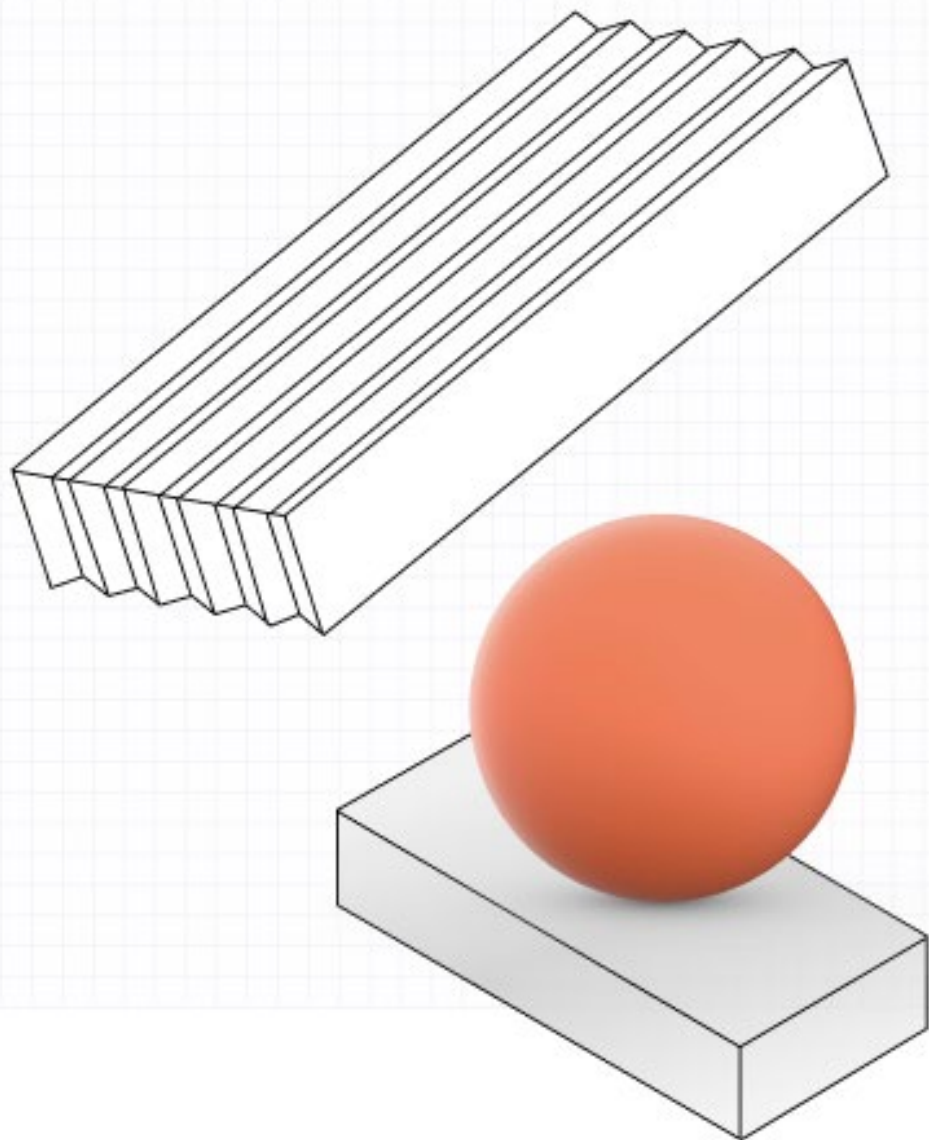


Source Gateway Services

Merchant Management API

API Specifications



API Version 1.3

July 2023

Credorax Bank Ltd is a credit institution established in Malta, operating under the trading name Finaro. Registered address Palazzo Homedes, 80 Strait St. Valletta VLT 1436, Malta. Number of registration C46342. Credorax Bank Ltd. is authorised and regulated by the Malta Financial Services Authority and the UK Prudential Regulatory Authority and Financial Conduct Authority with reference number: 710757. LEI code: 213800191YORJFDJF94.



Contents

Introduction.....	3
Gateway Interface.....	4
Implementing Merchant Management API	6
General API Flows.....	7
Business Flows.....	8
Message Format.....	10
Server-2-Server Notification	11
Requests and Responses.....	12
Create Merchant	13
Update merchant.....	36
Upload document	50
Get merchant status.....	53
Get Connectivity details	54
Appendix A: HMAC-SHA512 Request Signature.....	57
Appendix B: Products.....	59
Appendix C: Result Codes	61
Appendix D: Related Corporate and Merchant Individual Scenarios	62
Appendix E: Opeartion codes.....	64
Appendix F: Onboarding Documents.....	66
Change History.....	68
Need Support?	69



Introduction

The purpose of this document is to provide an in-depth description of the Finaro Merchant Management API. Finaro Merchant Management API is a proprietary platform for partners to onboard and manage merchants

The Merchant Management API is a simple-to-use RESTful API. The API operates as a basic request-response service where the client instructs the gateway to perform an operation and the platform replies with the request's acknowledgement. Once the merchant creation or update process is completed, the partner receives a notification with the merchant creation status.

Audience

The intended audience includes Finaro partners (PFs, PSPs) who wish to onboard their merchants via API.

API Version Control

The information provided in this document is accurate and reliable for standard processing as of its publication date. Any new implementations should thus avoid using previous versions of the API specification.

The API version number is a sequence-based identifier. Changes to the first part indicate major specification updates, while changes to the second part indicate minor updates.

The revision number reflects smaller specification changes, the correction of typing errors, or corrections that do not affect the API protocol itself.

Certification

All new implementations must go through a certification process in order to ensure the quality of their integrations and the integrity of merchant data.

An additional certification process will be required if new features are introduced.



Gateway Interface

Introduction

Merchant management requests are sent online and in real-time using the HTTPS (Hypertext Transfer Protocol - Secure) protocol. The platform protocol, in turn, exposes multiple request types, including Create, Update, Get, and Upload.

Uniform Resource Locator (URL) Addresses

Integration URL	<code>https://onboarding.int.sourcepayments.com/rest/setup/merchant/<action></code>
Production URL	<code>https://onboarding.sourcepayments.com/rest/setup/merchant/<action></code>

Security/Authentication

All HTTP requests must be sent over a secure TLS (Transport Layer Security) 2.0 channel. The merchant management platform does not authenticate the TLS/SSL (Secure Sockets Layer) session using a client-based certificate, and thus does not employ a regular type of session authentication. Instead, the client is first authenticated by its source IP alongside a secondary authentication check that employs a cipher sent in the request header and used for pre-processing verifications. See [Appendix A: HMAC-SHA512 Request Signature](#) for further details.

Health Checks

The health of the *Source* Merchant Management production and integration environments can be checked by accessing the following URLs:

Production Environment	<code>https://onboarding.sourcepayments.com/health</code>
Integration Environment	<code>https://onboarding.int.sourcepayments.com/health</code>

The service will then respond with a JavaScript Object Notation (JSON) message.

One of the following responses will be provided:

- `{"health":"OK"}`
- `{"health":"FAIL"}`



Follow these guidelines when using health checks:

- A **maximum** of one health check is permitted every 10 seconds
- If no response is received within 20 seconds, consider the request timed out
- The *Source* Merchant Management service should be considered unavailable after 3 consecutive service failures
- In any event of unexpected service interruption contact the Finaro Support Team immediately support@finaro.com.

Timeout Handling

If a request takes too long to return a response, a timeout is initiated by the Gateway application and a result code "007" is returned. If many timeout result codes are received, please contact our support at support@finaro.com.



Implementing Merchant Management API

As part of the integration process for Finaro's Merchant management API, the partner will go through the following steps:

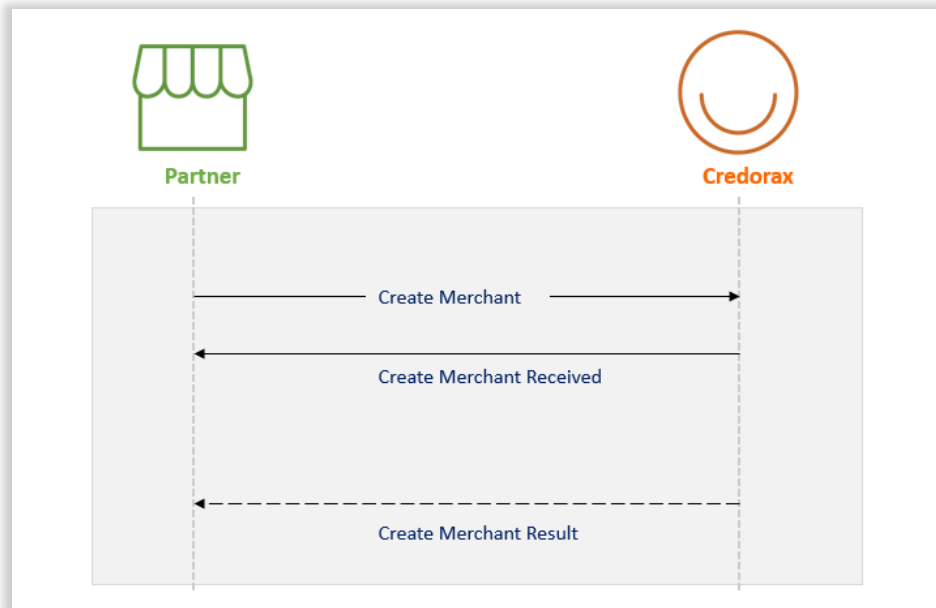
1. **Sign-up for the service:** contact your Finaro Solution Architect and request to start using the Merchant Management API service
2. **API review and preparation:**
 - a. Review the API specifications listed in this document and plan your development process
 - b. Whitelist IPs and URLs required for the service
3. **Integration & certification:**
 - a. You will receive credentials for the integration environment (a partner ID and a secret key)
 - b. Develop and test the API in integration environment
 - c. Run through the certification process to ensure everything works before you go live
4. **Go-live:** You will receive credentials for production environment, and you are ready to go live!



General API Flows

The Merchant Management API supports two types of flows: a 3-legged flow and a 2-legged flow

3-Legged Flows



Example 1 - 3-Legged Flow

1. The partner sends a request to the Credorax merchant management platform.
2. Credorax instantly responds with an acknowledgment that the request was received successfully. In case the request was not received successfully in this step (due to validation errors such as incorrect format in parameters, API structure etc.) the partner will receive a sync detailed response.
3. Once the request handling process is complete, the partner receives a notification with the request status. The notification is sent from Credorax notification engine. For more details about the Credorax notification engine please refer to the Data Open API Specifications guide on the Credorax Developer Portal: <https://epower.credorax.com>

2-Legged Flows

1. The partner sends a request to the Credorax merchant management platform.
2. Credorax instantly responds with a synchronous response with the information requested



Business Flows

The *Merchant Management API* supports the following business flows:

Main Business Flows

Create Merchant

A Create Merchant request creates a new merchant on Finaro's system. The merchant is automatically related to the requesting partner

Update Merchant

The Update Merchant request enables you to request an update of an existing merchant's details and settings, as described below:

- Update merchant details
- Add / cancel a merchant bank account (to which funds from processing are transferred)
- Add / cancel / update a merchant contact
- Add / Update a merchant individual
- Add / deactivate a product
- Update product configuration
- Update merchant pricing
- Add / cancel a merchant requirement
- Update the number of Card Present (POS) devices
- Update the address of Card Present (POS) devices
- Update the address of a related corporate

Upload Document

The Upload Document request enables a partner to send relevant documents related to the merchant onboarding or update process.

Get Merchant Status

The Get Merchant Status request retrieves the merchant onboarding status.

Get Connectivity Details



The Get Connectivity Details request retrieves the merchant connectivity details credentials to start processing after setup is completed. This API call is available only when the onboarding and setup process for the merchant is complete.

Additional Flows

Pricing Management

Pricing is managed during merchant creation and merchant update

When Creating a Merchant

When creating a merchant you must assign a pricing to the merchant. The basic pricing assignment is based on pre-defined pricing templates set up for your partner account (sometimes referred to as “fixed pricing” or “partner terms”).

1. Using the `partner_terms_id` define which fixed pricing template should be used for this merchant
2. Assign a unique pricing ID for the merchant you are creating, using the `merchant_terms_id`
3. Override the fixed terms with unique ones using the pricing functionality

When Updating a Merchant

You can make pricing updates using the pricing object in a merchant update flow. Refer to the [Update Merchant](#) chapter for more details.



Message Format

General

The Merchant Management API is a JSON-formatted REST API.

Note that the Merchant Management API structure and fields might be updated from time to time without prior notice. Such changes may include the addition of new objects, fields and values.

Headers

Each request should contain the following headers:

- Target host.
Host: onboarding.sourcepayments.com
- Content type – Payload content type, for POST requests.
Content-Type: application/json
- Content length – Payload length, for POST requests.
Example: Content-Length: 123
- Date – date in RFC-1123 format.
Example: Date: Tue, 8 Feb 2022 04:59:40 GMT
- Authorization header: contains the hashed string that signs the request. See [Appendix A: HMAC-SHA512 Request Signature](#) for more information.

Partner ID example: CEID-0000200332

Example

Host: onboarding.sourcepayments.com

Content-Type: application/json

Content-Length: 123

Date: Tue, 8 Feb 2022 04:59:40 GMT

Authorization: HMAC-SHA512 Credential= partnerID,

Signature=f8fa88335f76d3d2673575e3cb4e2c6bfcaf58869074fd9f42cbcd47a3be487626742eae68644e8acce565276392bf236353593f1c75926dce98a89d4ccc9ce4

Body

Valid JSON message. For more details about the possible objects and fields in a message see the [Requests & Responses chapter](#).



Server-2-Server Notification

General

The response from Finaro regarding merchant creation or merchant update is received after the request was successfully saved by Finaro's system. Finaro initially returns an acknowledgement with response code 099, indicating that the request was received and is being processed. The full and final response will be returned in a separate notification message, see [Appendix C: Result Codes](#)

The *Source* gateway expects to receive a response of "200: OK" status code (without body) in response to the notification it sends.

NOTE: To setup the notification service contact your Finaro Solution Architect to set up the notification URL.

Notification Format

The notification format is based on the response format of each request type (Create Merchant, Update Merchant, and Upload Document).

IP Allowlist

In order to accept notifications allow the following IP on the partner side:

PROD: 199.233.202.161, 199.233.203.161

INT: 52.209.227.163



Requests and Responses

The following chapters describe the required parameters for each type of API request flow.

The tables describe the API parameters and specify whether the field is mandatory (M), conditional (C), optional (O), or not used (-), according to the partner type (PSP or Payment Facilitator).



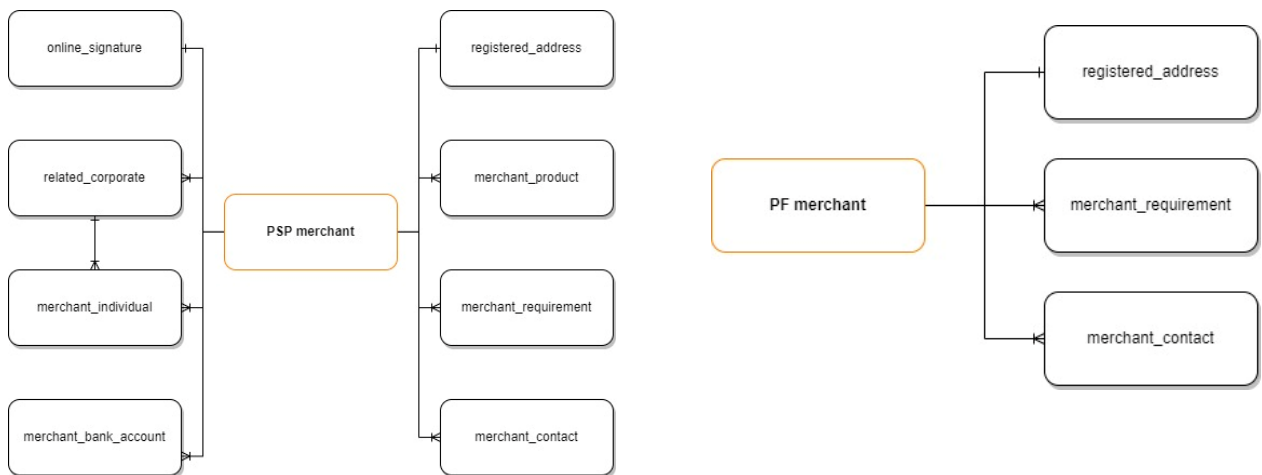
Create Merchant

PROD: [POST] <https://onboarding.sourcepayments.com/setup/merchant/create>

INT: <https://onboarding.int.sourcepayments.com/setup/merchant/create>

Request fields

The required request fields depend on the type of partner assigned to you by Finaro, PSP or PF (Payment Facilitator).



Root level fields and objects

The following table lists the root level fields and objects. The root level fields are listed in alphabetical order.

Parameter Name	Type	Length (min,max)	Description	PSP	PF
average_item_value_eur	[0-9]	(1,9)	The average value of an item sold by the merchant, in Euro (a number).	M	O
business_model_description	[A-Z a-z 0-9 - _():&/*<=?%,+]	(1, 32000)	A description of the merchant's activity. Can contain letters, numbers, or symbols	M	M
business_type	(corporate sole_trader)	(9,11)	Specifies whether the merchant's business is a corporation or a sole_trader	M	-
company_registration_name	[A-Za-z0-9.< (+ & *);-,%_>?:' =]	(1,50)	The official company registration name of the merchant's business. This field is mandatory when business_type = corporate	C	O



Parameter Name	Type	Length (min,max)	Description	PSP	PF
company_registration_number	[A-Z a-z 0-9]	(1,255)	The official company registration number of the merchant's business. This field is mandatory when business_type = corporate	C	O
country_of_incorporation	[A-Z]	(3,3)	The country where the merchant is incorporated, ISO alpha 3 Code. This field is mandatory when business_type = corporate.	C	O
date_of_incorporation	yyyy-mm-dd	(10,10)	The date when the merchant's business was incorporated. This field is mandatory when business_type = corporate For example: 2014-10-20	C	O
expected_monthly_turnover	[0-9]	(1,9)	The merchant's expected monthly turnover in Euro. A decimal number with 2 decimal places For example: 100.50	M	M
merchant_name	[A-Z a-z 0-9 - _():&/*<=?%+,]	(1,120)	The merchant's name Can contain letters, numbers, or symbols	M	M
merchant_bank_account	Object		A list object that can be sent more than once. It holds the merchant bank account details (also referred to as settlement bank account) This object includes a nested beneficiary address object	C	-
merchant_contact	Object		A list object that holds the merchant contact details. Send this object for at least one Primary Contact	M	M
merchant_individual	Object		A list object that holds the merchant individual details. This object needs to be sent at least once. <ul style="list-style-type: none"> Send this object for an Authorised Signatory of the merchant (= the individual who will sign the contract) Send this object for every shareholder who is an individual with over 25% holding in a corporate. Send this object for every director in the corporate 	M	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
merchant_product	Object		<p>A list object that holds the merchant_product details</p> <p>Send this object for every product the merchant needs to have in Finaro (accordint to the Partner products)</p> <p>This object includes a nested product_configuration object</p>	M	-
pricing	Object		<p>A list object that holds the merchant pricing details. Send this object at least once, for creating a merchant pricing from the partner's fixed pricing.</p> <p>This object includes a nested optional fee_list object.</p>	M	-
merchant_requirement	Object		<p>A list object that holds the merchant_requirement details (also referred to as payment channel).</p> <p>Send this object for every MCC, site (URL), and store of the merchant.</p> <p>This object includes a nested pricing object; and a nested pos_device object if the payment channel is CP</p>	M	M
online_signature	Object		Object that holds the online_signature details	M	-
registered_address	Object		Object that holds the registered_address details	M	M
related_corporate	Object		<p>Object that holds the related_corporate details. mandatory when business_type = corporate (in case of sole trader related_corporate object is not required)</p> <p>Send this object at least once for the onboarded merchant itself, and if relevant, then also for:</p> <ul style="list-style-type: none"> • Every merchant shareholder or director which is a corporate <p>See Appendix D: Related Corporate and Merchant Individual Scenarios.</p>	C	-

merchant bank account object

PSPs onboarding merchants must send at least one active bank account per merchant. This is where the funds from the merchant's processing activity are transferred to.

PFs are not require to send this object



Parameter Name	Type	Length (min,max)	Description	PSP	PF
account_number	[A-Z 0-9]	(1,18)	The merchant's bank account number. Either the IBAN or the account number should be sent.	C	-
active	[true false]	(4,5)	Specifies whether the merchant's bank account is currently active. In merchant create send 'true'	M	-
bank_country	[A-Z]	(3,3)	The country in which the merchant's bank is located. ISO alpha 3 Code. For example: GBR	M	-
bank_name	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(1,70)	The name of the merchant's bank.	M	-
beneficiary_address	object		Object that holds the beneficiary address details	O	-
beneficiary_name	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(1,50)	The name of the beneficiary of the merchant's bank account.	M	-
iban	[A-Z 0-9]	(1,40)	The IBAN number of the merchant's bank account, should be according to bank_country, Capital letters and numbers only. Either the IBAN or the account number should be sent	C	-
partner_terms_identifier	[A-Z a-z 0-9 -]	(9, 10)	The ID of the fixed merchant pricing terms you are using to assign pricing to this merchant. The ID always begins with a leading 'MPS-' and continues with max 6 digits. For example: MPS-660832 NOTE: you receive these IDs as part of your connectivity details before go-live	M	-
routing_number	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(1,9)	The routing number of the merchant's bank. Either the SWIFT code or the routing number should be sent.	C	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
settlement_currency	[A-Z]	(3,3)	The currency in which funds are deposited in the merchant's bank account. ISO alpha 3 Code For example: GBP	M	-
sort_code	[A-Z a-z 0-9 - =':?_%(),&+.<>/*]	(1,24)	The sort code of the merchant's bank account	O	-
swift	[A-Z 0-9]	(1,11)	The SWIFT code of the merchant's bank. Either the SWIFT code or the routing number should be sent.	C	-
use_merchant_address	[true false]	(4,5)	Specifies whether to use the merchant's address for the bank account's beneficiary address . If you set this parameter to "true", do not send a beneficiary address object. The beneficiary address details will be taken from the address fields of the merchant individual object.	M	-

beneficiary address object

This object needs to be sent only if beneficiary's bank account address is different than the merchant address

Parameter Name	Type	Length (min,max)	Description	PSP	PF
address_line_1	[A-Z a-z 0-9 - =':?_%(),&+.<>/*]	(1,255)	Beneficiary's street name	M	-
address_line_2	[A-Z a-z 0-9 - =':?_%(),&+.<>/*]	(1,255)	Beneficiary's street number	M	-
city	[A-Z a-z 0-9 - =':?_%(),&+.<>/*]	(1,80)	Beneficiary's city	M	-
country	[A-Z]	(3,3)	Beneficiary's country, ISO alpha 3 Code For example: GBR	M	-
postcode	[A-Z a-z 0-9 - =':?_%(),&+.<>/*]	(1,80)	Beneficiary's postal code	M	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
state	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,5)	Beneficiary's state	O	-

merchant contact object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
active	[true false]	(4,5)	Specifies whether the merchant's contact person is currently active. For contacts in merchant creation the value is 'true'	M	M
email	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,80)	Email of the merchant's contact person	M	M
first_name	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,40)	First name of the merchant's contact person	M	M
last_name	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,80)	Last name of the merchant's contact person	M	M
mobile_phone	[0-9]	(1,40)	Mobile phone number of the merchant's contact person	O	M
phone	[0-9]	(1,40)	Landline phone number of the merchant's contact person	M	M
role	Values from the list: <ul style="list-style-type: none"> • primary_contact • cbk_contact • general • insights 	(7,15)	This field specifies the Role of the merchant's contact person. <ul style="list-style-type: none"> • at least one contact who is a primary_contact must be sent (there canbe only one) • If the contact has several roles, enter them as a list separated by semicolons (";") • if role=general, no other roles can be indicated for the contact 	M	O

merchant individual object

PSPs must send this object at least once for either corporate or sole trader merchant types.

For a corporate type of merchant, send the details of the following individuals:



- An individual who is an **authorized signatory** of the merchant (the individual who will sign the contract); and
- For every **shareholder** with over 25% holding in a corporate; and
- For every **director** in the corporate.

For a sole trader type of merchant send at least one individual

Parameter Name	Type	Length (min,max)	Description	PSP	PF
active	[true false]	(4,5)	Specifies whether the individual is currently active For merchant creation individuals the value is 'true'	M	-
address_line_1	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,255)	Individual's street name	O	-
city	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,80)	Individual's city of residence	M	-
country	[A-Z]	(3,3)	Individual's country of residence ISO alpha 3 Code. For example: GBR	M	-
date_of_birth	YYYY-MM-DD	(10,10)	Individual's date of birth For example: 1988-04-28	M	-
first_name	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,25)	Individual's first name	M	-
holding_percent	[0-9.] Decimal: 3 digits + 2 decimal places	(1,6)	Specifies the Individual's percentage of ownership of the corporation. Mandatory if individual_type = 'UBO_ownership'	C	-
house_number	[0-9]	(1,9)	Individual's house number	O	-
individual_type	Value from the list: <ul style="list-style-type: none"> • director • UBO_ownership • authorised_signatory • sole_trader 	(8,20)	The individual's type. If the individual is of several types, enter them as a list separated by semicolons (";"). For example: shareholder;director	M	-
last_name	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,80)	Individual's last name	M	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
nationality	[A-Z]	(3,3)	Individual's nationality. ISO alpha 3 Code. For example: GBR	M	-
official_id_expiry_date	YYYY-MM-DD	(10,10)	The expiry date of the official ID used by the individual. For example: 2030-03-24	O	-
official_id_number	[A-Z a-z 0-9 -]	(1,50)	The ID number/passport of the official ID used by the individual	M	-
official_id_type	Value from the list: <ul style="list-style-type: none"> • Passport • ID Card • Driver's license • Residence permit • Asylum seekers document • Other document 		The type of official ID used by the individual	M	-
passport_place_of_issue	[A-Z a-z 0-9 - =':?_%,()&+. <> ; / *]	(1,255)	The place of issue of the Individual's passport Mandatory if official_id_type = 'Passport'	C	-
place_of_birth	[A-Z]	(3,3)	The Individual's place of birth ISO alpha 3 Code. For example: GBR	M	-
postcode	[A-Z a-z 0-9 - =':?_%,()&+. <> ; / *]	(1,11)	The individual's postal code	M	-
related_corporate	[A-Z a-z 0-9 -]	(1,80)	Specify here the company_registration_number of the corporate this individual is related to. The value in this field should match the company_registration_number in related_corporate object. If the type is corporate you must send the related corporate id. If the type is sole trader this field is optional.	C	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
state	[A-Z a-z 0-9 - =':?_%,()&+.<> /*]	(0,5)	The individual's state	O	-

merchant product object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
active	[true false]	(4,5)	Specifies whether the product needs to be setup as active for the merchant. In merchant create send 'true'	M	-
product	[A-Z a-z 0-9 - =':?_%,()&+.<> /*]	(3,80)	The type of product. See Appendix B: Product for the list of all possible products. Note that the partner must have the product in order for the merchant to have it.	M	-
product_configuration	Object		Object that holds the product configuration details (also referred to as <i>technical setup</i>). In case product configuration for the merchant is similar to the partner's- no need to send this Object (configuration will be inherited from the partner to the merchant)	O	-

product configuration object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
3ds_adviser_rba_amount_threshold	[0-9]{1,3}(?:[0-9]{1,2})	(0,6)	The threshold value for initiating 3D Secure when using a Risk Based approach. Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
3ds_adviser_smartguard_score_threshold	[0-9]	(0,4)	Specifies the risk score threshold for Source's SmartGuard service. Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-
3ds_enable_notification	[true false]	(4,5)	Specifies whether to implement a 3D Secure notification mechanism to automatically retrieve updates on the transaction processing progress without initiating another call to the gateway. Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-
3ds_notification_url	[A-Z a-z 0-9 - =':?_%,()&+.<> ;/*]	(0,255)	The URL to which 3D Secure notifications from Credorax should be sent. Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-
allowed_operations	[0-9]	(0,3)	List of all operations that the merchant should inherit from the partner. If all, leave empty. For example : 2,3,12,13	O	-
apm_allowed_operations	[A-Z a-z 0-9 - =':?_%,()&+.<> ;/*]	(0,3)	The payment methods allowed for this merchant Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-
apm_notification_addresses	[A-Z a-z 0-9 - =':?_%,()&+.<> ;/*]	(0,255)	The address to which APM notifications from Finaro should be sent. Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
apm_notification_mode	[true false]	(4,5)	Specifies whether to implement an APM notification mechanism for receiving an update about a transaction's status after the shopper's action. Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-
crx_acquiring_multi_capture	[true false]	(4,5)	Specifies whether the merchant may capture a payment multiple times (for example, for an order with items that can be shipped at different times)	O	-
hpp_code_3d_secure_required	[true false]	(4,5)	Specifies whether the merchant requires 3d Secure on its HPP code hosted payment page	O	-
hpp_code_enable_smartguard	[true false]	(4,5)	Specifies whether to enable SmartGuard if the merchant uses an HPP code solution	O	-
hpp_code_enable_token	[true false]	(4,5)	Specifies whether to enable token usage if the merchant uses an HPP code solution	O	-
hpp_code_shopping_cart	[A-Z a-z 0-9 - =':?_%,()&+. <> ; / *]	(0,255)	The type of shopping cart solution if the merchant uses an HPP code solution. The available options include: <ul style="list-style-type: none"> • Magento • WooCommerce 	O	-
hpp_form_3d_secure_required	[true false]	(4,5)	Specifies whether the merchant requires 3D Secure on its HPP form hosted payment page	O	-
hpp_form_enable_notification	[true false]	(4,5)	Specifies whether the merchant required registration for its HPP form notification service	O	-
hpp_form_enable_smartguard	[true false]	(4,5)	Specifies whether to enable SmartGuard if the merchant uses an HPP form solution	O	-
hpp_form_enable_token	[true false]	(4,5)	Specifies whether to enable token usage if the merchant uses an HPP form solution	O	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
hpp_form_skin_name	[A-Z a-z 0-9 - =':?_%,()&+.<>*/]	(0,255)	The name of the skin for setting up this merchant	O	-
pbl_allow_email	[true false]	(4,5)	Specifies whether the merchant is allowed to have the email PayByLink functionality	O	-
pbl_allow_sms	[true false]	(4,5)	Specifies whether the merchant is allowed to have the phone message PayByLink functionality	O	-
pbl_allow_vpos	[true false]	(4,5)	Specifies whether the merchant is allowed to have the email virtual POS functionality	O	-
pbl_countries_of_payers	[A-Z]	(3,3)	The countries which the payer can be from, in ISO 3-letter country code format	O	-
pbl_countries_of_payers_default	[A-Z]	(3,3)	The default payer country, in ISO 3-letter country code format	O	-
pbl_currency	[A-Z]	(3,3)	The currencies in which payment can be accepted, in ISO 3-letter currency code format	O	-
pbl_default_currencies	[A-Z]	(3,3)	The default currency or currencies, in ISO 3-letter currency code format. To specify multiple currencies, enter them as a comma separated list	O	-
pbl_languages	[A-Z a-z 0-9 - =':?_%,()&+.<>*/]	(0,2)	The languages in which the payment page can be sent, in ISO 2-letter language code format	O	-
pbl_language_default	[A-Z]	(0,2)	The default payment page language, in ISO 2-letter language code format	O	-
pbl_notification_address	[A-Z a-z 0-9 - =':?_%,()&+.<>*/]	(0,255)	The address to which payment notifications will be sent	O	-
pbl_notification_mode	[true false]	(4,5)	Specifies whether to activate notification mode	O	-
pbl_sender_email	[A-Z a-z 0-9 - =':?_%,()&+.<>*/]	(0,255)	The email address from which emails will be sent	O	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
pbl_skin_type	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(0,255)	The type of skin used for the payment page	O	-
pbl_thankyou_email_content	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(0,1024)	The content of the email which will be sent on behalf of the merchant after the payment is done. Relevant only if pbl_allow_email=true	O	-
pbl_thankyou_sms_content	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(0,1024)	The content of the message which will be sent on behalf of the merchant after the payment is done. Relevant only if pbl_allow_sms=true	O	-
smart_guard_blocking_threshold	[0-9]	(0,4)	The SmartGuard risk score threshold. Transactions with a SmartGuard risk score equal to or higher than this threshold will be blocked. Only relevant if the partner is registered for this service, and the merchant intends to use the service.	O	-
smart_guard_ml_standalone	[true false]	(4,5)	Indicates whether the merchant is enrolled for the standalone SmartGuard service	O	-
smart_guard_review_threshold	[0-9]	(0,4)	The SmartGuard transaction review threshold. Transactions with a SmartGuard risk score lower than this threshold will be automatically passed. Transactions with a SmartGuard risk score equal to or higher than this threshold, but lower than the smart_guard_blocking_threshold, will be passed and reviewed. Only relevant if the partner is registered for this service, and the merchant intends to use the service	O	-
token_skip_card_validation	[true false]	(4,5)	Indicates whether the merchant is allowed to create tokens without card validation	O	-

pos device object



This object is mandatory if you send a merchant_requirement object with a card present (CP) payment channel. It will be sent nested under the merchant_requirement object and will contain the POS device data.

Parameter Name	Type	Length (min,max)	Description	PSP	PF
address_line_1	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,80)	House number and street where the POS device is located	M	M
api_code	[0-9]	(1,4)	The code identifying the type of POS device as received from Finaro. For example: 009	M	M
city	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,80)	City in which the POS is located	M	M
country	[A-Z]	(3,3)	Country in which the POS is located ISO alpha 3 Code For example: GBR	M	M
number_of_devices	[0-9]	(1,4)	Number of POS devices of this type (=api_code), and at the this location	M	M
postcode	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,11)	Postal code of POS location	M	M
state	[A-Z a-z 0-9 - =':?_%(),&+.<> /*]	(1,5)	State in which the POS is located	O	O

pricing object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
fee_list	object	-	A list object that holds the fee name and the fee value of fees you want to have in this merchant pricing. Send this object to override fee values of the fixed pricing (also referred to as "partner terms") assigned to this merchant.	O	-
max_trx_value	[0-9.] Decimal: Max 10 digits + 2 decimal places	(1,13)	The maximum transaction value which is allowed for the merchants in this merchant pricing.	O	-
merchant_terms_identifier	[A-Z a-z 0-9 -]	(1,10)	Identifier assigned by the partner for the unique merchant pricing. This Identifier will be used to link the merchant requirements to the created merchant pricing, and to update the merchant pricing in the future.	M	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
partner_terms_identifier	[A-Z a-z 0-9 -]	(9, 10)	<p>The ID of the fixed merchant pricing terms you are using to assign pricing to this merchant.</p> <p>The ID always begins with a leading 'MPS-' and continues with max 6 digits. For example: MPS-660832</p> <p>NOTE: Enter here the same value as the one you entered in parameter partner_terms_identifier in the merchant bank account object.</p>	M	-

settlement_frequency	[A-Z a-z]	(5,12)	<p>The frequency in which the settlement is paid to the merchant in this merchant pricing.</p> <p>Supported values:</p> <ul style="list-style-type: none"> • Monthly • Weekly • Daily • Twice a week 	O	-
----------------------	-----------	--------	--	---	---

fee list object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
fee_name	[A-Z a-z]	(12,12)	<p>The fee name of the fee you want to set up for the merchant.</p> <p>Supported Values:</p> <ul style="list-style-type: none"> • Discount Fee 	M	-
fee_value	[0-9.] Decimal, Max 2 decimal places	(1,6)	<p>The value of the fee assigned for this merchant terms.</p> <p>Note: for discount fee the max value is 100 (reflecting the percentage)</p>	M	-



merchant requirement object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
active	[true false]	(4,5)	Specifies whether this merchant_requirement is active. Note: when creating a new merchant always send 'true'.	M	M
descriptor_city	[A-Z a-z 0-9 - =':?_%,()&+.<> /*]	(1,13)	The portion of the merchant descriptor that identifies the city in which the merchant's business is located	M	M
descriptor_dba_name	[A-Z a-z 0-9 - =':?_%,()&+.<> /*]	(1,22)	The portion of the merchant descriptor that identifies the merchant's "Doing Business As" (DBA) name	M	M
mcc_code	[0-9]	(4,4)	The requested merchant category code (MCC)	M	M
merchant_requirement_id	[A-Z a-z 0-9 - =':?_%,()&+.<> /*]	(1,85)	The partner's ID assigned for this merchant_requirement object. This parameter value should be 1 for the first merchant requirement ID and the consecutive ID number for each additional merchant requirement objects sent for the merchant	M	M
payment_channel	(WEB MOTO CP)	(2,4)	Indicates whether transactions are sent from WEB, MOTO or CP	M	M
pos_device	Object	-	Object that holds the pos_device details. Must be sent if the payment channel is CP.	C	C
pricing	Object	-	Object that holds the pricing details. When sending the pricing object under the merchant requirements, send only partner_terms_identifier and merchant_terms_identifier that is related to this merchant requirements	M	M
sub_merchant_id	[0-9]	(0,15)	The sub merchant ID assigned to the merchant. For example: 246338000011008	-	M



Parameter Name	Type	Length (min,max)	Description	PSP	PF
url	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(0,255)	The merchant's site URL. Relevant only if the payment channel is WEB. URL format: <ul style="list-style-type: none"> • must start with HTTP / HTTPS • must contain WWW. For example: https://www.nicepictures.dk/	C	C

online signature object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
date_timestamp	YYYY-MM-DDThh:mm:ss.000Z	(24,24)	The timestamp from the partner's system, recording when the partner request is sent Example: 2021-10-30T15:27:02.000Z	M	M
service_agreement_version	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(1,255)	The version of the service agreement to which this merchant is subject For example: v1	M	M
signer_ip	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(1,255)	The IP address of the signer of the merchant creation request.	M	M

registered address object

Parameter Name	Type	Length (min,max)	Description	PSP	PF
apartment_number	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(0,255)	The merchant's apartment number	O	O
city	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(1,80)	The merchant's city	M	M
country	[A-Z]	(3,3)	The merchant's country ISO alpha 3 Code. For example: GBR	M	M
house_number	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(1,255)	The merchant's house number	M	M
postcode	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(1,11)	The merchant's postal code	M	M
state	[A-Z a-z 0-9 - =':?_%,()&+.<>;/ *]	(0,5)	The merchant's state	O	O



Parameter Name	Type	Length (min,max)	Description	PSP	PF
street	[A-Z a-z 0-9 - =':?_%(),&+.<>:/ *]	(1,255)	The merchant's street	M	M

related corporate object

This object is only required for merchants defined as corporate (not sole traders). Use it to provide the merchant's company details as well as the details of any corporate director or shareholder

Parameter Name	Type	Length (min,max)	Description	PSP	PF
active	[true false]	(4,5)	Specifies whether this corporate is active. For merchant creation send 'true'.	M	-
address_line_1	[A-Z a-z 0-9 - =':?_%(),&+.<>:/ *]	(1,255)	The corporate's address line 1	M	-
city	[A-Z a-z 0-9 - =':?_%(),&+.<>:/ *]	(1,80)	The corporate's city	M	-
company_registration_name	[A-Z a-z 0-9 - =':?_%(),&+.<>:/ *]	(1,80)	The corporate's registration name	M	-
company_registration_number	[A-Z a-z 0-9 -]	(1,80)	The corporate's registration number	M	-
country	[A-Z]	(3,3)	The corporate's country ISO alpha 3 Code. For example: GBR	M	-
country_of_incorporation	[A-Z]	(3,3)	The corporate's country of incorporation ISO alpha 3 Code. For example: GBR	M	-
date_of_incorporation	YYYY-MM-DD	(10,10)	The corporate's data of incorporation For example: 2017-05-20	M	-
house_number	[0-9]	(1,3)	The corporate's house number	M	-
postal_code	[A-Z a-z 0-9 - =':?_%(),&+.<>:/ *]	(1,11)	The corporate's postal code	M	-



Parameter Name	Type	Length (min,max)	Description	PSP	PF
related_corporate	[A-Z a-z 0-9]	(1,255)	<ul style="list-style-type: none"> If this field is sent for a corporate with a holding in the merchant, specify here the company_registration_number of the merchant. If this field is sent for the onboarded merchant itself, send Null value. See Appendix D: Related Corporate and Merchant Individual Scenarios .	M	-
State	[A-Z a-z 0-9 - =':?_%(),&+.<>/*]	(0,5)	The corporate's state	O	-

Code Examples for Create Merchant

Create Merchant: Body Example

```
{
  "merchant_name": "test1",
  "business_type": "corporate",
  "business_model_description": "sales",
  "average_item_value_eur": "100",
  "expected_monthly_turnover": "45000",
  "company_registration_name": "test1",
  "company_registration_number": "12345",
  "country_of_incorporation": "GBR",
  "date_of_incorporation": "2021-07-06",
  "registered_address": {
    "city": "London",
    "country": "GBR",
    "street": "1 strawberry Drive",
    "house_number": "0",
    "postcode": "M30 8WE"
  },
  "online_signature": {
    "date_timestamp": "2021-10-30T15:27:02.000Z",
    "service_agreement_version": "v1",
    "signer_ip": "0"
  },
  "related_corporate": [
    {
      "company_registration_name": "test1",
```



```

    "company_registration_number": "12345",
    "country_of_incorporation": "GBR",
    "date_of_incorporation": "2021-07-06",
    "related_corporate": null,
    "active": true,
    "city": " London",
    "postal_code": "M30 8WE",
    "country": "GBR",
    "address_line_1": "1 strawberry Drive",
    "house_number": "0"
  }
],
"merchant_contact": [
  {
    "first_name": "Danny",
    "last_name": "Fox",
    "email": "dannyfox@gmail.com",
    "phone": "97250245544",
    "active": true,
    "role": "principal_contact"
  }
],
"merchant_individual": [
  {
    "active": true,
    "first_name": "dan",
    "last_name": "brill",
    "nationality": "GBR",
    "date_of_birth": "1983-04-15",
    "place_of_birth": "GBR",
    "official_id_type": "ID",
    "official_id_number": "fox834299GB9AZ60",
    "individual_type": "director",
    "city": "Londen",
    "postcode": "M30 8WE",
    "country": "GBR",
    "address_line_1": "1 strawberry Drive",
    "house_number": "0",
    "holding_percent": "100",
    "passport_place_of_issue": "ID",
    "official_id_expiry_date": "2028-04-05",
    "related_corporate": "12345"
  }
],
"merchant_product": [

```




```

    {
      "product": "Acquiring services",
      "active": true
    }
  ],
  "merchant_bank_account": [
    {
      "settlement_currency": "GBP",
      "active": true,
      "beneficiary_name": "test1",
      "bank_name": "HBC",
      "bank_country": "GBR",
      "iban": "GB99BUKB20454552546554",
      "swift": "GTRBGB58",
      "use_merchant_address": true,
      "partner_terms_identifier": "MPS-631500"
    }
  ],
  "pricing": [
    {
      "partner_terms_identifier": "MPS-631500",
      "merchant_terms_identifier": "SSS-555",
      "settlement_frequency": "Weekly",
      "max_trx_value": "9999",
      "fee_list": [
        {
          "fee_name": "Discount Fee",
          "fee_value": "0.3"
        }
      ]
    }
  ],
  "merchant_requirement": [
    {
      "pricing": {
        "partner_terms_identifier": "MPS-631500",
        "merchant_terms_identifier": "SSS-555",
      },
      "merchant_requirement_id": "1",
      "active": true,
      "payment_channel": "CP",
      "mcc_code": "7995",
      "descriptor_dba_name": "TEST1",
      "descriptor_city": "London",
      "pos_device": {
        "api_code": "009",
        "number_of_devices": "1",
        "city": "London",
      }
    }
  ]
}

```



```
        "country": "GBR",
        "address_line_1": "1 strawberry Drive",
        "postcode": "M30 8WE"
    },
    {
        "pricing": {
            "partner_terms_identifer": "MPS-631500",
            "merchant_terms_identifer": "SSS-555",
        },
        "merchant_requirement_id": "2",
        "active": true,
        "payment_channel": "MOTO",
        "mcc_code": "5072",
        "descriptor_dba_name": "test1",
        "descriptor_city": " London "
    }
]
}
```

Create Merchant: Response Examples

Create succeeded

Sync response (immediate):

```
{"crx_request_id": "c424665c9336410e9f7459031b1003ef", "original_request_time": "2022-10-28T13:41:44.097", "result": {"response_code": "000", "response_description": "Request is successful"}}
```

ASync response (Up to 4 hours), via webhook notification:

```
{"type": "Merchant onboarding API", "event_id": "b342417170704c4dbe54bcb9166569ed", "event_status_code": "03", "event_status_description": "Merchant was created successfully for merchant CEID-0000008335", "event_additional_fields": {"operation_type": "Merchant Creation", "result.merchant_id": "CEID-0000251426"}}
```



Create failed

Sync response (immediate):

```
{"crx_request_id": "2c2ff3e8beec4173b2affe1c2f890093", "original_request_time": "2022-10-28T13:42:09.571", "result": {"response_code": "012", "response_description": "At least one of input parameters is malformed.", "response_details": ["company_registration_name parameter has to be equal to the merchant upsert value in this case.", "company_registration_name parameter is required."]}}
```

ASync response (Up to 4 hours), via webhook notification:

```
{"type": "Merchant onboarding API", "event_id": "140761842829451e99625b3bab3b1b7b", "event_status_code": "05", "event_status_description": "Merchant was not created successfully for merchant CEID-0000008335", "event_additional_fields": {"operation_type": "Merchant Creation", "failure_message": "Account, You're creating a duplicate record. We recommend you use an existing record instead."}}
```



Update merchant

PROD: [POST] <https://onboarding.sourcepayments.com/setup/merchant/update/<merchant id>>

INT: <https://onboarding.int.sourcepayments.com/rest/setup/merchant/update/<merchant id>>

Request Fields

- When sending an Update merchant call, you must send all the root level fields listed below, as well as the relevant object(s) (depending on the type of update).
- For adding a new object
 - Active = true
 - Is_update= false
 - Send all the fields you would send in the creation
- For updating existing object:
 - Active = true
 - Is_update= true
 - Send only the fields you want to update as well as the field to identify the object
- For cancelling existing object:
 - Active = false
 - Is_update= true

Root level fields and objects

Parameter Name	Type	Length (min,max)	Description	Presence - PSP	Presence - PF
is_update	[true false]	(4,5)	At the root level, set this to is_update=true	M	M
merchant_id	[A-Z a-z 0-9 -]	(15,15)	The merchant ID, as provided in the notification returned to the partner once the merchant creation is successful. For example: CEID-0000190628	M	M



Parameter Name	Type	Length (min,max)	Description	Presence - PSP	Presence - PF
partner_id	[A-Z a-z 0-9 -]	(15,15)	The Source assigned ID of the partner creating the merchant For example: CEID-0000190325	M	M
merchant_bank_account	Object	-	A list object that can be sent more than once. It holds the merchant bank account details (also referred to as settlement bank account) This object includes a nested beneficiary address object	O	-
merchant_contact	Object	-	Object that holds the merchant contact details	O	O
merchant_individual	Object	-	Object that holds the merchant individual details	O	-
merchant_product	Object	-	Object that holds the merchant product details	O	-
merchant_requirement	Object	-	Object that holds the merchant requirement details (also referred to as Payment Channel)	O	O
registered_address	Object	-	Object that holds the registered address details	O	O
related_corporate	Object	-	Object that holds the related corporate details	O	-

merchant bank account object

Using this object in the Update merchant API call, you can add or cancel a merchant bank account (i.e., a settlement bank account).

- To add a settlement bank account, specify **active=true**, **is_update=false**, and all the fields you would send in the [merchant bank account](#) object of a **Create merchant** call.
- To cancel a settlement bank account, specify **active=false**, **is_update=true**, as well as the **settlement_currency** field to identify the bank account, as described in the table below:

Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence- PF
active	[true false]	(4,5)	Specifies whether a merchant bank account is being added (active=true), or cancelled (active=false)	M	-



Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence- PF
is_update	[true false]	(4,5)	<p>Specifies whether to create a new object of this type (is_update=false), or cancel an existing object (is_update=true). The existing object is identified by its settlement_currency.</p> <p>Specifies whether a merchant bank account is being added (active=true), or whether a merchant bank account is being cancelled (active=false).</p> <p>To create a new object of this type, send all the fields you would send in the merchant bank account object of a Create merchant call.</p>	M	-
settlement_currency	[A-Z]	(3,3)	<p>The currency in which funds are deposited in the merchant's bank account.</p> <p>ISO alpha 3 Code</p> <p>For example: GBP</p>	M	-

merchant contact object

Using this object in the Update merchant API call, you can add, update or cancel a merchant contact.

- To add a merchant contact, specify **active=true**, **is_update=false**, and the rest of the fields listed below.
- To update a merchant contact, specify **active=true**, **is_update=true**, the **email** field to identify the contact, and the fields you wish to update from those listed below.
- To cancel a merchant contact, specify **active=false**, **is_update=true**, as well as the **email** field to identify the contact.

Parameter Name	Type	Length (min,max)	Description	Presence - PSP	Presence - PF
active	[true false]	(4,5)	Specifies whether a merchant contact is being added or updated (active=true), or whether a merchant contact is being cancelled (active=false).	M	M
email	[A-Z a-z 0-9 - =':?_%,()&+.<>:/*]	(1,80)	Email of the merchant's contact person	M	M



Parameter Name	Type	Length (min,max)	Description	Presence - PSP	Presence - PF
first_name	[A-Z a-z 0-9 - =':?_%,()&+.<> ;/*]	(1,40)	First name of the merchant's contact person	M	M
is_update	[true false]	(4,5)	Specifies whether to create a new object of this type (is_update=false), overwrite the values of an existing object (is_update=true), or cancel an existing object (is_update=true). The existing object is identified by its email .	M	M
last_name	[A-Z a-z 0-9 - =':?_%,()&+.<> ;/*]	(1,80)	Last name of the merchant's contact person	M	M
mobile_phone	[0-9]	(1,40)	Mobile phone number of the merchant's contact person	O	M
phone	[0-9]	(1,40)	Landline phone number of the merchant's contact person	M	M
role	Values from the list: <ul style="list-style-type: none"> primary_contact cbk_contact general insights 	(7,15)	This field specifies the Role of the merchant's contact person. <ul style="list-style-type: none"> If the contact has several roles, enter them as a list separated by semicolons (";") if role=general, no other roles can be indicated for the contact 	M	O

merchant individual object

Using this object in the Update merchant API call, you can add or update a merchant individual.

- To add a merchant individual, specify **active=true**, **is_update=false**, and **all the fields** you would send in the [merchant individual](#) object of a **Create merchant** call.
- To update a merchant individual, specify **active=true**, **is_update=true**, the **official_id_number** to identify the individual, and the fields you wish to update from among those listed below.



Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence- PF
active	[true false]	(4,5)	Specifies whether a merchant individual is being added or updated (active=true)	M	-
address_line_1	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(1,255)	Individual's street name	O	-
city	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(1,80)	Individual's city of residence	M	-
country	[A-Z]	(3,3)	Individual's country of residence ISO alpha 3 Code. For example: GBR	M	-
holding_percent	[0-9.] Decimal: 3 digits + 2 decimal places	(1,6)	Specifies the Individual's percentage of ownership of the corporation.	O	-
house_number	[0-9]	(1,9)	Individual's house number	O	-
is_update	[true false]	(4,5)	Specifies whether to create a new object of this type (is_update=false), or overwrite the values of an existing object (is_update=true). The existing object is identified by its official_id_number .	M	-
official_id_expiry_date	YYYY-MM-DD	(10,10)	The expiry date of the official ID used by the individual. For example: 2030-03-24	O	-
official_id_number	[A-Z a-z 0-9 -]	(1,50)	The ID number/passport of the official ID used by the individual	M	-
passport_place_of_issue	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(1,255)	The place of issue of the Individual's passport Mandatory if official_id_type = 'Passport'	C	-
postcode	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(1,11)	The individual's postal code	M	-
state	[A-Z a-z 0-9 - =':?_%,()&+.<>;/*]	(0,5)	The individual's state	O	-



merchant_product object

Using this object in the Update merchant API call, you can add, deactivate or update product configuration of a merchant product.

- To add a merchant product, specify **active=true**, **is_update=false**, and all the fields you would send in the [merchant_product](#) object of a **Create merchant** call.
- To deactivate a merchant product, specify **active=false**, **is_update=true**, as well as the **product** field to identify the product.
- To update a merchant product configuration, specify **active=true**, **is_update=true**, the product field to identify the product, and a nested product_configuration object with updated fields.
product_configuration object fields are the same as in create merchant



Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence- PF
active	[true false]	(4,5)	Specifies whether a merchant product is being added, updated (active=true), or deactivated (active=false)	M	-
is_update	[true false]	(4,5)	Specifies whether to create a new object of this type (is_update=false), or deactivate / edit an existing object (is_update=true). The existing object is identified by its product . To create a new object of this type, send all the fields you would send in the merchant product object of a Create merchant call	M	-
product	[A-Z a-z 0-9 - =':?_%,()&+.< >;/*]	(3,80)	The type of product. See Appendix B: Product for the list of all possible products. Note that the partner must have the product in order for the merchant to have it.	M	-
product_configuration	Object		Object that holds the product configuration details (also referred to as <i>technical setup</i>). In case product configuration for the merchant is similar to the partner's- no need to send this Object (configuration will be inherited from the partner to the merchant)	O	-

merchant requirement object

Using this object in the Update merchant API call, you can add, update or cancel a merchant requirement, i.e., a payment channel:

- To add a payment channel, specify **active=true**, **is_update=false**, and all the fields you would send in the **merchant_requirement** object of a **Create merchant** call
- To update a payment channel, specify **active=true**, **is_update=true**, the **merchant_requirement_id** to identify the payment channel. If the update is for a CP payment channel send also a nested **pos_device** object with updated fields (you can change the number of POS devices and their address).
- To cancel a payment channel, specify **active=false**, **is_update=true**, as well as the **merchant_requirement_id** to identify the payment channel.



Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence- PF
active	[true false]	(4,5)	Specifies whether a merchant requirement is being added or updated (active=true), or whether a merchant requirement is being cancelled (active=false).	M	M
is_update	[true false]	(4,5)	Specifies whether to create a new object of this type (is_update=false), overwrite the values of an existing object (is_update=true), or cancel an existing object (is_update=true). The existing object is identified by its merchant_requirement_id	M	M
merchant_requirement_id	[A-Z a-z 0-9 - =!:_%,()&+.<> /*]	(1,85)	The partner assigned ID for this merchant_requirement object	M	M
pos_device	Object	-	Object that holds the pos_device details. Must be sent if the payment channel is CP.	C	C

pricing object

- Using this object in the Update merchant API call, you can add or update a merchant pricing for these attributes: discount fee, settlement frequency and maximum transaction value.
- Linking new pricing to existing merchant requirement will be done through creation of new merchant requirement with the required pricing identifier.

Parameter Name	Type	Length (min,max)	Description	Presence - PSP	Presence - PF
fee_list	object	-	A list object that holds the fee name and value the partner wants to have in this merchant pricing. Send this object in merchant update in order to update fee value in existing pricing, or to create a pricing with new fee value	O	-
is_update	[true false]	(4,5)	Specifies whether to create a new pricing (is_update=false), or overwrite the values of an existing pricing (is_update=true)	M	



Parameter Name	Type	Length (min,max)	Description	Presence - PSP	Presence - PF
max_trx_value	[0-9.] Decimal: Max 10 digits + 2 decimal places	(1,13)	The maximum transaction value which is allowed for the merchants in this merchant pricing	O	-
merchant_terms_identifier	[A-Z a-z 0-9 -]	(1,10)	Identifier assigned by the partner for this merchant pricing. This Identifier will be used in order to link the merchant requirements to created merchant pricing, and to update the merchant pricing in the future.	M	-
partner_terms_identifier	[A-Z a-z 0-9 -]	(9, 10)	The ID of the fixed merchant pricing terms you are using to assign pricing to this merchant. The ID always begins with a leading 'MPS-' and continues with max 6 digits. For example: MPS-660832 NOTE: you receive these IDs as part of your connectivity details before go-live	M	-
settlement_frequency	Value from the list: <ul style="list-style-type: none"> • Monthly • Weekly • Daily • Twice a week 	(5,12)	The frequency in which the settlement is paid for the merchant in this merchant pricing	O	-

registered address object

Using this object in the Update merchant API call, you can update a registered address.

Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence- PF
apartment_number	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(0,255)	The merchant's apartment number	O	O
city	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(1,80)	The merchant's city	M	M
country	[A-Z]	(3,3)	The merchant's country ISO alpha 3 Code. For example: GBR	M	M



Parameter Name	Type	Length (min,max)	Description	Presence-PSP	Presence-PF
house_number	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(1,255)	The merchant's house number	M	M
is_update	[true false]	(4,5)	Specifies to update the values of an existing object (is_update=true).	M	both
postcode	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(1,11)	The merchant's postal code	M	M
state	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(0,5)	The merchant's state	O	O
street	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(1,255)	The merchant's street	M	M

related corporate object

Using this object in the Update merchant API call, you can add or update an existing related corporate. The existing object is identified by its **company_registration_number**.

- To add a *related corporate*, specify **active=true**, **is_update=false**, and all the fields you would send in the *related_corporate* object of a **Create merchant** call.
- To update a *related corporate*, specify **active=true**, **is_update=true**, the **company_registration_number** to identify the related corporate, and the fields you wish to update from among those listed below.

Parameter Name	Type	Length (min,max)	Description	Presence-PSP	Presence-PF
active	[true false]	(4,5)	Specifies whether this corporate is active. For merchant creation send 'true'.	M	-
address_line_1	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(1,255)	The corporate's address line 1	M	-
city	[A-Z a-z 0-9 - =':?_%,()&+. <> ;/*]	(1,80)	The corporate's city	M	-
company_registration_number	[A-Z a-z 0-9 -]	(1,80)	The corporate's registration number	M	-
country	[A-Z]	(3,3)	The corporate's country ISO alpha 3 Code. For example: GBR	M	-
house_number	[0-9]	(1,3)	The corporate's house number	M	-



Parameter Name	Type	Length (min,max)	Description	Presence-PSP	Presence-PF
is_update	[true false]	(4,5)	Specifies whether to create a new object of this type (is_update=false), or overwrite the values of an existing object (is_update=true).	M	-
postal_code	[A-Z a-z 0-9 - =':?_%,()&+.<>*/]	(1,11)	The corporate's postal code	M	-
related_corporate	[A-Z a-z 0-9]	(1,255)	company_registration_number of the merchant.	M	-
State	[A-Z a-z 0-9 - =':?_%,()&+.<>*/]	(0,5)	The corporate's state	O	-



Merchant Update: Code Examples (Body)

Adding object- individual

```
{ "is_update": true,
  "merchant_individual": [
    { "first_name": "Spiritual",
      "last_name": "Individual",
      "nationality": "USA",
      "date_of_birth": "1988-07-02",
      "place_of_birth": "DNK",
      "official_id_type": "Passport",
      "official_id_number": "122999",
      "individual_type": "director",
      "city": "Esbjerg",
      "state": "IL",
      "postcode": "6710",
      "country": "DNK",
      "address_line_1": "Sandbakken",
      "address_line_2": "10",
      "holding_percent": "0",
      "passport_place_of_issue": "DNK",
      "active": true,
      "house_number": "412",
      "official_id_expiry_date": "2028-06-19"
    }
  ]
}
```

Modifying MR object (changing number of devices)

```
{ "is_update": true,
  "merchant_requirement": [
    { "is_update": true,
      "active": true,
      "merchant_requirement_id": "5",
      "pos_device": {
        "address_line_1": "tes_upd",
        "city": "NEW",
        "country": "ITA",

```



```

    "number_of_devices": "15",
    "postcode": "new",
    "state": "CA"
  } } ] ]

```

Merchant Update: Response Examples

Update succeeded

Sync response (immediate):

```

{"merchant_id": "CEID-0000251426", "crx_request_id": "1af81ad2cab74fc08e248de34e1c9720",
 "original_request_time": "2022-12-20T17:51:28.318", "result": { "response_code": "000",
 "response_description": "Request is successful" }}

```

ASync response (Up to 4 hours), via webhook notification:

```

{"type": "Merchant onboarding API", "event_id": "1af81ad2cab74fc08e248de34e1c9720",
 "event_status_code": "04", "event_status_description": "Merchant was updated successfully for merchant CEID-0000008335", "event_additional_fields": {"operation_type": "Merchant Update" }}

```

Update failed

Sync response (immediate):

```

{"merchant_id": "CEID-0000251426", "crx_request_id": "62b17a987afa4500bebe030d60d3388a",
 "original_request_time": "2022-12-20T17:58:17.646", "result": {"response_code": "012",
 "response_description": "At least one of input parameters is malformed.", "response_details":
 ["merchant_individual[0].first_name parameter is not allowed to update."
 ] }}

```




ASync response (Up to 4 hours), via webhook notification:

```
{ "type": "Merchant onboarding API", "event_id": "82b58722fcd647f6a7d51f5e392f7204",  
  "event_status_code": "06", "event_status_description": "Merchant was not updated successfully for merchant  
  CEID-0000008335", "event_additional_fields": {"operation_type": "Merchant Update",  
  "failure_message": "Related company not exist 33061103, The record you wanted to update does not exist  
  123456" }}
```



Upload document

PROD: `https://onboarding.sourcepayments.com/rest/setup/merchant/ upload/<merchant id>?related=<related>&file_type=<file type>&submitMerchant =<boolean>`

INT: `https://onboarding.int.sourcepayments.com/rest/setup/merchant/ upload/<merchant id>?related=<related>&file_type=<file type>&submitMerchant=<boolean>`

Supported files type: PDF, JPEG, JPG, PNG, TIFF

Maximum size: 8MB

URL Parameters

Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence - PF
merchant_id	[A-Z a-z 0-9 -]	(15,15)	The merchant ID, as provided in the notification returned to the partner once the merchant creation process is complete. For example: CEID-0000190628	M	-
file_type	[0-9]	(1,2)	A code specifying the type of file being uploaded. The available codes include:	M	-
related	[A-Z a-z 0-9 -]	(1,80)	This parameter relates the uploaded document either to an individual or a corporate that was sent in the merchant create <ul style="list-style-type: none"> In case the uploaded document is related to an individual, specify here the official_id_number from the relevant the merchant individual object In case the uploaded document is related to a corporate, specify here the company_registration_number of the relevant corporate from the related corporate object 	M	-



Parameter Name	Type	Length (min,max)	Description	Presence- PSP	Presence - PF
submitMerchant	[true false]	(4,5)	<ul style="list-style-type: none"> This parameter indicates the partner is submitting the merchant and the files to start onboarding process in finaro. This indication can be sent more than once if additional files for the merchant is required. Optional parameter- the partner will send 'true' indication only when the merchant is ready to start onboarding and needed files were provided (see appendix F for needed onboarding files). Else, the parameter will have 'false' or not sent at all. 	O	-

File Upload: Response Examples

file upload succeeded

Sync response (immediate):

```
{
  "merchant_id": "CEID-0000221963",
  "upload_id": "85b9e347ef0048fe927b4121eabe58c6",
  "original_request_time": "2022-10-28T13:43:38.812",
  "result": {
    "response_code": "000",
    "response_description": "File received"
  }
}
```

Sync response (immediate)- with submitMerchant=true:

```
{
  "merchant_id": "CEID-0000221963",
  "upload_id": "85b9e347ef0048fe927b4121eabe58c6",
  "original_request_time": "2022-10-28T13:43:38.812",
  "result": {
    "response_code": "000",
    "response_description": "File received and merchant is submitted"
  }
}
```

ASync response (Up to 4 hours), via webhook notification

```
{
  "type": "File upload",
  "result": {
    "event_id": "aba90de07e414966b3d258f5d5a8a5b8",
    "event_status_code": "00",
    "event_status_description": "File uploaded successfully for merchant CEID-0000200332.",
    "event_additional_fields": null
  }
}
```

file upload Failed

**Sync response (immediate):**

```
{"merchant_id": "CEID-0000221963111", "upload_id": "2d7748ae7b3847b1959be5c963192eaa",  
"original_request_time": "2022-11-08T09:26:27.193", "result": {"response_code": "012",  
"response_description": "File type is not supported"}}
```

ASync response (Up to 4 hours), via webhook notification:

```
merchantId=null, partnerId=CEID-0000008335, partnerRequestId=c892cb1e274c4c5caba7a03267d669ea,  
responseCode=01, responseCodeDescription=File upload failed for merchant CEID-0000221960.The record  
you wanted to update does not exist 256178., eventType=File upload result, messageForCustomer=null,  
credonoteRequest=null, credonoteResponse=null
```



Get merchant status

PROD: [https://onboarding.sourcepayments.com/rest/setup/merchant/ <merchant id>](https://onboarding.sourcepayments.com/rest/setup/merchant/<merchant id>)

INT: [https://onboarding.int.sourcepayments.com/rest/setup/merchant/ <merchant id>](https://onboarding.int.sourcepayments.com/rest/setup/merchant/<merchant id>)

Response fields

The possible merchant statuses include:

Status	Description
Qualification	Merchant is created (First onboarding status for MOB)
In Pre Approval	Merchant is submitted and pending for review and approval
Qualified	Merchant internal approval process in Finaro is done
Underwriting	Underwriting phase has started
Contract Management	Contract phase has started
Pending Operation Setup	Setup phase has started
Processing	Merchant's onboarding is completed and ready to process
Disqualified	Merchant is disqualified due to business considerations
Disconnected	The merchant is no longer processing with finaro
Closed Lost	The merchant is no longer relevant

Response example

```
{ "result":{
  "response_code": "000",
  "response_description": "Request is successful",
  "response_details": [
    {
      "merchant_id": "CEID-0000250867",
      "merchant_status": "Pending Operation Setup"
    }
  ]
}
```



Get Connectivity details

PROD:

https://onboarding.sourcepayments.com/rest/setup/merchant/merchant_connectivity/<merchantId>

Response fields

The following information for connectivity details will be retrieved for each merchant:

Field name	Description
merchant ID	The merchant ID connectivity details are retrieved for
gateway_mid	The gateway MID this merchant has. If the merchant has several gateway_mid all will be retrieved
channel	Specifies weather this GWMID is card present / card not present
descriptor	The descriptor assigned to the merchant by the partner in merchat create call
last_modified_date	Last modification date of the gateway MID or the terminal
secret_key	The secret key of the gateway MID in order to send transactions via the payment API
target_mid	The terminal ID of this gateway MID. If the gateway MID has several terminals all will be retrieved
max_transaction_value	Max transaction value allowed for this terminal as was set in onboarding underwriting phase
min_transaction_value	Min transaction value allowed for this terminal as was set in onboarding underwriting phase
mcc_mc	Approved MC MCC for this GW MID or terminal
mcc_visa	Approved VISA MCC for this GW MID or terminal
Processor	The processor of this terminal transactions
device_type	The device type of this terminal. Relevat only for card present
device_street	The device street of this terminal. Relevat only for card present
device_city	The device city of this terminal. Relevat only for card present
device_country	The device country of this terminal. Relevat only for card present
transaction_currencies	Allowed transaction currencies as were set int the fixed MPS
allowed_card_schemes	Allowed card schemes currencies as were set int the fixed MPS
allowed_operations	Allowed operations as were set in product configuration
skin_id	The selected Skin of the payment pahe for HPP code product
static_key	HPP Static key provided during onboarding for HPP code product



Response example

```
{ "crx_request_id": "489d8dd0d4064a7cb7e42ca4089cc43a",
  "original_request_time": "2023-01-25T07:45:50.739",
  "result": {
    "response_code": "000",
    "response_description": "Request is successful",
    "response_details": [
      {
        "merchant_id": "CEID-0000251508",
        "details": [
          {
            "gateway_mid": "10029503",
            "channel": "Card Present",
            "descriptor": "Faustino Lviv",
            "last_modified_date": "2022-12-26 12:40:17",
            "secret_key": "1672058313251atc8ojj",
            "Terminals": [
              {
                "target_mid": "R0000052",
                "card_present": true,
                "last_modified_date": "2022-12-26 12:40:17",
                "max_transaction_value": 2500,
                "min_transaction_value": 0,
                "mcc_mc": "5999",
                "mcc_visa": "5999",
                "processor": "CREDORAX",
                "device_type": "Ingenico iCT250",
                "device_street": "Ruska street",
                "device_city": "Lviv",
                "device_country": "United States",
                "transaction_currencies": [
                  "EUR",
                  "DKK",
                  "CZK",
                  "SEK",
                  "NOK"
                ],
                "allowed_card_schemes": [
                  "MasterCard",
                  "Visa"
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```



```
"last_modified_date": "2022-12-26 12:40:17",
"max_transaction_value": 2500,
"min_transaction_value": 0,
"mcc_mc": "5999",
"mcc_visa": "5999",
"processor": "CREDORAX",
"device_type": "Ingenico iCT250",
"device_street": "Ruska street",
"device_city": "Lviv",
"device_country": "United States",
"transaction_currencies": [
  "EUR",
  "DKK",
  "CZK",
  "SEK",
  "NOK"
],
"allowed_card_schemes": [
  "MasterCard",
  "Visa"
]
},
}
]
}
]
}
```




Appendix A: HMAC-SHA512 Request Signature

Every API request is associated with a package signature sent as an authorization header in order to ensure the authenticity of data transfer. This package signature, in turn, is calculated on a newline separated list of values using HMAC-SHA512 with the requestor's unique secret key.

The header template is:

Authorization: HMAC-SHA512 Credential=<clientId>, Signature=<HMAC SHA-512 in hex>

Calculating the Signature

1. Apply the HMAC-SHA512 hashing algorithm to the JSON body of the request and the merchant's secret key.
2. Append the result of step 1 to the request's authentication header according to the above header template.

Signature Calculation Example

- In order for a MOB request to successfully pass authorization, you need to supply two headers with the request: date and authorization

Date

Fri, 26 Aug 2022 12:14:30 GMT (the format is important)

Authorization

In order to compose an auth header, you are going to need the following information:

1. Request type: for create or update it's POST, for get_status it's GET
 2. Link where the request will be sent
 3. Date - the same that you add to the Date header
 4. JSON of the request. **Make sure to remove spaces and newlines in JSON.**
- All these elements are to be combined into a single string, with each component written from the next line, for example:

```
POST
```



```
https://onboarding.sourcepayments.com/setup/merchant/Fri, 26 Aug 2022 12:14:30 GMT
{
  "json": true
}
```

- Then you need to get a hash of this content. You can use a web tool, for example: <https://codebeautify.org/hmac-generator>.
- The result of applying HMAC-SHA512 to the request body and secret should look like this:
3464b92a69578ece8fd422cc355c7e4cf3948770065e19dc517c97b51f1f5192f180c920b02
449e1ad4734420dea9eac8f51952a880f53e18a453db6308ad50f
- Once you have the hashed value, add it to the request header in the following format:
Authorization: **HMAC-SHA512 Credential=YOURPARTNER, Signature=long_hashed_string**
YOURPARTNER - partner id in format CEID-0000000000
long_hashed_string - long string we received when transformed our request data



Appendix B: Products

This appendix lists all the possible products that can be specified in the "product" field of the [merchant_product](#) object:

Product name

3D Secure

3DS Adviser

Account Updater on Demand

Acquiring services

Amex (via Cybersource)

APM Enabler

Apple Pay

Astropay direct

Banorte bank (via Cybersource)

BNP (Via Computop)

Cal (Via Yaad Sarig)

Card present package

Chargeback prevention -Ethoca

Chargeback prevention -RDR

Credorax acquiring

Credorax on behalf

ECP

FirstData

Gateway Services

Google Pay

HPP Code

HPP Form

Isracard (Via Yaad Sarig)

Israeli acquiring - Leumi Card

Israeli Acquiring Processor (Via Yaad Sarig)

LPM Services

Nuvei

PayByLink



Product name

PayPal

Raiffeisen bank (via Cybersource)

Retry optimization

Smart guard

Spanish connectivity (Redsys)

Token

Token Account Updater

TSYS (via Cybersource)

UATP (via Cybersource)

Worldpay



Appendix C: Result Codes

This appendix lists all the possible result codes that can be returned in the “result” object and their corresponding descriptions:

Note: The result description is used for reference and may change from time to time.

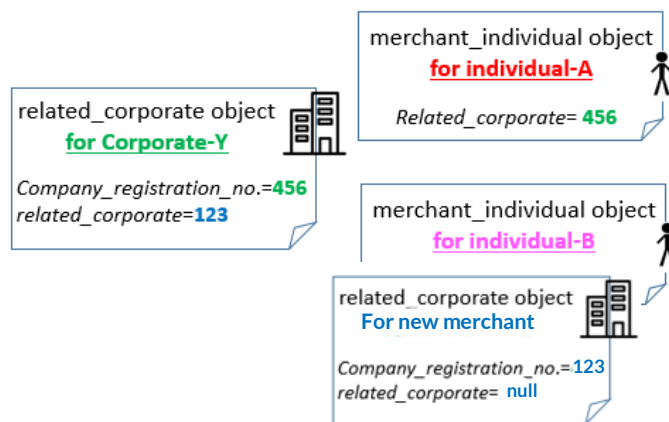
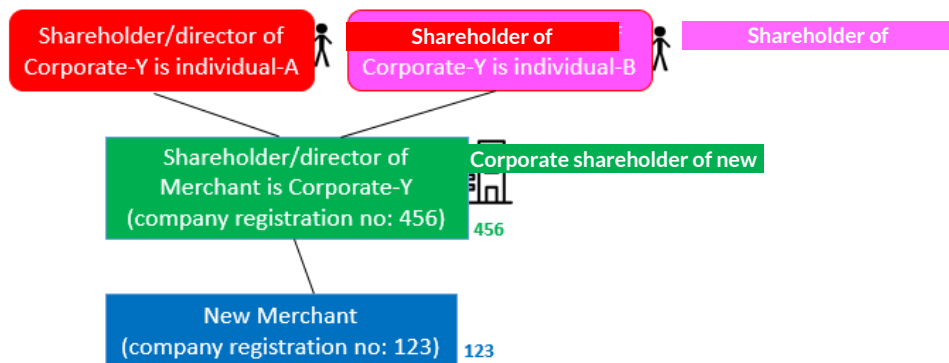
Result Code	Description
000	Request is successful
003	Merchant was created successfully for merchant “CEID-XXX”
004	Merchant was updated successfully for merchant “CEID-XXX”
005	Merchant was not created successfully for merchant “CEID-XXX”
006	Merchant was not updated successfully for merchant “CEID-XXX”
012	At least one of the following input parameters is malformed
014	At least one of the following input parameters is missing
016	Authentication failed due to invalid authentication credentials
020	Operation not allowed
009	Too many requests. We recommend an exponential backoff of your requests.
001	Failure due to a technical reason
007	Credorax request timeout
099	Request is being processed. A full response will be returned in a separate notification with the final response data



Appendix D: Related Corporate and Merchant Individual Scenarios

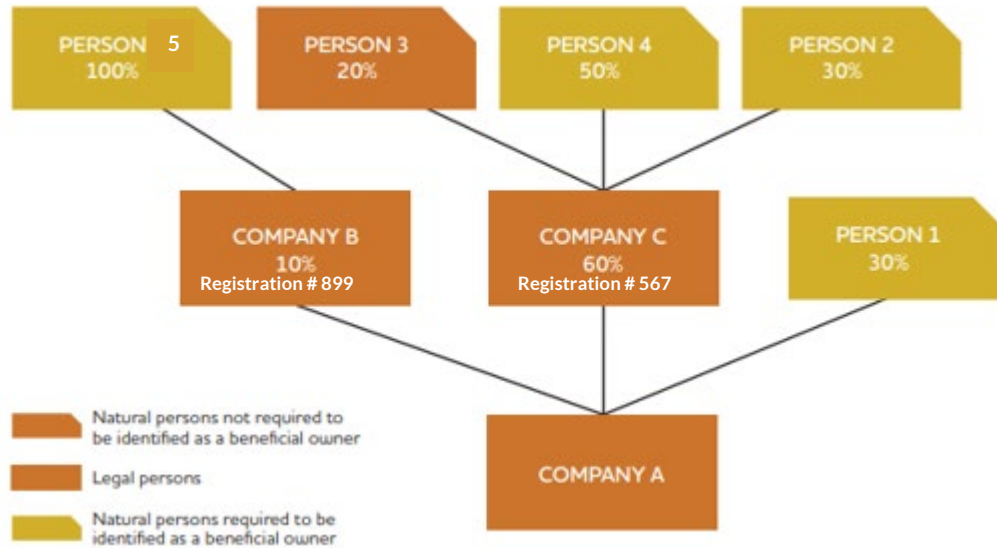
The following figures illustrates scenarios in which both a related_corporate object and a merchant_individual object are sent.

Scenario 1





Scenario 2



In this example need to send the following:

- Company A- this is the onboarded merchant. Send related_corporate object with company_reg_number=123 and related corporate=null
- Company B- Send related_corporate object with company_reg_number=899 and related corporate=123.
- Company C- Send related_corporate object with company_reg_number=567 and related corporate=123.
- Person 1- Send merchant_individual object for this person since he holds over 25% of company A
- Person 2+4 - Send merchant_individual object for this persons since they holds over 25% of company C
- Person 3-No need for merchant_individual object for this person since he holds less than 25% of company C
- Person 5- Send merchant_individual object for this person since he holds over 25% of company B



Appendix E: Operation codes

This appendix lists all the possible operation codes values that can be specified in the "allowed_operations" field of the product_configuration object

Operation code

- [12] - Use Token - Auth
- [13] - Use Token - Capture
- [14] - Token Auth Void
- [15] - Token Referral Credit
- [16] - Block Token
- [2] - Authorization
- [23] - Create Token with Sale
- [24] - Use Token - Recurring Sale
- [28] - Create Token - Auth
- [29] - Create Token - Capt
- [3] - Capture
- [301] - Fraud transaction update
- [32] - Use Token - Recurring Auth
- [33] - Use Token - Recurring Capt
- [34] Referral CFT
- [35] - Independent CFT
- [37] - Create Token independent CFT
- [38] - Use token independent CFT
- [4] - Authorization Void
- [46] - Use Token - Independent Credit
- [5] - Referral Credit
- [55] - ATC update
- [6] - Independent Credit
- [7] - Sale Void
- [8] - Refund Void
- [88] - 3DS only create token
- [89] - 3DS only use token
- [9] - Capture Void



Operation code

[92] - 3D Secure completion

[96] - versioning

[97] - Use token versioning

[98] - 3D Secure authentication Only



Appendix F: Onboarding Documents

This chapter lists the details for adding onboarding documents as part of the **upload document API request**.

Possible Values for the file_type Parameter

Code	File type
1	ID
2	Passport
3	Utility bill
4	Bank statement
5	Certificate of registration
6	Register of shareholders
7	Processing history
9	PCI SAQ
10	Deed of trust
11	License passporting
12	Certificate of incorporation
13	Proof of address
14	Register of directors
20	License to carry out the business
29	Power of attorney
99	Financial statements

Required Documents

The list below consists of the perlimanry list of KYC documents required from the merchant. Additional documents may be needed upon Finaro's review of the merchants.

Authorized Signatory

- Valid Proof of Identity- file code 1 or 2
- Proof of Residential Address (not older than 6 months)- file code 13
- Power of Attorney / Authorization to sign- file code 29



Majority Shareholder / UBO

- Valid Proof of Identity- file code 1 or 2
- Proof of Residential Address (not older than 6 months)- file code 13

Business Activity Documents

- Recent financials- file code 99
- PCI Documents- file code 9
- Business license (if relevant) - file code 20
- Processing history (last 6 months) - file code 7

Settlement

- Bank Statement in the name of the merchant- file code 4



Change History

Version	Subject/Date	Description
1.3	July 2023	Added pricing management functionality <ul style="list-style-type: none"> • merchant_bank_account object: Change of parameter name from <code>acquiring_pricing_id</code> to <code>partner_terms_identifier</code> • merchant_requirement object: Change of parameter name from <code>acquiring_pricing_id</code> to <code>partner_terms_identifier</code> • New mandatory pricing object
1.2	April 2023	Submit merchant function
1.1	March 2023	<ul style="list-style-type: none"> • Get connectivity API • Appendix F: onboarding documents list • URL field type • Apartment number field type
1.0	January 2023	First Release



Need Support?

Contact our Client Relations Centre 24/7 for any additional information or technical issue:

US: +1.617.715.1977

UK: +44.20.3608.1288

EU: +356 2778 0876

Email: support@finaro.com