Claude Message Batches API: 50% Pricing, Limit (2026)

Edition of November 13, 2026 — facts verified November 13, 2026. Current edition always at https://tygartmedia.com/claude-batch-api-pricing-limits/.

Direct Answer (November 13, 2026): The Anthropic Message Batches API cuts input and output token prices by **50%** on asynchronous batches (up to 100,000 requests or 256 MB per batch, 24-hour completion window): **Fable 5.2 $4/$20**, **Opus 5.5 $2/$10**, **Sonnet 5.5 $1/$5**, and **Haiku 5.5 $0.05/$0.25** per MTok (≤100K prompts; $0.25/$1.25 above 100K).

Updated October 28, 2026: Made infographic diagrams clickable to open full-size images.

Updated November 2, 2026 — added direct answers on batch processing costs and turnaround conditions, sourced from Anthropic's published pricing.

Last verified: November 13, 2026 (Pacific) against Anthropic Message Batches docs and API pricing.

The Message Batches API lets you submit up to 100,000 Claude requests in a single call and receive results asynchronously — at exactly 50% of standard token prices. Most batches finish in under an hour. Results remain downloadable for 29 days. This page covers every verified limit, the per-tier rate limit tables, and how batch pricing stacks with prompt caching.

Is batch processing cheaper on the Claude API?

Yes. Batch processing on the Claude API is 50% cheaper than standard synchronous API calls, providing an automatic 50% discount on both input tokens and output tokens across all supported Claude models.

The deciding condition is your workload's **turnaround tolerance**: requests submitted to the Message Batches API are queued and processed asynchronously rather than returning an instant response stream. While most batches complete within 1 to a few hours, Anthropic's processing window allows up to 24 hours for batch completion. If your application can tolerate asynchronous turnaround—such as overnight evaluations, bulk data extraction, backfill content generation, or offline embeddings—batching cuts your Claude API bill directly in half. If your application powers a user-facing chatbot or requires low-latency real-time responses, you must use standard API calls.

For an operational breakdown of queue architecture and real-world batching workflows, see our companion piece on Anthropic Batch API discount strategies and workload patterns.

Pricing: 50% off standard rates

Batch pricing — half-rate framing without sticky dollars.

Every token processed through the Message Batches API is billed at half the standard input and output price. No quality difference from synchronous requests — only timing. The table below shows verified batch prices for active models.

Model	Batch input (per MTok)	Batch output (per MTok)	Standard input (per MTok)	Standard output (per MTok)
Claude Fable 5.2	$4.00	$20.00	$8.00	$40.00
Claude Opus 5.5	$2.00	$10.00	$4.00	$20.00
Claude Sonnet 5.5	$1.00	$5.00	$2.00	$10.00
Claude Haiku 5.5 (≤100K / >100K)	$0.05 / $0.25	$0.25 / $1.25	$0.10 / $0.50	$0.50 / $2.50

Legacy models (Fable 5.1 at $5/$25 batch, Haiku 4.5 at $0.50/$2.50 batch, Opus 4.5–5, and Sonnet 4.5–5) remain listed at their original rates; the 50% batch discount applies to every model.

Source: platform.claude.com/docs/en/build-with-claude/batch-processing

Key limits at a glance

Key limits at a glance — stale-proof.

Limit	Value
Maximum requests per batch	100,000
Maximum batch payload size	256 MB
Typical completion time	Under 1 hour
Hard expiration window	24 hours from creation
Result retention period	29 days after creation
Zero Data Retention eligible	No
Results format	JSONL, streamed via `results_url`
Supported models	All active Claude models

A batch expires if processing has not completed within 24 hours. Any individual request within that batch that did not finish is marked `expired` — you are not billed for expired or errored requests. Batch results (the JSONL file) are accessible for download for 29 days after the batch was created; after that the batch object itself is still visible but results can no longer be downloaded.

Message Batches API rate limits

The Message Batches API has its own rate-limit pool, shared across all models, separate from the standard Messages API limits. The "processing queue" count refers to individual batch requests (not batches) that have been submitted but not yet completed by the model. Anthropic publishes one standard set of batch limits, shown below; usage tiers (Start, Build, Scale, Custom) now govern monthly spend caps rather than batch throughput.

Maximum requests per minute (RPM)	Max batch requests in processing queue	Max batch requests per batch
1,000	200,000	100,000

Source: platform.claude.com/docs/en/api/rate-limits

RPM here limits how fast you can make HTTP requests to the Batches API endpoints (create, retrieve, list, cancel). It does not limit how many individual requests inside a batch are processed per minute — that is governed by the queue cap above. If high demand causes processing to slow, more individual requests within a batch may reach the 24-hour expiration limit.

Stacking batch pricing with prompt caching

The Batches API documentation explicitly states that the 50% batch discount and prompt caching discounts stack. Cache writes incur a one-time cost at 1.25x the base input rate (5-minute TTL) or 2x (1-hour TTL); subsequent cache reads cost $0.20/MTok on Fable 5.2 (0.025x) and Opus 5.5 (0.05x), $0.10/MTok on Sonnet 5.5 (0.05x), and $0.01/$0.05 per MTok on Haiku 5.5 (0.1x). Because batches process asynchronously and may take longer than 5 minutes, Anthropic recommends using the 1-hour cache duration for batch requests that share large context.

The following example uses Claude Opus 5.5 (standard input: $4.00/MTok) to show what each token type costs in a batch with a 1-hour cached system prompt.

Token type	Multiplier applied	Effective price per MTok	How calculated
Uncached input (standard)	1x	$4.00	Baseline
Uncached input (batch)	0.5x	$2.00	50% batch discount
Cache write — 1h TTL (batch)	2x × 0.5x = 1x	$4.00	2x write cost, then 50% batch
Cache read (batch)	0.05x × 0.5x = 0.025x	$0.10	5% read cost, then 50% batch

Token type	Multiplier applied	Effective price per MTok	How calculated
Output (batch)	0.5x of $20.00	$10.00	50% batch discount on output

In practice: if you cache a 50,000-token system prompt once and then read it across 1,000 batch requests, the cache write costs $0.20 (50K tokens at $4.00/MTok effective), while 1,000 cache reads cost $5.00 total (50M tokens at $0.10/MTok). The same 50 million tokens without caching would cost $100 in batch input (50 MTok at the $2.00/MTok batch rate). Cache hit rates on batches vary; Anthropic's documentation notes typical rates of 30% to 98% depending on traffic patterns, since batch requests are processed concurrently rather than sequentially.

How results come back

How results come back from Message Batches.

When the batch finishes (or the 24-hour limit is reached), a `results_url` property is set on the batch object. Results are in JSONL format — one JSON object per line, in any order (not necessarily matching submission order). Each result carries the `custom_id` you assigned, plus a `result` object of type `succeeded`, `errored`, `canceled`, or `expired`. Streaming the results file rather than downloading it all at once is recommended for large batches. You are not billed for `errored`, `canceled`, or `expired` requests.

Does the Batches API count against my standard Messages API rate limits?

No. The Message Batches API has its own rate-limit pool that is tracked separately from the standard Messages API RPM, ITPM, and OTPM limits. You can use both simultaneously up to their respective limits.

What happens if my batch does not finish within 24 hours?

Any individual requests within the batch that did not complete are marked `expired`. You are not billed for those requests. The batch itself moves to `ended` status and whatever results did complete are available at the `results_url`.

Can I use extended thinking, tool use, or vision in a batch?

Yes. The Batches API supports vision, tool use (including server tools such as web search and code execution), system messages, multi-turn conversations, and extended thinking. The parameters not supported are `stream: true`, fast mode (`speed`), Threads parameters, and `max_tokens: 0`.

How long are batch results available for download?

Results are available for 29 days after the batch was created. After that window, the batch object remains visible in the Console and via the API, but the results file can no longer be downloaded.

Is the Batches API eligible for Zero Data Retention?

No. The Message Batches API is explicitly excluded from Zero Data Retention (ZDR). Data is retained under the feature's standard retention policy regardless of your organization's ZDR settings.

Run your batches from the Anthropic Console — API keys, billing, and a token-to-dollar cost estimator that shows what 50%-off batch pricing does to your actual workload.

Related on Tygart Media: tokens to words · Claude Code billing · how much Claude costs.

Primary & Reference Links Cited on This Page

- Anthropic Message Batches docs — `https://platform.claude.com/docs/en/build-with-claude/batch-processing`
- API pricing — `https://platform.claude.com/docs/en/about-claude/pricing`
- platform.claude.com/docs/en/api/rate-limits — `https://platform.claude.com/docs/en/api/rate-limits`
- Tygart Media: Anthropic Batch API discount strategies and workload patterns
- Tygart Media: https://tygartmedia.com/wp-content/uploads/2026/08/claude-api-token-meter-visual.jpg
- Tygart Media: https://tygartmedia.com/wp-content/uploads/2026/08/claude-rate-limits-why-ceiling.jpg
- Tygart Media: https://tygartmedia.com/wp-content/uploads/2026/08/anthropic-api-key-before-you-start.jpg

- Tygart Media: Anthropic Console
- Tygart Media: tokens to words
- Tygart Media: Claude Code billing
- Tygart Media: how much Claude costs