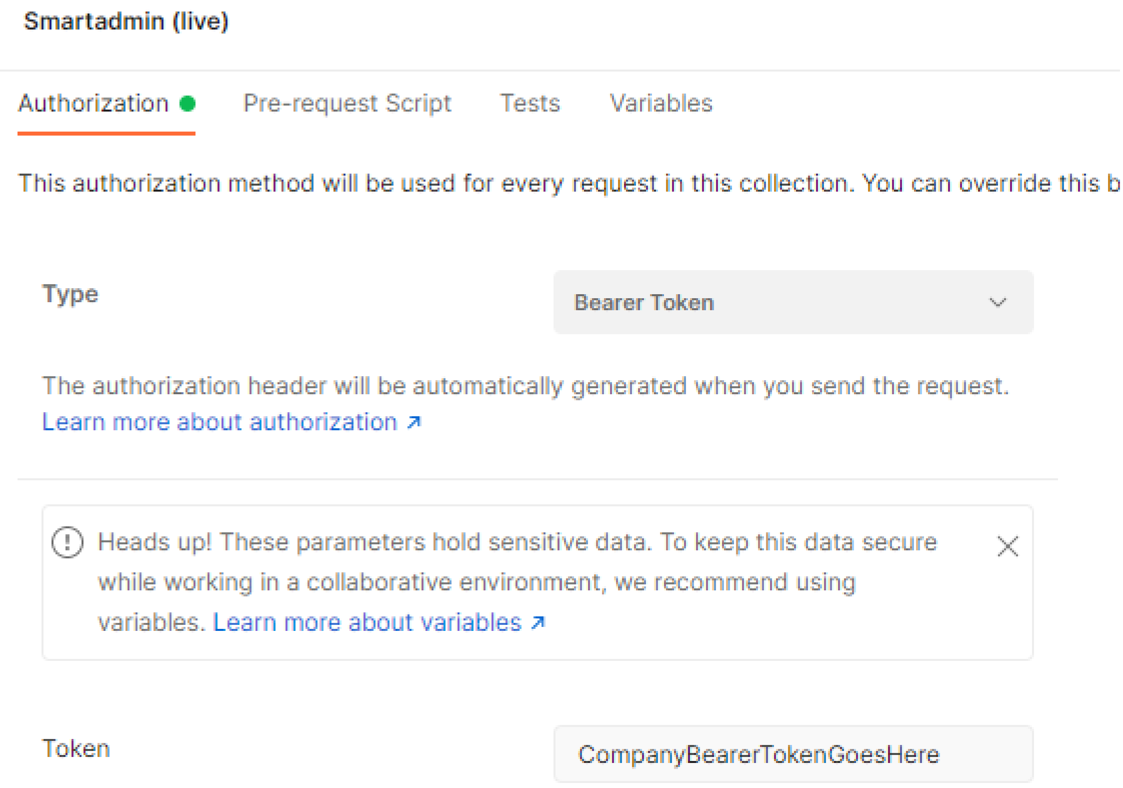
Neuro-Pay Checkout API

outlining the functionality and implementaiton of the paiwise checkout API

Authentification

In order to use the checkout API the company in question needs to generate an API key through the Neuro-Admin platform. Said API key should be sent as a bearer token in each request.



This authorization method will be used for every request in this collection. You can override this b

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Token: CompanyBearerTokenGoesHere

Data format

The API currently accepts and returns JSON data in the body

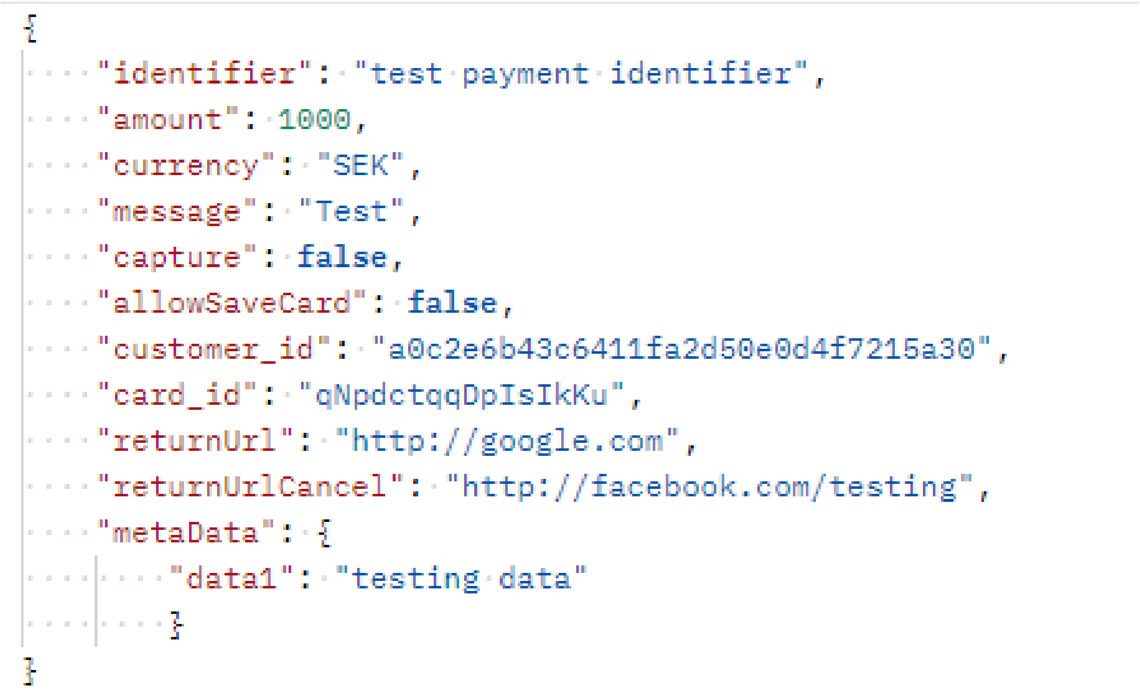
Requests

Collection of possible requests

Create [POST]

https://api.paiwise.com/checkout/create [POST]

identifier (optional)	String	Identifier of the checkout, if not passed the cid will get as reference
amount	int	Smallest nominee for the currency. Example: 30SEK is 3000 öre so value of this param should be 3000
currency	String	EUR, SEK, USD... etc
message	String	Message to be shown to customer in the checkout
capture (optional)	Boolean (default: true)	If capture is set to false, a manual capture has to be done later. In order to release the funds back to the client the Refund endpoint can be used.
allowSaveCard (optional)	Boolean (default: true)	If allowSaveCard is set to false, the user can't save the card in his card list for future usage.
returnUrl	String	URL where the customer should be returned to after checkout process is done.
returnUrlCancel	String	URL where the customer should be returned to if cancelling the checkout process
customer_id (optional)	String	If present activate the possibility to save the card during the payment
card_id (optional)	String	If present (customer_id required) do an automatic payment
template (optional)	String	For now only can be "iframe" which removes background image
metaData	Array	Key value pair array to store additional data together with the payment.



```
{
    "identifier": "test payment identifier",
    "amount": 1000,
    "currency": "SEK",
    "message": "Test",
    "capture": false,
    "allowSaveCard": false,
    "customer_id": "a0c2e6b43c6411fa2d50e0d4f7215a30",
    "card_id": "qNpdctqqDpIsIkKu",
    "returnUrl": "http://google.com",
    "returnUrlCancel": "http://facebook.com/testing",
    "metaData": {
        "data1": "testing data"
        }
}
```

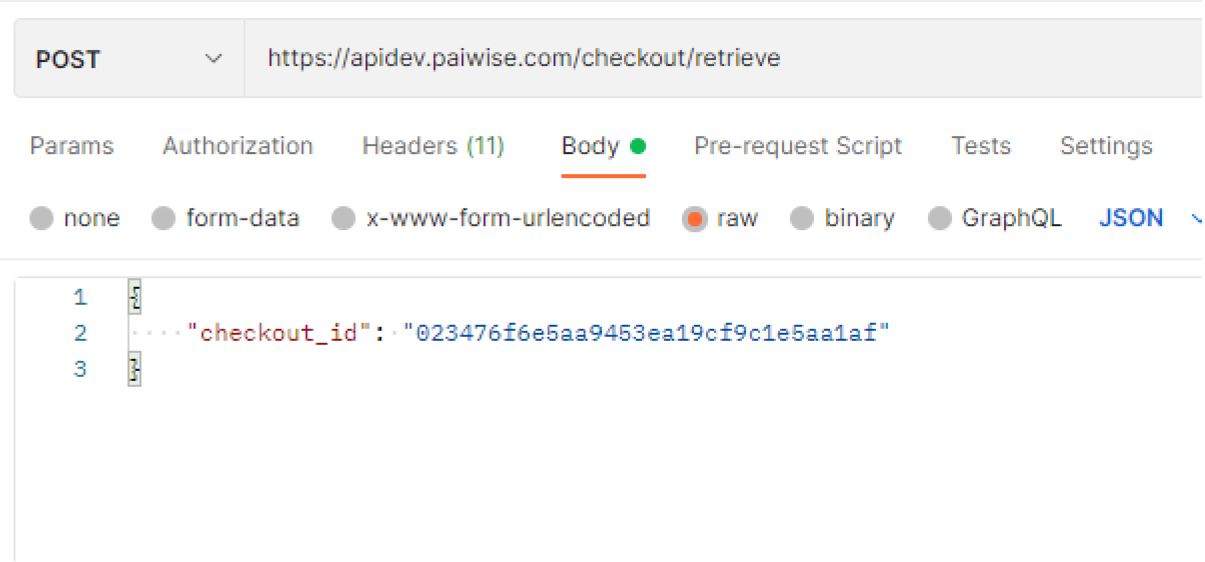
Requests

Retrieve [POST]

https://api.paiwise.com/checkout/retrieve [POST]

Values

checkout_id	String	The checkout_id received when creating a checkout object



POST https://apidev.paiwise.com/checkout/retrieve

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "checkout_id": "023476f6e5aa9453ea19cf9c1e5aa1af"
}
```

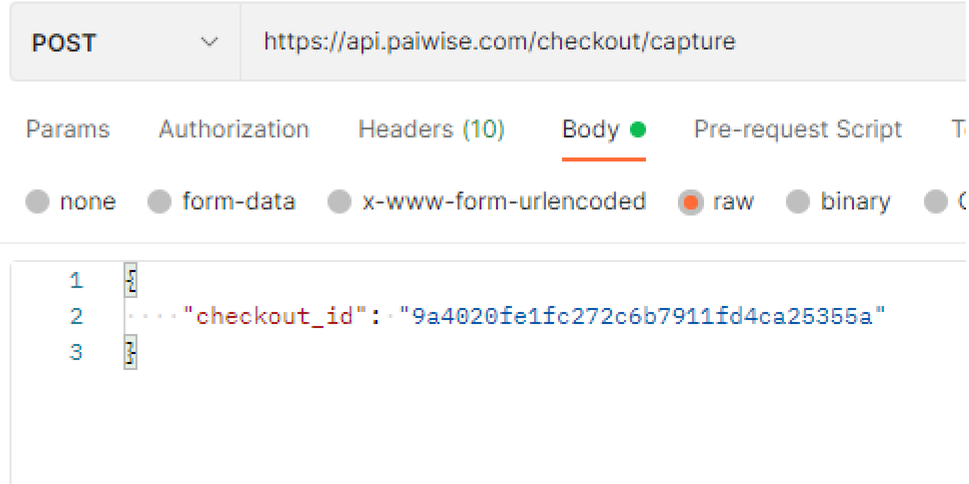
Capture [POST]

https://api.paiwise.com/checkout/capture [POST]

Capture is used if capture was set to false when creating a checkout object. And if status of the object is “reserved”

Values

checkout_id	String	The checkout_id received when creating a checkout object



POST https://api.paiwise.com/checkout/capture

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ (

```
{
    "checkout_id": "9a4020fe1fc272c6b7911fd4ca25355a"
}
```

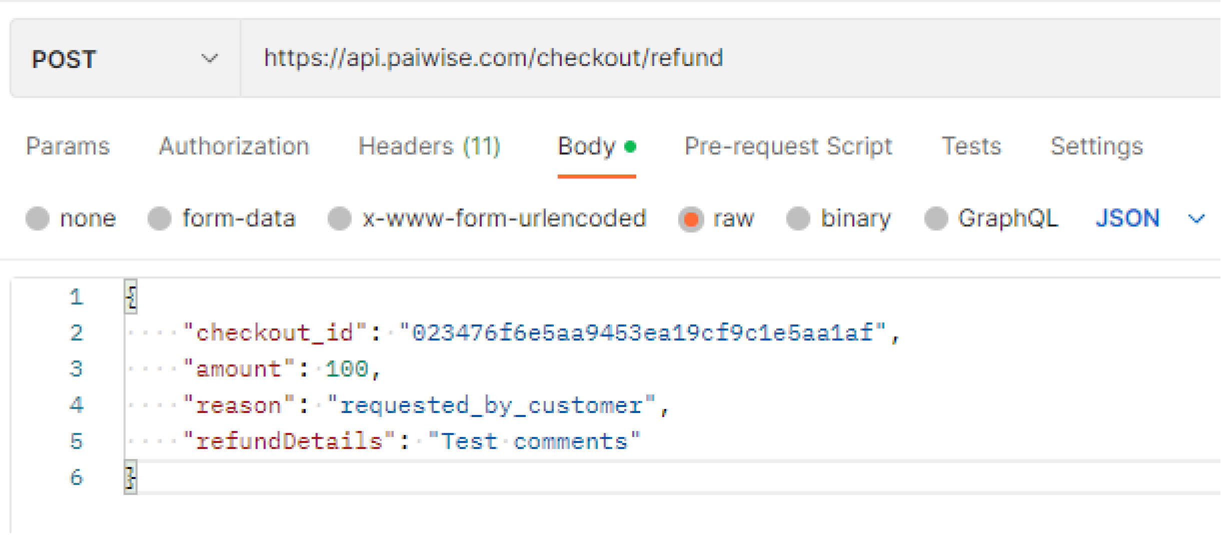
Refund [POST]

https://api.paiwise.com/checkout/refund [POST]

The refund can be used on objects that has status “paid” or “reserved” and will result in changing the status to either “refunded” or “reserve released”

Values

checkout_id	String	The checkout_id received when creating a checkout object
amount (optional)	Int	If an amount to be refunded is not set the whole amount will be refunded
reason (optional)	String	Reason of the refund. Value admitted: "duplicate", "requested_by_customer" or "fraudolent"
refundDetails (optional)	String	Comments about the refund. Free text



```
{
    "checkout_id": "023476f6e5aa9453ea19cf9c1e5aa1af",
    "amount": 100,
    "reason": "requested_by_customer",
    "refundDetails": "Test comments"
}
```

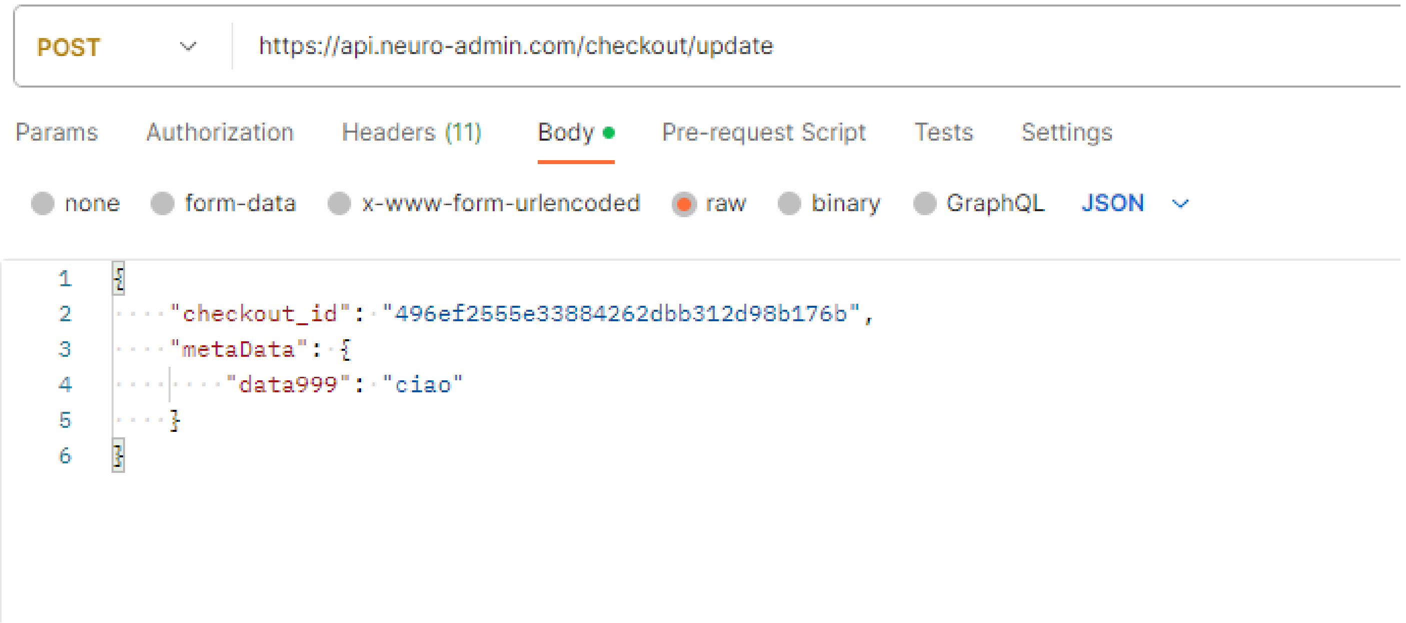
Update [POST]

https://api.paiwise.com/checkout/update[POST]

The update can be used on objects that has every status

Values

checkout_id	String	The checkout_id received when creating a checkout object
metaData	String	An object key-value (if value is empty, the relative key will be deleted from the object)



POST https://api.neuro-admin.com/checkout/update

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

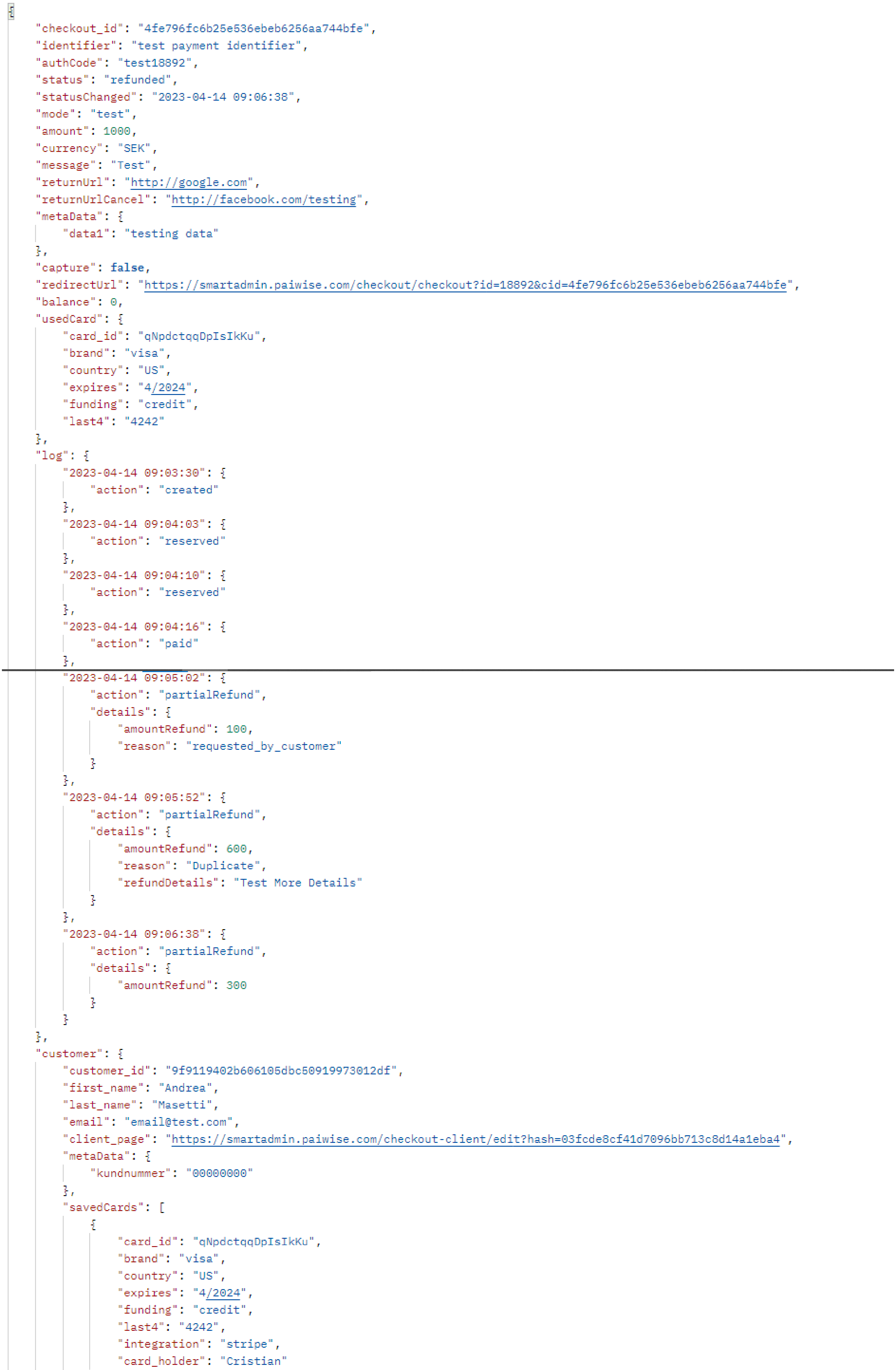
```
{
    "checkout_id": "496ef2555e33884262dbb312d98b176b",
    "metaData": {
        "data999": "ciao"
    }
}
```

Response

Response

The response object

For each endpoint the checkout object will be returned in its “new state”
Example:



```
{
    "checkout_id": "4fe796fc6b25e536ebeb6256aa744bfe",
    "identifier": "test payment identifier",
    "authCode": "test18892",
    "status": "refunded",
    "statusChanged": "2023-04-14 09:06:38",
    "mode": "test",
    "amount": 1000,
    "currency": "SEK",
    "message": "Test",
    "returnUrl": "http://google.com",
    "returnUrlCancel": "http://facebook.com/testing",
    "metaData": {
        "data1": "testing data"
    },
    "capture": false,
    "redirectUrl": "https://smartadmin.paiwise.com/checkout/checkout?id=18892&cid=4fe796fc6b25e536ebeb6256aa744bfe",
    "balance": 0,
    "usedCard": {
        "card_id": "qNpdctqqDpIsIkKu",
        "brand": "visa",
        "country": "US",
        "expires": "4/2024",
        "funding": "credit",
        "last4": "4242"
    },
    "log": {
        "2023-04-14 09:03:30": {
            "action": "created"
        },
        "2023-04-14 09:04:03": {
            "action": "reserved"
        },
        "2023-04-14 09:04:10": {
            "action": "reserved"
        },
        "2023-04-14 09:04:16": {
            "action": "paid"
        },
        "2023-04-14 09:05:02": {
            "action": "partialRefund",
            "details": {
                "amountRefund": 100,
                "reason": "requested_by_customer"
            }
        },
        "2023-04-14 09:05:52": {
            "action": "partialRefund",
            "details": {
                "amountRefund": 600,
                "reason": "Duplicate",
                "refundDetails": "Test More Details"
            }
        },
        "2023-04-14 09:06:38": {
            "action": "partialRefund",
            "details": {
                "amountRefund": 300
            }
        }
    },
    "customer": {
        "customer_id": "9f9119402b606105dbc50919973012df",
        "first_name": "Andrea",
        "last_name": "Masetti",
        "email": "email@test.com",
        "client_page": "https://smartadmin.paiwise.com/checkout-client/edit?hash=03fcde8cf41d7096bb713c8d14a1eba4",
        "metaData": {
            "kundnummer": "00000000"
        },
        "savedCards": [
            {
                "card_id": "qNpdctqqDpIsIkKu",
                "brand": "visa",
                "country": "US",
                "expires": "4/2024",
                "funding": "credit",
                "last4": "4242",
                "integration": "stripe",
                "card_holder": "Cristian"
```

Possible statuses are

- created
- reserved
- reserve released
- paid
- refunded
- partialRefund
- revoked
- failed
- canceled

Every date and time returned are in UTC.

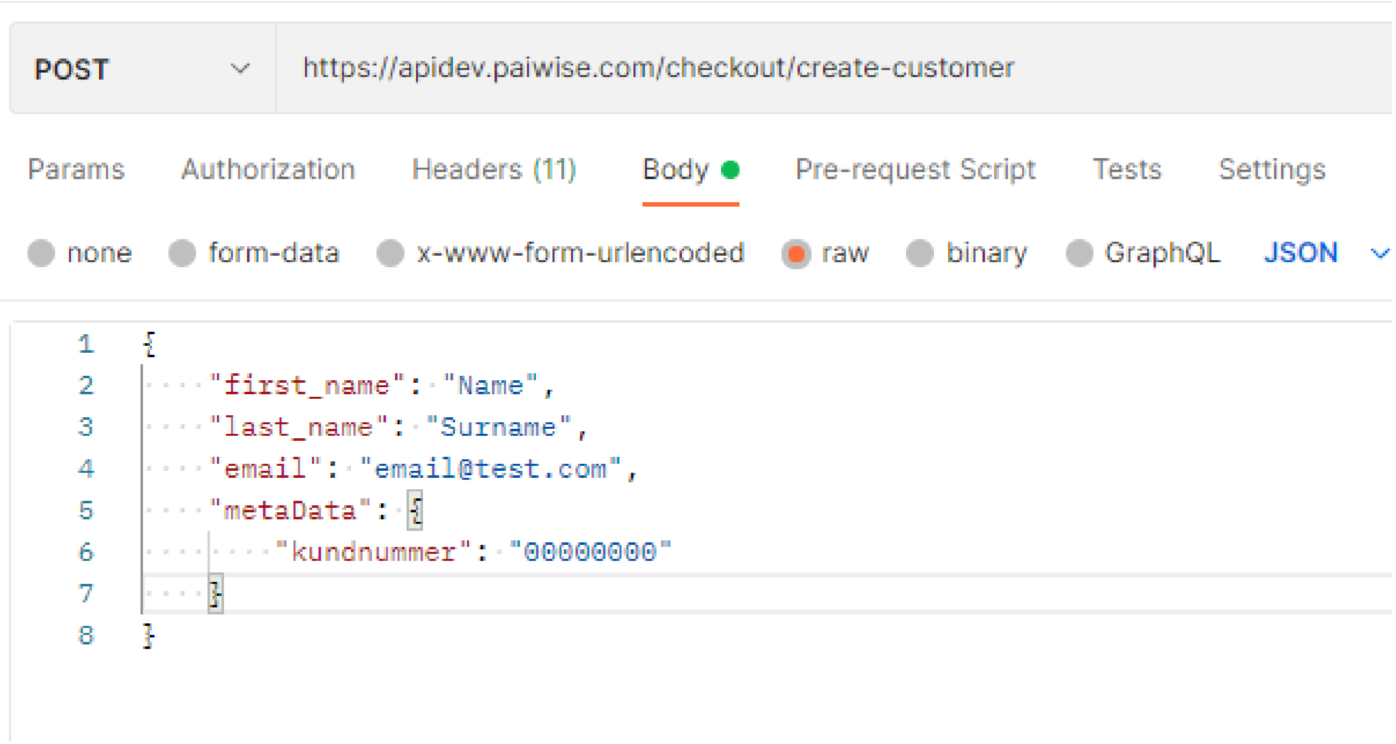
Customer's Request

Customer's Request

Create customer [POST]

https://api.paiwise.com/checkout/create-customer [POST]

first_name	String	Customer's last name
last_name	String	Customer's last name
email	String	Customer's email
metaData	Array	Key value pair array to store additional data together with the customer.



POST https://apidev.paiwise.com/checkout/create-customer

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

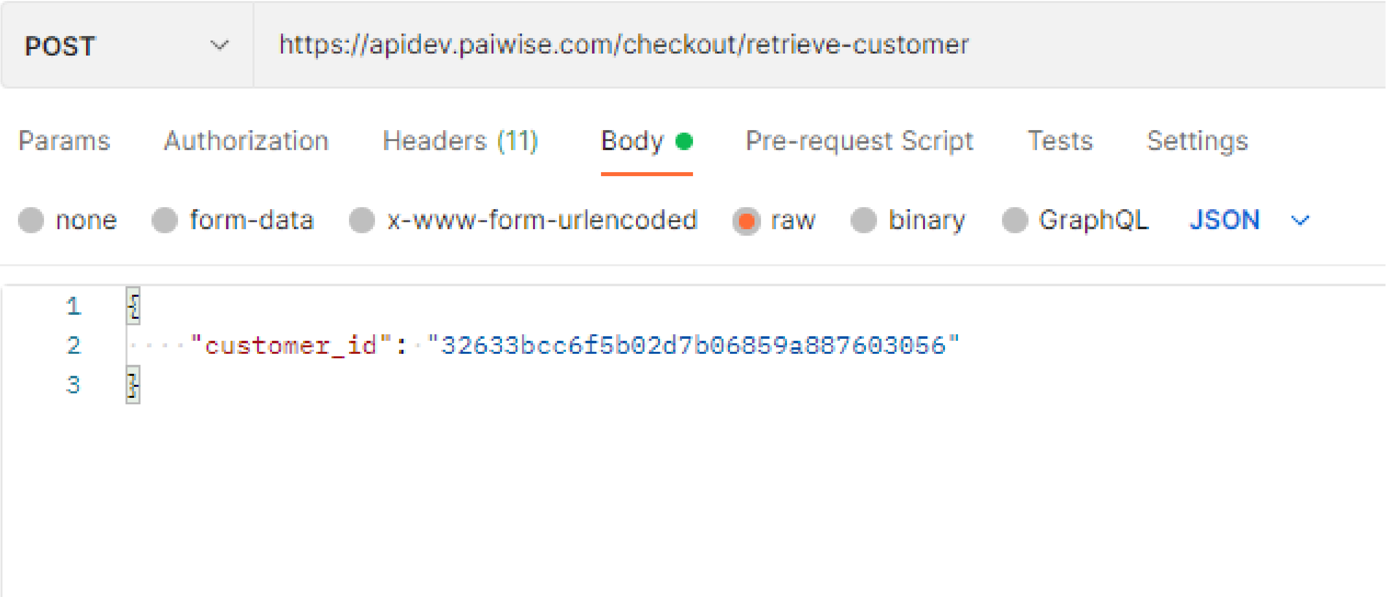
```
{
    "first_name": "Name",
    "last_name": "Surname",
    "email": "email@test.com",
    "metaData": {
        "kundnummer": "00000000"
    }
}
```

Customer's Request

Retrieve customer [POST]

https://api.paiwise.com/checkout/retrieve-customer [POST]

customer_id	String	Customer's id



POST https://apidev.paiwise.com/checkout/retrieve-customer

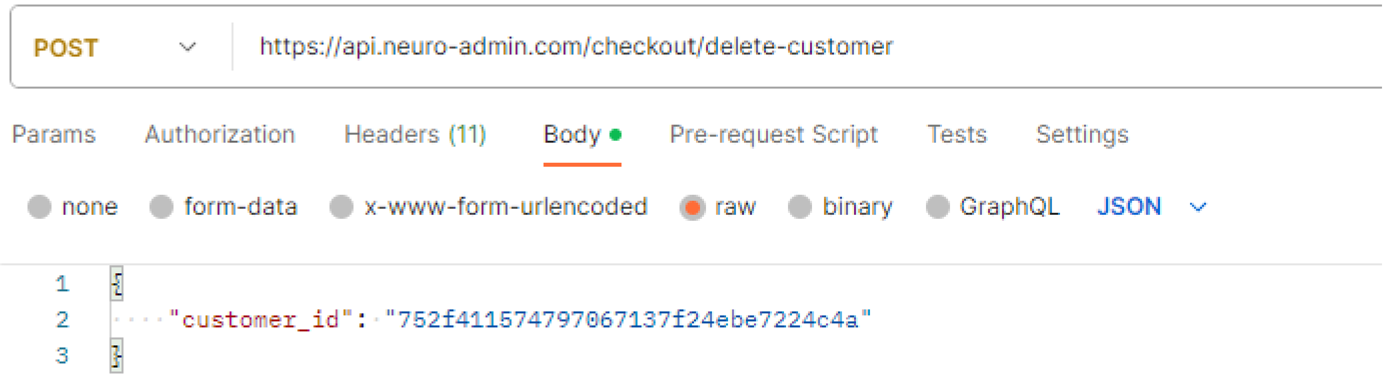
```
{
    "customer_id": "32633bcc6f5b02d7b06859a887603056"
}
```

Customer's Request

Delete customer [POST]

https://api.paiwise.com/checkout/delete-customer [POST]

customer_id	String	Customer's id



POST https://api.neuro-admin.com/checkout/delete-customer

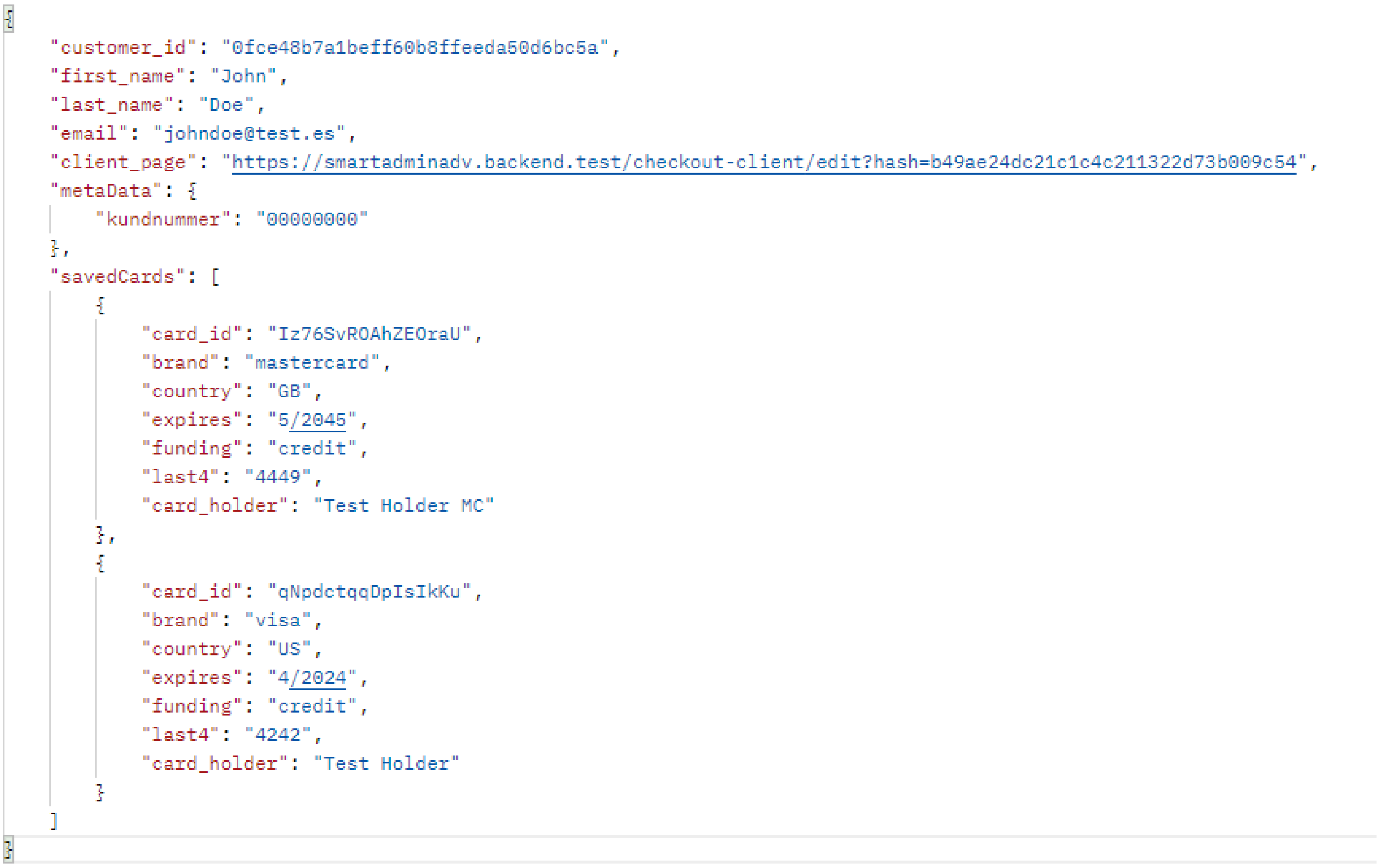
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "customer_id": "752f411574797067137f24ebe7224c4a"
}
```

Customer's Response

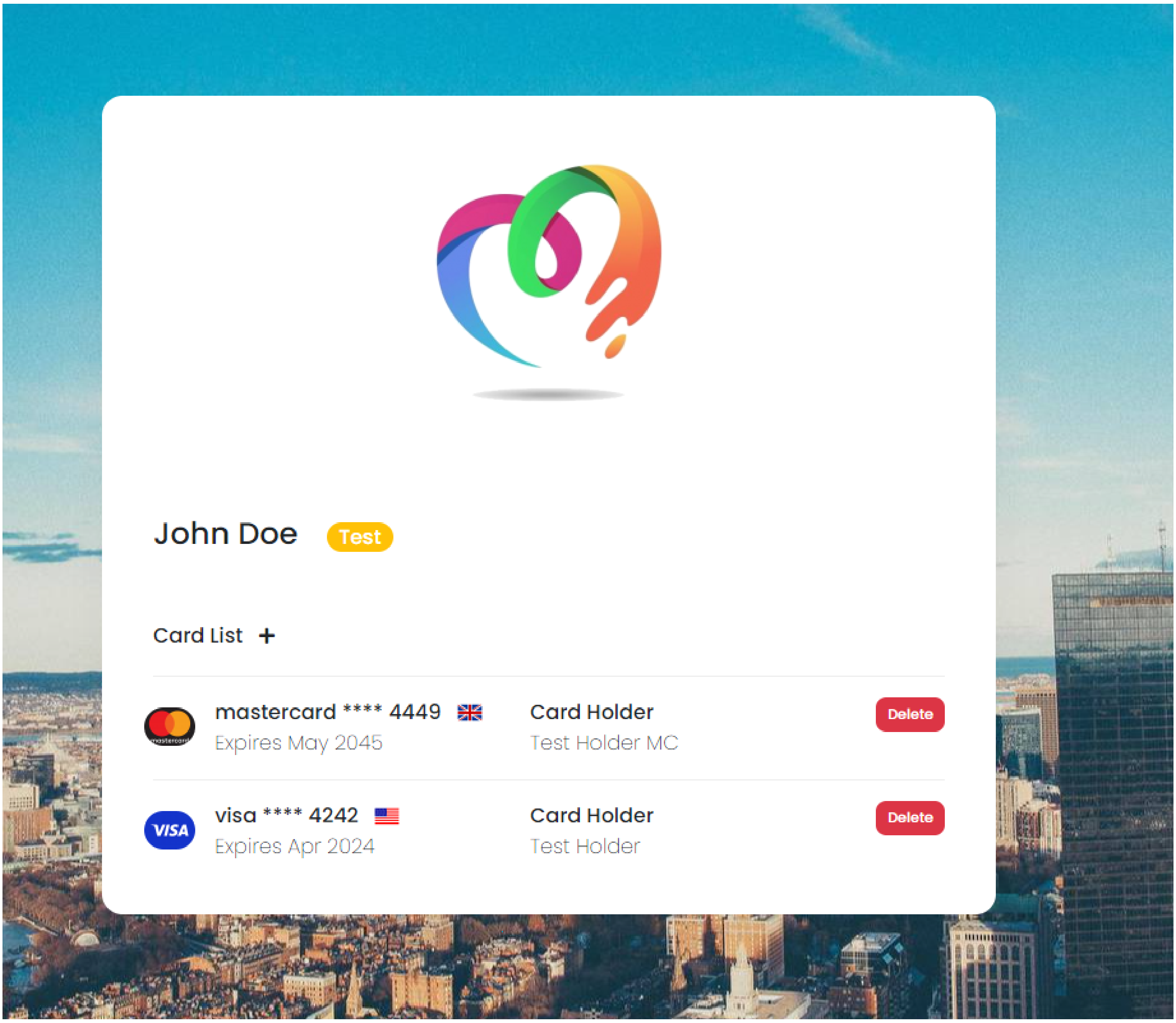
The response object

For each endpoint the customer object will be returned
Example:



```
{
    "customer_id": "0fce48b7a1beff60b8ffeeda50d6bc5a",
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoe@test.es",
    "client_page": "https://smartadminadv.backend.test/checkout-client/edit?hash=b49ae24dc21c1c4c211322d73b009c54",
    "metaData": {
        "kundnummer": "00000000"
    },
    "savedCards": [
        {
            "card_id": "Iz76SvROAhZEOraU",
            "brand": "mastercard",
            "country": "GB",
            "expires": "5/2045",
            "funding": "credit",
            "last4": "4449",
            "card_holder": "Test Holder MC"
        },
        {
            "card_id": "qNpdctqqDpIsIkKu",
            "brand": "visa",
            "country": "US",
            "expires": "4/2024",
            "funding": "credit",
            "last4": "4242",
            "card_holder": "Test Holder"
        }
    ]
}
```

The customer object also returns a "Client_page", following this link the customer can change the card configuration by adding/deleting and set the default a card for a future payment.



John Doe Test

Card List +

mastercard **** 4449
Expires May 2045

Card Holder
Test Holder MC

visa **** 4242
Expires Apr 2024

Card Holder
Test Holder

Postman collection

attached to this page is a postman collection for testing

All requests inherits the authentication from the collection. as the bearer token you add your API key test or live,

You can find file attachments in top left corner of this page