Building Your First MCP Server

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

Prerequisites check

- **You should have completed the MCP classification exercise from Chapter 2. You should also be co**
- **academy.kspl.tech | Koenig AI Academy**

What the server will do

- The server exposes one tool:
- list_project_files(root_label, relative_path)
- It lists files under a pre-approved project root. The user can ask Claude to inspect project st
- This is the key production lesson: the model should not decide the security boundary. The serve
- academy.kspl.tech | Koenig AI Academy

Minimal server shape

- An MCP server has identity, registered capabilities, and a transport. The SDK README shows a mi
- import { McpServer } from "@modelcontextprotocol/server";
- import { StdioServerTransport } from "@modelcontextprotocol/server/stdio";
- import * as z from "zod/v4";
- academy.kspl.tech | Koenig AI Academy

Why this is safer than `read_file(path)`

- A raw read_file(path) tool looks convenient. It is also an invitation to leak secrets. The mode
- Never rely on "Claude will know not to ask for secrets." A connector must enforce policy in cod
- academy.kspl.tech | Koenig AI Academy

Controlled errors

- Errors are part of the user experience. A stack trace is useful to an attacker and confusing to
- For this chapter, use three predictable errors:
- Unknown root label.
- Path escapes approved root.
- 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**