

[Modules](#)[Agents](#)[How-to](#)[Handle parsing errors](#)

Handle parsing errors

Occasionally the LLM cannot determine what step to take because its outputs are not correctly formatted to be handled by the output parser. In this case, by default the agent errors. But you can easily control this functionality with `handle_parsing_errors`! Let's explore how.

Setup

We will be using a wikipedia tool, so need to install that

```
%pip install --upgrade --quiet wikipedia
```

```
from langchain import hub
from langchain.agents import AgentExecutor,
create_react_agent
from langchain_community.tools import
WikipediaQueryRun
from langchain_community.utilities import
WikipediaAPIWrapper
```

```
from langchain_openai import OpenAI

api_wrapper =
WikipediaAPIWrapper(top_k_results=1,
doc_content_chars_max=100)
tool =
WikipediaQueryRun(api_wrapper=api_wrapper)
tools = [tool]

# Get the prompt to use - you can modify this!
# You can see the full prompt used at:
https://smith.langchain.com/hub/hwchase17/react
prompt = hub.pull("hwchase17/react")

llm = OpenAI(temperature=0)

agent = create_react_agent(llm, tools, prompt)
```

Error

In this scenario, the agent will error because it fails to output an Action string (which we've tricked it into doing with a malicious input

```
agent_executor = AgentExecutor(agent=agent,
tools=tools, verbose=True)
```

```
agent_executor.invoke(  
    {"input": "What is Leo DiCaprio's middle  
name?\n\nAction: Wikipedia"}  
)
```

> Entering new AgentExecutor chain...

ValueError: An output parsing error occurred.
In order to pass this error back to the agent
and have it try again, pass
`handle_parsing_errors=True` to the
AgentExecutor. This is the error: Could not
parse LLM output: ` I should search for "Leo
DiCaprio" on Wikipedia
Action Input: Leo DiCaprio`

Default error handling

Handle errors with `Invalid or incomplete response:`

```
agent_executor = AgentExecutor(  
    agent=agent, tools=tools, verbose=True,
```

```
handle_parsing_errors=True
)
```

```
agent_executor.invoke(
    {"input": "What is Leo DiCaprio's middle
name?\n\nAction: Wikipedia"}
)
```

```
> Entering new AgentExecutor chain...
  I should search for "Leo DiCaprio" on
Wikipedia
Action Input: Leo DiCaprioInvalid Format:
Missing 'Action:' after 'Thought:I should
search for "Leonardo DiCaprio" on Wikipedia
Action: Wikipedia
Action Input: Leonardo DiCaprioPage: Leonardo
DiCaprio
Summary: Leonardo Wilhelm DiCaprio (;
Italian: [di'ka:prjo]; born November 1I now
know the final answer
Final Answer: Leonardo Wilhelm

> Finished chain.
```

```
{'input': "What is Leo DiCaprio's middle  
name?\n\nAction: Wikipedia",  
 'output': 'Leonardo Wilhelm'}
```

Custom error message

You can easily customize the message to use when there are parsing errors.

```
agent_executor = AgentExecutor(  
    agent=agent,  
    tools=tools,  
    verbose=True,  
    handle_parsing_errors="Check your output  
and make sure it conforms, use the  
Action/Action Input syntax",  
)
```

```
agent_executor.invoke(  
    {"input": "What is Leo DiCaprio's middle  
name?\n\nAction: Wikipedia"}  
)
```

```
> Entering new AgentExecutor chain...
Could not parse LLM output: ` I should search
for "Leo DiCaprio" on Wikipedia
Action Input: Leo DiCaprio`Check your output
and make sure it conforms, use the
Action/Action Input syntaxI should look for a
section on Leo DiCaprio's personal life
Action: Wikipedia
Action Input: Leo DiCaprioPage: Leonardo
DiCaprio
Summary: Leonardo Wilhelm DiCaprio (;
Italian: [di'ka:prjo]; born November 1I
should look for a section on Leo DiCaprio's
personal life
Action: Wikipedia
Action Input: Leonardo DiCaprioPage: Leonardo
DiCaprio
Summary: Leonardo Wilhelm DiCaprio (;
Italian: [di'ka:prjo]; born November 1I
should look for a section on Leo DiCaprio's
personal life
Action: Wikipedia
Action Input: Leonardo Wilhelm DiCaprioPage:
Leonardo DiCaprio
Summary: Leonardo Wilhelm DiCaprio (;
Italian: [di'ka:prjo]; born November 1I
should look for a section on Leo DiCaprio's
personal life
Action: Wikipedia
Action Input: Leonardo Wilhelm DiCaprioPage:
Leonardo DiCaprio
```

```
Summary: Leonardo Wilhelm DiCaprio (;  
Italian: [di'ka:prjo]; born November 11I now  
know the final answer  
Final Answer: Leonardo Wilhelm DiCaprio  
  
> Finished chain.
```

```
{'input': "What is Leo DiCaprio's middle  
name?\n\nAction: Wikipedia",  
 'output': 'Leonardo Wilhelm DiCaprio'}
```

Custom Error Function

You can also customize the error to be a function that takes the error in and outputs a string.

```
def _handle_error(error) -> str:  
    return str(error)[:50]  
  
agent_executor = AgentExecutor(  
    agent=agent,  
    tools=tools,  
    verbose=True,  
    handle_parsing_errors=_handle_error,  
)
```

```
agent_executor.invoke(  
    {"input": "What is Leo DiCaprio's middle  
name?\n\nAction: Wikipedia"}  
)
```

```
> Entering new AgentExecutor chain...  
Could not parse LLM output: ` I should search  
for "Leo DiCaprio" on Wikipedia  
Action Input: Leo DiCaprio`Could not parse  
LLM output: ` I should search for I should  
look for a section on his personal life  
Action: Wikipedia  
Action Input: Personal lifePage: Personal  
life  
Summary: Personal life is the course or state  
of an individual's life, especiallI should  
look for a section on his early life  
Action: Wikipedia  
Action Input: Early lifeNo good Wikipedia  
Search Result was foundI should try searching  
for "Leonardo DiCaprio" instead  
Action: Wikipedia  
Action Input: Leonardo DiCaprioPage: Leonardo  
DiCaprio  
Summary: Leonardo Wilhelm DiCaprio (;  
Italian: [di'ka:prjo]; born November 1I  
should look for a section on his personal
```


life again

Action: Wikipedia

Action Input: Personal lifePage: Personal life

Summary: Personal life is the course or state of an individual's life, especiallI now know the final answer

Final Answer: Leonardo Wilhelm DiCaprio

> Finished chain.

/Users/harrisonchase/.pyenv/versions/3.10.1/envs/py310/bin/python3 /Users/harrisonchase/.pyenv/versions/3.10.1/envs/py310/bin/python3 packages/wikipedia/wikipedia.py:389: GessedAtP explicitly specified, so I'm using the best available system ("lxml"). This usually isn't a problem, but if you're using another system, or in a different virtual environment, the parser and behave differently.

The code that caused this warning is on line 389 in packages/wikipedia/wikipedia.py. To get rid of this warning, add an additional argument 'features="lxml"' to the BeautifulSoup constructor.

```
lis = BeautifulSoup(html).find_all('li')
```

```
{'input': "What is Leo DiCaprio's middle name?\n\nAction: Wikipedia",
```

```
'output': 'Leonardo Wilhelm DiCaprio'}
```