

Oracle In Person Payments Integration Guide

December, 2025, Version 1.1
Copyright © 2026, Oracle and/or its affiliates
Public



Purpose statement

This document provides an overview of features and enhancements included in release Oracle Payment Cloud Service - In Person Payments Terminal APIs.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Document Revision History

VERSION	DATE	DESCRIPTION OF CHANGE
1.0	2024-12-15	Initial version
1.1	2025-12-02	Introducing api version 2 of oracle payments terminal.

Table of contents

What is in person payments	6
How it works	7
Payment Methods	7
Terminal Serial Number or Workstation Identifier?	8
Step 1: Generate the Client Credentials Key	9
Authentication	9
OAuth 2.0 – Client Credentials Grant	9
Token Endpoint:	9
HTTP Headers:	9
Request Body:	9
Generating the Basic Authorization Header	9
Step 2: Implement in your point of sale required Terminal APIs	10
Step 3: Make a payment using Transaction Type 004	20
Example Transaction 004 API :	20
Step 4: Get the Payment Outcome	21
Transaction Receipt Object	22
Possible result codes that can be received in the result object:	28
Declined Payment Testing	29
Transaction Event Notifications	31
Webhook Subscription Configuration:	32
Webhook event parameters	33
Step 5: Test and Go Live	35
Oracle Payment Cloud Service Terminal APIs	36
Authorization Card Present (001)	37
Authorization Card Not Present (002)	39
Capture Authorization (003)	41
Authorization and Capture Card Present (004)	42
Authorization and Capture Card Not Present (005)	44
Full Refund / Cancellation:	46
Partial Refund (007)	47
Generate Token Card Present (008)	48
Generate Token Card Not Present (009)	50
Make Token Payment (010)	52
Oracle Payment Cloud Service – Webhook Specification	54
AuthorizaHon Webhook	54
AuthorizaHon Adjustment Webhook	55
Token Creation Webhook	55
Cancellation Webhook	56
Refund Webhook	57
Refund Failure Webhook	57

List of figures

Figure 1. Verifone P630,	6
--------------------------	---

List of tables

Table 1: Header Parameters	10
Table 2. Request Parameters	10
Table 3. Response Parametres	12
Table 4. Transaction Receipts Table	15
Table 5. Error Codes`	28
Table 6. Declined payment simulation	29
Table 7. webhook event parameters	33
Table 8. Delta changes	36

What is in person payments

Oracle Payment Cloud Service (OPCS) In Person Payments allows transactions to be processed via a payment terminal integrating with a point-of-sale solution. Several OPCS Terminal APIs are available to support different transaction types.

Figure 1. Verifone P630,



How it works

There are 10 OPCS Terminal APIs available:

1. Authorisation Card Present
 - a. Used in conjunction with '3'. This allows for an amount to be authorised for the presented card, but the transaction is not fulfilled so no money will be moved.
2. Authorisation Card Not Present
 - a. As described for '1' above but the card details are manually keyed in.
3. Capture Authorised Payment
 - a. Used in conjunction with '1' and '2' to fulfil the transaction that was previously authorised.
4. Authorisation and Payment Card Present
 - a. A complete transaction where the amount is authorised and then fulfilled in a single request.
5. Authorisation and Payment Card Not Present
 - a. As '4' above but the card details are manually keyed in.
6. Full Refund or Cancel
 - a. Depending on whether the transaction amount has been captured (money has moved from the issuer's account) this transaction will result in either a full refund (where it has been captured) or a cancel if not captured.
7. Partial Refund
 - a. Used to refund back to the shopper an amount less than the original transaction amount. Multiple partial refunds can be performed against a single transaction amount.
8. Generate Token Card Present
 - a. Used to validate the card details at zero amount or greater than zero and receive a token to make future payments in conjunction with '10'
9. Generate Toke Card Not Present
 - a. As described for '8' above but the card details are manually keyed in.
10. Make Token Payment
 - a. Using a previously generated token a payment can be made with the need of the original card for a specific shopper.

The point-of-sale application instigates the applicable API transaction request to Oracle Payment Cloud Service where it is validated and then processed resulting in the terminal 'lighting up' ready for the guest/shopper to present their card or key their card details in. The card details are verified, and response returned to the terminal and point of sale.

Payment Methods

The following payment methods are supported:

- American Express
- Union Pay
- Diner
- Discover
- JCB
- Mastercard
- Visa

Before you begin

The following requirements must be fulfilled before integrating with Oracle Payment Cloud Service In Person Payments:

1. General Requirements

- a. Oracle must be the designated Payment Provider.
- b. An Oracle Account Representative must be contacted to enable in person payment configurations. The organization must be onboarded to Oracle Payment Cloud Service (OPCS) and have access to a **test environment**.

2. Developer Setup Requirements

- a. The integrator must be registered as a Service Provider in OPCS (to enable channel creation).
- b. The integrator must provide:
 - i. Email addresses for users requiring OPCS credentials
 - ii. Webhook endpoint and HMAC key to receive payment events
- c. Oracle will share the following:
 - i. **Channel Code** (defines transaction handling attributes)
 - ii. **Location reference(s)** (identifiers for customer locations. It would be used for making token payments or processing a sale transaction for a different location to where the terminal is registered.)
 - iii. **Organization reference(s)** (identifiers for customer organization. It would be used for making token payments.)

3. Technical Requirements

- a. Test Terminal
- b. Test cards

Terminal Serial Number or Workstation Identifier?

To identify which Terminal to 'light' up to process a transaction, the integrating system must either pass: the unique Terminal identifier or another unique identifier (e.g. Media Access Control (MAC) Address) for the Workstation/POS device (Workstation) that started the transaction. This is a unique value made up by the model's name and its serial number, e.g. P400Plus-123456789. A Workstation/POS Device Identifier must be unique within OPCS.

For both cases, the terminal must initially be registered to a location. This process is completed when an Oracle customer onboards with the OPCS service. A single terminal can be shared with multiple workstations if required.

Where an identifier is of the POS Device this can be mapped as part of the terminal registration process.

Step 1: Generate the Client Credentials Key

Authentication

OAuth 2.0 – Client Credentials Grant

To authenticate API calls, use the OAuth 2.0 **Client Credentials** flow. Your service must obtain an access token from Oracle Identity Cloud Service (IDCS) using the **Client ID** and **Client Secret** provided for your environment.

Token Endpoint:

POST <https://idcs-0e86f01b7f094720af224335e43f84eb.identity.oraclecloud.com/oauth2/v1/token>

HTTP Headers:

```
Authorization: Basic <base64(client_id:client_secret)>
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
```

Request Body:

Use the appropriate environment code in the scope:

```
grant_type=client_credentials&scope=OPOP_<environment>/opop.payments
```

Generating the Basic Authorization Header

Before calling the token endpoint, you must generate the Base64-encoded value for the Authorization: Basic header

1): Concatenate the Client ID and Client Secret

Format:

```
clientId + ":" + clientSecret
```

Example:

```
12345:ABCDE
```

2): Base64-encode the resulting string

Command:

```
echo -n '12345:ABCDE' | base64
```

Output:

```
MTIzNDpBQkNERQ==
```

3): Use the encoded value in the Authorization header

Final header format:

```
Authorization: Basic MTIzNDpBQkNERQ==
```

Postman Script (Optional):

To automatically extract the token and store it in Postman collection variables:

```
const response = pm.response.json();
const access = response.access_token;
pm.collectionVariables.set("access_token", access);
```

Step 2: Implement in your point of sale required Terminal APIs

The Terminal API's have some required parameters that vary depending on the transaction type. These are summarised here.

Table 1: Header Parameters

Parameter Name	Required?	Description
AuthToken	Y	A base64 encoded combination of your clientId and clientSecret

Table 2. Request Parameters

Parameter Name	Required?	Data Type	Description
transactionType	Y	Number	The type of transaction being performed indicated in Oracle Payment Cloud Service Terminal APIs
terminalID	Y	String	The unique ID of the terminal to send this request to. Format: <i>[device model]-[serial number]</i>
workstationId	Y if terminalID not sent	String	Unique Identifier in the organization (e.g. MAC Address) of the POS device that made the request.
transactionDetails.amount	Y	Number	Transaction amount, to 2 decimal places. This value must be passed into the script on initialization, so that same value can be used here.
transactionDetails.currency	Y	String	Transaction Currency, 3-character ISO Code
locationReference	Y if processing transaction against a location that is not the one the terminal is registered too or for 010	String	Organization's own reference for the location. This is captured in Oracle Payment Cloud Service during onboarding when locations are added via API or the UI.
channelCode	Y	String	Channel code associated with the channel that was created for in person payments.
transactionReferences.clientTransactionId	Y	String	Integrating the system's reference to identify a transaction. Use a unique value per transaction. This will show as the POS Transaction Reference (also known as Merchant Reference) for the transaction in OPCS reports. Maximum 40 alphanumeric characters.

transactionReferences. originalOpcsTransactionId	Y, where transactionType = 003, 006, 007	String	OPCS's reference for the transaction that is being modified, e.g., manually captured, refunded partially, or in full. This will be the opcsTransactionId returned when the original transaction was requested.
transactionReferences. requestDateTime	Y	String	UTC Date and Time of the request for the transaction
shopperReference	Yes, where transactionType = 008, 009, 010	String	Unique reference defined by the publisher / service provider for the shopper. Min 3, Max 256
opcsToken	Yes, where transactionType = 010	String	Unique reference defined by the publisher / service provider for the shopper. Min 3, Max 256
recurringProcessingModel	Yes, where transactionType = 008	String	Always UnscheduleWdCardOnFile for recurring payment types.
organizationReference	Yes, where transactionType = 010	String	Organization's own reference. This is captured in Oracle Payment Cloud Service during onboarding when organizations are added via API or the UI.

Table 3. Response Parametres

Parameter Name	Parameter Name	Parameter Name	Data Type	Description
transactionType	-	-	Number	The type of transaction being performed indicated in Oracle Payment Cloud Service Terminal APIs
opcsToken	-	-	String	the token for the card received when performing transactionType = 008 or 009.
recurringProcessing Model	-	-	String	Always UnscheduledCardOnFile for recurring payment types.
shopperReference	-	-		Unique reference for the shopper that was sent in the request.
terminalId	-	-		The unique ID of the terminal sent in the request. Format: [device model]-[serial number].
workstationId	-	-		Unique identifier in the organization (e.g. MAC address) of the POS device making the request. Can only be used where a workstation is mapped to a single terminal. Mapping is performed via OPCS.
transactionDetails	-	-	Object	Object containing details of the transaction.
	amount	-	Number	Final transaction amount requested, in major and minor units.
	currency	-	String	Transaction currency (3-character ISO code).
transactionReferences	-	-	Object	Object containing all transaction reference identifiers.
	originalClient TransactionId	-	String	Publisher's unique identifier of a transaction being modified (used for modification types: 003, 006, 007).
	clientTransactionId	-	String	Unique identifier for the transaction sent in the request.

	originalOpcsTransactionId	-	String	OPCS unique identifier of a transaction being modified (003, 006, 007).
	opcsTransactionId	-	String	OPCS unique reference for the transaction request.
	originalPspReference	-	String	Payment processor's unique identifier of a modified transaction (003, 006, 007).
	pspReference	-	String	Payment processor's unique reference for the transaction.
	requestDateTime	-	String	UTC date and time when the transaction request was made.
	responseDateTime	-	String	UTC date and time of the transaction response.
paymentLocationId	-	-	String	Unique identifier for the location the transaction is associated with.
locationReference	-	-	String	Organization's internal reference for the location (if sent in the request).
operatorId	-	-	String	Identifier for the operator performing the transaction (if provided).
invoiceId	-	-	String	Reference used by the integrating system for the itemized list of purchases (if provided).
channelCode	-	-	String	Reference for the channel sent in the request.
approvalCode	-	-	String	Code assigned to a transaction approval by the acquirer.
rawAcquirerResult	-	-	Object	Object containing the raw acquirer response in an authorization response.
	avsResultRaw	-	String	Raw AVS (Address Verification System) result.
	cvcResultRaw	-	String	Raw CVC result.
	refusalReasonRaw	-	String	Raw refusal reason text.
	acquirerCode	-	String	Acquirer code or name.
	acquirerReference	-	String	Reference assigned by the acquirer.

result		-	Object	Object containing the result details.
	resultCode	-	String	
	resultReason	-	String	
	refusalReason	-	String	Decline reason (only shown for declined payments).
	authorizedAmount	-	String	Amount that has been authorized.
	currency	-	String	Currency of the authorized amount.
paymentMethodData		-	Object	
	cardData		Object	
		cardScheme Commercial	String	Indicates if the card is commercial.
		cardPaymentMethod	String	Payment method used.
		cardIssuingBank	String	Issuing bank name.
		cardIssuingCountry	String	Issuing country.
		cardIssuingCurrency	String	Issuing currency.
		cardBin	String	First 6 digits of the card number.
		issuerBin	String	First 8 digits of the card number (for 16+ digit cards).
		fundingSource	String	Funding source (e.g., CREDIT, DEBIT).
		cardSummary	String	Last 4 digits of the card number.
		expiryDate	String	Card expiry date.
		paymentMethodVariant	String	Payment method variant.
		coBrandedWith	String	Co-branded card information.
		entryMode	String	List of entry modes.
		paymentMethodType	String	Type of payment method (Card, Check, Mobile, etc.).
additionalMetaData			Object	Object containing additional metadata.
	inferredRefusalReason		String	Inferred refusal reason.
	threeDAuthenticatedResponse		String	Raw 3D secure authentication result.

shopperData			Object	Object containing shopper information.
	countryCode		String	Shopper country.
	cardHolderName		String	Card Holder name.
	transactionReceipt		Object	See <i>Transaction Receipt</i> .

Table 4. Transaction Receipts Table

1	2	3	4	Type	Description
Transaction Receipts				Object	
	documentQualifier			Object	Qualification of the document to print to the Cashier or the Customer. <ul style="list-style-type: none"> • SaleReceipt • CashierReceipt • CustomerReceipt • Document • Voucher • Journal
	requiredSignatureFlag			Object	
	outputContent			Object	Content to display or print.
		outputFormat		Enumeration	Format of the content to display or print. <ul style="list-style-type: none"> • MessageRef • Text • XHTML • BarCode
		outputText		Object	Content of text message to display or print
			Character Style	Enumeration	Character Style <ul style="list-style-type: none"> • Normal • Bold • Italic • Underline

			endOfLine Flag	Boolean	
			text	String	Header1
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Header 2
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Merchant Copy
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	filler
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Date
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Time
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	<Filler>
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Card

			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	PAN Sequence
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Pref. Name
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Card Type
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Payment Method
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Payment Variant
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Entry Mode
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	<Filler>
			Character Style	Enumeration	
			endOfLine Flag	Boolean	

			text	String	AID
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	MID
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	TID
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	PTID
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	<Filler>
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Auth. Code
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Tender Reference
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			Text	String	Reference
			Character Style	Enumeration	
			endOfLine Flag	Boolean	

			text	String	<Filler>
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	Type
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	TOTAL
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	APPROVED
			Character Style	Enumeration	
			endOfLine Flag	Boolean	
			text	String	<Filler>

Step 3: Make a payment using Transaction Type 004

When this transaction type is instigated, the API will expect several values to be passed to it to complete the transaction, which are detailed below. Once submitted, the request will be validated and, if successfully, the terminal will display the amount to pay and wait for the card to be presented. The card details will then be checked and if authorised the payment will be completed and a response returned.

Example Transaction 004 API :

```
{
  "transactionType": "004",
  "terminalId": "P400Plus-123456789",
  "transactionDetails": {
    "amount": 1000,
    "currency": "USD"
  },
  "transactionReference": {
    "clientTransactionId": "Test_123456",
    "requestDateTime" : "2023-09-01T09:00:00.000Z"
  },
  "channelCode": "Test_123456"
}
```

Step 4: Get the Payment Outcome

When the payment completes, an API response will be passed to the merchant client. This response will contain transaction details, the result, and other additional details passed from the Acquirer. An example response can be found below:

```
{
  "approvalCode": "123456",
  "channelCode": "ch123456",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "entryModes": [
        "CONTACTLESS"
      ],
      "expiryDate": "2/2028",
      "fundingSource": "CREDIT",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "CARD"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "refusalReasonRaw": "APPROVED"
  },
  "result": {
    "authorizedAmount": 2000,
    "currency": "USD",
    "resultCode": "SUCCESS"
  },
  "shopperData": {
    "cardHolderName": " /",
    "countryCode": "US"
  },
  "terminalId": "P400Plus-123456789",
  "transactionDetails": {
    "amount": 2000,
    "currency": "USD"
  },
  "transactionReceipts": [OBJECT EXAMPLE: Transaction Receipt Object, 22]
  },
  "requiredSignatureFlag": false
},
"transactionReferences": {
  "clientTransactionId": "123456",
  "opcsTransactionId": "ABC1234DEF",
  "pspReference": "ZVR2DNVWNV5FKQV5",
  "requestDateTime": "2024-08-02T14:54:16.296651Z",
  "responseDateTime": "2025-03-17T12:28:10.837901751Z"
},
"transactionType": "004"
}
```

Transaction Receipt Object

Within the response a transaction receipt object is present for customer and merchant

```

"transactionReceipts": [
  {
    "documentQualifier": "CASHIER_RECEIPT",
    "outputContent": {
      "outputFormat": "TEXT",
      "outputText": [
        {
          "text": "key=header1",
          "characterStyle": "BOLD",
          "endOfLineFlag": true
        },
        {
          "text": "key=header2",
          "characterStyle": "BOLD",
          "endOfLineFlag": true
        },
        {
          "text": "name=MERCHANT%20COPY&key=merchantTitle",
          "characterStyle": "BOLD",
          "endOfLineFlag": true
        },
        {
          "text": "key=filler",
          "characterStyle": "NORMAL",
          "endOfLineFlag": true
        },
        {
          "text": "name=Date&value=01%2f17%2f2024&key=txdate",
          "characterStyle": "NORMAL",
          "endOfLineFlag": true
        },
        {
          "text": "name=Time&value=02%3a28%3a08&key=txtime",
          "characterStyle": "NORMAL",
          "endOfLineFlag": true
        },
        {
          "text": "key=filler",
          "characterStyle": "NORMAL",
          "endOfLineFlag": true
        },
        {
          "text": "name=Card&value=%2a%2a%2a%2a9999&key=pan",
          "characterStyle": "NORMAL",
          "endOfLineFlag": true
        },
        {
          "text": "name=PAN%20seq.&value=33&key=panSeq",
          "characterStyle": "NORMAL",
          "endOfLineFlag": true
        },
        {
          "text": "name=Pref.%20name&value=mc%20en%20gbr%20gbp&key=preferredName",
          "characterStyle": "NORMAL",
          "endOfLineFlag": true
        }
      ]
    }
  }
]

```

```

    "text": "name=Card%20type&value=mc&key=cardType",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Payment%20method&value=mc&key=paymentMethod",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Payment%20variant&value=mc&key=paymentMethodVariant",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Entry%20mode&value=Contactless%20chip&key=posEntryMode",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=AID&value=A000000004101001&key=aid",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=MID&value=50&key=mid",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=TID&value=e285p-805455133&key=tid",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=PTID&value=66642984&key=ptid",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Auth.%20code&value=123456&key=authCode",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Tender&value=Vcsq001705487288008&key=txRef",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Reference&value=123456&key=mref",
    "characterStyle": "NORMAL",

```

```

        "endOfLineFlag": true
    },
    {
        "text": "key=filler",
        "characterStyle": "NORMAL",
        "endOfLineFlag": true
    },
    {
        "text": "name=Type&value=GOODS_SERVICES&key=txtype",
        "characterStyle": "NORMAL",
        "endOfLineFlag": true
    },
    {
        "text": "name=TOTAL&value=%24%2010.00&key=totalAmount",
        "characterStyle": "BOLD",
        "endOfLineFlag": true
    },
    {
        "text": "key=filler",
        "characterStyle": "NORMAL",
        "endOfLineFlag": true
    },
    {
        "text": "name=APPROVED&key=approved",
        "characterStyle": "BOLD",
        "endOfLineFlag": true
    },
    {
        "text": "key=filler",
        "characterStyle": "NORMAL",
        "endOfLineFlag": true
    }
    ]
},
"requiredSignatureFlag": false
},
{
"documentQualifier": "CUSTOMER_RECEIPT",
"outputContent": {
    "outputFormat": "TEXT",
    "outputText": [
        {
            "text": "key=header1",
            "characterStyle": "BOLD",
            "endOfLineFlag": true
        },
        {
            "text": "key=header2",
            "characterStyle": "BOLD",
            "endOfLineFlag": true
        },
        {
            "text": "name=CARDHOLDER%20COPY&key=cardholderHeader",
            "characterStyle": "BOLD",
            "endOfLineFlag": true
        },
        {
            "text": "key=filler",
            "characterStyle": "NORMAL",
            "endOfLineFlag": true
        },
        {

```

```

    "text": "name=Date&value=01%2f17%2f2024&key=txdate",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Time&value=02%3a28%3a08&key=txttime",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Card&value=%2a%2a%2a%2a9999&key=pan",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=PAN%20seq.&value=33&key=panSeq",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Pref.%20name&value=mc%20en%20gbr%20gbp&key=preferredName",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Card%20type&value=mc&key=cardType",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Payment%20method&value=mc&key=paymentMethod",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Payment%20variant&value=mc&key=paymentMethodVariant",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Entry%20mode&value=Contactless%20chip&key=posEntryMode",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=AID&value=A000000004101001&key=aid",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=MID&value=50&key=mid",
    "characterStyle": "NORMAL",

```

```

    "endOfLineFlag": true
  },
  {
    "text": "name=TID&value=e285p-805455133&key=tid",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=PTID&value=66642984&key=ptid",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Auth.%20code&value=123456&key=authCode",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Tender&value=Vcsq001705487288008&key=txRef",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Reference&value=123456&key=mref",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=Type&value=GOODS_SERVICES&key=txtype",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=TOTAL&value=%24%2010.00&key=totalAmount",
    "characterStyle": "BOLD",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },
  {
    "text": "name=APPROVED&key=approved",
    "characterStyle": "BOLD",
    "endOfLineFlag": true
  },
  {
    "text": "key=filler",
    "characterStyle": "NORMAL",
    "endOfLineFlag": true
  },

```

```
{
  "text": "name=Retain%20for%20your%20records&key=retain",
  "characterStyle": "NORMAL",
  "endOfLineFlag": true
},
{
  "text": "name=Thank%20you&key=thanks",
  "characterStyle": "NORMAL",
  "endOfLineFlag": true
}
]
},
"requiredSignatureFlag": false
```

Possible result codes that can be received in the result object:

Oracle Payments may return one of the following result codes in the result code field:

- **Success:** Payment completed successfully
- **Failure:** Payment failed due to configuration, authentication, or provider-level issues
- **Partial:** Payment was only partially authorized
- **Received:** The request is Received for transaction 006 and 007, status of the request is provided through webhook

Table 5. Error Codes`

Error Code	Error Category	Possible Reasons
00000- <PAYMENT_PROVIDER_CODE>	UNKNOWN	<ul style="list-style-type: none"> • Unknown
00001- <PAYMENT_PROVIDER_CODE>	INVALID_PAYLOAD	<ul style="list-style-type: none"> • Negative amount • Invalid currency • Invalid OCPS transaction id • Field length exceeded • Invalid characters • Invalid date time format • Mandatory fields not completed
00002- <PAYMENT_PROVIDER_CODE>	MISSING_LOOKUP	<ul style="list-style-type: none"> • Failed to find token • Failed to find location • Failed to find channel
00003- <PAYMENT_PROVIDER_CODE>	CROSS_ORG_TOKEN_USAGE	<ul style="list-style-type: none"> • Token from different organization
00004- <PAYMENT_PROVIDER_CODE>	AUTHENTICATION_ERRORS	<ul style="list-style-type: none"> • Failed to authenticate
00005- <PAYMENT_PROVIDER_CODE>	AUTHORIZATION_ERRORS	<ul style="list-style-type: none"> • Channel blocked • Service Provider blocked • Organization blocked • Credential not valid for channel • IP Address not allowed
00007- <PAYMENT_PROVIDER_CODE>	CONFIGURATION_MISSING	<ul style="list-style-type: none"> • Location's account holder is closed • Location's account holder is suspended

		<ul style="list-style-type: none"> Location does not belong to Channel
00008- <PAYMENT_PROVIDER_CODE>	DECLINED_PAYMENT	<ul style="list-style-type: none"> To be shown in resultReason of response to a declined payment: <ul style="list-style-type: none"> Cancel InvalidCard NotAllowed Refusal Error <p style="color: red;">Do not expose the details of specific refusal reasons to shoppers.</p>

Declined Payment Testing

It is possible to simulate declined payments. For Transaction Types: 001, 002, 004, 005, 008, 009, pass a Transaction Amount ending in the numerical value shown in column "Test Value". E.g. simulate blocked card enter transaction amount of 1125

Table 6. Declined payment simulation

Test Value	resultReason	Acquirer Response
121	Refusal	CANCELLED
122	Refusal	ACQUIRER_FRAUD
123	Refusal	DECLINED
124	Refusal	NOT_ENOUGH_BALANCE
125	Refusal	BLOCK_CARD
126	Refusal	CARD_EXPIRED
127	Refusal	INVALID_AMOUNT
128	Invalid Card	INVALID_CARD
129	Refusal	NOT_SUPPORTED
130	Refusal	ERROR
131	Refusal	DECLINED

132	Refusal	DECLINED
133	Refusal	REFERRAL
134	WrongPIN	INVALID_PIN (cancel the payment on the device when pin requested to get failure response)
135	Refusal	PIN_TRIES_EXCEEDED
136	Refusal	ISSUER_UNAVAILABLE
137	Refusal	WITHDRAWAL_AMOUNT_EXCEEDED
138	Refusal	WITHDRAWAL_COUNT_EXCEEDED
139	Refusal	NOT_ENOUGH_BALANCE. (Where partial authorisation is permitted a fail will not occur but rather partial payment)
142		To simulate contactless limit exceeded on terminal
144	Refusal	NOT_SUBMITTED
145	Refusal	FRAUD_CANCELLED
146	Refusal	TRANSACTION_NOT_PERMITTED
147		CVC_DECLINED
148	Refusal	RESTRICTED_CARD
149	Refusal	REVOCATION_OF_AUTH
150	Refusal	DECLINED_NON_GENERIC
151	Refusal	ISSUE_SUSPECTED_FRAUD
152	Refusal	NO_CHECKING_ACCOUNT
153	Refusal	NO_SAVINGS_ACCOUNT
154	Cancel	219 Shopper cancelled ctls fallback. (Simulate fallback to card present, tap card, then cancel when prompted to present card)
155		MOBILE_PIN_REQUIRED

(Test environment only allows to cancel transaction)		
157	Refusal	AUTHENTICATION_REQUIRED
158	Refusal	BAN_CURRENT_AID
159	Refusal	ONLINE_PIN_REQUIRED

Transaction Event Notifications

Oracle Payment Cloud Service can see webhooks to notify for the transaction events in this table. An URL endpoint and HMAC key must be provided so that it can be configured.

Notifications for which Transactions

	001 / 002 Auth. Only CP/CNP*	003 Manual Capture	004 / 005 Auth. & Capture CP/CNP*	006 Full Refund / Cancellation	007 Partial Refund	008 / 009 Get Token CP/CNP*	010 Token Payment
AUTHORIZATION	✔		✔			✔	✔
AUTHORIZATION ADJUSTMENT					✔ Where done before Capture		
AUTHORIZATION EXPIRE	✔ if not captured within 30 days						
CONTRACT CREATED						✔	
CAPTURE		✔	✔				✔
CAPTURE_FAILURE		✔ If it then fails	✔ If it then fails				✔ If it then fails
REVERSAL				✔ when received			
REFUNDED					✔ when received		
REFUND FAILURE				✔ If it then fails	✔ If it then fails		

*CP = Card Present, CNP = Card Not Present

Webhook Subscription Configuration:

Before Oracle can configure webhook delivery for your integration, the following information must be provided by the customer.

1. General Requirements

- a. You must determine which payment events you want to receive from Oracle Payment Cloud Service (OPCS).
- b. You must provide a secure, publicly accessible HTTPS endpoint to receive webhook notifications.
- c. Your system must support at least one of the supported authentication methods (Basic, Bearer, or OAuth-based).

2. Webhook Setup Requirements

You need to provide the following details:

a. Events to Subscribe To

Specify the list of event types you want to receive (for example: AUTHORISATION, AUTHORISATION_ADJUSTMENT, EXPIRE, CAPTURE, etc.).

b. Notification Endpoint

Provide the full URL of the endpoint where OPCS will send webhook messages.

c. Authentication Method

Choose one of the supported authentication mechanisms and provide the required information:

i. Basic Authentication

Username, Password

ii. Bearer Token Authentication

Static Bearer token

iii. OAuth-Based Authentication (Basic_With_Bearer)

OAuth token URL, Username, Password, (OPCS will use these credentials to fetch the Bearer token.)

Webhook event parameters

Table 7. webhook event parameters

1	2	Definition
live		false if not live, otherwise true.
event		Defines the type of event notification being sent. See event types below.
		AUTHORIZATION Indicates the result of a payment authorization request.
		AUTHORIZATION_ADJUSTMENT Represents an adjustment of the authorized amount in a payment request (e.g., partial refund before capture).
		AUTHORIZATION_EXPIRE Indicates that uncaptured amounts from an authorization have expired.
		CONTRACT_CREATED Sent when a recurring contract has been set up
		CAPTURE Indicates outcome of a request to capture a payment. Unless a CAPTURE_FAILURE is received, assume success.
		CAPTURE_FAILURE Indicates that a capture has failed as reported by the card scheme.
		REVERSAL Represents the result of a full reversal, leads to a cancellation if before capture, or a refund if after capture
		REFUNDED Indicates the result of a refund request.
		REFUND_FAILURE Indicates that a refund has failed due to rejection by the card scheme
eventDateTime		UTC date and time of the event in ISO 8601 format (yyyy-MM-DDThh:mm:ssTZD)
eventSuccess		Indicates whether the event was successful or not.
eventMessage		Message related to eventSuccess field
originalOpcsTransactionId		Included where a transaction has been modified and is OPCS's identifier for that

		original transaction that is being modified.
opcsTransactionId		OPCS unique reference for the transaction request.
originalClientId		The original transaction identifier the publisher / service provider's manages. Applicable where that transaction is being modified.
clientId		Integrator identifier for a transaction request.
originalPspReference		The original payment processor PSP Reference for the transaction, returned where a modification occurs against an existing transaction.
pspReference		Payment processor PSP Reference for the transaction request. Where CONTRACT_CREATED Notification refers to the opcsToken
channelCode		Channel code the transaction is linked to
amount		
	currency	ISO 3 character currency, e.g. USD
	value	Amount related to transaction in minor units
opcsToken		The OPCS Token returned on a token request. Will be included in AUTHORIZATION and CONTRACT_CREATED.
shopperReference		Shopper reference that was passed when a token was requested Will be included in AUTHORIZATION and CONTRACT_CREATED.
pspAdditionalData		
	cardBin	card bin
	cardIssuingCurrency	Issuing currency
	cardIssuingBank	Issuing Bank
	cardPaymentMethod	Payment method
	cardIssuingCountry	Issuing Country
	fundingsource	funding source
	shopperCountry	
	acquirerCode	
	cvcResultRaw	
	avsResultRaw	
	refusalReasonRaw	
	threeDOfferedResponse	
	threeDAuthenticatedResponse	

Step 5: Test and Go Live

Before going live, test all the use cases you intend to encounter in your live environment. When you are ready to go live, you need to:

1. Apply for a live account
2. Configure your live account
3. Submit a request to add payment methods in your live customer area
4. Switch from test to live endpoints

Oracle Payment Cloud Service Terminal APIs

Delta Changes between v1 and v2

Table 8. Delta changes

Area	V1	V2	Action required	Impact/Notes
Endpoint	api/v1/payments	api/v2/payments	Update base path and any client routes	Customers need to start using new endpoint
Location Reference	Optional for txnType 010	Mandatory for txnType 010.	Send locationRefrence for txnType 010.	Required for merchant and store-level mapping.
Organization Reference	Optional for txnType 010	Mandatory for txnType 010.	Send organizationRefrence for txnType 010.	Required for merchant and store-level mapping.

Authorization Card Present (001)**Endpoint**`/api/v2/payments/`**Method:** POST**Request**

```
{
  "transactionType": "001",
  "terminalId": "P400Plus-123456789",
  "channelCode": "ch123456",
  "locationReference": "Test_Loc1",
  "transactionDetails": {
    "currency": "USD",
    "amount": 6000
  },
  "transactionReferences": {
    "clientTransactionId": "12346",
    "requestDateTime": "2023-11-06T14:54:16.296651Z"
  }
}
```

Response

```
{
  "approvalCode": "123456",
  "channelCode": "ch123456",
  "locationReference": "Loc1",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "entryModes": [
        "CONTACTLESS"
      ],
      "expiryDate": "2/2028",
      "fundingSource": "CREDIT",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "CARD"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "refusalReasonRaw": "APPROVED"
  },
  "result": {
    "authorizedAmount": 6000,
    "currency": "USD",
    "resultCode": "SUCCESS"
  },
  "shopperData": {
    "cardHolderName": " /",
    "countryCode": "US"
  }
}
```

```
    "terminalId": "P400Plus-123456789",
    "transactionDetails": {
      "amount": 6000,
      "currency": "USD"
    },
    "transactionReceipts": [OBJECT EXAMPLE: Transaction Receipt Object, 22],
    "transactionReferences": {
      "clientTransactionId": "123456",
      "opcsTransactionId": "ABC1234DEF",
      "pspReference": "VBTSRJJQ35GSKTV5",
      "requestDateTime": "2023-11-06T14:54:16.296651Z",
      "responseDateTime": "2025-03-17T12:26:02.985506694Z"
    },
    "transactionType": "001"
  }
}
```

Authorization Card Not Present (002)

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "002",
  "terminalId": "P400Plus-123456",
  "channelCode": "ch123456",
  "transactionDetails": {
    "currency": "USD",
    "amount": 5800
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "requestDateTime": "2024-11-06T14:54:16.296651Z"
  }
}
```

Response

```
{
  "approvalCode": "054831",
  "channelCode": "ch123456",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "coBrandedWith": "pulse,nyce",
      "entryModes": [
        "KEYED"
      ],
      "expiryDate": "2/2028",
      "fundingSource": "CREDIT",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "CARD"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "acquirerReference": "N75L57767RT",
    "avsResultRaw": "2",
    "cvcResultRaw": "P",
    "refusalReasonRaw": "AUTHORISED"
  },
  "result": {
    "authorizedAmount": 5800,
    "currency": "USD",
    "resultCode": "SUCCESS"
  },
  "shopperData": {
    "countryCode": "US"
  }
}
```

ORACLE

```
    },
    "terminalId": "P400Plus-123456789",
    "transactionDetails": {
      "amount": 5800,
      "currency": "USD"
    },
    "transactionReceipts": [OBJECT EXAMPLE: Transaction Receipt Object, 22]
  ]
  },
  "requiredSignatureFlag": false
}
],
"transactionReferences": {
  "clientTransactionId": "123456",
  "opcsTransactionId": "ABC1234DEF",
  "pspReference": "Q5Z3WWZD6FJJXQ65",
  "requestDateTime": "2024-11-06T14:54:16.296651Z",
  "responseDateTime": "2025-03-17T12:27:19.228179414Z"
},
"transactionType": "002"
}
```

Capture Authorization (003)

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "003",
  "channelCode": "ch123456",
  "transactionDetails": {
    "currency": "USD",
    "amount": 843
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "originalOpcsTransactionId": "FED6543CBA",
    "requestDateTime": "2024-02-05T10:44:16.296651Z"
  }
}
```

Response

```
{
  "channelCode": "ch123456",
  "result": {
    "authorizedAmount": 843,
    "currency": "USD",
    "resultCode": "SUCCESS"
  },
  "transactionDetails": {
    "amount": 843,
    "currency": "USD"
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "opcsTransactionId": "ABC1234DEF",
    "originalOpcsTransactionId": "FED6543CBA",
    "originalPspReference": "VBTSRJJQ35GSKTV5",
    "pspReference": "M2JLCNW3T3TPBC75",
    "requestDateTime": "2024-02-05T10:44:16.296651Z",
    "responseDateTime": "2025-03-17T12:56:29.335747724Z"
  },
  "transactionType": "003"
}
```

Authorization and Capture Card Present (004)

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "004",
  "terminalId": "P400Plus-123456",
  "channelCode": "ch123456",
  "transactionDetails": {
    "currency": "USD",
    "amount": 2000
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "requestDateTime": "2024-08-02T14:54:16.296651Z"
  }
}
```

Response

```
{
  "approvalCode": "123456",
  "channelCode": "ch123456",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "entryModes": [
        "CONTACTLESS"
      ],
      "expiryDate": "2/2028",
      "fundingSource": "CREDIT",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "CARD"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "refusalReasonRaw": "APPROVED"
  },
  "result": {
    "authorizedAmount": 2000,
    "currency": "USD",
    "resultCode": "SUCCESS"
  },
  "shopperData": {
    "cardHolderName": " /",
    "countryCode": "US"
  },
  "terminalId": "P400Plus-123456789",
  "transactionDetails": {
    "amount": 2000,
    "currency": "USD"
  }
}
```

```
    },
    "transactionReceipts": [OBJECT EXAMPLE: Transaction Receipt Object, 22]
      },
      "requiredSignatureFlag": false
    }
  ],
  "transactionReferences": {
    "clientTransactionId": "123456",
    "opcsTransactionId": "ABC1234DEF",
    "pspReference": "ZVR2DNVWNV5FKQV5",
    "requestDateTime": "2024-08-02T14:54:16.296651Z",
    "responseDateTime": "2025-03-17T12:28:10.837901751Z"
  },
  "transactionType": "004"
}
```

Authorization and Capture Card Not Present (005)**Endpoint**`/api/v2/payments/`**Method:** POST**Request**

```
{
  "transactionType": "005",
  "terminalId": "P400Plus-123456",
  "channelCode": "ch123456",
  "transactionDetails": {
    "currency": "USD",
    "amount": 1365
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "requestDateTime": "2024-01-17T11:28:16.296651Z"
  }
}
```

Response

```
{
  "approvalCode": "087357",
  "channelCode": "ch123456",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "coBrandedWith": "pulse,nyce",
      "entryModes": [
        "KEYED"
      ],
      "expiryDate": "2/2028",
      "fundingSource": "CREDIT",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "CARD"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "acquirerReference": "NSB6L9669L5",
    "avsResultRaw": "2",
    "cvcResultRaw": "P",
    "refusalReasonRaw": "AUTHORISED"
  },
  "result": {
    "authorizedAmount": 1365,
    "currency": "USD",
    "resultCode": "SUCCESS"
  },
  "shopperData": {
    "countryCode": "US"
  },
  "terminalId": "P400Plus-123456789",
}
```

```
"transactionDetails": {  
  "amount": 1365,  
  "currency": "USD"  
},  
"transactionReceipts": [OBJECT EXAMPLE: Transaction Receipt Object, 22]  
  },  
  "requiredSignatureFlag": false  
  }  
],  
"transactionReferences": {  
  "clientTransactionId": "123456",  
  "opcsTransactionId": "ABC1234DEF",  
  "pspReference": "STR9QNVWNV5FKQV5",  
  "requestDateTime": "2024-01-17T11:28:16.296651Z",  
  "responseDateTime": "2025-03-17T12:28:56.378193402Z"  
},  
"transactionType": "005"  
}
```

Full Refund / Cancellation:

If transaction has been captured a full amount refund will be issued else the transaction will be a cancel/void.

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "006",
  "terminalId": "P400Plus-123456",
  "transactionReferences": {
    "clientTransactionId": "123456",
    "originalOpcsTransactionId": "FED6543CBA",
    "requestDateTime": "2024-07-02T14:54:16.296651Z"
  },
  "channelCode": "ch123456"
}
```

Response

```
{
  "channelCode": "ch123456",
  "result": {
    "resultCode": "RECEIVED"
  },
  "terminalId": "P400Plus-123456789",
  "transactionReferences": {
    "clientTransactionId": "123456",
    "opcsTransactionId": "ABC1234DEF",
    "originalOpcsTransactionId": "FED6543CBA",
    "originalPspReference": "L85SPLVWNV5FKQV5",
    "pspReference": "XR5CCLGZZPLLGB75",
    "requestDateTime": "2024-07-02T14:54:16.296651Z",
    "responseDateTime": "2025-03-17T12:59:04.801666198Z"
  },
  "transactionType": "006"
}
```

Partial Refund (007)

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "007",
  "reference": "test-reference",
  "transactionDetails": {
    "amount": 2,
    "currency": "USD"
  },
  "transactionReferences": {
    "clientTransactionId": "partialrefundrbe_after003--30ndJan2025",
    "originalOpcsTransactionId": "iphsJEmr9f",
    "requestDateTime": "2025-01-30T16:35:44.000Z"
  },
  "channelCode": "channelCode"
}
```

Response

```
{
  "channelCode": " channelCode ",
  "locationReference": "TEST_location",
  "result": {
    "resultCode": "RECEIVED"
  },
  "transactionDetails": {
    "amount": 2,
    "currency": "USD"
  },
  "transactionReferences": {
    "clientTransactionId": "client",
    "opcsTransactionId": "Gh4ERTijrp",
    "originalOpcsTransactionId": " iphsJEmr9f ",
    "originalPspReference": "K32rb6GP8RPWCN75",
    "pspReference": "CHD9FJ3edXKX84V5",
    "requestDateTime": "2025-01-30T16:35:44.000Z",
    "responseDateTime": "2025-10-28T10:23:56.738624Z"
  },
  "transactionType": "007"
}
```

Generate Token Card Present (008)

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "008",
  "recurringProcessingModel": "UnscheduledCardOnFile",
  "shopperReference": "Test123",
  "terminalId": "P400Plus-123456",
  "transactionDetails": {
    "amount": 1253,
    "currency": "USD"
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "requestDateTime": "2024-03-12T16:04:16.296651Z"
  },
  "channelCode": "ch123456"
}
```

Response

```
{
  "approvalCode": "123456",
  "channelCode": "ch123456",
  "opcsToken": "P3XZTJQCM29F2S75",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "entryModes": [
        "CONTACTLESS"
      ],
      "expiryDate": "2/2028",
      "fundingSource": "CREDIT",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "CARD"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "refusalReasonRaw": "APPROVED"
  },
  "recurringProcessingModel": "UnscheduledCardOnFile",
  "result": {
    "authorizedAmount": 1253,
    "currency": "USD",
    "resultCode": "SUCCESS"
  },
  "shopperData": {
    "cardHolderName": " /",
    "countryCode": "US"
  },
  "shopperReference": " Test123",
}
```

```
    "terminalId": "P400Plus-123456789",
    "transactionDetails": {
      "amount": 1253,
      "currency": "USD"
    },
    "transactionReceipts": [OBJECT EXAMPLE: Transaction Receipt Object, 22],
    "transactionReferences": {
      "clientTransactionId": "123456",
      "opcsTransactionId": "ABC1234DEF",
      "pspReference": "C7ZM3ZZD6FJJXQ65",
      "requestDateTime": "2024-03-12T16:04:16.296651Z",
      "responseDateTime": "2025-03-17T12:29:42.057845508Z"
    },
    "transactionType": "008"
  }
}
```

Generate Token Card Not Present (009)

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "009",
  "recurringProcessingModel": "UnscheduledCardOnFile",
  "shopperReference": "Test123",
  "terminalId": "P400Plus-123456",
  "transactionDetails": {
    "amount": 1253,
    "currency": "USD"
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "requestDateTime": "2024-03-12T16:04:16.296651Z"
  },
  "channelCode": "ch123456"
}
```

Response

```
{
  "approvalCode": "055685",
  "channelCode": "ch123456",
  "opcsToken": "P3XZTJQCM29F2S75",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "coBrandedWith": "pulse,nyce",
      "entryModes": [
        "KEYED"
      ],
      "expiryDate": "2/2028",
      "fundingSource": "CREDIT",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "CARD"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "acquirerReference": "FF55LVNVZ99",
    "avsResultRaw": "2",
    "cvcResultRaw": "P",
    "refusalReasonRaw": "AUTHORISED"
  },
  "recurringProcessingModel": "UnscheduledCardOnFile",
  "result": {
    "authorizedAmount": 1253,
    "currency": "USD",

```

```
    "resultCode": "SUCCESS"
  },
  "shopperData": {
    "countryCode": "US"
  },
  "shopperReference": "Test123",
  "terminalId": "P400Plus-123456789",
  "transactionDetails": {
    "amount": 1253,
    "currency": "USD"
  },
  "transactionReceipts": [OBJECT EXAMPLE: Transaction Receipt Object, 22],
  "transactionReferences": {
    "clientTransactionId": "123456",
    "opcsTransactionId": "ABC1234DEF",
    "pspReference": "P8PJ89XK2MBFKQV5",
    "requestDateTime": "2024-03-12T16:04:16.296651Z",
    "responseDateTime": "2025-03-17T12:34:00.330757699Z"
  },
  "transactionType": "009"
}
```

Make Token Payment (010)

Endpoint

/api/v2/payments/

Method: POST

Request

```
{
  "transactionType": "010",
  "opcsToken": "P3XZTJQCM29F2S75",
  "recurringProcessingModel": "UnscheduledCardOnFile",
  "shopperReference": "Test123",
  "transactionDetails": {
    "amount": 0,
    "currency": "USD"
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "requestDateTime": "2024-03-12T16:04:16.296651Z"
  },
  "channelCode": "ch123456",
  "organizationReference": "VV0IXLLJ4IQ1",
  "locationReference": "test_location"
}
```

Response

```
{
  "channelCode": "ch123456",
  "locationReference": " test_location ",
  "opcsToken": "1234",
  "paymentMethodData": {
    "cardData": {
      "cardBin": "541333",
      "cardIssuingCountry": "BE",
      "cardIssuingCurrency": "USD",
      "cardPaymentMethod": "mc",
      "cardSchemeCommercial": "unknown",
      "cardSummary": "9999",
      "expiryDate": "2/2028",
      "issuerBin": "54133300",
      "paymentMethodVariant": "mc"
    },
    "paymentMethodType": "STORED_VALUE"
  },
  "rawAcquirerResult": {
    "acquirerCode": "TestPmmAcquirer",
    "acquirerReference": "5HZ4554FGHH",
    "avsResultRaw": "2",
    "cvcResultRaw": "P",
    "refusalReasonRaw": "AUTHORISED"
  },
  "recurringProcessingModel": "UnscheduledCardOnFile",
  "result": {
    "authorizedAmount": 0,
    "currency": "USD",
  }
}
```

```
    "resultCode": "SUCCESS"
  },
  "shopperData": {},
  "shopperReference": "Test123",
  "transactionDetails": {
    "amount": 0,
    "currency": "USD"
  },
  "transactionReferences": {
    "clientTransactionId": "123456",
    "opcsTransactionId": "ABC1234DEF",
    "pspReference": "G3GWHSQC97RF6975",
    "requestDateTime": "2024-03-12T16:04:16.296651Z",
    "responseDateTime": "2025-03-17T12:53:05.615212325Z"
  },
  "transactionType": "010"
}
```

Oracle Payment Cloud Service – Webhook Specification

AuthorizaHon Webhook

```
{
  "live": "false",
  "event": "AUTHORIZATION",
  "eventDateTime": "2023-09-11T11:52:15+02:00",
  "eventSuccess": "true",
  "eventMessage": "",
  "opcsTransactionId": "123456ABCD",
  "clientTransactionId": "123456",
  "pspReference": "A456C987ZZ",
  "channelCode": "CHCTS001",
  "amount": {
    "currency": "USD",
    "value": 100
  },
  "pspAdditionalData": {
    "cardBin": 54133,
    "cardIssuingCurrency": "USD",
    "cardIssuingBank": "null",
    "cardPaymentMethod": "mc",
    "cardIssuingCountry": "US",
    "fundingSource": "CREDIT",
    "tokenTxVariant": "null",
    "untokenisedCardSummary": "null",
    "cardHolderName": "TEST",
    "shopperCountry": "US",
    "acquirerCode": "TestPmmAcquirer",
    "cvcResultRaw": "Unknown",
    "avsResultRaw": "Unknown",
    "refusalReasonRaw": "APPROVED",
    "threeDOfferedResponse": "null",
    "threeDAuthenticatedResponse": "null"
  }
}
```

```
} }
```

AuthorizaHon Adjustment Webhook

```
{
  "live": "false",
  "event": "AUTHORISATION_ADJUSTMENT",
  "eventDateTime": "2025-05-13T14:57:34.000+0000",
  "eventSuccess": true,
  "eventMessage": "Acquirer Error",
  "opcsTransactionId": "3DyY7K9uzq",
  "originalOpcsTransactionId": "3DyY7K9uzq",
  "pspReference": "VRCJBX79632WQT65",
  "channelCode": "ccorcAOMKY9",
  "originalPspReference": "JPXRPZ5RPXFH9PT5",
  "clientTransactionId": "partialrefundrbe_after004--12May2025",
  "amount": {
    "value": 3325,
    "currency": "USD"
  }
}
```

Token Creation Webhook

```
{
  "live": "false",
  "event": "CONTRACT_CREATED",
  "eventDateTime": "2025-05-23T10:28:47.000+0000",
  "eventSuccess": true,
  "eventMessage": "",
  "pspReference": "AAAAAAAAAAAAAAAAAKpZ+zpcUmsyZlu1bFYGNrD6NJ+UscYNCZG+pBS3uNjU=",
  "originalPspReference": "XVFLWRMJGwVV9ST5",
  "opcsTransactionId": "DP11HQI2Ea",
  "opcsToken": "AAAAAAAAAAAAAAAAAKpZ+zpcUmsyZlu1bFYGNrD6NJ+UscYNCZG+pBS3uNjU=",
  "clientTransactionId": "test1234",
  "shopperReference": "shopper_1",
  "channelCode": "ccextU7CJ80",
  "amount": {
    "value": 300,
    "currency": "USD" } }
```

Cancellation Webhook

```
{
  "live": "false",
  "event": "REVERSAL",
  "eventDateTime": "2025-04-16T15:36:54.000+0000",
  "eventSuccess": true,
  "eventMessage": "",
  "originalOpcsTransactionId": "BditTy1tjs",
  "opcsTransactionId": "BditTy1tjs",
  "clientTransactionId": "Happy_Path_002_08thApr2025",
  "pspReference": "DQZ9Z394CNTZJMV5",
  "orginalPspReference": "V8JW44RLQSWQQ2V5",
  "channelCode": "ccorcaAOMKY9",
  "amount": {
    "value": 0,
    "currency": "EUR"
  },
  "shopperData": {
    "shopperCountry": "US"
  }
}
```

Refund Webhook


```
{
  "live": "false",
  "event": "REFUNDED",
  "eventDateTime": "2025-05-09T17:39:43.000+0000",
  "eventSuccess": false,
  "eventMessage": "Requested refund amount too high",
  "originalOpcsTransactionId": "dFvKV3AHdM",
  "opcsTransactionId": "iHY6ShyhTB",
  "clientTransactionId": "partialrefundrbe_afterEH004--09May2025",
  "pspReference": "ZQFS6XKVT3XW24V5",
  "orginalPspReference": "C8K69VNG6GH45G75",
  "channelCode": "ccorcaAOMKY9",
  "amount": {
    "value": 6436,
    "currency": "USD"
  }
}
```


Refund Failure Webhook


```
{
  "live": "false",
  "event": "REFUND_FAILURE",
  "eventDateTime": "2025-04-09T13:28:49.000+0000",
  "eventSuccess": true,
  "eventMessage": "Refund Failed",
  "originalOpcsTransactionId": "x87BXSLpOZ",
  "opcsTransactionId": "x87BXSLpOZ",
  "pspReference": "GDBPQ2ZKB6N49W65",
  "orginalPspReference": "FJ9LQPCF7FQRX2V5",
  "channelCode": "ccextU7CJ80",
  "amount": {
    "value": 119900,
    "currency": "USD"
  }
}
```

Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2025, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.