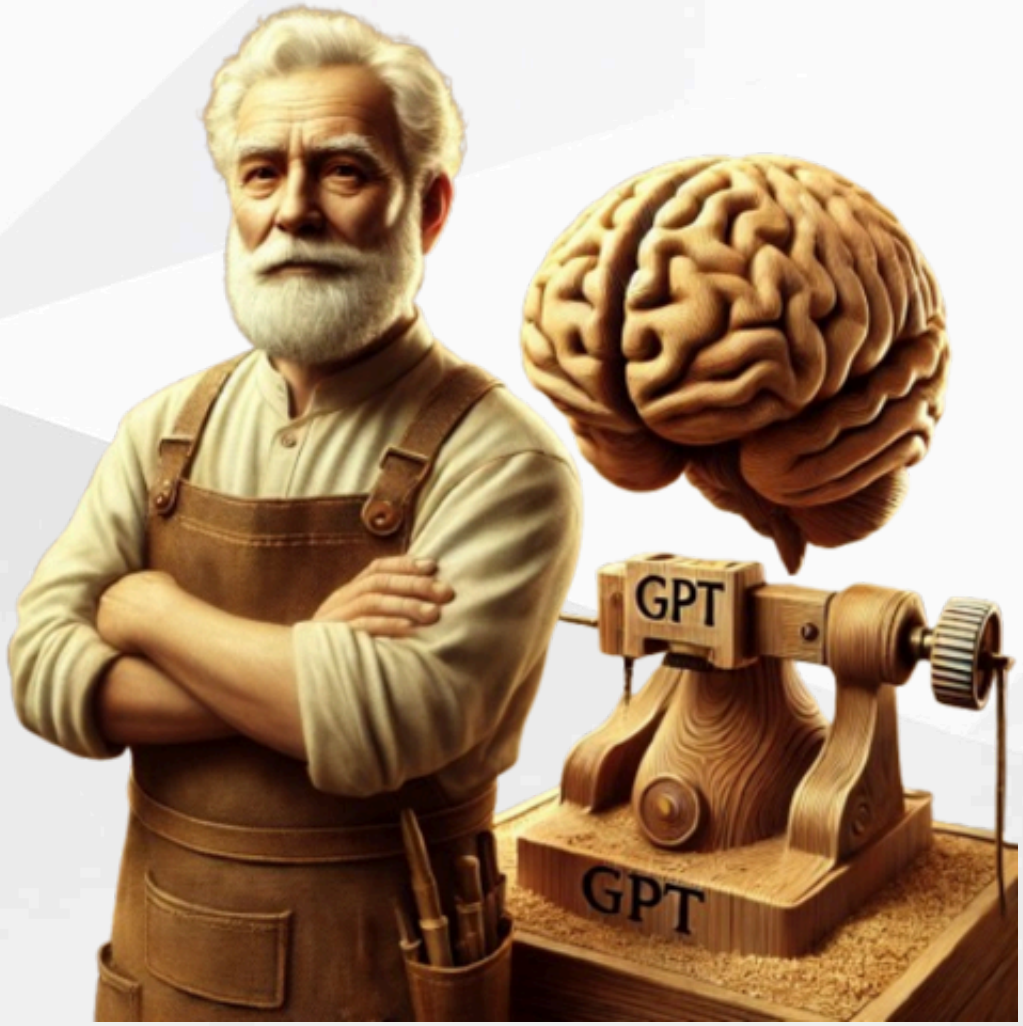
Developing Open LLM applications with

Apache OpenServerless

Lesson 3

Form and Display support



Form and Display Support

- OpenAI api
- A chat class
- Redis
- History Class
- An stateful assistant

OpenAI API

OpenAI API Intro

- The first API developed
  - the "de-facto" standard
- Everyone uses it
  - We can use it with private AI also
  - You can hook in other providers

OpenAI API: create a connection

- `base_url`: the location of the api key server
- `api_key`: the authentication key (if required)

```
import os, openai
# ollama configuratin
host =  os.getenv("OLLAMA_HOST")
api_key = os.getenv("AUTH")
base_url = f"https://{api_key}@{host}/v1"
# accessing to the server
client = openai.OpenAI(
    base_url = base_url,
    api_key = api_key,
)
```

OpenAI API: `messages` is a list of message

- a message:

```
message = {
    "role": "user",
    "content": "What is the capital of Italy"
}
```

- Roles:
  - `system`: configuration
  - `user`: user requests
  - `assistant`: assistant responses

OpenAI API: completions

Request: `messages` is a list of message:

```
MODEL= "llama3.1:8b"
messages = [message]
res = client.chat.completions.create(
    model=MODEL,
    messages=[message],
)
```

Response:

```
res.choices[0].message.content
```

OpenAI API: streaming

- Add `stream: True`

```
res = client.chat.completions.create(
    model=MODEL,
    messages=[message],
    stream = True
)
```

Receive a stream:

```
for m in res:
   print(m.choices[0].delta.content)
```

Chat Class

Classes in Python

```
class Counter:
  def __init__(self):
    self.value = 0

  def count(self):
    self.value += 1
    return self.value
```

Esempio:

```
c = Counter()
c.count()
c.count()
```

Wrapping OpenAI

- Inspecting the code

  `!code packages/assistant/api/chat.py`

- Testing the Chat class

```
import sys ; sys.path.append("packages/assistant/api") ; import chat
ch = chat.Chat({})
ch.add("system:I tell the country and you tell me the capital.")
ch.complete()
ch.add("user:Italy")
ch.complete()
ch.add("user:France")
ch.complete()
ch.messages
```

Redis

History Class

Assistant