

Prompting for Expertise with the Persona Pattern

When building agents that can leverage specialized expertise, we have a powerful tool at our disposal: the ability to have the LLM adopt different expert personas to provide specialized insights using the **Persona Pattern**. The `prompt_expert` tool serves as our foundation, allowing us to create interactions with experts who have specific backgrounds, expertise, and perspectives. Let's explore how to build specialized expert tools for a software development workflow.

First, let's look at our base expert consultation tool:

```
@register_tool()
def prompt_expert(action_context: ActionContext,
description_of_expert: str, prompt: str) -> str:
    """
    Generate a response from an expert persona.

    The expert's background and specialization should be thoroughly
    described to ensure
    responses align with their expertise. The prompt should be
    focused on topics within
    their domain of knowledge.

    Args:
        description_of_expert: Detailed description of the expert's
background and expertise
        prompt: The specific question or task for the expert

    Returns:
        The expert's response
    """
    generate_response = action_context.get("llm")
    response = generate_response(Prompt(messages=[
        {"role": "system",
         "content": f"Act as the following expert and respond
accordingly: {description_of_expert}"},
        {"role": "user", "content": prompt}]
```

