



<https://verifone.cloud/docs/online-payments/api-integration/affirm>

Updated: 07-Nov-2025

Affirm

Overview

Affirm offers flexible financing options for your shoppers to pay over time, while also ensuring you to receive the full payment upfront.

This documentation will detail how you can take payments with Affirm using [Verifone's eCommerce API](#).

Availability

- Country: United states of America
- Currency: USD (US Dollar)
- Channel: Instore and E-commerce

Benefits

Expand Your Reach

- Access 50M+ high-intent shoppers in Affirm's growing customer network.
- Join 337,000+ merchants already using Affirm to attract new customers and drive repeat business.

Boost Conversions & Order Value

- 28% fewer abandoned carts than with other BNPL providers—keep browsers from bouncing.
- 70%+ lift in average order values, as shoppers feel empowered to buy more with flexible payment plans.
- 76% of customers say they wouldn't have purchased without Affirm—capture sales that might otherwise be lost.

Smarter Checkout, More Flexibility

- Adaptive Checkout™ technology automatically tailors' payment options—from 4 interest-free biweekly payments to installments up to 36 months—helping every shopper find a plan that fits.

Simplified Operations

- When integrated through Verifone, you get one unified report for all payment methods, including Affirm—streamlined reconciliation and clearer insights.

- Seamless integration means a faster setup and a smoother customer experience.

Give shoppers a smarter, more transparent way to pay—and turn flexibility into loyalty.

Integrations

Checkout Integrations

Redirect or iFrame — choose the flow that works best for your business.

Offer Affirm's flexible payments directly in your checkout experience, keeping customers engaged and conversions high—without compromising on security or speed.

Ecom API (Server-to-Server)

Take full control with our Ecommerce API for a fully customizable integration.

Create seamless, branded payment experiences, manage transactions, and handle authorizations directly through your own backend—ideal for advanced merchants and enterprise partners.

POS Integration

Bring Affirm into your in-store checkout experience.

Empower customers to pay overtime—right at the terminal—using Verifone's POS integration. A unified experience across channels means shoppers can buy with confidence anywhere they shop.

Plugins for Popular Platforms

Get up and running quickly with pre-built plugins for leading ecommerce platforms like WooCommerce, Magento.

No heavy lifting — just install, configure, and start offering flexible payment options in minutes.

Before you get started

Starting from scratch? Follow these steps to get started.

- Have a [Verifone Central](#) account in either our [Sandbox or Production](#) environment with API Access.
 - Don't have a Verifone Central Account? Contact your [regional sales team](#) to get started.
 - Forgot your password? Click the [portal link for your region](#) and [reset your password](#).
- Have an Affirm payment provider contract setup by Verifone administration.

Collect your Verifone Credentials (environment variables)

- [Select an environment](#) (Sandbox or Production)
 - Make sure to set up the base of the URL (environment) to be configurable for when you are ready to proceed to production.
- Collect your [Payment Provider Contract ID](#)
 - The Payment Provider Contract (PPC) represents the merchant's contract with their acquiring partner, including key details such as the Merchant ID.

Take payments with Affirm

Decide for REDIRECT or MODAL checkout flow mode

- **REDIRECT** (default): Users are redirected to Affirm site, where they authenticate and go through the checkout flow. Once they complete checkout, they're returned to the merchant site.
- **MODAL**: Users go through the Affirm flow in a modal window while remaining on merchant site. Modal checkout with the help of `affirm.js` will enable client-side callbacks so that merchant can do post checkout confirmation procedures on merchant site.

Define and add a merchant reference Recommended

Use the **merchant_reference** field to specify an alphanumeric ID of your choosing. Typically, this corresponds to an order number from your inventory management system or web platform, allowing you to easily reconcile transaction reports with your orders.

For example: “Merchant_reference”: “Order-123456789”

Capture now

Default capture flow behavior is *true* which means that authorization and capture is performed in one step (*auto-capture*) by Verifone. If `capture_now` flag is set to *false* while initiating the payment the capture has to be executed as a separate process after the authorization. See [Capture an authorized Affirm payment](#).

Payments with Affirm in REDIRECT mode

Step 1: Create an Affirm Payment

Using the Initiate a Affirm payment API call, we can generate a URL for the customer to visit and pay with Affirm.

API Reference: <https://verifone.cloud/api-catalog/verifone-ecommerce-api#tag/Ecom-Payments/operation/affirmInitTransaction>

Request Method: POST

URL {environment}/api/v2/transactions/affirm

Headers: **x-vfi-api-idempotencyKey** – unique UUID to identify the transaction

Body:

```
{
  "payment_provider_contract": "{{ppc}}",
  "merchant_reference": "Order Sale 123",
  "amount": 60000,
  "redirect_url": "https://verifone.cloud/",
  "cancel_url": "https://verifone.cloud/",
  "currency_code": "USD",
  "locale": {
    "country_code": "US"
  },
  "capture_now": true,
}
```

Response:

```
{
"amount": 60000,
"created_at": 1729047737.9813294,
"merchant_reference": " Order Sale 123",
"status": "INITIATED",
"id": "a520d533-e56c-4751-b9da-e448e12a298f",
"payment_url":
"https://sandbox.affirm.com/products/checkout?public_api_key=UFM3YKEQ6FX62&checko
/
"processor": "AFFIRM",
"payment_product": "AFFIRM",
"payment_product_type": "Affirm"
}
```

Explanation of fields:

- **id** - the ID of the transaction.
- **payment_url** - the URL which can be accessed from any browser. It should be used to redirect the shopper
- **redirect_url** – The URL the customer will be redirected back to after completing the payment successfully.
- **cancel_url** – The URL the customer will be redirect back to if the cancel button is selected in the top left corner.
- **capture_now** – whether ‘auto-capture’ or not. See [Capture now](#).

Save these details later for the next steps:

1. payment_url
2. checkout_ari from payment_url
3. id (This the transaction ID)

Step 2: Redirect the customer to the payment URL

Redirect the customer to the payment_url from [step 1](#).

The customer will then need to authenticate with Affirm via a Mobile number or a Passkey.

See [Affirm Checkout Dialog](#).

Step 3: Identify the returning customer

After completing the affirm payment, the customer will be posted to the redirect_url or cancel_url.

In the **POST** message made to your browser, Affirm will return the checkout_ari as checkout_token:

Name	×	Headers	Payload	Preview	Response
payment-confirmed.html					
	▼	Form Data	view parsed		
			checkout_token=QD6U6W3UJRGYBS0C		

Use the checkout_token returned here to match against the corresponding checkout_ari & id (transaction_id) received in [step 1](#).

Step 4: Query the transaction result with Read Transaction

Using the id (transaction id) from [step 1](#), perform a **Read Transaction** call to know the transaction result.

API Reference: <https://verifone.cloud/api-catalog/verifone-ecommerce-api#tag/Transaction/operation/readTransaction>

Request Method: GET

URL: {environment}/api/v2/transaction/{transaction_id}

Response:

```
{
  "id": "a520d533-e56c-4751-b9da-e448e12a298f",
  "amount": "600.00",
  "currency_code": "USD",
  "created_at": "2024-10-22T01:35:22.676Z",
  "country_code": "US",
  "shipping_information": {},
  "customer_details": {
    "billing": {},
    "shipping": {}
  },
  "merchant_reference": "Order 123456",
  "status": "SALE AUTHORISED",
  "processor_reference": "B3312-7BII",
  "cvv_present": false,
  "processor_payer_id": "8saASadmd2e12l3dsm",
  "dynamic_descriptor": "Order Sale",
  "shopper_interaction": "ecommerce",
  "merchant_id": "1234567",
  "payment_summary": {
    "captured_amount": "600.00"
  },
  "transaction_status": "AUTHORISED",
  "transaction_type": "SALE",
  "entity_id": "432dj44-c41b-4c9b-964e-9cbf9da729d9",
  "acquirer_merchant_id": "ASHQWIIQQ222444"
}
```

With a transaction_status of AUTHORISED we know the payment was successfully completed by the customer.

Payments with Affirm in MODAL mode

Step 1: Create an Affirm Payment

Verifone performs the Affirm Direct Checkout internally and provides the data to let you launch the Affirm modal by calling Affirm Javascript SDK to show checkout page.

Add the mode flag to the common request body (see [REDIRECT](#) part). Checkout ID and Public Api Key are now additional parts of the response.

API Reference: <https://verifone.cloud/api-catalog/verifone-ecommerce-api#tag/Ecom-Payments/operation/affirmInitTransaction>

Request Method: POST

URL: {{env}}/api/v2/transactions/affirm

Headers: **x-vfi-api-idempotencyKey** – unique UUID to identify the transaction

Body:

```
{ ...  
"mode": "MODAL"  
}
```

Response:

```
{ ...  
"checkoutId": "{{Checkout ID}}",  
"publicApiKey": "{{Public API Key}}",  
}
```

Explanation of fields:

- **mode** – when **MODAL** mode shall be used this flag must be set to *MODAL* to indicate the checkout flow type towards Verifone.
- **checkout_id** – a unique identifier representing the Payment Host checkout ID. This value is available only in **MODAL** mode.
- **public_api_key** – the merchant's public api key. This value is available only in **MODAL** mode.

Save these details later for the next steps:

1. checkoutId
2. publicApiKey

Step 2: Initialize and open Affirm checkout page

Prepare the config object for Affirm Javascript library affirm.js to be used with initialization function.

Affirm API Reference:

<https://docs.affirm.com/payments/docs/afjs-reference>

<https://docs.affirm.com/payments/docs/open-affirm-checkout>

<https://docs.affirm.com/payments/docs/modal-vs-redirect-checkout>

```
var _affirm_config = {  
public_api_key: public_key,
```

```
script: "https://cdn1-sandbox.affirm.com/js/v2/affirm.js"
}
```

Explanation of fields:

- **public_api_key** – your public API key received in [step 1](#).
- **script** – link to Affirm's JavaScript library 'affirm.js' to be used in MODAL mode. Replace sandbox environment URL in script parameter with live value after integration.
 - Sandbox: <https://cdn1-sandbox.affirm.com/js/v2/affirm.js>
 - Production: <https://cdn1.affirm.com/js/v2/affirm.js>

Initiate the checkout process by calling '**affirm.checkout(checkoutObject)**' with Checkout object:

```
var checkoutObject = {
  "checkoutAri": checkoutId,
  "metadata": {
    "mode": "modal"
  }
}
```

Explanation of fields:

- **checkoutAri** – a unique identifier representing the Payment Host checkout ID received in [step 1](#).
- **mode** – indicates the checkout flow type *MODAL* towards Affirm.

Optional: create callback functions to receive the feedback of the checkout process and to handle the events:

```
var callbacks = {
  onFail:function(error) {
    // Error handling
  },
  onSuccess:function(checkout) {
    // Success procedures
  },
  onOpen:function(token) {
    // Initialization successful
  },
  onValidationError:function(checkout_validation_error) {
    // Validation error handling
  }
}
```

Now call '**affirm.checkout.open(callbacks)**' to open the Affirm checkout portal. See [Affirm Checkout Dialog](#).

See the [HTML test page example](#) which starts the modal checkout flow on button pressing. Checkout ID and Public Api Key have to be provided.

Step 3: Receive Callback and Complete the Transaction

Callback onOpen will be received when the customer has loaded the Affirm checkout flow.

When receiving the OnSuccess callback finish the transaction by sending *Complete Affirm Payment* message.

In case of OnFail or onValidationError callback add a refusal_reason to the completion request to inform Verifone about the negative result.

API Reference: <https://verifone.cloud/api-catalog/verifone-ecommerce-api#tag/Payment-Modifications/operation/affirmCompleteTransaction>

Request Method: POST **URL:** {environment}/api/v2/transactions/affirm_complete

Path parameters: id – string (example: a520d533-e56c-4751-b9da-e448e12a298f)

Headers: x-vfi-api-idempotencyKey - unique UUID to identify the transaction

Body:

```
{
  "merchant_reference": "7a1db7a8-6f24-4bc5-a51b-cef33fc05140",
  // add refusal reason only in negative case!
  "refusal_reason": "ACQUIRER_COMMS_FAILURE"
}
```

Response:

```
{
  "id": "a520d533-e56c-4751-b9da-e448e12a298f",
  "authorization_expiration": "auth transaction expiration",
  "amount": 150,
  "currency_code": "USD",
  "created_at": "2019-08-24T14:15:22Z",
  "created_by": "ID of creator",
  "customer": "ID of a customer",
  "merchant_reference": "merchant reference",
  "processor": "AFFIRM",
  "payment_product": "AFFIRM",
  "payment_product_type": "Affirm",
  "status": "AUTHORIZED",
  "status_reason": "status reason",
  "geo_location":
  [
    52.370216,
    4.895168
  ],
  "country_code": "EU"
}
```

Explanation of fields:

- **id** – the ID of the transaction.
- **merchant_reference** – the reference specified by the merchant to identify the transaction.
- **refusal_reason** – the failure reason, only to be set if NOT received the onSuccess callback. Transaction state is set to DECLINED at Verifone side if available in the request.
- **status** – the status of the authorization. With a status of AUTHORIZED we know the payment was successfully completed by the customer.

Affirm Checkout Dialog

Note for sandbox testing:

- Any valid US number format is accepted e.g **2349990000**
- The verification code is set to **123456**

After authentication, the customer chooses their payment plan and confirms the bank account to be charged.



In proud partnership with **PortalCST KlarnaRE IBC**
Store

Big purchase? No problem.

With Affirm, you can buy now and pay over time.

Checking your loan eligibility **won't affect your credit score.**

MOBILE NUMBER
(234) 999-0000

We'll send you a verification code via SMS.

☐ Remember me



Continue

[New number? Update your account.](#)

Or, if you have a passkey



Log in with passkey

By continuing, I agree to Affirm's [Terms of Service](#) and authorize Affirm to obtain, use, and share consumer reports about me. See our [Privacy Policy](#) to learn more about our privacy practices.

[Verify](#)[Pick a plan](#)[Review details](#)

Purchase amount \$600.00

Choose a payment plan

We're not big on fees, so we don't charge any.

Lowest interest



\$210.08 every month for 3 months

Interest \$30.24

Total of payments \$630.24



\$108.93 every month for 6 months

Interest \$53.57

Total of payments \$653.57

Lowest monthly payment



\$58.49 every month for 12 months

Interest \$101.91

Total of payments \$701.91

[Plans are subject](#) to final eligibility and approval.

Choose this plan



Verify

Pick a plan

Review details

Purchase amount \$600.00

Payment plan

Change

Lowest monthly payment	
\$58.49 every month for 12 months	
Interest (29.99% APR)	\$101.91
Total of payments	\$701.91

Review and pay

12 total payments ^

● Due next month	\$58.49
○ December 22, 2024	\$58.49
+8 more payments	
○ September 22, 2025	\$58.49
○ October 22, 2025	\$58.52

These due dates are close estimates

AutoPay (optional) ⓘ

ON

PAYMENT METHOD - ACH

Bank account 3214



By continuing, you agree to the [Autopay Authorization](#).

☐

I have reviewed and agree to the [Truth in Lending Disclosure](#), [Loan Agreement](#), PortalCST KlarnaRE IBC Store refund policy, and have received the [Credit Score Disclosure](#).

Confirm

Example of HTML page embedding affirm JavaScript library affirm.js

```
<html>
  <head>
    <title>Affirm Test</title>
    <script type="text/javascript">
      function initialize(public_key){
        _affirm_config = {
          public_api_key: public_key,
          script: "https://cdn1-sandbox.affirm.com/js/v2/affirm.js"
        };
        (function(m,g,n,d,a,e,h,c){var b=m[n]||{};k=document
.createElement(e),p=document.getElementsByTagName(e)[0],l=function(a,b,c){
return function(){a[b]._.push([c,arguments])}};b[d]=l(b,d,"set");var
f=b[d];b[a]={};b[a]._=[];f._=[];b._=[];b[a][h]=l(b,a,h);b[c]=function()
{b._.push([h,arguments])};a=0;for(c=
"set add save post open empty reset on off trigger ready setProduct".split(
" ");a<c.length;a++)f[c[a]]=l(b,d,c[a]);a=0;for(c=["get","token","url",
"items"];a<c.length;a++)f[c[a]]=function(){};k.async=
!0;k.src=g[e];p.parentNode.insertBefore(k,p);delete
g[e];f(g);m[n]=b})(window,_affirm_config,"affirm","checkout","ui","script",
"ready","jsReady");
      }
      function showDialog(){
        var public_key = document.getElementById("public_key"
).value;
        initialize(public_key);
        var checkoutId = document.getElementById("checkout-id"
).value;
        var checkoutInfo = {
          "checkoutAri": checkoutId,
```

```

        "metadata": {
            "mode": "modal"
        }
    };
    var callbacks = {
        onOpen: function(res){
            alert("Open");
            console.log(res);
        },
        onValidationError: function(res){
            alert("Validation Error");
            console.log(res);
        },
        onFail: function(res){
            alert("Failed");
            console.log(res);
        },
        onSuccess: function(res){
            alert("OnSuccess");
            console.log(res);
        }
    }
    affirm.checkout(checkoutInfo);
    affirm.checkout.open(callbacks);
}
</script>
</head>
<body>
    <input type="text" id="public_key" value="WEP5F08QZR50F2RQ">
    <input type="text" id="checkout-id" value="UNM4DMNALG3QER7B">
    <input type="button" value="Pay" onclick="showDialog()">
</body>
</html>

```

Additional steps

Capture an authorized Affirm payment

Use the transaction id to capture an Affirm Sale transaction: <https://verifone.cloud/docs/online-payments/payment-actions/capturing-authorisation>.

Refund an Affirm payment

Use the transaction id to refund an Affirm Sale transaction: <https://verifone.cloud/docs/online-payments/payment-actions/refunds>.

Perform a Void on an Affirm payment

Use the transaction id to void an Affirm Sale transaction: <https://verifone.cloud/docs/online-payments/payment-actions/void>.