

Snap & Bill

Third-Party Reading Ingestion API

Submit and retrieve meter readings via token-authenticated REST endpoints

Version	Base URL	Format	Date
v1.0	https://api.snapnbill.com/api/v1	JSON only	April 2026

Contents

1. Overview
2. Authentication
3. IP Whitelisting
4. Endpoints — Submit Readings (POST)
5. Endpoints — Retrieve Previous Readings (GET)
6. Idempotency
7. Error Reference
8. HTTP Status Codes
9. Changelog & Contact

1 Overview

The Snap & Bill Reading Ingestion API lets authorised third-party systems push meter readings directly into the platform without requiring a user login, JWT flow, or CSV upload. Each integration receives a long-lived bearer token that is scoped to a specific *service provider* and *site*.

The API supports **partial success** — a single call may accept some readings while rejecting others. Every response contains per-row status information so your system can react individually to each result.

- i All endpoints accept and return **JSON only**. Always send `Content-Type: application/json` and `Accept: application/json`.

2 Authentication

Pass your integration token as a standard HTTP **Bearer token**:

```
Authorization: Bearer <your-integration-token>
```

TOKEN BEHAVIOUR

- Tokens are **long-lived** — no refresh is needed.
- Each token maps to exactly one service provider and one site; those values are derived server-side and do not need to be sent in the request body.
- A token may have an optional expiry date (`expires_at`) configured at creation time.
- Tokens can be **revoked** at any time by your Snap & Bill administrator.

- ⚠ Treat your token like a password. The raw token value is displayed **only once** at creation time and is never stored in plain text. If you lose it, a new token must be issued.

3 IP Whitelisting

Every integration token is associated with one or more allowed source IP addresses or CIDR ranges. Requests arriving from an IP not on the whitelist receive a `403 Forbidden` response, even if the token is otherwise valid.

CIDR EXAMPLE	EFFECT
203.0.113.45	Single IPv4 address
203.0.113.0/24	Entire /24 subnet
0.0.0.0/0	Any IP — useful for testing environments only

 Inform your Snap & Bill administrator of your egress IP address(es) before you go live. Using `0.0.0.0/0` in production is strongly discouraged.

4 Submit Readings

POST

/api/v1/integrations/readings

Submit one or more meter readings

Request body

```
{
  "request_id":      "partner-unique-id-001", // required, your idempotency key
  "service_type_id": 2,                       // required, must be valid for the site
  "readings": [                                // required, non-empty array
    {
      "barcode":      "ABC123456",            // required
      "current_unit": 452.7,                  // required, numeric
      "reading_date": "2026-03-25"           // required, YYYY-MM-DD, not in future
    }
  ]
}
```

FIELD REFERENCE

FIELD	TYPE	REQUIRED	NOTES
request_id	string	required	Your unique identifier for this submission. Used for idempotency — replaying the same <code>request_id</code> returns the original result without reprocessing. Max 255 chars.
service_type_id	integer	required	Must correspond to a valid service type (e.g. 1 = Water, 2 = Electricity). Will be validated against platform data.
readings	array	required	One or more reading objects. Must not be empty. Duplicate barcodes within one request are rejected.
readings[].barcode	string	required	The meter barcode identifier. Must exist in the platform and belong to the site associated with your token.
readings[].current_unit	numeric	required	Current meter reading value. Must be $\geq$ the previous recorded unit for that meter.
readings[].reading_date	date	required	Date of reading in <code>YYYY-MM-DD</code> format. Cannot be a future date.

Success response — all accepted

```
{
  "status": "accepted",
  "request_id": "partner-unique-id-001",
  "accepted_count": 2,
  "rejected_count": 0,
  "results": [
    {
      "barcode": "ABC123456",
      "status": "accepted",
      "open_session_id": 4567
    }
  ]
}
```

Partial success response

```
{
  "status":      "partial_success",
  "request_id":  "partner-unique-id-001",
  "accepted_count": 1,
  "rejected_count": 1,
  "results": [
    {
      "barcode":      "ABC123456",
      "status":      "accepted",
      "open_session_id": 4567
    },
    {
      "barcode":  "ZZZ999",
      "status":  "rejected",
      "code":    "BARCODE_NOT_FOUND",
      "message": "Barcode does not exist"
    }
  ]
}
```

OVERALL STATUS VALUES

VALUE	MEANING	HTTP CODE
accepted	All readings accepted	200
partial_success	Some accepted, some rejected	200
rejected	All readings rejected (or payload-level rejection)	422

5 Retrieve Previous Readings

GET /api/v1/integrations/readings

Fetch last recorded readings for all meters on the site

Returns the most recent recorded unit and date for every meter lot associated with your token's site and service provider. Use this to determine the baseline before submitting new readings.

QUERY PARAMETERS

PARAMETER	TYPE	REQUIRED	NOTES
service_type_id	integer	optional	Filter results to a specific service type. Omit to return all service types for the site.

Response

```
{
  "service_provider_id": 7,
  "service_provider_name": "Pieter Both Syndic Services",
  "site_id": 3,
  "site_name": "Aqua Springs",
  "site_fullname": "SYNDICAT DES COPROPRIÉTAIRES DE AQUA SPRINGS",
  "readings": [
    {
      "barcode": "ABC123456",
      "service_type_id": 2,
      "previous_unit": 448.2,
      "last_reading_date": "2026-02-28"
    }
  ]
}
```

6 Idempotency

The API enforces idempotency at the `(token, request_id)` level. If a request is retried with the same `request_id`, the **original response is returned unchanged** — no duplicate sessions or readings are created.

REPLAYED RESPONSE INDICATOR

An idempotent replay returns the original result verbatim with one extra field added:

```
{
  "status": "accepted",
  "request_id": "partner-unique-id-001",
  "accepted_count": 1,
  "rejected_count": 0,
  "replayed": true,
  "results": [ /* ... original results ... */ ]
}
```

i The HTTP status code on a replay mirrors the **original** response — a replayed rejection returns `422`, a replayed acceptance returns `200`. Always check the `replayed` field to distinguish a cached result from a fresh one.

✓ Use a new `request_id` (e.g. a UUID or a hash of your batch) for each distinct submission. Reuse the same `request_id` only when retrying after a network failure on a request you are unsure was processed.

7 Error Reference

Per-reading error codes

These codes appear in `results[].code` when a single reading is rejected:

CODE	MEANING	RESOLUTION
BARCODE_NOT_FOUND	The barcode value does not exist in the platform	Verify the barcode against the GET endpoint or your provisioned list
LOT_NOT_FOUND	The barcode exists but is not linked to any lot for your site and service type	Contact your Snap & Bill administrator to verify the meter assignment
READING_BELOW_PREVIOUS	The submitted <code>current_unit</code> is less than the previously recorded value	Use GET /readings to retrieve the current baseline and resubmit
DUPLICATE_BARCODE_IN_PAYLOAD	The same barcode appears more than once in the same request	Deduplicate your readings array before sending

Payload-level errors (HTTP 422)

These errors cause the entire request to be rejected before per-row processing:

```
{
  "message": "The readings field is required."
}
```

- Missing required fields (`request_id` , `service_type_id` , `readings`)
- Invalid `service_type_id` (does not exist in platform)
- `readings` is empty or not an array
- `reading_date` is in the future or invalid format (must be `YYYY-MM-DD`)
- `current_unit` is not numeric

8 HTTP Status Codes

CODE	DESCRIPTION
200 OK	Request processed. Check <code>status</code> field — may be <code>accepted</code> , <code>partial_success</code> , or <code>rejected</code> .
401 Unauthorized	Token is missing, invalid, revoked, or expired.
403 Forbidden	Token is valid but the source IP is not on the whitelist.
422 Unprocessable Entity	Payload failed schema validation (missing fields, invalid formats, etc.).
429 Too Many Requests	Rate limit exceeded. Back off and retry.
500 Internal Server Error	Unexpected server error. Contact support with the request time and your <code>request_id</code> .

9 Changelog & Contact

VERSION	DATE	NOTES
1.0	April 2026	Initial release — POST/GET readings, token management, idempotency, IP whitelisting

For integration support, provisioning of tokens, or to report an issue, contact support@mycloud.mu.