Managing Split Capture Transactions

Learn about managing transaction states in your payment gateway.

Overview

Transactions represent Affirm loans issued to the end-user. You can interact with transactions via the Transaction API to change that transaction's state, update metadata, or retrieve details.

Authorization

Authorization occurs after a user completes the Affirm checkout flow and returns to the merchant site. Authorizing a transaction generates a `transaction_id` (i.e. A1B2-C3D4), that you'll use to reference this loan moving forward.

The Authorize resource endpoint creates a loan and reserves the funds. When you authorize, Affirm will generate a `transaction_id` that you'll use to reference the transaction moving forward. You must authorize a transaction to fully create it.

❗ **Note**

> If you don't authorize a charge, it won't be considered active. This means the user won't see the loan, and you won't be able to capture the funds. For this reason, you should `authorize` the loan as soon as you receive a `checkout` token.

cURL

```
curl https://sandbox.affirm.com/api/v1/transactions \
     -X POST \
     -u "<public_api_key>:<private_api_key>" \
     -H "Content-Type: application/json" \
     -d '{"transaction_id": "<checkout_token>","order_id": "JKLM4321"}'
```

You should receive a response that looks like this with the `transaction_id`.

JSON

```
{
   "amount": 6100,
   "amount_refunded": 0,
   "authorization_expiration": "2019-01-01T00:00:00Z",
   "checkout_id": "A1B2C3D4E5F6G7H8",
   "created": "2019-01-01T00:00:00Z",
   "currency": "USD",
   "id": "A1B2-C3D4",
   "order_id": "JKLM4321",
   "provider_id": 1,
   "status": "authorized"
}
```

After successfully authorizing a transaction and receiving the authorized amount, the user will be redirected to the `user_confirmation_url` page on your site, which will do the following:

- Validate that the authorized amount equals the order total
- Redirect the customer to the order confirmation page or display an order confirmation message
- Store the `transaction_id`
- Mark the order payment as pending

If the authorization fails, the user will be redirected to the `user_cancel_url` page on your site, which will do the following:

- Redirect the customer to your order error page or display that the order was incomplete.
- Potentially store this checkout attempt, which isn't required on our end.

📘**Note**

> You should only authorize an Affirm loan once for the entire amount of the the transaction being purchased.

Managing transactions

You can integrate the `Transaction` endpoints into your back-end order management system, where you normally fulfill orders and process payments.

The Transaction API allows you to manage the transaction through various states such as authorized, captured, and voided.

- Capture transactions
- Void transactions
- Refund transactions

You can use the read and update states if you want to read or update loan information.

- Read transactions
- Update transactions

Capture a transaction

After fulfilling an order, you must send a Capture Transaction request to Affirm in order to capture or settle the funds. You'll want to perform this activity from your secure back-end systems. You'll need the charge_id provided in the Authorization API response to capture a previous authorization. There aren't any required fields that need to be stored from the `Capture` response.

Capturing funds is similar to capturing a credit card transaction. After capturing the loan, we do the following:

- Notify the customer that the loan has been captured and that their first payment will be due in the future.
- Pay the merchant within 2-3 business days.

For Split Captures, you may `Capture` the loan in your desired increments by specifying the `amount` parameter for each unique capture request within the authorization window.

❗ Amount

> If you don't pass an amount in the capture in the `Capture` request, the entire amount will be captured.

Authorization window

All captures must occur within the set authorization window. Once the auth window expires, uncaptured amounts will be refunded to the customer's loan, and will no longer be capturable. Work with your technical counterpart to ensure the auth window is set appropriately per your shipping timeframes. The default auth window is otherwise set to 30 days.

cURL

```
curl
https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":"50000", "order_id": "{order_id}", "shipping_carrier":
"USPS", "shipping_confirmation": "1Z23223"}'

You will then receive back a response with the confirmation.
```

JSON

```
{
 "fee": 600,
 "created": "2016-03-18T00:03:44Z",
 "order_id": "JKLM4321",
 "currency": "USD",
 "amount": 50000,
 "type": "capture",
 "id": "O5DZHKL942503649",
 "transaction_id": "6dH0LrrgUaMD7Llc"
}
```

Multiple captures, refunds, and voids

To capture funds in increments, send a Capture Transaction API request for the specified amount within the authorization window. For example, if your authorization window is 30 days, you can make a unique `Capture` request for each item in the order within that 30 day period. You can also use the Void endpoint to void uncollected funds, and the Refund endpoint to process refunds for partially captured funds.

Examples

Below we'll cover some examples of common Split Capture use cases and walk you through submitting multiple `Capture` requests. These examples follow a fictional business, Prometheus Furniture, that sells home furnishings and appliances. Due to the high demand for their furniture, they typically ship items for multiple orders at different times.

Example 1: Capture all funds

Prometheus Furniture just fulfilled an order for a sofa priced at $400 and a computer desk priced at $600 for a total of $1000. In the example below, the sofa is ready to ship immediately, while the computer desk will take longer.

First, they'll authorize the loan for the full $1000.

Authorization

```
curl https://sandbox.affirm.com/api/v1/transactions \
     -X POST \
     -u "<public_api_key>:<private_api_key>" \
     -H "Content-Type: application/json" \
     -d '{"transaction_id": "<checkout_token>","order_id": "JKLM4321"}'
```

Next, Prometheus Furniture sends a `Capture` request specifying the `amount` of $400 on Day 1 of the loan.

Day 1

```
#First capture for $400
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":40000, "order_id": "JKLM4321", "shipping_carrier": "USPS",
"shipping_confirmation": "1Z23223"}'
```

26 days later, when the computer desk is ready to ship, Prometheus Furniture submits another capture request specifying the remaining amount of $600.

Day 26

```
#Second capture for $600
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":60000, "order_id": "JKLM4321", "shipping_carrier": "USPS",
"shipping_confirmation": "1Z23223"}'
```

Example 2: Capture less than the full amount with a partial void

Sometimes, customers request a loan greater than the amount they spend due to a canceled item or tax adjustment. In the example below, a customer takes out a $1000 loan, but orders two bar stools for $300, a patio furniture for $500, and a chair for $200. After authorizing the full loan amount, Prometheus captures the funds for the bar stools, specifying $300.

Day 1

```
#First capture for $300
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":30000, "order_id": "JKLM4421", "shipping_carrier": "USPS",
"shipping_confirmation": "1Z23223"}'
```

When the patio furniture is ready to ship on day 13, they'll capture the second amount for the patio furniture.

```
Day 13
#Second capture for $500
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":50000, "order_id": "JKLM4421", "shipping_carrier": "USPS",
"shipping_confirmation": "1Z23223"}'
```

The chair that was initially ordered was canceled before being shipped. As a result, Prometheus Furniture wouldn't capture the $200, but rather submit a `Void` request within the authorization window for the remaining amount.

Day 20

```
#Void $200
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/void \
    -X POST \
    -H "Content-Type: application/json" \
    -u "{public_api_key}:{private_api_key}" \
    -d '{"amount":20000}'
```

Example 3: Captures all funds with a refund

In this example, the customer decided to purchase a new rug for $200 and a new refrigerator for $800. After the initial authorization, Prometheus Furniture sends the first capture request for $200.

Day 1

```
#First capture for $200
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":20000, "order_id": "JKLM5431", "shipping_carrier": "USPS",
"shipping_confirmation": "1Z23223"}'
```

Prometheus can get the fridge to the customer a little earlier than normal and captures the second amount for $800 on day 4.

Day 4

```
#Second capture for $800
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":80000, "order_id": "JKLM5431", "shipping_carrier": "USPS",
"shipping_confirmation": "1Z23223"}'
```

The customer was excited to get their items quickly but didn't quite like how the rug clashed with the colors in their living room. Luckily, Prometheus Furniture has a great refund policy and initiated a refund for the customer when they found out. This time, they'll use the Refund API to credit the $200 back to the customer.

```
Day 21
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/refund \
    -X POST \
    -H "Content-Type: application/json" \
    -u "{public_api_key}:{private_api_key}" \
    -d '{"amount": 20000}'
```

Although Prometheus Furniture ended up capturing the full loan, they were able to refund the requested amount back to the customer which will be applied at the end of the loan.

Example 4: Partially captured loan needs to be voided and refunded

For this example, a customer ordered a dining table for $1000 and 6 dining chairs for $1200. Prometheus Furniture then captures the first $1000 for the dining table.

Day 1

```
#First capture for $1000
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/capture \
     -X POST \
     -u "{public_api_key}:{private_api_key}" \
     -H "Content-Type: application/json" \
     -d '{"amount":100000, "order_id": "JKLM5555", "shipping_carrier":   "USPS",
"shipping_confirmation": "1Z23223"}'
```

Once the dining table arrives at the customer's residence, they realize it's too large for their dining area and decide to return it. Now that the table is returned, there is no need for the dining chairs and the customer requests to cancel the unfulfilled order. To cancel the order, the first step is to void the additional authorized but uncaptured amount of $1200.

Day 7

```
#Voiding the authorized but uncaptured loan amount of $1200
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/void \
    -X POST \
    -H "Content-Type: application/json" \
    -u "{public_api_key}:{private_api_key}" \
    -d '{"amount":120000}
```

After voiding the authorized but uncaptured transaction amount of $1200, the next step is to refund the captured loan amount of $1000 for the dining table.

Day 7

```
#Refunding the captured loan amount of $1000
curl https://sandbox.affirm.com/api/v1/transactions/{transaction_id}/refund \
    -X POST \
    -H "Content-Type: application/json" \
    -u "{public_api_key}:{private_api_key}" \
    -d '{"amount": 100000}'
```

Because the loan had been partially captured, but the customer decided to return the table and cancel the chairs order, the authorized transaction amount of $1200 needed to be voided first, followed by a refund of $1000 for the partial amount of the loan that had already been captured.