## **Authorize.Net Refund Transaction**

In the case that a refund is required for a previously captured transaction, use the Refund Transaction method. This method will refund the transaction on Authorize.Net for either an existing card or dynamically to the card details provided then create an additional transaction on the existing payment on OrderCloud.io.

If the refund transaction is successful, but there is an error while creating the transaction on OrderCloud.io, the Authorize.Net transaction will be voided.

### Refund Transaction Request

This method requires either CardDetails.CreditCardID (for a previously created card) or CardDetails.CardNumber and CardDetails.ExpirationDate (to create a new card). It also requires CardDetails.PaymentID referencing the previously created payment being refunded.

POST https://api.ordercloud.io/v1/integrationproxy/authorizenet HTTP/1.1

Authorization: bearer insert\_access\_token\_here

Content-Type: application/json; charset=UTF-8

{

"BuyerID": "...",

“OrderID”: “...”,

“TransactionType”: “refundTransaction”,

“CardDetails”: {

“PaymentID”: “...”

“CreditCardID”: “...”,

“CardholderName”: “...”,

“CardType”: “...”,

“CardNumber”: “...”,

“ExpirationDate”: “...”,

“CardCode”: “...”

}

}

### Refund Transaction Response

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

{

"ChargeStatus": "...",

“CreditCardID”: “...”,

“PaymentID”: “...”,

“TransactionID”: “...”,

“Messages”: [

{

"code": "1",

"description": "..."

}

]

}

### Error Handling

During the refund process, the Authorize.Net refund transaction will be created first, followed by the transaction on OrderCloud.io. In the case that the OrderCloud.io step fails, the refund will be voided automatically on Authorize.Net.

Errors will return the exact response directly from the Authorize.Net or OrderCloud.io endpoint that failed. However, if any required fields are missing, a 400 error will be returned before any of the update process is executed.

#### Validation Responses

In the case that a required field is missing from your request, the following response will be returned containing a unique ErrorCode and Message as well as the request body sent during the call. The possible ErrorCodes and Messages are listed below.

HTTP/1.1 400 Bad Request

Content-Type: application/json

{

"ErrorCode": "...",

"Message": "...",

"Data": {

“...Request Body...”

}

}

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **Message** | **Status Code** |
| RefundTransaction.OrderIDRequired | OrderID is required to refund a transaction. | 400 |
| RefundTransaction.PaymentIDRequired | CardDetails.PaymentID is required to refund a transaction. | 400 |
| RefundTransaction.BuyerIDRequired | BuyerID is required to refund a transaction. | 400 |
| RefundTransaction.CardDetailsRequired | CardDetails.CreditCardID OR CardDetails.CardNumber and CardDetails.ExpirationDate are required to refund a transaction. | 400 |

#### OrderCloud.io Error Responses

If an incorrect BuyerID was provided:

HTTP/1.1 404 Not Found

Content-Type: application/json

{

"Errors": [

{

"ErrorCode": "NotFound",

"Message": "Buyer not found: 1234",

"Data": null

}

]

}

If an incorrect CardDetails.PaymentID was provided:

HTTP/1.1 404 Not Found

Content-Type: application/json

{

"Errors": [

{

"ErrorCode": "NotFound",

"Message": "Payment not found: 1234",

"Data": null

}

]

}

If an incorrect OrderID was provided:

HTTP/1.1 404 Not Found

Content-Type: application/json

{

"Errors": [

{

"ErrorCode": "NotFound",

"Message": "Order not found: 1234",

"Data": null

}

]

}