AI for Software Dev

Video #4 – AI Skills

Jim Fawcett & Mike Corley

AI Skills

- AI Skills are specialized prompts and resources that are available for chats :
  - Are defined for specific domain areas, e.g., quality analysis, documentation, ...
  - These files get mounted into the LLM container for the current session from your account storage.
  - Are loaded automatically when a prompt is related to the skill domain.
- Skills are directly supported only by the Anthropic platform
  - You can simulate skills for ChatGPT and Gemini by creating, for each skill, an agent that reads from a local skills directory and injects the skill files as a prompt, using the platform messaging API

Advantages

- AI skills allow you to provide training like information that is used across all chats.
- This specialization is a powerful way to channel the LLM operations to your specific needs.
  - For example, a code-smells skill can define specific things to look for and analyze, e.g., long-methods, lacking-documentation, inefficient-code, …
- You are essentially making the LLM pay attention to specific things in your domain.

Example – Code Smells

- The implementation-smells skill provides a Skill.md file
  - Contains information about code organization, naming and documentation, control flow, error handling, and language specific topics.
- The skill also provides a smell-catalog.md file
  - Contains code organization smells, God class/object, feature envy, …
    - These terms are standard labels for specific code smells
- Running on RustDirNav project directory resulted in:
  - 1 high severity smell (code bug)
  - 5 medium severity smells (code maintainability)
  - 4 low severity smells (non-idiomatic code)

Example – Prompt and Partial Response

```
 2      ?
 3      |
 4     ?▉?      Claude Code v2.1.45
 5   ▌?▉▉?▌     Sonnet 4.6 · Claude Pro
 6 ??▉▉▉▉??   C:\github\Jimfawcett\NewSite\code\AI
 7   ?? ??
 8
 9 ? analyze code smells for /DemoAgent-Claude2/Test/RustDirNav
10
11 ? Skill(implementation-smells)
12   ?  Successfully loaded skill
13
14 ? Searched for 1 pattern, read 3 files (ctrl+o to expand)
15
16 ? Now let me run the automated detector on both source files in parallel.
17
18 ? 2 Bash agents finished (ctrl+o to expand)
19     ├─ Run smell detector on lib.rs · 1 tool use · 5.6k tokens
20     │   ?  Done
21     └─ Run smell detector on test1.rs · 1 tool use · 5.6k tokens
22         ?  Done
23
24 ? Now I have everything needed for a thorough manual + automated analysis.
25
26   ---
27   Implementation Smells Analysis - RustDirNav
28
29   Files analyzed: src/lib.rs (259 lines), examples/test1.rs (77 lines)
30
```

Conclusion

- This fourth presentation focuses on Anthropic AI Skills
  - You can find all the details here: https://jimfawcett.github.io/NewSite/Code/CodeBites_SkillsAI.html
- The first presentation explores:
  - AI Chat Bots
  - Use web application defined by platform to communicate with remote 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
- The third presentation explores:
  - AI Consoles
  - Can read from and write to local files
  - Injects prompts and receives replies using build in messaging API