

# Checkout Order

- [Create Checkout Session - Card, Vipps, Apple & Google Pay](#)
- [Create Session for Invoice Order](#)
- [Hosted Checkout](#)

# Create Checkout Session - Card, Vipps, Apple & Google Pay

The Create Checkout Session API enables merchants to generate secure, one-time checkout sessions for customers. This ensures a quick and PCI-compliant payment process without requiring the customer to create an account or save a payment method.

## Key use cases

- **E-commerce:** Generate links for one-off product sales.
- **Services:** Request upfront payments (consulting, events, classes).

## Prerequisites

Before you start the integration, make sure you have:

### 1. API Access:

- A valid API key and Bearer Token from Front Payment
- Access to the demo and production environments

### 2. Merchant Setup:

- Your merchant account configured with Front Payment
- Enabled payment methods (Vipps, Google Pay, Apple Pay, Visa, Mastercard)

### 3. Technical Requirements:

- Ability to make HTTPS API calls
- Secure storage of tokens and keys
- Callback endpoints to handle payment status updates

### 4. Test Environment:

- For testing, contact [nafees.faraz@frontpayment.no](mailto:nafees.faraz@frontpayment.no) to gain access to the demo environment

# Step 1: Create Payment URL

To initiate a payment, your system will need to call our create endpoint to generate a payment URL. This URL will redirect your users to payment gateway.

## Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/orders/regular/submit
```

## Authorization

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

The request body should be a JSON object containing details about the order, customer, and callback URLs.

```
{
  "products": [
    {
      "name": "Test Product",
      "productId": "1234",
      "quantity": 1,
      "rate": 4500,
      "discount": 0,
      "tax": 12,
      "amount": 4500
    }
  ],
  "orderSummary": {
    "subTotal": 4017.86,
    "totalTax": 482.14,
    "totalDiscount": 0.00,
    "grandTotal": 4500.00,
  }
}
```

```
"shippingCost": 0.00
},
"referenceNo": "",
"customerReference": "",
"orderDate": "1754556624",
"withCustomer": true,
"customerDetails": {
  "type": "private",
  "countryCode": "+47",
  "msisdn": "46567468",
  "email": "kari@nordmann.no",
  "name": "Kari Nordmann",
  "preferredLanguage": "en",
  "personalNumber": null,
  "organizationId": null,
  "address": {
    "street": "Luramyrvеien 65",
    "zip": "4313",
    "city": "Sandnes",
    "country": "NO"
  }
},
"submitPayment": {
  "via": "visa"
},
"callback": {
  "callbackUrl": "https://your-callback-url.com/callback",
  "success": "https://your-callback-url.com/success",
  "failure": "https://your-callback-url.com/failure"
}
}
```

## Validation Rules

Ensure your payload adheres to the following validation rules:

Field	Type	Description
<code>products.*.name</code>	string	<b>Required.</b> The name of the product.

Field	Type	Description
<code>products.*.productId</code>	string	<b>Optional.</b> The unique ID of the product.
<code>products.*.quantity</code>	numeric	<b>Required.</b> Quantity of the product.
<code>products.*.rate</code>	numeric	<b>Required.</b> Rate per unit of the product.
<code>products.*.discount</code>	numeric	<b>Optional.</b> Discount applied to the product.
<code>products.*.tax</code>	numeric	<b>Required.</b> Tax rate must be (e.g., 0, 12, 15, 25), Unless you have other configuration unless otherwise configured.
<code>products.*.amount</code>	numeric	<b>Required.</b> Total amount for the product line item.
<code>orderSummary.subTotal</code>	numeric	<b>Required.</b> Subtotal of all products before tax and discount.
<code>orderSummary.totalTax</code>	numeric	<b>Required.</b> The total tax for the order.
<code>orderSummary.totalDiscount</code>	numeric	<b>Required.</b> Total discount for the order.
<code>orderSummary.grandTotal</code>	numeric	<b>Required.</b> Grand total of the order.
<code>orderSummary.shippingCost</code>	numeric	<b>Optional.</b> Shipping cost of order.
<code>orderDate</code>	string	<b>Required.</b> Unix timestamp for the Date of the order, which must be current or future date.
<code>referenceNo</code>	string	<b>Optional.</b> Any reference information from your side. example: Order Uuid generated from your application.
<code>customerReference</code>	string	<b>Optional.</b> Customer reference
<code>orderFrom</code>	string	<b>Conditionally Required</b> if <code>fpgoUuid</code> is present. If provided, the value must be <code>PARTNER</code> . This indicates that the request originates from a registered partner and is intended to update an existing record.
<code>fpgoUuid</code>	string	<b>Optional</b> Use this to <b>prevent duplicates</b> . Pass the <code>orderUuid</code> from a previous response to update that specific order. If omitted, a new order is created.
<code>withCustomer</code>	boolean	<b>Required.</b> If <b>withCustomer</b> is true then you must provide customer details

Field	Type	Description
customerDetails.type	string	The customer type. <b>Required</b> if <b>withCustomer</b> is <i>true</i> . Must be either <code>`private`</code> or <code>`corporate`</code> .
customerDetails.countryCode	string	Country code for the customer's phone number (e.g., "+47"). <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
customerDetails.msisdn	string	Mobile Subscriber MSISDN Number (phone number). <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
customerDetails.email	string	Customer's email address. <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
customerDetails.name	string	Customer's full name. <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
customerDetails.preferredLanguage	string	<b>Optional</b> . Customer preferred language. Available languages are <code>en</code> , <code>no</code> , <code>sv</code> , <code>da</code> , <code>de</code> . If nothing is given it will set default to <code>no</code> .
customerDetails.personalNumber	string	<b>Optional</b> . Customer's personal identification number, must be 11 characters.
customerDetails.organizationId	numeric	<b>Required</b> if customer type is <i>corporate</i> . Must be alphanumeric.
customerDetails.address.street	string	Street address of the customer. <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
customerDetails.address.zip	string	Zip code of the customer's address. <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
customerDetails.address.city	string	City of the customer's address. <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
customerDetails.address.country	string	ISO Alpha-2 country code (e.g., "NO"). Custom validation <code>IsoAlpha2Country</code> applies. <b>Required</b> if <b>withCustomer</b> is <i>true</i> .
submitPayment.via	string	<b>Required</b> . The payment method. Available payment methods <code>vipps</code> , <code>visa</code> , <code>mastercard</code> , <code>applepay</code> , or <code>googlepay</code> .
callback.callbackUrl	url	<b>Required</b> . The URL to which Front Payment will send updates. Must be a valid url.
callback.success	url	<b>Required</b> . The URL to redirect to upon successful payment. Must be a valid url.

Field	Type	Description
callback.failure	url	<b>Required.</b> The URL to redirect to upon failed payment. Must be a valid url.

## Response

### Success Response (HTTP 201)

A successful request will return a 201 Created status with the following JSON payload:

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "Order Submitted Successfully",
  "is_data": true,
  "data": {
    "orderId": "ODR123456789",
    "customerUuid": "CSRT40567996",
    "paymentUrl": "https://v1.checkout.bambora.com/a403d3df20af4888bd8f7dd38f3cd7f1"
  }
}
```

## Error Responses

### HTTP 500: Internal Dependency Error

```
{
  "status_code": 500,
  "status_message": "Internal Dependency Error",
  "message": "Internal Error Occurred Please Try Again Later",
  "is_error": true,
  "errors": {
    "happenedAt": "String",
    "internalErrorDetails": "Array"
  }
}
```

### HTTP 510: Execution Exception

```
{
  "status_code": 510,
  "status_message": "Execution Exception Occurred",
  "message": "Something Went Wrong",
  "is_error": true,
  "errors": "Array"
}
```

## Step 2: Redirect to the Payment Gateway

After you successfully complete Step 1, you'll receive a **paymentUrl**. Redirect the user to this payment gateway, so they can make payment and complete the transaction.

After the user completes their payment, our system redirects them back to your application:

- If the payment is successful, they are redirected to the **success URL** you provided.
- If the payment fails, they are redirected to the **failure URL** you provided.

Additionally, our system will send a notification to the **callbackUrl** you gave in your initial request payload, updating your system on the payment status.

## Notifications via Callback URL

For `paymentLink` order, after payment completed successfully, we will notify your server via the `callbackUrl` provided by you. Follow the link below to learn how to handle callback data from your side.

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

## Best Practices

- Always validate amounts on your backend before marking payment as successful.
- Use **webhooks (`callbackUrl`)** as your source of truth, not just redirects.
- Ensure `orderDate` is a valid Unix timestamp and not expired.
- For corporate customers, `organizationId` is mandatory.

# Create Session for Invoice Order

This endpoint allows you to **export an order as an invoice** to **Front Payment Go**. Upon successful submission, it returns:

- **Order UUID**
- **Customer UUID**
- **Bank ID Verification URL** (for customer identity verification)

## Request

- **Endpoint:** `https://demo-api.frontpayment.no/api/v1/connect/orders/invoice/submit`
- **Method:** `POST`
- **Authorization:** `Bearer <token>`

## Validation Rules

Ensure your payload adheres to the following validation rules:

Field	Type	Description
<code>products.*.name</code>	<code>string</code>	<b>Required.</b> The name of the product.
<code>products.*.productId</code>	<code>string</code>	<b>Optional.</b> The unique ID of the product.
<code>products.*.quantity</code>	<code>numeric</code>	<b>Required.</b> Quantity of the product.
<code>products.*.rate</code>	<code>numeric</code>	<b>Required.</b> Rate per unit of the product.
<code>products.*.discount</code>	<code>numeric</code>	<b>Optional.</b> Discount applied to the product.
<code>products.*.tax</code>	<code>numeric</code>	<b>Required.</b> Tax rate must be (e.g., 0, 12, 15, 25), Unless you have other configuration unless otherwise configured.
<code>products.*.amount</code>	<code>numeric</code>	<b>Required.</b> Total amount for the product line item.

Field	Type	Description
<code>orderSummary.subTotal</code>	numeric	<b>Required.</b> Subtotal of all products before tax and discount.
<code>orderSummary.totalTax</code>	numeric	<b>Required.</b> The total tax for the order.
<code>orderSummary.totalDiscount</code>	numeric	<b>Required.</b> Total discount for the order.
<code>orderSummary.grandTotal</code>	numeric	<b>Required.</b> Grand total of the order.
<code>orderDate</code>	string	<b>Required.</b> Unix timestamp for the Date of the order.
<code>referenceNo</code>	string	<b>Optional.</b> Any reference information from your side. example: Order Uuid generated from your application.
<code>customerReference</code>	string	<b>Optional.</b> Customer reference
<code>orderFrom</code>	string	<b>Conditionally Required</b> if <code>fppoUuid</code> is present. If provided, the value must be <code>PARTNER</code> . This indicates that the request originates from a registered partner and is intended to update an existing record.
<code>fppoUuid</code>	string	<b>Optional</b> Use this to <b>prevent duplicates</b> . Pass the <code>orderUuid</code> from a previous response to update that specific order. If omitted, a new order is created.
<code>invoiceInterval</code>	numeric	<b>Optional</b> Invoice interval (allowed: 0,1,2).
<code>separateInvoices</code>	boolean	<b>Optional</b> Whether invoices should be separated.
<code>invoiceFeeApplicable</code>	boolean	<b>Required</b> Whether invoice fee applies.
<code>invoiceMaturity</code>	numeric	<b>Optional</b> Invoice maturity period.
<code>settings.secureDetails</code>	boolean	<b>Optional</b> Enable secure details.
<code>customerDetails.type</code>	string	<b>Required</b> Customer type ( <code>private</code> / <code>corporate</code> )
<code>customerDetails.countryCode</code>	string	<b>Required</b> Country code for the customer's phone number (e.g., "+47")
<code>customerDetails.msisdn</code>	string	<b>Required</b> Mobile Subscriber MSISDN Number (phone number).
<code>customerDetails.email</code>	email	<b>Required</b> Customer's email address.
<code>customerDetails.name</code>	string	<b>Required</b> Customer full name.

Field	Type	Description
customerDetails.personalNumber	string	<b>Required</b> if customer type is <code>private</code> . Customer's personal identification number, must be 11 characters.
customerDetails.organizationId	string	<b>Required</b> if customer type is <code>corporate</code> . Must be alphanumeric.
customerDetails.preferredLanguage	string	<b>Optional</b> Customer preferred language. Available languages are <code>en</code> , <code>no</code> , <code>sv</code> , <code>da</code> , <code>de</code> . If nothing is given it will set default to <code>no</code> .
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.
callback.callbackUrl	url	<b>Required</b> The URL to which Front Payment will send updates. Must be a valid url.
callback.success	url	<b>Required</b> The URL to redirect to upon successful payment. Must be a valid url.
callback.failure	url	<b>Required</b> The URL to redirect to upon failed payment. Must be a valid url.

## Example Request

```
{
  "products": [
    {
      "name": "Router",
      "productId": "R_1",
      "quantity": "1",
      "rate": 40,
      "discount": 0,
      "tax": "0",
      "amount": 40
    }
  ]
}
```

```
],
"orderSummary": {
  "subTotal": "40",
  "totalTax": "0",
  "totalDiscount": "0.00",
  "grandTotal": "40"
},
"orderDate": "1755150488",
"customerDetails": {
  "type": "private",
  "countryCode": "+47",
  "msisdn": "46567468",
  "email": "zahid@yopmail.com",
  "name": "Zahidul",
  "preferredLanguage": "en",
  "personalNumber": "28038712383",
  "organizationId": "",
  "address": {
    "street": "Klosterenget 144",
    "zip": "7030",
    "city": "Trondheim",
    "country": "NO"
  }
},
"referenceNo": "Dhaka",
"customerReference": "3500",
"invoiceInterval": 0,
"separateInvoices": true,
"invoiceFeeApplicable": true,
"invoiceMaturity": 10,
"callback": {
  "callbackUrl": "https://wp.frontpayment.no/?order_identifier=rRbl1FWZG59o&order_status=success",
  "success": "https://wp.frontpayment.no/?order_identifier=rRbl1FWZG59o&order_status=success",
  "failure": "https://frontpayment.no/?order_identifier=rRbl1FWZG59o&order_status=failed"
}
}
```

---

## Example Success Response

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "orderCreatedSuccessfully",
  "is_data": true,
  "data": {
    "orderId": "ODR4286244937",
    "customerUuid": "CSRT3419523642",
    "bankIdUrl":
"https://auth.current.bankid.no/precheck/auth?...state=eyJvcmlclV1aWQiOiJPRFI0Mjg2MjQ0OTM3liwiY29ubmVjdCl6InllcyJ9"
  }
}
```

## Other Responses

### Internal Error (500)

```
{
  "status_code": 500,
  "status_message": "Internal Dependency Error",
  "message": "Internal Error Occurred Please Try Again Later",
  "is_error": true,
  "errors": {
    "happenedAt": "String",
    "internalErrorDetails": "Array"
  }
}
```

### Execution Exception (510)

```
{
  "status_code": 510,
  "status_message": "Execution Exception Occurred",
  "message": "Something Went Wrong",
  "is_error": true,
  "errors": "Array"
}
```

}

# Notifications via Callback URL

[Go To](#) [Notication Via Callback Url](#) [Page](#)

# Hosted Checkout

This documentation is intended for third-party developers and partners who want to integrate their systems with our secure Hosted Checkout platform.

Our **Hosted Checkout** provides a fast, compliant, and seamless payment experience for your customers, ensuring transactions meet industry security and regulatory standards (e.g., PCI-DSS). It is designed to minimize integration effort while maximizing flexibility and user trust.

## Key features include:

- A secure, pre-built checkout page for quick deployment
- Multiple payment options, including Vipps, Google Pay, Apple Pay, Visa, and Mastercard
- Invoice and credit-check flows with BankID verification
- Automatic notifications via callback URLs for status updates
- Built-in compliance and security measures to protect sensitive data

We are continuously expanding our payment ecosystem. More payment methods and features will be added over time, ensuring your integration stays current with market needs.

This guide will walk you through the integration process step by step—from creating a payment link and redirecting users to the checkout page, to handling notifications and ensuring a smooth payment experience.

**Compliance Note:** Front Payment's Hosted Checkout is designed to follow applicable standards, including PCI-DSS and BankID security requirements. Ensure your integration handles tokens and customer data securely and in line with local regulations.

## Prerequisites

Before you start the integration, make sure you have:

### 1. API Access:

- A valid API key and Bearer Token from Front Payment
- Access to the demo and production environments

### 2. Merchant Setup:

- Your merchant account configured with Front Payment
- Enabled payment methods (Vipps, Google Pay, Apple Pay, Visa, Mastercard)

### 3. Technical Requirements:

- Ability to make HTTPS API calls
- Secure storage of tokens and keys
- Callback endpoints to handle payment status updates

#### 4. Test Environment:

- For testing, contact `nafees.faraz@frontpayment.no` to gain access to the demo environment

## Step 1: Create a Payment Link

To initiate a payment, your system will need to call our `create` endpoint to generate a payment link. This link will redirect your users to our secure hosted checkout page.

### Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/hosted/orders/payment-link/create
```

### 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

The request body should be a JSON object containing details about the order, customer, and callback URLs.

```
{
  "products": [
    {
      "name": "Router",
      "productId": "R_1",
      "quantity": "1",
      "rate": 5,
      "discount": 0,
      "tax": "0",
      "amount": 5
    }
  ]
}
```

```
],
"orderSummary": {
  "subTotal": "5",
  "totalTax": "0",
  "totalDiscount": "0.00",
  "grandTotal": "5"
},
"orderDate": "1756278578",
"dueDateForPaymentLink": "1756278578",
"customerDetails": {
  "type": "private",
  "countryCode": "+47",
  "msisdn": "46567468",
  "email": "john_doe@yopmail.com",
  "name": "John Doe",
  "personalNumber": null,
  "organizationId": null,
  "address": {
    "street": "Klosterenget 144",
    "zip": "7030",
    "city": "Trondheim",
    "country": "NO"
  }
},
"checkoutLanguage": "en",
"referencesNo": null,
"customerNotes": null,
"callback": {
  "callbackUrl": "https://your-callback-url.com/callback?order_identifier=rRbl1FWZG59o&order_status=failed",
  "success": "https://your-site-url.com/?order_identifier=rRbl1FWZG59o&order_status=success",
  "failure": "https://your-site-url.com/?order_identifier=rRbl1FWZG59o&order_status=failed"
}
}
```

## Validation Rules

Ensure your payload adheres to the following validation rules:

Field	Type	Description
<code>products.*.name</code>	<code>string</code>	<b>Required.</b> Name of the product.

Field	Type	Description
<code>products.*.productId</code>	string	<b>Optional.</b> Unique identifier for the product.
<code>products.*.quantity</code>	numeric	<b>Required.</b> Quantity of the product.
<code>products.*.rate</code>	numeric	<b>Required.</b> Rate per unit of the product.
<code>products.*.discount</code>	numeric	<b>Optional.</b> Discount applied to the product.
<code>products.*.tax</code>	numeric	<b>Required.</b> Tax rate (e.g., 0, 12, 15, 25). Unless you have other configuration.
<code>products.*.amount</code>	numeric	<b>Required.</b> Total amount for the product line item.
<code>orderSummary.subTotal</code>	numeric	<b>Required.</b> Subtotal of all products before tax and discount.
<code>orderSummary.totalTax</code>	numeric	<b>Required.</b> Total tax for the order.
<code>orderSummary.totalDiscount</code>	numeric	<b>Required.</b> Total discount for the order.
<code>orderSummary.grandTotal</code>	numeric	<b>Required.</b> Grand total of the order.
<code>orderDate</code>	string	<b>Required.</b> Unix timestamp for the Date of the order, which must be current or future date.
<code>dueDateForPaymentLink</code>	string	<b>Required.</b> Unix timestamp for the due date of the payment link.
<code>orderFrom</code>	string	<b>Conditionally Required</b> if <code>fpgoUuid</code> is present. If provided, the value must be <code>PARTNER</code> . This indicates that the request originates from a registered partner and is intended to update an existing record.
<code>fpgoUuid</code>	string	<b>Optional</b> Use this to <b>prevent duplicates</b> . Pass the <code>orderUuid</code> from a previous response to update that specific order. If omitted, a new order is created.
<code>customerDetails.type</code>	string	<b>Required.</b> Customer type must be either <code>private</code> or <code>corporate</code> .
<code>customerDetails.countryCode</code>	string	<b>Required.</b> Country code for the customer's phone number (e.g., "+47").
<code>customerDetails.msisdn</code>	string	<b>Required.</b> Mobile Subscriber ISDN Number (phone number).
<code>customerDetails.email</code>	email	<b>Required.</b> Customer's email address.

Field	Type	Description
customerDetails.name	string	<b>Required.</b> Customer's full name.
customerDetails.personalNumber	string	<b>Optional</b> Customer's personal identification number, must be 11 characters.
customerDetails.organizationId	numeric	<b>Required</b> if customerDetails.type is corporate. Must be number
customerDetails.address.street	string	<b>Optional.</b> Street address of the customer.
customerDetails.address.zip	string	<b>Optional.</b> Zip code of the customer's address.
customerDetails.address.city	string	<b>Optional.</b> City of the customer's address.
customerDetails.address.country	string	<b>Optional.</b> ISO Alpha-2 country code (e.g., "NO"). Custom validation IsoAlpha2Country applies.
checkoutLanguage	string	<b>Optional.</b> Customer checkout page language. Available languages are en, no, sv, da, de. If nothing is given it will set default to no.
referencesNo	string	<b>Optional.</b> Any reference number for the order.
customerNotes	string	<b>Optional.</b> Any notes from the customer.
callback.callbackUrl	url	<b>Required.</b> The URL we will notify upon payment status changes.
callback.success	url	<b>Required.</b> The URL to redirect the user to if the payment is successful.
callback.failure	url	<b>Required.</b> The URL to redirect the user to if the payment fails.

## Response

A successful request will return a 201 Created status with the following JSON payload:

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "Payment Link Created Successfully",
  "is_data": true,
  "data": {
    "orderId": "ODR344175661",
```

```
"customerUuid": "CSRT197366289",  
"paymentUrl": "https://demo.frontpayment.no/order/hosted/ODR344175661/checkout"  
}  
}
```

The `paymentUrl` in the response is crucial for the next step.

---

## Step 2: Redirect to the Payment Page

After successfully creating a payment link, your system should redirect your user to the `paymentUrl` received in the response from Step 1. This will take your user to our secure hosted checkout page, where they can complete the payment.

### Payment Options

On the hosted checkout page, users will be presented with two primary payment options:

#### 1. Payment Methods (Vipps, Google Pay, Visa, Mastercard)

If the user selects one of the standard payment methods:

- They will be redirected to a secure payment page where they can enter their credentials (e.g., card details, mobile payment app details).
- Upon successful completion of the payment, the user will be redirected to the Front Payment success page.
- This success page will feature a prominent "**Back To Site**" button. Clicking this button will redirect the user back to your system, using the `callback.success` URL you provided in the initial request. If not clicked on this button, user will be automatically redirected into this `callback.success` URL after 5 seconds.

#### 2. Pay By Invoice

The **invoice distribution method** is determined based on the information provided in the request. The system follows this priority order:

1. **EHF or E-invoice (Preferred)** – Requires a valid P-number or organization number.
2. **Email** – Used if EHF/E-invoice cannot be delivered.
3. **Postal Mail** – Used if no valid email address is available or email delivery fails.

If none of the above delivery methods are successful, our customer service team will notify the client to resolve the issue.

**When a *Private Customer* chooses the "Pay By Invoice" option:**

- They will be redirected to a **Bank ID verification page** to verify their identity.
- Upon successful Bank ID verification, a **credit check** will be performed in the background by Front Payment.
- If the credit check yields a positive score (minimum 315), the user will proceed to a **document signing flow** to finalize the invoice agreement.
- Once the document is signed, an invoice will be created, and the user will be redirected to the Front Payment success page.
- Similar to the payment methods flow, this success page will also include a "**Back To Site**" button, which will redirect the user back to your system via the provided `callback.success` URL.

#### **When a *Corporate Customer* chooses the "Pay By Invoice" option:**

- For corporate customer, BankID flow will be skipped and an Invoice will be automatically created after the user fills up all the information & press "**Pay Now**" button.

## Notifications via Callback URL

For `invoice` order and customer type is `private`, after BankID verification is completed successfully our system will notify you via the `callbackUrl` provider by you.

For `invoice` order and customer type is `corporate`, the order will be directly invoiced after user select invoice method from the payment page and frontpayment will notify you via `callbackUrl`.

For `paymentLink` order, after payment completed successfully, we will notify your server via the `callbackUrl` provided by you.

For an invoice, our system will also notify you for any future status changes in our system via `callbackUrl`. Follow the link below to learn how to handle callback data from your side.

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