

Order Management

- [Send Payment Link](#)
- [Send Invoice](#)
- [Send E-Faktura](#)
- [Send EHF Invoice](#)
- [Resend Payment Link](#)
- [Cancel Order](#)
- [Refund Order](#)
- [Get All Order Status](#)
- [Get Order Status By UUID](#)
- [Get Order Details By UUID](#)
- [Get Invoice Number By UUID](#)
- [Delayed Charge](#)

Send Payment Link

With this API endpoint, you can export orders to Front Payment Go and get Order Uuid and Customer Uuid in return.

Endpoint

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

Authorization

To access this endpoint, include a `Bearer Token` in the `Authorization` header of your request. You can obtain this token from Frontpayment.

Example Authorization Header: `Authorization: Bearer YOUR_FRONTPAID_BEARER_TOKEN`

Request Payload

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

```
{
  "products": [
    {
      "name": "Hair Wash",
      "productId": "VFDDF",
      "quantity": "1",
      "rate": 51,
      "discount": 0,
      "tax": "0",
      "amount": 51
    }
  ],
  "orderSummary": {
    "subTotal": "51.00",
    "totalTax": "0.00",
```

```
"totalDiscount": "0.00",
"grandTotal": "51.00"
},
"sendOrderBy": {
  "sms": false,
  "email": false
},
"orderDate": "1703040812",
"customerDetails": {
  "type": "private",
  "countryCode": "+47",
  "msisdn": "46567468",
  "email": "test@yopmail.com",
  "firstName": "",
  "name": "Kari Nordmann",
  "preferredLanguage": "en",
  "personalNumber": "12345678901",
  "address": {
    "street": "Luramyrvеien 65",
    "zip": "4313",
    "city": "Sandnes",
    "country": "NO"
  }
},
"invoiceAsPaymentOption": true,
"referenceNo": null,
"customerReference": null,
"secureDetails": true,
"callback": {
  "callbackUrl": "https://example.com/callback-url"
}
}
```

Validation Rules

Make sure your request meets the following requirements:

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

Field	Type	Description
products.*.productId	string	Optional Unique identifier for the product.
products.*.quantity	numeric	Required Quantity of the product.
products.*.rate	numeric	Required Rate per unit of the product.
products.*.discount	numeric	Optional Discount applied to 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.
orderSummary.subTotal	numeric	Required Subtotal of all products before tax and discount.
orderSummary.totalTax	numeric	Required Total tax for the order.
orderSummary.totalDiscount	numeric	Required Total discount for the order.
orderSummary.grandTotal	numeric	Required Grand total of the order.
sendOrderBy.sms	boolean	Nullable If you want to send order payment link via customer phone number.
sendOrderBy.email	boolean	Nullable If you want to send order payment link via email.
orderDate	string	Required Provide Current / Future Date as Unix timestamp for the Date of the order.
customerDetails.countryCode	string	Required if customerDetails.email is not given. Country code for the customer's phone number (e.g., "+47").
customerDetails.msisdn	string	Required if customerDetails.email is not given. Mobile Subscriber ISDN Number (phone number).
customerDetails.email	string	Required if customerDetails.msisdn is not given. Customer's email address.
customerDetails.firstName	string	Optional Customer's First name.
customerDetails.name	string	Required Customer's full name.
customerDetails.preferredLanguage	string	Optional Customer preferred language. Available languages are en, no, sv, da, de. If nothing is given it will set default to no.

Field	Type	Description
<code>customerDetails.type</code>	string	Required You can provide customer type is <code>private</code> or <code>corporate</code> .
<code>customerDetails.personalNumber</code>	string	Optional Customer's personal identification number, must be exactly 11 characters containing only numbers and cannot contain spaces. When Customer type is <code>private</code> then you can use this for add personal number.
<code>customerDetails.organizationId</code>	string	Conditional Required Organization identification number, must contain only numbers and cannot contain spaces. When Customer type is <code>corporate</code> then this field is required. Otherwise you can add this as <code>null</code> or remove from payload.
<code>customerDetails.address.street</code>	string	Required Street address of the customer.
<code>customerDetails.address.zip</code>	string	Required Zip code of the customer's address.
<code>customerDetails.address.city</code>	string	Required City of the customer's address.
<code>customerDetails.address.country</code>	string	Required ISO Alpha-2 country code (e.g., <code>NO</code>). Custom validation <code>IsoAlpha2Country</code> applies.
<code>invoiceAsPaymentOption</code>	boolean	Required Enable Invoice as payment option then set <code>true</code> .
<code>referenceNo</code>	string	Nullable Any reference number.
<code>customerReference</code>	string	Nullable Any value for customer reference.
<code>secureDetails</code>	boolean	Optional. If <code>secureDetails</code> is set to <code>True</code> , the order details will not be accessible until the user's PNumber has been verified through BankID.
<code>callback.callbackUrl</code>	url	Optional To receive real-time notifications on order state changes, you must provide a callback url. This is an server-to-server <code>HTTP GET</code> request.

Response

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

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "orderAddedSuccessfully",
  "is_data": false,
  "data": {
    "uuid": "ODR3506777330",
    "customerUuid": "CSRT3463048878"
  }
}
```

API returns a `500` or `510` error, it means something failed on the server side

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

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

Notifications via Callback URL

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.

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.

Send Invoice

Introduction

This API acts as the bridge between your application and **Front Payment**, making it effortless to create invoices automatically. Instead of manually entering customer and order information, you can simply send the details directly from your system. This ensures a smooth and quick process, so that once a transaction or order is complete, a precise invoice is instantly generated and delivered to the right customer. It streamlines your billing process and helps you keep everything organized.

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.

Use Cases

Here are a few ways your application can use this API:

- **Automated E-commerce Billing**

When a customer completes a purchase on your online store, your application can instantly use this API to generate and send an invoice to their email, creating a seamless and immediate billing process.

- **Subscription Management**

For services with recurring billing, the API can be used to automatically generate and send invoices to customers at regular intervals (e.g., monthly or annually), eliminating the need for manual billing cycles.

- **Service & Project Invoicing**

After a project or service is completed, your application can use the API to automatically draft and send a detailed invoice to the client, including all project details, labor costs, and materials.

- **Financial & Accounting System Integration**

Your application can use this API to sync order and billing data directly with an accounting system, ensuring all financial records are up-to-date and accurate without manual data entry.

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. Technical Requirements:

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

3. Test Environment:

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

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/orders/invoice/create
```

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

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

```
{
  "products": [
    {
      "name": "Hair Wash",
      "productId": "VFDDF",
```

```
    "quantity": "1",
    "rate": 51,
    "discount": 0,
    "tax": "0",
    "amount": 51
  }
],
"orderSummary": {
  "subTotal": "51.00",
  "totalTax": "0.00",
  "totalDiscount": "0.00",
  "grandTotal": "51.00"
},
"orderDate": "1703040812",
"customerDetails": {
  "type": "private",
  "countryCode": "+47",
  "msisdn": "46567468",
  "email": "test@yopmail.com",
  "firstName": "",
  "name": "Kari Nordmann",
  "preferredLanguage": "en",
  "personalNumber": "12345678901",
  "organizationId": null,
  "address": {
    "street": "Luramyrvеien 65",
    "zip": "4313",
    "city": "Sandnes",
    "country": "NO"
  }
},
"invoiceInterval": 0,
"invoiceMaturity": 10,
"invoiceFeeApplicable": true,
"separateInvoices": true,
"referenceNo": null,
"customerReference": null,
"callback": {
  "callbackUrl": "https://example.com/callback-url"
}
```

```
}
```

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.*.quantity	numeric	Required Quantity of the product.
products.*.rate	numeric	Required Rate per unit of the product.
products.*.discount	numeric	Optional Discount applied to 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.
orderSummary.subTotal	numeric	Required Subtotal of all products before tax and discount.
orderSummary.totalTax	numeric	Required Total tax for the order.
orderSummary.totalDiscount	numeric	Required Total discount for the order.
orderSummary.grandTotal	numeric	Required Grand total of the order.
orderDate	string	Required Unix timestamp for the Date of the order.
customerDetails.countryCode	string	Required Country code for the customer's phone number (e.g., "+47").
customerDetails.msisdn	string	Required Mobile Subscriber ISDN Number (phone number).
customerDetails.email	email	Required Customer's email address.
customerDetails.firstName	email	Optional Customer's first name.
customerDetails.name	email	Required Customer's full name.

Field	Type	Description
customerDetails.preferredLanguage	string	Optional 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.type	string	Required You can provide customer type is <code>private</code> or <code>corporate</code> .
customerDetails.personalNumber	string	Optional Customer's personal identification number, must be exactly 11 characters containing only numbers and cannot contain spaces. When Customer type is <code>private</code> then you can use this for add personal number.
customerDetails.organizationId	string	Conditional Required Organization identification number, must contain only numbers and cannot contain spaces. When Customer type is <code>corporate</code> then this field is required. Otherwise you can add this as <code>null</code> or remove from payload.
customerDetails.address.street	string	Required Street address of the customer.
customerDetails.address.zip	string	Required Zip code of the customer's address.
customerDetails.address.city	string	Required City of the customer's address.
customerDetails.address.country	string	Required ISO Alpha-2 country code (e.g., <code>NO</code>). Custom validation <code>IsoAlpha2Country</code> applies.
invoiceInterval	numeric	Optional Default value is = <code>0</code> . You can change it to <code>0</code> , <code>1</code> , <code>2</code> . Daily = <code>0</code> , Once a month = <code>1</code> , Twice a month = <code>2</code> .
invoiceFeeApplicable	boolean	Required Default value is <code>true</code> .
invoiceMaturity	numeric	Optional If specified, the value must be <code>10</code> for private customers; for corporate customers, the value may be <code>14</code> , <code>30</code> , or <code>45</code> .
separateInvoices	boolean	Optional Default value is <code>true</code> .
referenceNo	string	Nullable Any reference number.
customerReference	string	Nullable Any value for customer reference.

Field	Type	Description
callback.callbackUrl	url	Optional To receive real-time notifications on order state changes, you must provide a callback url. This is an server-to-server <code>HTTP GET</code> request.

Response

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

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "orderAddedSuccessfully",
  "is_data": false,
  "data": {
    "uuid": "ODR3506777330",
    "customerUuid": "CSRT3463048878"
  }
}
```

API returns a `500` or `510` error, it means something failed on the server side

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

```
{
  "status_code": 510,
  "status_message": "Execution Exception Occurred",
  "message": "somethingWentWrong",
  "is_error": true,
}
```

```
"errors": "Array"  
}
```

Notifications via Callback URL

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.

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 `Notication Via Callback Url` Page](#)

Send E-Faktura

Introduction

This API provides a way for your application to automate the creation of **e-invoices** for private customers. It allows you to programmatically send all necessary customer and product information to **Front Payment**. This eliminates the need for manual data entry, ensuring that e-invoices are generated quickly and accurately, which is essential for businesses that need to handle a high volume of transactions with individual customers.

Key Benefits

- **Increased Efficiency:** Automates the entire invoicing process, saving significant time and resources that would otherwise be spent on manual billing.
- **Enhanced Accuracy:** By directly transferring data from your application, the risk of human error in invoice creation is eliminated, ensuring that all details are correct.
- **Improved Customer Experience:** Provides customers with a seamless and instant way to receive and manage their invoices through their preferred digital channels.
- **Real-Time Status Updates:** The callback feature allows your application to receive instant notifications on the status of each invoice, enabling you to manage the billing workflow proactively.

Use Cases

- **E-commerce:** Instantly invoice customers for online purchases.
- **Recurring Billing:** Automate invoices for subscriptions and membership fees.
- **Service-based Business:** Easily bill clients for project or freelance work.
- **Utility & Bill Payments:** Send regular e-invoices for monthly services.

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. Technical Requirements:

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

3. Test Environment:

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

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/orders/faktura/create
```

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

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

```
{
  "products": [
    {
      "name": "Hair Wash",
      "productId": "VFDDF",
      "quantity": "1",
      "rate": 51,
      "discount": 0,
      "tax": "0",
      "amount": 51
    }
  ],
  "orderSummary": {
```

```
"subTotal": "51.00",
"totalTax": "0.00",
"totalDiscount": "0.00",
"grandTotal": "51.00"
},
"orderDate": "1703040812",
"customerDetails": {
  "countryCode": "+47",
  "msisdn": "46567468",
  "email": "test@yopmail.com",
  "name": "Kari Nordmann",
  "preferredLanguage": "en",
  "personalNumber": "12345678901",
  "address": {
    "street": "Luramyrvеien 65",
    "zip": "4313",
    "city": "Sandnes",
    "country": "NO"
  }
},
"invoiceInterval": 0,
"invoiceMaturity": 10,
"invoiceFeeApplicable": true,
"separateInvoices": true,
"referenceNo": null,
"customerReference": null,
"callback": {
  "callbackUrl": "https://example.com/callback-url"
}
}
```

Validation Rules

Make sure your request meets the following requirements:

Field	Type	Description
<code>products.*.name</code>	string	Required Name of the product.
<code>products.*.productId</code>	string	Optional Unique identifier for the product.

Field	Type	Description
products.*.quantity	numeric	Required Quantity of the product.
products.*.rate	numeric	Required Rate per unit of the product.
products.*.discount	numeric	Optional Discount applied to 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.
orderSummary.subTotal	numeric	Required Subtotal of all products before tax and discount.
orderSummary.totalTax	numeric	Required Total tax for the order.
orderSummary.totalDiscount	numeric	Required Total discount for the order.
orderSummary.grandTotal	numeric	Required Grand total of the order.
orderDate	string	Required Unix timestamp for the Date of the order.
customerDetails.countryCode	string	Required Country code for the customer's phone number (e.g., "+47").
customerDetails.msisdn	string	Required Mobile Subscriber ISDN Number (phone number).
customerDetails.email	email	Required Customer's email address.
customerDetails.name	email	Required Customer's full name.
customerDetails.preferredLanguage	string	Optional Customer preferred language. Available languages are en, no, sv, da, de. If nothing is given it will set default to no.
customerDetails.personalNumber	string	Required Customer's personal identification number, must be exactly 11 characters containing only numbers and cannot contain spaces.
customerDetails.address.street	string	Required Street address of the customer.
customerDetails.address.zip	string	Required Zip code of the customer's address.
customerDetails.address.city	string	Required City of the customer's address.

Field	Type	Description
customerDetails.address.country	string	Required ISO Alpha-2 country code (e.g., <code>NO</code>). Custom validation <code>IsoAlpha2Country</code> applies.
invoiceInterval	numeric	Optional Default value is = <code>0</code> . You can change it to <code>0</code> , <code>1</code> , <code>2</code> . Daily = <code>0</code> , Once a month = <code>1</code> , Twice a month = <code>2</code> .
invoiceFeeApplicable	boolean	Required Default value is <code>true</code> .
invoiceMaturity	numeric	Optional If specified, the value must be <code>10</code> .
separateInvoices	boolean	Optional Default value is <code>true</code> .
referenceNo	string	Nullable Any reference number.
customerReference	string	Nullable Any value for customer reference.
callback.callbackUrl	url	Required To receive real-time notifications on order state changes, you must provide a callback url. This is an server-to-server <code>HTTP GET</code> request.

Response

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

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "orderAddedSuccessfully",
  "is_data": false,
  "data": {
    "uuid": "ODR3506777330",
    "customerUuid": "CSRT3463048878"
  }
}
```

API returns a `500` or `510` error, it means something failed on the server side

```
{
  "status_code": 500,
  "status_message": "Internal Dependency Error",
}
```

```
"message": "internalErrorOccurredPleaseTryAgainLater",
"is_error": true,
"errors": {
  "happenedAt": "String",
  "internalErrorDetails": "Array"
}
}
```

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

Notifications via Callback URL

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.

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 `Notication Via Callback Url` Page](#)

Send EHF Invoice

The **Send EHF Invoice** endpoint allows merchants to generate and send standardized electronic invoices (EHF) to corporate customers in Norway. EHF invoices conform to national and PEPPOL regulations, enabling seamless delivery to customer accounting systems and public sector entities. This API integrates Front Payment's digital invoicing with your order workflow — encapsulating products, billing details, due dates, and delivery preferences into an EHF-enabled invoice.

Upon success, you'll receive an `orderUuid` and `customerUuid` to track the invoice lifecycle. To check invoice status go to [Get Order Status By UUID](#) page

Key Benefits

- **Automate invoicing workflows** and reduce manual billing overhead.
- **Ensure compliance** with Norwegian EHF (PEPPOL) standards.
- **Deliver invoices directly** to recipients' systems via the PEPPOL network.
- **Monitor invoice status** or reconcile with callbacks or queries.

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. Technical Requirements:

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

3. Test Environment:

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

Endpoint

POST https://demo-api.frontpayment.no/api/v1/connect/orders/ehf/create

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

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

```
{
  "products": [
    {
      "name": "Hair Wash",
      "productId": "VFDDF",
      "quantity": "1",
      "rate": 51,
      "discount": 0,
      "tax": "0",
      "amount": 51
    }
  ],
  "orderSummary": {
    "subTotal": "51.00",
    "totalTax": "0.00",
    "totalDiscount": "0.00",
    "grandTotal": "51.00"
  },
  "orderDate": "1703040812",
  "customerDetails": {
    "countryCode": "+47",
    "msisdn": "46567468",
    "email": "test@yopmail.com",
    "name": "Kari Nordmann",
  }
}
```

```

"preferredLanguage": "en",
"organizationId": "123456789",
"address": {
  "street": "Luramyrvеien 65",
  "zip": "4313",
  "city": "Sandnes",
  "country": "NO"
}
},
"invoiceInterval": 0,
"invoiceMaturity": 10,
"invoiceFeeApplicable": true,
"separateInvoices": true,
"referenceNo": null,
"customerReference": null,
"callback": {
  "callbackUrl": "https://example.com/callback-url"
}
}

```

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.*.quantity	numeric	Required Quantity of the product.
products.*.rate	numeric	Required Rate per unit of the product.
products.*.discount	numeric	Optional Discount applied to 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.

Field	Type	Description
orderSummary.subTotal	numeric	Required Subtotal of all products before tax and discount.
orderSummary.totalTax	numeric	Required Total tax for the order.
orderSummary.totalDiscount	numeric	Required Total discount for the order.
orderSummary.grandTotal	numeric	Required Grand total of the order.
orderDate	string	Required Unix timestamp for the Date of the order.
customerDetails.countryCode	string	Required Country code for the customer's phone number (e.g., "+47").
customerDetails.msisdn	string	Required Mobile Subscriber ISDN Number (phone number).
customerDetails.email	email	Required Customer's email address.
customerDetails.name	email	Required Customer's full name.
customerDetails.preferredLanguage	string	Optional 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.organizationId	string	Required Organization identification number, must contain only numbers and cannot contain spaces
customerDetails.address.street	string	Required Street address of the customer.
customerDetails.address.zip	string	Required Zip code of the customer's address.
customerDetails.address.city	string	Required City of the customer's address.
customerDetails.address.country	string	Required ISO Alpha-2 country code (e.g., <code>NO</code>). Custom validation <code>IsoAlpha2Country</code> applies.
invoiceInterval	numeric	Optional Default value is = <code>0</code> . You can change it to <code>0</code> , <code>1</code> , <code>2</code> . Daily = <code>0</code> , Once a month = <code>1</code> , Twice a month = <code>2</code> .
invoiceFeeApplicable	boolean	Required Default value is <code>true</code> .
invoiceMaturity	numeric	Optional Default value is <code>14</code> . if you want to set the value then give <code>14</code> , <code>30</code> or <code>45</code> .
separateInvoices	boolean	Optional Default value is <code>true</code> .

Field	Type	Description
referenceNo	string	Nullable Any reference number.
customerReference	string	Nullable Any value for customer reference.
callback.callbackUrl	url	Required To receive real-time notifications on order state changes, you must provide a callback url. This is an server-to-server <code>HTTP GET</code> request.

Response

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

```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "orderAddedSuccessfully",
  "is_data": false,
  "data": {
    "uuid": "ODR3506777330",
    "customerUuid": "CSRT3463048878"
  }
}
```

API returns a `500` or `510` error, it means something failed on the server side

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

```
{
  "status_code": 510,
```

```
"status_message": "Execution Exception Occurred",  
"message": "somethingWentWrong",  
"is_error": true,  
"errors": "Array"  
}
```

Notifications via Callback URL

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.

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 `Notication Via Callback Url` Page](#)

Resend Payment Link

The **Resend Payment Link** endpoint allows your system to resend the payment link associated with an existing order to a customer. This is useful when the original link was lost, not received, expired, or needs to be sent again.

You can resend the link via:

- **SMS** (requiring `countryCode` and `msisdn`)
- **Email** (requiring a valid `email` address)

Importantly, the SMS or email does **not** need to match the contact details already stored with the customer profile. The payment link can be sent to a **new phone number** or a **different email address**, making it flexible for scenarios where the customer wants to use an alternative contact method.

This ensures that the customer can always receive the payment link, even if their original phone number or email is unavailable.

You will find details about the endpoint URL, authentication, request structure, validation rules, and sample responses below.

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/orders/resend/{{ORDER_UUID}}
```

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

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

```
{
  "countryCode": +47,
  "msisdn": "xxxxxxxx",
  "email": "example-email@email.com"
}
```

Validation Rules

Make sure your request meets the following requirements:

Field	Type	Description
countryCode	string	Conditional Required Country code for the customer's phone number (e.g., "+47").
msisdn	string	Conditional Required Mobile Subscriber MSISDN Number (phone number). If you want to resend order payment link via customer phone number.
email	email	Conditional Required Customer's email address. If you want to resend order payment link via email.

Response

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

```
{
  "status_code": 202,
  "status_message": "OK",
  "message": "resentOrderSuccessfully",
  "is_data": true,
  "data": null
}
```

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

```
{
  "status_code": 404,
  "status_message": "Not Found",
}
```

```
"message": "orderNotFound",
"is_error": false,
"errors": null
}
```

API returns a `417` error, it means requested payload is not valid.

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

API return a `400` error, it means your requested order is already `PAID` or `CANCELLED`.

```
{
  "status_code": 400,
  "status_message": "Conflict of Business Logic",
  "message": "orderStatusAlreadyPaid",
  "is_error": false,
  "errors": null
}
```

```
{
  "status_code": 400,
  "status_message": "Conflict of Business Logic",
  "message": "orderStatusAlreadyCancelled",
  "is_error": false,
  "errors": null
}
```

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"
```

```
}
```

Cancel Order

This API allows your application to **cancel a specific order**. By providing the order's unique ID and a reason for the cancellation, you can programmatically update the order's status. This is crucial for managing order changes, such as when a customer requests a cancellation, an item is out of stock, or an order needs to be stopped for any other reason.

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/orders/cancel/{{ORDER_UUID}}
```

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

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

```
{
  "cancellationNote": "Your cancellation Note here"
}
```

Validation Rules

Make sure your request meets the following requirements:

Field	Type	Description
<code>cancellationNote</code>	<code>string</code>	Required Order cancellation note.

Response

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

```
{
  "status_code": 202,
  "status_message": "OK",
  "message": "cancelledOrderSuccessfully",
  "is_data": false,
  "data": null
}
```

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

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

API returns a `417` error, it means requested payload is not valid.

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

API return a `400` error, it means your requested order is already `PAID`.

```
{
  "status_code": 400,
  "status_message": "Conflict of Business Logic",
  "message": "orderStatusAlreadyPaid",
  "is_error": false,
}
```

```
"errors": null
}
```

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"
}
```

Refund Order

The **Refund Order** endpoint enables merchants to initiate either full or partial refunds for an order using its `Order UUID`. Depending on your business workflow, you can refund the entire order or only specific items. Upon successful submission, the API responds with a `202 Accepted`, indicating that your refund request has been accepted and is pending processing.

This endpoint is ideal for scenarios such as:

- **Returns & Exchanges:** Revert payment for returned or exchanged items.
- **Order Modifications:** Adjust invoices or correct billing mistakes.
- **Partial Cancellations:** Process refunds for specific products rather than full orders.

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/orders/refund/{{ORDER_UUID}}
```

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

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

```
{
  "type": "regular",
  "grandTotal": 10,
  "products": [
    {
      "id": 5410,
      "amount": 10
    }
  ]
}
```

```
]
}
```

Retrieve Product ID from [Get Order Details API](#).

Validation Rules

Make sure your request meets the following requirements:

Field	Type	Description
<code>type</code>	<code>string</code>	Required Using type. Available types are <code>regular</code> and <code>invoiced</code>
<code>grandTotal</code>	<code>numeric</code>	Required Grand total of the refunded amount.
<code>products.*id</code>	<code>numeric</code>	Required Order product id. From which product you want to refund.
<code>products.*.amount</code>	<code>numeric</code>	Required Refund amount for the product.

Response

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

```
{
  "status_code": 202,
  "status_message": "OK",
  "message": "orderRefundedSuccessfully",
  "is_data": true,
  "data": null
}
```

Error Response

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

```
{
  "status_code": 404,
```

```
"status_message": "Not Found",
"message": "orderNotFound",
"is_error": false,
"errors": 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": "somethingWentWrong",
  "is_error": true,
  "errors": "Array"
}
```

Others refund rejections errors

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

```
{
  "status_code": 400,
  "status_message": "Conflict of Business Logic",
```

```
"message": "refundRejectionForRefundRequestGreaterThanOrderAmount",  
"is_error": false,  
"errors": null  
}
```

```
{  
  "status_code": 400,  
  "status_message": "Conflict of Business Logic",  
  "message": "refundRejectionForProductAmountExceed",  
  "is_error": true,  
  "errors": "Array"  
}
```

```
{  
  "status_code": 400,  
  "status_message": "Conflict of Business Logic",  
  "message": "refundRejectionForWeeklyThresholdExceed",  
  "is_error": true,  
  "errors": null  
}
```

```
{  
  "status_code": 400,  
  "status_message": "Conflict of Business Logic",  
  "message": "refundRejectionForRequestAmountThresholdExceed",  
  "is_error": true,  
  "errors": null  
}
```

Get All Order Status

This API allows your application to **retrieve the status of your orders** in the FrontGo system. You can get a complete overview of all orders or filter for specific statuses, such as `PAID`, `CANCELLED`, or `INVOICED`. This is a vital tool for maintaining a real-time, comprehensive view of your payment operations and keeping your internal systems synchronized with Front Payment.

Endpoint

- **Fetch all orders:**

```
GET https://demo-api.frontpayment.no/api/v1/connect/orders/status
```

- **Fetch orders with a specific status such as "Paid":**

```
https://demo-api.frontpayment.no/api/v1/connect/orders/status?type=Paid
```

Query Parameters

`type` (optional): Filter the orders by their status. You can use one of the following values:

- **SENT**
- **PAID**
- **CANCELLED**
- **EXPIRED**
- **INVOICED**
- **REFUNDED**
- **PARTIAL REFUNDED**

Authorization

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

Example:

```
Authorization: Bearer YOUR_FRONTPAYMENT_BEARER_TOKEN
```

Response

A successful request will return a `200` status with the following JSON payload:

```
{
  "status_code": 200,
  "status_message": "OK",
  "message": "Order Status Retrieved Successfully",
  "is_data": true,
  "data": [
    {
      "uuid": "String",
      "status": "String"
    },
    {
      "uuid": "String",
      "status": "String"
    }
  ]
}
```

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"
}
```

Get Order Status By UUID

The **Get Order Status by UUID** endpoint lets you query the latest processing state of a single order using its `ORDER_UUID`. It's useful for polling after a checkout, reconciling webhook events, or debugging edge cases in your order flow.

Typical Uses

- **Post-payment confirmation:** Verify status after redirect/webhook before fulfilling.
- **Support tooling:** Let agents paste a UUID to see live status.

Endpoint

```
GET https://demo-api.frontpayment.no/api/v1/connect/orders/status/{{ORDER_UUID}}
```

Authorization

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

Example:

```
Authorization: Bearer YOUR_FRONTPAYMENT_BEARER_TOKEN
```

Response

A successful request will return a `200` status with the following JSON payload:

```
{
  "status_code": 200,
  "status_message": "OK",
  "message": "Order Status Retrieved Successfully",
  "is_data": true,
  "data": {
    "uuid": "String",
    "status": "String",
```

```
"orderAmount": 40,  
"refundedAmount": 0  
}  
}
```

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

```
{  
  "status_code": 404,  
  "status_message": "Not Found",  
  "message": "orderNotFound",  
  "is_error": false,  
  "errors": null  
}
```

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"  
}
```

Get Order Details By UUID

This API enables your application to **retrieve detailed information for a single order** using its unique Order UUID. It provides a complete snapshot of any specific transaction, including its status, products, customer details, and payment summary. This is essential for customer support, order tracking, and offering users a comprehensive view of their purchases.

Use Cases

- **Customer Service:** When a customer has a query, a support agent can quickly use the Order UUID to access all transaction details, enabling prompt and accurate responses.
- **Order History:** Your application can display a detailed order history to users, including itemized lists and total costs, for transparency and convenience.
- **Internal Auditing:** Pull specific order details for auditing, accounting, or reconciliation purposes, ensuring accuracy and compliance.

Endpoint

```
GET https://demo-api.frontpayment.no/api/v1/connect/orders/details/{{ORDER_UUID}}
```

Authorization

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

Example:

```
Authorization: Bearer YOUR_FRONTPAYMENT_BEARER_TOKEN
```

Response

A successful request will return a `200 OK` status with the following JSON payload:

```
{
  "status_code": 200,
  "status_message": "OK",
  "message": "orderRetrievedSuccessfully",
```

```
"is_data": true,
"data": {
  "type": "String",
  "orderId": "String",
  "invoiceNumber": "String",
  "status": "String",
  "orderDate": "String",
  "paymentLinkDueDate": "String",
  "productList": [
    {
      "name": "String",
      "productId": "String",
      "quantity": "Integer",
      "rate": "Float",
      "discount": "Float",
      "tax": "Float",
      "amount": "Float"
    }
  ],
  "orderSummary": {
    "subTotal": "Float",
    "tax": "Float",
    "discount": "Float",
    "grandTotal": "Float"
  },
  "customerDetails": {
    "type": "String",
    "uuid": "String",
    "countryCode": "String",
    "msisdn": "String",
    "email": "String",
    "name": "String",
    "preferredLanguage": "String",
    "personalNumber": "String",
    "organizationId": "String",
    "address": {
      "street": "String",
      "zip": "String",
      "city": "String",
      "country": "String"
    }
  }
}
```

```
    }  
  },  
}  
}
```

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

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

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"  
}
```

Get Invoice Number By UUID

The **Get Invoice Number By UUID** endpoint enables your system to fetch the invoice number associated with a specific order, identified by its UUID. This can help in reconciling billing records, displaying invoice references to customers, or integrating with accounting systems.

Important: An invoice number is returned only when the order is an Invoiced order and in one of the following statuses:

- **Invoiced**
- **Inkasso**
- **Purring**
- **Paid**
- **Refunded**

If the order has not yet reached an invoiced-type status, this API will not return an invoice number.

You will find the endpoint path, authentication requirements, and example responses below.

EndPoint

```
GET https://demo-api.frontpayment.no/api/v1/connect/orders/invoice-number/{{ORDER_UUID}}
```

Authorization

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

Example:

```
Authorization: Bearer YOUR_FRONTPAYMENT_BEARER_TOKEN
```

Response

A successful request will return a `200` status with the following JSON payload:

```
{
  "status_code": 200,
  "status_message": "OK",
```

```
"message": "orderInvoiceNoRetrievedSuccessfully",
"is_data": true,
"data": {
  "orderId": "ODR2005869234",
  "invoiceNumber": "99900753",
  "status": "REFUNDED"
}
}
```

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

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

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"
}
```

Delayed Charge

With this API endpoint, you can charge from order using Order UUID.

Endpoint

```
POST https://demo-api.frontpayment.no/api/v1/connect/orders/delayed/charge/{{ORDER_UUID}}
```

Authorization

To access this endpoint, include a `Bearer Token` in the `Authorization` header of your request. You can obtain this token from Frontpayment.

Example Authorization Header: `Authorization: Bearer YOUR_FRONTPAID_BEARER_TOKEN`

Response

A successful request will return a `200 OK` status with the following JSON payload:

```
{
  'status_code' => 200,
  'status_message' => 'OK',
  'message' => 'orderDelayedChargedSuccessfully',
  'is_data' => true,
  'data' => null,
}
```

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

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

```
}
```

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

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