

# Submit Reservation

The **Submit Reservation API** allows merchants to create a reservation for a customer and generate a payment link to be sent to the customer. Once created, the customer can complete payment with Visa and Mastercard debit or credit cards.

This API is designed for seamless integration into booking, order management, and e-commerce workflows where you want to lock in an order before payment is finalized.

## Typical Flow

1. Submit a reservation request with customer and order details.
2. Send payment link from FrontGO with order and merchant details.
3. Redirect the customer to the provided checkout URL.
4. Receive status updates (e.g., `Reserved`, `Captured`, `Charged`) via your callback endpoint.

---

## Prerequisites

Before integrating the **Submit Reservation API**, ensure you have the following:

- **API Access:** A valid merchant account with Front Payment and access to the API environment (Demo or Production).
- **Authentication:** A Bearer token issued by Front Payment, included in the `Authorization` header of every request.
- **Callback Endpoint:** A publicly accessible HTTPS endpoint to receive real-time payment status notifications (recommended).
- **Basic Setup:**
  - Ability to send HTTPS `POST` requests with JSON payloads.
  - Correct handling of numeric totals and validation rules (e.g., customer type, tax fields, 11-digit personal numbers).
  - Server time synchronized to ensure accurate timestamps for order and payment link expiry.
- **Testing Environment :** Access to the demo API from the following URL to validate your integration end to end before going live.

```
https://demo-api.frontpayment.no
```

# Reservation Lifecycle

Once a reservation is successfully created and its status is updated to **Reserved**, you have several options for handling the reserved funds:

- **Capture:** Capture the reserved amount either in full or partially. Capturing is typically used to secure payment when goods or services are delivered. The reservation period varies depending on your business type but generally lasts between **7 and 31 days**. [See Capture API Reference](#)
- **Partial Capture and Release:** If you capture only part of the reserved amount, the remaining balance is automatically released back to the customer within **1-3 days**.
- **Charge:** It is also possible to charge the customer up to **90 days** after the reservation has been confirmed. After this period, there is **no guarantee** that the reserved funds will still be available for charging. [See Charge API Reference](#)

---

These rules provide flexibility in aligning payment capture with your operational processes, while ensuring that customers are not left with indefinite reservations on their accounts.

## Submit Reservation

### Endpoint

```
POST [https://demo-api.frontpayment.no/api/v1/connect/reservations/submit](https://demo-api.frontpayment.no/api/v1/connect/reservations/submit)
```

### 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

```
{  
  "customerDetails": {
```

```
"uuid": "",
"type": "private",
"countryCode": "+880",
"msisdn": "1724313009",
"email": "nafees@yopmail.com",
"name": "Nafees",
"preferredLanguage": "en",
"personalNumber": "19635464524",
"organizationId": null,
"address": {
  "street": "Dhaka",
  "zip": "3500",
  "city": "Cumilla",
  "country": "NO"
},
"orderDate": "1756871295",
"dueDateForPaymentLink": "1756871295",
"referenceNo": null,
"customerReference": null,
"sendOrderBy": {
  "sms": false,
  "email": true
},
"products": {
  "0": {
    "name": "Test",
    "productId": null,
    "rate": 1,
    "tax": "0",
    "amount": 1
  }
},
"orderSummary": {
  "subTotal": 1,
  "totalTax": 0,
  "grandTotal": 1
},
"chargeValidity": "55",
"customerNotes": null,
```

```

"tnc": null,
"settings": {
  "secureDetails": false,
  "isChargePartiallyRefundable": false
},
"callback": {
  "callbackUrl": "https://wp.frontpayment.no"
}
}

```

## Validation Rules

Field	Type	Description
customerDetails	array	<b>Required.</b> Customer details information.
customerDetails.type	string	<b>Required.</b> Customer type (private or corporate).
customerDetails.countryCode	string	<b>Required.</b> Country dialing code (e.g., +880).
customerDetails.msisdn	string	<b>Required.</b> Mobile Subscriber ISDN Number (phone number).
customerDetails.email	email	<b>Required.</b> Valid customer email address.
customerDetails.name	string	<b>Required.</b> Full name of the customer.
customerDetails.preferredLanguage	string	<b>Required.</b> Preferred language (e.g., en, no).
customerDetails.personalNumber	string	<b>Optional.</b> Customer's personal identification number, must be 11 characters.
customerDetails.organizationId	string	<b>Required</b> if <code>customerDetails.type</code> is <code>corporate</code> . Must be number
customerDetails.address	array	<b>Required.</b> Customer address details.
customerDetails.address.street	string	<b>Required.</b> Street address of the customer.
customerDetails.address.zip	string	<b>Required.</b> Zip code of the customer's address.
customerDetails.address.city	string	<b>Required.</b> City of the customer's address..
customerDetails.address.country	string	<b>Required.</b> ISO Alpha-2 country code (e.g., "NO"). Custom validation <code>IsoAlpha2Country</code> applies.
orderDate	string	<b>Required.</b> Unix timestamp for the Date of the order.
dueDateForPaymentLink	string	<b>Required.</b> Unix timestamp for payment link expiry. Must be current or future timestamp.
referenceNo	string	<b>Optional.</b> Internal reference number.
customerReference	string	<b>Optional.</b> Customer's reference number.

Field	Type	Description
sendOrderBy	array	<b>Required.</b> Defines communication channels (SMS, email).
sendOrderBy.sms	boolean	<b>Required.</b> Whether to send via SMS.
sendOrderBy.email	boolean	<b>Required.</b> Whether to send via Email.
products	array	<b>Required.</b> List of products in the order.
products.*.name	string	<b>Required.</b> Name of the product.
products.*.productId	string	<b>Optional.</b> Unique identifier for the product.
products.*.rate	numeric	<b>Required.</b> Rate per unit of the product.
products.*.tax	numeric	<b>Required.</b> Tax rate (e.g., 0, 12, 15, 25). Unless you have other configuration.
products.*.amount	numeric	<b>Required.</b> Quantity of product.
orderSummary.subTotal	numeric	<b>Required.</b> Subtotal of products.
orderSummary.totalTax	numeric	<b>Required.</b> Total tax amount.
orderSummary.grandTotal	numeric	<b>Required.</b> Final payable amount.
customerNotes	string	<b>Optional.</b> Customer-provided notes.
tnc	string	<b>Optional.</b> Terms & Conditions.
chargeValidity	string	<b>Optional.</b> Must be numeric string.
settings	array	<b>Optional.</b> Additional settings.
settings.secureDetails	boolean	<b>Optional.</b> If <code>secureDetails</code> is set to <code>True</code> , the reservation details will not be accessible until the user's PNumber has been verified through BankID.
settings.isChargePartiallyRefundable	boolean	<b>Optional.</b> Whether partial refunds are allowed.
callback	array	<b>Optional.</b> Callback configuration.
callback.callbackUrl	url	<b>Optional.</b> If the value is true charge will be party refundable, otherwise it will be fully refundable.

## Example Success Response

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "Reservation Submitted Successfully",
  "is_data": true,
  "data": {
    "customerUuid": "CSRT1358046874",
    "reservationUuid": "RES1943140879",
```

```
"checkoutUrl": "https://stg.frontpayment.no/reservations/details/RES1943140879"
}
}
```

## Error Response

401 Unauthorized: Missing or invalid Bearer token. Other validation errors will return appropriate HTTP error codes (e.g., 400 Bad Request) along with error messages specifying the invalid or missing fields.

```
{
  "status_code": 500,
  "status_message": "Internal Dependency Error",
  "message": "internalErrorOccurredPleaseTryAgainLater",
  "is_error": true,
  "errors": {
    "happenedAt": "String",
    "internalErrorDetails": "Array"
  }
}
```

```
{
  "status_code": 510,
  "status_message": "Internal Dependency Error",
  "message": "Payment Gateway Error(Submit Payment): Failed to Create Payment Link",
  "is_error": true,
  "errors": {
    "Payment Gateway Error(Submit Payment): Failed to Create Payment Link"
  }
}
```

## Redirect to Checkout

After receiving a successful response, the customer should be redirected to the `checkoutUrl` provided. At checkout, the customer can choose from supported payment methods:

- **Visa**
- **Mastercard**

Once the payment is completed successfully, the user will be redirected to a success page.

---

# Callback Notification

The `callbackUrl` is an endpoint on your server that our system will call via an `HTTP GET` request whenever the status of the specified order changes from its initial state. For `reservation` the `callbackUrl` will be triggered for three status changed: `Reserved`, `Captured`, `Charged`.

See the link below to understand how to work with the callback URL on your side and how to verify the request sent from our side.

[Go To `Notification Via Callback Url` Page](#)

---

Revision #41

Created 18 September 2023 10:26:32 by Admin

Updated 30 September 2025 12:35:23 by Nayamot Ullah