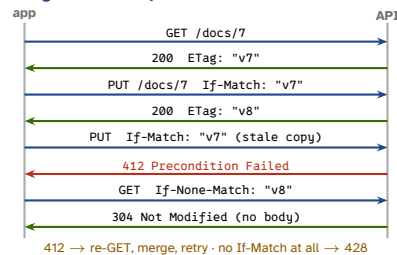
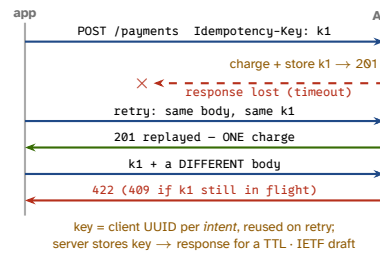


In one line: An API is a **contract you can never fully take back**: model **resources** (nouns) and let **methods** carry the verbs, make every write **safe to retry**, make every change **additive**, and judge a Swift API by **clarity at the point of use**. Old app versions live for years.

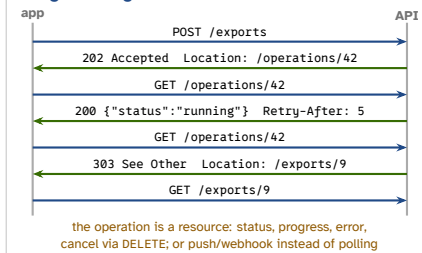
Etag: cache + optimistic lock



Idempotency-Key: retry a POST



Long-running: 202 + status



HTTP: resources, methods, status

Nouns, plural, shallow: /users/7/orders, not /getUserOrders?id=7 (RPC-in-URL, Richardson L0); past ~2 levels go flat + filter (/orders?user=7). Non-CRUD action: POST /orders/9/cancel (or a cancellation sub-resource). Root is always an **object**, never a bare array (no room for next/meta later).

Method	Safe	Idem.	Use
GET/HEAD	yes	yes	read; cacheable
PUT	no	yes	full replace at a known URI (client picks id)
DELETE	no	yes	2nd call may 404 — the state is the same
POST	no	no	create / action — needs an Idempotency-Key
PATCH	no	no	Merge Patch (RFC 7396, null = remove) / JSON Patch (6902)

2xx 200 body · 201 + Location · 202 accepted, not done · 204 no body
4xx 400 malformed · 401 who are you? · 403 known, not allowed · 404 (or hide existence) · 409 state conflict · 412 If-Match failed · 422 valid JSON, invalid values · 429 + Retry-After
5xx 500 bug · 502 bad upstream reply · 503 overloaded + Retry-After · 504 upstream timeout

Retry only idempotent requests on 5xx/429/timeout, exponential backoff + jitter.

Errors — RFC 9457 problem details (2023, obsoletes 7807):

```
422 Unprocessable Content Content-Type: application/problem+json
{"type":"https://api.example.com/probs/validation","status":422,
 "title":"Invalid request","detail":"email is not an address",
 "instance":"/signups/81","errors":[{"field":"email"}]}
```

Clients branch on type (a URI; default about:blank), never on detail text; extra members allowed; never leak stack traces.

Pagination. Offset (?offset=40&limit=20): page jumps, but **drifts** (insert → duplicate row, delete → skipped row) and costs O(offset). Cursor/keyset (?after=<opaque>): stable under writes, index-fast, no jumps; needs a **total order** — sort key + unique tie-breaker (created_at, id), the cursor encodes the last pair. Feeds → cursor. Filters/sort as query params (?status=open&sort=created), cap limit, next link in the body or Link: rel="next" (RFC 8288).

Mobile: latency, not bandwidth, dominates — **chunky beats chatty** (one call per screen, not 1 + N). **BFF** (Backends for Frontends, Sam Newman): a per-client backend that aggregates and shapes; plus gzip/br, field selection, delta sync (?since=), ETags.

	REST	GraphQL	gRPC
Wire	HTTP + JSON	one POST endpoint	HTTP/2 + protobuf
Contract	OpenAPI (opt.)	typed schema	.proto + codegen
Caching	HTTP/CDN, free	hard: persisted queries	none built in
Stream	SSE / WebSocket	subscriptions	4 kinds incl. bidi
Cost	chatty screens	N+1 resolvers, query cost	no browser (gRPC-Web)
Fits	public, cacheable	many shapes per client	service-to-service

Evolution — versions + compatibility

- URI** /v2/ (visible, routable) · **media type** (vnd.x.v2+json) · **date-pinned** header (Stripe) · **none**, additive only (GraphQL @deprecated, protobuf).
- Safe:** new endpoint, optional request field, response field. **Breaking:** remove/rename, change type/unit, newly required, newly nullable, a **new enum value**, a changed default.
- Never repurpose a field** (protobuf: never reuse a tag). **Tolerant reader** (Fowler): ignore unknown fields, unknown values → .unknown. Retire: Sunset (RFC 8594) + a min-app-version gate.

Swift library API

```
public enum Status: String, Decodable, Sendable {
    case active, suspended, unknown // tolerant reader
    public init(from d: Decoder) throws {
        let raw = try d.singleValueContainer().decode(String.self)
        self = Status(rawValue: raw) ?? .unknown
    }
    // items.remove(at: i) · a.sort() mutates, a.sorted() returns
```

- Swift API Design Guidelines:** “Clarity at the point of use is your most important goal”, over brevity; **omit needless words**; name by **role**; factories make...; Bools as assertions (isEmpty); conversions unlabeled (Int64(x)).
- Progressive disclosure:** common case in one line (URLSession.shared.data(from:)), power via default arguments.
- Return some P to keep the concrete type changeable; every public is forever — widen later, never narrow.
- SemVer MAJOR** for a protocol requirement without a default or a new public enum case. ABI stable Swift 5.0, module stability 5.1 (.swiftinterface, BUILD_LIBRARY_FOR_DISTRIBUTION: non-@frozen enums need @unknown default). Rename: @available(*, deprecated, renamed:).

Interview traps · questions

- ! If-Match mismatch is **412**, not 409. An ETag is opaque, not a hash.
- ! A new enum case breaks JSON decoders and Swift switches alike.
- ! no-cache = store but revalidate; no-store = never store.

- Safe “pay” on a flaky network? — Idempotency-Key; server replays.
- Two editors, one doc? — If-Match; 412 → re-fetch, merge.
- Slow export? — 202 + Location of an operation resource.

Remember: nouns + HTTP verbs · retry-safe writes · add, never change · cursor for feeds · clarity at the point of use.