

Day91_LangChain_HuggingFace_QA

September 27, 2025

1 LangChain – Q&A with Memory (Updated for HuggingFaceEndpoint)

This notebook will guide you from **understanding LangChain** to **building a simple AI assistant** that can **remember past questions**.

Note: LangChain is **actively developed**. Libraries, methods, and classes may change frequently. If you encounter errors, always **check the official LangChain documentation**: <https://www.langchain.com/docs/> and make sure your packages are up-to-date.

2 What is LangChain?

LangChain is like a **toolkit to build smart AI assistants**.

It combines:

- **Brain (LLM)** – The AI that generates answers
- **Prompts** – Instructions you give to the AI
- **Chains** – Step-by-step workflows
- **Agents** – Smart decision-makers that can use tools
- **Memory** – Allows the AI to remember past conversations

Analogy: Think of LangChain as **your AI friend** with a **brain, notepad, and access to tools**.

Tip: LangChain frequently updates APIs and methods. If something doesn't work, check the official docs for the latest method or parameter names.

3 Key Components

1. **LLM (Large Language Model)** – Brain that understands and generates text
2. **PromptTemplate** – Template for asking questions clearly

3. **Chain** – Series of steps to answer questions
4. **Agent** – Chooses the right tool for the task
5. **Memory** – Remembers conversation history

4 Installation

Run this in a notebook cell:

```
!pip install langchain transformers langchain-huggingface
```

5 Get HuggingFace API Token

1. Go to [HuggingFace](#) and log in or create an account
2. Go to **Settings** → **Access Tokens** → **New token**
3. Copy your token
4. Set it in Python:

```
import os
os.environ["HUGGINGFACEHUB_API_TOKEN"] = "YOUR_TOKEN_HERE"
```

Replace "YOUR_TOKEN_HERE" with your actual token.

6 Import Libraries

```
from langchain_huggingface import HuggingFaceEndpoint
from langchain.chains import ConversationChain
from langchain.prompts import PromptTemplate
from langchain.memory import ConversationBufferMemory
```

Note: If you get an import error, check LangChain docs—the import paths may have changed in updates.

7 Load the Language Model (Brain)

```
llm = HuggingFaceEndpoint(
    endpoint_url="https://api-inference.huggingface.co/models/google/flan-t5-large",
    model_kwargs={"temperature":0, "max_length":256}
)
```

This model is our **smart friend** who can answer questions in simple words.

8 Create a Prompt Template

```
prompt = PromptTemplate(
    input_variables=["question"],
    template="Answer the following question in simple words: {question}"
)
```

This is like writing **clear instructions** on a sticky note for our AI friend.

9 Add Memory

```
memory = ConversationBufferMemory()
```

Memory acts like a **notepad** where the AI remembers everything you ask.

10 Create the Q&A Chain with Memory

```
conversation = ConversationChain(
    llm=llm,
    memory=memory,
    prompt=prompt
)
```

Now the AI can **answer questions and remember previous ones**.

11 Ask Questions

```
# First question
answer1 = conversation.run("What is LangChain?")
print("AI:", answer1)

# Second question
answer2 = conversation.run("Can you explain its key components?")
print("AI:", answer2)

# Third question referring to previous context
answer3 = conversation.run("Which part is like the brain?")
print("AI:", answer3)
```

Notice how the AI **uses memory** to answer related questions.

12 See the Conversation History

```
print("Conversation History:\n")
print(memory.buffer)
```

You will see all your previous questions and AI answers, just like a chat history.

13 Next Steps / Ideas

- Explore **Agents** to let AI automatically choose tools
- Connect **APIs** (e.g., weather, calculator, Wikipedia)
- Use **custom prompts** for different answer styles (simple, detailed, humorous)
- Add **more advanced memory** to remember user preferences over long chats