

Backend Developer English Essentials

Must-know terms and phrases for backend developers — APIs, databases, system design, architecture patterns.

<https://coderslingo.com/resources/cheatsheets/backend-developer-essentials/>

API & web

endpoint — A single URL + method the server exposes, e.g. POST /users.

REST — A style for HTTP APIs built around resources and standard verbs (GET/POST/PUT/DELETE).

resource — A thing the API exposes (a user, an order), usually mapped to a URL path.

payload — The data sent in a request or response body, typically JSON.

status code — The 3-digit result of an HTTP request (200 OK, 404 Not Found, 500 error).

idempotent — An operation you can repeat with no extra effect — GET, PUT and DELETE should be idempotent.

rate limit — A cap on how many requests a client may send in a time window; over it returns 429.

pagination — Returning a large result set in pages, via offset/limit or a cursor.

webhook — A callback URL your server calls when an event happens, instead of the client polling.

polling — Repeatedly asking the server "is it ready yet?" instead of being notified.

authentication — Proving who you are (a token, an API key, credentials).

authorization — Deciding what an authenticated user is allowed to do.

CORS — Browser rules controlling which origins may call your API from JavaScript.

serialization — Turning an in-memory object into a transmittable format like JSON; the reverse is deserialization.

middleware — Code that runs on every request before the handler — auth, logging, parsing.

Databases

query — A request to read or change data, usually written in SQL.

index — A lookup structure that speeds up reads on a column but slows writes.

transaction — A group of operations that all succeed or all roll back together (ACID).

migration — A versioned, scripted change to the database schema.

N+1 query — A performance bug: one query for a list, then one extra query per item.

connection pool — A reusable set of open DB connections, so each request doesn't open its own.

foreign key — A column that references a row in another table, enforcing a relationship.

normalization — Structuring tables to remove duplicated data.

denormalization — Deliberately duplicating data to make reads faster.

ORM — A library that maps database rows to objects in your code.

full table scan — Reading every row because no index could be used — usually slow.

deadlock — Two transactions each waiting on a lock the other holds, so neither proceeds.

replication — Copying data to replica databases for read scaling or failover.

sharding — Splitting one dataset across several databases by some key.

cardinality — How many distinct values a column has — high cardinality indexes well.

Architecture & systems

monolith — A single deployable application containing all the logic.

microservice — A small, independently deployable service owning one capability.

queue — A buffer that holds messages so producers and consumers can work at different speeds.

message broker — Infrastructure (Kafka, RabbitMQ) that routes messages between services.

cache — Fast storage holding copies of expensive-to-compute or expensive-to-fetch data.

cache invalidation — Removing or refreshing stale cached data — famously hard to get right.

load balancer — A component that spreads incoming traffic across multiple server instances.

throughput — How much work the system handles per unit time (requests per second).

latency — How long a single request takes to respond.

bottleneck — The single slowest part that limits overall throughput.

stateless — A service that keeps no per-client data, so any instance can handle any request.

idempotency key — A client-supplied id that lets the server safely deduplicate retried requests.

eventual consistency — A model where replicas converge over time; reads may briefly be stale.

circuit breaker — A guard that stops calling a failing dependency to avoid cascading failures.

dead letter queue — Where messages go after repeatedly failing to be processed.

graceful degradation — Keeping core features working when a non-critical dependency is down.

backpressure — Signalling upstream to slow down when a consumer can't keep up.

race condition — A bug where the result depends on the unpredictable timing of concurrent operations.

tech debt — Shortcuts in the code that will cost time later if not paid back.

observability — Being able to understand system behaviour from its logs, metrics and traces.

Key phrases used at work

"This change is backward-compatible — no API consumers need to update."

"Heads up: removing this field is a breaking change, so we'll need a deprecation window."

"I found a few N+1 queries in the ORM — added eager loading to fix them."

"The endpoint needs to be idempotent so a retried request doesn't create duplicates."

"The trade-off here is between read latency and write consistency."

"The database is the bottleneck at our current traffic — let's add a read replica."

"I'd argue we should use optimistic locking here rather than pessimistic locking."

"There's a potential race condition if two requests hit this endpoint at the same time."

"The happy path is covered — I still need tests for the error cases."

"Let's add a guard clause for the empty-list case before we merge."

"Nit: I'd rename this to `userCount` for clarity, but up to you."

"LGTM — nice solution to the N+1 problem."

"That's a valid concern, but it's out of scope for this PR — I'll open a follow-up ticket."

"We're accepting eventual consistency here, so reads may return slightly stale data."

"Failed messages end up in the dead letter queue for investigation."

"Yesterday: merged the JWT refresh endpoint (PR #312). Today: rate-limiting middleware. No blockers."

"Blocked on PR #287 — waiting for review from the platform team. Flagging in case we need to escalate."

"Quick question before I proceed: should this return a 404 or a 403 when the user lacks access?"

"The profiler shows this query is doing a full table scan — we can add a composite index."

"This is pragmatic for now, but let's log it as tech debt and revisit next quarter."

