



☰ Core Concepts > CLI

Core Concepts

CLI

Learn how to use the CrewAI CLI to interact with CrewAI.

CrewAI CLI Documentation

The CrewAI CLI provides a set of commands to interact with CrewAI, allowing you to create, train, run, and manage crews & flows.

Installation

To use the CrewAI CLI, make sure you have CrewAI installed:

Terminal

```
pip install crewai
```



Core Concepts > CLI

Terminal

```
crewai [COMMAND] [OPTIONS] [ARGUMENTS]
```

Available Commands

1. Create

Create a new crew or flow.

Terminal

```
crewai create [OPTIONS] TYPE NAME
```

TYPE : Choose between "crew" or "flow"

NAME : Name of the crew or flow

Example:

Terminal



Core Concepts > CLI

2. Version

Show the installed version of CrewAI.

Terminal

```
crewai version [OPTIONS]
```

`--tools` : (Optional) Show the installed version of CrewAI tools

Example:

Terminal

```
crewai version  
crewai version --tools
```

3. Train

Train the crew for a specified number of iterations.



Core Concepts > CLI

`-n, --n_iterations INTEGER` : Number of iterations to train the crew (default: 5)

`-f, --filename TEXT` : Path to a custom file for training (default: "trained_agents_data.pkl")

Example:

Terminal

```
crewai train -n 10 -f my_training_data.pkl
```

4. Replay

Replay the crew execution from a specific task.

Terminal

```
crewai replay [OPTIONS]
```

`-t, --task_id TEXT` : Replay the crew from this task ID, including all subsequent tasks

Example:



Core Concepts > CLI

5. Log-tasks-outputs

Retrieve your latest crew.kickoff() task outputs.

Terminal

```
crewai log-tasks-outputs
```

6. Reset-memories

Reset the crew memories (long, short, entity, latest_crew_kickoff_outputs).

Terminal

```
crewai reset-memories [OPTIONS]
```

`-l, --long` : Reset LONG TERM memory

`-s, --short` : Reset SHORT TERM memory

`-e, --entities` : Reset ENTITIES memory



Core Concepts > CLI

Terminal

```
crewai reset-memories --long --short  
crewai reset-memories --all
```

7. Test

Test the crew and evaluate the results.

Terminal

```
crewai test [OPTIONS]
```

`-n, --n_iterations INTEGER` : Number of iterations to test the crew (default: 3)

`-m, --model TEXT` : LLM Model to run the tests on the Crew (default: "gpt-4o-mini")

Example:

Terminal



Core Concepts > CLI

8. Run

Run the crew or flow.

Terminal

```
crewai run
```

❗ Starting from version 0.103.0, the `crewai run` command can be used to run both standard crews and flows. For flows, it automatically detects the type from `pyproject.toml` and runs the appropriate command. This is now the recommended way to run both crews and flows.

❗ Make sure to run these commands from the directory where your CrewAI project is set up. Some commands may require additional configuration or setup within your project structure.

9. Chat

Starting in version 0.98.0, when you run the `crewai chat` command, you start an interactive session with your crew. The AI assistant will guide you by asking for necessary inputs to execute the crew. Once all inputs are provided, the crew will execute its tasks.



Core Concepts > CLI

```
crewai chat
```

ⓘ Ensure you execute these commands from your CrewAI project's root directory.

ⓘ IMPORTANT: Set the `chat_llm` property in your `crew.py` file to enable this command.

```
@crew
def crew(self) -> Crew:
    return Crew(
        agents=self.agents,
        tasks=self.tasks,
        process=Process.sequential,
        verbose=True,
        chat_llm="gpt-4o", # LLM for chat orchestration
    )
```

10. API Keys



Core Concepts > CLI

Initial API key providers

The CLI will initially prompt for API keys for the following services:

OpenAI

Groq

Anthropic

Google Gemini

SambaNova

When you select a provider, the CLI will prompt you to enter your API key.

Other Options

If you select option 6, you will be able to select from a list of LiteLLM supported providers.

When you select a provider, the CLI will prompt you to enter the Key name and the API key.

See the following link for each provider's key name:

[LiteLLM Providers](#)



Core Concepts > CLI

Powered by Mintlify