

Signature Verification

CONTENTS

Table of Contents

Signature Verification	1
Overview	1
Related Documentation	1
Return URL (browser response)/Inquiry API	2
Webhook Response	4

Overview

This documentation explains how to verify signature which is sent in:

1. Return URL (browser response)
2. Inquiry API
3. Webhook response

Related Documentation

This guide should be used together with the additional documents as described below.

Document	Description
HashGeneration.pdf	Logic and algorithm to generate the signature.

Return URL (browser response)/Inquiry API

1. Sample response received:
 - dia_secret is the parameter where the signature is sent which will be used to verify in the further steps.

```
{
  "merchant_id": "106598",
  "merchant_access_code": "4a39a6d4-46b7-474d-929d-21bf0e9ed607",
  "unique_merchant_txn_id": "TestNode3222",
  "pine_pg_txn_status": "4",
  "txn_completion_date_time": "18/03/2024 04:44:49 PM",
  "amount_in_paisa": "1000",
  "txn_response_code": "1",
  "txn_response_msg": "SUCCESS",
  "acquirer_name": "BILLDESK",
  "pine_pg_transaction_id": "14635747",
  "captured_amount_in_paisa": "1000",
  "refund_amount_in_paisa": "0",
  "payment_mode": "3",
  "mobile_no": "",
  "udf_field_1": "",
  "udf_field_2": "",
  "udf_field_3": "",
  "udf_field_4": "",
  "Acquirer_Response_Code": "0300",
  "Acquirer_Response_Message": "DEFAULT",
  "parent_txn_status": "",
  "parent_txn_response_code": "",
  "parent_txn_response_message": "",
  "dia_secret": "156A7BD91DCC0A7BD9D080FDC900581A7BC65D8B17A535E24CE6A042B93DF7C9",
  "dia_secret_type": "SHA256"
}
```

2. Removal of parameters:
 - The following parameters have to be excluded from the payload before moving to the next step
 - o dia_secret
 - o dia_secret_type

```
{
  "merchant_id": "106598",
  "merchant_access_code": "4a39a6d4-46b7-474d-929d-21bf0e9ed607",
  "unique_merchant_txn_id": "TestNode3222",
  "pine_pg_txn_status": "4",
  "txn_completion_date_time": "18/03/2024 04:44:49 PM",
  "amount_in_paisa": "1000",
  "txn_response_code": "1",
  "txn_response_msg": "SUCCESS",
  "acquirer_name": "BILLDESK",
  "pine_pg_transaction_id": "14635747",
  "captured_amount_in_paisa": "1000",

```

```

"refund_amount_in_paisa": "0",
"payment_mode": "3",
"mobile_no": "",
"udf_field_1": "",
"udf_field_2": "",
"udf_field_3": "",
"udf_field_4": "",
"Acquirer_Response_Code": "0300",
"Acquirer_Response_Message": "DEFAULT",
"parent_txn_status": "",
"parent_txn_response_code": "",
"parent_txn_response_message": "",
}

```

3. Sorting the payload

- The payload has to be sorted into alphabetical order
- Sample sorted payload:

```

Acquirer_Response_Code=0300
Acquirer_Response_Message=DEFAULT
acquirer_name=BILLDESK
amount_in_paisa=1000
captured_amount_in_paisa=1000
merchant_access_code=4a39a6d4-46b7-474d-929d-21bf0e9ed607
merchant_id=106598
mobile_no=
parent_txn_response_code=
parent_txn_response_message=
parent_txn_status=
payment_mode=3
pine_pg_transaction_id=14635747
pine_pg_txn_status=4
refund_amount_in_paisa=0
txn_completion_date_time=18/03/2024 04:44:49 PM
txn_response_code=1
txn_response_msg=SUCCESS
udf_field_1=
udf_field_2=
udf_field_3=
udf_field_4=
unique_merchant_txn_id=TestNode3222

```

4. Convert the payload into & separated string

```

Acquirer_Response_Code=0300&Acquirer_Response_Message=DEFAULT&acquirer_name=BILLDESK&amount_in_paisa=1000&captured_amount_in_paisa=1000&merchant_access_code=4a39a6d4-46b7-474d-929d-21bf0e9ed607&merchant_id=106598&mobile_no=&parent_txn_response_code=&parent_txn_response_message=&parent_txn_status=&payment_mode=3&pine_pg_transaction_id=14635747&pine_pg_txn_status=4&refund_amount_in_paisa=0&txn_completion_date_time=18/03/2024 04:44:49 PM&txn_response_code=1&txn_response_msg=SUCCESS&udf_field_1=&udf_field_2=&udf_field_3=&udf_field_4=&unique_merchant_txn_id=TestNode3222

```

5. Hashing the payload

- Pass the above payload through SHA256 algorithm along with the MID secret to generate the signature.

```
156A7BD91DCC0A7BD9D080FDC900581A7BC65D8B17A535E24CE6A042B93DF7C9
```

6. Match the generated signature with the received signature.

Webhook Response

1. Sample response received:
 - X-verify is the parameter in the headers where the signature is sent which will be used to verify in the further steps.

```
x-verify{{ FF0014009BE78864DA6880349F1F2D273DE6920B4480B65C3EF8D20A76990409}}
```

```
{
  "event_name": "payment.captured",
  "merchant_response": {
    "merchant_id": "113484",
    "merchant_access_code": "7f532770-f8a7-46f8-a463-182727a29350",
    "unique_merchant_txn_id": "104943038807791693",
    "pine_pg_txn_status": "4",
    "txn_completion_date_time": "29/11/2023 12:18:49 PM",
    "amount_in_paisa": "20000",
    "txn_response_code": "1",
    "txn_response_msg": "SUCCESS",
    "acquirer_name": "HDFC",
    "pine_pg_transaction_id": "7831007",
    "captured_amount_in_paisa": "20188",
    "refund_amount_in_paisa": "0",
    "payment_mode": "CREDIT_DEBIT_CARD",
    "parent_txn_status": "",
    "parent_txn_response_code": "",
    "parent_txn_response_message": "",
    "masked_card_number": "*****1112",
    "card_holder_name": "mojiz",
    "salted_card_hash": "B6B6A7CE1E6E2AA0DD7C028385446A3BBADCEE026A283859C69F5D2B8CC645AD",
    "rrn": "425847096720",
    "auth_code": "999999"
  }
}
```

2. Convert the above payload into a without spaces:

```
{
  "event_name":"payment.captured","merchant_response":{"merchant_id":"113484","merchant_access_code":"7f532770-f8a7-46f8-a463-182727a29350","unique_merchant_txn_id":"104943038807791693","pine_pg_txn_status":"4","txn_completion_date_time":"29/11/2023 12:18:49 PM","amount_in_paisa":"20000","txn_response_code":"1","txn_response_msg":"SUCCESS","acquirer_name":"HDFC","pine_pg_transaction_id":"7831007","captured_amount_in_paisa":"20188","refund_amount_in_paisa":"0","payment_mode":"CREDIT_DEBIT_CARD","parent_txn_status":"","parent_txn_response_code":"","parent_txn_response_message":"","masked_card_number":"*****1112","card_holder_name":"mojiz","salted_card_hash":"B6B6A7CE1E6E2AA0DD7C028385446A3BBADCEE026A283859C69F5D2B8CC645AD","rrn":"425847096720","auth_code":"999999"}}
}
```

3. Convert the payload into base64 format:

eyJldmVudF9uYWY1IjJoicGF5bWVudC5jYXB0dXJlZClslm1lcmNoYW50X3Jlc3Bvbmliljpb7lm1lcmNoYW50X2kljoiMTeZNdg0liwibWVyYzhhbncRfYWNjZXNzX2NvZGUoIl3jZUzUjc3MC1mOGE3LTQ2ZjgtYTQ2My0xODI3MjdhdHJkZnTAilCj1bmldWVfbWVyY2hhbnRfdHhuX2kljoiMTA0OTQzMdMA0DA3NzkxNkZklwiicGluZV9wZW19OeG5fc3Rh dHVzljoiNClslnR4bl9jb21wbGV0aW9uX2X2Rh dGvfddGltZSI6jl5LzExLzlWmJjMgMTI6MT6Tg6NDkgUE0iLCJhbW91bnRfaW5fcGFpc2E0iioyMDAwMCIslsnR4bl9yZXNwb25zZV9t c2ciOiJVNU DRVN TIiwiYW NxdWlyZXJfbm FtZSI6khErKMiLCJwaW5lX3BnX3RyYw5zYWNOaW9uX2kljoiNzg zMTAwNyIsImNhcHR1cmVkX2Ftb3VudF9pbl9wYWlzYSI6jlwMTg4liiwicVmdW5kX2Ftb3VudF9pbl9wYWlzYSI6jlAlJCJwXltZV50X21vZGUoIlDUKVsEvSRfREVCVSrRfQ0FSRClsInBhcnVudF90eG5fc3Rh dHVzljoiiliwicGfyZV50X3R4bl9yZXNwb25zZV9jb2RlljoiiliwicGfyZV50X3R4bl9yZXNwb25zZV9tZXNzYWNdlloiliwibWFza2VkX2NhcmRhdhcnVtYmVylloijQwKioqKioqKioqMTexMilsImNhcmlRfaG9sZGVYX25hbWUIOiJlb2ppeilSl nNhbHRIZF9jYXJkX2hhc2giOiJCNkl2QTdT RTFFNk UyQU EwREQ3QzAyODM4NTQONk EzQkJBRENFRTAyNkEyODM4NTIDNjIGN UqY QjhDQzY0NUFEliwcnJuljo iNDI1ODQ3MDk2NzlwliwiYXV0aF9jb2RlljoiOTk5OTk5In19

4. Hashing the payload

- Pass the base64 payload through SHA256 algorithm along with the MID secret to generate the signature.

FF0014009BE78864DA6880349F1F2D273DE6920B4480B65C3EF8D20A76990409

- Match the generated signature with the received signature.