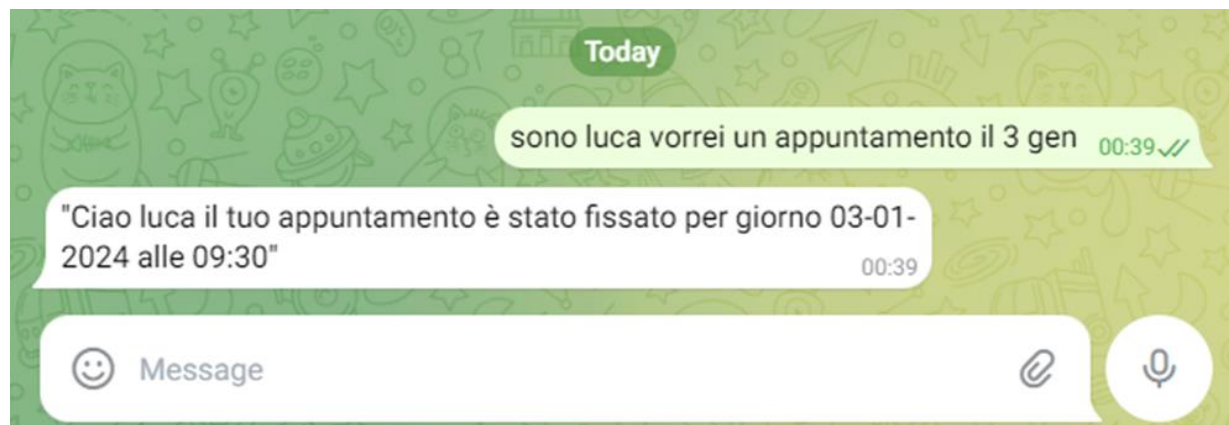
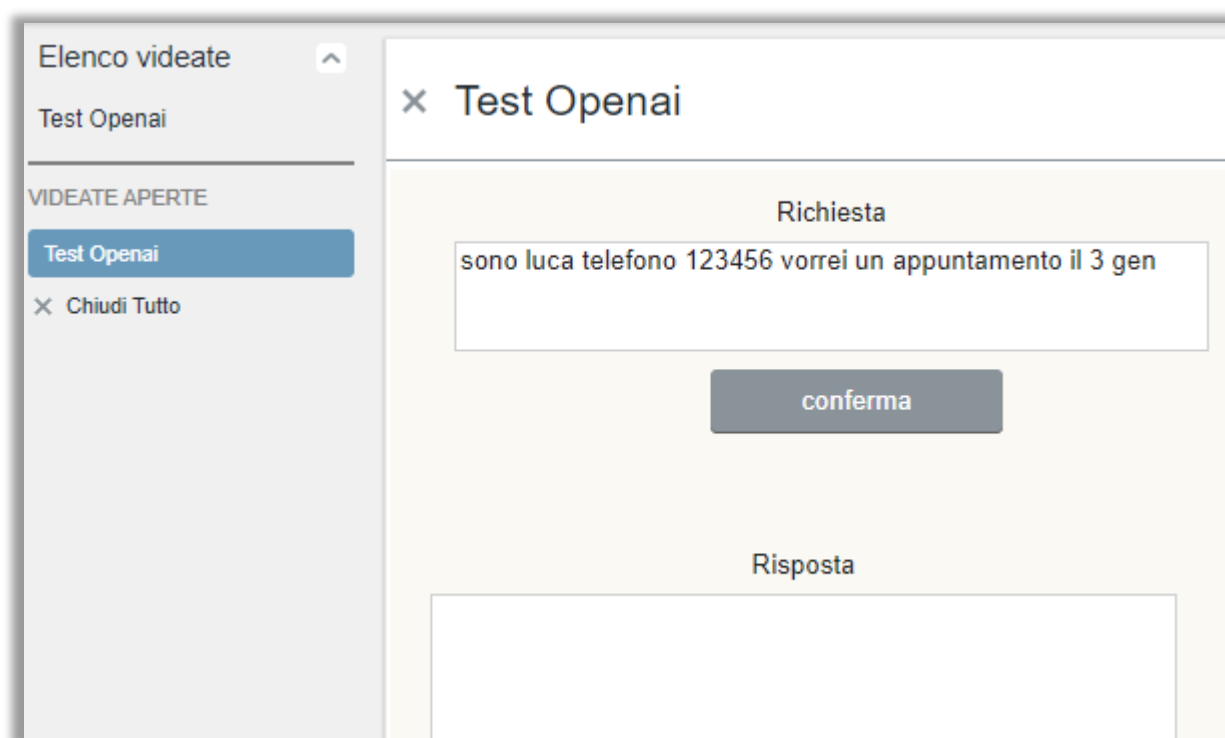


Esempio di collaborazione Inde-Assistente

Questo è il dialogo sul BOT che ho creato su Telegram, avrei potuto usare **Inde** ma ci sono casi in cui è preferibile l'uso di Telegram.



Posso ovviamente fare una cosa equivalente con Inde:

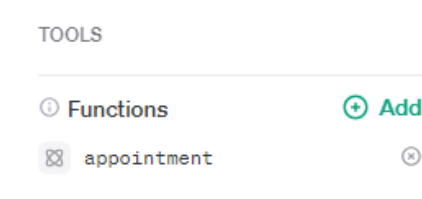


Elaborazione con INDE

Nel playground (<https://platform.openai.com/playground>) ho creato e istruito l'assistente:



Ho aggiunto all'assistente una funzione nei tools:



Ho caricato i dati della funzione:

La funzione ha un nome che la identifica e dei parametri, ma soprattutto una descrizione che ne chiarisce lo scopo.

```
{
  "name": "appointment",
  "parameters": {
    "type": "object",
    "properties": {
      "nome": {
        "type": "string",
        "description": "nome del soggetto"
      },
      "telefono": {
        "type": "string",
        "description": "numero del telefono"
      },
      "data": {
        "type": "string",
        "description": "data approssimativa e.g. domani"
      },
      "ora": {
        "type": "string",
        "description": "ora, se assente 00:00"
      }
    }
  },
  "required": [
    "nome",
    "telefono"
  ],
  "description": "la funzione fissa un appuntamento come richiesto dal cliente se c'è la disponibilità o un appuntamento nel primo giorno disponibile"
}
```

Quindi l'assistente ha tutte le informazioni che gli servono per operare.

Nel momento in cui gli trasmettiamo il messaggio dell'utente nel testo vede "appuntamento" e quindi sceglie fra le funzioni che gli abbiamo fornito quella più adatta, in questo caso "appointment".

Poi controlla quali parametri sono richiesti dalla funzione e li estrae dal messaggio.

A questo punto ci chiede di eseguire la funzione.

Ecco cosa ci arriva dall'assistente:

```
"required_action": {
  "type": "submit_tool_outputs",
  "submit_tool_outputs": {
    "tool_calls": [
      {
        "id": "call_xSBlyZ2d89Fahg4r3aNfmBcZ",
        "type": "function",
        "function": {
          "name": "appointment",
          "arguments": "{\"nome\":\"luca\", \"telefono\":347123456, \"data\":\"03/01/2024\", \"ora\":\"00:00\"}"
        }
      }
    ]
  }
}
```

...

Eseguiamo la nostra funzione

```
public date time GestioneAppuntamento(
```

```
string nome    //
```

```
string telefono //
```

```
string data    //
```

```
string ora     //
```

```
)
```

```
{
```

```
date time dt = #31/12/1999 02:00#
```

```
...
```

```
return dt
```

```
}
```

Supponiamo che la nostra funzione restituisca: 03/01/2024 09:30. Possiamo comporre ad esempio questo messaggio di risposta: **"Ciao luca, ecco il tuo appuntamento: 03/01/2024 09:30, se hai problemi contattaci!"**.

L'assistente fondamentale ci aiuta grazie alla sua capacità di comprendere il linguaggio naturale ma non solo, potremmo usare il tool **code interpreter** per esempio per fargli eseguire calcoli strutturali durante la fase di costruzione o ristrutturazione di un edificio, in questo caso non gli forniremo noi la funzione, se la scrive da solo!

```
{
  "instructions": "Sei un ingegnere strutturale. Quando ti viene chiesto, scrivi ed esegui il codice per soddisfare le richieste.",
  "tools": [{"type": "code_interpreter"}],
  "model": "gpt-4-1106-preview",
  "file_ids": ["file_123abc456"]
}
```

Pensate a quante cose possiamo aggiungere ai nostri applicativi che i nostri clienti neanche si sognano!