

Capture Reservation

The **Capture Reservation** endpoint is used to convert a previously reserved (authorized) amount into an actual charge — either in full or partially — by referencing the reservation's unique identifier (UUID). Capturing is typically performed when goods or services are delivered, ensuring that the funds are secured from the customer's account.

Key behaviors

- **Full or Partial Capture:** You may capture the entire reserved amount, or you may choose to only capture part of it.
- **Automatic Release:** If only part of the reserved amount is captured, the remainder is automatically released back to the customer's account within 1-3 days.
- **Reservation Validity Window:** The original reservation remains valid for a limited period (often between 7 and 31 days, depending on your business rules). After that, the reservation may expire and no longer be capturable.

Use this endpoint when you're ready to finalize payment for what was reserved (or portions thereof), once delivery or service fulfillment is confirmed.

You will find details about the method, authentication, request fields, validation rules, and possible responses below.

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/reservations/capture/{{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": {
      "id": 298,
      "amount": 5
    },
    "1": {
      "id": 299,
      "amount": 50
    }
  },
  "grandTotal": 55,
  "additionalText" : "My additional Text for capture"
}
```

Retrieve Product ID from [Get Reservation API](#)

Validation Rules

Make sure your request meets the following requirements:

Field	Type	Description
<code>products.*.id</code>	numeric	**Required** Reservation product id. From which product you want to capture
<code>products.*.amount</code>	numeric	**Required** Captured amount for the product
<code>grandTotal</code>	numeric	**Required** Grand total of the captured amount.
<code>additionalText</code>	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": "reservationCapturedSuccessfully",
}
```

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

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

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

Revision #14

Created 3 January 2024 01:31:25 by Admin

Updated 1 October 2025 09:48:34 by Nayamot Ullah