JUPITER PRIME

Consulting • Training • Advisory

جوبيتر برايم للاستشارات والتدريب

COURSE BROCHURE

LLM API Integration and Function Calling Training Course

jupiterprime.uk

info@jupiterprime.uk

Registered in the United Kingdom and Saudi Arabia

LLM API Integration and Function Calling Training Course

LLM API Integration and Function Calling is a 1-week training course for developers and architects who connect large language model APIs to business systems through REST services, structured outputs and tool calls. Each participant leaves with a tested LLM integration service blueprint, covering schemas, retrieval, error handling and an evaluation set. Jupiter Prime Training built the course around hands-on coding labs.

Why LLM API Integration and Function Calling Matters Now

Language model features are moving from chat windows into order systems, service desks and internal portals. That shift demands reliable JSON schema output, safe function calling, controlled token cost and protected API keys. Fragile prompt-only integrations break under load or return malformed data. The course aligns design choices with ISO/IEC 42001 for AI management and the NIST AI Risk Management Framework AI RMF for measuring and treating model risk.

Who Gains Most from LLM API Integration and Function Calling

- Backend developers adding model calls to existing REST services
- Solution architects designing integration patterns for language model features
- Technical product owners who set accuracy, latency and cost targets
- Platform engineers responsible for API gateways, secrets and monitoring
- QA engineers building evaluation sets for model-driven features

Teams gain most when a developer and an architect attend together and agree one reference design.

LLM API Integration and Function Calling Capabilities

Participants move from experimental prompts to production-grade integration code.

- Design REST endpoints that wrap model calls with validation and timeouts — service consumers
- Enforce structured outputs with JSON schema and repair malformed responses — downstream systems
- Define function calling tools with clear parameters and permission limits — security teams
- Build embeddings retrieval that grounds answers in approved documents — content owners
- Apply rate limits, backoff and token budgets to control cost — finance and operations
- Measure answer quality with an eval set before each release — product owners

Key LLM API Integration and Function Calling Terms

- Structured outputs — model responses constrained to a declared format so code can parse them without guesswork.
- JSON schema — a machine-readable definition of fields, types and required values used to validate model output.
- Function calling — a pattern where the model proposes a named tool and arguments, and application code decides whether to run it.
- Embeddings retrieval — finding relevant passages by comparing numeric vectors, then passing them to the model as context.
- Exponential backoff — retrying a refused or failed request after growing delays to respect provider rate limits.
- Eval set — a fixed collection of inputs and expected results used to score model behaviour after every change.

LLM API Integration and Function Calling Day by Day

Day 1: Model APIs and Request Design

- Request and Response Anatomy — Choosing Parameters for Each Task
- System Instructions and Message Roles — Controlling Behaviour Predictably
- Token Counting and Context Limits — Sizing Inputs Safely
- Streaming Versus Single Responses — Matching User Experience Needs
- ISO/IEC 42001 and NIST AI RMF — Integration Risk Responsibilities

Day 2: Structured Outputs and Validation

- JSON Schema Design — Fields, Types and Required Values
- Structured Output Modes — Constraining Model Responses
- Validation and Repair Loops — Handling Malformed Output
- Extraction Patterns — Turning Free Text into Records
- Schema Versioning — Keeping Consumers Stable Through Change

Day 3: Function Calling and Tool Orchestration

- Tool Definition Writing — Names, Descriptions and Parameters
- Argument Checking — Rejecting Unsafe or Incomplete Calls
- Multi-Step Tool Loops — Chaining Calls with Clear Exit Rules
- Permission Boundaries — Read-Only Versus Write Actions
- Human Approval Points — Confirming High-Impact Operations

Day 4: Retrieval, Reliability and Cost

- Embeddings Retrieval Pipelines — Chunking, Indexing and Ranking
- Rate Limits and Exponential Backoff — Surviving Provider Throttling
- Token Cost Budgets — Caching and Model Tier Selection
- API Key Vault Practice — Rotation, Scoping and Leak Prevention
- Logging and Tracing — Diagnosing Slow or Wrong Answers

Day 5: Practising the LLM Integration Service Blueprint

- Logistics Case Lab — Shipment Queries Through Function Calling
- Banking Case Lab — Structured Extraction from Customer Letters
- Eval Set Construction — Scoring Accuracy Before Release
- Load and Failure Drill — Testing Backoff and Fallback Paths
- LLM Integration Service Blueprint — Final Design for Review

Applying LLM API Integration and Function Calling After the Course

- Wrap one existing model call in your codebase with schema validation, backoff and an eval set within the next sprint.
- Present the LLM integration service blueprint to your architecture board as the reference pattern for new model features.

LLM API Integration and Function Calling Questions FAQ

What is LLM API integration and function calling, and why does it matter now?

It is the engineering of reliable connections between language model APIs and business software, using schemas, tools and retrieval. It matters now because model features are entering core systems, where malformed output, runaway cost or leaked keys cause real damage.

Which roles gain most from LLM API integration and function calling training?

Backend developers, solution architects, technical product owners, platform engineers and QA engineers gain most. Working knowledge of one programming language and REST APIs is expected.

How does LLM API integration and function calling differ from a prompt writing course?

Prompt writing courses focus on wording requests for people using chat tools. This course focuses on code: schemas, tool definitions, retries, retrieval pipelines and testing inside production services.

What do you bring back from LLM API integration and function calling training?

You bring back an LLM integration service blueprint with schemas, tool definitions, retry rules and an eval set. You also return with lab code you can adapt immediately.

LLM API Integration and Function Calling: The Way Forward

The LLM integration service blueprint first guides the next model feature your team ships. Architects use it to review designs, developers reuse its schemas and retry rules, and product owners track its eval scores at each release. After the first production quarter, revisit token budgets and tool permissions, since real usage often differs from lab assumptions.