CASHlib PAY CASH ONLINE

Search...

Introduction

Merchant Interface Description

Cashlib Flow

Resource information

Error Codes

Payment Process - Iframe

Kyc Process - Iframe

Tests to be made before going LIVE

Endpoints

Endpoints to implement

Merchant Cashlib (1.3.0)

Download OpenAPI specification: Download

This document is the technical specification of the interface between CASHlib and e-Merchant websites.

Introduction

CASHlib permits a person to buy a **voucher** at a Distributor **Point Of Sale (POS)**. Then he can pay his purchases on affiliated e-merchant web sites with the vouchers he has previously bought. Vouchers can also be bought on-line.

CASHlib voucher is a paper ticket with a 16 digit **Code** associated with a **value** and an **expiry_date** (and other attributes).

User has to enter the 16 digit code on the e-merchant website to make his payment. If the purchase amount is lower than voucher value then the voucher **balance** is updated with the remaining value and the user may later make other payments with this voucher. If the purchase amount is higher than user's voucher balance then the user may add other vouchers to reach the payment amount.

Several e-merchant websites may belong to the same Group which is called a Partner.

Vouchers issued by CASHlib can be branded (Partner's brand with vouchers to be used only for purchases in the Partners websites) or non-branded (ie CASHlib). Processing branded or non-branded vouchers is the same.

CASHlib vouchers are anonymous. However, **User** is "identified" when purchasing on Merchant websites by his **IP address** and by his **name/firstname** and **birthdate**. Purchase limits are imposed to users depending on their experience with CASHlib. No limits are set for **VIP** users but they have to provide **KYC** (Know Your Customer) information.

A **crosscountry** voucher is bought in a country and may be used for purchasing in another country.

Periodically (eg monthly), CASHlib computes Partners settlements. This settlement is mainly based on total amount of purchases made on the Partners websites less Cashlib fees. Cashlib fees depend on transaction volume.

CASHlib management and administration is made via a Backoffice web site which is accessible to Merchants (with access only to their area) and CASHlib administrators/managers (full view).

Merchant Interface Description

The parts detail the interface between CASHlib and E-Merchants.

This interface relies on an https/TLSV1 link (V1.2).

CASHlib provides 4 web services (API) to affiliated E-Merchants in order to purchase with CASHlib vouchers:

- Voucher_Payment
- Transaction_Info
- Voucher_Directpayment
- Kyc_Directpayment

Voucher_Payment permits to open an IFRAME on the user's web browser and entirely processes the payment with the connected customer. After completing the payment process, the customer is returned to the e-merchant URL either success_url if payment is successful or cancel_url if the payment failed or has been cancelled by customer.
(If the user has not entered all necessary voucher codes within 1 hour then the payment is rejected: ie payment is rejected by CASHlib if the user enters his voucher codes more than 1 hour after voucher_payment request has been received by CASHlib.)

Then, the integration of payment with CASHlib vouchers is very easy for the Merchant.

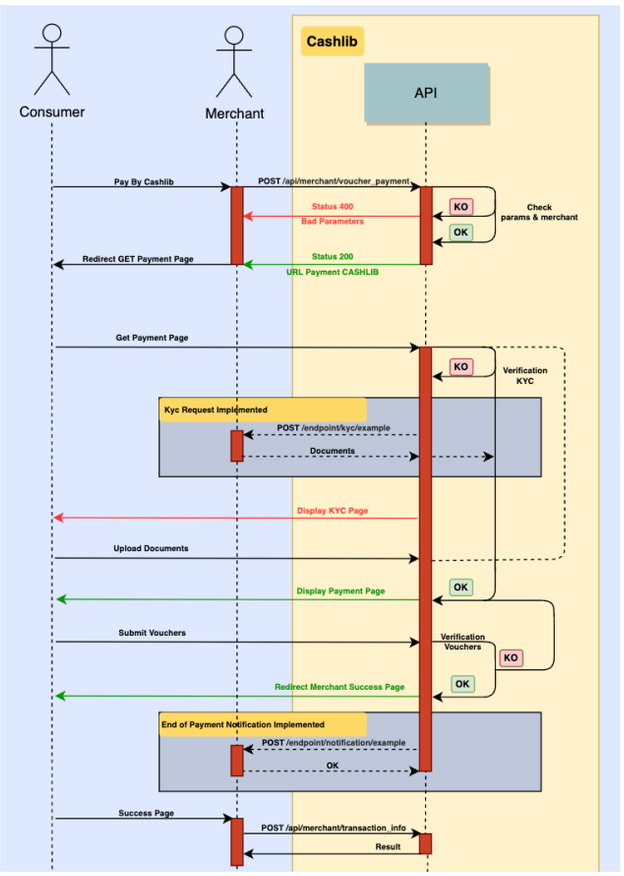
Transaction_Info permits to check a Transaction status. If the user quits (without cancelling) during the payment process, then the transaction may stay in "pending" status on the Merchant web site which should call this webservice to get actual transaction status.

Voucher_Directpayment permits to make a payment with up to 5 voucher codes via a server to server communication.

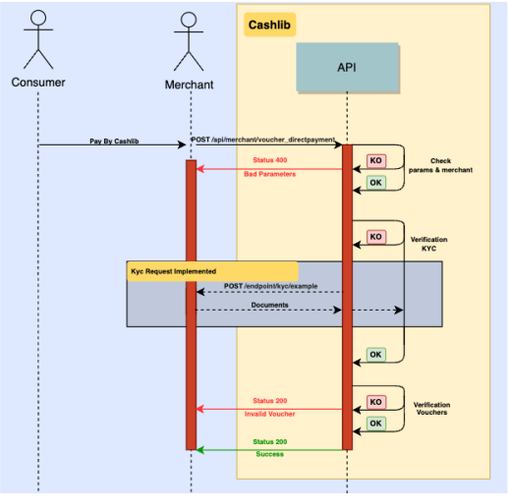
Kyc_Directpayment permits to know if the user can make the direct payment and if not to allow through an iframe to make his Kyc.

Cashlib Flow

FLOW Cashlib Iframe DOWNLOAD FLOW



FLOW Cashlib Direct Payment DOWNLOAD FLOW



Resource information

These APIs are based on standard REST-LIKE functions sending requests with POST methods. Responses are JSON formatted. They are protected with an API Key which should be included in the request header. 1 API Key is assigned to each E-Merchant by Cashlib.

In order to call the REST CASHlib APIs using POST method, a function has to be created. You have to use TLSv1.X to call the API.

Parameters are based on ISO standards

- Currency : ISO 4217 ; http://www.xe.com/iso4217.php
- Country code : ISO 3166-1 alpha-3 ; http://en.wikipedia.org/wiki/ISO_3166-1_alpha-3
- Characters : UTF-8
- Reference for dates is UTC/GMT +1 (Europe/Paris). Date format is « yyyy-mm-dd hh:ii:ss». Time is in 24H mode.

Error Codes

The CASHlib API attempts to return appropriate http status codes for every request.

In addition to HTTP error code, the CASHlib API contains functional error codes and messages. For example:

- HTTP Status Code: 400 Bad Request
- Response content: `{ "error_message" : "This voucher is not a test voucher.", "status" : 42}`

HTTP Status Code :

Codes	Text	Description
200	OK	Success
400	BAD REQUEST	Wrong Parameters
403	FORBIDDEN	Authentication Failed
404	NOT FOUND	The page / endpoint doesn't exist
500	INTERNAL SERVER ERROR	Technical Error

Codes :

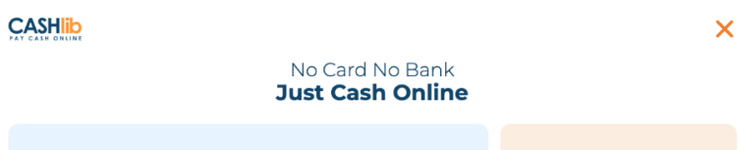
Codes	Text	Description
1	The customer must send us his KYC	Customer reached the legal limit and we need his KYC to continue
11	Your account has been suspended.	**mid** does exist but is not active
12	Could not authenticate you	Your API Key is missing or your API is not linked to the **mid**
13	Missing required parameter	
16	Duplicated transaction_id	transaction_id parameter already exists for this merchant.
17	Incorrect format parameter	
18	Too many requests received	
50	Not permitted.	IP address is blacklisted
52	Customer's IP address is inhibited.	IP Address already exists and is inhibited
53	Wrong Purchase amount	Purchase amount is greater Maximal than Transaction Amount
57	Payment IFRAME was canceled by customer	
58	Too many attempts to enter voucher code	
59	Timeout on IFRAME display	
60	Transaction already paid	
61	Error remaining Amount	
62	Invalid currency	Transaction currency is different than partner's currency.
70	Transaction not found	
80	Invalid voucher code	
81	Not sufficient funds	Cumulated voucher balance is below payment amount

Payment Process - Iframe

This appendix details the customer payment process after he has entered the CASHlib payment IFRAME.

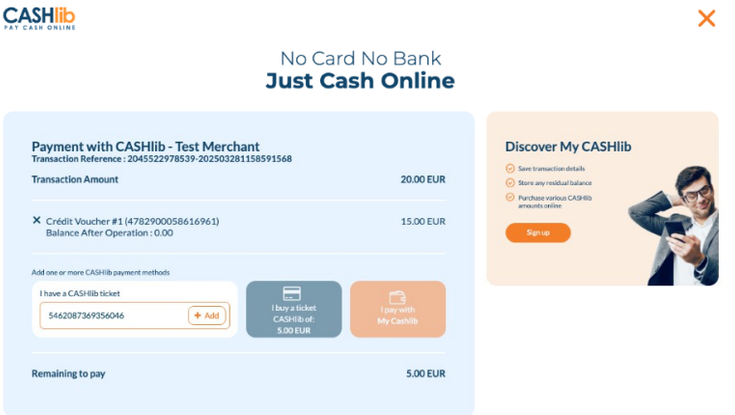
This is shown only for information since the entire payment process is managed by CASHlib IFRAME.

When the customer enters the CASHlib payment IFRAME, the following window is displayed with Merchant name and purchase amount. Customer may enter his voucher code and click on the « Add » button



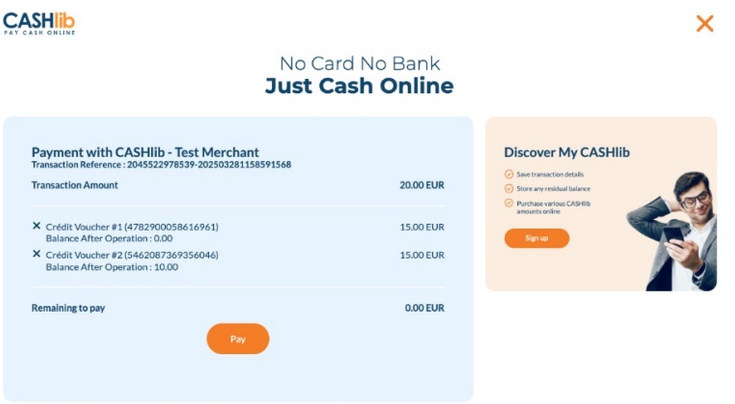
Then, the voucher code is included in the list of Vouchers. Voucher current balance and voucher futurebalance (when the payment is completed) are also displayed.

If the voucher current balance is lower than purchase amount then the customer may add up to 9 other voucher codes to the List of vouchers until the purchase amount is reached (ie "Still to be paidamount"= 0).



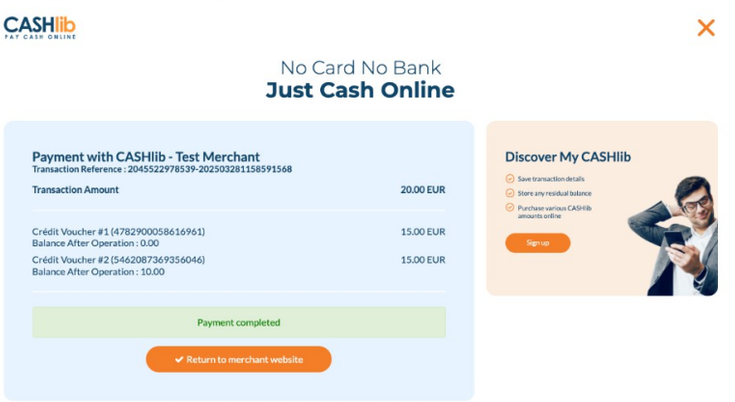
When the Purchase amount is reached then the "Add" voucher button is replaced with "Pay" buttonin order to actually make the payment with vouchers in the list.

If the user clicks on "Cancel and Return to merchant website" then the user is returned to the"cancel_url" you have sent in the voucher_payment request (with status information).



After clicking on "Pay" button, an "End of payment" notification is sent to the notification URL (if and only if this notification URL has been defined in the MID settings).

A "Voucher(s) debited" message is then displayed.



After the user clicks on "Click here to complete purchase", the user is redirected to success_url you have sent in the voucher_payment request (with status information).

Important notes:

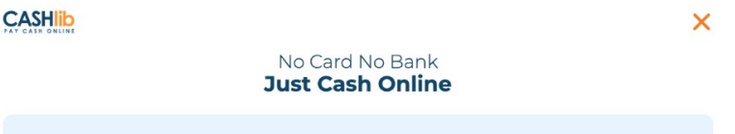
- In the merchant payment page design, User should NOT have the capability to quit CASHlib payment IFRAME without clicking on the buttons within the IFRAME
- CASHlib payment page may also be opened in "full page" mode. Then above recommendation is done.

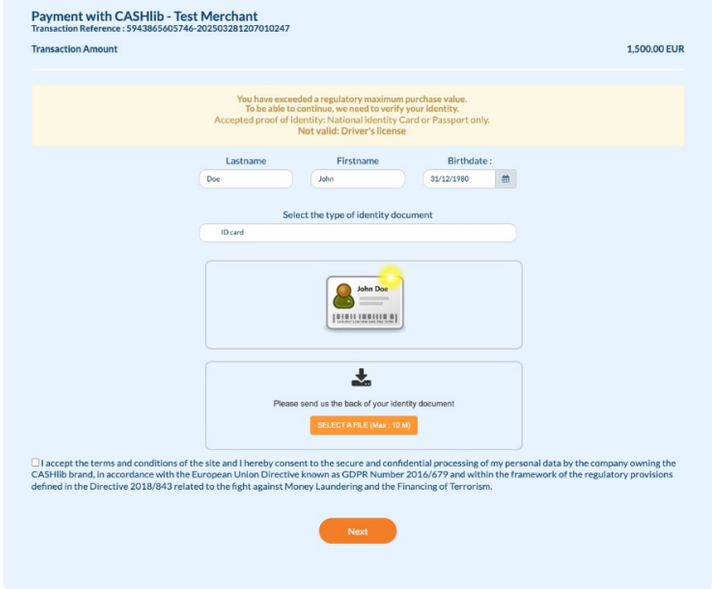
Kyc Process - Iframe

This appendix details the customer kyc process after he has entered the CASHlib payment IFRAME.

When a user reaches the payment limits, we need to have his KYC, and instead of displaying the payment page we ask for its documents.

Here is an example of a page that is displayed :





To become a VIP the user must provide in the case of CASHLIB Iframe his identity documents (ID card or passport) as well as his proof of address.
To become a PLATINIUM player the user must complete an information form regarding the source of wealth.

Tests to be made before going LIVE

In order to check there is no problem in your implementation of CASHlib merchant interface (voucher_payment webservice), we request the Merchant to make the following tests.

- 1. Run a CASHlib payment as fast as possible (prepare your voucher code before launching CASHlib payment)
- 2. Run a CASHlib payment, enter your voucher code and click on the "Pay" button. When the "Purchase details" page is displayed with "voucher(s) debited", do NOT click on the "Click here to complete purchase" button but close your web browser window.
- 3. Launch a CASHlib payment. When the "Purchase" page is displayed, do NOT enter any voucher code but close your web browser window

After the 3 tests are completed, then send us the references of the test transactions (transaction_id) and the status of these transactions that you have on YOUR server side.

With above test transactions, we can tell you whether your implementation of CASHlib API is correct.

Endpoints

Voucher Payment

This request permits to make a payment with a CASHlib voucher.

When the consumer has completed the payment process, his web browser is returned either to success_url or to cancel_url (URL sent by merchant in voucher_payment request) depending on whether the payment is successful or cancelled/failed.

Returned information to the given url (either success_url or cancel_url depending on transaction status) is the following. `{ 'transaction_id': '1234567890123456789 0', 'transaction_reference': '8268001197211-20230424155515070 2', 'status': 0, 'purchase_amount': 1000, 'currency': 'EUR' }`

HEADER PARAMETERS

apikey required	string	

REQUEST BODY SCHEMA: application/json
required

transaction_id required	string	<= 64 characters Merchant's transaction id
mid required	string	<= 64 characters Merchant identifier (Defined by CASHLIB)
purchase_amount required	integer	Amount In Cents (ex 1000 for 10€)
currency required	string	= 3 characters Currency iso3
ipaddress required	string	Consumer's IP address
name required	string	<= 255 characters Consumer's name
firstname required	string	<= 255 characters Consumer's firstname
birthdate required	string	Consumer's birthdate (yyyy-mm-dd)
email required	string	<= 255 characters Consumer's email
phone_number required	string	<= 14 characters Consumer's phone number
address required	string	<= 255 characters Consumer's Address
zipcode required	string	<= 14 characters Consumer's Zipcode
city required	string	<= 255 characters Consumer's City
country required	string	= 3 characters Consumer's country of residence
success_url required	string	<= 255 characters Url Redirection after success payment
cancel_url required	string	<= 255 characters Url Redirection after cancel payment

POST /api/merchant/voucher_payment

Request samples

Payload

Content type
application/json

Copy


```
{
   "transaction_id": "1234567890123456789 0",
   "mid": "20230101XX",
   "purchase_amount": 1000,
   "currency": "EUR",
   "ipaddress": "192.168.1.1",
   "name": "Doe",
   "firstname": "John",
   "birthdate": "1970-12-31",
   "email": "john.doe@mail.com",
   "phone_number": "+44123456789",
   "address": "41 Boulevard Royal",
   "zipcode": "L-2449",
   "city": "Luxembourg",
   "country": "FRA",
   "success_url": "https://merchant-page-example.com/redirect/success
   "cancel_url": "https://merchant-page-example.com/redirect/cancel",
   "redirect_target": "_self"
}
```


Response samples

200 400

Content type
application/json

Copy


```
{
   "transaction_id": "1234567890123456789 0",
   "transaction_reference": "8268001197211-20230424155515070 2",
   "status": 0,
   "error_message": "",
   "iframe_url": "https://api-test.cashlib.com/purchase/S5s8wZyaPqD0a
}
```

redirect_target — string
Enum: "_self" "_parent" "_top"
Type of redirection

Responses

> 200 successful operation

> 400 bad request

Transaction Info

Transaction_Info permits to check a transaction status. If the user quits (without cancelling) during the payment process, then the transaction may stay in "pending" status on the Merchant web site which should call this webservice to get actual transaction status.

Please note that you should not call this webservice during more than 1 hour after the transaction started and you have to wait at least 10 minutes between each call.

HEADER PARAMETERS

apikey (required) — string

REQUEST BODY SCHEMA: application/json
required

transaction_id (required) — string <= 64 characters
Merchant's transaction id

mid (required) — string <= 64 characters
Merchant identifier (Defined by CASHLIB)

Responses

> 200 successful operation

> 400 bad request

POST /api/merchant/transaction_info

Request samples

Payload

Content type
application/json

Copy

```
{
   "transaction_id": "1234567890123456789O",
   "mid": "20230101XX"
}
```

Response samples

200 400

Content type
application/json

Copy

```
{
   "transaction_id": "1234567890123456789O",
   "transaction_reference": "8268001197211-20230424155515070 2",
   "status": 0,
   "error_message": "",
   "transaction_status": "Pending",
   "purchase_amount": 1000,
   "currency": "EUR"
}
```

Voucher Direct Payment Deprecated

Voucher_directpayment permits to make a payment with up to 5 voucher codes via a server to server communication.

HEADER PARAMETERS

apikey — string

REQUEST BODY SCHEMA: application/json
required

transaction_id (required) — string <= 64 characters
Merchant's transaction id

test — string <= 1 characters
Y if test transaction

mid (required) — string <= 64 characters
Merchant identifier (Defined by CASHLIB)

amount (required) — integer
Amount In Cents (ex 1000 for 10€)

currency (required) — string = 3 characters
Currency iso3

ipaddress (required) — string
Consumer's IP address

name (required) — string <= 255 characters
Consumer's name

firstname (required) — string <= 255 characters
Consumer's firstname

birthdate (required) — string
Consumer's birthdate (yyyy-mm-dd)

email (required) — string <= 255 characters
Consumer's email

phone_number (required) — string <= 14 characters
Consumer's phone number

address (required) — string <= 255 characters
Consumer's Address

zipcode (required) — string <= 14 characters
Consumer's Zipcode

city (required) — string <= 255 characters
Consumer's City

country (required) — string = 3 characters
Consumer's country of residence

voucher_code_1 (required) — string = 16 characters
1st voucher code 16 digits

voucher_code_2 — string = 16 characters
2nd voucher code 16 digits

voucher_code_3 — string = 16 characters
3rd voucher code 16 digits

voucher_code_4 — string = 16 characters
4th voucher code 16 digits

voucher_code_5 — string = 16 characters
5th voucher code 16 digits

Responses

POST /api/merchant/voucher_directpayment

Request samples

Payload

Content type
application/json

Copy

```
{
   "transaction_id": "1234567890123456789O",
   "test": "N",
   "mid": "20230101XX",
   "amount": 1000,
   "currency": "EUR",
   "ipaddress": "192.168.1.1",
   "name": "Doe",
   "firstname": "John",
   "birthdate": "1970-12-31",
   "email": "john.doe@mail.com",
   "phone_number": "+44123456789",
   "address": "41 Boulevard Royal",
   "zipcode": "L-2449",
   "city": "Luxembourg",
   "country": "FRA",
   "voucher_code_1": "1234567890123456",
   "voucher_code_2": "1234567890123456",
   "voucher_code_3": "1234567890123456",
   "voucher_code_4": "1234567890123456",
   "voucher_code_5": "1234567890123456"
}
```

Response samples

200 400

Content type
application/json

Example
VoucherDirectPaymentResponseSuccess

Copy

```
{
   "status": 0,
   "transaction_reference": "8268001197211-20230424155515070 2",
   "transaction_id": "1234567890123456789O",
   "error_message": "",
   "purchase_amount": 1000,
   "currency": "EUR"
}
```

> 200 successful operation

> 400 bad request

KYC Direct Payment

Kyc_directpayment permits to know if the customer can do the transaction, if he reached compliance limit this return an url to allow the customer to send the KYC to CASHLIB.

HEADER PARAMETERS

- `apikey` — string

REQUEST BODY SCHEMA: application/json
required

- `mid` (required) — string <= 64 characters
  Merchant identifier (Defined by CASHLIB)
- `purchase_amount` (required) — integer
  Amount In Cents (ex 1000 for 10€)
- `currency` (required) — string = 3 characters
  Currency iso3
- `ipaddress` (required) — string
  Consumer's IP address
- `name` (required) — string <= 255 characters
  Consumer's name
- `firstname` (required) — string <= 255 characters
  Consumer's firstname
- `birthdate` (required) — string
  Consumer's birthdate (yyyy-mm-dd)
- `email` (required) — string <= 255 characters
  Consumer's email
- `phone_number` (required) — string <= 14 characters
  Consumer's phone number
- `redirect_url` — string <= 255 characters
  Url Redirection after kyc process done
- `redirect_target` — string
  Enum: `"_self"` `"_parent"` `"_top"`
  Type of redirection

Responses

> 200 successful operation

> 400 bad request

POST /api/merchant/kyc_directpayment

Request samples

Payload

Content type
application/json

Copy

```
{
    "mid": "20230101XX",
    "purchase_amount": 1000,
    "currency": "EUR",
    "ipaddress": "192.168.1.1",
    "name": "Doe",
    "firstname": "John",
    "birthdate": "1970-12-31",
    "email": "john.doe@mail.com",
    "phone_number": "+44123456789",
    "redirect_url": "https://merchant-page-example.com/redirect/kyc",
    "redirect_target": "_self"
}
```

Response samples

200 400

Content type
application/json

Copy

```
{
    "status": 0,
    "error_message": "No Kyc required."
}
```

Current Balance

Current Balance to get current balance.

HEADER PARAMETERS

- `apikey` — string

REQUEST BODY SCHEMA: application/json
required

- `mid` (required) — string <= 64 characters
  Merchant identifier (Defined by CASHLIB)

Responses

> 200 successful operation

> 400 bad request

POST /api/merchant/balance

Request samples

Payload

Content type
application/json

Copy

```
{
    "mid": "20230101XX"
}
```

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": 0,
  - "result": {
        "balance": 105000,
        "balance_billed": 60000,
        "balance_not_billed": 45000,
        "fees_balance_not_billed": 5000
    }
}
```

Endpoints to implement

These is the endpoints to be implemented by the merchant

End of payment Notification

When payment is completed, an "End of payment" notification is sent to the notification URL if and only if the notification URL has been defined in the MID settings.
This notification includes the same parameters as sent to the success_url.

HEADER PARAMETERS

- `apikey` — string

REQUEST BODY SCHEMA: application/json
required

POST /endpoint/notification/example

Request samples

Payload

Content type
application/json

Copy

```
{
    "status": 0,
```

status required	integer	Status 0 = SUCCESS
error_message	string	Error message blank = SUCCESS
transaction_id required	string	Merchant's transaction id
transaction_reference required	string	CASHlib transaction uid
purchase_amount required	integer	Amount In Cents (ex 1000 for 10€)
currency required	string = 3 characters	Currency iso3

Responses

> 200 successful operation

```
"error_message": "",
"transaction_id": "1234567890123456789O",
"transaction_reference": "8268001197211-20230424155515O702",
"purchase_amount": 1000,
"currency": "EUR"
}
```

Response samples

200

Content type
application/json

Copy

```
"OK"
```

Kyc Request

When a customer reached limits when calling **voucher_payment** or **voucher_directpayment**, a "Kyc request" is sent to ask the KYC for the customer if and only if the notification URL has been defined in the MID settings. If KYCs are not retrieved in this way, the user will be asked to fill in his KYCs during the payment process.

HEADER PARAMETERS

apikey	string

REQUEST BODY SCHEMA: application/json
required

name required	string <= 255 characters	Consumer's name
firstname required	string <= 255 characters	Consumer's firstname
birthdate required	string	Consumer's birthdate (yyyy-mm-dd)
email required	string <= 255 characters	Consumer's email
transaction_id required	string	Merchant's transaction id
transaction_reference required	string	CASHlib transaction uid

Responses

> 200 successful operation

POST /endpoint/kyc/example

Request samples

Payload

Content type
application/json

Copy

```
{
    "name": "Doe",
    "firstname": "John",
    "birthdate": "1970-12-31",
    "email": "john.doe@mail.com",
    "transaction_id": "1234567890123456789O",
    "transaction_reference": "8268001197211-20230424155515O702"
}
```

Response samples

200

Content type
application/json

Copy

```
{
    "id_recto": "iVBORw0KGgoAAAANSUhEUgAAAAgAAAAIAQMAAAD+wSzIAAAABlBMV
    "id_verso": "iVBORw0KGgoAAAANSUhEUgAAAAgAAAAIAQMAAAD+wSzIAAAABlBMV
    "selfie": "iVBORw0KGgoAAAANSUhEUgAAAAgAAAAIAQMAAAD+wSzIAAAABlBMVEX
    "proof_of_address": "iVBORw0KGgoAAAANSUhEUgAAAAgAAAAIAQMAAAD+wSzIA
}
```