

Agents

Detailed guide on creating and managing agents within the CrewAI framework.

Overview of an Agent

In the CrewAI framework, an Agent is an autonomous unit that can:

- Perform specific tasks
- Make decisions based on its role and goal
- Use tools to accomplish objectives
- Communicate and collaborate with other agents
- Maintain memory of interactions
- Delegate tasks when allowed

Agent Attributes

Attribute	Parameter	Type	Description
Role	role	str	Defines the agent's function and expertise within the crew.
Goal	goal	str	The individual objective that guides the agent's decision-making.
Backstory	backstory	str	Provides context and personality to the agent, enriching interactions.
LLM (optional)	llm	Union[str, LLM, Any]	Language model that powers the agent. Defaults to the model specified in OPENAI_MODEL_NAME or "gpt-4".

Attribute	Parameter	Type	Description
Tools (optional)	tools	List[BaseTool]	Capabilities or functions available to the agent. Defaults to an empty list.
Function Calling LLM (optional)	function_calling_llm	Optional[Any]	Language model for tool calling, overrides crew's LLM if specified.
Max Iterations (optional)	max_iter	int	Maximum iterations before the agent must provide its best answer. Default is 20.
Max RPM (optional)	max_rpm	Optional[int]	Maximum requests per minute to avoid rate limits.
Max Execution Time (optional)	max_execution_time	Optional[int]	Maximum time (in seconds) for task execution.
Verbose (optional)	verbose	bool	Enable detailed execution logs for debugging. Default is False.
Allow Delegation (optional)	allow_delegation	bool	Allow the agent to delegate tasks to other agents. Default is False.
Step Callback (optional)	step_callback	Optional[Any]	Function called after each agent step, overrides crew callback.
Cache (optional)	cache	bool	Enable caching for tool usage. Default is True.
System Template (optional)	system_template	Optional[str]	Custom system prompt template for the agent.
Prompt Template (optional)	prompt_template	Optional[str]	Custom prompt template for the agent.
Response Template (optional)	response_template	Optional[str]	Custom response template for the agent.
Allow Code Execution (optional)	allow_code_execution	Optional[bool]	Enable code execution for the agent. Default is False.
Max Retry Limit (optional)	max_retry_limit	int	Maximum number of retries when an error occurs.

Attribute	Parameter	Type	Description
			Default is 2.
Respect Context Window (optional)	respect_context_window	bool	Keep messages under context window size by summarizing. Default is True.
Code Execution Mode (optional)	code_execution_mode	Literal["safe", "unsafe"]	Mode for code execution: 'safe' (using Docker) or 'unsafe' (direct). Default is 'safe'.
Multimodal (optional)	multimodal	bool	Whether the agent supports multimodal capabilities. Default is False.
Inject Date (optional)	inject_date	bool	Whether to automatically inject the current date into tasks. Default is False.
Date Format (optional)	date_format	str	Format string for date when inject_date is enabled. Default is "%Y-%m-%d" (ISO format).
Reasoning (optional)	reasoning	bool	Whether the agent should reflect and create a plan before executing a task. Default is False.
Max Reasoning Attempts (optional)	max_reasoning_attempts	Optional[int]	Maximum number of reasoning attempts before executing the task. If None, will try until ready.
Embedder (optional)	embedder	Optional[Dict[str, Any]]	Configuration for the embedder used by the agent.
Knowledge Sources (optional)	knowledge_sources	Optional[List[BaseKnowledgeSource]]	Knowledge sources available to the agent.
Use System Prompt (optional)	use_system_prompt	Optional[bool]	Whether to use system prompt (for o1 model support).

```
from crewai import Agent
```

```
from crewai_tools import SerperDevTool
```

```
# Create an agent with all available parameters
agent = Agent(
    role="Senior Data Scientist",
    goal="Analyze and interpret complex datasets to provide actionable
    insights",
    backstory="With over 10 years of experience in data science and machine
    learning,"
        "you excel at finding patterns in complex datasets.",
    llm="gpt-4", # Default: OPENAI_MODEL_NAME or "gpt-4"
    function_calling_llm=None, # Optional: Separate LLM for tool calling
    verbose=False, # Default: False
    allow_delegation=False, # Default: False
    max_iter=20, # Default: 20 iterations
    max_rpm=None, # Optional: Rate limit for API calls
    max_execution_time=None, # Optional: Maximum execution time in
    seconds
    max_retry_limit=2, # Default: 2 retries on error
    allow_code_execution=False, # Default: False
    code_execution_mode="safe", # Default: "safe" (options: "safe", "unsafe")
    respect_context_window=True, # Default: True
    use_system_prompt=True, # Default: True
    multimodal=False, # Default: False
    inject_date=False, # Default: False
    date_format="%Y-%m-%d", # Default: ISO format
    reasoning=False, # Default: False
```

```
max_reasoning_attempts=None, # Default: None  
tools=[SerperDevTool()], # Optional: List of tools  
knowledge_sources=None, # Optional: List of knowledge sources  
embedder=None, # Optional: Custom embedder configuration  
system_template=None, # Optional: Custom system prompt template  
prompt_template=None, # Optional: Custom prompt template  
response_template=None, # Optional: Custom response template  
step_callback=None, # Optional: Callback function for monitoring  
)
```