agentchat.contrib.gpt_assistant_agent

GPTAssistantAgent Objects

```
class GPTAssistantAgent(ConversableAgent)
```

An experimental AutoGen agent class that leverages the OpenAI Assistant API for conversational capabilities. This agent is unique in its reliance on the OpenAI Assistant for state management, differing from other agents like ConversableAgent.

__init__

```
def __init__(name="GPT Assistant",
             instructions: Optional[str] = None,
             llm_config: Optional[Union[Dict, bool]] = None,
             overwrite_instructions: bool = False,
             **kwargs)
```

Arguments:

- `name` *str* - name of the agent. It will be used to find the existing assistant by name. Please remember to delete an old assistant with the same name if you intend to create a new assistant with the same name.
- `instructions` *str* - instructions for the OpenAI assistant configuration. When instructions is not None, the system message of the agent will be set to the provided instructions and used in the assistant run, irrespective of the overwrite_instructions flag. But when instructions is None, and the assistant does not exist, the system message will be set to AssistantAgent.DEFAULT_SYSTEM_MESSAGE. If the assistant exists, the system message will be set to the existing assistant instructions.
- `llm_config` *dict or False* - llm inference configuration.
    - assistant_id: ID of the assistant to use. If None, a new assistant will be created.
    - model: Model to use for the assistant (gpt-4-1106-preview, gpt-3.5-turbo-1106).
    - check_every_ms: check thread run status interval
    - tools: Give Assistants access to OpenAI-hosted tools like Code Interpreter and Knowledge Retrieval, or build your own tools using Function calling. ref https://platform.openai.com/docs/assistants/tools
    - file_ids: files used by retrieval in run
- `overwrite_instructions` *bool* - whether to overwrite the instructions of an existing assistant. This parameter is in effect only when assistant_id is specified in llm_config.
- `kwargs` *dict* - Additional configuration options for the agent.
    - verbose (bool): If set to True, enables more detailed output from the assistant thread.
    - Other kwargs: Except verbose, others are passed directly to ConversableAgent.

can_execute_function

```
def can_execute_function(name: str) -> bool
```

Whether the agent can execute the function.

reset

```
def reset()
```

Resets the agent, clearing any existing conversation thread and unread message indices.

clear_history

```
def clear_history(agent: Optional[Agent] = None)
```

Clear the chat history of the agent.

Arguments:

- `agent` - the agent with whom the chat history to clear. If None, clear the chat history with all agents.

pretty_print_thread

```python
def pretty_print_thread(thread)
```

Pretty print the thread.

oai_threads

```python
@property
def oai_threads() -> Dict[Agent, Any]
```

Return the threads of the agent.

assistant_id

```python
@property
def assistant_id()
```

Return the assistant id

get_assistant_instructions

```python
def get_assistant_instructions()
```

Return the assistant instructions from OAI assistant API

delete_assistant

```python
def delete_assistant()
```

Delete the assistant from OAI assistant API

find_matching_assistant

```python
def find_matching_assistant(candidate_assistants, instructions, tools,
                            file_ids)
```

Find the matching assistant from a list of candidate assistants. Filter out candidates with the same name but different instructions, file IDs, and function names. TODO: implement accurate match based on assistant metadata fields.

✏ Edit this page

Community

Discord ⬈

Twitter ⬈