BillMitra24

BillMitra24 API & Webhooks Guide

GST Billing, Inventory & Accounting Software

Edition: November 2026 · English / हिंदी

API & webhooks developer guide

Connect Shopify, Tally, your website, a mobile app or any software to BillMitra24 with the REST API and webhooks.

1. Create an API key

1. Login as Admin → Settings → API & Integrations.
2. Enter a name (e.g. Shopify sync), choose Read only or Read & Write, optional expiry, click Generate key.
3. Copy the key immediately — it is shown only once. Only a secure hash is stored. Revoke it any time.
4. API access must be included in your plan.

2. Basics

- Base URL: `https://YOUR-DOMAIN/api/v1`
- Authentication: header `X-Api-Key: bm24_live_…` (or Authorization: Bearer …)
- JSON in and out. Dates as yyyy-MM-dd. Enums as text (e.g. "UPI", "Customer").
- Lists are paged: page, pageSize (max 200) → { page, pageSize, total, data }.
- Rate limit: 120 requests per minute per key (HTTP 429 when exceeded).
- Errors: `400` validation, `401` missing/invalid key, `403` read-only key or plan without API, `404` not found — body: `{ "error": "...", "message": "..." }`

3. Endpoints

	Path	What it does
GET	`/me`	Your business, plan and key info
GET	`/currencies`	Active currencies with exchange rates
GET	`/bank-accounts`	Cash and bank accounts
GET	`/parties?q=&type=Customer&page=1&pageSize=50`	List / search parties with balance
GET	`/parties/{id}`	One party
GET	`/parties/{id}/ledger?from=&to=`	Party ledger with running balance
POST	`/parties`	Create a party (write)
PUT	`/parties/{id}`	Update a party (write)
GET	`/items?q=&page=1`	List / search items with stock
GET	`/items/{id}`	One item
GET	`/items/{id}/stock-history`	Stock movements of an item
POST	`/items`	Create an item (write)

	Path	What it does
PUT	`/items/{id}`	Update an item (write)
POST	`/items/{id}/stock`	Adjust stock + / - (write)
GET	`/documents?type=sales-invoice&from=&to=&status=&partyId=`	List invoices or any document type
GET	`/documents/{id}`	One document with lines
POST	`/documents`	Create an invoice / quotation / purchase etc. (write)
POST	`/documents/{id}/cancel`	Cancel a document (write)
GET	`/payments?type=In&from=&to=`	List payments
GET	`/payments/{id}`	One payment with allocations
POST	`/payments`	Record a payment in / out (write)
GET	`/reports/summary?from=&to=`	Sales, purchase, expense, receivable & payable totals
GET	`/reports/outstanding`	Party-wise outstanding
GET	`/reports/stock`	Stock summary and value

Document types: `sales-invoice, quotation, sales-return, credit-note, delivery-challan, proforma-invoice, purchase-invoice, purchase-return, debit-note, purchase-order`

In document lines send itemId or itemCode to use the item price, HSN and GST automatically; or send description, rate and taxRate for a free-text line. Stock, party balance, cash/bank and webhooks update exactly like the app.

4. Examples

curl

```
curl -H "X-Api-Key: bm24_live_xxxxxxxx" https://YOUR-DOMAIN/api/v1/me

curl -X POST https://YOUR-DOMAIN/api/v1/documents \
  -H "X-Api-Key: bm24_live_xxxxxxxx" -H "Content-Type: application/json" \
  -d '{
    "type": "sales-invoice",
    "partyId": 12,
    "date": "2026-11-23",
    "lines": [
      { "itemCode": "RUG-58", "quantity": 2 },
      { "description": "Installation", "quantity": 1, "rate": 1500, "taxRate": 18, "hsn":
"9965" }
    ],
    "amountReceived": 20000,
    "paymentMode": "UPI"
  }'
```

C# (.NET)

```
using var http = new HttpClient { BaseAddress = new Uri("https://YOUR-DOMAIN/api/v1/") };
http.DefaultRequestHeaders.Add("X-Api-Key", "bm24_live_xxxxxxxx");

// list unpaid invoices
var json = await http.GetStringAsync("documents?type=sales-invoice&status=Unpaid");

// create a party
var res = await http.PostAsJsonAsync("parties", new {
    name = "Agarwal Handicrafts", mobile = "9414012345", gstin = "08AAGFA1234L1Z9",
stateCode = "08"
});
res.EnsureSuccessStatusCode();
```

JavaScript

```
const api = (path, opts = {}) => fetch("https://YOUR-DOMAIN/api/v1/" + path, {
  ...opts, headers: { "X-Api-Key": "bm24_live_xxxxxxxx", "Content-Type":
"application/json", ...(opts.headers || {}) }
}).then(r => r.json());

const stock = await api("reports/stock");
await api("payments", { method: "POST", body: JSON.stringify({ type: "In", partyId: 12,
amount: 5000, mode: "UPI" }) });
```

5. Webhooks

- Settings → API & Integrations → Webhooks: enter your URL, choose events, Save, then Send test event.
- Events: `document.created`, `document.cancelled`, `payment.created`, `party.created`, `item.created`
- Each POST has headers X-BillMitra24-Event and X-BillMitra24-Signature (HMAC-SHA256 of the body with your signing secret). Failed deliveries are retried 3 times.

```
// C#: verify a webhook (ASP.NET Core)
var body = await new StreamReader(Request.Body).ReadToEndAsync();
var sig = Request.Headers["X-BillMitra24-Signature"].ToString();   // "sha256=..." or hex
using var h = new HMACSHA256(Encoding.UTF8.GetBytes(webhookSecret));
var expected =
Convert.ToHexString(h.ComputeHash(Encoding.UTF8.GetBytes(body))).ToLowerInvariant();
if (!sig.EndsWith(expected)) return Unauthorized();
var evt = Request.Headers["X-BillMitra24-Event"];   // document.created, payment.created
...
```

6. Good practice

- Use a separate key for every app and give read-only where possible.
- Never put the key in browser or mobile app code that users can see — call the API from your server.
- Check Last used and Requests on the API page; revoke unknown keys.