

Strands API Reference

Version 0.1.x

Generated: 2025-06-18

Agent Class

The Agent class is the core component of the Strands framework.

Constructor Parameters:

- model: The language model to use (BedrockModel, OpenAIModel, etc.)
- system_prompt: Instructions defining agent behavior
- tools: List of callable tools the agent can use
- memory: Memory system for context retention

Methods:

- `__call__(message: str) -> AgentResult`: Send a message to the agent
- `reset()`: Clear conversation history
- `add_tool(tool)`: Add a new tool to the agent

Example:

```
from strands import Agent  
agent = Agent(system_prompt='You are helpful')  
response = agent('Hello!')
```

Model Classes

BedrockModel:

- AWS Bedrock integration for Claude, Llama, and other models
- Requires AWS credentials and Bedrock access
- model_id: 'claude-3.7-sonnet', 'llama-3.1-8b', etc.

OpenAIModel:

- OpenAI API integration for GPT models
- Requires OPENAI_API_KEY environment variable
- model_id: 'gpt-4', 'gpt-3.5-turbo', etc.

OllamaModel:

- Local model execution with Ollama
- No API keys required, runs on your machine
- model_id: 'llama3.2', 'mistral', 'phi3', etc.

All models support:

- Streaming responses
- Token counting
- Custom parameters

Creating Tools

Tools extend agent capabilities with custom functions.

Creating a Tool:

```
from strands import tool
```

```
@tool
```

```
def calculate_sum(a: int, b: int) -> int:
```

```
    """Add two numbers together."""
```

```
    return a + b
```

Tool Requirements:

- Must have type hints for all parameters
- Must have a docstring describing the function
- Should handle errors gracefully
- Return JSON-serializable data

Advanced Tool Features:

- Async tool support
- Tool validation
- Custom error handling
- Tool composition