KAJA Payments

Our API is organized around REST. It has predictable resource-oriented URLs, accepts form-encoded request bodies, returns JSON-encoded responses, and uses standard HTTP response codes, authentication, and verbs.

General Information

Authentication

Our API uses **Access Token** to authenticate requests. You can view and manage your **Access Token** in the Dashboard.

The method we use for authentication is Bearer Authentication

All API requests must be made over HTTPS. Calls made over plain HTTP will fail. API requests without authentication will also fail.

> Warning `Don't share your Access Token with anybody.`

Bearer Authentication

Bearer Authentication is an HTTP authentication scheme that involves security tokens called bearer tokens. The client must send this token in the Authorization header when making requests to protected resources.

`Authorization: Bearer`

Errors

We use conventional HTTP response codes to indicate the success or failure of an API request.

The `2XX` range indicates success.

The `2XX` range indicates success.

The `4XX` range indicates an error that failed given the information provided (e.g., a required parameter was omitted, etc.).

The `5XX` range indicates an error with our servers (these are rare).

In addition, `name`, `message`, `code`, and `status` will be applied to the error response.

Example Response

```json
{
    "name":"Unprocessable entity",
    "message":"Amount cannot be blank.",
    "code":422,
    "status":422
}
```

Responses

Our response will be divided into 3 sections:

`items` - contains the actual result of your request

`_meta` - contains meta information of your request like total results, number of pages, etc.

`_links` - contains links to the next page, last page, etc.

```json
{
    "items": [
        // Results goes here
    ],
    "_links": {
        "self": {
            "href": "/path?page=1&per-page=10"
        },
        "next": {
            "href": "/path?page=2&per-page=10"
```

View More

Filters

Filters will help you find what you search.

Parameters	Description
lt	Stands for the less-than sign <
gt	Stands for the greater-than sign >
lte	Stands for the less-than or equals sign <=
gte	Stands for the greater-than or equals sign >=
eq	Stands for the equals sign =
neq	Stands for the not equals sign !=
in	IN
nin	NOT IN
and	AND

View More

Use `NULL` as a value if you want to select empty values.

> Only fields marked with `Filterable` can be used for filtering the results.

AUTHORIZATION Bearer Token

Token	XXX

GET Example - gte

https://example.kajadashboard.com/path?filter[createdOn][gte]=2023-01-01 00:00:00

Find all records created after `2023-01-01 00:00:00`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Filters

PARAMS

filter[createdOn][gte]	2023-01-01 00:00:00

Example Request Example - gte

Example Request Example - gte

curl

```
curl --location -g 'https://example.kajadashboard.com/path?filter[createdOn][gte]=2023-01-01%2000%3A00%3A00'
```

Example Response

Body Headers (0)

No response body
This request doesn't return any response body

GET Example - between

https://example.kajadashboard.com/path?filter[and][0][createdOn][gte]=2019-01-01 00:00:00&filter[and][0][createdOn][lte]=2019-01-01 23:59:59

Find all records created between `2019-01-01 00:00:00` and `2019-01-01 23:59:59`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Filters

This request is using Bearer Token from folder Filters

PARAMS

filter[and][0][createdOn][gte]	2019-01-01 00:00:00
filter[and][0][createdOn][lte]	2019-01-01 23:59:59

Example Request Example - between

curl

```
curl --location -g 'https://example.kajadashboard.com/path?f…  View More  …][createdOn][gte]=2019-01-01%2000%3A00%3A00&filter[and][0][cre
```

Example Response

Body Headers (0)

No response body
This request doesn't return any response body

GET Example - like

GET Example - like

https://example.kajadashboard.com/path?filter[customer.email][like]=example.com

Find all records whose email contains `example.com`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Filters

PARAMS

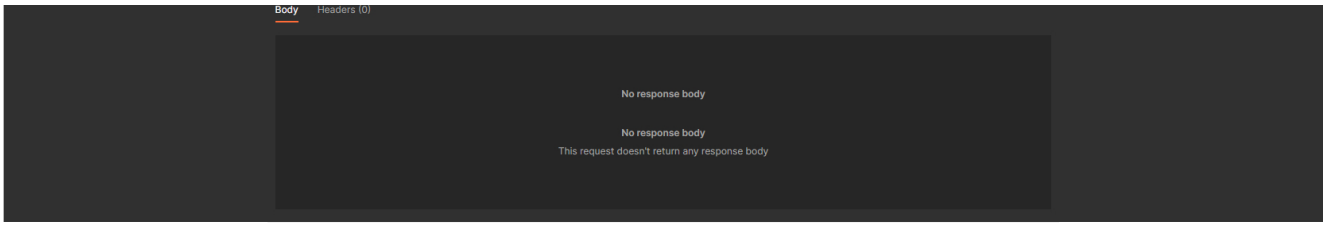
filter[customer.email][like]	example.com

Example Request Example - like

curl

```
curl --location -g 'https://example.kajadashboard.com/path?filter[customer.email][like]=example.com'
```

Example Response



GET Example - in

https://example.kajadashboard.com/path?filter[id][in][]=1&filter[id][in][]=2

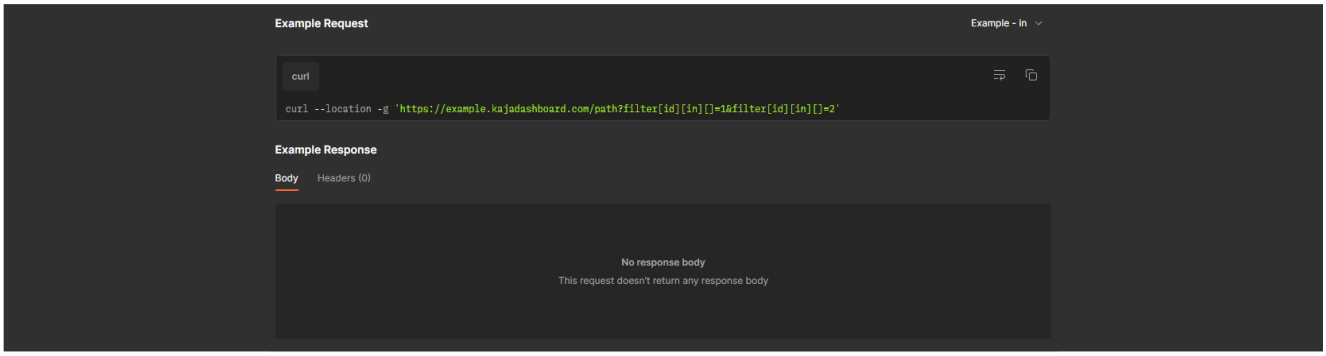
Find all records with IDs `1` or `2`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Filters

PARAMS

filter[id][in][]	1
filter[id][in][]	2



GET Example - non-empty

https://example.kajadashboard.com/path?filter[not][id]=NULL

Find all records with ID with non-empty values (ie. NOT NULL)

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Filters

PARAMS

filter[not][id]	NULL

Pagination

If there are too many results of your request, we will enable pagination of the results.
Two parameters can control pagination `page` and `per-page`.

`page` - The page number you want to get. By default is `1`.

`per-page` - Items per page. The value can be between `1` and `50`. By default is `30`.

`per-page` - Items per page. The value can be between `1` and `50`. By default is `30`.

AUTHORIZATION Bearer Token

Token	XXX

GET 10 results per page

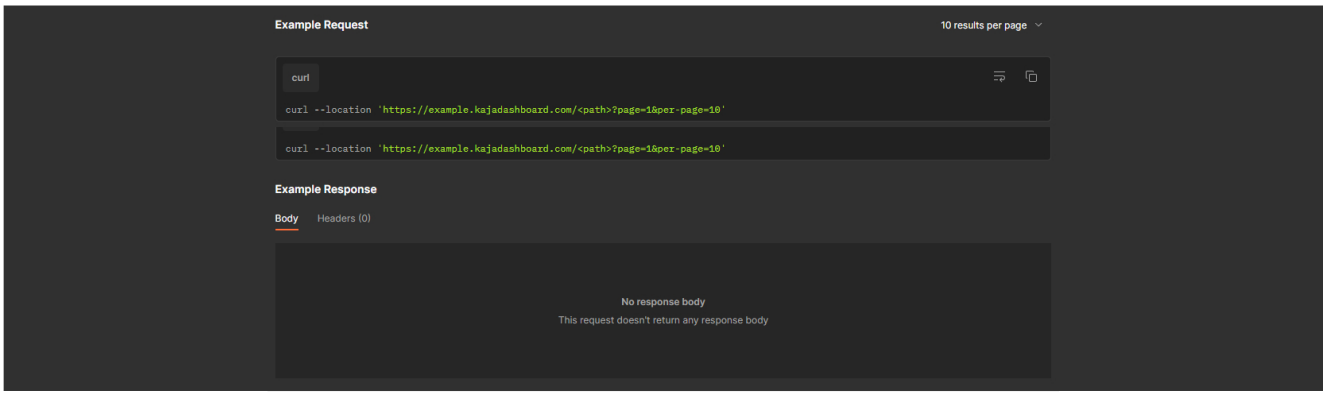
https://example.kajadashboard.com/<path>?page=1&per-page=10

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Pagination

PARAMS

page	1
per-page	10



Data Types

Type	Description
Boolean	`Yes` or `No`
Integer	A number which is not a fraction; a whole number. `1`, `2`, `3`...
Number	Floating point numbers. `1.5`, `2.9`, `3.59`...
String	A string is a collection of characters, can contain alphanumeric characters.
ENUM	Enumeration type or ENUM for short, is a data type to categorize named values.
ENUM	Enumeration type or ENUM for short, is a data type to categorize named values.
Date	Date is in the following format `YYYY-MM-DD` (Example: 2019-02-17) Always in `UTC` time zone
Datetime	Date and time is in the following format `YYYY-MM-DD HH:MM:SS` (Example: 2019-02-17 20:05:34) Always in `UTC` time zone.
Time	Time is in the following format `HH:MM:SS` (Example: 20:05:34)

View More

Merchants

AUTHORIZATION Bearer Token

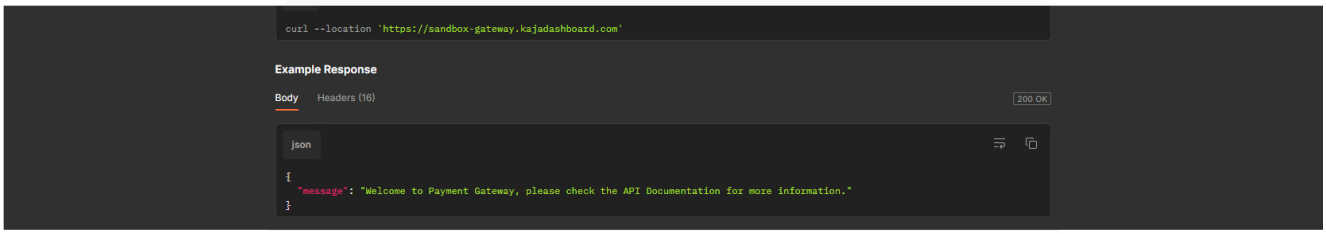
Token	XXX

Sandbox

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET Sandbox Payment Gateway



Card Payments

> In the payment gateway, your requests need to always include the system `terminal.id`. Within itself the system `terminal.id` contains DBA, MID, and terminal TID information, this way we always ensure that you are using the correct MID and TID if you have more than one.

AUTHORIZATION Bearer Token

Token	XXX

Card Payment Webhooks

AUTHORIZATION Bearer Token

Card Payment Webhooks

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Card Payments

Payment Transaction Created

Notification upon the creation of a payment transaction.

json

```
{
    ...
    "module": "transaction",
    "action": "create",
    ...
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Card Payments

Refund Transaction Created

Notification upon refund creation.

json

json

```
{
    ...
    "module": "transaction",
    "action": "refund",
    ...
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Card Payments

Recurring Transaction Created

Notification upon recurring creation.

json

```
{
    ...
    "module": "transaction",
    "action": "reccuring",
    ...
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Card Payments

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Card Payments

POST Authorization

https://gateway.kajadashboard.com/payment/auth

The Auth method initializes an authorization-only transaction. Authorization-only transactions are not included in the batch for settlement.

A separate capture request must be submitted in order for the transaction to be settled.

> The suggested authorization period is **10 days**. You need to capture a payment within the authorization period to collect the funds of the transaction. Note, the authorization period can vary by issuer and/or card brand, it is your responsibility to ensure whatever window period is appropriate for your use case.

> **It is important to note that these requests need to always include the system `terminal.id` for which you are making a payment.**

Object

Fields	Description	Type	Filterable	Required	Readonly	Level 1	Level 2	Level 3
id	Unique ID of the payment record	Integer	Yes	No	Yes	No	No	No
terminal	Terminal Object	Object	No	Yes	No			
terminal.id	Dashboard Terminal ID	Integer	Yes	Yes	No	Yes	Yes	Yes
amount	Payment amount	Number	No	Yes	No	Yes	Yes	Yes
source	Transaction amount	ENUM	Yes	Yes	No	Yes	Yes	Yes
source	Transaction source `Internet`, `Phone` and `Mail`	ENUM	Yes	Yes	No	Yes	Yes	Yes
origin	Origin of the transaction (i.e. Website, CRM, etc)	String	Yes	No	No	No	No	No
level	Desired level	ENUM	Yes	No	No	Yes	Yes	No

View More

> If you pass the `level` parameter in your request, we will validate the field requirements of the level you have requested. In the event of an incomplete validation we will return an error with the missing fields.

> If a transaction has a `batch` information in the response, this means that this transaction has batched.

Products Object

Fields	Description	Type	Filterable	Required	Readonly	Level 1	Level 2	Level 3
id	Product ID	String	No	No	No	No	No	Yes
name	Product Name	String	No	No	No	No	No	Yes
price	Product Price	Number	No	No	No	No	No	Yes
quantity	Product Quantity	Integer	No	No	No	No	No	Yes
measurementUnit	Product Measurment Unit	String	No	No	No	No	No	Yes
commodityCode	Product commodity code. It should contain a maximum of 12 characters. Only letters, numbers and	String	No	No	No	No	No	Yes

View More

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

json

```
{
  "terminal": {
    "id": 24
  },
  "amount": "1.99",
  "source": "Internet",
  "level": 1,
  "card": {
    "name": "John Wick",
    "number": 5309572686772257,
```

View More

Example Request Auth

curl

```
curl --location 'https://gateway.kajadashboard.com/payment/auth' \
--data-raw '{
  "terminal": {
    "id": 24
  },
  "amount": "1.99",
  "source": "Internet",
  "level": 1,
  "card": {
    "name": "John Wick",
```

View More

Example Response

Body Headers (1) 200 OK

json

```
{
  "id": 876,
  "amount": "1.99",
  "origin": null,
  "type": "Auth",
  "level": 1,
  "authCode": "VTLMC1",
  "parent": {
    "id": null
  },
```

View More

POST Sale

https://gateway.kajadashboard.com/payment/sale

https://gateway.kajadashboard.com/payment/sale

The Sale method creates a new authorization and a capture request by default.

> It is important to note that these requests need to always include the system `terminal.id` for which you are making a payment.

Sale transactions are settled automatically while auth-only transactions are not included in the batch for settlement.

> Keep in mind the required fields in order to initiate a sale transaction. If you want to perform level 2 or level 3 the required fields are outlined below.

Fields	Level 2	Level 3	Visa	Mastercard	AMEX (Level 2 Only)
order.id	No	Yes	Yes	Yes	No
order.date	No	Yes	Yes	No	No
order.invoice	No	Yes	Yes	No	No
order.summaryCommodityCode	No	Yes	Yes	No	No
order.description	No	No	No	No	Yes
order.supplierRefNo	No	No	No	No	Yes
order.customer.id	No	No	No	No	Yes
order.customer.vat	No	Yes	Yes	No	No
order.shipping.fromZip	No	Yes	Yes	Yes	No
order.shipping.toZip	No	Yes	Yes	Yes	Yes
order.shipping.toCountry	No	Yes	Yes	Yes	No
order.shipping.cost	No	Yes	Yes	Yes	No

View More

Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the payment record	Integer	Yes	No	Yes
terminal	Terminal Object	Object	No	Yes	No
terminal.id	Dashboard Terminal ID	Integer	Yes	Yes	No
amount	Payment amount	Number	No	Yes	No
source	Transaction source `Internet`, `Phone` and `Mail`	ENUM	Yes	Yes	No
origin	Origin of the transaction (i.e. Website, CRM, etc)	String	Yes	No	No
level	Desired level of the payment `1`, `2` or `3`	ENUM	Yes	Yes	No
threeds	3DS Object	Object	No	Yes	No
threeds.id	The ID that is	Integer	View More	Yes	No

If you pass the `level` parameter in your request, we will validate the field requirements of the level you have requested. In the event of an incomplete validation we will return an error with the missing fields.

If a transaction has a `batch` information in the response, this means that this transaction has batched.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

json


```
{
  "terminal": {
    "id": 24
  },
  "amount": "2.10",
  "source": "Internet",
  "level": 1,
  "card": {
    "name": "John Wick",
    "number": 5309572686772257,
```


View More

Example Request Sale

curl


```
curl --location 'https://gateway.kajadashboard.com/payment/sale' \
--data-raw '{
  "terminal": {
    "id": 24
  },
  "amount": "2.10",
  "source": "Internet",
  "level": 1,
  "card": {
    "name": "John Wick",
```


View More

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": 874,
  "amount": "2.10",
  "origin": null,
  "type": "Auth",
  "level": 1,
  "authCode": "VTLMC1",
  "parent": {
    "id": null
  },
```


View More

POST Capture

https://gateway.kajadashboard.com/payment/<id>/capture

The capture method marks an Auth transaction for settlement, therefore the transaction ID is needed. This method also allows adjustments of the transaction amount.

If your request does not include the `amount`, our system will settle the full transaction with no changes in the `amount` field and vice versa.

You can perform **multiple capture requests** for a specific auth transaction until you reach the authorized amount. After that, you must submit a new authorization request to capture additional amounts.

Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the capture record	Integer	Yes	No	Yes
terminal	Terminal Object	Object	No	Yes	No
terminal.id	Dashboard Terminal ID	Integer	Yes	Yes	No
amount	Amount of the capture	Number	No	No	No
partial	Partial Object	Object	No	No	No
partial.total	Total payments	Integer	No	No	No
partial.sequence	Current payment sequence number	Integer	No	No	No
sendReceipt	Send receipt	Integer	No	No	No

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

```
{
    "amount": "1.99",
    "partial": {
        "total": null,
        "sequence": null
    }
}
```

Example Request Capture

```
curl --location --request POST 'https://gateway.kajadashboard.com/payment/876/capture' \
--data ''
```

Example Response

Body Headers (1) 200 OK

```
{
  "amount": "1.99",
  "partial": {
    "total": null,
    "sequence": null
  }
}
```

POST Partial Capture

https://gateway.kajadashboard.com/payment/<id>/capture

We provide the functionality to create partial captures.

If `partial` is included in the request the gateway marks the specified transaction amount for settlement.

We provide the functionality to create partial captures.

If `partial` is included in the request the gateway marks the specified transaction amount for settlement.

> When performing partial capture it is required to pass the `sequence` and `total` to indicate how many partial captures will be performed.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

```
{
    "terminal": {
        "id": 24
    },
    "amount": 0.98,
    "partial": {
        "sequence": "1",
        "total": "2"
    }
}
```

Example Request Partial Capture

```
curl --location 'https://gateway.kajadashboard.com/payment/capture/880' \
--header 'Content-Type: application/json' \
--data '{
    "terminal": {
        "id": 24
    },
    "amount": 0.90,
    "partial": {
        "sequence": "1",
        "total": "2"
```

View More

Example Response

Body Headers (1) 200 OK

```
{
  "amount": "0.90",
  "partial": {
    "total": "2",
    "sequence": "1"
  }
}
```

POST Refund

https://gateway.kajadashboard.com/payment/<id>/refund

The refund method is used to refund/void an authorized or settled transaction. This method will void or reverse the transaction based on the card network.

> **Important:** We do not allow refunds without `transaction.id` and an `amount` that exceeds the captured of the original transaction.

We support additional data for a refund request. If no amount is specified in the request, the system will return the full amount of the original transaction. You need to

We support additional data for a refund request. If no amount is specified in the request, the system will return the full amount of the original transaction. You need to provide any discount and tax amount that was in the original transaction to perform a partial refund.

> **Important:** In a refund request if the `amount` is present but the `order.tax` and `order.discounts` amounts are not present, the `amount` must be less than or equal to the subtotal amount of the original order.

The system will create a new transaction id for all refund transactions.

- If the original transaction has been settled, the new transaction id created by the return will be processed offline and will not generate an auth code.
- If the original transaction has not been settled, we link the original transaction to the return transaction. The gateway automatically reverses the transaction and an auth code will be generated.

Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the refund record	Integer	Yes	No	Yes
terminal	Terminal Object	Object	No	Yes	No
terminal.id	Dashboard Terminal ID	Integer	Yes	Yes	No
amount	Amount of the refund	Number	No	No	No

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

json

```
{
    "id": 883,
    "amount": "1.99",
    "origin": "API",
    "type": "Return",
    "level": 1,
    "authCode": null,
    "parent": {
        "id": 882
    },
```

View More

Example Request Refund

curl

```
curl --location --request POST 'https://gateway.kajadashboard.com/payment/882/refund
' \
--header 'Content-Type: application/json' \
--data ''
```

Example Response

Body Headers (1) 200 OK

json

```
{
  "id": 883,
  "amount": "1.99",
  "origin": "API",
  "type": "Return",
  "level": 1,
  "authCode": null,
  "parent": {
    "id": 882
  },
```

View More

POST Partial Refund

https://gateway.kajadashboard.com/payment/<id>/refund

You are allowed to perform a partial refund by providing a lower `amount` than the original captured transaction if no `amount` is provided the transaction will be fully refunded. You need to provide any discount and tax amount that was in the original transaction to perform a partial refund.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

json

```
{
  "terminal": {
    "id": 24
  },
  "amount": "1.10",
  "source": "Internet",
  "level": 1,
  "card": {
    "name": "John Wick",
    "number": 514631220000035,
```

View More

```
    "name": "John Wick",
    "number": 514631220000035,
```

View More

Example Request Partial Refund

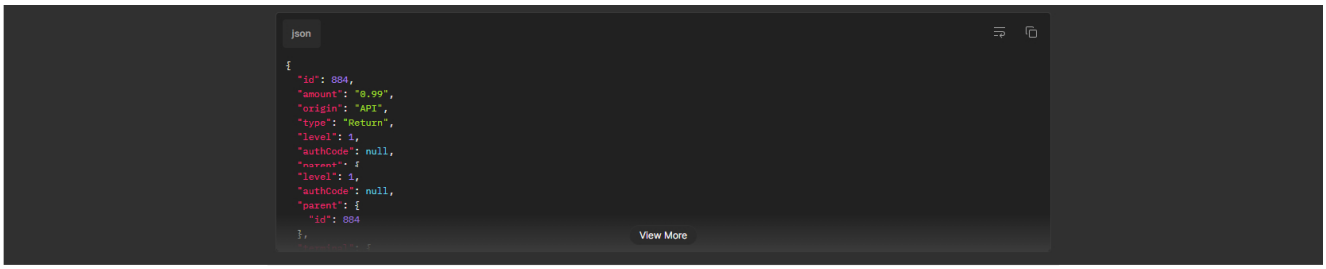
curl

```
curl --location 'https://gateway.kajadashboard.com/payment/884/refund
' \
--data '{
  "terminal": {
    "id": 24
  },
  "amount": "1.10",
  "source": "Internet",
  "level": 1,
  "card": {
```

View More

Example Response

Body Headers (1) 200 OK



POST Soft/Dynamic Descriptor

https://gateway.kajadashboard.com/payment/sale

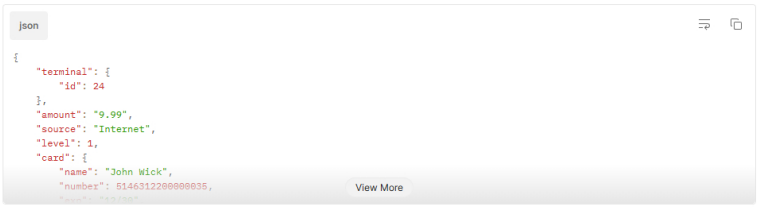
Soft Descriptor or Dynamic Descriptor functionality gives you the ability to modify the statement descriptor on a per-transaction basis. Using this feature, you gain the ability to change this information according to your discretion and needs.

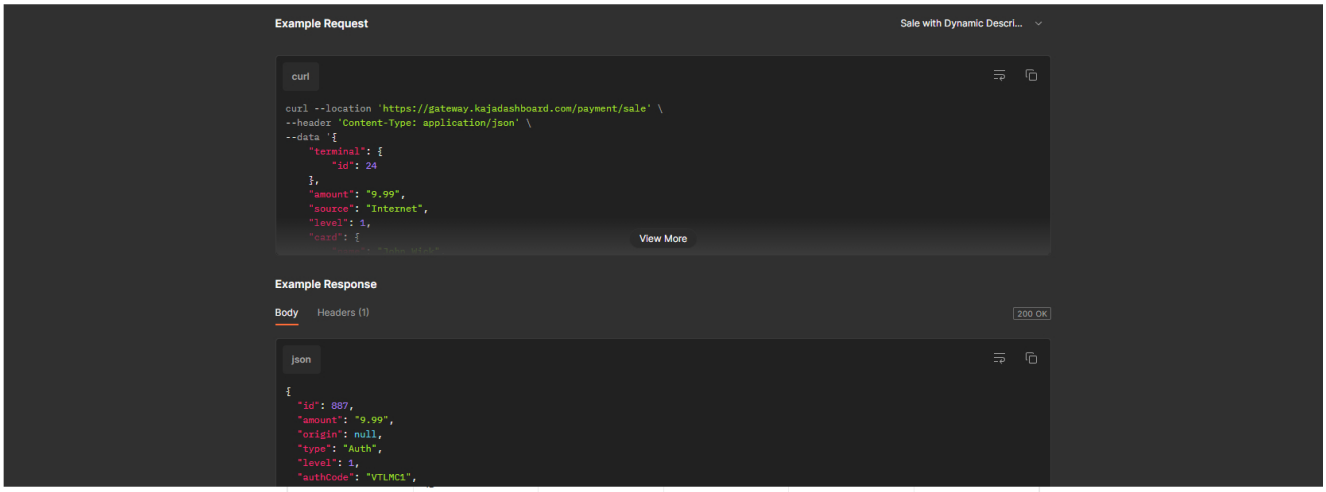
> In order to use this functionality always include `order.description` in your **Sale** or **Auth** requests.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)





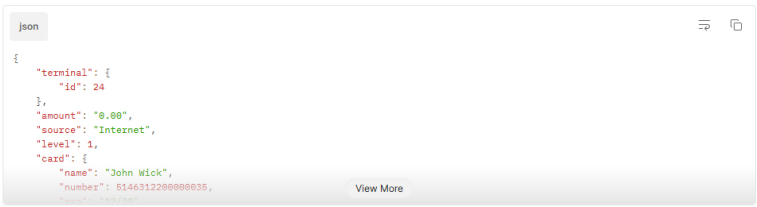
amount	Amount of the payment	Number	No	Yes	No
source	Where the card details come from `Internet` or `MOTO`	ENUM	Yes	Yes	No
card	Card details	Object	No	Yes	No
card.name	Cardholder name	String	No	Yes	No
card.number	Card number	Integer	No	Yes	No
card.exp	Card expiration date `mm/yy`	String	No	Yes	No
card.save	Save the card details	ENUM		Yes	No

View More

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)



```
curl --location 'https://gateway.kajadashboard.com/payment/card-authentication' \
--header 'Content-Type: application/json' \
--data '{
    "terminal": {
        "id": 24
    },
    "amount": "0.00",
    "source": "Internet",
    "level": 1,
    "card": {
```

View More

Example Response

Body Headers (1) 200 OK

```
{
  "amount": "0.00",
  "partial": {
    "total": null,
    "sequence": null
  }
}
```

card.name	Cardholder name	String	No	Yes	No
card.number	Card number	Integer	No	Yes	No
card.exp	Card expiration date `mm/yy`	String	No	Yes	No
card.cvv	Card security code	Integer	No	Conditional	No

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

```
{
    "terminal": {
        "id": 24
    },
    "source": "Internet",
    "card": {
        "name": "John Wick",
        "number": 5146312200000035,
        "exp": "12/30",
        "cvv": "998"
```

View More

Example Request Card Tokenization

```
curl --location 'https://gateway.kajadashboard.com/payment/generate-token' \
--header 'Content-Type: application/json' \
--data '{
    "terminal": {
        "id": 24
    },
    "source": "Internet",
    "card": {
        "name": "John Wick",
        "number": 5146312200000035,
```

View More

Example Response

Body Headers (1) 200 OK

```
{
  "id": "890",
  "amount": 0,
  "origin": "API",
  "type": "GenerateToken",
  "level": 1,
  "authCode": null,
  "parent": {
    "id": null
  },
```

View More

POST Card Tokenization with Global Token

https://gateway.kajadashboard.com/payment/token-exchange

https://gateway.kajadashboard.com/payment/token-exchange

Currently we support two types of card tokens. Regular tokens and `Global Tokens`. The difference between the regular and the global token is that the regular token **can only be used on the terminal on which the card was tokenized** whereas the Global Token can be used across all of your terminals.

By default a card is tokenized with a regular token specifically issued for the terminal on which the payment was made. In order to receive `Global Tokens` instead of regular tokens the setting must be explicitly enabled in your terminal settings.

If you have regular token/s which you want to exchange for global tokens (or vice versa) you can use the **token-exchange** endpoint.

> To exchange a regular token for a global token, your terminal MUST have the Global Tokens Functionality Enabled.

> **IMPORTANT** The token exchange functionality is not a tokenization endpoint. If you have a regular token issued while the Global Token functionality was disabled for your terminal, then you will not receive a global token for it. In order to do so, you have to tokenize the card again.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)

```
{
    "terminal": {
```

```
    },
    "token": "XXXiuS1XiX9w44XX"
}
```


POST Sale with a Fee Terminal

https://gateway.kajadashboard.com/payment/sale

In some cases, there can be 2 terminals involved in one transaction. We call this functionality a Fee Terminal.

You can utilize this endpoint if your merchant account is setup to have a separate fee terminal. The transaction will be divided into two separate authorizations and captures.

You can utilize this endpoint if your merchant account is setup to have a separate fee terminal. The transaction will be divided into two separate authorizations and captures.

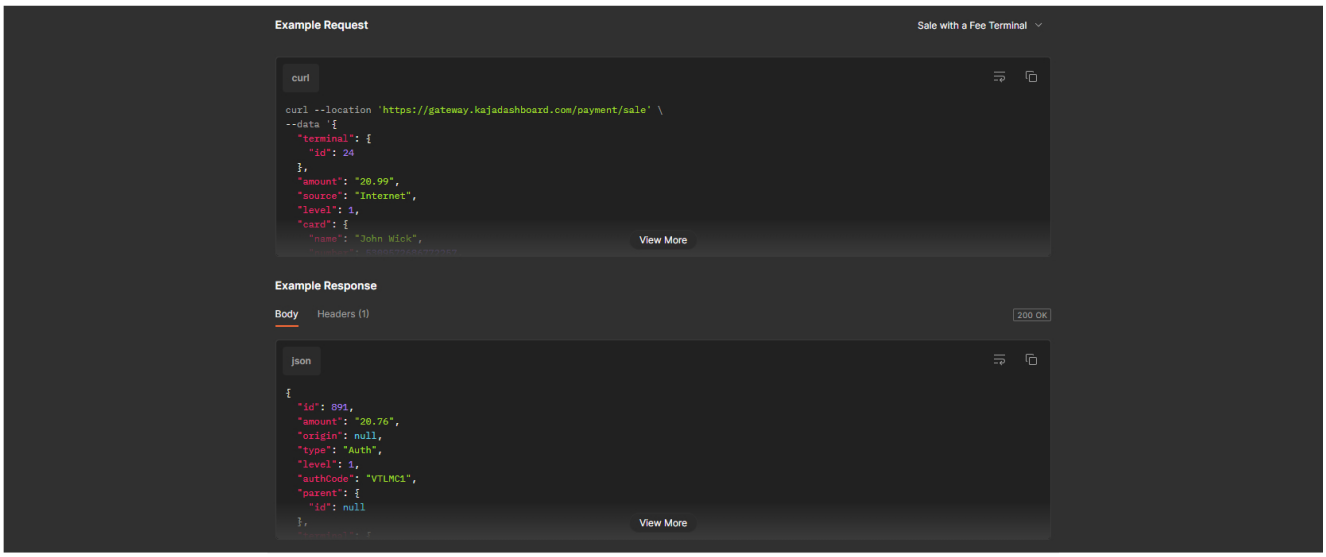
AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Card Payments

Body raw (json)


```
{
    "terminal": {
        "id": 24
    },
    "amount": "20.99",
    "source": "Internet",
    "level": 1,
    "card": {
        "name": "John Wick",
        "number": 6389572686772257,
```

View More



Hosted Form

With our Gateway embedded hosted form functionality, you can achieve quick, efficient, and secure online transactions, without the worry of PCI compliance.

The overall process to generate and prepare a hosted gateway payment form is as follows.

1. Create a POST request with the required parameters for your needs
2. We will return the code of the hosted form and the URL in the response

1. Create a POST request with the required parameters for your needs
2. We will return the code of the hosted form and the URL in the response

Hosted Form Object

Fields	Description	Type	Required
dba	Dba Object	Object	Yes
dba.id	DBA ID	Integer	Yes
terminal	Terminal Object	Object	Yes

terminal.id	Terminal ID	Integer	Yes
threeds	Activate 3D Secure `Required`, `Disabled` or `RequiredIfAvailable`	ENUM	No
amount	Payment Amount	Number	No
feeType	The type of sales tax applied `amount` or `percent`	ENUM	No
fee	The sales tax value applied to the order	Number	No
externalId	Unique record identifiers fro		No

View More

> The Amount and External ID are dynamic parameters that can still be provided after the form is generated.
> If you want to dynamically provide Amount and External ID you can do so at the end of the URL using &amount=&externalId= or you can provide them in the corresponding javascript parameters. Keep in mind that if the amount is not provided the client will be prompted to enter it.

The return URL also gives you the option to change the mapping of specific parameters (, , and) based on your needs.
The parameters that we have are as follows: Success, Failed, and Cancelled.

> **IMPORTANT** If you have set `sendReciept` to "Yes" `requestContactInfo` needs to be also "Yes" as well or no receipt will be generated.

Dynamic Contact and Billing Information

Option 1

We give you the ability to provide the contact and billing information in the request to generate the form. This will return you the form with those fields pre-filled. Example requests and responses are provided.

Option 2

We also give you the ability to dynamically set Contact and Billing information in your environment without the need to constantly generate new forms this is how:

> If you set `requestContactInfo` and/or `requestBillingInfo` to Yes, you can dynamically fill in the 'contactInfo' and/or 'billingInfo' objects which are present in the options of the response code of the generated form.

json


```
"data": "eyJkYmFJZCI6MTEyLCJ0ZXJtaW5hbElkIjoxMDksInRocmVlZHMiOiJEaXNhYmxlZCIsImV4dGVybmFsSWQiOiJleGFtcGxlIiwicmV0dXJuVXJsIjoiaHR
"amount": 2,
"fee": 2,
"feeType": "amount",
"contactInfo": {
    "name": "John Doe",
    "email": "john.doe@example.com",
    "phone": "+1 800 555 111"
},
"billingInfo": {
```


View More

> Or you can add the base64 encoded version of 'contactInfo' and/or 'billingInfo' JSON.stringified objects in the URL. Like in the following JS example:

javascript

```
let url = "https://kajadashboard.com/gateway/public/form?data=eyJkYmFJZCI6MTEyLCJ0ZXJtaW5hbElkIjoxMDksInRocmVlZHMiOiJEaXNhYmxlZCI
let contactInfoObject = {
    "name": "John Doe",
    "email": "john.doe@example.com",
    "phone": "+1 800 555 111"
};
let contactInfoHash = btoa(JSON.stringify(contactInfoObject));
let billingInfoObject = {
    "country": "United States",
    "address": "Example address 1",
```

View More

AUTHORIZATION Bearer Token

Token	XXX

POST Generate

https://kajadashboard.com/api/gateway/hosted-form

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Hosted Form

Body raw (json)

json

```
{
    "dba": {
        "id": 9
    },
    "terminal": {
        "id": 24
    },
    "threeds": "Disabled",
    "amount": 2,
    "feeType": "amount",
```

View More

Example Request Generate

curl

```
curl --location 'https://kajadashboard.com/api/gateway/hosted-form' \
--header 'Content-Type: application/json' \
--data '{
    "dba": {
        "id": 9
    },
    "terminal": {
        "id": 24
    },
    "threeds": "Disabled",
```

View More

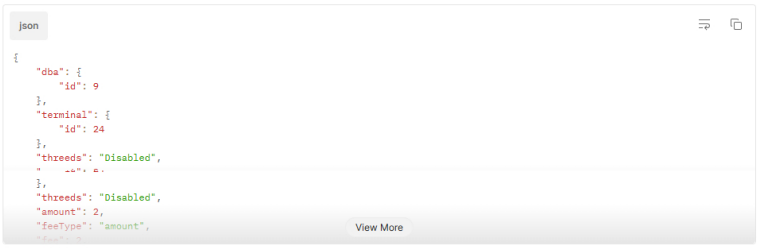
POST Generate with Billing and Contact Information

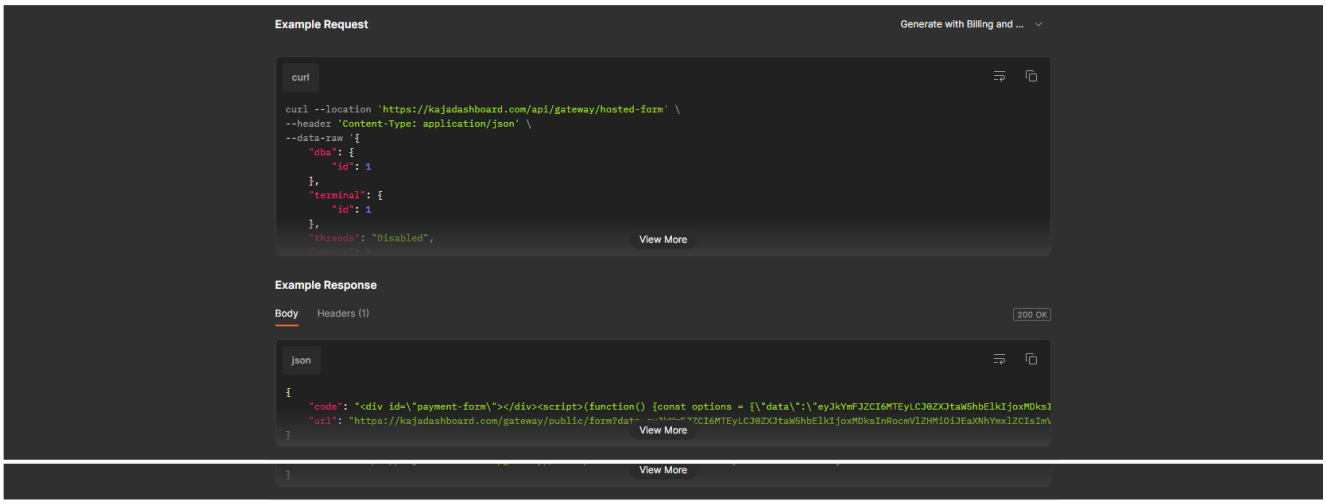
https://kajadashboard.com/api/gateway/hosted-form

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Hosted Form

Body raw (json)





Hosted Fields

Hosted Fields allows you to create a highly customized payment page for your website or application. All fields are completely customizable to fit your web page design and branding. In order to use this functionality you need to have an API token with permission `Credit/Debit Card Payments Read and Write` assigned to it.

AUTHORIZATION Bearer Token

Token	XXX

How it Works

Everytime when a payment should be made, you must include our `HostedFields.js` asset into your website, then issue an `Hosted Fields Token` with which a payment can be processed. You then initialize a payment form (with the `Hosted Fields Token`) by providing the required configuration and styling for it. We will render the payment form the way you requested it and process the payment when the customer submits the form with all required data (card number, cvv, etc.). In order to make it fully functional you will need to handle all payment form events that occur on your end. We're going to notify you of these events (validation failure, validation success, 3DS started, processing start and so on), but you will need to handle them accordingly (e.g., updating the UI when required) to make the payment experience user friendly. Hosted Fields Asset URL:

`https://kajadashboard.com/js/HostedFields.js`

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

This folder is using Bearer Token from folder Hosted Fields

Hosted Fields Token

A `Hosted Fields Token` is required to process a hosted fields payment without exposing your API Token.

> Only a single payment can be made with one Hosted Fields Token. This means that for every payment you need to get fresh Hosted Fields Token

After the payment is complete, you can use the `Hosted Fields Token` to get the tokenized card if card saving was allowed during the payment process.
To get an `Hosted Fields Token`, you need to make a call to this endpoint:

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

POST Hosted Fields Token

https://kajadashboard.com/api/hosted-fields/token

Supported Parameters:

Fields	Description	Type	Default	Required
expiration	Token expiration in minutes. The maximum	Integer	30	No

Fields	Description	Type	Default	Required
expiration	Token expiration in minutes. The maximum allowed value is 30	Integer	30	No
terminal	The ID of your terminal on which the payment will be processed. Must be an active terminal using our Gateway	Integer	N/A	Yes
domain	The Website domain on which the token will be used. Please use full URL Name. E.g. https://my.website.com	String	N/A	Yes
saveCard	Whether you want to retrieve the tokenized card after the payment is made. Allowed values: `required`, `optional`, `disabled`	String	disabled	No
3ds	Enable or Disable 3DS	boolea...	false	No

View More

`expiration` Parameter

Controls the expiration time of the token. Defaults to 15 minutes if not provided. The maximum allowed value is 30. If card saving (card tokenization) was allowed by the user during the payment process (or if you made it a requirement), you must retrieve the tokenized card while the `Hosted Fields Token` is still active.
For example if you issue an `Hosted Fields Token` with an expiration time of 10 minutes, then the payment and card token retrieval must occur within this time limit. Otherwise, you will need to issue a new `Hosted Fields Token`. **If the payment is made within the time limit but the card token is not retrieved within the same period, it will be lost and will not be available to you**

`terminal` Parameter

Specifies the terminal ID on which the payment will be processed. The terminal must be in `Active` status and must be an internal terminal, pointing to our gateway.

`domain` Parameter

Specifies the URL on which the `Hosted Fields Token` will be used and the payment form will be rendered. You need to specify the full URL. E.g: `https://my.website.com`. If your website is accessed with a port number, you need to provide it too. E.g: `https://my.website.com:3000`

`saveCard` Parameter

Specifies whether you want to save the tokenized card after the payment is made. Accepted values: `required`, `optional`, `disabled`.

`required`

The card will be tokenized and available for you to fetch after the payment is processed. A message will be rendered above the payment form submit button informing the user that their card will be saved. You are responsible for requesting the tokenized card after the payment.

`optional`

Card tokenization will be optional. The user will be presented with a checkbox to explicitly confirm that they want their card to be saved for future payments. If unchecked the card token will not be available for you.

`disabled`

This is the default value of the `saveCard` parameter. Card tokenization will not be performed and the card token will not be available to you.

`3ds` Parameter

Enables or disables the 3DS Verification when it is set as optional on the terminal. If the terminal you specified requires 3DS Verification, then the 3DS will be enabled by default and you wont be able to disable it in your hosted fields form. If 3DS verification is disabled on your terminal, then the 3DS will also be disabled. If you still request 3DS verification for the `Hosted Fields Token` when 3DS is disabled, an exception will be thrown.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Hosted Fields

Body raw (json)

json

```
{
    "expiration": 15,
    "terminal": 3,
    "domain": "https://my.website.com",
    "saveCard": "required",
    "3ds": false
}
```

Example Request Hosted Fields Token

curl

```
curl --location 'https://kajadashboard.com/api/hosted-fields/token' \
--header 'Content-Type: application/json' \
--data '{
    "expiration": 15,
    "terminal": 3,
    "domain": "https://my.website.com",
    "saveCard": "required",
    "3ds": false
}'
```

Example Response

Body Headers (1)

json

```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvZGFzaGJvYXJkLmludCIsInN1YiI6IkhvcSRlZCBGaWVsZHMgQWNjZ
```

```
  "issued_at": 1741701742,
  "expiration": 15,
  "expires_at": "2025-03-11 14:17:22 UTC"
}
```


View More

Integration Guide

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

Step 1: Include the JavaScript Library

After you have your AccessToken you can start integrating the hosted fields form into your website. First, add the `HostedFields` asset script tag to access the JavaScript Library:

```javascript
<script src="https://kajadashboard.com/js/HostedFields.js"></script>
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

Step 2: Add HTML Elements for all payment fields

Step 2: Add HTML Elements for all payment fields

You need some HTML elements that will act as containers in which the separate hosted fields will be rendered. Here's an example structure:

```html
<div id="card-number"></div>
<div id="card-expiration"></div>
<div id="card-cvv"></div>
<div id="card-holder-name"></div>
<div id="submit-button"></div>
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

Step 3: Set the AccessToken and Configure the rest of the form

Main Configuration

Now you are ready to render the hosted fields form on your website. Start by setting the `Hosted Fields Token`, which you've obtained from your backend together with the payment amount. After that, you must specify all hosted fields containers together with the payment amount.

```javascript
<script>
  const form = HostedFields.create({
```

```javascript
<script>
  const form = HostedFields.create({
    token: AccessToken,
    amount: 25, // Provide the payment amount
    fields: {
      "cardNumber": {
        target: "#card-number" // This is the ID of the div element that we created above
      },
      "cardExpiration": {
        target: "#card-expiration"
```

View More

> Keep in mind that the `fields` object property names must be exactly as shown above: `cardNumber, cardExpiration, cardCvv, cardHolderName, submit`

Setting Shipping data

StartFragment

In case your `Hosted Fields Token` is issued with **3DS Param** set to `true` and your customer provides an AMEX (American Express) Card you will be required to provide shipping data. You will also be informed for this via an event called **hostedFields.shippingRequired** which will occur when the customer enters his card. You can provide the shipping data in two ways:

1. During initial form creation via the `shipping` property of the configuration object, see the example above.
2. Using the `setShippingInfo` method on the form instance:

```json
form.setShippingInfo({
  address1: 'Example Address Line #1',
  country: 'US',
  city: 'Chicago',
  state: 'Illinois',
  zip: '60639'
})
```

You can use this method together with the **hostedFields.shippingRequired** event. When receiving this event you might want to explicitly check if the `shippingRequired` flag is set to `true` on the event object. In case the user provides an AmEx card you will receive the event with flag equal to `true`, but if he changes the card to a non-AmEx card you will receive the event with flag equal to false.

```json
form.addEventListener('hostedFields.shippingRequired', e => {
  if(e.detai.shippingRequred) {
    form.setShippingInfo({
      address1: 'Hello world',
      country: 'US',
      city: 'Hello city',
      state: 'Illinois',
      zip: '40005'
    })
  }
})
```

The shipping object has the following structure:

Field	Data	Type	Required
shipping.address1	Shipping Address	string	Yes
shipping.country	Shipping Country. You can use full country name or ISO2/ISO3 Code	string	Yes
shipping.city	City	string	Yes
shipping.state	State. Required when country is US	string	Mixed
shipping.zip	ZIP	string	Yes

> For the 3DS Integration the `shipping` data is synonymous to the billing data. In case you collect and represent it as `billing` information in your app, it's still the same for your hosted fields integration. Just provide that billing data as `shipping` using the `setShippingInfo` or during the initialization with the correct object structure

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

Step 4: Add styling

Manually Providing CSS

The object you pass to the `create` function of the `HostedFields` library supports a second property called `styles`. There, you can provide all the CSS styles needed to render the forms in your design. E.g:

```html
<script>
  const form = HostedFields.create({
    token: 'YOUR-ACCESS-TOKEN',
    fields: {
    },
    styles: {
      "input[type=text]": {
        'width': '100%',
        'padding': '10px',
        'border-radius': '7px',
```

View More

At this point your form should be rendered and visible on your webpage.

At this point your form should be rendered and visible on your webpage.

> Whenever an input is considered invalid we will add `invalid` className to the class attribute of the input. If the input is considered valid we will add `valid` className to the class attribute.

Automatically Detecting your CSS

If you don't want to provide the CSS Manually, we can detect it automatically for you. In order for this to work you need to provide us the input's HTML with already applied styles/classes and use `useTargetStyle` flag inside the field config. E.g:

```html
<div id="card-number">
  <input type="text" class="form-control my-input-class" id="card-number-input">
</div>
<div id="card-expiration"></div>
<div id="card-cvv"></div>
<div id="card-holder-name"></div>
<div id="submit-button"></div>
<script>
  HostedFields.setToken(AccessToken);
  const form = HostedFields.create({
```

View More

In the above example we create an input with our own styles (provided by the classes `form-control` and `my-input-clss`) and an id `card-number-input`, then we inform the config that this element (`#card-number-input`) will be the target element for the iframe providing the card number input and that it needs to use its styles to render the input. And you are ready to go, the iframe will copy your styles and apply them for you.

> Keep in mind that your original `input` element will be removed and the iframe will be inserted inside its parent. In the example above the parent is `div#card-number`.

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

Step 5: Listen for Events

The `form` variable returned by our JavaScript library is an instance of **EventTarget** object. This means that you can **addEventListener** to all of the various events it emits while the user interacts with it or while the payment is processed. All of the emitted events are **CustomEvents** and they might have payload containing event related data under `event.detail` Object. Here's a full list:

Event Name	Description	Detail
hostedFields.ready	The form is initialized and rendered	N/A
hostedFields.error	Configuration or Initialization Error. E.g. `Invalid Hosted Fields Token`, `Invalid Domain`, `Server Error` etc.	{message: string}
hostedFields.validation	Validation result for a given field. Occurs while user is typing and after a payment form is submitted.	{valid: bool, field: string, message: string}
hostedFields.shippingRequired	Whether or not the shipping data is required. Occurs when provided card is AmEx and 3DS is enabled	{shippingRequired: bool}
card.type	Brand of the credit/debit card provided used by the user	string
submit.processing	Payment processing has started. The occurs when user submits the payment form	N/A

View More

Example event handlers:

```javascript
```

```javascript
const formMessage = document.getElementById('#formMessage'); // Assume this is you notification message

// We initialized the form variable in Step 4.
form.addEventListener('3ds.start', () => formMessage.innerHtml = '3DS Verification Started');

// Form submitted event
form.addEventListener('submit.processing', () => formMessage.innerHtml = 'Your payment is being processed');

// Payment Result Event
form.addEventListener('submit.result', e => {
```

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

Step 6: Get the tokenized card

When a payment is completed and you did set **saveCard** parameter to `required` or `optional` (and the user gave the permission) the card token will be available for you to fetch. To do so you need make a POST request to following endpoint:

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

POST **Get the tokenized card**

`https://kajadashboard.com/api/hosted-fields/card-token`

> You need to perform this call before your `Hosted Fields Token` expires.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Hosted Fields

Body raw (json)

```json
{
    "accessToken": "ACCESS-TOKEN-USED-FOR-PAYMENT"
}
```

Example Request Get the tokenized card

```curl
curl --location 'https://kajadashboard.com/api/hosted-fields/card-token' \
--header 'Content-Type: application/json' \
--data '{
    "accessToken": "ACCESS-TOKEN-USED-FOR-PAYMENT"
}'
```

Example Response

Body Headers (1)

```json
{
    "token": "TOKENIZED-CARD-STRING",
    "bin": 412493, // First 6 numbers of the card
    "exp": "2030-12-31"
}
```

Add Payment Method Page

If you want to use the hosted fields functionality to create an `Add Payment Method` page to your website the process is the same as described above. Just make sure you set the `amount` parameter to **0**. And when issuing the `Hosted Fields Token` pass the **saveCard** parameter as `required`. We will make sure to render the correct submit button text so the users know they aren't paying anyting.

```javascript
<script>
  const form = HostedFields.create({
    token: 'YOUR-ACCESS-TOKEN',
    amount: 0, // Provide 0 as payment amount
    fields: {...},
    token: 'YOUR-ACCESS-TOKEN',
    amount: 0, // Provide 0 as payment amount
    fields: {...},
    styles: {...}
  });
  </script>
```

After the user submits the form the **tokenized card will be available**

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

Testing your integration

You can use our sandbox environment to test your integration.

`https://sandbox.kajadashboard.com`

Create your API Token there with permission `api-creditcard-payment-read-write` and use it to issue `Hosted Fields Token` and render a hosted fields form on your website. You can use our **test cards** to make payments and fetch tokenized cards.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Hosted Fields

CustomerVault

Customer Vault securely stores sensitive customer billing information, allowing subsequent transactions to be processed without the need for you to recollect the information or store it on your own system. This makes for a quick and simple billing process.

Customer Vault securely stores sensitive customer billing information, allowing subsequent transactions to be processed without the need for you to recollect the information or store it on your own system. This makes for a quick and simple billing process.

The Customer Vault API offers the following capabilities:

- Create and modify customer profiles.
- Securely store customer payment information, such as card numbers, in the Customer Cards, and link it to the customer profile.
- Store multiple payment cards for each customer.
- Use tokens to process transactions.

You can **create**, **update**, and **delete** customer information from the customer vault.

Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the customer	Integer	Yes	No	Yes
dba	DBA Object	Object	No	Yes	No
dba.id	ID	Integer	Yes	Yes	No
firstName	First name	String	Yes	No	No
lastName	Last name	String	Yes	No	No
company	Company	String	Yes	No	No
email	Email	String	Yes	Yes	No
website	Website	String	No	No	No
phone	Phone	String	Yes	Yes	No
altPhone	Alt phone	String	No	No	No
identificator	Internal customer ID on the merchants	String	Yes	Yes	No

View More

AUTHORIZATION Bearer Token

Token	XXX

Sandbox

IMPORTANT Please be aware the CustomerVault URL is different than what is referenced above in the merchant section of this documentation. Please ensure to use the correct URL endpoint for CustomerVault as shown below.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder CustomerVault

GET Sandbox Dashboard

https://sandbox.kajadashboard.com<path>

For testing purposes, please use our sandbox endpoint.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

For testing purposes, please use our sandbox endpoint.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request Sandbox Dashboard

curl

```
curl --location 'https://sandbox.kajadashboard.com<path>'
```

Example Response

Body Headers (0)

No response body

This request doesn't return any response body

Hosted Card Form

With our Customer Vault embedded hosted form functionality, you can achieve quick, efficient, and secure customer card tokenization without the worry of PCI compliance.

With our Customer Vault embedded hosted form functionality, you can achieve quick, efficient, and secure customer card tokenization without the worry of PCI compliance.

The overall process to generate and prepare a hosted customer vault card form is as follows.

1. Create a POST request with the required parameters for your needs
2. We will return the code of the hosted form and the URL in the response
3. You can embed this form to collect customer card data in your vault securely.

Hosted Form Object

Fields	Description	Type	Required
dba	Dba Object	Object	Yes
dba.id	DBA ID	Integer	Yes
terminal	Terminal Object	Object	Yes
terminal.id	Terminal ID	Integer	Yes
customerVault	Customer Vault Object	Object	No
customerVault.id	The id of the Customer in the Customer Vault	Integer	No
returnUrl	The return URL after payment completion (i.e success/error page)	String	No
returnUrlNavigation	How to redirect the URL (Inside/Outside of frame) `top` or `self`	ENUM	No

> **IMPORTANT** If you do not supply `customerVault.id` , the generated code that we supply will include two additional parameters `customerVaultId` and `customerVaultToken` which you will need to dynamically provide when you are embeding the code. This information is needed to identify the correct customer in your Customer Vault.

In the return URL we will pass the Customer ID and Card ID as well as the status. The parameters that we have are: **Success** and **Cancelled**

AUTHORIZATION Bearer Token

Token XXX

POST Generate

https://kajadashboard.com/api/customer-vault-card/form

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Hosted Card Form

Body raw (json)

json

```
{
  "dba": {
    "id": 3
  },
  "terminal": {
    "id": 2
  },
  "customerVault": {
    "id": 1
  },
  "customerVault": {
    "id": 1
  },
```

View More

Example Request Generate

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault-card/form' \
--header 'Content-Type: application/json' \
--data '{
  "dba": {
    "id": 3
  },
  "terminal": {
    "id": 2
  },
  "customerVault": {
```

View More

Example Response

Body Headers (1) 200 OK

json

```
{
  "code": "<div id=\"customer-vault-add-card\"></div><script>(function() {const options = {\"data\": \"eyJkYmFJZCI6MywidGVybWluYWxJZCI
  "url": "https://kajadashboard.com/customerVault/public/form
}
```

View More

View More

Customers

Use this endpoint to manage all of your customers in the Customer Vault.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder CustomerVault

GET List

https://kajadashboard.com/api/customer-vault

List all of your customers

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request List

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault'
```


```
curl --location 'https://kajadashboard.com/api/customer-vault'
```


Example Response

Body Headers (1) 200 OK

json


```
{
  "items": [
    {
      "id": "1",
      "dba": {
        "id": "1"
      },
      "firstName": "John",
      "lastName": "Wick",
      "company": null,
```


View More

GET View

https://kajadashboard.com/api/customer-vault/<customerId>

Use this endpoint to view information for a specific customer.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request View

Example Request View

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault/1'
```

Example Response

Body Headers (1) 200 OK

json


```
{
    {
    "id": "1",
    "dba": {
        "id": "1"
    },
    "firstName": "John",
    "lastName": "Wick",
    "company": null,
    "email": "johnwick@email.com",
```


View More

POST Create

https://kajadashboard.com/api/customer-vault

Creating a Customer can be achieved using this endpoint

Creating a Customer can be achieved using this endpoint

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

json

```
{
    "dba": {
        "id": "1"
    },
    "firstName": "John",
    "lastName": "Wick",
    "email": "johnwick@email.com",
    "phone": "+18187740010",
    "description": "Description to help me identify the customer"
}
```

Example Request Create

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault' \
--data-raw '{
    "dba": {
        "id": "1"
    },
    "firstName": "John",
    "lastName": "Wick",
    "email": "johnwick@email.com",
    "phone": "+18187740010",
    "description": "Description to help me identify the customer"
}'
```

Example Response

Body Headers (1) 200 OK

json

```
{
  "id": 1,
  "dba": {
    "id": 1
  },
  "firstName": "John",
  "lastName": "Wick",
  "company": null,
  "email": "johnwick@email.com",
  "website": null,
```


View More

PUT Update

https://kajadashboard.com/api/customer-vault/<customerId>

The update endpoint is used to modify the customer's information in the Customer Vault. You can modify personal details and company information of the customer in the vault.

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

json


```
{
    "phone": "+18187267744"
}
```


Example Request Update

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/customer-vault/1' \
--data '{
    "phone": "+18187267744"
}'
```

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": 1,
  "dba": {
    "id": 1
  },
  "firstName": "John",
  "lastName": "Wick",
  "company": null,
  "email": "johnwick@email.com",
  "website": null,
```


View More

DELETE Archive

https://kajadashboard.com/api/customer-vault/<customerId>/archive

When a specific customer in the vault has transactional history, that customer cannot be deleted. In this case, the customer needs to be archived.

> **NOTE** We allow a customer to be returned to active status from archived.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request Archive

curl

```
curl --location --request DELETE 'https://kajadashboard.com/api/customer-vault/1/archive' \
--data ''
```

Example Response

Example Response

Body Headers (1) 200 OK

json


```
{
    true
}
```


DELETE Delete

https://kajadashboard.com/api/customer-vault/<customerId>

If a specific customer in the vault has no transactional history your can delete the customer using this endpoint.

> **IMPORTANT** You cannot restore a deleted customer. If you are unsure of future use we suggest archiving the customer.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request Delete

curl

```
curl --location --request DELETE 'https://kajadashboard.com/api/customer-vault/1/' \
--data ''
```

Example Response

Body Headers (1) 200 OK

```
{
    true
}
```

Cards

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder CustomerVault

GET List Cards

https://kajadashboard.com/api/customer-vault/<customerId>/cards

List all stored payment information for a customer. This will return an object of all the stored payment card records on file for a specific customer in the vault.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

This request is using Bearer Token from folder CustomerVault

Example Request List Cards

```
curl --location 'https://kajadashboard.com/api/customer-vault/1/cards'
```

Example Response

Body Headers (1) 200 OK

```
{
  "items": [
    {
      "id": "2",
      "billing": {
        "id": "1"
      },
      "number": 4489,
      "bin": {
        "bin": 411111,
```

View More

GET View

https://kajadashboard.com/api/customer-vault/<customerId>/card/<id>

https://kajadashboard.com/api/customer-vault/<customerId>/card/<id>

To view a single card record for a specific customer use the following:

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request View a specific payment c...

```
curl --location 'https://kajadashboard.com/api/customer-vault/1/card/1'
```

Example Response

Body Headers (1) 200 OK

```
{
  "id": "2",
  "billing": {
    "id": "1"
  },
  "number": 4489,
  "bin": {
    "bin": 411111,
    "brand": "Visa",
    "organization": "Jpmorgan Chase Bank, N.a.",
```

View More

```
    "bin": 411111,
    "brand": "Visa",
    "organization": "Jpmorgan Chase Bank, N.a.",
```

View More

POST Add Card

https://kajadashboard.com/api/customer-vault/<customerId>/card

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

```
json
{
    "billing": {
        "id": 34
    },
    "terminal": {
        "id": 1593
    },
    "cvv": "996",
    "exp": "12/31",
    "holderName": "John Wick",
```

View More

Example Request Add Card

```
curl
curl --location 'https://kajadashboard.com/api/customer-vault/1/card' \
--data '{
    "billing": {
        "id": 2
    },
    "terminal": {
        "id": 1
    },
    "cvv": "996",
    "exp": "12/31",
```

View More

Example Response

Body Headers (1) 200 OK

```
json
{
  "id": 1,
  "number": 6466,
  "bin": {
    "bin": 411111,
    "brand": "Visa",
    "organization": "Jpmorgan Chase Bank, N.a.",
    "type": "Credit",
    "category": null,
    "country": {
```

View More

PUT Update Exp Date

PUT Update Exp Date

https://kajadashboard.com/api/customer-vault/<customerId>/card/<cardId>/change-exp-date

You can utilize this endpoint to update/change the expiration date of a stored and tokenized card in the Customer vault

> **IMPORTANT** Exp.Date Format [MM/YY]. 12/30

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

```
json
{
    "exp": "12/30"
}
```

Example Request Update Exp Date

```
curl
curl --location --request PUT 'https://kajadashboard.com/api/customer-vault/1/card/12/change-exp-date' \
--data '{
    "exp": "12/30"
}'
```

Example Response

Body Headers (1) 200 OK

```
json
{
    true
}
```

PUT Update Status

https://kajadashboard.com/api/customer-vault/<customerId>/card/<cardId>/change-status

By utilizing this endpoint Card Status can be changed to `Active` or `Inactive`.

> You cannot change status of an expired card.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

```
json
{
    "status": "Active"
```

```
    "status": "Active"
}
```

Example Request — Update Status

```
curl --location --request PUT 'https://kajadashboard.com/api/customer-vault/1/card/12/change-status' \
--data '{
    "status": "Active"
}'
```

Example Response

Body | Headers (1) — 200 OK

```
{
    true
}
```

DELETE Archive

https://kajadashboard.com/api/customer-vault/<customerId>/card/<id>/archive

If a specific payment card record is present in the transactional history of a customer we do not allow for it to be deleted. In this case, you can archive the payment

If a specific payment card record is present in the transactional history of a customer we do not allow for it to be deleted. In this case, you can archive the payment card information record.

> **NOTE** We allow a customer's payment card record to be returned to active status from archived - use our **Unarchive** endpoint.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request — Archive a payment card

```
curl --location 'https://kajadashboard.com/api/customer-vault/1/card/1/archive' \
--data '{
    "billing": {
        "id": 2
    },
    "terminal": {
        "id": 1
    },
    "cvv": "996",
    "exp": "12/31",
```

View More

Example Response

Body | Headers (1) — 200 OK

```
{
    true
}
```

PUT Unarchive

https://kajadashboard.com/api/customer-vault/<customerId>/card/<id>/unarchive

If you want to return an archived card to an active status use the following:

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request — Unarchive a payment card

```
curl --location 'https://kajadashboard.com/api/customer-vault/1/card/1/unarchive' \
--data '{
    "billing": {
        "id": 2
    },
    "terminal": {
        "id": 1
    },
    "cvv": "996",
    "exp": "12/31",
```

View More

Example Response

Body | Headers (1) — 200 OK

```
{
    true
}
```

DELETE Delete

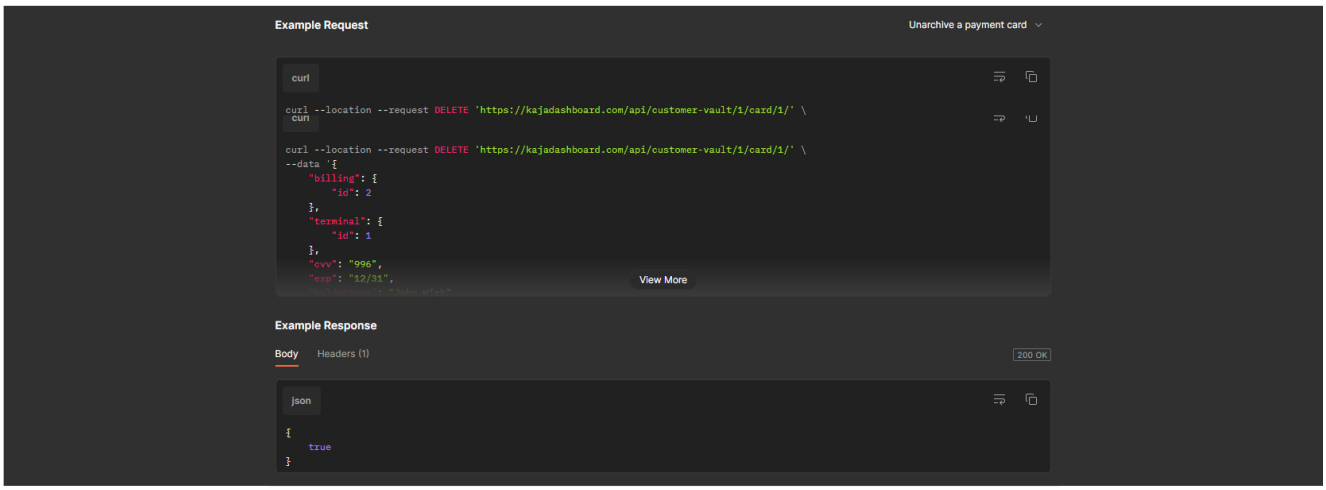
https://kajadashboard.com/api/customer-vault/<customerId>/card/<id>

If a specific payment card is not present in the transactional history of a customer you can delete it.

IMPORTANT You cannot restore a deleted payment card record.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault



Billing Information

You can use this endpoint for customer billing-related operations in the customer vault. You can create, update, delete, or archive customer billing information.

Billing Information Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID	Integer	Yes	No	Yes
id	Unique ID	Integer	Yes	No	Yes
firstName	First name	String	Yes	Yes	No
lastName	Last name	String	Yes	Yes	No
address	Address	String	Yes	Yes	No
state	State `Example: CA, NY, etc...`	String	Yes	No	No
country	Country `Example: US, UK, etc...`	String	Yes	Yes	No
city	City	String	Yes	Yes	No
zip	ZIP code	String	Yes	Yes	No
archived	`Yes` or `No`	ENUM	Yes	No	Yes
createdOn	Datetime of creation	Datetime		No	Yes

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder CustomerVault

GET List

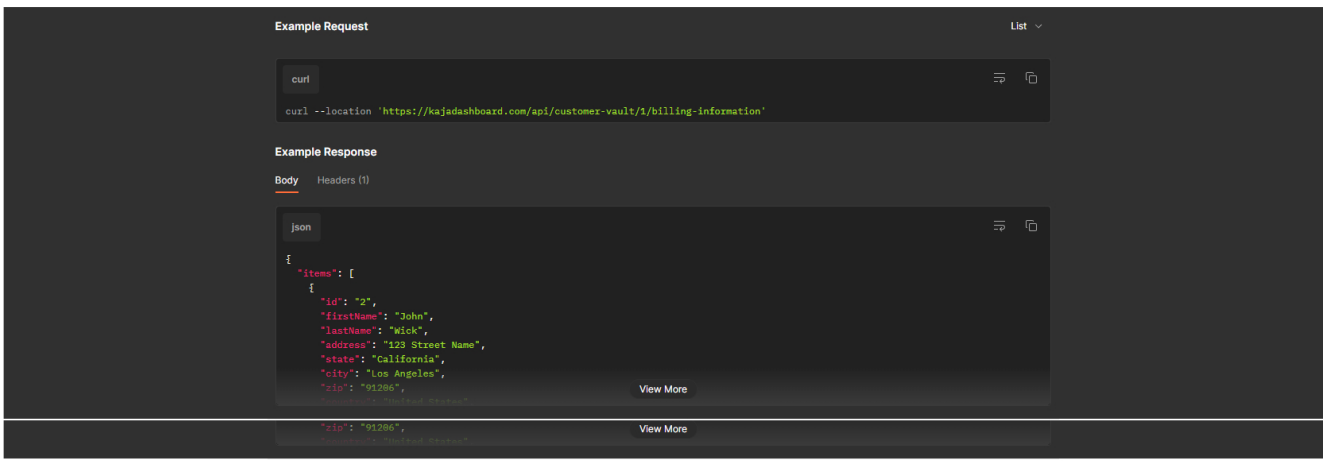
https://kajadashboard.com/api/customer-vault/<customerId>/billing-information

https://kajadashboard.com/api/customer-vault/<customerId>/billing-information

This will return an object of all the billing information records on file for a specific customer in the vault.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault



GET View

https://kajadashboard.com/api/customer-vault/<customerId>/billing-information/<id>

Use this endpoint to view a single billing record for a specific customer

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request View

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault/<customerId>/billing-information/<id>'
```

Example Response

Body Headers (1)

json

```
{
  "id": 136,
  "firstName": "Marty",
  "lastName": "McFly",
  "address": "9303 Lyon Drive",
  "state": "California",
  "city": "Hill Valley",
  "zip": "95420",
  "country": "United States",
  "archived": "No",
```

View More

PUT Update

https://kajadashboard.com/api/customer-vault/<customerId>/billing-information/<id>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

json

```
{
    "firstName": "Emmett",
    "lastName": "Brown",
    "state": "CA"
}
```

Example Request Update

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/customer-vault/38/billing-information/34' \
--data '{
    "firstName": "Emmett",
    "lastName": "Brown",
    "state": "CA"
}'
```

Example Response

Body Headers (1)

json

```
{
  "id": 136,
  "firstName": "Emmett",
  "lastName": "Brown",
  "address": "9303 Lyon Drive",
  "state": "California",
  "city": "Hill Valley",
  "zip": "95420",
  "country": "United States",
  "archived": "No",
```

View More

POST Create

https://kajadashboard.com/api/customer-vault/<customerId>/billing-information

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

json

```
{
    "firstName":"John",
    "lastName":"Wick",
    "address":"123 Street Name",
    "city":"Los Angeles",
    "zip":"91206",
    "state":"CA",
    "country":"US"
}
```

Example Request Create

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault/38/billing-information' \
--header 'Content-Type: application/json' \
--data '{
   "firstName":"Marty",
   "lastName":"McFly",
   "address":"123 Street Name",
   "city":"Los Angeles",
   "zip":"91286",
   "state":"CA",
   "country":"US"
}'
```

Example Response

Body Headers (1)

json


```
{
  "id": "365",
  "firstName": "Marty",
  "lastName": "McFly",
  "address": "123 Street Name",
  "state": "California",
  "city": "Los Angeles",
  "zip": "91286",
  "country": "United States",
  "archived": null,
```


View More

Shipping Information

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder CustomerVault

GET List

https://kajadashboard.com/api/customer-vault/<customerID> /shipping-information

GET List

https://kajadashboard.com/api/customer-vault/<customerID> /shipping-information

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request List

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault/<customerID>/shipping-information'
```

Example Response

Body Headers (1)

json


```
{
    "items": [
        {
            "id": 28,
            "firstName": "John",
            "lastName": "Wick",
            "address": "123 Sesame St",
            "state": "California",
            "city": "Los Angeles",
            "zip": "91286",
```


View More

GET View

https://kajadashboard.com/api/customer-vault/<customerID>/shipping-information/<id>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request View

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault/<customerID>/shipping-information/<id>'
```

Example Response

Body Headers (1)

json


```
{
  "id": 28,
  "firstName": "John",
  "lastName": "Wick",
  "address": "123 Sesame St",
  "state": "California",
  "city": "Los Angeles",
  "zip": "91286",
  "firstName": "John",
  "lastName": "Wick",
  "address": "123 Sesame St",
  "state": "California",
  "city": "Los Angeles",
  "zip": "91286",
  "country": "United States",
  "note": "Example Note",
```


View More

PUT Update

https://kajadashboard.com/api/customer-vault/<customerID>/shipping-information/<id>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

```json
{
    "firstName": "John",
    "lastName": "Wick",
    "state": "CA"
}
```

Example Request Update

```curl
curl --location --request PUT 'https://kajadashboard.com/api/customer-vault/<customerID>/shipping-information/<id>' \
--data '{
   "address":"111 New Street Name",
   "city": "San Diego"
}'
```

Example Response

Body Headers (1)

```json
{
  "id": 28,
  "firstName": "John",
  "lastName": "Wick",
  "address": "111 New Street Name",
  "state": "California",
  "city": "San Diego",
  "zip": "91206",
  "country": "United States",
  "note": "Example Note",
```

View More

POST Create

https://kajadashboard.com/api/customer-vault/<customerID>/shipping-information

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

```json
{
    "firstName":"John",
    "lastName":"Wick",
    "address":"123 Street Name",
    "city":"Los Angeles",
    "zip":"91206",
    "state":"CA",
    "country":"US"
}
```

Example Request Create

```curl
curl --location 'https://kajadashboard.com/api/customer-vault/<customerID>/shipping-information' \
--data '{
    "firstName":"John",
    "lastName":"Wick",
    "address":"123 Street Name",
    "city":"Los Angeles",
    "zip":"91206",
    "state":"CA",
    "country":"US"
}'
```

Example Response

Body Headers (1)

```json
{
  "id": "30",
  "firstName": "John",
  "lastName": "Wick",
  "address": "1234 Street Name",
  "state": "California",
  "city": "Paris",
  "zip": "91206",
  "country": "United States",
  "note": null,
```

View More

Recurring Payment

Recurring Payments

Easily and securely set up billing that occurs at specific intervals. If you have a customer that decides to opt for recurring billing, the system can set up an order that recurs at the decided period and automatically uses the customer's payment details to process subsequent payments

The recurring transaction can be set with its own start and end date and time, billing frequency (complete freedom with cron expressions) and total number of payments. When recurring is configured for a customer in the system, a new transaction is sent on the set date for every billing cycle for the requested duration, without any customer or merchant intervention.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder CustomerVault

GET List

https://kajadashboard.com/api/customer-vault/38/recurring-payments

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request List

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault/38/recurring-payments'
```

Example Response

Body Headers (0)

Text


```
{
  "items": [
    {
      "id": 1,
      "name": "Monthly subscription",
      "description": "Cinema Membership",
      "amount": "3.11",
      "execute": {
        "frequency": 1,
        "period": "month"
```


View More

GET View

https://kajadashboard.com/api/customer-vault/38/recurring-payment/1

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Example Request View

curl

```
curl --location 'https://kajadashboard.com/api/customer-vault/38/recurring-payment/1'
```

Example Response

Body Headers (0)

Text

Text


```
{
  "id": 1,
  "name": "Monthly subscription",
  "description": "Cinema Membership",
  "amount": "3.11",
  "execute": {
    "frequency": 1,
    "period": "month"
  },
  "valid": {
```


View More

PUT Update

https://kajadashboard.com/api/customer-vault/38/recurring-payment/1

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)

json

```
{

    "name": "Monthly subscription",
    "description": "Cinema Membership",
    "amount": "3.11"
}
    "description": "Cinema Membership",
    "amount": "3.11"
}
```

Example Request Update

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/customer-vault/38/recurring-payment/1' \
--data '{
```

```
    "name": "Monthly subscription",
    "description": "Cinema Membership",
    "amount": "3.11"
}'
```


Example Response

Body Headers (0)


```
Text

{
  "id": 1,
  "name": "Monthly subscription",
  "description": "Cinema Membership",
  "amount": "3.11",
  "execute": {
    "frequency": 1,
    "period": "month"
  },
  "valid": {
                                                    View More
  },
  "valid": {
                                                    View More
```


POST Create

https://kajadashboard.com/api/customer-vault/38/recurring-payment

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder CustomerVault

Body raw (json)


```
json

{
    "name": "Monthly subscription",
    "description": "Cinema Membership",
    "amount": 3,
    "execute": {
        "frequency": 1,
        "period": "month"
    },
    "valid": {
        "from": "2020-01-01",
                                                    View More
```


Example Request Create


```
curl

curl --location 'https://kajadashboard.com/api/customer-vault/38/recurring-payment' \
--data '{
    "name": "Monthly subscription",
    "description": "Cinema Membership",
    "amount": 3,
    "execute": {
        "frequency": 1,
        "period": "month"
    },
    "valid": {
                                                    View More
```


Example Response

Body Headers (1)


```
json

{
  "id": "1",
  "name": "Monthly subscription",
  "description": "Cinema Membership",
  "amount": "3.00",
  "execute": {
    "frequency": 1,
    "period": "month"
  },
  "valid": {
                                                    View More
```


Transaction Reporting

Transaction Reporting

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET List

https://gateway.kajadashboard.com/payments

Transaction Reporting Object

Fields	Description	Type	Filterable	Required	Readonly
externalId	Unique record identifiers from a system outside of Payment Gateway	String	Yes	No	No
terminal	Terminal Object	Object	No	Yes	Yes
terminal.id	Dashboard Terminal ID	Integer	Yes	Yes	Yes
parent	Parent Object	Object	No	No	Yes
parent.id	Parent ID of a Payment Request	Integer	Yes	No	Yes
type	Request type	String	Yes	Yes	Yes
card	Card Object	Object	No	No	Yes

card.name	Cardholder Name	String	Yes	No	No
card.number	Card Number	String	Yes	No	No
amount	Amount of the capture	Number	Yes	Yes	No

View More

Use this endpoint to list all transactions for all of your `terminal.id`.

> Here as an exception you do not need to include the `terminal.id` in your request. Using this endpoint we will return all your transactions with their corresponding `terminal.id`.

Filter Example:

`https://gateway.maverickpayments.com/payment/?filter[terminal.id][eq]=1`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request List

curl

```
curl --location 'https://gateway.kajadashboard.com/payments'
```

Example Response

Body Headers (1) 200 OK

json


```
{
    "items": [
        {
            "id": 863,
            "amount": "2.00",
            "origin": "HostedForm",
            "type": "Auth",
            "level": 1,
            "authCode": "TAS364",
            "parent": {
```


View More

GET View

https://gateway.kajadashboard.com/payment/<id>

Use this endpoint to view information for a specific transaction record.

> You need to provide the `transaction.id` in your request.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request View

curl

```
curl --location 'https://gateway.kajadashboard.com/payment/863'
```

Example Response

Body Headers (1) 200 OK

Body Headers (1) 200 OK

json


```
{
  "id": 863,
  "amount": "2.00",
  "origin": "HostedForm",
  "type": "Auth",
  "level": 1,
  "authCode": "TAS364",
  "parent": {
    "id": null
  },
```


View More

Response Codes

Responses to any request clearly indicate if the request is processed successfully. We have integrated the following logic and groups in our response codes.

Status Object

Fields	Description	Type	Filterable	Required	Readonly	Level 1	Level 2	Level 3
status	Status Object	Object	No	No	Yes			
status.status	`Approved`, `Declined`, `Error`	String	Yes	No	Yes	No	No	No
	`Error`							
status.reason	Status reason	String	No	No	Yes	No	No	No

Approval Response

This is the only response that we will return only a status for it. Do not expect a status.reason here

Example:

json


```
"status": {
        "status": "Approved",
```

```
    "reason": null
}
```

Decline Response

The decline response will include a status.reason giving you more details on the type of decline.

Example:

```json
"status": {
        "status": "Declined",
        "reason": Insufficient Funds
    }
```

Here is a list of all possible decline codes:

Status Reason Message	Details
Duplicate Request (Approved previously)	The transaction was already performed and approved. Verify if the request was submitted twice for the same transaction ID or external reference number.
Reversal Not Allowed	The transaction is not authorized for reversal. This error may occur because the transaction was settled, declined, or already reversed.
Return Not Allowed	The transaction is not authorized for return. This error may occur because the transaction was not settled, was declined, or was already reversed.
Supervisor Override Required	
Modify Transaction Not Allowed	The transaction is not authorized for modification. This error may occur because the transaction was already settled, or was declined.
Possible Duplicate Request	This is a duplicate request. The credentials for this transaction (i.e. amount, card number, or same service) are the same as another transaction submitted less than one minute apart.
Duplicate Request (Reversed previously)	View More request with the same credentials (amount, card number, or service) hit the server twice within a minute.

Error Response

These are rare and are usually returned due to incorrect request formatting or general system errors.

Status Reason Message	Details
Format Error field details	Check the formatting of your request parameters
Customer requested stop of all recurring payments from specific merchant	
Transaction is in process. Please try again after some time.	Simultaneous actions cannot be performed on the same customer record. The application is unable to perform edits on a customer record while the record is in use.
An error has occurred. Your request couldn't be processed because of a technical issue. Please try again later. We apologize for the inconvenience. Our technical team has been notified	This is a system general error and you can contact us if you are experiencing these errors for more information.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Verification Response Codes

In addition to our status response codes, we have verification codes within the transaction response.

Card Verification Object

Fields	Description	Type	Filterable	Required	Readonly	Level 1	Level 2	Level 3
card.verification	Verification object	Object	No	No	Yes			
card.verification.cvv	The card security code verification response.	String	No	No	Yes	No	No	No
card.verification.address	The Address Verification System	String	No	No	Yes	No	No	No
ion.address	Verification System response.							

CVV Verification Response

Response Code	Response Description
M	CVV2 verification successful
P	CVV2 verification not performed
U	CVV2 verification not available
N	CVV2 verification fail/mismatch
S	CVV2 code not present on card

Address Verification Response

Address Verification Code is a one-character field that contains the address verification result code. This code defines the result of the address verification performed by the issuer on the address submitted in a transaction request.

Response Code	Response Definition
0	Address verification was not requested.
A	The street address submitted matches what is on file.
B	The street address submitted matches what is on file. The ZIP code does not match what was on file.
C	This service is not supported.

D	The street address submitted matches what is on file.
F	The street address and ZIP code submitted match what is on file. This applies to UK transactions.
G	The issuer does not participate in this service. This applies to non-US issuers.
I	The AVS information was not verified.
M	The street address submitted matches what is on file.
N	View More …er the street address or ZIP code match what is on file.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

3DSecure

3DSecure (3DS) is a fraud prevention security protocol that allows consumers to shop online more securely. It works by verifying that the consumer entering the credit card information on an e-commerce site is the owner or person authorized to use that credit card. How this works depends on which version of the 3DS protocol is being used. Using 3DS can provide you with chargeback protection and lower interchange rates.

> 3DS will be enabled at the merchant level. If you want to utilize this protocol contact us. Some merchants may be required to use 3DS by default for all transactions.

Integration

The following JavaScript needs to be embedded on your checkout page to collect browser data information. You will need to pass the

The following JavaScript needs to be embedded on your checkout page to collect browser data information. You will need to pass the collected data from the JavaScript in the `browserData` field in the **Create** method when creating 3DS authentication. This information is needed to increase the level of security and accuracy of the 3DS authentication.

javascript

```
<script src="/js/main.js"></script>
<script>
    ((function() {document.querySelector("#browserData").value = getBrowserData();}))();
</script>
```

> ***#browserData*** is just an example field you need to change it according to your HTML form.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Version

StartFragment

There are 2 versions of 3DS that we support. Based on what version of 3DS is offered by the issuing bank the authentication of the transaction goes through a different flow.

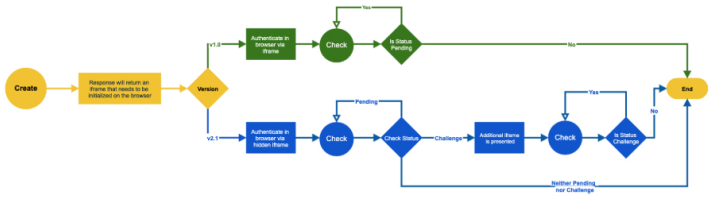
- **v1.0** - the original version that relies on redirects to iFrame and popup to complete consumer authentication. The authentication process verifies the consumer via a challenge in a window.
- **v2.1** - this is the newest version of the protocol. It's faster, less consumer obtrusive, and offers multiple ways to authenticate including passive, biometric, and two-factor authentication methods.

Verification process

Verification process

The overall process can be summarized as follows. You initiate a **Create** that will return a unique `id` and an iframe that needs to be presented in the browser. The iframe can be either hidden or visible based on the 3DS version. After that, you initiate **Check** with the provided `id`. The result will either return a `status` which will indicate the outcome of the 3DS authentication. If the status is successful you will proceed in creating a Sale or an Auth using the provided `id` in `threeds.id`.

> The version will be included in the response to the Create command.



StartFragment

v1.0 Flow

- **Create** a 3DS authentication request using card data
- We send the 3DS authentication request via form submission in an iframe in the browser
- If a challenge is required, it appears in the browser in the same iframe
- Utilizing the **Check** command will return `status` for that 3DS authentication
- Proceed by creating a **Sale** or an **Auth** for this 3DS authentication

v2.2 Flow

- Create a 3DS authentication request using card data
- We send the 3DS authentication request via form submission in a hidden iframe in the browser
- Utilize the **Check** command to verify if authentication is successful or additional challenge is required
- If an additional challenge is required we present it in a new iframe that needs to be shown in the browser
- Utilizing the **Check** command will return `status` for that 3DS authentication
- Proceed by creating a **Sale** or an **Auth** for this 3DS authentication

Always initiate Check for a specific `id` to verify the `status` before proceeding to a Sale or an Auth. Once a status changes from the initial **Pending** it will never change again. The acceptable statuses for proceeding to an **Auth** or **Sale** are **Success**, **Attempted**, and **Not Enrolled**.

EndFragment

EndFragmen

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Create

This will initiate the 3DS process and `id`, `version` and `iframe` will be presented in the response.

> Keep in mind that in very few cases a **406 - Not Acceptable** error can be returned. This happens when certain cards cannot be verified at the moment. If you are not required to use 3DS authentication as a merchant you can still proceed in processing the payment without Utilizing 3DS authentication and with no liability shift.

Object

Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the capture record	Integer	Yes	No	Yes
terminal	Terminal Object	Object	No	Yes	No
terminal.id	Dashboard Terminal ID	Integer	Yes	Yes	No
amount	Amount of the sale/auth	Number	No	Yes	No
card	Card Object	Object	No	Yes	No
card.number	Card number	Integer	No	Yes	No
card.exp	Card expiration date	Date	No	Yes	No
card.holderName	First name + Last Name	String	No	Discover Only	No
browserData	Data generated by the provided JS	Object	No	Yes	No

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

POST Create

POST Create

https://gateway.kajadashboard.com/3ds/create

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Body raw (json)

```json
{
    "terminal": {
        "id": 24
    },
    "amount": "45.00",
    "card": {
        "number": 5555555555554444,
        "exp": "12/25"
    },
    "browserData": {
```

View More

Example Request Create

```curl
curl --location 'https://gateway.kajadashboard.com/3ds/create' \
--data '{
    "terminal": {
        "id": 24
    },
    "amount": "45.00",
    "card": {
        "number": 5555555555554444,
        "exp": "12/25"
    },
```

View More

Example Response

Body Headers (1) 200 OK

```json
{
  "id": "1",
  "version": "2.1.0",
  "iframe": {
    "url": "https://gateway.kajadashboard.com/3ds/iframe/1/1/c3329956d4e9cbf189dc9b62c4bdabc6",
    "hidden": true
  },
  "params": {
    "url": "https://acs-server-sandbox.3dsintegrator.com/v2/fingerprint",
    "threeDSMethodData": "eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW
```

View More

Check

For 3DS v1.0 this will return the `status` for the specific `id`.

For 3DS v2.2 if the additional challenge is needed this will return a new `iframe` along with the `status` for the specific `id`.

A	Network stepped in for the authentication because the issuer was not participating	Yes
I	Data Share - Informational only; 3DS	No

	requestor challenge preference acknowledged	
N	Not Authenticated /Account Not Verified; Transaction denied	No
U	Authentication/ Account Verification Could Not Be Performed; Technical or other problems.	No
C	Challenge Required; Additional authentication is still required from the user	Depends on if the client completes the challenge and Issuer successfully authenticates the transaction
R	Authentication/ Verification Rejected; Issuer	No

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET Check

https://gateway.kajadashboard.com/3ds/22/check

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Body raw (json)

json

```
{
    "terminal": {
        "id": 24
    },
    "amount": "45.00",
    "card": {
        "number": 5555555555554444,
        "exp": "12/25"
    },
    "browserData": {
```

View More

Example Request Check

curl

```
curl --location --request GET 'https://gateway.kajadashboard.com/3ds/22/check' \
--data '{
    "terminal": {
        "id": 24
    },
    "amount": "45.00",
    "card": {
        "number": 5555555555554444,
        "exp": "12/25"
    },
```

View More

Example Response

Body Headers (1)

json

```
{
  "id": "1",
  "status": {
    "code": "Y",
    "description": "Authentication Successful"
  }
}
```

Test Cards

We offer you test cards for the sandbox environment so you can confirm your integration works before using real card data.

Credit Cards / Pans	Networks	Type of Transaction	Status/Success
4005519200000004 4124939999999990 4444333322221111	Visa	Frictionless	Y
4009348888881881 4485660000000007	Visa	Frictionless	N
4012000033330026 4485666666666668	Visa	Challenge/Prompt	? - Depends on the challenge completion
4012000033330026 4485666666666668	Visa	Challenge/Prompt	? - Depends on the challenge completion
4012000077777777 4124930001898619	Visa	-	Error: Card Not Enrolled
4166676667666746	Visa	Frictionless	R - Issuer rejected authentication and requested authorization not to be attempted
4917300800000000	Visa Electron	Frictionless	Y
5555341244441115 5406004444444443	MasterCard	Frictionless	Y

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Account Updater

Account Updater is a feature that automatically requests updates for tokenized cards on file in the event that a customer's card expires or is replaced. This will help you avoid or reduce billing issues.

> Account updater is not enabled on your account by default. Contact us so that we can set it up and ensure that this feature is configured in a way that best suits your business.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Card Compatibility

Whether a card can be updated using Account Updater is solely dependent on the customer's issuing bank and that bank's participation in this service. Sometimes even if the customer's card is compatible, it might not be eligible for automatic updates if the issuing bank does not participate.

> Prepaid, gift cards and cards processed with ApplePay or GooglePay are not supported.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

How It Works

Account Updater works on a transaction level whether you initiate a **Sale**, **Auth**, or **Return**. The card needs to be tokenized, the `card.save` need to be `Yes` first, and then all future transactions are going to be checked for card updates, if any. In the event that the card has been updated, we will automatically charge the new card and will return the new details in the `card` object along with some additional status details.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

POST Sale with Account Updater

https://gateway.kajadashboard.com/payment/sale

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Body raw (json)

json

```
{
    "terminal": {
        "id": 24
    },
    "amount": "1.10",
    "source": "Internet",
    "level": 1,
    "card": {
        "name": "John Wick",
        "number": 4000000000000051,
```

View More

Example Request Sale with Account Updater

curl

Example Request Sale with Account Updater

curl

```
curl --location 'https://gateway.kajadashboard.com/payment/sale' \
--header 'Content-Type: application/json' \
--data '{
    "terminal": {
        "id": 24
    },
    "amount": "1.10",
    "source": "Internet",
    "level": 1,
    "card": {
```

View More

Example Response

Body Headers (1)

json

```
{
  "id": 894,
  "amount": "1.10",
  "origin": null,
  "type": "Auth",
  "level": 1,
  "authCode": "TAS183",
  "parent": {
    "id": null
  },
```

View More

ACH Payments

The Automated Clearing House (**ACH**) is an electronic funds-transfer system that facilitates payments in the U.S. The ACH is run by the National Automated Clearing House Association (**NACHA**).

- **Debit:** Removes funds from the customer account and funds to the business account.

- **Credit:** Removes funds from the business account to fund credit to the customer.

AUTHORIZATION Bearer Token

Token	XXX

General Information

We offer a secure, streamlined ACH payment processing system. This endpoint enables you to make secure bank-to-bank transfers. We offer one-time and recurring ACH payments for various payment scenarios including corporate, consumer, telephone, and web payments.

ACH Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the ACH record	Integer	Yes	No	Yes
amount	Amount of the transaction including tax	Number	No	Yes	No
tax	The tax included in tax	Number	No	No	No
tax	The tax included in the amount in absolute value	Number	No	No	No
checkNumber	Check Number	String	Yes	No	No
traceNumber	Trace Number	String	Yes	No	Yes
transactionID	Transaction ID	Integer	Yes	No	Yes
SECCode	`PPD`, `CCD`, `WEB` or `TEL`	ENUM	No	Yes	No
transactionType	`Debit` or `Credit`	ENUM	View More	Yes	No

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

Transactions

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

SEC Codes

Standard Entry Class (SEC) code is a three letter code that describes how a customer or business authorized an ACH transaction.

Standard Entry Class (SEC) code is a three letter code that describes how a customer or business authorized an ACH transaction.

You will need to provide the SEC Code when creating an ACH transaction.

Here is a list of SEC codes we support.

> **IMPORTANT** Note that some SEC codes do not allow addendaText

Code	Debit	Credit	Addenda Text
`PPD`	Yes	Yes	Allowed
`CCD`	Yes	Yes	Allowed
`WEB`	Yes	No	Not Allowed
`TEL`	Yes	No	Not Allowed

Please note: The SEC codes that a merchant is able to process with is determined in underwriting of a merchant. If a SEC code is utilized that has not been approved for the merchant, then the transaction will be rejected. If additional SEC codes are needed, then a ticket may be opened in the dashboard to request the additional option from our underwriting team.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

Statuses

When you have successfully created an ACH Transaction, it will be given one of the following statuses, depending on the case.

When you have successfully created an ACH Transaction, it will be given one of the following statuses, depending on the case.

Status	Description
Voided	The item was voided upon entry for various administrative purposes. This is very rare, and typically warrants an interactive login or call to support
Hold	The item was accepted but placed on **hold**. This can be common and typically occurs if your daily processing limits have been exceeded. A held item is typically held for a reason and can only be released by us.
Pending	This is the "so far so good" status which means your item will be processed soon.
Submitted	A variant of **Pending**, consider **Pending** and **Submitted** to be identical. This is the most common status you will see.

Once your item has been accepted (status = **Pending** or **Submitted**) there are a few more statuses that get applied to it:

Status	Description
Transmitted	This occurs when the item has been sent out to the ACH network. The most significant thing to know about this status is that your item can no longer be edited or **VOIDED**.
Settled	For DEBIT items, this means that the debited funds have been

	successfully settled in your account. For CREDIT items, this means that the pre-funding requirements for your credit have been met and the credit is "all good."
Returned	This status occurs when an item has been returned for one of many different reasons.

different reasons.

> **Returned** is considered a "rested" status. Once a transaction has this status, it **never leaves it.**

Verification Statuses

We have internal advanced prescreen ACH payment tools and based on your account settings some transaction types will require verification. We use these to reduce the risk of fraud and returns. This service will process a submitted Routing and Account Number against our database containing positive and negative ACH/check writing history. It will validate the account is open and in good standing while also returning information about critical accounts and transactions.

In most cases, you will see your `verification.status` in **Pending**, meaning no additional verification was required, and if needed we can perform a manual check for a specific ACH transaction record.

> **IMPORTANT** If your account is setup with mandatory validation and your ACH transactions do not pass our validation tool we will not process them.

Status	Description
Pending	The transaction awaits verification.
APIError	There has been a problem with the verification API. If this occurs please contact us to investigate the problem.
Fail	The transaction has failed our verification processes.
Pass	The transaction has passed our verification processes.

Sources

Every ACH transaction record in our dashboard has a source. These are the current sources:

Source	Description
Dashboard	The transaction is initialized by the dashboard interface.
Imported	The transaction is initialized by the import CSV file.
Hosted	The transaction is initialized by the hosted form.
API	The transaction is initialized by the current ACH API.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

Actions

Actions give you the possibility to update your ACH records in specific statuses.

The available actions are as follows

Action	Description
void	This is used to void a transaction. This action will change the status of a transaction to voided. Keep in mind that void can't be used before a transaction is submitted for processing.
cancel	This is used to change a recurring transaction record to a "canceled" status.
verify	This is used to pre-verify a transaction before you submit it. We use several different algorithms for this verification process.

> **IMPORTANT** You can void a transaction only in these statuses:
>
> 1. Held

1. Held

2. Pending
3. Submitted

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

POST Action

`https://kajadashboard.com/api/ach/<id>/<action>`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request Void

```
curl
curl --location --request POST 'https://dashboard.example.com/api/ach/334/cancel'
```

Example Response

Body Headers (1) 200 OK

Body Headers (1) 200 OK

```
json
[
  true
]
```

History Log

Utilizing this endpoint you can see the full history log of an ACH transaction record.

Fields	Description	Type	Filterable	Required	Readonly
date	Date and time of the log	Datetime	Yes	No	Yes
description	Description of the log	String	No	No	Yes

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

GET Log

https://kajadashboard.com/api/ach/<id>/log

https://kajadashboard.com/api/ach/<id>/log

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request Log

curl

```
curl --location 'https://kajadashboard.com/api/ach/335/log'
```

Example Response

Body Headers (1) 200 OK

json

```
{
    "items": [
        {
            "date": "2023-04-13 08:48:24",
            "description": "Transaction Settled"
        },
        {
            "date": "2023-04-12 21:07:35",
            "description": "Transaction sent to origination File ID: 22287"
        },
```

View More

GET List

View More

GET List

https://kajadashboard.com/api/ach

Use this endpoint to list all ACH transactions in your dashboard.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request List

curl

```
curl --location 'https://kajadashboard.com/api/ach'
```

Example Response

Body Headers (1) 200 OK

json

```
{
  "items": [
    {
      "id": 326,
      "amount": "9.99",
      "tax": "1.99",
      "SECCode": "PPD",
      "accountName": "Test Wick",
      "accountNumber": "XXXXX7890",
      "accountType": "Checking",
```

View More

GET View

https://kajadashboard.com/api/ach/325

View specific ACH transaction records.

In the example we are viewing transaction with `id` 325

AUTHORIZATION Bearer Token

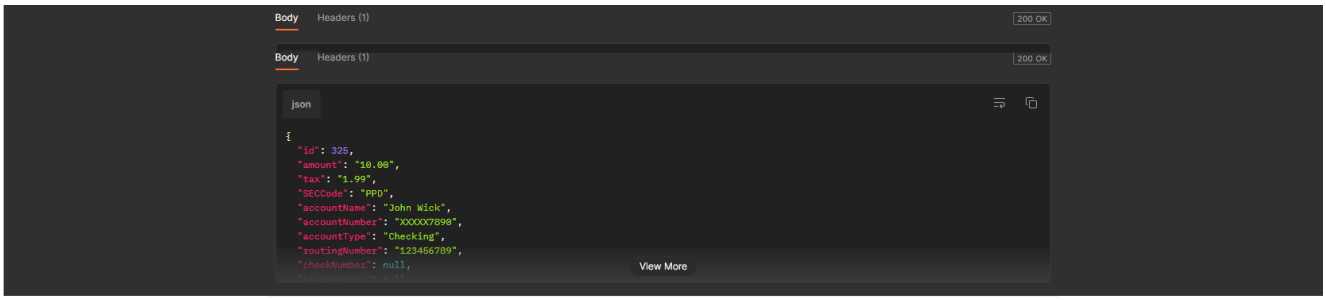
This request is using Bearer Token from folder ACH Payments

Example Request View

curl

```
curl --location 'https://kajadashboard.com/api/ach/325'
```

Example Response



POST Create ACH

https://kajadashboard.com/api/ach

Use this endpoint to create ACH Transactions.

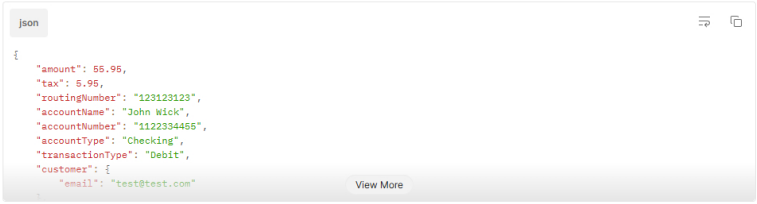
ACH Object

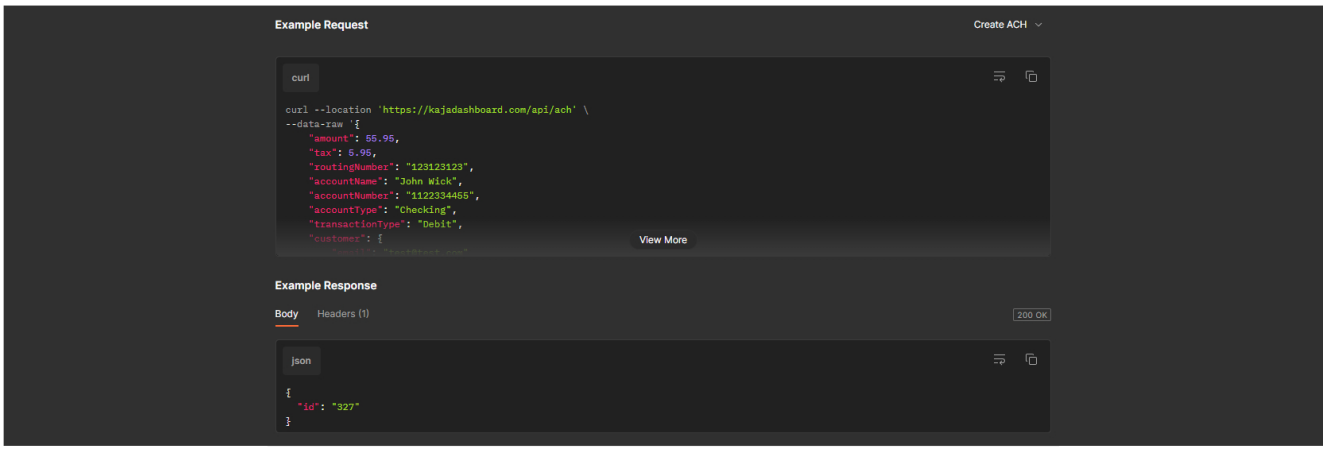
Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the ACH record	Integer	Yes	No	Yes
amount	Amount of the transaction including tax	Number	No	Yes	No
tax	The tax included in the amount in	Number	No	No	No
tax	The tax included in the amount in absolute value	Number	No	No	No
checkNumber	Check Number	String	Yes	No	No
routingNumber	Routing Number	String	Yes	Yes	No
accountName	Name of the account	String	No	Yes	No
accountNumber	Access Number	String	Yes	Yes	No
accountType	`Checking` or `Savings`	ENUM	No	Yes	No
traceNumber	Trace Number	String	Yes	No	Yes
transactionID	Transaction ID	Integer	View More	No	Yes

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)





POST Create ACH & Save Customer

POST Create ACH & Save Customer

https://kajadashboard.com/api/ach

When creating a new ACH transaction record you have the ability to save the customer and payment information to your ACH Customers for future payments. In order to do this you need to pass the `customer.save` as `Yes`.

> **IMPORTANT** When you want to save a customer from an initial transaction the `customer` object fields become required.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

json


```
{
    "amount": 9.99,
    "tax": 1.99,
    "SECCode": "PPD",
    "routingNumber": "123456789",
    "accountName": "Jack Wick's Account",
    "accountNumber": "1234567890",
    "accountType": "Checking",
    "transactionType": "Debit",
    "customer": {
```


View More

Example Request Create ACH & Save Custo...

curl


```
curl --location 'https://kajadashboard.com/api/ach' \
--data-raw '{
    "amount": 9.99,
    "tax": 1.99,
    "SECCode": "PPD",
    "routingNumber": "123456789",
    "accountName": "Jack Wick\'s Account",
    "accountNumber": "1234567890",
    "accountType": "Checking",
    "transactionType": "Debit",
```


View More

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": 3642099,
  "amount": "9.99",
  "tax": "1.99",
  "SECCode": "PPD",
  "accountName": "John Wick's Account",
  "accountNumber": "XXXXX7890",
  "accountType": "Checking",
  "routingNumber": "123456789",
  "checkNumber": null,
```


View More

POST Create ACH for an existing Customer

POST Create ACH for an existing Customer

https://kajadashboard.com/api/ach

An example request with an existing customer in the Customer Vault. Keep in mind that these are the minimum required parameters. Feel free to add any additional parameters as needed.

The Customer may has multiple accounts, when `account.id` is not passed, the system will get first available.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

json


```
{
    "amount": 9.95,
    "transactionType": "Debit",
    "dba": {
        "id": "9"
    },
    "customer": {
        "id": "42"
    }
}
```


Example Request Create ACH for an existin...

curl


```
curl --location 'https://kajadashboard.com/api/ach' \
--data '{
    "amount": 9.95,
    "transactionType": "Debit",
    "dba": {
        "id": "9"
    },
    "customer": {
        "id": "42"
    }
}'
```


Example Response

Body Headers (1) 200 OK

json


```
{
  "id": "332"
}
```


POST Create ACH for an existing Customer and specific Account

https://kajadashboard.com/api/ach

An example request with an existing customer account in the Customer Vault.

An example request with an existing customer account in the Customer Vault.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

```
json
{
    "amount": 55.95,
    "transactionType": "Debit",
    "dba": {
        "id": "5"
    },
    "customer": {
        "id": "1"
    },
    "account": {
```

View More

Example Request Create ACH for an existin...

```
curl
curl --location 'https://kajadashboard.com/api/ach' \
--data '{
    "amount": 55.95,
    "transactionType": "Debit",
    "dba": {
        "id": "5"
    },
    "customer": {
        "id": "1"
    },
```

View More

Example Response

Body Headers (1) 200 OK

```
json
{
  "id": "332"
}
```

DELETE Remove

https://kajadashboard.com/api/ach/

A transaction can be deleted if there is no status set to it from our system.

Once a transaction gets into these statuses **Pending, Submitted** or **Held** you can only perform a **VOID** action request.

> **IMPORTANT** If the transaction is in these statuses:
> **Transmitted**, **Settled** or **Returned** you cannot perform any actions to it.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request Delete

```
curl
curl --location --request DELETE 'https://kajadashboard.com/api/ach/332'
```

Example Response

Body Headers (1) 200 OK

```
json
null
```

Hosted Form

With our ACH embedded hosted form functionality you can achieve quick, efficient and secure online transactions.

The overall process to generate and prepare a hosted gateway payment form is as follows.

1. Create a POST request with the required parameters for your needs.
2. We will return the code of the hosted form, html form, html link and the url in the response.

Field	Description	Type	Required
dba	DBA Object	Object	Yes
dba.id dba	DBA ID DBA Object	Integer Object	Yes Yes
dba.id	DBA ID	Integer	Yes
amount	Payment Amount	Number	No
memo	Short note or memo	String	No
returnUrl	The return url after payment completion (i.e. success/error page)	String	No
useLogo	Use the logo of your merchant profile `Yes` or `No`	ENUM	No
sendReceipt	Send a customized receipt after successful payment `Yes` or `No`	ENUM	No
lock	Use this to lock certain fields. Available: `amount` , `memo`	Array	No

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

POST Example Request

https://kajadashboard.com/api/ach/hosted-form

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

```
json
{
    "dba":{
        "id":1
    },
    "amount":2,
    "memo":"Invoice 123",
    "returnUrl":"https://example.com",
    "useLogo":"Yes",
    "sendReceipt":"Yes"
}
```

Example Request Example Request

```
curl
curl --location 'https://kajadashboard.com/api/ach/hosted-form' \
--header 'Content-Type: application/json' \
--data '{
    "dba":{
        "id":1
    },
    "amount":2,
    "memo":"Invoice 123",
    "returnUrl":"https://example.com",
    "useLogo":"Yes",
```

View More

Example Response

Body Headers (0)

```
Text
{
  "url": "https://kajadashboard.com/ach/public/form?data=eyJpZCI6MSwidGhlbWUiOiJsaWdodCIsImFtb3VudCI6MiwibWVtbyI6Ikludm9pY2UgMTIzIiwic
  "htmlForm": "<form action=\"https://kajadashboard.com/ach/public/form\" method=\"GET\"><input type=\"hidden\" name=\"id\" value=\"1\
  "htmlLink": "<a href=\"https://kajadashboard.com/public/form?data=eyJpZCI6MSwidGhlbWUiOiJsaWdodCIsImFtb3VudCI6MiwibWVtbyI6Ikludm9pY2
  "hostedForm": "<div id=\"maverick\"></div><script>(function() { new MaverickClient({ target: 'maverick', url: e
}
```

View More

Customers

Use this endpoint to store customer information in your dashboard for quick ACH payment management.

Customers Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the customer	String	Yes	No	Yes
firstName	First name of the customer	String	Yes	Yes	No
lastName	Last name of the customer	String	Yes	Yes	No
lastName	Last name of the customer	String	Yes	Yes	No
phone	Phone number of the customer	String	Yes	No	No
email	Email address of the customer	String	Yes	No	No
address1	Address of the customer line 1	String	No	Yes	No
address2	Address of the customer line 2	String	No	No	No
city	City of the customer	String		Yes	No

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

GET List Customers

https://kajadashboard.com/apl/ach/customer

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request List Customers

Example Request List Customers

curl

```
curl --location 'https://kajadashboard.com/api/ach/customer'
```

Example Response

Body Headers (1) 200 OK

json


```
{
    "items": [
        {
            "id": 45,
            "firstName": "John",
            "lastName": "Wick",
            "phone": "+1 818-727-1234",
            "email": "john.wick@example.com",
            "address1": "9024 Some Ave",
            "address2": null,
```


View More

GET View specific Customer

https://kajadashboard.com/api/ach/customer/<id>

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request View specific Customer

curl

```
curl --location 'https://kajadashboard.com/api/ach/customer/45'
```

Example Response

Body Headers (1)

json


```
{
  "id": 45,
  "firstName": "John",
  "lastName": "Wick",
  "phone": "+1 818-727-1234",
  "email": "john.wick@example.com",
  "address1": "9024 Some Ave",
  "address2": null,
  "city": "LA",
  "state": "CA",
```


View More

POST Create Customer

https://kajadashboard.com/api/ach/customer

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

json


```
{
    "firstName": "John",
    "lastName": "Wick",
    "phone": "+1 818-777-7777",
    "email": "john@wick.com",
    "address1": "26520 Agoura Road 1st Floor",
    "address2": "",
    "city": "CALABASAS",
    "state": "CA",
    "zipCode": "91302",
```


View More

Example Request Create Customer

curl

```
curl --location 'https://kajadashboard.com/api/ach/customer' \
--data-raw '{
    "firstName": "John",
    "lastName": "Smith",
    "phone": "+1 818-777-7777",
    "email": "john@wick.com",
    "address1": "123 Agoura Road 1st Floor",
    "address2": "",
    "city": "CALABASAS",
    "state": "CA",
```

View More

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": "46",
  "firstName": "John",
  "lastName": "Smith",
  "phone": "+1 818-777-7777",
  "email": "john@wick.com",
  "address1": "123 Agoura Road 1st Floor",
  "address2": null,
  "city": "CALABASAS",
  "state": "CA",
```


View More

PUT Update Customer

https://kajadashboard.com/api/ach/customer/36

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

json

```
{
    "accountNumber": "1234567899",
    "routingNumber": "123456789",
    "address1": "26520 Agoura Road 1st Floor"
}
```

Example Request Update Customer

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/ach/customer/36' \
--data '{
    "accountNumber": "1234567899",
    "routingNumber": "123456789",
    "address1": "26520 Agoura Road 1st Floor"
}'
```

Example Response

Body Headers (1)

json

```
{
  "id": 36,
  "accountName": "John Wick",
  "accountNumber": "******7899",
  "accountType": "Checking",
  "routingNumber": "123456789",
  "accounts": [
    {
      "id": 32,
      "name": "John Wick",
```

View More

DELETE Delete a Customer

https://kajadashboard.com/api/ach/customer/<id>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request Delete a Customer

curl

```
curl --location --request DELETE 'https://kajadashboard.com/api/ach/customer/46'
```

Example Response

Body Headers (1) 200 OK

Body Headers (1) 200 OK

json

```
[]
```

Customer Accounts

Account Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the account	String	Yes	No	Yes
name	Name of the account	String	Yes	Yes	No
type	`Checking` or `Savings`	ENUM	Yes	Yes	No
routingNumber	Routing Number	String	Yes	Yes	No
accountNumber	Access Number	String	Yes	Yes	No

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

GET List Customer Accounts

https://kajadashboard.com/api/ach/customer74/account

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request List Customer Accounts

curl

```
curl --location 'https://kajadashboard.com/api/ach/customer/74/account'
```

Example Response

Body Headers (1) 200 OK

json


```
{
    "items": [
        {
            "id": 32,
            "name": "John Wick",
            "type": "Checking",
            "accountNumber": "******7890",
            "routingNumber": "123456789"
        }
    ],
```


View More

POST Create Customer Account

https://kajadashboard.com/api/ach/customer/74/account

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

json


```
{
    "name": "Jack Wick",
    "type": "Savings",
    "accountNumber": "987654321",
    "routingNumber": "123456789"
}
```


Example Request Create Customer Account

curl

curl

```
curl --location 'https://kajadashboard.com/api/ach/customer/74/account' \
--data '{
    "name": "Jack Wick",
    "type": "Savings",
    "accountNumber": "987654321",
    "routingNumber": "123456789"
}'
```

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": "36",
  "name": "Jack Wick",
  "type": "Savings",
  "accountNumber": "******4321",
  "routingNumber": "123456789"
}
```


PUT Update Customer Account

https://kajadashboard.com/api/ach/customer/74/account/36

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw (json)

json


```
{
    "type": "Checking"
}
```


Example Request Update Customer Account

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/ach/customer/74/account/36' \
--data '{
    "type": "Checking"
}'
```

Example Response

Body Headers (1)

json


```
{
  "id": 36,
  "name": "Jack Wick",
  "type": "Checking",
  "accountNumber": "*****4321",
  "routingNumber": "123456789"
}
```


DELETE Delete a Customer Account

https://kajadashboard.com/api/ach/customer/74/account/36

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Body raw


```
{
    "name": "Jack Wick",
    "type": "Savings",
    "accountNumber": "987654321",
    "routingNumber": "123456789"
}
```


Example Request Delete a Customer Account

curl

curl

```
curl --location --request DELETE 'https://kajadashboard.com/api/ach/customer/74/account/36'
```

Example Response

Body Headers (1) 200 OK

json


```
[]
```


Bank Information

Utilizing this endpoing you can view/verify the bank inforamtion associated with an ABA/Routing Number.

ABA Object

Fields	Description	Type	Filterable	Required	Readonly
id	ABA/Routing Number	Integer	Yes	No	Yes
bank	Name of the bank	String	No	No	Yes

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

GET View

https://kajadashboard.com/api/ach/aba/<aba>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request View

curl

```
curl --location 'https://kajadashboard.com/api/ach/aba/011000015'
```

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": "011000015",
  "bank": "FEDERAL RESERVE BANK OF BOSTON"
}
```



```
  "bank": "FEDERAL RESERVE BANK OF BOSTON"
}
```


ACH Settlements

Utilizing this endpoint you can view your ACH settlement details

Settlement Object

Fields	Description	Type

settlement	Settlement Object	Object
settlement.id	Settlement ID	Integer
settlement.date	Settlement Date	DateTime
settlement.amount	Settlement Amount	Number
settlement.type	Settlement Type `Origination File Settlement` , `Late Return` , `ATMVerify Fee Only Settlement` , `Reserved for Future Use` , `Reserve Release` , `Transaction Fee Only Settlement`	ENUM
approvedDebits	Approved Debit Amount	Number
approvedCredits	Approved Credit Amount	Number
fees	Fees Object View More	Object

fees Fees Object View More Object

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

GET List

https://kajadashboard.com/api/ach/settlements/<dbaId>

List settlements for a specific `dba.id`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request List

curl

```
curl --location 'https://kajadashboard.com/api/ach/settlements/1'
```

Example Response

Body Headers (1) 200 OK

json

```
{
  "items": [
    {
      "settlement": {
        "id": 88757,
        "date": "2023-07-20 08:40:00",
        "amount": "3500.00",
        "type": "Origination File Settlement"
      },
      "approvedDebits": "3500.00",
```

View More

GET View

https://kajadashboard.com/api/ach/settlements/<dbaId>/details/<settlementId>

Get settlement details for the given `dba.id` and `settlement.id`

Settlement Details Object

Fields	Description	Type	Filterable
transaction	Transaction Object	Object	No
transaction.id	Transaction ID	Integer	Yes
transaction.type	Transaction Type `debit` , `credit`	ENUM	No
amount	Total Amount	Number	Yes
customerId	Customer ID	String	No
accountName	Account Name	String	Yes

accountName Account Name String Yes

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder ACH Payments

Example Request View

curl

```
curl --location 'https://kajadashboard.com/api/ach/settlements/1/details/88757'
```

Example Response

Body Headers (1) 200 OK

json

```
{
  "items": [
    {
      "transaction": {
        "id": 92297589,
        "type": "credit"
      },
      "amount": "3500.00",
      "customerId": "D1732A4527467",
      "accountName": "Checking"
```

View More

ACH Webhooks

All the ACH group of events you can subscribe to.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

New ACH Record

Notification when a new ACH record is created.

Event Identification

json

```
{
    ...
    "module": "ach",
    "action": "create",
    "data": achObject,
    ...
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

ACH Status Update

Notification when an ACH status is updated.

Event Identification

json

```
{
    ...
    "module": "ach",
    "action": "updateStatus",
    "data": achObject,
    ...
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

New ACH Customer

Notification when a new ACH Customer has been successfully added.

json

```
{
    ...
    "module": "ach",
```

json

```
{
    ...
    "module": "ach",
    "action": "createCustomer",
    "data": achObject,
    ...
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder ACH Payments

Reporting

You can use this endpoint for all the available reports through our API.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Sandbox

IMPORTANT Please be aware the reporting section URL is different than what is referenced above in the merchant section of this documentation. Please ensure to use the correct URL endpoint for reporting as shown below.

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

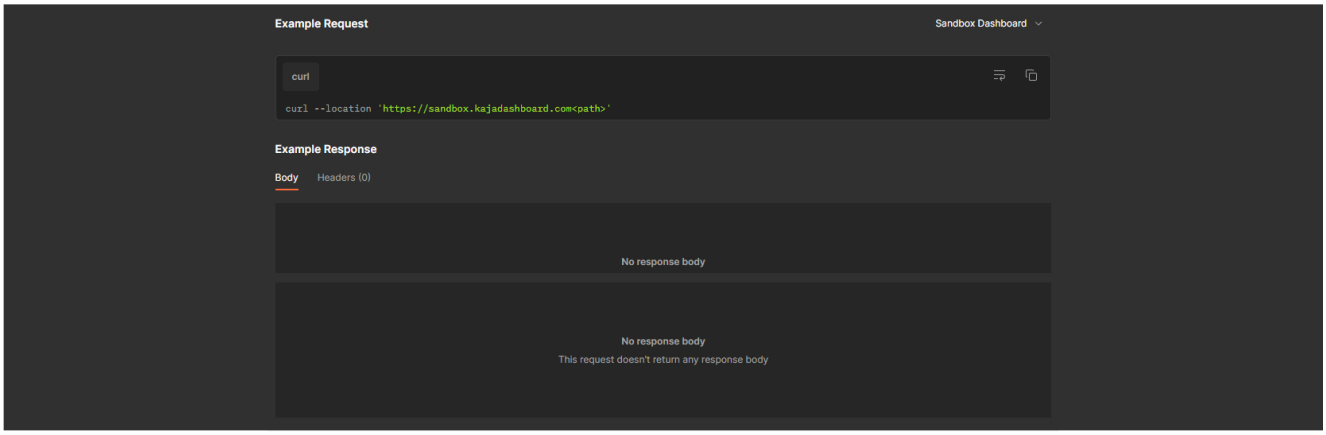
GET Sandbox Dashboard

https://sandbox.kajadashboard.com<path>

For testing purposes, please use our sandbox endpoint.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants



Authorizations

Returns a list of all authorizations for a specific `dba.id`

> **IMPORTANT** If no `date` is specified in the request as a filter we will return authorizations for the last 5 days from the current date.

Authorizations Object

Fields	Description	Type	Filterable	Required	Readonly
id	Auth Identification Number	Integer	No	No	Yes
card	Card details	Object	No	No	Yes
card.number	Card number	Integer	Yes	No	Yes
card.bin	BIN Object	Object	No	No	Yes
card.bin.bin	Bank Identification Number	Integer	Yes	No	Yes
card.bin.brand	Brand of the card	String	Yes	No	Yes
card.bin.brand	Brand of the card	String	Yes	No	Yes
card.bin.type	`Debit`, `Credit` or `ChargeCard`	ENUM	No	No	Yes
card.bin.category	Category of the card	String	No	No	Yes
card.bin.organization	Organization	String	No	No	Yes
card.bin.country	Country	Object		No	Yes

View More

Authorization Statuses

Below you can find some of the most common authorization statuses:

> Keep in mind that we may return other statuses as well this is just for your quick reference.

Status	Description
Successful approval/completion	The authorization has been successfully approved
Insufficient funds	The authorization was declined due to insufficient funds on the card
Do not honor	The authorization has been completely rejected by the issuing bank
Pick up card	The authorization has been Pick up card and needs to be held if possible. (Usually a stolen card)
Refer to Issuer	The authorization has been declined. The customer should contact their bank for more information
Invalid account number	The authorization has been declined. The card number has been incorrectly entered

AUTHORIZATION Bearer Token

Token	XXX

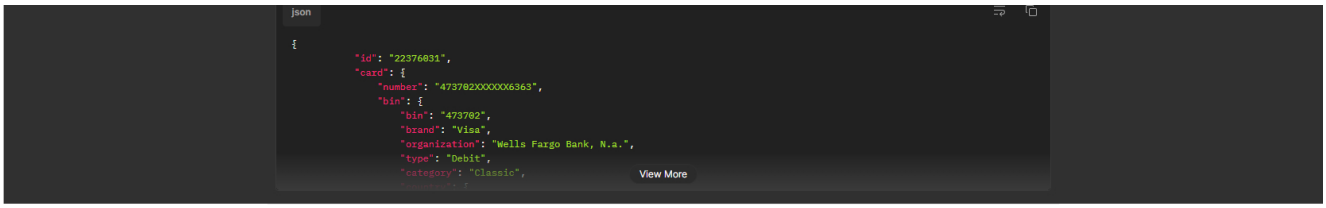
GET List

https://kajadashboard.com/api/reporting/authorizations/<dbaId>

> **IMPORTANT** If no `date` is specified in the request as a filter we will return authorizations for the last 5 days from the current date.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Authorizations



GET List with Filters

https://kajadashboard.com/api/reporting/authorizations/<dbald>?filter[card.bin.brand]=Visa&filter[terminal.number]=7000&filter[date][gte]=2020-03-24 00:00:00&filter[date][lte]=2020-03-24 23:59:59

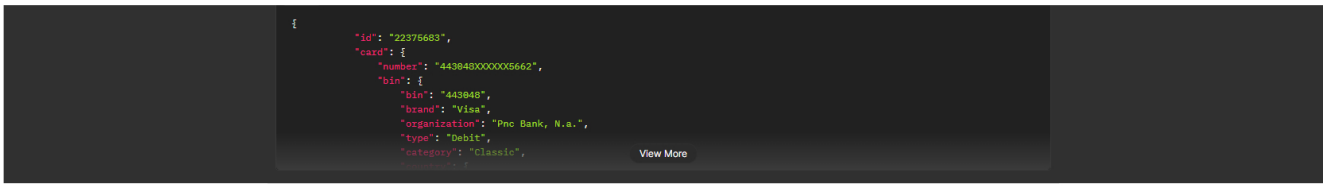
Example request to list all authorizations for a specific `dbaId` , `card.bin.brand.visa` and `terminal.number` between a date range

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Authorizations

PARAMS

filter[card.bin.brand]	Visa
filter[terminal.number]	7000
filter[date][gte]	2020-03-24 00:00:00



Batches

View batches for a specific `dbaId` . You can view the full batch summary for a specific date or ID and also all the transactions that are included in that batch.

> **IMPORTANT** If no `batch.date` is specified in the request as a filter we will return results for the last 5 days from the current date.

Batch Object

Fields	Description	Type	Filterable	Required	Readonly
id	Batch Identification Number	Integer	No	No	Yes
card	Card details	Object	No	No	Yes
card.number	Card number	Integer	Yes	No	Yes
card.bin	BIN Objecct	Objecct	No	No	Yes
card.bin.bin	Bank Identification Number	Integer	No	No	Yes
card.bin.bin	Bank Identification Number	Integer	No	No	Yes
card.bin.brand	Brand of the card	String	Yes	No	Yes
card.bin.type	`Debit` , `Credit` or `ChargeCard`	ENUM	No	No	Yes
card.bin.category	Category of the card	String	No	No	Yes
card.bin.organization	Organization	String	No	No	Yes
card.bin.country	Country	Object	No	No	Yes

View More

AUTHORIZATION Bearer Token

Token	XXX

GET List

https://kajadashboard.com/api/reporting/batches/<dbald>

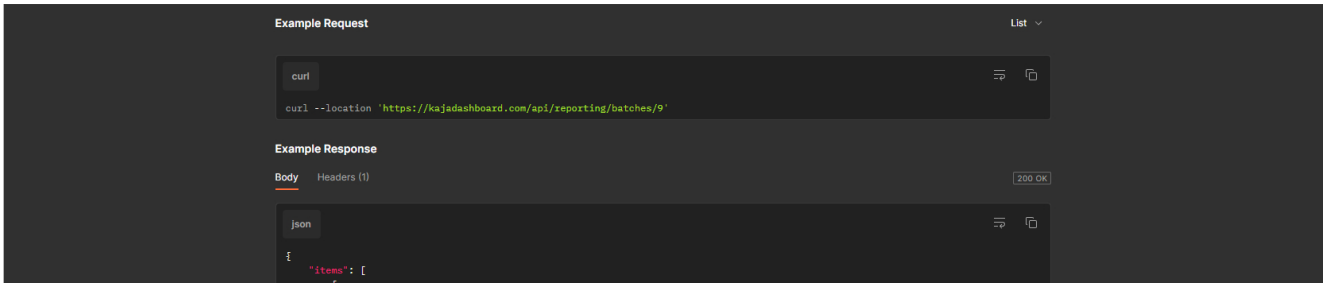
Use this endpoint to list all the batches for a specific `dba.id`

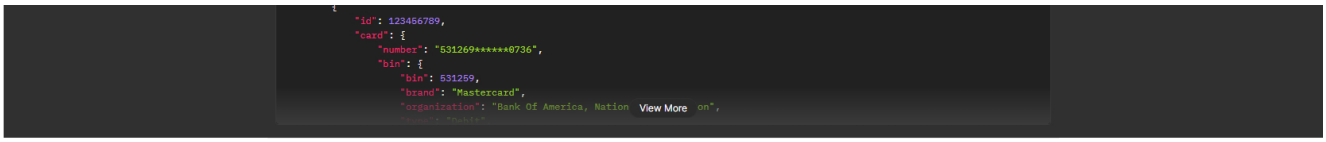
> **IMPORTANT** If no `batch.date` is specified in the request as a filter we will return results for the last 5 days from the current date.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Batches

This request is using Bearer Token from folder Batches





GET List with Filters

https://kajadashboard.com/api/reporting/batches/<dba id>?filter[card.bin.brand]=Visa&filter[date][gte]=2020-03-24&filter[date][lte]=2020-03-24

https://kajadashboard.com/api/reporting/batches/<dba id>?filter[card.bin.brand]=Visa&filter[date][gte]=2020-03-24&filter[date][lte]=2020-03-24

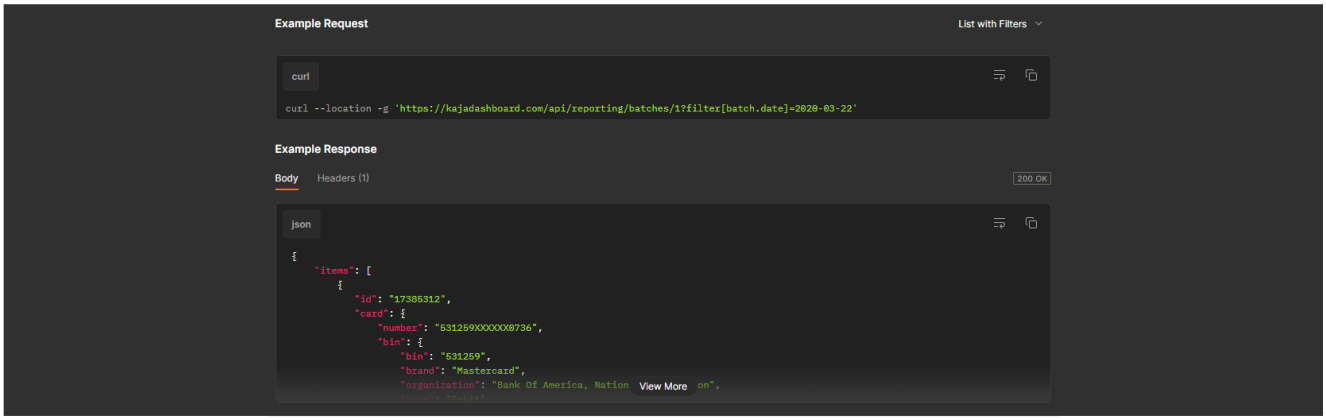
Example request to list all the transactions the are included in a batch for a specific date

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Batches

PARAMS

filter[card.bin.brand]	Visa
filter[date][gte]	2020-03-24
filter[date][lte]	2020-03-24

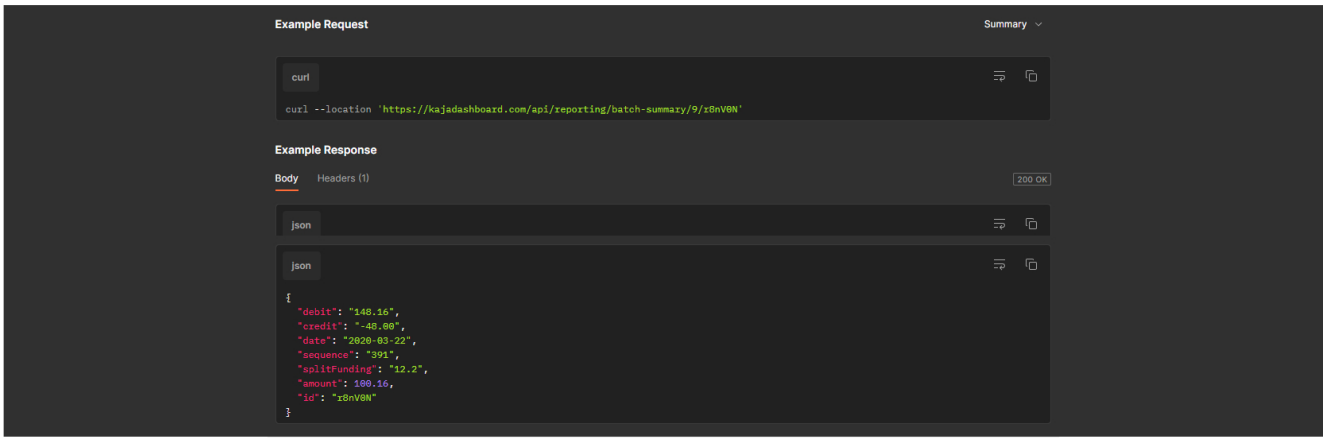


GET Summary

https://kajadashboard.com/api/reporting/batch-summary/<dba.id>/<batch.id> or <batch.date>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Batches



Payouts

You can utilize this endpoint to list all payout information for a specific `dba.Id`

Payouts List Object

Fields	Description	Type	Filterable	Required	Readonly
processingDate	Date	Date	Yes	No	Yes
deposit	Deposit Amount	Number	No	No	Yes
feeTotalAmount	Fee Amount	Number	No	No	Yes
chargeback	Chargeback Object	Object	No	No	Yes
chargeback.totalCount	Chargeback Count	Integer	No	No	Yes
chargeback.totalAmount	Chargeback Amount	Number	No	No	Yes
chargeback.totalAmount	Chargeback Amount	Number	No	No	Yes
reservedAmount	Reserve Amount	Number	No	No	Yes

releasedAmount	Released Amount	Number	No	No	Yes
batches	Batch Object	Object	No	No	Yes
batches.date	Batch Date	Date	No	No	Yes

View More

AUTHORIZATION Bearer Token

Token	XXX

GET List

https://kajadashboard.com/api/reporting/payouts/acq/<dbaId>?filter[processingDate][gte]=2023-04-01&filter[processingDate][lte]=2023-04-30

> **NOTE** Keep in mind that running this request without specifying a processingDate filter will return all available records for the current month

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Payouts

PARAMS

filter[processingDate][gte]	2023-04-01
filter[processingDate][lte]	2023-04-30

Example Request List with Date Range

curl

```
curl --location -g 'https://kajadashboard.com/api/reporting/A...9?filter[processingDate][gte]=2023-04-01&filter[processingDate
```

View More

Example Response

Body Headers (1) 200 OK

json


```
{
  "items": [
    {
      "processingDate": "2023-04-12",
      "deposit": 9731.14,
      "feeTotalAmount": 0,
      "chargeback": {
        "totalCount": 0,
        "totalAmount": 0
      },
```


View More

GET View

https://kajadashboard.com/api/reporting/payout/acq/<dbaId>/<processingDate>

View detailed payout information for a specific date

Payouts View Object

Fields	Description	Type	Filterable	Required	Readonly
processingDate	Date	Date	Yes	Yes	Yes
deposit	Deposit Amount	Number	No	No	Yes
fees	Fee Object	Object	No	No	Yes
fees.items	Fees Items Object	Object	No	No	Yes
fees.items.cardType	Fee Card Type	String	No	No	Yes
fees.items.amount	Fee Amount	Number	No	No	Yes
fees.totalAmount	Fees Total Amount	Number	No	No	Yes
chargebacks	Chargebacks Object	Object	No	No	Yes
chargebacks.items	Chargebacks Items Object	Object	No	No	Yes
chargebacks.items.referenceNumber	Chargeback Reference Number	String	No	No	Yes
chargebacks.items.amount	Chargeback Amount	Number	No	No	Yes
chargebacks.totalCou	Chargebacks Total	Integer		No	Yes

View More

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Payouts

Example Request View

curl

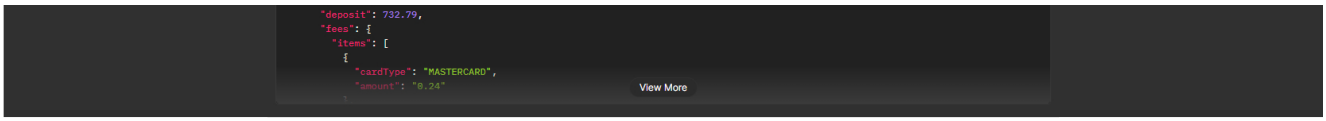
```
curl --location 'https://kajadashboard.com/api/reporting/payout/acq/1/2023-04-05'
```

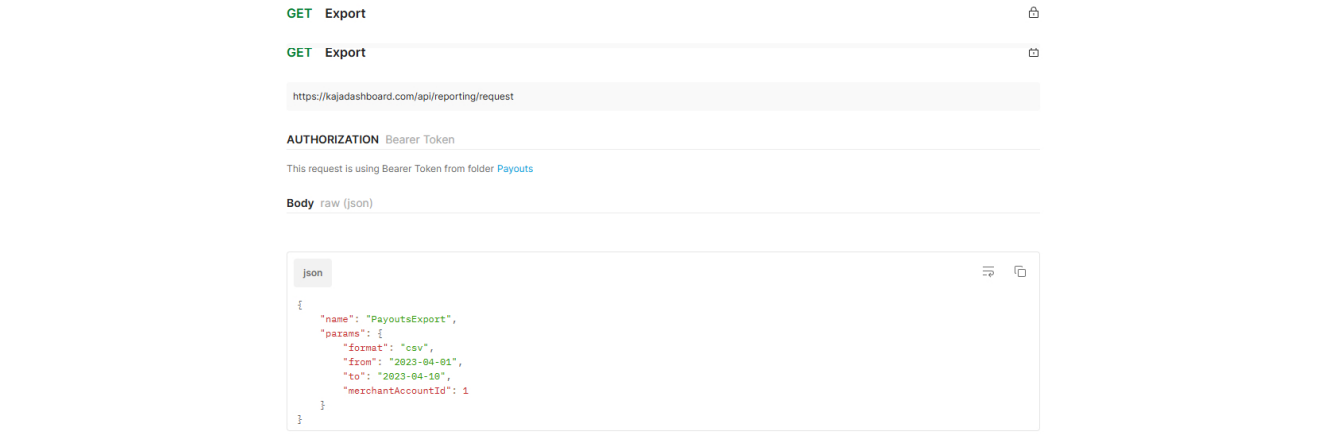
Example Response

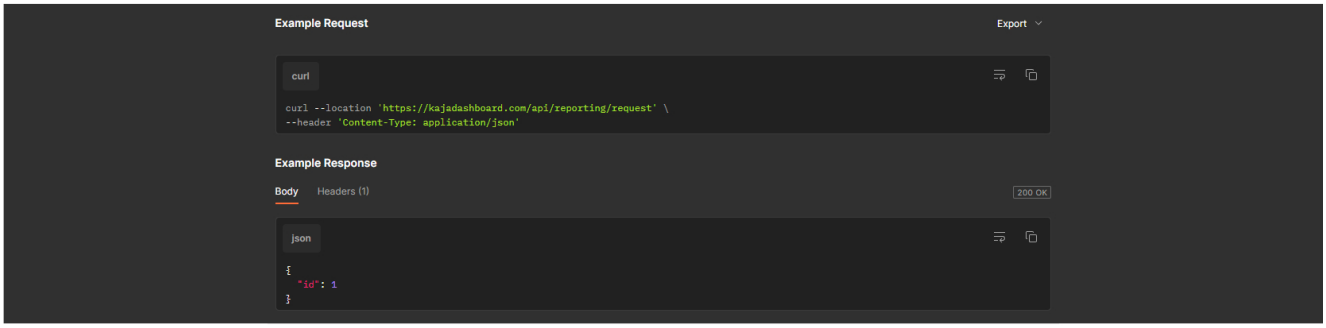
Body Headers (1) 200 OK

json


```
{
  "items": [
    {
      "processingDate": "2023-04-05",
```







Chargebacks

You can use this endpoint to manage all chargeback-related information.

You can do the following:

- List all chargebacks for a specific `dba.id`
- View specific chargeback information for `chargeback.id`
- Dispute a `case.id` for a specific `chargeback.id`

How to use our Chargeback system

Our chargeback system is designed around chargeback cases. When the first case arrives we create a chargeback with an `id.` Every new case that arrives for a specific chargeback `id` updates the chargeback information and gets listed in the `cases` object.

> **IMPORTANT** A specific `chargeback.id` can have more than one cases associated with it, as new information can arrive.

We suggest the current flow:

1. List your chargebacks. If a certain `chargeback.id` has `attention` =Yes, you can dispute it and upload evidence.

1. List your chargebacks. If a certain `chargeback.id` has `attention` =Yes, you can dispute it and upload evidence.

2. View the specific `chargeback.id`, this will list one or more cases associated with the chargeback. Make note of the `case.id` that has `attention` =Yes.
3. Use our **Attachments** and upload temporary file/files for the dispute.
4. **Create** a dispute to a `case.id` for a specific `chargeback.id`
5. **Preview** the created dispute if needed before submitting it.
6. Use the Dispute **Submit** endpoint to submit the dispute file to a specific `chargeback.id`

Chargeback Object

Fields	Description	Type
id	The unique ID that is assigned by our system to every chargeback	Integer
dba	Object	Object
dba.id	The ID of the DBA	Integer
type	`Chargeback` or `Retrieval`	ENUM
amount	The amount of the retrieval or chargeback. This is not always equal to the amount of the transaction.	Number
card	Object	Integer
card.brand	Card Brand	String
card.number	Card Number	String
date	Object	Object
date.posted	The post date a… / the issuer. View More	Date

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Disputes

By utilizing this endpoint you can dispute specific chargeback cases.

The process is as follows:

1. First, use our standard Attachments endpoint but you need to pass an additional URL parameter `temporary`. This will return a unique `id`.
2. After you have your temporary file id you can create the Dispute for a specific `chargeback.id` and the desired `case.id`.

You can see the examples below

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

POST Upload a temporary file

https://kajadashboard.com/api/attachment/upload/temporary

In order to upload a temporary file and submit a dispute to a chargeback case you need to utilize our standard Attachments endpoint and pass an additional URL parameter `temporary`

In order to upload a temporary file and submit a dispute to a chargeback case you need to utilize our standard Attachments endpoint and pass an additional URL parameter `temporary`

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request — Upload a temporary file

curl

```
curl --location 'https://kajadashboard.com/api/attachment/upload/temporary' \
--form '=@"7L-_Am9Yd/disputeinfo.pdf"'
```

Example Response

Body Headers (1)

json

```
{
  "id": "47c2fd1ebac89e2d5244b847c9b869fb",
  "name": "disputeinfo.pdf",
  "type": "application/pdf",
  "size": 245311
}
```

POST Create a Dispute

POST Create a Dispute

https://kajadashboard.com/api/reporting/chargeback/<chargebackId>/dispute/<casesId>/create

After uploading the file/files you need to create the dispute by assigning all the temporary `attachment.id` to the `case.id`.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request — Create a Dispute

curl

```
curl --location 'https://kajadashboard.com/api/reporting/chargeback/5545/dispute/6233665/create' \
--header 'Content-Type: application/json' \
--data '["47c2fd1ebac89e2d5244b847c9b869fb"]'
```

Example Response

Body Headers (1) 200 OK

json

```
{
  "disputeId": "2023101033545_20230516123226"
}
```

GET Preview

GET Preview

https://kajadashboard.com/api/reporting/chargeback/<chargebackId>/dispute/<casesId>/preview/<disputeId>

Use this endpoint if you want to preview your dispute before submitting it.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request — Preview

curl

```
curl --location 'https://kajadashboard.com/api/reporting/chargeback/5545/dispute/6233665/preview/2023101033545_20230516123226'
```

Example Response

Body Headers (0) 200 OK

No response body

This request doesn't return any response body

POST **Submit**

https://kajadashboard.com/api/reporting/chargeback/<chargebackId>/dispute/<casesId>/submit/<disputeId>

Use this endpoint to submit your dispute

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request Submit

curl

```
curl --location --request POST 'https://kajadashboard.com/ap                chargeback/5545/dispute/6233665/submit/2023101033545_202305161
```

View More

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": 5336147,
  "name": "IN_5671_CASES_2023101033545_20230516123226.pdf",
  "tyne": "application/pdf",
  "id": 5336147,
  "name": "IN_5671_CASES_2023101033545_20230516123226.pdf",
  "type": "application/pdf",
  "size": 16616,
  "createdOn": "2023-05-16 12:34:37"
}
```


Webhooks

Use webhooks to be notified about events that happen in your account.

Webhook Object

Fields	Description	Type
id	Unique ID of the event	Integer
module	Event group	String
action	Event identifier	String
date	Date of the event	Datetime
data	Event Data	Object

Example

javascript

```
{
    "id": "1",
    "module": "ach",
    "action": "create",
    "date": "2019-08-03 19:31:55",
    "data": {...}
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Signatures

We sign the webhook events that are sent to your endpoints. We do so by including a signature in each event's `Webhook-Signature` header. This allows you to verify that the events were sent by us, not by a third party.

> **IMPORTANT** You are not required to implement this kind of signature check, but it's **strongly recommended**.

`Webhook-Signature` is a SHA-512 sum of following fields `<webhookSignature><id><module><action><date>`

You can view and manage your **Webhook Signature** in your Dashboard. Keep in mind that every **Webhook URL** has its own `<webhookSignature>`

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Events

Groups of events you can subscribe to.

AUTHORIZATION Bearer Token

Groups of events you can subscribe to.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Chargeback

Chargeback group of events you can subscribe to.

Chargeback Object

Fields	Description	Type
id	The unique ID assigned by our system to every case.	Integer
dba	Object	Object
dba.id	The ID of the DBA	Integer
identifier	The Merchant Identification Number (MID)	Integer
disputeId	We assign a unique ID to the first case of a dispute. Every related case update will have a unique id and the same disputeId.	Integer
merchantRefId	A reference number provided by merchant or their vendor for this transaction.	String
merchantEmail	Merchant email address	String
number	The unique number assigned by the bank system to every case.	String
type	`Chargeback o` View More `]`	ENUM

Example

json

```
{
    "id": "200",
    "dba": {
        "id": "1217"
    },
    "identifier": "201100000000",
    "disputeId": "100",
    "merchantRefId": "0123456",
    "merchantEmail": "jhon@example.com",
    "number": "2019248118022",
    "type": "Chargeback"
```

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

New Chargeback Record

Notification when a new Chargeback record is created.

Event Identification

json

Event Identification

json

```
{
    ...
    "module": "chargeback",
    "action": "create",
    "data": chargebackObject,
    ...
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

Chargeback Update

Notification when a Chargeback is updated.

Event Identification

json

```
{
    ...
    "module": "chargeback",
    "action": "update",
    "data": chargebackObject,
    ...
}
```

```
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

View

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET List

https://kajadashboard.com/api/reporting/chargebacks

You can utilize this endpoint to list all chargebacks

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request List

curl

curl --location 'https://kajadashboard.com/api/reporting/chargebacks'

curl

curl --location 'https://kajadashboard.com/api/reporting/chargebacks'

Example Response

Body Headers (1) 200 OK

json


```
{
  "items": [
    {
      "id": 5545,
      "dba": {
        "id": 9
      },
      "type": "Chargeback",
      "amount": "39.00",
      "card": {
```


View More

GET View

https://kajadashboard.com/api/reporting/chargeback/<id>

By utilizing this endpoint you can view detailed information for a specific chargeback

Fields	Description	Type
id	The unique ID that is assigned by our system to	Integer

Fields	Description	Type
id	The unique ID that is assigned by our system to every case.	Integer
dba	Object	Object
dba.id	The ID of the DBA	Integer
mid	The Merchant Identification Number (MID)	Integer
merchantRefId	A reference number provided by the merchant or their vendor for this transaction.	String
merchantEmail	Merchant email address	String
type	`Chargeback` or `Retrieval`	ENUM
visaRDR	`Yes` or `No` - Identifies if the chargeback is Visa RDR	ENUM
authCode	The authorization code passed with the settlement record	String
acquirerReferenceNumber	An identification number for a credit card transaction assigned by the acquirer.	Object
gateway	Object	Object

View More

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request View

curl

curl --location 'https://kajadashboard.com/api/reporting/chargeback/5545'

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": 5545,
  "dba": {
    "id": 9
  },
  "mid": 40111950369,
  "merchantRefId": "SSS-2023040518402732",
  "merchantEmail": "",
  "type": "Chargeback",
  "visaRDR": "No",
```


View More

Statements

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET List

https://kajadashboard.com/api/reporting/statements/<dbaId>

https://kajadashboard.com/api/reporting/statements/<dbaId>

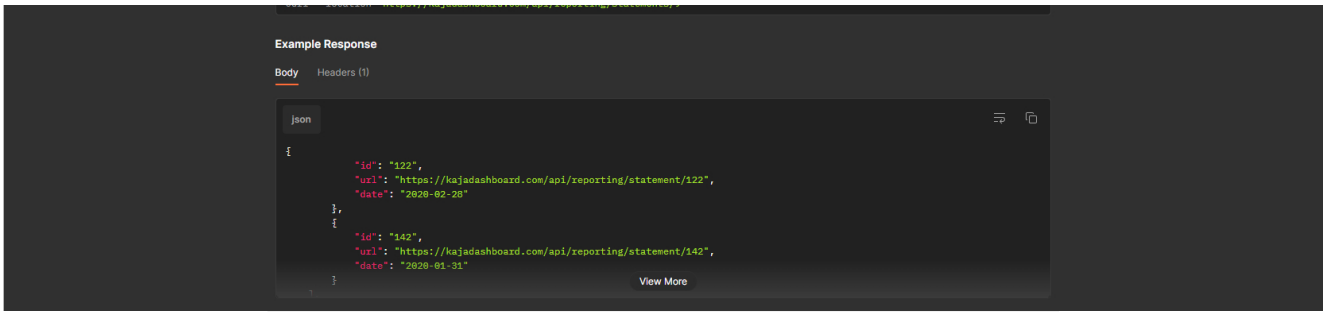
AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request List

curl

curl --location 'https://kajadashboard.com/api/reporting/statements/9'



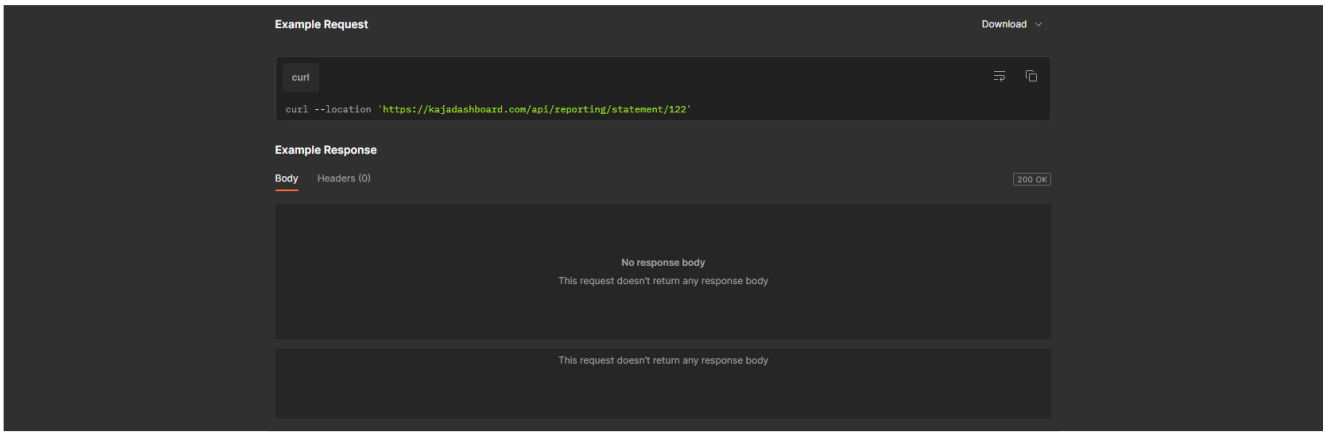
GET Download

https://kajadashboard.com/api/reporting/statement/<statementId>

To download a specific statement, use its corresponding `id`. All the statements are in pdf format.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants



Reserve

You can use this endpoint to view reserve activity for a specific `dba.id`

Reserve Object

Fields	Description	Type	Filterable	Required	Readonly
id	Reserve id	Integer	No	No	Yes
type	Reserve amount activity type `Debit` or `Credit`	ENUM	No	No	Yes
amount	The amount that is either credited or debited to the reserve.	Number	No	No	Yes
reserve	Current accumulated reserve amount	Number	No	No	Yes
description	Reserve funding activity type	ENUM	No	No	Yes
date	Datetime of reserve activity	Datetime	No	No	Yes

Reserve Descriptions

Below you can find all the reserve descriptions that we return

Status	Description
Calculated Reserve	An amount has been withheld and added to the reserve
Calculated Release	An amount has been released from the reserve
One Time Reserve	One time amount has been withheld and added to the reserve
One Time Release	One time amount has been released from the reserve
Record Adjustment Reserve	One time adjustment amount has been withheld and added to the reserve (usually due to an accounting error)
Record Adjustment Release	One time adjustment amount has been released from the reserve (usually due to an accounting error)
Transfer Funds on Reserve	An amount has been transferred to the reserve. This is usually done when a merchant begins with a fixed reserve

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET List

https://kajadashboard.com/api/reporting/reserve/<dbaId>

https://kajadashboard.com/api/reporting/reserve/<dbaId>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request List

curl

```
curl --location 'https://kajadashboard.com/api/reporting/reserve/9'
```

Example Response

Body Headers (1)

json


```
{
    "items": [
        {
            "id": "1700",
            "type": "Credit",
            "amount": "293.65",
            "reserve": "89908.71",
            "description": "Calculated Reserve",
            "date": "2018-06-06 20:22:22"
        },
```


View More

Fraud Report

At the end of each month, we create a customized and highly detailed fraud report for each merchant. To download it use the following endpoint by providing the `dbaId` and `date` for the desired fraud report pdf.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET Generate

https://kajadashboard.com/api/reporting/fraud/<dbaId>/<year-month>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request Generate

curl

```
curl --location 'https://kajadashboard.com/api/reporting/fraud/9/2023-01'
```

Example Response

Example Response

Body Headers (0) 200 OK

No response body

This request doesn't return any response body

Visa Verifi

PRODUCT SUMMARY

Once a Merchant obtains a merchant account (MID) through us and is registered through us on Verifi*, they can leverage this API to access Order Insight and Notification services to drive dispute prevention. This guide includes details needed to integrate with us and to leverage Verifi services. There are two ways you can integrate with us and utilize Visa Verifi's services.

1. Directly utilizing our Payment Gateway services
2. Using an external third-party Payment service.

**please contact us if wanting to register as a merchant for Verifi services*

PRODUCT OVERVIEW

Order Insight provides Issuers and Cardholders with real-time information covering purchase details and Merchant information to enrich the Cardholder experience, eliminate billing confusion, and prevent friendly fraud. A single integration allows data to be shared and used for three distinct purposes – Cardholder digital experience, Call Center Experience, and pre-dispute Compelling Evidence fraud dispute deflection.

Compelling Evidence introduces new rules that help merchants combat first-party misuse (friendly fraud) by weighting a cardholder's non-fraud prior transactions with the merchant as evidence of likely legitimacy of the disputed transaction. If a clear link between the merchant

Compelling Evidence introduces new rules that help merchants combat first-party misuse (friendly fraud) by weighting a cardholder's non-fraud prior transactions with the merchant as evidence of likely legitimacy of the disputed transaction. If a clear link between the merchant and the cardholder can be identified, the issuing bank is much more likely to rule the dispute in favor of the business; however, submitting evidence does not *guarantee* an auto-win as issuing banks still have discretion as to whether a dispute is ruled in favor of a merchant.

Visa's qualifying criteria to demonstrate the link is as follows:

For all disputes with Visa reason code 10.4, merchants can increase their odds of winning by providing two prior transactions with the same cardholder where:

- No fraud activity has been reported
- The same payment credential was used
- Processing occurred more than 119 calendar days and fewer than 366 calendar days prior to the disputed transaction processing date

Merchants must provide:

1. Product descriptions for the disputed charge and the two chosen transactions - reference product object table below.
2. The device ID, device fingerprint or the IP address of the customer
3. Proof that one or more of the following match across both non-fraud transactions and the disputed transaction:
    a. Customer email

b. Delivery address
c. Device ID / device fingerprint
d. IP address

Our systems will detect and prefill other required information and will upon receiving qualifying dispute query for previous transactions that meet the required criteria.

To increase chances of qualifying, update your integration to share information included in the following sections for the Verifi API.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

This folder is using Bearer Token from folder Merchants

Submiting External Transactions

If you are not utilizing our payment gateway services and still wish to participate in the Visa Verifi program you can do so by sending us all of your transactions and the corresponding necessary information. To do so you can utilize this endpoint and leverage our systems to do the rest automatically. The transactions that you submit will become available for lookup through the Visa Verifi program.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

POST **Submit**

https://kajadashboard.com/api/visa-verifi/<merchantAccountId>/transaction

Fields	Description	Type	Filterable	Required
terminalId	ID of the terminal for which you're submitting external transaction	Integer	No	Yes
transactionId	Unique ID of the transaction in the external system	Integer	String	No
amount	Total Amount of the transaction	Decimal	No	Yes
date	Date of the transaction	Datetime	No	Yes
date	Date of the transaction	Datetime	No	Yes
cardBin	The first six digits of the card used in the transaction	Integer	No	Yes
cardNumber	The last four digits of the card used in the transaction	Integer	No	Yes
cardExpDate	Card Expiration Date	Date	No	Yes
authCode	Transaction Auth Code	String	No	No
arn	Transaction ARN Number	String	No	No

View More

Product Object

This is the structure of each product object. When submitting the external transaction please provide a list of these objects inside `order.products` field.

Fields	Description	Type	Filterable	Required
name	Name of the purchased product	String	No	Yes
price	Price of a single product	Decimal	No	Yes
quantity	Quantity of purchased products	Integer	No	Yes

Example Product Object

json

```
{
    "name": "Hosting",
    "price": "24.00",
    "quantity": "1.00",
}
```

Tax Object

This is the structure of each tax object. When submitting the external transaction please provide a list of these objects inside `order.taxses` field.

Fields	Description	Type	Filterable	Required
type	Type of tax	String	No	Yes
amount	Tax amount	Decimal	No	Yes
rate	Tax rate	Integer	No	Yes
category	Tax category	String	No	Yes

Example Tax Object

json

```
{
    "type": "VAT",
    "amount": "2.40",
    "rate": "10.00",
    "category": "VAT"
}
```

Billing Object

This is the structure of the billing object. It provides data about the billed person. When submitting the external transaction please provide it under `order.billing` field.

Fields	Description	Type	Filterable	Required
name	Name of the billed person	String	No	Yes
country	Country	String	No	Yes
state	State	String	No	No
city	City	String	No	Yes
street	Street	String	No	Yes

zip	Zip	String	Number	No

Example Billing Object

```json
{
    "name": "Craig  Dewayne",
    "country": "United States",
    "state": "Arkansas",
    "city": "Pea Ridge",
    "street": "1302 Spruce St",
    "zip": "72751"
}
```

Shipping Object

This is the structure of the Shipping object. It provides data about the order shipping. When submitting the external transaction please provide it under `order.shipping` field.

Fields	Description	Type	Filterable	Required
fromZip	Shipping from zip	String	Number	No
toAddress	Delivery address information	String	No	Yes
toCity	Delivery city information	String	No	No
toZip	Shipping to zip	String	Number	No
toCountry	Shipping Country	String	No	Yes

Example Shipping Object

```json
{
    "toCountry": "United States",
    "toCity": "Pea Ridge",
    "toAddress": "1302 Spruce St",
    "toZip": "72751",
    "fromZip": "31151"
}
```

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request Submit

```curl
curl --location --request POST 'https://kajadashboard.com/api/visa-verifi/<merchantAccountId>/transaction'
```

Example Response

Body Headers (0)

No response body

No response body

This request doesn't return any response body

GET List

https://kajadashboard.com/api/visa-verifi/<mercantAccountId>/order-insights

You can utilize this endpoint to list all Visa Verifi requests either as OrderInsight or Compelling evidence

List all requests from Visa Verifi for a specific merchant's account for a specific period.

Response Object

Fields	Description	Type	Filterable	Required	Readonly
id	ID	Integer	No	No	Yes
insightId	Insight ID in Visa System	Integer	No	No	Yes
source	Source of the request: OI=orderInsight, CI=compelling evidence	String	Yes	No	Yes
cardBin	The first 6 digits of the Card	Integer	No	No	Yes
cardLast4	The last 4 digits of the Card	Integer	No	No	Yes
paymentType	Card Brand	String	No	No	Yes
transactionDate	TransactionDate according to VISA	Date	No	No	Yes
transactionAmount	Transaction amount	Decimal	No	No	Yes
acquirerBin	Visa-assigned Bank Identification Number	Integer	No	No	Yes

View More

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Example Request List

```curl
curl --location 'https://kajadashboard.com/api/visa-verifi/2/order-insights/'
```

Example Response

Body Headers (0)

Text


```
{
    "items": [
        {
            "id": 8,
            "insightId": "efdc350e-5dcf-43ae-b925-5c076b558#19",
            "source": "OI",
            "cardBin": 453922,
            "insightId": "efdc350e-5dcf-43ae-b925-5c076b558#19",
            "source": "OI",
            "cardBin": 453922,
            "cardLast4": "4489",
            "paymentType": "VISA",
            "transactionDate": "2023-08-23 11:28:22",
```


View More

Other Reports

Based on the reports that are available to you, you can utilize this endpoint to export data-intensive reports from our system.

> **IMPORTANT** Reports are executed in background mode and you need to check the status of each report utilizing or Status Reports endpoint.

Reporting Object

Fields	Description	Type	Filterable	Required	Readonly
name	Unique Name of the report	String	No	Yes	No
params	Report parameters	Object	No	Yes	No
params.format	File format of the report. Example: `csv`, `xlsx`	ENUM	No	Yes	No
params.xxx	Other report parameters	String	No	No	No

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

GET Request a Report

https://kajadashboard.com/api/reporting/request

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants

Body raw

```
{
    "name": "ResidualExport",
    "params": {
        "format": "csv",
        "date": "2022-02",
        "isPublished": "Yes",
        "isHeld": "No",
        "dbaId": 1,
    }
```

Example Request Request a Report

curl

```
curl --location 'https://kajadashboard.com/api/reporting/request' \
--header 'Content-Type: application/json'
```

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": 1
}
```


GET Check Status

https://kajadashboard.com/api/reporting/status/<reportId>

Use this endpoint to check the status of a specific requested report.

AUTHORIZATION Bearer Token

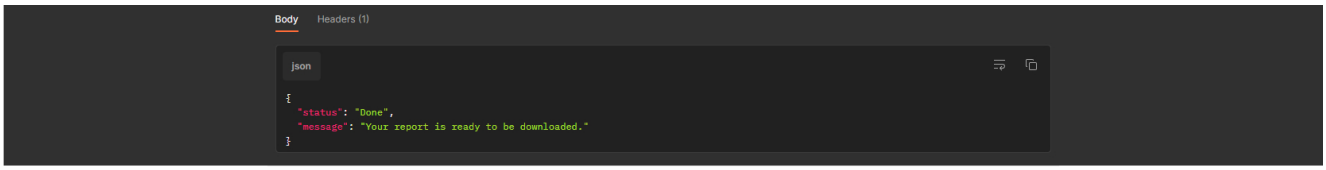
This request is using Bearer Token from folder Merchants

Example Request Check Status

curl

```
curl --location 'https://kajadashboard.com/api/reporting/status/1'
```

Example Response



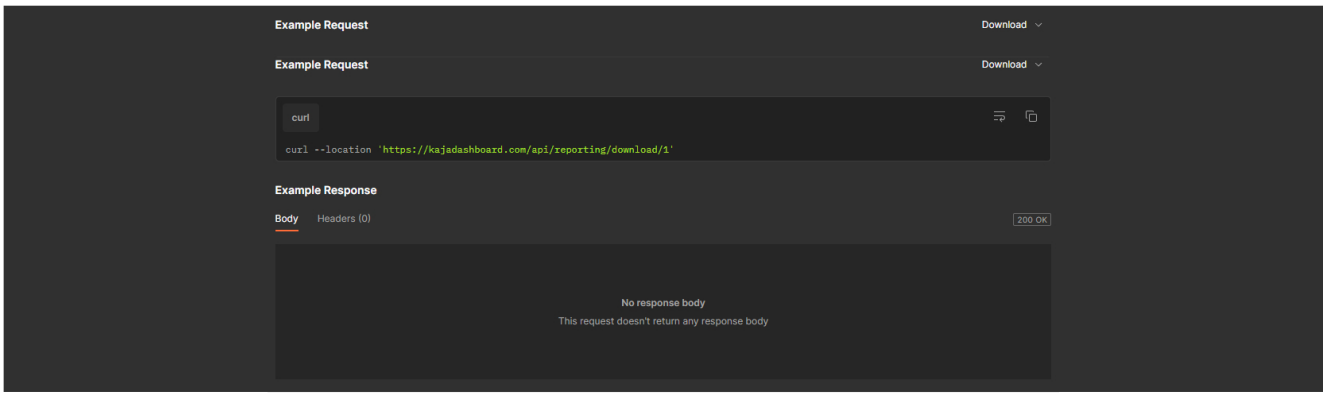
GET Download

https://kajadashboard.com/api/reporting/download/<reportId>

After checking the status of a requested report and when it is ready to be downloaded, you can utilize this endpoint.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Merchants



Gateway Test Cards

We offer you test cards for the sandbox environment so you can confirm your integration works before using real card data.

> **IMPORTANT** For AVS you can use the Address: 8320 and ZIP: 85284

Card Number	Network	Expiration Date	CVV
4539225011794489	VISA	12/30	999
4957587837877027	VISA	12/30	999
4957587837877027	VISA	12/30	999
4632633008802809	VISA	12/30	999
4539225011794430	VISA	12/30	999
5296345017707556	MASTERCARD	12/30	998
5309572686772257	MASTERCARD	12/30	998
6011421812165456	DISCOVER	12/30	996
6011891662320957	DISCOVER	12/30	996
340200784265287	AMEX	12/30	9997
371632530176123	AMEX	12/30	9997
4111111111111111	VISA	12/30	999
4000111111111115	VISA	12/30	999

View More

Partial Approval

There are cases where you will encounter partial approvals, such as gift cards etc. To test this use the following:

Card Number	Network	Expiration Date	CVV	Amount	Approved Amount
2223000048400011	MASTERCARD	12/25	998	11.10	5.55

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Merchants

This folder is using Bearer Token from folder Merchants

Platforms

Platforms represent different types of partners, such as resellers and third-party solution providers, utilizing our platform for mutual merchants. Board your merchants, view reporting, receive webhooks for important updates, process payments on behalf of, and perform other valuable functions to offer a more integrated and seamless payment experience by leveraging our dashboard and infrastructure.

Sandbox

GET Sandbox Dashboard

https://sandbox.kajadashboard.com<path>

For testing purposes, please use our sandbox endpoint.

```
curl
curl --location 'https://sandbox.kajadashboard.com<path>'
```

Example Response

Onboarding

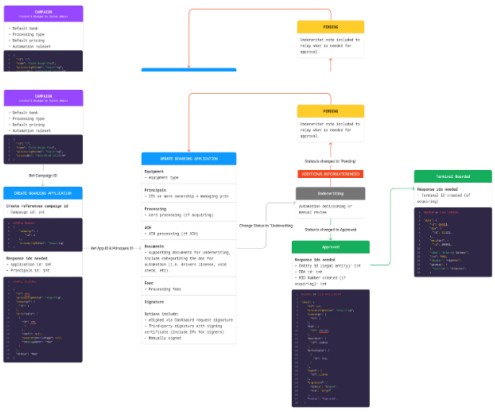
For platforms, such as resellers which generally include ISO and ISV partners, this enables the ability to systematically onboard merchants via our API. This enables platforms to better manage the interface and experience for boarding their merchants to our system.

Boarding Applications

This endpoint provides an interface for working with Boarding Applications. Each application belongs to a pre-created campaign. A campaign controls several aspects of a Boarding Application, such as pricing fees, the type of processing and several other parameters.

The overall boarding process is as follows:

1. Create a Boarding Application and get the Boarding Application `id`
2. Update the Boarding Application data using the `id`
   a. Update Principal using the `principals.id`
   b. Add Principals or Mass Update/Insert Principals if needed
   c. Update the Processing/ACH section
   d. Upload necessary documents and assign them to the Boarding Application
   e. Update Fee where necessary
3. Request **eSign** or use **AutoSign** functionality to get needed signatures from principals.
4. Update application `status` to `Underwriting` or use **AutoUnderwriting** functionality
5. Check application `status` for changes.



Expand Flow Chart

> Keep in mind that after creating an application you can get the `url` and send it for merchant completion. The link will initiate a web interface which allows the Boarding Application to be completed by the merchant online.

Application Object

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the Boarding Application	Integer	Yes	No	Yes
processingMethod	`Acquiring`, `ACH`, `All`	ENUM	Yes	Yes	Yes
campaign	Object	Object	No	Yes	Yes
campaign.id	Unique ID of the Campaign to which the Boarding Application is assigned	Integer	Yes	Yes	Yes
company	Object	Object	No	Yes	Yes
company.name	Corporate / Legal Name	String	Yes	Yes	No
company.type	`Individual`, `Partnership`,	ENUM	No	Yes	No

View More

AUTHORIZATION Bearer Token

Token	XXX

Campaigns

A Boarding Application always starts by choosing a campaign and `processingMethod`. We have two main types of campaigns `Standard` or `Other`.

A standard campaign is a pre-fixed system campaign with standard pricing fees (Interchange or 3 Tier). Other is used for all the other types of campaigns where we do special conditions, pricing fees etc. The `processingMethod` determines the method of processing that a merchants wants to apply for.

Fields	Description	Type	Filterable	Required	Readonly
id	Unique ID of the campaign	Integer	Yes	No	Yes
name	Name of the campaign	String	Yes	No	Yes
type	`Standard` or `Other`	ENUM	Yes	No	Yes

processingMethod	`Acquiring`, `ACH` or `All`	ENUM	Yes	No	Yes
createdOn	Date and Time of creation	DateTime	Yes	No	Yes

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

GET List

https://kajadashboard.com/api/boarding-application/campaigns

Use this endpoint to list all Campaigns

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request List

curl

```
curl --location 'https://kajadashboard.com/api/boarding-application/campaigns'
```

Example Response

Body Headers (15) 200 OK

json


```
{
  "items": [
    {
      "id": "2",
      "name": "3 Tier",
      "type": "Standard",
      "processingMethod": "All",
      "createdOn": "2019-02-05 19:27:49"
    },
    {
```


View More

Equipment

Equipment

You can utilize this endpoint to perform several operations in the Equipment section of the Boarding Application

Equipment Object

Fields	Description	Type	Filterable	Required	Readonly
equipment	Equipment Info	Object	No	No	No
equipment.used	Equipment Used Object	Object	No	No	No
equipment.used.value	Equipment Used Value	String	No	No	No
equipment.used.locked	Equipment Used Locked Flag `Yes` or `No`	Enum	No	No	No
equipment.note	Equipment Note Object	Object	No	No	No
equipment.note.value	Equipment Note Value	String	No	No	No
equipment.note.locked	Equipment Note Locked Flag `Yes` or `No`	Enum	No	No	No

View More

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

Terminals

When creating a boarding application we give you the ability to add existing terminals.

By utilizing this endpoint you can perform several operations with the Existing terminals section of the Boarding Application.

> **IMPORTANT** You can only create, update, and delete Terminals information on pre-approved applications.

Equipment Terminal Object

Fields	Description	Type	Filterable	Required	Readonly
tid	TID	Integer	No	No	No
vnumber	V#	String	No	No	No
gateway	`Internal` or `Other`	ENUM	No	No	No
note	Note	String	No	No	No

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

GET List

https://kajadashboard.com/api/boarding-application/<id>/equipment/terminals

List all terminals that have been added to a Boarding Application. These terminals will be sent to the merchant when the application is approved.

```
curl --location 'https://kajadashboard.com/api/boarding-application/1/equipment/terminals' \
--header 'Content-Type: application/json'
```

Example Response

Body Headers (1) 200 OK

```
json
{
    "tid": 7000,
    "vnumber": "1",
    "gateway": "Internal",
    "note": "Gateway 1"
},
{
    "tid": 7001,
    "vnumber": "2",
    "gateway": "Other",
```

View More

GET View

https://kajadashboard.com/api/boarding-application/<id>/equipment/terminal/<tid>

View a specific Terminal in the Equipment section of the Boarding Application

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

This request is using Bearer Token from folder Boarding Applications

Example Request View

```
curl
curl --location 'https://kajadashboard.com/api/boarding-application/1/equipment/terminal/7000' \
--header 'Content-Type: application/json'
```

Example Response

Body Headers (1) 200 OK

```
json
{
  "tid": 7000,
  "vnumber": "1",
  "gateway": "Internal",
  "note": "Gateway 1"
}
```

GET View

https://kajadashboard.com/api/boarding-application/<id>/equipment

Use this endpoint to view the equipment settings on a Boarding Application.

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request View

```
curl
curl --location 'https://kajadashboard.com/api/boarding-application/1/equipment'
```

Example Response

Body Headers (1) 200 OK

```
json
{
  "equipment": {
    "used": {
      "value": "",
      "locked": "No"
    },
    "note": {
      "value": "",
      "locked": "No"
    },
```

View More

PUT Update

PUT Update

https://kajadashboard.com/api/boarding-application/<id>/equipment

Use this endpoint the update the equipment section of a Boarding Application

Just as in the dashboard, most of the fields are lockable.

Depending on permission you might not be able to update locked fields.

> **Important:** Attempting to update a locked field or a lock flag without the appropriate permissions will simply return the old values for these fields as they won't be updated.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

```
{
    "equipment":{
        "used":{
            "value":"Equipment Name"
        },
        "note":{
            "value":"note"
        },
        "providedBy":{
            "value":"Deployment Center"
```

View More

Example Request Update

```
curl --location --request PUT 'https://kajadashboard.com/api/boarding-application/1/equipment' \
--data '{
    "equipment": {
        "providedBy": {
            "value": "ISO"
        }
    },
    "ebt": {
        "locked": "No"
    },
```

View More

Example Response

Body Headers (1)

```
{
  "equipment": {
    "used": {
      "value": "Equipment Name",
      "locked": "No"
    },
    "note": {
      "value": "note",
      "locked": "No"
    },
```

View More

```
      "locked": "No"
    },
```

View More

List

Use this endpoint to list all Boarding Applications that you have available.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

GET List Boarding Applications

https://kajadashboard.com/api/boarding-application

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request List Boarding Applications

```
curl --location 'https://kajadashboard.com/api/boarding-application'
```

Example Response

Example Response

Body Headers (1) 200 OK

```
{
  "items": [
    {
      "id": 606,
      "processingMethod": "Acquiring",
      "company": {
        "name": null
      },
      "dba": {
        "name": null
```

View More

Create

An application needs to be created first before proceeding to updating with the necessary fields. We have 3 types of Boarding Applications based on the processing type that the merchant is applying for.

1. An Acquiring Boarding Application
2. An ACH Boarding Application
3. A hybrid, Acquiring and ACH Boarding Application.

> Keep in mind that the number of required fields are dependent on the `processingMethod`. For example if the Boarding Application is for Acquiring there is no need to send ACH section. Also for an Acquiring Boarding Application based on the `processing.sales` values we might require additional fields if the merchant has 70% or more `processing.sales.mail` + `processing.sales.internet`.

> When creating a boarding application via AppConnect token, you must explicitly provide an **agent id**.

> When creating a boarding application via AppConnect token, you must explicitly provide an **agent id**.

The Agent ID which you want to set on the boarding application must have authorized the AppConnect. Example:

```javascript
...
"agent": {
    "id": 654
}
```

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

POST Create Boarding Application

https://kajadashboard.com/api/boarding-application

Creating an Acquiring Boarding Application from a standard Campaign with `ID` 206 as an example.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

```json
{
    "campaign": {
        "id": 206
    },
    "processingMethod": "Acquiring"
}
```

Example Request Create Boarding Application

```curl
curl --location 'https://kajadashboard.com/api/boarding-application' \
--data '{
    "campaign": {
        "id": 206
    },
    "processingMethod": "Acquiring"
}'
```

Example Response

Body Headers (1) 200 OK

```json
{
  "id": 607,
  "processingMethod": "Acquiring",
  "mcc": null,
  "campaign": {
    "id": 206
  },
  "company": {
    "name": null,
    "type": null,
```

View More

Delete

You can use this endpoint to Delete a boarding application.

> **NOTE** This functionality is permission based, and in some case we might return an exception

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

DELETE Delete Boarding Application

https://kajadashboard.com/api/boarding-application/<id>

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request Delete Boarding Application

Example Request Delete Boarding Application

```curl
curl --location --request DELETE 'https://kajadashboard.com/api/boarding-application/599'
```

Example Response

Body Headers (1) 200 OK

```json
[]
```

View

You can either view the full Boarding Application or only a specific section.

> Keep in mind that based on the `processingMethod` the Processing/ACH section can be null.

The sections of the Boarding Application that are available to be viewed/updated separately are as follows:

- **Principals**
- **Processing**
- **ACH**
- **Documents**
- **Fees**
- **Documents**
- **Fees**

Each section and all endpoints associated with it are described in detail in our developers documentation

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

GET View Boarding Application

https://kajadashboard.com/api/boarding-application/<id>

You can utilize this endpoint for viewing the full Boarding Application

View Boarding Application with `id` 607

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request View Boarding Application

curl

```
curl --location 'https://kajadashboard.com/api/boarding-application/607'
```

Example Response

Example Response

Body Headers (1) 200 OK

json

```
{
  "id": 607,
  "processingMethod": "Acquiring",
  "mcc": null,
  "campaign": {
    "id": 206
  },
  "company": {
    "name": null,
    "type": null,
```

View More

Update

For testing purposes, please use our sandbox endpoint below.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

PUT Update Boarding Application

https://kajadashboard.com/api/boarding-application/608

PUT Update Boarding Application

https://kajadashboard.com/api/boarding-application/608

In the below example, we are updating only the `company.name` of the Boarding Application.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

json

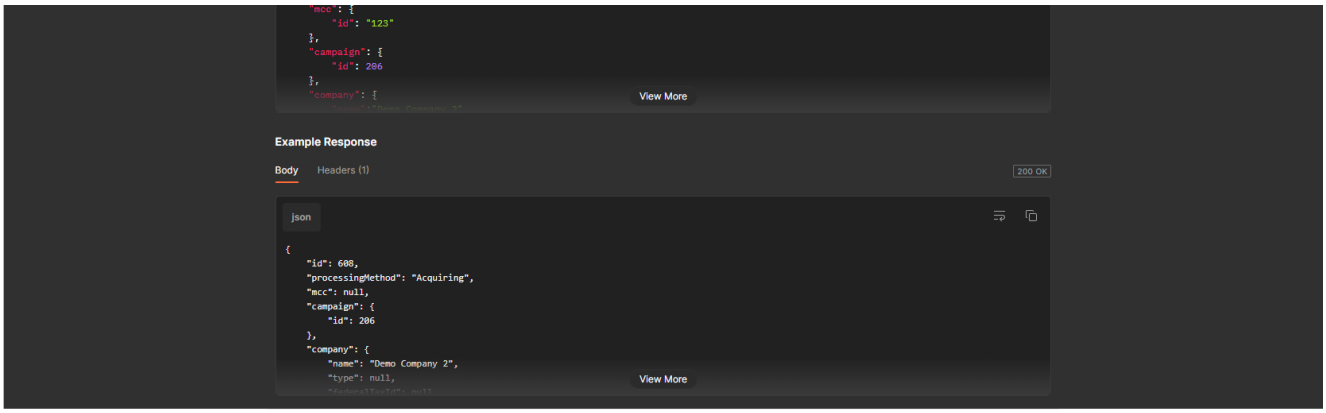
```
{
    "processingMethod": "Acquiring",
    "mcc": {
        "id": "123"
    },
    "campaign": {
        "id": 206
    },
    "company": {
        "name":"Demo Company 2"
```

View More

Example Request Update Boarding Applicati...

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/boarding-application/608' \
--data '{
    "processingMethod": "Acquiring",
```



Principals

You can use this endpoint to execute all necessary operations in the Principals section of the Boarding Application. You can do the following:

You can use this endpoint to execute all necessary operations in the Principals section of the Boarding Application. You can do the following:

- List all Principals
- Create an additional Principal
- Update Principal informaion
- Delete a Principal

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

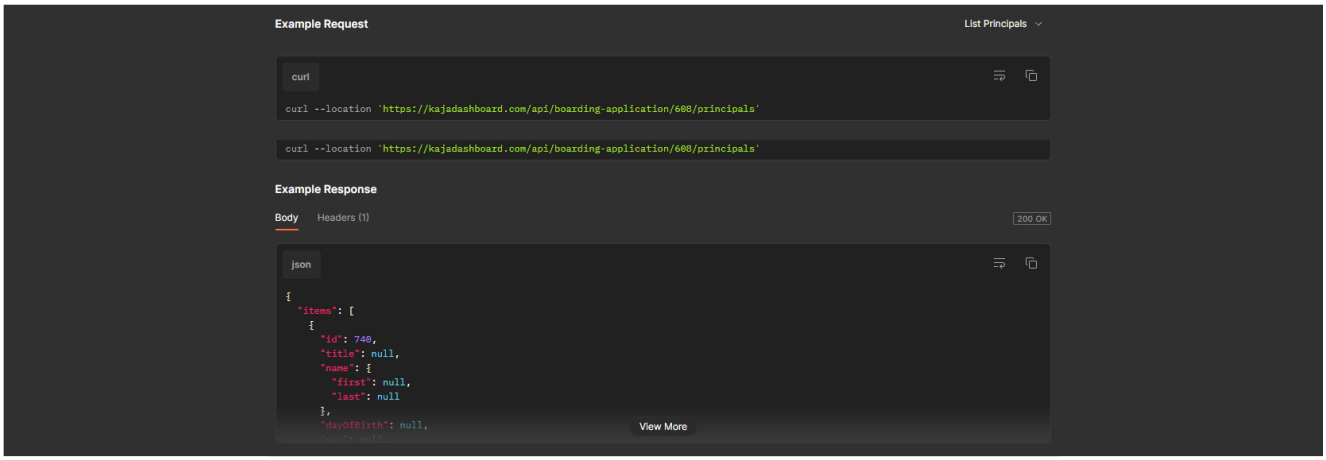
GET List Principals

https://kajadashboard.com/api/boarding-application/608/principals

List all Principals for Boarding Application 608.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications



POST Create Principal

https://kajadashboard.com/api/boarding-application/608/principal

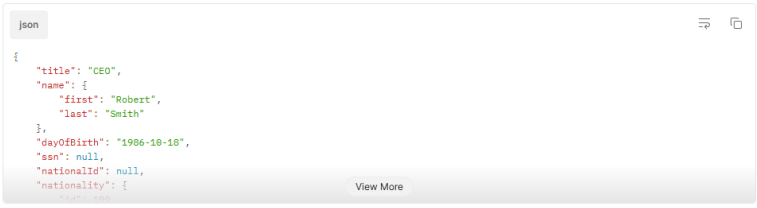
Add a second Principal to Boarding Application 608

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

Body raw (json)



```
curl --location 'https://kajadashboard.com/api/boarding-application/608/principal' \
--data '{
    "title": "CEO",
    "name": {
        "first": "Robert",
        "last": "Smith"
    },
    "dayOfBirth": "1986-10-18",
    "ssn": null,
    "nationalId": null,
```

View More

Example Response

Body Headers (1) 200 OK

json


```
{
  "id": 741,
  "title": "CEO",
  "name": {
    "first": "Robert",
    "last": "Smith"
  },
  "dayOfBirth": "1986-10-18",
  "ssn": null,
  "nationalId": null,
```


View More

PUT Update a Single Principal

https://kajadashboard.com/api/boarding-application/607/principal/741

You can utilize this endpoint to update information on a single principal in the Principals section of the Boarding Application.

Updating the `dayOfBirth` for Principal with `id` 741 for Boarding Application 607.

> **NOTE** Only send the parameters that you want to update, there is no need to send a full request with all the parameters for the Principal.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

Body raw (json)

json

```
{
    "dayOfBirth": "1986-10-21"
}
```

Example Request Update Principals

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/boarding-application/607/principal/741' \
--data '{
    "dayOfBirth": "1986-10-21"
}'
```

Example Response

Body Headers (1) 200 OK

json


```
[
  {
    "id": 741,
    "title": "CEO",
    "name": {
      "first": "Robert",
      "last": "Smith"
    },
    "dayOfBirth": "1986-10-21",
    "ssn": null,
```


View More

PUT Mass update Principals

https://kajadashboard.com/api/boarding-application/608/principals

You can utilize this endpoint to mass add, update or delete principal information.

> **IMPORTANT** Only send the parameters that you want to update for a specific principal, there is no need to send a full request with all the parameters for the Principal. However, if you have more than one principal on the Boarding Application you will need to provide all the Principal id's even if there are no changes to them or else those id's that are not provided in the request will be deleted.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

json

```
[
    {
        "id": 744,
        "dayOfBirth": "1986-10-21"
    }
]
```


```
    }
]
```


Example Request Update Principals

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/boarding-application/608/principals' \
--data '[
    {
        "id": 744,
        "dayOfBirth": "1986-10-21"
    },
    {
        "id": 745
    }
]'
```

Example Response

Body Headers (1) 200 OK

json

```
[
  {
    "id": 744,
    "title": "CEO",
    "name": {
      "first": "Robert",
      "last": "Smith"
    },
    "dayOfBirth": "1986-10-21",
    "ssn": null,
```

View More

DELETE Delete Principal

https://kajadashboard.com/api/boarding-application/607/principal/741

Delete Principal with `id` 741 for Boarding Application 607.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request Delete Principal

curl

```
curl --location --request DELETE 'https://kajadashboard.com/api/boarding-application/607/principal/741' \
--data ''
```

Example Response

Body Headers (1) 200 OK

json

json

```
true
```

Processing

You can use this endpoint to execute all necessary operations in the Processing section of the Boarding Application. You can do the following:

- View all the information in the Processing section
- Update information in the Processing section

`IMPORTANT: If you want to update multiple Bank Accounts read the Bank Accounts section.`

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

GET View Processing section

https://kajadashboard.com/api/boarding-application/607/processing

You can use this endpoint to view the necessary information in the Processing section of the Boarding Application.

Viewing only the Processing section of Boarding Application with `id` 607

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

This request is using Bearer Token from folder Boarding Applications

Example Request View Processing section

curl

```
curl --location 'https://kajadashboard.com/api/boarding-application/607/processing'
```

Example Response

Body Headers (1) 200 OK

json

```
{
  "bank": {
    "accountNumber": null,
    "routingNumber": null
  },
  "volumes": {
    "monthlyTransactionAmount": null,
    "avgTransactionAmount": null,
    "maxTransactionAmount": null
  },
```

View More

PUT Update Processing section

https://kajadashboard.com/api/boarding-application/607/processing

https://kajadashboard.com/api/boarding-application/607/processing

You can use this endpoint to update the necessary information in the Processing section of the Boarding Application.

Update the `bank.accountNumber` and `bank.routingNumber` in the Processing section of Boarding Application with `id` 607

> **NOTE** You do not need to send the all the parameters in the Processing section. You can send only the parameters that you are updating.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

```json
{
    "bank": {
        "accountNumber": "222222233",
        "routingNumber": "444444444"
    }
}
```

Example Request — Update Processing section

```curl
curl --location --request PUT 'https://kajadashboard.com/api/boarding-application/607/processing' \
--data '{
    "bank": {
        "accountNumber": "222222233",
        "routingNumber": "444444444"
    }
}'
```

Example Response

Body Headers (1) — 200 OK

```json
{
  "bank": {
    "accountNumber": "222222233",
    "routingNumber": "444444444"
  },
  "banks": [
    {
      "id": 357,
      "type": "All",
      "accountNumber": "222222233",
```

View More

Merchant Category Codes (MCCs)

In the finance ecosystem, Merchant Category Codes (also known as MCCs) are used to classify businesses by the type of products or services they provide.

For example, restaurants, hospitals and grocery stores have different MCCs. These MCC codes are used for calculating interchange fees, authorizing payments, and preventing fraud.

The MCC of sellers get configured during onboarding; it's important that your sellers have MCCs that match the services they're offering.

The below endpoint provides a list of MCCs numbers, their description and associated system identifier.
The MCC of sellers get configured during onboarding; it's important that your sellers have MCCs that match the services they're offering.

The below endpoint provides a list of MCCs numbers, their description and associated system identifier.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

GET List MCCs

https://kajadashboard.com/api/boarding-application/mcc

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request — List MCCs

```curl
curl --location 'https://kajadashboard.com/api/boarding-application/mcc'
```

Example Response

Body Headers (0)

Text

Text

```
{
    "items": [
        {
            "id": 1030,
            "number": "7994",
            "description": "Sweepstakes and social games"
        },
        {
            "id": 1027,
            "number": "7994",
```

View More

ACH

You can use this endpoint to execute all necessary operations in the ACH section of the Boarding Application. You can do the following:

- View all the information in the ACH section
- Update information in the ACH section

`IMPORTANT: If you want to update multiple Bank Accounts read the Bank Accounts section.`

AUTHORIZATION Bearer Token

This folder is using Bearer Token from folder Boarding Applications

GET View ACH section

https://kajadashboard.com/api/boarding-application/609/ach

GET View ACH section

https://kajadashboard.com/api/boarding-application/609/ach

You can use this endpoint to view the necessary information in the ACH section of the Boarding Application.

Viewing only the ACH section of Boarding Application with `id` 609

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Example Request View ACH section

curl

```
curl --location 'https://kajadashboard.com/api/boarding-application/609/ach'
```

Example Response

Body Headers (1) 200 OK

json


```
{
  "bank": {
    "accountNumber": null,
    "routingNumber": null
  },
  "volumes": {
    routingNumber : null
  },
  "volumes": {
    "monthlyTransactions": null,
    "monthlyTransactionAmount": null,
    "avgTransactionAmount": null,
    "maxTransactionAmount": null,
```


View More

PUT Update ACH section

https://kajadashboard.com/api/boarding-application/609/ach

You can use this endpoint to update the necessary information in the ACH section of the Boarding Application.

Update the `bank.accountNumber` and `bank.routingNumber` in the ACH section of Boarding Application with `id` 609

NOTE You do not need to send all the parameters in the Ach section. You can send only the parameters that you are updating.

AUTHORIZATION Bearer Token

This request is using Bearer Token from folder Boarding Applications

Body raw (json)

json

```
{
    "bank": {
        "accountNumber":"1122334455",
        "routingNumber":"112233440"
    }
}
```

Example Request Update ACH section

curl

```
curl --location --request PUT 'https://kajadashboard.com/api/boarding-application/609/ach' \
--data '{
    "bank": {
        "accountNumber":"1122334455",
        "routingNumber":"112233440"
```