



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

API Design: REST, GraphQL, and gRPC

Class Duration

14 hours of live training delivered over 2 days.

Student Prerequisites

- Software development experience (any language)
- Experience consuming HTTP APIs from code
- Basic familiarity with JSON
- Some experience building a web service endpoint is helpful
- No prior OpenAPI, GraphQL, or gRPC experience required

Target Audience

This course is designed for backend and full-stack developers who build APIs that other people — and increasingly other machines and AI agents — depend on. It suits developers who have shipped endpoints but never been taught API design as a discipline, tech leads establishing API standards across teams, and architects choosing between REST, GraphQL, and gRPC for new systems. The course is language-agnostic: examples use HTTP, JSON, and schema languages rather than any single framework.

Description

An API is a promise, and most APIs are promises made carelessly. This course teaches API design as a deliberate discipline, starting where good HTTP APIs start: resource modeling, the Richardson maturity model, and the actual semantics of HTTP methods, status codes, caching, and content negotiation. From there it works through the patterns every production API needs — pagination, filtering, and sorting; versioning strategies and their costs; and error design done properly with RFC 9457 Problem Details, the standard that obsoleted RFC 7807. Participants adopt a contract-first workflow, authoring OpenAPI 3.1 documents (with a look at what OpenAPI 3.2 adds) that drive mock servers, validation, and generated clients.

Day two broadens the picture. Participants design authentication and authorization with OAuth 2.0, OpenID Connect, and API keys; apply rate limiting; and design webhooks and asynchronous APIs documented with AsyncAPI. The course then gives GraphQL and gRPC honest, non-evangelical



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

treatment: schema design and the situations where GraphQL genuinely wins, and protobuf-based gRPC for service-to-service communication. It closes with the organizational layer — governance, style guides, and breaking-change management — and a forward-looking module on designing APIs for AI consumption, including the Model Context Protocol (MCP) and what makes an API legible to an LLM-driven agent.

Learning Outcomes

- Model resources and relationships for an HTTP API and evaluate designs against the REST maturity model
- Apply HTTP semantics correctly: methods, status codes, idempotency, caching headers, and content negotiation
- Design pagination (offset and cursor), filtering, and sorting that hold up under real data volumes
- Compare versioning strategies — URI, header, and evolution without versioning — and choose deliberately
- Design consistent, machine-readable errors using RFC 9457 Problem Details
- Author OpenAPI 3.1 contracts and run a contract-first workflow with mocking, linting, and code generation
- Select and implement auth patterns: OAuth 2.0 flows, OpenID Connect, and API keys, each in its proper place
- Apply rate limiting and quota patterns and communicate limits to clients
- Design reliable webhooks and asynchronous APIs, documented with AsyncAPI
- Design a GraphQL schema and articulate when GraphQL is the right choice — and when it is not
- Define gRPC services with protobuf and manage schema evolution for service-to-service APIs
- Establish API governance: style guides, review processes, deprecation policy, and breaking-change management
- Design APIs that AI agents can consume effectively, including exposing capabilities via MCP

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Software Requirements

- Text editor or IDE with OpenAPI support (VS Code recommended)
- Node.js LTS or Python 3.12+ for tooling
- An HTTP client: curl, HTTPie, Bruno, or Postman
- Spectral or similar OpenAPI linter
- Docker or Podman for running mock servers (optional)
- Modern web browser

Training Topics

Foundations of API Design

- APIs as contracts and products
- Resource modeling: nouns, relationships, granularity
- The Richardson maturity model in practice
- Consistency, predictability, and the principle of least surprise

HTTP Semantics

- Methods, safety, and idempotency
- Status codes that mean what they say
- Headers, caching, ETags, and conditional requests
- Content negotiation and media types

Collection Patterns

- Offset versus cursor pagination and their tradeoffs
- Filtering and query parameter design
- Sorting, sparse fieldsets, and partial responses
- Bulk and batch operation patterns

Versioning Strategies

- URI versioning, header versioning, and media-type versioning
- Designing for evolution: additive change and tolerant readers
- Deprecation signaling and sunset timelines
- What actually counts as a breaking change

Error Design

- RFC 9457 Problem Details (successor to RFC 7807)
- Problem types, extension members, and error catalogs
- Validation errors and reporting multiple problems



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Errors that help client developers instead of frustrating them

Contracts and Contract-First Workflows

- OpenAPI 3.1: structure, JSON Schema alignment, components
- What OpenAPI 3.2 adds and migration considerations
- Design-first versus code-first, honestly compared
- Linting with style rules, mock servers, generated clients
- Contract testing and keeping the spec true

Authentication and Authorization

- OAuth 2.0 flows: which one for which client
- OpenID Connect for identity
- API keys: where they are appropriate and how to handle them
- Scopes, audiences, and token validation
- Common auth design mistakes

Rate Limiting and Operational Concerns

- Rate limiting algorithms and quota design
- Communicating limits: headers and retry guidance
- Timeouts, retries, and idempotency keys

Webhooks and Async APIs

- Webhook design: delivery, signing, retries, ordering
- Polling, long-polling, SSE, and WebSockets compared
- Event-driven API documentation with AsyncAPI
- Designing events: payloads, envelopes, and versioning

GraphQL

- Schema design: types, queries, mutations, connections
- Resolvers, the N+1 problem, and dataloaders
- Pagination with the connection pattern
- Security: depth limiting and cost analysis
- When GraphQL is the right choice — and when it is not

gRPC and Protobuf

- Protobuf messages, services, and field numbering
- Unary and streaming RPCs
- Schema evolution and compatibility rules
- gRPC for service-to-service; gRPC-Web and transcoding at the edge



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Governance and APIs for AI

- API style guides and design review processes
- Breaking-change detection in CI
- Designing APIs for AI agent consumption: clear naming, rich descriptions, predictable errors
- The Model Context Protocol (MCP): exposing APIs as tools
- Documentation that both humans and LLMs can use