# API & Web Services Glossary

## API (Application Programming Interface)

A set of rules that lets software applications talk to each other.

## Endpoint

The unique URL where an API resource is found.

## Request

The message sent to an API asking for data or action.

## Response

The reply from the API with data or status.

## Query Parameter

Extra data in a URL to filter results (?status=active).

## Path Parameter

Variable in the URL path (/users/{id}).

## Payload

The request body data sent with POST or PUT.

## Headers

Extra information in requests or responses (auth, content-type).

## API Key

A unique code that authenticates API requests.

## SDK

Software kit that helps developers work with APIs.

## API Call

One request made to an API.

## API Integration

Connecting systems using APIs.

## REST API

API style that uses HTTP methods to access resources.

## RPC API

Remote Procedure Call — runs functions remotely.

## SOAP API

Protocol using XML for strict and secure communication.

## Synchronous vs Asynchronous

Sync waits for reply. Async continues without waiting.

## Latency

Time it takes for API to reply.

## Throughput

How many requests the API can handle per second.

## Uptime

Percentage of time API is available.

## Caching

Storing responses to speed up calls.

## Pagination

Breaking data into pages (?page=1&limit=50).

## Filtering

Restricting results (?status=active).

## Sorting

Ordering results (?sort=created\_at).

## Rate Limiting

Restricting number of requests in time frame.

## Throttling

Slowing down requests to prevent overload.

## API Versioning

Running multiple API versions (/v1, /v2).

## OAuth 2.0

Secure delegated access without sharing passwords.

## JWT (JSON Web Token)

Compact signed token for authentication.

## Mutual TLS

Two-way certificate verification for secure access.

## API Monitoring

Tracking performance and errors.

## API Logging

Recording request and response details.

## API Gateway

System that manages API traffic, routing, and security.

## Service Discovery

Finding APIs dynamically in a network.

## API Mocking

Simulating API responses for testing.

## API Deprecation

Phasing out old API versions.

## API Sandbox

Safe environment for testing APIs.

## API Orchestration

Combining multiple calls into one process.

## REST Methods

GET = read, POST = create, PUT = replace, PATCH = update, DELETE = remove.

## HTTP Status Codes

200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 404 Not Found, 500 Server Error.

## WSDL

Web Services Description Language — blueprint of SOAP service.

## SOAP Envelope

Root element of a SOAP message.

## SOAP Header

Optional section with metadata (auth, routing).

## SOAP Body

Main request or response data.

## SOAP Fault

Error in a SOAP response.

## UDDI

Registry for discovering web services.

## WS-Security

Standard for securing SOAP with encryption and signatures.

## API Security

Protecting APIs from unauthorized use and attacks.

## Authentication

Verifying who is making the request.

## Authorization

Deciding what they can access.

## HTTPS

Secure HTTP with encryption.

## IP Whitelisting

Allowing access only from approved IPs.

## CORS

Rules about which domains can call your API.

## Token Expiration

Setting tokens to expire after time.

## Token Refresh

Getting a new token without logging in again.

## HMAC Authentication

Verifying requests with hashed codes.

## Encryption at Rest

Storing data in encrypted form.

## Encryption in Transit

Encrypting data while moving between systems.

## Replay Protection

Blocking repeated captured requests.

## RBAC

Role-Based Access Control — restrict by role.

## ABAC

Attribute-Based Access Control — restrict by attributes.

## API Firewall

Layer that blocks suspicious requests.

## DDoS Protection

Defending against flooding attacks.

## Zero Trust

Never trust, always verify every request.

## Penetration Testing

Simulating attacks to find weaknesses.

## Fuzz Testing

Sending random/invalid inputs to test API.

## Compliance

Meeting rules like GDPR, HIPAA.

## API Governance

Rules for API security, usage, quality.

## API SLA

Service Level Agreement — defines guarantees.

## GraphQL

Query language — client asks for exactly what it needs.

## GraphQL Mutation

Changing data in GraphQL.

## GraphQL Subscription

Real-time updates in GraphQL.

## gRPC

Fast framework using Protocol Buffers.

## Protobuf

Binary data format used by gRPC.

## API Facade

Simplifies backend complexity for client.

## API Adapter

Translates requests between systems.

## API Proxy

Forwards requests and responses.

## Composite API

Combines multiple service responses.

## API Chaining

Running multiple calls in sequence.

## API Virtualization

Simulated APIs when real ones are offline.

## API Load Balancing

Distributes requests across servers.

## API Failover

Switching to backup servers on failure.

## Blue-Green Deployment

Switching between two identical environments for updates.

## Canary Release

Rollout to small group before full release.

## Backward Compatibility

New APIs still support old clients.

## Forward Compatibility

Old clients can work with future changes.

## API Lifecycle

Design → Develop → Test → Deploy → Monitor → Retire.

## Webhook

Server pushes updates to client.

## Polling

Client checks repeatedly for updates.

## API Mesh

Network of APIs managed as one system.

## API Federation

Combining multiple APIs into one.

## Chaos Testing

Intentionally breaking things to test resilience.

## Self-Healing API

API that recovers from failures automatically.

## API Analytics

Tracking API usage trends.

## API Sustainability

Designing APIs to use less energy and cost.