

Databricks: Agent Bricks – Agentic AI, tools and governance

Databricks: Agent Bricks – Agentic AI is a hands-on course on designing and building AI agents in Databricks. The program focuses on an agent that selects and calls tools on its own: retrieval with AI Search, Unity Catalog functions or controlled SQL queries, while keeping control over which data and actions it can access.

Participants go through the whole process: from deciding whether the problem needs an agent or whether RAG is enough, through defining tools and building the agent, to analyzing its behavior in MLflow Tracing, testing for unwanted calls and designing permissions. The course ends with an agent prototype with a limited set of tools and a plan for deploying it safely. The exercises use RetailOps data and documents (a fictional retail company whose data we use for all exercises).



Training recipients

The training is intended for technical staff who want to build AI agents in Databricks that work on company data and tools.

The course is particularly suitable for:

- AI and ML engineers who have experience with RAG and want to move on to agents,
- developers building assistants that perform actions in company systems,
- AI solution architects responsible for security and permissions,
- teams planning a production deployment of agents who need a way to test them.

The training will be especially valuable for organizations that want to use agents but need clear rules on what an agent can access and how to check its behavior.



Benefits

- RAG or an agent – you will learn when an agent is justified and how to limit its autonomy to what the use case needs.
- Agent tools – you will learn to give an agent access to AI Search, Unity Catalog functions and controlled SQL queries, each with a clear contract.
- Building an agent – you will build your own agent with tool calling in Agent Bricks, the Databricks platform for building agents in code.
- Permissions and security – you will learn the principle of least privilege for an agent and how to limit data access through Unity Catalog.
- Testing and evaluation – you will learn to analyze agent behavior in MLflow Tracing, evaluate its answers and test for unwanted behavior.
- The path to production – you will learn the options for deploying an agent and prepare a plan for the next steps.



Training program

Module – RAG or an agent: choosing the architecture

Module goal: the participant knows how an agent differs from RAG and when it is worth building one.

- Agent – a language model that decides on its own which tool to call (a search engine, a database query, a business function) to complete a task
- Levels of solutions – a simple chain, RAG, an agent with tools, several cooperating agents; more capability, but also more cost and risk
- Limits of autonomy – what the agent can do on its own and what needs human approval
- An agent with no programming (Genie Agent) vs. a custom agent written in code (Agent Bricks); the earlier Knowledge Assistant and Supervisor Agent have been legacy since October 2026
- Choosing a model – which models can call tools
- Lab: the participant chooses an architecture for three RetailOps use cases

2. Module – Agent tools

Module goal: the participant prepares tools that the agent can use safely.

- Unity Catalog functions as tools – for example, “check order status”, written in SQL or Python, with permission control
- Document search (AI Search) as an agent tool
- Genie Agent (formerly Genie space) as a tool for questions about data in tables (demo)
- Controlled SQL queries – the agent supplies only the parameters for a query prepared in advance (demo)
- Tool description – the name, description and parameters decide whether the model picks the right tool

- Lab: the participant creates two functions (order status, return policy) and checks how the agent uses them

3. Module - Building an agent

Module goal: the participant builds a working agent and understands how it makes decisions.

- The agent loop - the model picks a tool, gets the result and decides on the next step or the answer
- An agent project in Agent Bricks - a template created by the Agent Bricks CLI (Beta), the list of tools and their permissions in the agent.toml file, the DurableAgentServer agent server
- Agent-building libraries: LangGraph, OpenAI Agents SDK (short comparison)
- Connecting Unity Catalog functions as agent tools
- Testing the agent in AI Playground and in a notebook
- Lab: the participant builds an agent with one tool and analyzes its decisions

4. Module - Workshop 1: an agent with search and a business function

Module goal: the participant combines several tools in one agent.

- Workshop: a RetailOps agent that searches the documentation and checks order data through a Unity Catalog function. The outcome is an agent that performs several steps and backs its answer with sources.
- Analyzing failed calls and improving tool descriptions

5. Module - Identity, permissions and data boundaries

Module goal: the participant designs the agent's permissions so it has access to no more than it needs.

- On whose behalf the agent acts - a technical account (service principal) or the permissions of the user asking the question (user authorization, formerly on-behalf-of-user - Public Preview)
- Least privilege - access only to the functions, tables and indexes it needs
- Row filters and column masks in Unity Catalog - how they affect the agent's answers
- Unity Gateway - access control for models and MCP servers, limits, budgets, usage and cost tracking (GA); protective rules (service policies), such as content filters and sensitive data detection (Beta) - overview
- Prompt injection - protection against instructions hidden in documents and data
- Lab: the participant designs the agent's permissions and checks its answers from a second account, if the environment provides a second test account; otherwise a trainer demo

6. Module - Quality evaluation and security testing

Module goal: the participant can check whether the agent works correctly and safely before it reaches users.

- Tracing (MLflow Tracing) - which tools were called, with which parameters, how long it took and what it cost (we assume familiarity from DBX-RAG and focus on tool calls)
- Automatic evaluation - answer correctness, choice of the right tool, compliance with company rules
- Testing for unwanted behavior (red teaming) - attempts to get around instructions, questions about out-of-scope data
- Fallback behavior - refusing, handing the case over to a person, a limit on the number of steps
- Workshop 2: the participant completes a prepared set of 10 test cases, runs the evaluation and analyzes how the agent behaved. The outcome is a quality report and a list of fixes.

7. Module - Deployment and next steps

Module goal: the participant knows how to make the agent available to users and monitor it after deployment.

- Deploying the agent with the agentbricks deploy command to Agent Runtime on Databricks Apps (recommended for new agents); deployment as a Model Serving endpoint is legacy – when you will still see it and how to migrate
- Monitoring after deployment – logging requests and responses, evaluating quality on real traffic
- MCP – a standard for connecting tools to agents; MCP servers provided by Databricks and external ones, governed in Unity Gateway (the Genie One MCP server – GA, the others – Beta) – demo
- Agent memory – conversation history and facts remembered across conversations (agent memory and sessions in Lakebase, Beta) – overview; plan for the next version of the solution



Expected preparation of the participant

- Completion of DBX-RAG (GenAI: RAG with AI Search) or equivalent experience with AI Search, language models and MLflow
- Knowledge of Python at the level of functions and scripts
- Basic knowledge of Unity Catalog (permissions; row filters and masking are briefly recapped)

To increase the comfort of work and training's effectiveness we suggest using an additional monitor. The lack of additional monitor does not exclude participation in the training, however, it significantly influences the comfort of work during classes.

You can find information and requirements of participation in Distance Learning trainings at:

<https://www.altkomakademia.pl/en/how-it-works/about-courses/>



Training Includes

Training includes

- access to the Altkom Akademia student portal
- a prepared Databricks environment with an AI Search index, a model endpoint and RetailOps scenario data for the duration of the training
- notebooks and exercise materials in electronic form

Note: During selected training sessions, the Academy provides Participants with access to paid tools and services necessary to complete the training program (in particular Azure, Fabric, and other cloud resources). The Participant is required to use the provided resources solely within the scope of the training agenda and in accordance with the trainer's instructions. Use of these resources beyond the training scope—particularly for private projects, testing additional services, modifying the environment configuration, or undertaking activities that generate costs unrelated to the exercises—will result in the Participant being charged for costs exceeding the allocated limit, based on expenses incurred by the

Academy.



Language

- Training: English
- Materials: English

Duration

1 days / 7 hours

Examination description

Szkolenie nie zawiera formalnego egzaminu. Zaleca się jednak podejście do egzaminu certyfikacyjnego Databricks Certified Data Engineer Associate jako kolejny krok rozwojowy.