

D:\myprojects\nlp\agent\agent.py

```
1 # pip install python-dot-env openai
2
3 from openai import OpenAI
4 import os
5 from dotenv import load_dotenv
6 import re, json, webbrowser
7 from datetime import datetime
8
9 # Load environment variables
10 load_dotenv()
11
12 # Create an instance of the OpenAI class
13 openai_client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
14
15 #=====
16 def generate_text_with_conversation(messages, model = "gpt-4o"):
17     response = openai_client.chat.completions.create(
18         model=model,
19         messages=messages
20     )
21     return response.choices[0].message.content
22
23 def get_response_time(url):
24     time1 = datetime.now()
25     webbrowser.open(url)
26     time2 = datetime.now()
27
28     return time2-time1
29
30 def extend_search_new(text, span):
31     start, end = span
32     nest_count = 1 # Starts with 1 since we know '{' is at the start position
33     for i in range(end, len(text)):
34         if text[i] == '{':
35             nest_count += 1
36         elif text[i] == '}':
37             nest_count -= 1
38             if nest_count == 0:
39                 return text[start:i+1]
40     return text[start:end]
41
42 def extract_json(text_response):
43     pattern = r'\{.*?\}'
44     matches = re.finditer(pattern, text_response, re.DOTALL)
45     json_objects = []
46
47     for match in matches:
48         json_str = extend_search_new(text_response, match.span())
49         try:
50             json_obj = json.loads(json_str)
51             json_objects.append(json_obj)
52         except json.JSONDecodeError:
53             continue
54
55     return json_objects if json_objects else None
56 #=====
57
```

```

58 system_prompt = """
59     You run in a loop of Thought, Action, PAUSE, Action_Response.
60     At the end of the loop you output an Answer.
61
62     Use Thought to understand the question you have been asked.
63     Use Action to run one of the actions available to you - then return PAUSE.
64     Action_Response will be the result of running those actions.
65
66     Your available actions are:
67         get_response_time:
68             e.g. get_response_time: naver.com
69                 Returns the response time of a website
70
71     Example session:
72         Question: what is the response time for naver.com?
73
74         Thought: I should check the response time for the web page first.
75
76         Action:
77             {
78                 "function_name": "get_response_time",
79                 "function_parms": {
80                     "url": "naver.com"
81                 }
82             }
83
84         PAUSE
85
86         You will be called again with this:
87             Action_Response: 00:00:00023
88
89         You then output:
90             Answer: The response time for naver.com is 00:00:0.005 seconds.
91
92     """
93
94 user_prompt = "What is the response time for naver.com?"
95
96 messages = [
97     {"role": "system", "content": system_prompt},
98     {"role": "user", "content": user_prompt},
99 ]
100
101 available_actions = {
102     "get_response_time": get_response_time
103 }
104
105 turn_count = 1
106 max_turns = 5
107
108 while turn_count < max_turns:
109     print (f"\nLoop: {turn_count}")
110     print("-----")
111     turn_count += 1
112
113     response = generate_text_with_conversation(messages, model="gpt-4o")
114
115     print("*** response : \n",response)
116
117     json_function = extract_json(response)

```

```
118
119     if json_function:
120         function_name = json_function[0]['function_name']
121         function_parms = json_function[0]['function_parms']
122         if function_name not in available_actions:
123             raise Exception(f"Unknown action: {function_name}: {function_parms}")
124         print(f" *** running {function_name} {function_parms}")
125         action_function = available_actions[function_name]
126         #call the function
127         result = action_function(**function_parms)
128         function_result_message = f"\nAction_Response: {result}"
129
130         messages.append({"role": "user", "content": function_result_message})
131         print(function_result_message)
132     else:
133         break
```