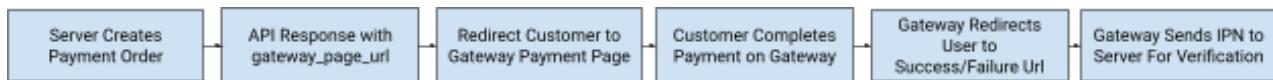


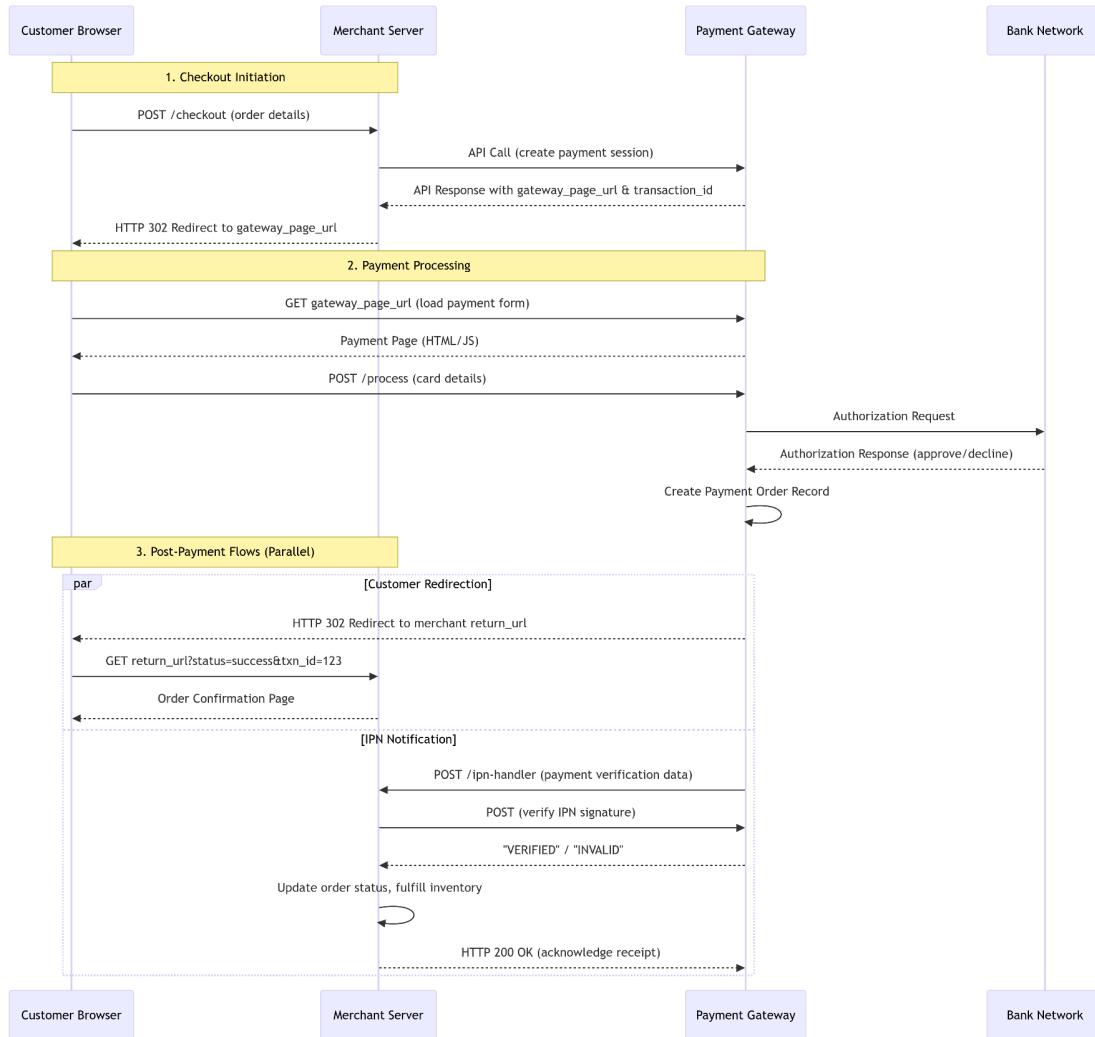
[Sample] Payment Gateway API Integration Documentation

This document outlines the process for integrating with the Payment Gateway API to process payment orders. The integration involves sending a detailed order request to the gateway and then redirecting the customer to a hosted payment page to complete the transaction.

Integration Workflow

The payment process follows a standard server-side integration flow:





1. Initiate Payment Order

This is the primary API call to create a payment session. You will send detailed order, customer, and product information in a single **POST** request.

- **Endpoint:** `POST /payment/api/v1/p/service/api/payment/processing/payment-order`
- **Base URLs:**
 - Staging: <https://api-stage.tnextpay.com>
 - Local: <http://192.168.169.162:8094> (for local development)
- **Content-Type:** `application/json`

Authentication

Authentication is handled directly within the Header body. You must generate **SIGNATURE** and **DIGEST** alone with the following headers

Field	Data Type	Description	Required
X-TNPG-TIMESTAMP	string	Your gateway API username.	Yes
X-TNPG-HOST	string	Your gateway API password.	Yes
X-TNPG-TARGET-API	string	A unique signature for request validation.	Yes
X-TNPG-MERCHANT-ID	string	Your gateway Merchant ID	Yes
X-TNPG-API-KEY	string	Your gateway API Key.	Yes
X-TNPG-SIGNATURE	string	Generated Signature using api key and api secret	Yes
X-TNPG-DIGEST	string	Generated Digest using api key and api secret	Yes

Request Body

The request body is a comprehensive JSON object containing all necessary information for the transaction.

Field Name	Parent Object	Data Type	Description	Required
order_id	(root)	string (UUID)	A unique identifier for the order in your system.	Yes
order_information	(root)	object	Contains core payment details.	Yes
payable_amount	order_information	float	The total amount the customer needs to pay.	Yes
currency_code	order_information	string	The 3-letter currency code (e.g., "BDT").	Yes
ipn_url	(root)	string	Instant Payment Notification (IPN) URL. The gateway will send a silent POST request here with the final payment status.	Yes
success_url	(root)	string	URL to redirect the customer to after a successful payment.	Yes

cancel_url	(root)	string	URL to redirect the customer to if they cancel the payment.	Yes
failure_url	(root)	string	URL to redirect the customer to if the payment fails.	Yes
promotion_information	(root)	object	Optional details for payment channel promotions.	No
preferred_channel	promotion_information	string	Preferred payment channel (e.g., "VISA").	No
allowed_bin	promotion_information	string	Comma-separated list of allowed card BINs.	No
discount_detail	(root)	object	Optional details about applied discounts.	No
mdf_1 to mdf_6	(root)	string	Merchant-defined fields for passing custom data.	No
customer_information	(root)	object	Details of the customer making the payment.	Yes
product_information	(root)	object	Details of the product being purchased.	Yes

shipment_information	(root)	object	Details related to the shipment of physical goods.	No
----------------------	--------	--------	--	----

Sample Request Header

```
"X-TNPG-TIMESTAMP: Mon, 09 Feb 2026 07:47:49 GMT"
"X-TNPG-HOST: http://192.168.169.162:8094",
"X-TNPG-TARGET-API: POST /p/service/api/payment/processing/payment-order",
"X-TNPG-MERCHANT-ID: M12345", "X-TNPG-API-KEY: test",
"X-TNPG-SIGNATURE: kh1Z3x+wZYAJxoZ1dMHqXpAfsI2FLiUuj28WcVIsl\A=",
"X-TNPG-DIGEST: SHA-256=3jAQ+6GzudF14RFEvj6KAo1RmTF8D9FJR0Dz5vIhWQ8="
```

Sample Request Body

```
{
    "order_id": "75_1769071647",
    "order_information": {
        "payable_amount": "10.00",
        "currency_code": "BDT"
    },
    "ipn_url": "http://my-ipn-url",
    "success_url": "http://my-success-url",
    "cancel_url": "http://my-cancel-url",
    "failure_url": "http://my-failure-url",
    "promotion_information": {
        "preferred_channel": "",
        "allowed_bin": ""
    },
    "discount_detail": null,
    "mdf_1": "Order #75",
    "mdf_2": "Product Info: {\"product_name\":\"Test\",\"product_category\":\"Uncategorized\"}",
    "customer_information": {
        "name": "asif sanaullah",
        "email": "me.sanaullah.asif@gmail.com",
        "primaryAddress": "Dhaka",
        "secondaryAddress": "",
        "city": "Dhaka",
        "state": "BD-13",
        "postcode": "",
        "country": "Bangladesh",
        "contact_number": "01818482731",
        "fax": "01818482731"
    },
    "product_information": {
        "product_name": "Test",
        "product_category": "Uncategorized"
    }
}
```

2. API Responses

The API will respond differently depending on whether the payment order was created successfully or if there was an error.

Successful Response (200 OK)

If the request is valid, the API returns a JSON object containing the details of the created payment order and, most importantly, the URL to redirect the user to.

Field Name	Parent Object	Description
payment_order_id	order_detail	The unique identifier for the payment order in the gateway's system.
order_status	order_detail	The initial status of the order (e.g., "UNATTEMPTED").
session_id	order_detail	A unique session ID for the payment page.
gateway_page_url	(root)	The URL to which you must redirect the customer to complete the payment.
token	token_response	A JWT token, which may be used for subsequent verification calls.

Sample Successful Response Body:

```
{
  "order_detail": {
    "payment_order_id": "01K9PV56SPWPMNA9A1BRVP5SRF",
    "order_status": "UNATTEMPTED",
    "session_id": "01K9PV56VKQ3MVN8JGA827F7WZ",
    "order_id": "b55e0e9e-7265-4559-94fe-a7058cd66ddb"
  },
  "token_response": {
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWliOiJORDVIVmFtb282dGZxeEQyYIZVT01BYkZyOEt1ZnNOYSIsImIhdCI6MTc2Mjc3NzAxMywiZXhwIjoxNzYyODYzNDEzfQ.bvgX3uTZmHrx_evsxejSK0-h006unluz7Qr3DO4jJKo",
    "type": "Bearer",
    "expiresAt": "2025-11-11T12:16:52Z"
  },
  "gateway_page_url": "https://payment-stage.tnextpay.com/01K9PV56VKQ3MVN8JGA827F7WZ"
}
```

Failed Response (4xx/5xx)

If the request fails due to an error (e.g., validation, duplicate order), the API will return a JSON object with error details.

Field Name	Description
reason	A human-readable explanation of the error.
message	A machine-readable error code (e.g., "DUPLICATE_PAYMENT_ORDER").
statusText	The HTTP status text (e.g., "BAD_REQUEST").
statusCode	The HTTP status code (e.g., 400).

Sample Failed Response Body:

```
{
  "submitTimeUtc": "2026-02-09T07:02:24Z",
  "code": 401,
  "errors": [
    {
      "code": "INVALID_DIGEST",
      "reason": "Invalid digest."
    }
  ],
  "status": "UNAUTHORIZED"
}
```

3. Inquiry of Payment

4. Verification of payment

Verification request and response will be here...

This is the primary API call to verify a payment session. You will send detailed order, customer, and product information in a single **POST** request.

- **Endpoint:** **POST** /payment/api/v1/p/service/api/payment/processing/verify.
- **Base URLs:**
 - Staging: <https://api-stage.tnextpay.com>
 - Local: <http://192.168.169.162:8094> (for local development)

- **Content-Type:** application/json

Authentication

**** No provision yet, but will be added further**

Request Body

The request body is a comprehensive JSON object containing all necessary information for the transaction.

Field Name	Parent Object	Data Type	Description	Required
paymentOrderId	(root)	string	A unique signature for request validation.	Yes

Sample Request Body

Header as Mentioned above

```
{ "paymentOrderId": "01KH0NXZQNNF3CNZN48DKY99JW" }
```

Response Body

The API will respond differently depending on whether the payment order was created successfully or if there was an error.

Successful Response (200 OK)

If the request is valid, the API returns a JSON object containing the details of the created payment order and, most importantly, the URL to redirect the user to.

Field Name	Parent Object	Description	Example
status	transaction_info	Payment Status	1. APPROVED 2. DECLINED 3. CANCELLED 4. FAILED
status_code	transaction_info	Status code of payment	1. 1002

			2. 1003 3. 1004 4. 1005
order_id	transaction_info	Merchant order id sent during create payment	
order_tracking_id	transaction_info	PSO order id align with order_id (Unique)	
transaction_date	transaction_info	Date of transaction Format : Y-m-d His	
bank_transaction_id	transaction_info	Transaction id from payment vendor (i.e. bkash, Visa, MasterCard)	
amount	transaction_info		
currency	transaction_info	Currency chosen in the merchant portal . Now only BDT/USD allowed	
currency_amount	transaction_info	Converted amount (if currency selected other than BDT)	
currency_type	transaction_info		
merchant_amount	transaction_info	Initiated amount by merchant	

Sample Successful Response Body:

```
{
    "status": "SUCCESS",
    "code": 201,
    "data": {
        "transaction_info": {
            "status": "APPROVED",
            "status_code": 1002,
            "order_id": "6989913a70e61",
            "order_tracking_id": "01KH0NYM2YTDRC6XBXP1VT8MPG",
            "transaction_date": "2026-02-09T13:48:48:437 GMT+0600",
            "bank_transaction_id": "DB9300ESCF",
            "amount": 10,
            "currency": "BDT",
            "currency_amount": 10,
            "currency_type": "BDT",
            "merchant_amount": 10
        }
    },
    "submitTimeUtc": "2026-02-09T07:48:51Z"
}
}
```

Sample Transaction not found

```
{
    "reason": "Payment detail not found.",
    "statusCode": 400,
    "statusText": "BAD_REQUEST",
    "message": "PAYMENT_DETAIL_NOT_FOUND"
}
```

5. Integration Best Practices

1. **Server-Side Implementation:** All API calls to initiate the payment must be made from your secure server-side backend. Never expose your API credentials (`username`, `password`, `signature`) in client-side code.
2. **Redirect the Customer:** Upon a successful API response, immediately redirect the user's browser to the `gateway_page_url`. Do not display this URL as a link for the user to click.
3. **Handle IPNs:** The most reliable way to confirm a payment is via the Instant Payment Notification (IPN) sent to your `ipn_url`. Your IPN endpoint should be secure, able to receive POST data, and resilient to network issues (implement logic to handle duplicate or missed notifications).
4. **Use Staging for Testing:** Thoroughly test your integration using the staging environment before going live.
5. **Error Handling:** Implement robust error handling to parse failed API responses and display user-friendly messages on your website. Check the `statusCode` and `message` fields to determine the cause of the failure.

6. Error Code

Error Code	Error Description	Bank code/Description	Remarks