

PayPal First billing agreement transaction

Overview

This tutorial takes you through the steps required to do your first transaction using PayPal billing agreement API with Verifone.

What is a billing agreement and how does it work?

Billing agreement API enables a customer and a merchant to enter a contract.

The merchant can bill a customer any time, without requiring a customer to login and authorize the transaction.

With a Billing Agreement the amount and the frequency of transactions are not fixed.

Examples

- An agreement to pay eBay seller fees for listing items on eBay
- An agreement between Uber and customer

In this scenarios, a customer gives their agreement only once, and later they do not have to authorize every time to PayPal. Billing the customer is done completely by merchant's side.

Step 1: Set up

Before making a transaction, you need:

- A PayPal account
- A valid JWT Token received from the entity service
- paymentProviderContract. It is created as part of onboarding merchant into Verifone via the Common Portal.

Step 2: Create a billing agreement token

First, we need to call

POST request URL: `/billingAgreement/initiate`

Body:

```
{
  "ppcUid": "21261b6f-07df-4b63-3f4f-7fa0b83a20db",
  "description": "Stored PayPal account with The Partner merchant",
  "enableAddressChange": true,
  "returnUrl": "https://68f8497efb9ce8aeef3ed419c6ef0597.m.pipedream.net/success",
  "cancelUrl": "https://68f8497efb9ce8aeef3ed419c6ef0597.m.pipedream.net/cancel"
}
```

Response:

```
{
  "approvalUrl": "https://www.sandbox.paypal.com/agreements/approve?ba_token=BA-29L89918T33851208",
  "billingToken": "BA-29L89998T33851208"
}
```

Step 3: Approve token

Copy the approval Url from the step 2. Open the link in a browser. Login and approve the billing agreement token.

Note: Store billingToken as you will need it in step 4 for creating a billing agreement.

Step 4: Create a billing agreement

After approving with PayPal, use POST request with PPC and billingToken to create a Billing agreement.

/billingAgreement/create

Body:

```
{
  "billingToken": "BA-29L89998T33851208",
  "ppcUid": "21261b6f-07df-4b63-3f4f-7fa0b83a20db"
}
```

Response:

```
{
  "agreementId": {
    "id": "B-1WW844616B620123",
    "state": "ACTIVE",
    "description": "Stored PayPal account with The Partner merchant",
    "merchant": {
      "payeeInfo": {
        "email": "merchant@business.example.com"
      }
    },
    "payer": {
      "payerInfo": {
        "email": "payer@business.example.com",
        "firstName": "John",
        "lastName": "Doe",
        "payerId": "VJRXDDV4UWZ2W"
      }
    },
    "plan": {
      "merchantPreferences": {}
    },
    "createTime": "2021-03-03T11:56:27.000Z",
    "updateTime": "2021-03-03T11:56:27.000Z"
  }
}
```

Note: You need to store the billing agreement ID, as you need it for making subsequent transactions.

Step 5: Integrate PayPal Risk Mitigation components

FraudNet is a JavaScript library developed by PayPal and embedded into a merchant's web page to collect browser-based data to help reduce fraud. Upon checkout, these data elements are sent directly to PayPal Risk Services for fraud and risk assessment.

To integrate FraudNet, embed a short code snippet in the merchant website and add a custom header to the Verifone call. See [Integrating FraudNet](#).

Please note that FraudNet is for desktop browsers only. For risk analysis data gathered on mobile devices, please refer to [Magne's documentation](#).

Step 6: Set Transaction Context Overview

The Set Transaction Context API is a standalone service that works in conjunction with payment APIs. Use this API to set the transaction context to send additional data about a customer to PayPal before a customer transaction is processed. PayPal uses this data to complete a pre-transaction risk management evaluation.

See [PayPal Risk Analysis](#).

Step 7: Create a transaction

Create a transaction that later will be used for capturing using billing agreement.

Note: Send the `trackingId` used in the Set transaction context API, and the Risk mitigation components in the header `paypalFraudId`. This would enable PayPal to make a decision by combining the information collected using Risk Mitigation components (FraudNet and Magne's), the Set transaction context API, and the create transaction API.

/transactions

Body:

```
{
  "context": {
    "paymentProviderContract": "26761b6f-07df-4b63-9f4f-7fa0b83a20db"
  },
  "intent": "AUTHORIZE",
  "amount": {
    "currencyCode": "USD",
    "value": 20000
  },
  "applicationContext": {
    "returnUrl": "https://68f8497efb9ce8aef3ed419c6ef0597.m.pipedream.net/success",
    "cancelUrl": "https://68f8497efb9ce8aef3ed419c6ef0597.m.pipedream.net/cancel"
  },
  "shipping": {
    "address": {
      "country": "IN",
      "postalCode": "91",
      "countrySubdivision": "IN-MH",
      "city": "yyy",
      "addressLine1": "add1",
      "addressLine2": "add2"
    }
  }
}
```

```
{
  "fullName": "JamesSmith"
}
```

Response:

```
{
  "id": "e432ddb2-82a8-4213-a7da-8a7c0a3952de",
  "status": "INITIATED",
  "orderId": "1W065188KT702530J",
  "createdAt": "2021-03-25T13:56:27Z",
  "approvalUrl": "https://www.sandbox.paypal.com/checkoutnow?token=1W065188KT702530J"
}
```

Step 8: Authorize transaction by billing agreement

Use transaction ID from step 5 and agreement ID from step 4.

```
/transactions/{TRANSACTION_ID}/authorize
```

Body:

```
{
  "agreementId": "B-1WW844616B620123"
}
```

Response:

```
{
  "id": "e123ddb2-82a8-4213-a7da-8a7c0a3952de",
  "authorizationId": "2JM15603JS454230B",
  "createdAt": "2021-03-25T14:00:46Z",
  "expiresAt": "2021-04-23T14:00:46Z",
  "status": "AUTHORISED",
  "payer": {
    "payerId": "VJRXDDV4UWZ4W",
    "shippingAddress": {
      "country": "IN",
      "postalCode": "400097",
      "countrySubdivision": "Maharashtra",
      "city": "Mumbai",
      "addressLine1": "Flat no. 507 Wing A Raheja Residency",
      "addressLine2": "Film City Road"
    }
  }
}
```

Step 9: Capture transaction by billing agreement

Use transaction ID from step 5 and agreement ID from step 4.

```
/transactions/{TRANSACTION_ID}/capture
```

Body:

```
{  
  "agreementId": "B-1WW844616B620123"  
}
```

Response:

```
{  
  "id": "9512303c-c7a2-4417-8cdd-92a617890053",  
  "captureId": "6CB11430732996717",  
  "createdAt": "2021-03-25T14:07:23Z",  
  "isFinalCapture": true,  
  "status": "AUTHORISED"  
}
```