

1. Modifying the conversational.py file

Memory Initialization:

- Replaced the Langchain REACT Agent's memory with `XAgentMemory` for consistent memory handling.

Callback Handling:

- Introduced `XAgentCallbackHandler` for managing callbacks specific to the XAgent framework.
- Added the XAgent callback handler to the list of callbacks during agent initialization.

XAgent Initialization:

- Replaced the Langchain REACT Agent initialization with `XAgent` instantiation.
- Configured XAgent with the necessary parameters, including the language model, agent type, memory, error handling, system message, output parser, and callbacks.

Handling Errors:

- Saved context in `XAgentMemory` in case of an exception, providing input and chat history for further analysis.

Voice-Related Code:

- Retained most of the existing voice-related code, adapting it to work seamlessly with XAgent.

Sending AI Messages:

- Utilized XAgent's output (`res`) to create an AI message for history tracking and communication via the `chat_pubsub_service`.

2. Modifying the dialogue_agent_with_tools.py file

Memory Initialization:

- Initialized `memory` with `XAgentMemory` instead of `ZepMemory` for consistent memory handling with XAgent.

Callback Handling:

- Introduced `XAgentCallbackHandler` for managing callbacks specific to the XAgent framework.
- Added the XAgent callback handler to the list of callbacks during agent

initialization.

XAgent Initialization:

- Replaced the initialization of the Langchain REACT Agent with `XAgent` instantiation.
- Configured XAgent with the necessary parameters, including the tools, language model, agent type, memory, error handling, system message, output parser, and callbacks.

Handling Errors:

- Saved context in `XAgentMemory` in case of an exception, providing input and chat history for further analysis.

Additional XAgent Properties:

- Set `system_message` and `output_parser` for XAgent, adapting them to the new framework.

Sending Messages:

- Utilized XAgent's output (`res`) to create an `AIMessage` for further processing or communication.

3. Modifying the test.py file

Step 1: Replaced Langchain Imports with XAgent Imports

Replaced Langchain-related imports with XAgent imports at the beginning of the testing script.

Step 2: Modified the `agent_factory` Function

Updated the `agent_factory` function to initialize an instance of the XAgent instead of the Langchain REACT Agent.

Step 3: Set Up Mock Objects and Dependencies

In the `setUp` method of the test class, set up mock objects and dependencies, including the XAgent-related ones.

Step 4: Modified Test Method

Updated the test method to use the XAgent-based `agent_factory`.

