

Terminology

REST (Representational State Transfer) has become the de facto standard for building web [APIs](#). Its stateless nature, resource-based approach, and standard HTTP methods (GET, POST, PUT, DELETE) provide a clean, predictable interface for client-server communication.

□ Authentication & Security

API security starts with robust authentication mechanisms. From API keys and JWT tokens to OAuth 2.0 flows, choosing the correct authentication method is critical for protecting your endpoints while maintaining usability.

□ API Design Principles

Good API design follows predictable patterns:

Use nouns for resource endpoints (/users instead of /getUsers)

Implement consistent error handling and status codes

Version your APIs to maintain backward compatibility

Design with scalability in mind, considering rate limiting and caching strategies

□ Documentation Standards

Clear, comprehensive documentation is the cornerstone of API adoption. OpenAPI (formerly Swagger) has emerged as the industry standard for describing REST APIs, enabling both human-readable documentation and automated code generation.

□ Testing & Monitoring

A robust API testing strategy encompasses:

Unit tests for individual endpoints

Integration tests for API workflows

Performance testing under load

Continuous monitoring for availability and response times

These concepts form the foundation of modern API development.

Revision #4

Created 8 February 2025 21:55:44 by sedawk

Updated 10 February 2025 17:19:53 by sedawk