

AI for Software Dev

Video #2 – AI Agents

Jim Fawcett & Mike Corley

AI Agents

- Agents are software applications that:
 - Focus on local projects
 - Define commands for frequent operations
 - Sequences of actions that support project development
 - Defined by the project domain
 - Provide helper functions to read from and write to project files
 - Send prompts and receive responses from a remote LLM
 - Defined by a platform key
 - Uses platform web messaging API

Advantages

- Unlike chat bots, agents only send prompts for AI processing
 - No need to paste code text or zip files at end of prompt
 - That means fewer tokens transferred, faster processing, and more reliable actions since those are defined in user application code
- Agents can be defined using any language for which there are messaging APIs
 - Python, Rust, C++, C#, and JavaScript
 - Existing libraries can be imported by the agent source to make development more productive and have more domain knowledge

Example – Basic Support for Development

- The example agent, **dev_agent.py**, provides support for:
 - File analysis
 - Project improvement suggestions
 - Show project directory tree
 - List project files
 - Generate README markdown file
 - Custom prompts
- It was generated using a claude.ai prompt and sent to local download
- It executes in a Windows 11 PowerShell terminal

Example – Prompt & Partial Response

You: `/analyze ./src/lib.rs`

 Analyzing...

Agent: ## Analysis of `./src/lib.rs`

1. Purpose and Functionality

This is a generic directory navigation library (`rust_dir_nav`) that provides a configurable way to traverse directory trees and process files/directories. Key features:

- **Generic Design**: Uses a trait-based approach (`DirEvent`) allowing users to define custom handlers for files and directories
- **Pattern Matching**: Filters files by extensions (patterns)
- **Configurable Traversal**: Options for recursive traversal and hiding empty directories
- **Cross-platform**: Handles Windows/Linux path separator differences

The library follows a depth-first search algorithm to traverse directories and calls user-defined handlers for matching files.

Conclusion

- This third presentation focuses on AI Command Line Interface (CLI)
 - You can find all the details here:
https://jimfawcett.github.io/NewSite/Code/CodeBites_AgentAI.html
- The first presentation explores:
 - AI Chat Bots
 - Use web application defined by platform to communicate with LLM
 - Cannot read or write local files
- The second presentation explores:
 - AI Agents
 - Use local application to communicate with remote LLM
 - Uses HTTPS-based API
 - Can read from and write to local file system