



☰ Get Started > Quickstart

Get Started

Quickstart

Build your first AI agent with CrewAI in under 5 minutes.

Build your first CrewAI Agent

Let's create a simple crew that will help us `research` and `report` on the `latest AI developments` for a given topic or subject.

Before we proceed, make sure you have `crewai` and `crewai-tools` installed. If you haven't installed them yet, you can do so by following the [installation guide](#).

Follow the steps below to get crewing! 🚀

1 Create your crew

Create a new crew project by running the following command in your terminal. This will create a new directory called `latest-ai-development` with the basic structure for your crew.

Terminal

```
crewai create crew latest-ai-development
```

2 Modify your `agents.yaml` file

[Get Started](#) > [Quickstart](#)

agents.yaml

```
# src/latest_ai_development/config/agents.yaml
researcher:
  role: >
    {topic} Senior Data Researcher
  goal: >
    Uncover cutting-edge developments in {topic}
  backstory: >
    You're a seasoned researcher with a knack for uncovering the latest
    developments in {topic}. Known for your ability to find the most relevant
    information and present it in a clear and concise manner.

reporting_analyst:
  role: >
    {topic} Reporting Analyst
  goal: >
    Create detailed reports based on {topic} data analysis and research findings.
  backstory: >
    You're a meticulous analyst with a keen eye for detail. You're known for
    your ability to turn complex data into clear and concise reports, making
    it easy for others to understand and act on the information you provide.
```

3 Modify your `tasks.yaml` file

tasks.yaml

```
# src/latest_ai_development/config/tasks.yaml
research_task:
  description: >
    Conduct a thorough research about {topic}
    Make sure you find any interesting and relevant information given
```



Get Started > Quickstart

```
reporting_task:
  description: >
    Review the context you got and expand each topic into a full section f
    Make sure the report is detailed and contains any and all relevant inf
  expected_output: >
    A fully fledge reports with the mains topics, each with a full section
    Formatted as markdown without ````
  agent: reporting_analyst
  output_file: report.md
```

4 Modify your `crew.py` file

crew.py

```
# src/latest_ai_development/crew.py
from crewai import Agent, Crew, Process, Task
from crewai.project import CrewBase, agent, crew, task
from crewai_tools import SerperDevTool

@CrewBase
class LatestAiDevelopmentCrew():
    """LatestAiDevelopment crew"""

    @agent
    def researcher(self) -> Agent:
        return Agent(
            config=self.agents_config['researcher'],
            verbose=True,
            tools=[SerperDevTool()]
        )

    @agent
    def reporting_analyst(self) -> Agent:
        return Agent(
```



Get Started > Quickstart

```
def research_task(self) -> Task:
    return Task(
        config=self.tasks_config['research_task'],
    )

@task
def reporting_task(self) -> Task:
    return Task(
        config=self.tasks_config['reporting_task'],
        output_file='output/report.md' # This is the file that will be conta
    )

@crew
def crew(self) -> Crew:
    """Creates the LatestAiDevelopment crew"""
    return Crew(
        agents=self.agents, # Automatically created by the @agent decorator
        tasks=self.tasks, # Automatically created by the @task decorator
        process=Process.sequential,
        verbose=True,
    )
```

5 Feel free to pass custom inputs to your crew

For example, you can pass the `topic` input to your crew to customize the research and reporting.

main.py

```
#!/usr/bin/env python
# src/latest_ai_development/main.py
import sys
from latest_ai_development.crew import LatestAiDevelopmentCrew
```



Get Started > Quickstart

```
'topic': 'AI Agents'  
}  
LatestAiDevelopmentCrew().crew().kickoff(inputs=inputs)
```

6 Set your environment variables

Before running your crew, make sure you have the following keys set as environment variables in your `.env` file:

An OpenAI API key (or other LLM API key): `OPENAI_API_KEY=sk-...`

A Serper.dev API key: `SERPER_API_KEY=YOUR_KEY_HERE`

7 Lock and install the dependencies

Lock the dependencies and install them by using the CLI command but first, navigate to your project directory:

Terminal

```
cd latest-ai-development  
crewai install
```

8 Run your crew

To run your crew, execute the following command in the root of your project:

Terminal

```
crewai run
```

9 View your final report



Get Started > Quickstart

output/report.md

Comprehensive Report on the Rise and Impact of AI Agents in 2024

1. Introduction to AI Agents

In 2024, Artificial Intelligence (AI) agents are at the forefront of innov

2. Benefits of AI Agents

AI agents bring numerous advantages that are transforming traditional work

- **Task Automation**: AI agents can carry out repetitive tasks such as da
- **Improved Efficiency**: By quickly processing large datasets and perfor
- **Enhanced Decision-Making**: AI agents can analyze trends and patterns

3. Popular AI Agent Frameworks

Several frameworks have emerged to facilitate the development of AI agents

- **Autogen**: A framework designed to streamline the development of AI ag
- **Semantic Kernel**: Focuses on natural language processing and understa
- **Promptflow**: Provides tools for developers to create conversational a
- **Langchain**: Specializes in leveraging various APIs to ensure agents c
- **CrewAI**: Aimed at collaborative environments, CrewAI strengthens team
- **MemGPT**: Combines memory-optimized architectures with generative capa

These frameworks empower developers to build versatile and intelligent age

4. AI Agents in Human Resources

AI agents are revolutionizing HR practices by automating and optimizing ke

- **Recruiting**: AI agents can screen resumes, schedule interviews, and e
- **Succession Planning**: AI systems analyze employee performance data an
- **Employee Engagement**: Chatbots powered by AI can facilitate feedback

As AI continues to evolve, HR departments leveraging these agents can real



Get Started > Quickstart

- **Investment Decisions**: AI agents provide stock predictions and analysis.

The incorporation of AI agents into finance is fostering a more responsive

6. Market Trends and Investments

The growth of AI agents has attracted significant investment, especially a

Conversely, corporations like Microsoft are taking strides to integrate AI

7. Future Predictions and Implications

Experts predict that AI agents will transform essential aspects of work li

- Enhanced integration of AI agents across all business functions, creating
- Continued advancement of AI technologies, resulting in smarter, more ada
- Increased regulatory scrutiny to ensure ethical use, especially concerni

To stay competitive and harness the full potential of AI agents, organizat

8. Conclusion

The emergence of AI agents is undeniably reshaping the workplace landscape

Note on Consistency in Naming

The names you use in your YAML files (`agents.yaml` and `tasks.yaml`) should match the method names in your Python code. For example, you can reference the agent for specific tasks from `tasks.yaml` file. This naming consistency allows CrewAI to automatically link your configurations with your code; otherwise, your task won't recognize the reference properly.

Example References



Get Started > Quickstart

agents.yaml

```
email_summarizer:  
  role: >  
    Email Summarizer  
  goal: >  
    Summarize emails into a concise and clear summary  
  backstory: >  
    You will create a 5 bullet point summary of the report  
  llm: mixtal_llm
```



Note how we use the same name for the agent in the `tasks.yaml` (`email_summarizer_task`) file as the method name in the `crew.py` (`email_summarizer_task`) file.

tasks.yaml

```
email_summarizer_task:  
  description: >  
    Summarize the email into a 5 bullet point summary  
  expected_output: >  
    A 5 bullet point summary of the email  
  agent: email_summarizer  
  context:  
    - reporting_task  
    - research_task
```

Use the annotations to properly reference the agent and task in the `crew.py` file.

Annotations include:

`@agent`



Get Started > Quickstart

`@callback`

`@output_json`

`@output_pydantic`

`@cache_handler`

crew.py

```
# ...
@agent
def email_summarizer(self) -> Agent:
    return Agent(
        config=self.agents_config["email_summarizer"],
    )

@task
def email_summarizer_task(self) -> Task:
    return Task(
        config=self.tasks_config["email_summarizer_task"],
    )
# ...
```



In addition to the [sequential process](#), you can use the [hierarchical process](#), which automatically assigns a manager to the defined crew to properly coordinate the planning and execution of tasks through delegation and validation of results. You can learn more about the core concepts [here](#).

Replay Tasks from Latest Crew Kickoff

CrewAI now includes a replay feature that allows you to list the tasks from the last run and replay from a specific one. To use this feature, run.



Get Started > Quickstart

Reset Crew Memory

If you need to reset the memory of your crew before running it again, you can do so by calling the reset memory feature:

```
crewai reset-memory
```

This will clear the crew's memory, allowing for a fresh start.

Deploying Your Project

The easiest way to deploy your crew is through [CrewAI Enterprise](#), where you can deploy your crew in a few clicks.

Was this page helpful?

< Installation

Agents >

Powered by Mintlify