

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](#)

Revision #36

Created 26 March 2024 15:11:35 by Admin

Updated 19 March 2026 09:57:19 by Admin