🏠   Modules   Callbacks   Logging to file

Logging to file

This example shows how to print logs to file. It shows how to use the `FileCallbackHandler`, which does the same thing as `StdOutCallbackHandler`, but instead writes the output to file. It also uses the `loguru` library to log other outputs that are not captured by the handler.

```python
from loguru import logger

from langchain.callbacks import FileCallbackHandler
from langchain.chains import LLMChain
from langchain.llms import OpenAI
from langchain.prompts import PromptTemplate

logfile = "output.log"

logger.add(logfile, colorize=True, enqueue=True)
handler = FileCallbackHandler(logfile)

llm = OpenAI()
prompt = PromptTemplate.from_template("1 + {number} = ")

# this chain will both print to stdout (because verbose=True) and write to
'output.log'
# if verbose=False, the FileCallbackHandler will still write to
'output.log'
chain = LLMChain(llm=llm, prompt=prompt, callbacks=[handler],
verbose=True)
answer = chain.run(number=2)
logger.info(answer)
```

API Reference:

- FileCallbackHandler from `langchain.callbacks`
- LLMChain from `langchain.chains`
- OpenAI from `langchain.llms`
- PromptTemplate from `langchain.prompts`

```
> Entering new LLMChain chain...
Prompt after formatting:
1 + 2 =


  [32m2023-06-01 18:36:38.929 [0m |  [1mINFO     [0m |
[36m__main__ [0m: [36m<module> [0m: [36m20 [0m -  [1m

3 [0m




> Finished chain.
```

Now we can open the file `output.log` to see that the output has been captured.

```
pip install ansi2html > /dev/null
```

```python
from IPython.display import display, HTML
from ansi2html import Ansi2HTMLConverter

with open("output.log", "r") as f:
    content = f.read()

conv = Ansi2HTMLConverter()
html = conv.convert(content, full=True)

display(HTML(html))
```

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title></title>
<style type="text/css">
.ansi2html-content { display: inline; white-space: pre-wrap; word-wrap:
break-word; }
```

```
.body_foreground { color: #AAAAAA; }
.body_background { background-color: #000000; }
.inv_foreground { color: #000000; }
.inv_background { background-color: #AAAAAA; }
.ansi1 { font-weight: bold; }
.ansi3 { font-style: italic; }
.ansi32 { color: #00aa00; }
.ansi36 { color: #00aaaa; }
</style>
</head>
<body class="body_foreground body_background" style="font-size: normal;" >
<pre class="ansi2html-content">


<span class="ansi1">&gt; Entering new LLMChain chain...</span>
Prompt after formatting:
<span class="ansi1 ansi32"></span><span class="ansi1 ansi3 ansi32">1 + 2 =
</span>

<span class="ansi1">&gt; Finished chain.</span>
<span class="ansi32">2023-06-01 18:36:38.929</span> | <span
class="ansi1">INFO    </span> | <span class="ansi36">__main__</span>:<span
class="ansi36">&lt;module&gt;</span>:<span class="ansi36">20</span> -
<span class="ansi1">

3</span>

</pre>
</body>

</html>
```