

Verify a payment method

In this guide, you will learn how to verify a payment method prior to use without holding funds from the consumer's account.

Before you begin

Payment verification is not available for all payment methods. Refer to the **Notes** column in the table of the [Payment methods overview](#) page for additional information.

Determine what you need to verify:

- Payment account validation — does the card exist and is it active?
- Card address (AVS) validation — does the address provided match?
- Card CVV validation — does the CVV provided match?
- Card email validation (American Express only) — does the email address provided match?
- Card phone number validation (American Express only) — does the phone number provided match?

Tip

Verification for ACH is available in the US. More information about card verifications including regions and applicable cards can be found in the [cards](#) payment method page.

Verify a card

Verifying a card validates a consumer's payment account and determines if the account number is in good standing.

The following example shows the required fields for a card verification request:

HTTP method: POST

Endpoint: /verifications

```
json
Copy
{
  "merchant": {
    "merchantSoftware": {
      "companyName": "Payment Company",
      "productName": "Application Name"
    }
  },
  "currency": "USD",
  "paymentMethodType": {
```

```
    "card": {
      "accountNumber": "4012000033330026",
      "expiry": {
        "month": "5",
        "year": "2027"
      }
    }
  }
}
```

Response:

```
json
Copy
{
  "transactionId": "722bb5fd-7c27-442e-a9da-3f8a3c15b468",
  "requestId": "fc90352e-ebef-4b19-8621-d439cf27cd07",
  "currency": "USD",
  "responseStatus": "SUCCESS",
  "responseCode": "APPROVED",
  "responseMessage": "Transaction approved by Issuer",
  "hostMessage": "Transaction accepted",
  "paymentMethodType": {
    "card": {
      "cardTypeName": "VISA",
      "cardType": "VI",
      "maskedAccountNumber": "401200XXXXXX0026",
      "cardTypeIndicators": {
        "issuanceCountryCode": "USA",
        "isLevel3Eligible": false,
        "isDurbinRegulated": false
      },
      "networkResponse": {
        "addressVerificationResult": "NOT_REQUESTED",
        "networkTransactionId": "013228692165455"
      }
    }
  }
}
```

```
},
"merchant": {
  "merchantId": "000017904371",
  "merchantSoftware": {
    "companyName": "Payment Company",
    "productName": "Application Name",
    "version": "1.235"
  },
  "merchantCategoryCode": "4899"
},
"transactionDate": "2023-08-16T23:30:43.033Z",
"initiatorType": "CARDHOLDER",
"accountOnFile": "NOT_STORED",
"hostReferenceId": "r8dnofa5dUdnWiwQuGYC",
"approvalCode": "tst262"
}
```

An HTTP status of 200 indicates a successful response. The following fields indicate result:

- responseStatus
- responseCode
- responseMessage

Tip

Refer to [Online payments response codes](#) for more information on these fields.

AVS details

To verify card AVS details, include the following fields in addition to the required fields above:

- accountHolder.billingAddress.line1
- accountHolder.billingAddress.line2
- accountHolder.billingAddress.city
- accountHolder.billingAddress.state
- accountHolder.billingAddress.postalCode
- accountHolder.billingAddress.countryCode

For AVS validation, the following fields indicate result:

- addressVerificationResult
- addressVerificationResultCode

CVV details

To verify card CVV details, include the following fields in addition to the required fields above:

- `paymentMethodType.card.cvv`

For CVV validation, the following fields indicate result:

- `cardVerificationResult`
- `cardVerificationResultCode`

Email details

To verify card **accountHolder** email details (American Express only), include the following fields in addition to the required fields above:

- `accountHolder.email`

For card email validation, the following fields indicate result:

- `emailVerificationResult`
- `emailVerificationResultCode`

Phone details

To verify card `accountHolder` phone details (American Express only), include the following fields in addition to the required fields above:

- `accountHolder.phone.countryCode`
- `accountHolder.phone.phoneNumber`

For card phone number validation, the following fields indicate result:

- `phoneVerificationResult`
- `phoneVerificationResultCode`

Retrieve verification details

After making a verification request, you can review the verification details. Complete the following steps to retrieve verification details:

Send a GET request to the `/verifications` endpoint one of two ways:

1. Use the `requestId` returned in the original response as a query parameter in a new GET request.
2. Use the `transactionId` returned in the original response as a path parameter in a new GET request.

Refund a payment

Send a refund to return captured funds to a consumer. Except for a standalone refund, the Online Payments API uses the payment method from a previous payment for your refund.

In this guide, learn how to issue the following types of refunds:

- Full refund
- Partial refund
- Standalone refund
- Multi-capture order refund

Note

We process refund authorizations on your behalf as part of a refund call.

Before you begin

Before you send a full or partial refund, obtain the `transactionId` of the original payment transaction.

To refund a partial amount:

- Determine that the amount of the refund is less than the amount of the original payment.
- The total amount of multiple partial refunds must not exceed the amount of the original payment.

To process a multi-capture order refund (for example, to refund one shipment from a split shipment order):

- The amount of the refund must be less than or equal to the amount of the capture (referred to by `captureId`), not the amount of the authorization.

Full refund

Refer to the following steps to refund the full amount of a previous transaction:

1. Reference the original payment transaction.
2. Send the original `transactionId` value in the `paymentMethodType.transactionReference.transactionReferenceId` field.
3. Omit the amount.

The following is an example of a full refund request:

HTTP method: POST

Endpoint: `/refunds`

json
Copy

```

{
  "paymentMethodType": {
    "transactionReference": {
      "transactionReferenceId": "624e78df-a159-4f26-9d20-1f81841e15e7"
    }
  },
  "merchant": {
    "merchantSoftware": {
      "companyName": "J.P. Morgan Chase",
      "productName": "Application Name",
      "version": "1.235"
    }
  }
}

```

Confirm a successful response with an HTTP 200 status. The following fields indicate the result:

- responseStatus
- responseCode
- responseMessage

Tip

For values and the description of fields responseStatus, responseCode, and responseMessage, refer to [Online Payments response codes](#).

Response:

```

json
Copy
{
  "transactionId": "948a288a-fe83-43ed-99eb-c1ca7afa1b3e",
  "requestId": "974c5764-e851-442b-8074-fd3d9c8ba08d",
  "transactionState": "AUTHORIZED",
  "amount": 1234,
  "currency": "USD",
  "responseStatus": "SUCCESS",
  "responseCode": "APPROVED",
  "responseMessage": "Transaction approved by Issuer",
  "transactionReferenceId": "624e78df-a159-4f26-9d20-1f81841e15e7",
  "remainingRefundableAmount": 0,

```

```
"approvalCode": "tst862",
"hostMessage": "Approved",
"initiatorType": "CARDHOLDER",
"accountOnFile": "NOT_STORED",
"transactionDate": "2024-07-31T14:21:36.253Z",
"merchant": {
  "merchantId": "998482157632",
  "merchantSoftware": {
    "companyName": "J.P. Morgan Chase",
    "productName": "Application Name",
    "version": "1.235"
  },
  "merchantCategoryCode": "4899"
},
"paymentMethodType": {
  "card": {
    "cardTypeName": "VISA",
    "cardType": "VI",
    "maskedAccountNumber": "411234XXXXXX4113",
    "cardTypeIndicators": {
      "issuanceCountryCode": "USA",
      "isLevel2Eligible": false,
      "isLevel3Eligible": false,
      "isDurbinRegulated": false,
      "cardProductTypes": [
        "COMMERCIAL",
        "PINLESS_DEBIT"
      ]
    }
  },
  "networkResponse": {
    "addressVerificationResult": "NOT_REQUESTED",
    "additionalData": {
      "electronicCommerceIndicator": "7",
      "marketSpecificData": "B",
      "productId": "F",
```

```
        "posEntryMode": "01",
        "returnAci": "T",
        "validationCode": "CELO"
    },
    "networkTransactionId": "014213692168435"
},
"isBillPayment": true
}
},
"hostReferenceId": "QZH7049D0Nma9Mg2jyHDX5",
"paymentRequest": {
    "paymentRequestId": "624e78df-a159-4f26-9d20-1f81841e15e7",
    "paymentRequestStatus": "CLOSED",
    "authorizations": [
        {
            "authorizationId": "624e78df-a159-4f26-9d20-1f81841e15e7",
            "amount": 1234,
            "transactionStatusCode": "CAPTURED",
            "authorizationType": "INITIAL"
        }
    ],
    "captures": [
        {
            "captureId": "624e78df-a159-4f26-9d20-1f81841e15e7",
            "amount": 1234,
            "transactionStatusCode": "CLOSED",
            "captureRemainingRefundableAmount": 0
        }
    ],
    "refunds": [
        {
            "refundId": "948a288a-fe83-43ed-99eb-c1ca7afa1b3e",
            "amount": 1234,
            "transactionStatusCode": "CLOSED"
        }
    ]
}
```



```
]
}
}
```

Partial refund

Refund a portion of the original transaction amount to a consumer. Refer to the following steps to refund a partial amount:

1. Send the original transactionId value in the paymentMethodType.transactionReference.transactionReferenceId field.
2. Include the amount to refund.

Note

The amount of the refund must be less than the amount of the original payment. Process multiple partial refunds as long as you do not refund more, in total, than the amount of the original payment.

The following is an example of a partial refund request:

HTTP method: POST

Endpoint: /refunds

```
JSON
Copy
{
  "amount": 900,
  "currency": "USD",
  "paymentMethodType": {
    "transactionReference": {
      "transactionReferenceId": "2150955d-47f7-4770-b6b7-6a8d0cbc114f"
    }
  },
  "merchant": {
    "merchantSoftware": {
      "companyName": "Payment Company",
      "productName": "Application Name",
      "version": 1.235
    }
  }
}
```

Response:

json
Copy

```
{
  "transactionId": "2150955d-47f7-4770-b6b7-6a8d0cbc114f",
  "requestId": "9f4068e9-81b2-40aa-80fc-010e96526b21",
  "transactionState": "AUTHORIZED",
  "amount": 900,
  "currency": "USD",
  "responseStatus": "SUCCESS",
  "responseCode": "APPROVED",
  "responseMessage": "Transaction approved by Issuer",
  "transactionReferenceId": "12b2f378-40c8-425d-be1c-71971c21104d",
  "remainingRefundableAmount": 3000,
  "approvalCode": "tst698",
  "hostMessage": "Approved",
  "initiatorType": "CARDHOLDER",
  "accountOnFile": "NOT_STORED",
  "transactionDate": "2025-03-11T20:34:28.278Z",
  "merchant": {
    "merchantId": "000017904374",
    "merchantSoftware": {
      "companyName": "Payment Company",
      "productName": "Application Name",
      "version": "1.235"
    },
    "merchantCategoryCode": "4899"
  },
  "paymentMethodType": {
    "card": {
      "cardTypeName": "VISA",
      "cardType": "VI",
      "maskedAccountNumber": "411234XXXXXX4113",
      "networkResponse": {
        "addressVerificationResult": "NOT_REQUESTED",
        "additionalData": {
          "electronicCommerceIndicator": "7"
        }
      }
    }
  }
}
```

```
    }
  },
  "isBillPayment": true
}
},
"hostReferenceId": "t79zEVoiZ3ZLMZi3Aftym5",
"paymentRequest": {
  "paymentRequestId": "12b2f378-40c8-425d-be1c-71971c21104d",
  "paymentRequestStatus": "CLOSED",
  "authorizations": [
    {
      "authorizationId": "12b2f378-40c8-425d-be1c-71971c21104d",
      "amount": 3900,
      "transactionStatusCode": "CAPTURED",
      "authorizationType": "INITIAL"
    }
  ],
  "captures": [
    {
      "captureId": "12b2f378-40c8-425d-be1c-71971c21104d",
      "amount": 3900,
      "transactionStatusCode": "CLOSED",
      "captureRemainingRefundableAmount": 3000
    }
  ],
  "refunds": [
    {
      "refundId": "2150955d-47f7-4770-b6b7-6a8d0cbc114f",
      "amount": 900,
      "transactionStatusCode": "CLOSED"
    }
  ]
}
}
```

Standalone refund

A standalone refund processes independently from your previous transactions and can be processed in real time. The refund applies directly to the consumer's payment method.
Attention

Process standalone refunds for card payments only.

The following is an example of a standalone refund request:

HTTP method: POST

Endpoint: /refunds

```
json
Copy
{
  "merchant": {
    "merchantSoftware": {
      "companyName": "Payment Company",
      "productName": "Application Name",
      "version": "1.235"
    }
  },
  "amount": 1234,
  "currency": "USD",
  "initiatorType": "CARDHOLDER",
  "accountOnFile": "NOT_STORED",
  "paymentMethodType": {
    "card": {
      "accountNumber": "4012000033330026",
      "expiry": {
        "month": 5,
        "year": 2027
      }
    }
  }
}
```

Response:

```
json
Copy
```

```
{
  "transactionId": "94d1dd5b-2c81-4896-a159-c726af81883a",
  "requestId": "e8cc1a4a-3869-420f-866c-02c53a370182",
  "transactionState": "AUTHORIZED",
  "amount": 1234,
  "currency": "USD",
  "responseStatus": "SUCCESS",
  "responseCode": "APPROVED",
  "responseMessage": "Transaction approved by Issuer",
  "approvalCode": "tst825",
  "hostMessage": "Approved",
  "initiatorType": "CARDHOLDER",
  "accountOnFile": "NOT_STORED",
  "transactionDate": "2024-07-31T14:16:04.869Z",
  "merchant": {
    "merchantId": "000017904371",
    "merchantSoftware": {
      "companyName": "Payment Company",
      "productName": "Application Name",
      "version": "1.235"
    },
    "merchantCategoryCode": "4899"
  },
  "paymentMethodType": {
    "card": {
      "cardTypeName": "VISA",
      "cardType": "VI",
      "maskedAccountNumber": "401200XXXXXX0026",
      "cardTypeIndicators": {
        "issuanceCountryCode": "USA",
        "isLevel2Eligible": false,
        "isLevel3Eligible": false,
        "isDurbinRegulated": false,
        "cardProductTypes": [
          "COMMERCIAL",

```

```
        "PINLESS_DEBIT"
      ]
    },
    "networkResponse": {
      "addressVerificationResult": "NOT_REQUESTED",
      "additionalData": {
        "electronicCommerceIndicator": "7",
        "productId": "F",
        "posEntryMode": "01",
        "returnAci": "T",
        "validationCode": "V7J0"
      },
      "networkTransactionId": "014213692168401"
    }
  }
},
"hostReferenceId": "S6JU1UmjVPOQFCmUjthEF7",
"paymentRequest": {
  "paymentRequestId": "94d1dd5b-2c81-4896-a159-c726af81883a",
  "paymentRequestStatus": "CLOSED",
  "refunds": [
    {
      "refundId": "94d1dd5b-2c81-4896-a159-c726af81883a",
      "amount": 1234,
      "transactionStatusCode": "CLOSED"
    }
  ]
}
}
```

Multi-capture refund

To refund a multi-capture order (for example, a split shipment), create a separate refund to the /refunds endpoint for each captureId of the previous payment. Refer to the following steps for a multi-capture refund:

1. Send the captureId value in the paymentMethodType.transactionReference.transactionReferenceId field.
2. The amount of the refund must be less than or equal to the amount of the capture.

The following is an example of a multi-capture refund request:

HTTP method: POST

Endpoint: /refunds

```
json
Copy
{
  "paymentMethodType": {
    "transactionReference": {
      "transactionReferenceId": "cf0bd1da-0cff-4bf0-a295-d1310c46c513"
    }
  },
  "merchant": {
    "merchantSoftware": {
      "companyName": "Payment Company",
      "productName": "Application Name",
      "version": 1.235
    }
  }
}
```

Response:

```
json
Copy
{
  "transactionId": "c3bf3975-aec4-4ca4-bf86-c05a7ce94150",
  "requestId": "7e0e0d62-6107-4b29-8582-276e74f7362e",
  "transactionState": "AUTHORIZED",
  "amount": 100,
  "currency": "USD",
  "responseStatus": "SUCCESS",
  "responseCode": "APPROVED",
  "responseMessage": "Transaction approved by Issuer",
  "transactionReferenceId": "cf0bd1da-0cff-4bf0-a295-d1310c46c513",
  "remainingRefundableAmount": 0,
}
```

```
"approvalCode": "tst688",
"hostMessage": "Approved",
"initiatorType": "CARDHOLDER",
"accountOnFile": "NOT_STORED",
"transactionDate": "2023-08-21T20:33:46.507Z",
"merchant": {
  "merchantId": "000017904371",
  "merchantSoftware": {
    "companyName": "Payment Company",
    "productName": "Application Name",
    "version": "1.235"
  },
  "merchantCategoryCode": "4899"
},
"paymentMethodType": {
  "card": {
    "cardTypeName": "VISA",
    "cardType": "VI",
    "maskedAccountNumber": "411234XXXXXX4113",
    "cardTypeIndicators": {
      "issuanceCountryCode": "USA",
      "isDurbinRegulated": false,
      "cardProductTypes": [
        "COMMERCIAL",
        "PINLESS_DEBIT"
      ]
    },
    "networkResponse": {
      "addressVerificationResult": "NOT_REQUESTED"
    },
    "isBillPayment": true
  }
},
"hostReferenceId": "Vx8VBdcN0XOgmzRuN4tVq2",
"paymentRequest": {
```



```
"paymentRequestId": "cf0bd1da-0cff-4bf0-a295-d1310c46c513",
"paymentRequestStatus": "CLOSED",
"authorizations": [
  {
    "authorizationId": "cf0bd1da-0cff-4bf0-a295-d1310c46c513",
    "amount": 1234,
    "transactionStatusCode": "CAPTURED",
    "authorizationType": "INITIAL"
  }
],
"captures": [
  {
    "captureId": "cf0bd1da-0cff-4bf0-a295-d1310c46c513",
    "amount": 100,
    "transactionStatusCode": "CLOSED",
    "captureRemainingRefundableAmount": 0
  },
  {
    "captureId": "9883af20-b3b9-4b39-9b6e-6cce1b8790a0",
    "amount": 100,
    "transactionStatusCode": "CLOSED",
    "captureRemainingRefundableAmount": 100
  },
  {
    "captureId": "d9921bb1-ee4e-4038-bf8b-a0005fe7aff1",
    "amount": 100,
    "transactionStatusCode": "CLOSED",
    "captureRemainingRefundableAmount": 100
  }
],
"refunds": [
  {
    "refundId": "c3bf3975-aec4-4ca4-bf86-c05a7ce94150",
    "amount": 100,
    "transactionStatusCode": "CLOSED"
  }
]
```

```
}  
  ]  
}  
}
```

Retrieve refund details

After you process a refund, refer to the following steps to obtain the refund details:

1. Send a GET request to the /refunds endpoint in one of two ways:
 1. Use the requestId from the original response as a query parameter in a new GET request.
 2. Use the transactionId from the original response as a path parameter in a new GET request.