Observability and Logging in MCP

- **academy.kspl.tech | Koenig AI Academy**

Prerequisites check

- You should have a working local MCP server with at least one tool from Chapter 3 and one resour
- academy.kspl.tech | Koenig AI Academy

Operational logs vs audit events

- Use two categories:
- Operational logs help engineers debug reliability: latency, exception class, retry count, depen
- Audit events help reviewers reconstruct sensitive actions: actor, tool name, target object, aut
- They can go to the same logging pipeline, but they should be distinguishable. A timeout reading
- academy.kspl.tech | Koenig AI Academy

The minimum useful log event

- **Every tool call should emit:**
- **event: stable event name, such as mcp.tool.completed.**
- **tool_name: the MCP tool called.**
- **request_id: correlation ID from the host or generated server-side.**
- **academy.kspl.tech | Koenig AI Academy**

Add a wrapper

- Wrap tool handlers instead of copying logging code into every tool.
- async function withToolLogging(
- toolName: string,
- handler: () => Promise
- academy.kspl.tech | Koenig AI Academy

Audit examples

- **Read-only file listing may not need a durable audit record in a toy project. A legal document r**
- **Audit records should focus on business meaning:**
- **"event": "audit.connector.action_requested",**
- **"actor": "user_123",**
- **academy.kspl.tech | Koenig AI Academy**

Try it next

- **Deploy a gateway with RBAC, rate limits, and JSONL auditing**
- **academy.kspl.tech | Koenig AI Academy**