZ231C5G9W6 Merchant refund id is not defined	● POAFM14L3V5SI Merchant payment id is not defined	R\$1.00 R\$1.00	BR	XX0000000	11/05/2024, 2:27 PM	11/05/2024, 2:37 PM		⋮

User will be able to check refund information details and events related to them.

Payment details

[×](#)[Details](#) [Refunds](#) [Events](#)[+ New refund](#)

Merchant	XX0000000		
TPP Transaction Id	25000294155		
Merchant Payment Id	Not defined		
IATA Payment Id	PLW19JJNT50VX		
Settlement Id			
Amount	₹13.00		
Country	IN	Locale	en-GB
Description	Lambda Airlines		
PNR	Not defined		
Departure Date	Not defined		
Status	<div><div></div> SETTLED</div>		
Compliance Done	<div><div></div></div>		
Creation date	02/04/2025, 9:48 AM		
Finish date	02/04/2025, 9:48 AM		
Last update date	02/04/2025, 9:50 AM		
Clearance date	-		

Notes:

N/A



10.16 Testing Scenario IPTS-ADM-04. User can download CSV report for payments/refunds

Objective:

Ensure that user have the capability to download CSV reports appropriately filtered for authorized merchants. This report is provided to airlines to reconcile transactions with IATA PAY and can be provided in using different methods. This test scenario covers airline user access to IATA PAY Admin Portal to download report.

Preconditions:

Merchant user(s) involved in testing, and requiring access to download CSV files detailing payments and refunds created in IATAPAY, should ensure that their profiles include the following attributes:

- The user(s) have been duly registered in the IATAPAY Admin Portal.
- The user(s) have been assigned the role of "Read Transactions" by the authorised merchants in the IATAPAY Admin Portal.

Test Data:

N/A

Steps to Reproduce:

Step 1: Log in to IATAPAY Admin Portal. Link to [IATA PAY Admin Portal](#).

Step 2: User can view "Payments" and "Refunds" tabs.

Step 3: User can filter "Payments" and "Refunds" tabs.

Step 4: User can click on "Export payments to CSV" button and "Export transactions to CSV" button in Payments tab and "Export refunds to CSV" button in Refunds tab.

Expected Result:

The user can filter "Payments" tab in the IATAPAY Admin Portal:



Payments

[+ New](#) [Multiple Refund](#) [Export payments to CSV](#) [Export transactions to CSV](#) [Refresh](#)

Status	ID	Amount	Country	Merchant	Creation Date ↑↓	Finish Date ↑↓	Clearance Date ↑↓	Filters
All	ID	Amount	All	All	Range	Range	Range	
●	PM1UD8XDIXKMN IP82X0IA60YRJ29C7OZO PNR is not defined	COP1.00	CO	XX0000000	12/09/2024, 5:46 PM	12/09/2024, 6:03 PM		
●	PGV1IWB4UG87I VN6VID91O5CUU55JE53 PNR is not defined	COP1.00	CO	XX0000000	12/09/2024, 5:53 PM	12/09/2024, 5:55 PM		
●	PM4XNE35ICQXY ZW18PWR2ISX04PNJOBO PNR is not defined	COP1.00	CO	XX0000000	12/09/2024, 5:07 PM	12/09/2024, 5:07 PM		
●	P8VN0PEMBTPON Merchant payment id is not defined PNR is not defined	€1.00	DE	XX0000000	12/09/2024, 4:45 PM	12/09/2024, 4:45 PM		
●	PTMEB7GV9GTL6 UETUHPBZ6YEX4H33EK	₹1.01	IN	XX0000007	12/09/2024, 4:48 PM	12/09/2024, 4:48 PM		
●	PIQ758RXYAY4Q F6WQ029FA4QU4ITDJAN	₹1.01	IN	XX0000006	12/09/2024, 4:47 PM	12/09/2024, 4:47 PM		
●	PRLSZ5W5MC4JO Merchant payment id is not defined PNR is not defined	€1.00	DE	XX0000000	12/09/2024, 4:44 PM	12/09/2024, 4:44 PM		

The user can click on “Export payments to CSV” and “Export transactions to CSV” buttons and automatically retrieve a CSV file containing the details of payments filtered accordingly:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	iataTransactionId,merchantId,merchantTransactionId,creationDateTime,finishDateTime,clearanceDateTime,amount,currency,accountCountry,status,failureCode,failureDetails,type,PNR																					
2	PIGKHAEDRCE5,XX0000000,6WQICEMU04SBSDLG08IO,2024-02-28T10:10:33,2024-02-28T10:10:56,1.01,EUR,DE,SETTLED,,PAYMENT,																					
3	P38ENM2H75EC,XX0000000,GIUG9ILLDZE5566C97AH,2024-02-28T09:32:38,2024-02-28T09:36:18,1.01,EUR,DE,SETTLED,,PAYMENT,																					
4	P75XBOMR973G6,XX0000000,MED5OG1XVMASYBVVSZ4,2024-02-28T09:16:10,2024-02-28T09:17:30,1.01,EUR,DE,SETTLED,,PAYMENT,																					
5	P960TA9RXWW0,XX0000000,P9G2DC57E29V4E4L959,2024-02-23T00:16:28,2024-02-23T00:16:59,1.01,EUR,DE,SETTLED,,PAYMENT,																					
6	PAV8SMIJ2VW0N,XX0000000,43RJVVPY8PCYNLERXFKQ,2024-02-15T09:55:11,2024-02-15T09:55:42,1.01,EUR,DE,SETTLED,,PAYMENT,																					
7	PL7821X37U5XK,XX0000000,V8MXMN4A8BQJ1YDYBV,2024-02-09T11:48:04,2024-02-09T11:48:37,1.01,EUR,DE,SETTLED,,PAYMENT,																					
8	PWKBRBIC8GQKM,XX0000000,I2LEU5LVDVP05DKHB858,2024-02-08T09:38:07,2024-02-08T09:39:33,1.01,EUR,DE,SETTLED,,PAYMENT,																					
9	PN2JFBR190AJ9,XX0000000,,2024-02-06T11:30:38,2024-02-06T11:32:47,12.00,EUR,DE,SETTLED,,PAYMENT,																					
10	PN5K6X04R69JW,XX0000000,,2024-02-05T16:32:11,2024-02-05T16:35:20,12.00,EUR,DE,SETTLED,,PAYMENT,																					
11	PKF8IKO2FMFC3,XX0000000,DC6DQSNCPWBWMOZIR8PX,2024-02-02T01:36:26,2024-02-02T01:37:23,1.01,EUR,DE,SETTLED,,PAYMENT,																					
12	PJVRPXPSETCNE,XX0000000,8ZVQV0F53IFTYUUL7SQV,2024-02-02T01:33:39,2024-02-02T01:34:18,1.01,EUR,DE,SETTLED,,PAYMENT,																					
13	PQ5ASA6M3IP4F,XX0000000,B293BAF1FFNQKLBFOF2B,2024-01-31T09:30:41,2024-01-31T09:31:37,1.01,EUR,DE,SETTLED,,PAYMENT,																					
14	P072RTAN51MWK,XX0000000,,2024-01-24T17:12:42,2024-01-24T17:17:19,10.00,EUR,DE,SETTLED,,PAYMENT,																					
15	PDICQSSZJRY16,XX0000000,,2024-01-24T15:49:14,2024-01-24T15:50:40,4.00,EUR,DE,SETTLED,,PAYMENT,																					
16	P6KR4L38NWWY6,XX0000000,,2024-01-24T15:43:22,2024-01-24T15:48:37,4.00,EUR,DE,SETTLED,,PAYMENT,																					
17	PGBKFUEQD5F0C,XX0000000,Q057HMFJCOZZEL9H2NWO,2024-01-18T06:25:03,2024-01-18T06:25:28,1.01,EUR,DE,SETTLED,,PAYMENT,																					
18	PQW1EWZSR7G3,XX0000000,2PRMP2B9B8T2Y408MO30,2024-01-18T05:49:41,2024-01-18T05:52:27,1.01,EUR,DE,SETTLED,,PAYMENT,																					
19	PAWHDDVD5RIFU,XX0000000,0XMKVDSKAIY7HZK09WZJ,2024-01-17T02:31:18,2024-01-17T02:32:29,1.01,EUR,DE,SETTLED,,PAYMENT,																					
20	PB3E6NRSWDWC,XX0000000,ZCISQYSSR8PPYG2TMM6X,2024-01-16T17:27:31,2024-01-16T17:28:08,1.01,EUR,DE,SETTLED,,PAYMENT,																					
21	PNTRJ4WP9YRFS,XX0000000,,2024-01-16T10:19:13,2024-01-16T10:19:52,3.00,EUR,DE,SETTLED,,PAYMENT,																					
22	P1TAUIY911PP4,XX0000000,VRGE16R19EC62XNOKKRO,2023-11-23T12:17:56,2023-11-23T12:19:04,1.01,EUR,DE,SETTLED,,PAYMENT,																					
23	PV5GR8E1XOFKS,XX0000000,6N9DMF9N3BDBVTLKAF,2023-11-21T16:35:39,2023-11-21T16:36:24,1.01,EUR,DE,SETTLED,,PAYMENT,																					
24	PNW18LIFYFLX1,XX0000000,NIIUG3ORYZHWUNYX2QJ,2023-11-17T12:10:30,2023-11-17T12:11:23,1.01,EUR,DE,SETTLED,,PAYMENT,																					
25	P52RBNQGDQHQ,XX0000000,UQMC2RGORIRUUVZU82MG,2023-11-08T11:58:50,2023-11-08T11:59:26,1.01,EUR,DE,SETTLED,,PAYMENT,																					
26	P72D9I9R9E0IG,XX0000000,KHDO4JOKVWY53W7E6KBA,2023-11-02T18:23:03,2023-11-02T18:23:40,1.01,EUR,DE,SETTLED,,PAYMENT,																					
27	PNV1JBG4YEFX,XX0000000,0Y3KIBGG75Z0BYBQ08IRW,2023-10-30T14:45:59,2023-10-30T14:46:31,1.01,EUR,DE,SETTLED,,PAYMENT,																					
28	PNWX85HIMZAH3C,XX0000000,8YKAMARJ2ERJOP9S23MX,2023-10-30T09:28:17,2023-10-30T09:28:55,1.01,EUR,DE,SETTLED,,PAYMENT,																					

Also, airline user can filter “Refunds” tab in the IATAPAY Admin Portal:



Refunds

[Single Refund](#) [Multiple Refund](#)

Single

[Export refunds to CSV](#) [Refresh](#)

Status	Refund ID	Payment Status / ID	Amount	Country	Merchant	Creation Date ↑↓	Finish Date ↑↓	Clearance Date ↑↓	Filters
All	Refund ID	Payment ID	Amount	All	All	Ra...	Ra...	Ra...	Q
●	RYD4CTDMHMMUZW Merchant refund id is not defined	● PJM2HN486GZ2R Merchant payment id is not defined	€2.00 €2.00	AT	RR1230000	12/03/2024, 12:28 PM	12/06/2024, 12:37 AM		⋮
●	R9PMDG9MLLKEX Merchant refund id is not defined	● P7ACSP099NXSB Merchant payment id is not defined	€6.25 €6.25	ES	XX0000004	12/04/2024, 9:07 AM			⋮
●	RVUKALIQ4SZGW Merchant refund id is not defined	● PR6JPJFRCT5HN Merchant payment id is not defined	€2.00 €2.00	AT	RR1230000	12/03/2024, 12:26 PM	12/03/2024, 12:36 PM		⋮
●	RTQ7PBT7JE592 Merchant refund id is not defined	● P47PD2I5Y4JBI A51LZFHFWNDG2LT02LRP	€1.01 €1.01	ES	XX0000004	11/08/2024, 11:54 AM	12/05/2024, 1:18 PM		⋮
●	RMB72UJ2CWYNB Merchant refund id is not defined	● PX2GUFZQFUEPD Merchant payment id is not defined	THB2.00 THB4.00	TH	JJ0000000	11/04/2024, 11:33 AM	11/29/2024, 11:35 AM		⋮
●	RQ1B8VXMAIOST Merchant refund id is not defined	● PK4ATL2Y31607 Merchant payment id is not defined	HK\$1.00 HK\$1.00	HK	AB1230000	11/13/2024, 10:38 AM	12/04/2024, 7:31 PM		⋮
●	RCPZ231C5G9W6	● POAFM14L3V5SI	R\$1.00			11/05/2024,	11/05/2024,		⋮

The user can click on “Export refunds to CSV” button and automatically retrieve a CSV file containing the details of refunds filtered accordingly:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	IataTransactionId	merchantId	merchantTransactionId	creationDateTime	finishDateTime	clearanceDateTime	amount	currency	accountCountry	status	failureCode	failureDetails	type	PNR	original payment	IATAPAYID						
2	RQNGOMULTY79	XX0000000	2023-09-21T15:52:51	2023-09-21T15:56:13	-0.89	EUR	DE	SETTLED	REFUND	PPMCRA4CVLR59												
3	RMPWWZGAATVG	XX0000000	2023-09-14T15:26:54	2023-09-14T15:31:14	-1.00	EUR	DE	SETTLED	REFUND	PYBQMIL29X2Y1												
4	R20J76CMBRI6S	XX0000000	2023-07-24T19:41:03	2023-07-24T19:42:14	-50.00	EUR	DE	SETTLED	REFUND	PIKC3455A10OZ												
5	RBG25JWL7HNQG	XX0000000	2023-06-20T09:19:17	2023-06-20T09:21:17	-1.00	EUR	DE	SETTLED	REFUND	P943JUE6PICVG												
6	RWB2NZU7ZLOP8	XX0000000	2023-06-19T08:37:05	2023-06-19T08:38:15	-0.89	EUR	DE	SETTLED	REFUND	PCY4L4840TYSR												
7	RUQD04YE92DC1	XX0000000	2023-06-19T07:55:03	2023-06-19T07:57:17	-0.89	EUR	DE	SETTLED	REFUND	PWODTIFSFQDZ												
8	RQABTNLOE2EGX	XX0000000	2023-06-14T08:36:39	2023-06-14T08:37:16	-0.89	EUR	DE	SETTLED	REFUND	P782JHF9SKUIQ												
9	R0554ZQET4USC	XX0000000	2023-06-05T15:52:01	2023-06-05T15:54:13	-0.10	EUR	DE	SETTLED	REFUND	P518816YL158I												
10	R2H70UXX88A6I	XX0000000	2023-06-05T15:43:02	2023-06-05T15:46:13	-0.89	EUR	DE	SETTLED	REFUND	P518816YL158I												
11	RSFGCJZVBQ2Q	XX0000000	2023-05-31T17:07:24	2023-05-31T17:08:13	-1.01	EUR	DE	SETTLED	REFUND	PAGA2XC9B3AF8												
12	R8KFKYD7CB49Z	XX0000000	2023-05-25T11:31:10	2023-05-25T11:33:13	-0.89	EUR	DE	SETTLED	REFUND	PMP369XKTMU34												
13	RKNQIP6IK1F2	XX0000000	2023-04-26T04:00:32	2023-04-26T04:01:12	-0.89	EUR	DE	SETTLED	REFUND	P5D002Y9AHR3W												
14	RDCDF0OH23IUQ	XX0000000	2023-04-21T07:12:38	2023-04-21T07:13:12	-0.89	EUR	DE	SETTLED	REFUND	P5Z0WQ0QXFS213												
15	RRGAJ2Z0GEYSO	XX0000000	2022-05-23T09:23:38	2023-04-19T13:25:02	-0.12	EUR	DE	SETTLED	REFUND	P7Z4SFIUW2TTI												
16	RAXS0ZSKX1KYV	XX0000000	2023-04-11T09:34:13	2023-04-11T09:35:15	-0.89	EUR	DE	SETTLED	REFUND	PFSK9E96EUIINT												
17	RXJL30LIK1LS8	XX0000000	2023-03-13T13:14:05	2023-03-13T13:15:13	-0.50	EUR	DE	SETTLED	REFUND	P0491JTCTRCQA												
18	R3UGWP9VYH1JQ	XX0000000	2023-03-13T13:11:27	2023-03-13T13:12:13	-0.89	EUR	DE	SETTLED	REFUND	P0491JTCTRCQA												
19	RKZ5X8H07D9QJ	XX0000000	2023-03-09T09:01:57	2023-03-09T11:13:26	-0.89	EUR	DE	SETTLED	REFUND	PEOGC64MIZAFF												
20	R190VXL2C3GIV	XX0000000	2023-03-07T10:24:17	2023-03-09T11:13:26	-0.50	EUR	DE	SETTLED	REFUND	PDJ1GDB99ZUGZ												
21	R6TD9VZDL720N	XX0000000	2023-02-22T11:49:57	2023-02-22T11:50:13	-0.50	EUR	DE	SETTLED	REFUND	PBP3AUESJC3EO												
22	R4564JSV96KV3	XX0000000	2023-02-22T11:41:22	2023-02-22T11:42:14	-0.89	EUR	DE	SETTLED	REFUND	PBP3AUESJC3EO												
23	RJQZ33XZDHHG	XX0000000	2023-02-08T14:21:30	2023-02-08T14:22:12	-0.50	EUR	DE	SETTLED	REFUND	POI9S3L7HBVG9												
24	RWBDX80SUGPYR	XX0000000	2023-02-08T08:20:06	2023-02-08T08:21:12	-1.89	EUR	DE	SETTLED	REFUND	P5G6LELNNX9T												
25	RE58TOKGPPYZG	XX0000000	2023-02-01T12:46:21	2023-02-01T12:47:13	-0.89	EUR	DE	SETTLED	REFUND	PIIGEAQCTREER												
26	RNODVWUJ0TTCK	XX0000000	2023-01-30T12:39:22	2023-01-30T12:41:16	-0.50	EUR	DE	SETTLED	REFUND	P6KAKI9G21V2R												
27	R7MP3ADTEDWEP	XX0000000	2023-01-25T12:42:01	2023-01-25T12:43:13	-0.89	EUR	DE	SETTLED	REFUND	PRDE0E1RSS0YB												
28	RF4TD4AR399M5	XX0000000	2023-01-25T12:42:01	2023-01-25T12:43:13	-0.89	EUR	DE	SETTLED	REFUND	PRDE0E1RSS0YB												

Pass/Fail Criteria:

The test will be regarded as successful if the information filtered within the IATAPAY Admin Portal matches that exported in the CSV file.

Notes:

N/A