Listing Payments & Refunds via API

Audience: integrator engineers reconciling payments against an external system (POS, accounting, or back-office).

What Is This?

Pixalink records a payment whenever a customer pays for something — a reservation deposit, a paid-membership purchase, or a credit-wallet top-up. Refunds are stored in the same place, linked back to the payment they reverse.

The `/api/payments` collection is the single, read-only feed of **every** payment and refund in an organisation, regardless of which feature produced it. Use it for end-of-day batch reconciliation across all payment types, or to look up one payment by your own reference.

Operation	Method	Endpoint
List payments & refunds	`GET`	`/api/payments`
Show a single payment	`GET`	`/api/payments/{id}`

Both endpoints are authenticated with a Sanctum bearer token, scoped to a single organisation. You never send `organisation_id` — results are always limited to the token's organisation. The acting user must hold the **View Any Payment** permission for the list endpoint (and **View Payment** for the single-record endpoint), or the request returns `403`.

Base URL: `https://{pixalink-domain}`

Headers

```
Authorization: Bearer <token>
Accept: application/json
```

Reference

List payments & refunds

```
GET /api/payments
```

Returns a paginated collection, newest first by default. Narrow and shape the results with the query parameters below.

Filters

Filter	Notes
`filter[type]`	`payment` or `refund`.
`filter[status]`	Integer status code — e.g. `1` = success, `0` = pending, `2` = failed (full list below).
`filter[method]`	`commerce_pay`, `manual`, `credit`, `cash`, `card`, `e_wallet`, ...
`filter[reference_id]`	Exact match on the transaction reference you sent (returned as `reference_id`). The fastest way to find one specific payment.
`filter[customer_id]`	**New.** Returns only payments where the paying customer matches this id. Safely ignores non-customer payers, so an unrelated record sharing the same numeric id is never returned.
`filter[payable_type]`	Fully-qualified class of what was paid for. Examples: `App\Models\Reservation`, `App\Models\CustomerPaidMembership`.
`filter[created_at]`	Date with an operator, e.g. `>2026-01-01`. Supports `>`, `<`, `=`.
`filter[completed_at]`	Date with an operator, e.g. `>2026-01-01`.

Sorting

```
sort=-created_at
```

Allowed sort fields: `created_at`, `completed_at`, `amount`. Prefix with `-` for descending. The default is `-created_at` (newest first).

Includes

Include	Returns
`payer`	**New.** The paying customer, as the full customer object (id, name, contact details, points, credits, tier, and so on). `null` for the rare payment with no customer attached.
`refunds`	All refund records linked to this payment.
`refundingPayment`	The original payment that a refund record reverses.

Combine includes with a comma: `include=payer,refunds`.

Paging

`per_page` sets the page size (default `15`). The response carries standard `links` and `meta` paging blocks.

Example

```
GET /api/payments?filter[customer_id]=42&filter[type]=payment&include=payer&sort=-created_at
```

Response (200)


```
{
  "data": [
    {
      "id": 1,
      "type": "payment",
      "method": "commerce_pay",
      "reference_id": "TXN-1234567890",
      "amount": 150.0,
      "currency_code": "MYR",
      "status": 1,
      "is_manual": false,
      "payment_type": null,
      "source": null,
      "attempted_at": "2026-01-01T00:00:00+00:00",
      "completed_at": "2026-01-01T00:01:00+00:00",
      "failed_at": null,
      "failure_reason": null,
      "created_at": "2026-01-01T00:00:00+00:00",
      "payer": {
        "id": 42,
        "name": "John Doe",
        "email": "john@example.com",
        "phone_number": "+60188888888",
        "gender": null,
        "date_of_birth": null,
        "source": null,
        "status": null,
        "current_point": 2500,
        "created_at": "2025-12-01T00:00:00.000000Z",
        "updated_at": "2026-01-01T00:00:00.000000Z",
        "notes": null,
        "is_manually_assign_tier": false,
        "current_credits": 150,
        "birthday_month": null,
        "space_id": 1,
        "tier_id": 1,
        "organisation_id": 2
      }
    }
  ],
  "links": { "first": "…", "last": "…", "prev": null, "next": null },
  "meta": { "current_page": 1, "per_page": 15, "total": 1 }
}
```


Show a single payment

```
GET /api/payments/{id}?include=payer
```

Returns the same object shape as one element of the list, wrapped in `data`. Accepts the same `include` values (`payer`, `refunds`, `refundingPayment`).

A payment that belongs to another organisation returns **`404`** (not `403`), so ids cannot be probed across tenants.

Response fields

Field	Type	Meaning
`id`	integer	Unique identifier of the payment.
`type`	string	`payment` or `refund`.
`method`	string	How the payment was taken (`commerce_pay`, `manual`, `credit`, ...).
`reference_id`	string | null	The transaction reference you supplied, if any.
`amount`	number	Amount in major currency units (e.g. `150.0`, not cents).
`currency_code`	string	ISO currency code, e.g. `MYR`.
`status`	integer	Status code (see table below).
`is_manual`	boolean	`true` when the payment was recorded manually, with no gateway.
`payment_type`	string | null	Business payment type, when recorded.
`source`	string | null	Where the payment originated, when recorded (e.g. `api`).
`attempted_at`	datetime | null	When the payment was attempted.
`completed_at`	datetime | null	When the payment completed.
`failed_at`	datetime | null	When the payment failed.
`failure_reason`	string | null	Reason for failure, when failed.
`created_at`	datetime	When the record was created.
`payer`	object | null	The paying customer — only when `include=payer`.
`refunds`	array	Linked refund records — only when `include=refunds`.
`refunding_payment`	object | null	The original payment a refund reverses — only when `include=refundingPayment`.

Status codes

`status` is an **integer**, not a string. The common values:

Code	Meaning
`0`	Pending
`1`	Success
`2`	Failed
`3`	Cancelled
`9`	Authorized
`11`	Refunded

Good to Know

Find a payment fastest by your own reference. If you stored a transaction reference when the payment was taken, `GET /api/payments?filter[reference_id]=POS-12345` is an exact lookup and the quickest path to a single record.

Filtering by customer is payer-based. `filter[customer_id]` matches the customer recorded as the payer on the payment. Payments where the customer sits one level deeper (for example, attached only through the thing being paid for) are not matched by this filter — fetch those through that feature's own endpoint instead.

Gateway internals are never exposed. Fields such as the gateway provider name and the raw gateway request and response are deliberately left out of this feed, so third-party gateway artefacts never leak to integrators. You only ever see fields relevant to reconciliation.

Amounts are in major units. `amount` is already in ringgit (or the record's currency), not cents — `150.0` means RM 150.00.

Need Help?

If a payment you expect is missing or a filter returns nothing, check that your token belongs to the right organisation and holds the View Any Payment permission, then confirm the `reference_id` you are searching for matches exactly. For anything else, contact the Pixalink support team with the payment `id` or your transaction reference.