

Charge Reservation

The **Charge Reservation** endpoint enables you to initiate a merchant-initiated payment transaction **outside** of the originally reserved amount, using the customer's card tokenization data. In contrast to a *capture*, which merely converts a reserved authorization into a charge, a *charge* can be invoked independently — even after the reservation window — subject to certain limits and conditions.

Use this endpoint when:

- You wish to charge a customer after services have been delivered, or at a later time, beyond the original reservation window.
- You have a valid card token associated with the customer (resulting from tokenization during the reservation process) and want to debit their card directly.
- You want flexibility in timing: the charge may be performed up to **90 days** after reservation confirmation (or within a period specified by the merchant when creating the reservation). After this timeframe, the availability of the reserved funds can no longer be guaranteed.

Distinction from “Capture”

- A *capture* operation draws from the previously reserved authorization.
- A *charge*, however, is independent and can exceed (or be separate from) the reserved amount, as long as valid payment credentials exist.

You will find details about endpoint usage, authentication, request schema, validation rules, and standard responses below.

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/reservations/charge/{RESERVATION_UUID}
```

Authentication

Include a **Bearer Token** in the `Authorization` header. You can obtain this token from **Front Payment**.

Example:

Authorization: Bearer YOUR_FRONTPAYMENT_BEARER_TOKEN

Request Payload

Send the following parameters as a JSON object in the request body:

```
{
  "products": {
    "0": {
      "name": "Charge QA",
      "productId": null,
      "rate": 150,
      "tax": "0",
      "amount": 150
    }
  },
  "grandTotal": 150,
  "additionalText" : "My additional Text for capture"
}
```

Validation Rules

Make sure your request meets the following requirements:

Field	Type	Description
products.*.name	string	Required Name of the product.
products.*.productId	string	Optional Unique identifier for the product.
products.*.rate	numeric	Required Rate per unit of the product.
products.*.tax	numeric	Required Tax rate must be (e.g., 0, 12, 15, 25), Unless you have other configuration.
products.*.amount	numeric	Required Total amount for the product line item.
grandTotal	numeric	**Required** Grand total of the captured amount.
additionalText	string	**Optional** Captured note.

Response

A successful request will return a `202 OK` status with the following JSON payload:

```
{
  "status_code": 202,
  "status_message": "OK",
  "message": "reservationChargedSuccessfully",
  "is_data": true,
  "data": {
    "uuid": "String"
  }
}
```

API returns a `404` error, it means requested order with `RESERVATION_UUID` could not be found in our system.

```
{
  "status_code": 404,
  "status_message": "Not Found",
  "message": "reservationNotFound",
  "is_data": false,
  "data": null
}
```

API return a `417` error, it means request payload validation failed.

```
{
  "status_code": 417,
  "status_message": "Client Error",
  "message": "payloadValidationErrors",
  "is_error": true,
  "errors": "Array"
}
```

API returns a `510` error, it means something failed on the server side

```
{
  "status_code": 510,
  "status_message": "Execution Exception Occurred",
}
```

```
"message": "Something Went Wrong",
"is_error": true,
"errors": "Array"
}
```

Other Rejection Errors

```
{
  "status_code": 404,
  "status_message": "Not Found",
  "message": "paymentCardNotFound",
  "is_data": false,
  "data": null
}
```

```
{
  "status_code": 400,
  "status_message": "Conflict of Business Logic",
  "message": "paymentChargeRunwayExceed",
  "is_data": false,
  "data": null
}
```

```
{
  "status_code": 400,
  "status_message": "Conflict of Business Logic",
  "message": "paymentChargeDeadlineExceed",
  "is_data": false,
  "data": null
}
```

Revision #14

Created 3 January 2024 01:36:56 by Admin

Updated 1 October 2025 09:49:04 by Nayamot Ullah