+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 05/19/2026

Sign up

SGLang Training

3 days (21 hours)

Overview

SGLang is a serving- and agent-oriented framework for orchestrating LLMs with high performance. It enables the construction of reliable inference pipelines (chat, RAG, tools) while optimizing latency and throughput.

This training aims to empower participants to design LLM applications in SGLang: structuring prompts, managing context, calling tools, and setting up multi-step workflows. The approach emphasizes reproducibility, observability, and best deployment practices.

The course is centered around workshops and demos: creating an inference service, implementing a mini-RAG, adding safeguards, and then conducting load testing. Deliverables: SGLang scripts, serving configuration, test suites, and an industrialization checklist.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Install and configure a working SGLang environment.
- Write SGLang programs for chat, RAG, and multi-step workflows.
- Integrate tools (functions) and handle errors/structured responses.
- Optimize latency and throughput using serving parameters and best practices.
- Test, trace, and package a production-ready API.

Target Audience

- Python developers

- ML/LLM engineers
- Data engineers
- Architects/Tech leads

Prerequisites

- Proficiency in Python (functions, modules, environments)
- Basic understanding of HTTP APIs and JSON formats
- Basics of LLMs (prompting, tokens, context)
- Knowledge of Git and the command line

Technical prerequisites

- Computer with at least 16 GB of RAM (32 GB recommended)
- Linux or macOS; Windows is possible via WSL2
- Python 3.10+, pip/venv, code editor (VS Code or equivalent)
- Access to an NVIDIA GPU (CUDA) recommended for local inference; otherwise, CPU/remote server execution

SGLang Training Program

[Day 1 - Morning]

Getting started with SGLang and the runtime environment

- Positioning SGLang: objectives (LLM serving, agents, workflows) and production use cases
- Installing and validating the environment: Python, dependencies, GPU/CPU, variables, and basic configuration
- Understanding the execution model: SGLang scripts, runtime, session and context management
- Writing your first structured prompts: roles, templates, parameters, and typed outputs
- Hands-on workshop: Run your first SGLang script and obtain usable JSON output.

[Day 1 - Afternoon]

Output validation and prompt composition

- Structuring reliable responses: format constraints, required fields, application-side validation
- Composing steps: chaining subtasks, reusing variables and utility functions
  utility functions
- Reducing hallucinations: guidelines, safeguards, rephrasing strategies, and checks
- Context management: segmentation, summaries, short-term vs. long-term memory, and token limits
- Hands-on workshop: Building an “analysis > extraction > reporting” pipeline with strictly structured output.

[Day 2 - Morning]

Tool-based calls and application integration

- Defining tools: input/output schemas, serialization, and error handling
- Orchestrating external calls: REST APIs, databases, business functions, and internal services
- Tool selection strategies: routing, rules, priorities, and fallback
- Observability: logs, traces, metrics, and request/response correlation
- Hands-on workshop: Add a “product search” tool and generate a final response that is justified and traceable.

[Day 2 - Afternoon]

Performance and serving: latency, throughput, and costs

- Understanding performance levers: batching, parallelism, caching, streaming, and concurrency management
- Configuring inference: temperature, top-p, max tokens, stop sequences, and impact on quality/latency
- Optimizing prompts: token reduction, reusable templates, and “diff” prompts
- Resource management: GPU memory, limits, timeouts, and protection against costly requests
- Hands-on workshop: Measure latency and throughput, then apply 3 optimizations and compare the results.

[Day 3 - Morning]

Quality, testing, and security of LLM applications

- Implementing a testing strategy: test cases, assertions, non-regression tests, and golden outputs
- Assessing quality: criteria, scoring, format validation, and semantic checks
- Securing inputs: prompt injection, sensitive data, filtering, and content policies
- Handling failures: retries, backoff, timeouts, controlled degradation, and user messages
- Hands-on workshop: Create an automated test suite and harden a workflow against injection.

[Day 3 - Afternoon]

Deployment: packaging, deployment, and operations

- Scaling a project: structure, environment-specific configuration, and secret management
- Exposing a service: endpoints, API contracts, quotas, authentication, and logging
- Production monitoring: dashboards, alerts, cost tracking, and error analysis
- Maintenance plan: prompt versioning, migrations, A/B testing, and rollbacks
- Hands-on workshop: Deliver a production-ready SGLang mini-service (API + monitoring + runbook).

Target Audience

This training is intended for both individuals and companies, large or small, wishing to train their teams in a new advanced IT technology or to acquire specific business knowledge or modern methods.

Assessment upon enrollment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Practical Course: 60% Practical, 40% Theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and reflection sessions, and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been properly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training program.