

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

Revision #15

Created 10 January 2024 12:37:39 by Admin

Updated 8 September 2025 15:14:10 by Admin