DEVELOPER DOCS

Noble Web Studio

Operator codes | Get API keys

RECHARGE & BILL PAYMENT API · V1

Recharge API Integration Guide

Prepaid mobile, DTH, postpaid, electricity, gas, FASTag and other BBPS bill payments, all through one REST API. Your recharge portal, app or reseller panel sends a JSON request and gets a JSON response, plus a callback when the final status arrives.

BASE URL `https://api.noblewebstudio.com/recharge/v1`

Draft for review. Highlighted values (like this) are sample values. Before publishing, the tech team should confirm the live base URL, header names, field names, and operator/circle codes.

ON THIS PAGE

Quick start | Authentication | Recharge API | Status API | Balance API | Callback URL | Status & error codes | Bill payment fields | Operator codes | Circle codes | Code samples | Go-live checklist | FAQ

Quick start

Most partners finish integration and go live in one to three working days.

STEP 1

Sign the agreement & KYC

Share company KYC documents. You get an API key and a secret on your partner panel.

STEP 2

Whitelist your server

Add your server's static IP and your callback URL in API Settings. Requests from other IPs are rejected.

STEP 3

Test on UAT

Use the UAT base URL. Test success, failure and pending cases, and check that your callback works.

STEP 4

Load wallet & go live

Load wallet & go live

Switch to the live base URL, add funds to your wallet, and start with a ₹10 test recharge.

Authentication

Send credentials in request headers, not in the URL. Query-string credentials end up in server logs and browser history. Every request must come from a whitelisted IP.

HEADER	VALUE	NOTES
`X-Api-Key`	Your API key	Shown on the partner panel
`X-Api-Secret`	Your API secret	Keep it on your server only. Never put it in an app or JavaScript.
`Content-Type`	`application/json`	For all POST requests

Recharge API

Use this endpoint for mobile, DTH, postpaid and bill payment transactions. The operator code decides the service.

`POST` `/recharge`

Request body

FIELD	TYPE		DESCRIPTION	EXAMPLE
`operator_code`	string	REQUIRED	Operator code from the operator list	`JIO`
`number`	string	REQUIRED	Mobile number, DTH subscriber ID or consumer number	`9876543210`
`amount`	number	REQUIRED	Amount in rupees (INR)	`299`

`client_ref_id`	string	REQUIRED	Your unique transaction ID, 6 to 40 characters. Use a new ID for every request.	`NWS20261105001`
`circle_code`	string	OPTIONAL	Telecom circle from the circle list. Recommended for ported (MNP) numbers.	`RJ`
`param1` , `param2`	string	OPTIONAL	Extra biller fields (see bill payment fields)	`110`
`callback_url`	string	OPTIONAL	Overrides the callback URL saved in API Settings for this one request	`https://…/cb`

REQUEST Copy

```
{
  "operator_code": "JIO",
  "number": "9876543210",
  "amount": 299,
  "client_ref_id": "NWS20261105001",
  "circle_code": "RJ"
}
```

RESPONSE · 200 Copy

```
{
  "status": "SUCCESS",
  "code": "TXN_SUCCESS",
  "message": "Recharge successful",
  "txn_id": "NW8823410057",
  "client_ref_id": "NWS20261105001",
  "operator_ref": "JIO7740112983",
  "number": "9876543210",
  "amount": 299,
  "commission": 8.97,
  "balance": 45210.50,
  "timestamp": "2026-11-05T14:32:08+05:30"
}
```

Response fields

FIELD	DESCRIPTION
`status`	`SUCCESS` , `FAILED` or `PENDING` . Your system should act on this field.
`code`	Machine-readable reason (see codes)
`txn_id`	Noble Web Studio transaction ID. Quote it in support tickets.
`client_ref_id`	The ID you sent, returned as is
`operator_ref`	Operator's reference number. Show it to the customer as proof of recharge.

`commission`	Commission credited for this transaction (₹)
`balance`	Wallet balance after this transaction (₹)

Status API

Check the latest status of any transaction using your `client_ref_id`. Use it for pending transactions and daily reconciliation.

```
GET  /status?client_ref_id=NWS20261105001
```

RESPONSE · 200 Copy

```
{
  "status": "PENDING",
  "code": "TXN_PENDING",
  "txn_id": "NW8823410057",
  "client_ref_id": "NWS20261105001",
  "operator_ref": null,
  "amount": 299,
  "created_at": "2026-11-05T14:32:08+05:30",
  "updated_at": "2026-11-05T14:32:09+05:30"
}
```

Rate limit: one status check per transaction every 60 seconds. For final updates, rely on the callback.

Balance API

Returns your current wallet balance. Check it before sending large batches.

```
GET  /balance
```

RESPONSE · 200 Copy

```
{
  "status": "SUCCESS",
  "balance": 45210.50,
```

```
  "currency": "INR",
  "as_of": "2026-11-05T14:35:00+05:30"
}
```

Callback URL

When a `PENDING` transaction becomes final, or a successful recharge is later reversed by the operator, we send a POST request to your callback URL. Return HTTP `200` within 10 seconds. If we don't get it, we retry up to 3 times: after 1, 5 and 15 minutes.

CALLBACK PAYLOAD (POST, JSON) Copy

```
{
  "event": "txn.updated",
  "status": "SUCCESS",
  "txn_id": "NW8823410057",
  "client_ref_id": "NWS20261105001",
  "operator_ref": "JIO7740112983",
  "amount": 299,
  "signature": "c1f9…e07a"
}
```

Verify the signature

`signature` = HMAC-SHA256 of `txn_id|client_ref_id|status|amount`, using your API secret as the key. Reject any callback whose signature doesn't match. That way nobody else can mark your orders successful.

PHP CALLBACK HANDLER Copy

```
// callback.php
$data = json_decode(file_get_contents('php://input'), true);
$base = $data['txn_id'].'|'.$data['client_ref_id'].'|'.$data['status'].'|'.$data['amoun
$expected = hash_hmac('sha256', $base, getenv('NWS_API_SECRET'));

if (!hash_equals($expected, $data['signature'])) { http_response_code(401); exit; }

// 1. Find the order by client_ref_id
// 2. Update only if it is still PENDING (or SUCCESS -> FAILED for reversals)
// 3. On FAILED, refund the retailer's wallet
http_response_code(200);
echo 'OK';
```

Status & error codes

Transaction status

STATUS	MEANING	WHAT YOUR SYSTEM SHOULD DO
SUCCESS	Operator confirmed the recharge	Mark the order successful and save `operator_ref`
FAILED	Recharge rejected; the amount is back in your wallet	Refund your retailer or customer
PENDING	Operator hasn't given a final response yet	Don't retry and don't refund. Wait for the callback or check the Status API.

Error codes

HTTP	CODE	MEANING
401	`AUTH_FAILED`	API key or secret is wrong
403	`IP_NOT_WHITELISTED`	Request came from an IP not added in API Settings
409	`DUPLICATE_REF_ID`	This `client_ref_id` was already used
422	`INVALID_OPERATOR`	Operator code is wrong or the operator is inactive
422	`INVALID_AMOUNT`	Amount isn't a valid plan or is outside the operator's limits
422	`INVALID_NUMBER`	Number format is wrong for this operator
402	`LOW_BALANCE`	Wallet balance is lower than the amount
503	`OPERATOR_DOWN`	Operator is temporarily unavailable. Try again later.
429	`RATE_LIMITED`	Too many requests. Slow down and retry.

Bill payment fields

Bill payment fields

Some billers need more than one identifier. Send these in `param1` and `param2` on the same Recharge endpoint. For electricity and gas, fetch the bill amount first so the customer pays the right amount.

BILLER TYPE	NUMBER	PARAM1	PARAM2
Landline	Landline number	STD code	Account number (if asked)
Electricity (most DISCOMs)	Consumer / K number	—	—
Electricity (MSEDCL)	Consumer number	Billing unit	—
Piped gas	Customer / BP number	Bill group (if asked)	—
Insurance premium	Policy number	Date of birth (DD-MM-YYYY)	—
FASTag	Vehicle number	—	—

Operator codes

Common operators are listed below. The full list (all electricity boards, gas, FASTag issuers, insurance) is in the partner panel under API Settings → Operator List, and can be downloaded as CSV.

OPERATOR	CODE	SERVICE
Jio	`JIO`	Prepaid
Airtel	`AIR`	Prepaid
Vi (Vodafone Idea)	`VI`	Prepaid
BSNL Top-up	`BSNLT`	Prepaid
BSNL Special Tariff	`BSNLS`	Prepaid
Jio Postpaid	`JIOPP`	Postpaid

Airtel Postpaid	`AIRPP`	Postpaid
Vi Postpaid	`VIPP`	Postpaid
Airtel Digital TV	`ADTH`	DTH
Tata Play	`TPDTH`	DTH
Dish TV	`DISH`	DTH
Sun Direct	`SUN`	DTH
Jaipur Vidyut Vitran Nigam (JVVNL)	`JVVNL`	Electricity
Ajmer Vidyut Vitran Nigam (AVVNL)	`AVVNL`	Electricity
Jodhpur Vidyut Vitran Nigam (JdVVNL)	`JDVVNL`	Electricity
UPPCL (Urban)	`UPPCLU`	Electricity
UPPCL (Rural)	`UPPCLR`	Electricity
MSEDCL Maharashtra	`MSEDCL`	Electricity
BSES Rajdhani	`BSESR`	Electricity
Indraprastha Gas	`IGL`	Gas

Circle codes

Two-letter circle codes based on telecom licence areas. Send them in `circle_code`.

CIRCLE	CODE	CIRCLE	CODE
Andhra Pradesh & Telangana	`AP`	Madhya Pradesh & Chhattisgarh	`MP`
Assam	`AS`	Maharashtra & Goa	`MH`
Bihar & Jharkhand	`BR`	Mumbai	`MU`
Delhi NCR	`DL`	North East	`NE`
Gujarat	`GJ`	Odisha	`OR`

Haryana	HR	Punjab	PB
Himachal Pradesh	HP	Rajasthan	RJ
Jammu & Kashmir	JK	Tamil Nadu (incl. Chennai)	TN
Karnataka	KA	UP East	UE
Kerala	KL	UP West & Uttarakhand	UW
Kolkata	KO	West Bengal	WB

Code samples

Replace the key and secret with your own, and use a new `client_ref_id` each time.

PHP | Node.js | Python | cURL

PHP

Copy

```php
$payload = [
  'operator_code' => 'JIO',
  'number'        => '9876543210',
  'amount'        => 299,
  'client_ref_id' => 'NWS' . date('YmdHis') . rand(100, 999),
  'circle_code'   => 'RJ',
];
$ch = curl_init('https://api.noblewebstudio.com/recharge/v1/recharge');
curl_setopt_array($ch, [
  CURLOPT_POST           => true,
  CURLOPT_POSTFIELDS     => json_encode($payload),
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_TIMEOUT        => 90,
  CURLOPT_HTTPHEADER     => [
    'Content-Type: application/json',
    'X-Api-Key: ' . getenv('NWS_API_KEY'),
    'X-Api-Secret: ' . getenv('NWS_API_SECRET'),
  ],
]);
$res = json_decode(curl_exec($ch), true);
curl_close($ch);

switch ($res['status'] ?? 'PENDING') {   // timeout = treat as pending
```

```
    case 'SUCCESS': /* mark success, save operator_ref */ break;
    case 'FAILED':  /* refund retailer */             break;
    default:        /* wait for callback */
  }
```

Go-live checklist

- Use a **new `client_ref_id`** for every request. Reuse it only to check status.
- Set the HTTP timeout to **90 seconds**. A timeout means pending, not failed. Never refund on a timeout.
- Never resend a pending recharge. A second request can recharge the customer twice.
- Verify the callback signature before you update any order.
- Handle reversals: a `SUCCESS` can later become `FAILED` through a callback.
- Log every request and response with `client_ref_id` and `txn_id` for daily reconciliation.
- Keep the API secret on your server. Never put it in an Android app, frontend code or a public Git repo.

FAQ

▶ **Can one API handle mobile, DTH and electricity bills?**