STARTBUTTON API DOC

Copy

Webhook

Webhooks allow you to set up a notification system that can be used to receive updates on certain requests made to the Startbutton API.

With webhooks, Startbutton sends updates to your server when the status of your request changes. You will typically listen to these events on your webhook URL - a `POST` endpoint, the URL needs to parse a JSON request and return a `200 OK`.

We send webhooks for the following transaction types:

1. Collections
2. Transfers
3. Conversions

Supported events for payment collection:

1. collection.verified
2. collection.completed

Verified | Completed


```
{
  "event": "collection.verified",
  "data": {
    "transaction": {
      "_id": "65042a1a0d32920xxxxxxxxx",
      "transType": "collection",
      "status": "verified",
      "merchantId": "64c7bd870821e83xxxxxxxxx",
      "transactionReference": "be6eaxxxxxxx",
      "customerEmail": "test@customer.com",
      "userTransactionReference": "aedxxxx",
      "paymentCode":
"a78a18df1fdc1fa7f2fb665b54afa21cf1c0d81b741f44527989f492e2a60
94cdb24281fb7d577c4636978xxxxxxxxxx",
      "isRecurrent": false,
      "postProcess": null,
      "gatewayReference": null,
      "createdAt": "2023-09-15T09:55:38.492Z",
      "updatedAt": "2023-09-15T09:57:30.522Z",
      "feeAmount": 115500,
      "narration": "Approved",
      "amount": 1030000,
      "currency": "ZAR"
    },
    "authorizationCode": null
  }
}
```


`amount` is in fractional unit

Supported events for transfer:

- transfer.pending
- transfer.successful
- transfer.failed
- transfer.reversed

Successful | Successful (... | Pend... | Fai... | Rever...


```
{
  "event": "transfer.successful",
  "data": {
    "transaction": {
      "_id": "65042e420d3292066xxxxxxx",
      "transType": "transfer",
      "status": "successful",
      "feeAmount": 15,
      "merchantId": "64c7bd870821e831xxxxxxxx",
      "transactionReference": "6342d3xxxxxx",
      "isRecurrent": false,
      "gatewayReference": "6342d3xxxxxx",
      "createdAt": "2023-09-15T10:13:22.438Z",
      "updatedAt": "2023-09-15T10:13:25.212Z",
      "amount": 5000,
      "currency": "NGN",
      "recipient": {
        "recipientName": "JOHN DOE JAMES",
        "currency": "NGN",
        "institutionType": "nuban",
        "institutionName": "Zenith Bank",
        "institutionNumber": "2007xxxxxx",
        "_id": "64f4fb7e605c5280xxxxxxx"
      }
    },
    "authorizationCode": null
  }
```


Supported events for Conversion :

1. conversion.successful
2. conversion.pending
3. conversion.failed

Successful | Pending | Failed


```
{
  "event": "conversion.successful",
  "data": {
    "transaction": {
      "_id": "66xxxxxxxxxxxx",
      "transType": "conversion",
      "status": "successful",
      "fromAmount": 1000,
      "toAmount": 1593050,
      "fromCurrency": "USD",
      "toCurrency": "NGN",
      "merchantId": "65xxxxxxxxxx",
      "transactionReference": "dbxxxxxxx",
      "isRecurrent": false,
      "createdAt": "2024-08-25T23:44:33.513Z",
      "updatedAt": "2024-08-25T23:44:37.346Z",
      "amount": 100000,
      "currency": "USD",
      "feeAmount": null
    },
    "authorizationCode": null
  }
}
```


Webbook Samples for Under and Overpaid transactions:

> Overpayment webhook behavior:

Underpayment | First Webhook for Ov... | Second Webhook for ...


```
{
  "event": "collection.verified",
  "data": {
    "transaction": {
      "_id": "67d946xxxx",
      "transType": "collection",
      "status": "successful",
      "merchantId": "64xxxxxxxxxxx",
      "transactionReference": "09026725031811120xxxxxxxx",
      "customerEmail": "test@customer.com",
      "paymentPartnerId": "65fbxxxxxxx",
      "isRecurrent": false,
      "postProcess": null,
      "createdAt": "2025-03-18T10:12:08.298Z",
      "updatedAt": "2025-03-18T10:12:08.298Z",
      "amount": 20000,
      "currency": "NGN",
      "feeAmount": null
    },
    "authorizationCode": null,
    "extraInformation": {
      "originalReference": "c066f854fa8a",
      "userTransactionReference": "BTS2013",
      "expectedAmount": 200000,
      "paymentCollectionType": "UNDERPAYMENT",
      }
    "payerInformation": {
      "sessionId": "090267250xxxxxxxxx",
      "accountNumber": "**********",
      "bankName": "KUDA MICROFINANCE BANK"
    }
  }
}
```

If we have any issues sending you a webhook, we retry the webhook 5 times.

If a `reference` is passed when initiating a transaction; a `userTransactionReference` will be returned in your webhook for complete transactions as well as under or overpayments.

The `userTransactionReference` will have the value passed in as reference during initialization

Webhook Verification

To ascertain that the request you received on your webhook is legit and not a bad actor, it's recommended that the webhook response is verified.

Events sent from Startbutton carry the `x-startbutton-signature` header. The value of this header is a `HMAC SHA512` signature of the event payload signed using your secret key. Verifying the header signature should be done before processing the event.

Here is a sample code showing the webhook verification

Node (Express) | Java... | Py... | Ruby ... | P...

```
var crypto = require('crypto');
var secret = process.env.MERCHANT_SECRET_KEY;
// Using Express
app.post("/my/webhook/url", function(req, res) {
    //validate event
    const hash = crypto.createHmac('sha512',
secret).update(JSON.stringify(req.body)).digest('hex');
    if (hash == req.headers['x-startbutton-signature']) {
    // Retrieve the request's body
    const event = req.body;
    // Do something with event
    }
    res.send(200);
});
```

Final Notes

To wrap up your webhook implementation, here is the ideal flow your webhook endpoint should follow

1. Verify request is from us using secret key and payload sent
2. Re-query transaction status after hit
3. Send 200 status back immediately and handle complex logic on your end
4. Ensure webhook responses are idempotent. Save the response and ensure value isn't give twice. since you can get multiple calls for a transaction.

Previous
IP Whitelisting

Next
Transaction Status

Last updated 19 days ago