API Testing Checklist

A checklist for validating API contracts, status codes, schema behavior, authentication, authorization, data rules, and operational resilience.

Purpose

Use this checklist when testing new APIs, changed endpoints, integrations, service contracts, or backend behavior that supports user-facing workflows.

1. Contract and Schema

- Request methods, routes, required headers, query parameters, path parameters, and body fields match the documented contract.
- Required, optional, nullable, defaulted, deprecated, and unknown fields are handled consistently.
- Response schema includes expected types, field names, nesting, arrays, pagination metadata, and error structures.
- Backward compatibility is considered for existing consumers, versions, mobile apps, and partner integrations.

2. Status Codes and Errors

- Success, validation failure, authentication failure, authorization failure, not found, conflict, rate limit, and server error cases are tested.
- Error responses include useful messages, stable error codes, correlation identifiers, and safe detail levels.
- Invalid JSON, missing body, malformed parameters, unsupported media types, and oversized payloads are handled safely.
- Error handling avoids leaking secrets, stack traces, internal hostnames, or implementation details.

3. Data and Business Rules

- Boundary values, empty values, duplicate values, special characters, localization, time zones, and date ranges are covered.
- Create, read, update, delete, partial update, idempotency, retry, and conflict behavior are validated where relevant.
- Database state, cache behavior, eventual consistency, and downstream synchronization are considered.
- Test data setup and cleanup are repeatable, isolated, and safe for shared environments.

4. Security and Permissions

- Authentication is tested for valid, missing, expired, malformed, revoked, and wrong-audience tokens.
- Authorization checks cover roles, ownership, tenant boundaries, object-level access, and privilege escalation attempts.
- Sensitive fields are masked, omitted, encrypted, or protected according to product and compliance requirements.
- Rate limits, abuse controls, CORS behavior, and replay-sensitive operations are reviewed.

5. Reliability and Observability

- Timeouts, retries, circuit-breaking behavior, slow dependencies, partial outages, and degraded responses are considered.
- Logs, metrics, traces, and correlation identifiers make failures diagnosable without exposing sensitive data.
- Performance expectations are tested for common payload sizes, peak scenarios, and critical workflows.
- API tests are categorized into smoke, contract, integration, regression, negative, and monitoring candidates.