履歴付き対話チュートリアル

はじめに

本書は、本サービスが提供する履歴付き対話APIによって対話機能を実現するクライアントアプリケーションを作成するためのチュートリアルです。

本書の対象読者は以下を想定しています。

- 本サービスと連携したシステムや製品開発を行う開発者

なお、本書ではPythonのサンプルコードを記載しておりますが、本サービスのAPIはREST形式のため、他の言語からもご利用いただけます。
サポートしているAPIとリクエスト・レスポンスの詳細については、APIリファレンスを参照してください。

履歴付き対話APIとは

LLMを使った対話機能を利用するAPIです。OpenAI APIよりも高度な過去の会話履歴を利用し、LLMと一般対話を行うことができます。

チュートリアルの流れ

本チュートリアルの流れは以下です。

curlコマンドによるAPI呼び出し

1. 新規に会話を開始する
2. 過去の会話履歴を踏まえて会話する
3. 履歴を使わずに単発で会話する
4. streaming形式で会話する

Pythonプログラムによる API呼び出し

1. requestsライブラリを利用して新規に会話を開始する

curlコマンドによるAPI呼び出し

APIの実行方法

以下のcurlコマンドを用いて、APIを呼び出します。APIの詳細は「Generative AI Cloud（SaaS）APIリファレンス」を参照してください。
後述のリクエストパラメータを用いることで、履歴を扱った会話を行うことができます。

```
curl -X POST https://api.genai-api.nec-cloud.com/genai-api/v1/chat -H "Content-Type: application/json" ¥
-d "<リクエストパラメータ>" -H "x-nec-genai-client-id:<ユーザId>" -H "Authorization: <API Key>"
```

新規に会話を開始する

以下は新規に会話を始める場合に指定するリクエストパラメータの例です。
historyIdにnewを指定することで新規会話を開始することができます。

```
{
  "userContent": "こんにちは、田中です",
```

```
  "systemContent": "あなたはAIアシスタントです",
  "historyId": "new",
  "model": "cotomi-fast-v2.0"
}
```

応答がJson形式で返却されます。

```
{"answer": "こんにちは田中さん、何かお手伝いすることはありますか？",
 "historyId":"historyid_12345678-90ab-cdef-ghij-zzz"}
```

過去の会話履歴を踏まえて会話する

「新規に会話を開始する」で行った会話の内容を踏まえて、会話を続ける方法について説明します。

以下は過去の会話履歴を利用した会話のリクエストパラメータの例です。
historyIdに過去の会話呼び出し時に応答で取得したIDを指定します。
なお、historyIdは会話履歴一覧APIからも取得できます。

```
{
  "userContent": "私の名前をおぼえていますか",
  "systemContent": "あなたはAIアシスタントです",
  "historyId": "historyid_12345678-90ab-cdef-ghij-zzz",
  "model": "cotomi-fast-v2.0"
}
```

応答がJson形式で返却されます。
historyIdは指定時と同じ値が返却されます。以降、同じ履歴での会話を続ける場合は、同じhistoryIdの値を指定してください。

```
{"answer": "田中さんですよね、覚えています。何かお手伝いすることはありますか？",
 "historyId":"historyid_12345678-90ab-cdef-ghij-zzz"}
```

履歴を使わず単発で会話する

以下は履歴機能を使用しない対話のリクエストパラメータの例です。

historyIdは指定不要です。

```
{
  "userContent": "こんにちは、田中です",
  "systemContent": "あなたはAIアシスタントです",
  "oneshot": true,
  "model": "cotomi-fast-v2.0"
}
```

応答がJson形式で返却されます。historyIdは返却されません。

```
{"answer": "こんにちは田中さん、何かお手伝いすることはありますか？"}
```

streaming形式で会話する

streaming形式で会話することで、応答をまとめてではなく、順次行うように指定することができます。
リクエストパラメータで、**stream**と**streamNum**を指定します(streamNumは任意)。

```
{
  "userContent": "こんにちは",
  "systemContent": "あなたはAIアシスタントです",
  "historyId": "historyid_12345678-90ab-cdef-ghij-zzz",
  "model": "cotomi-fast-v2.0",
```

```
  "stream": true,
  "streamNum": 1
}
```

streaming形式の応答では、以下のSSEの返却仕様に従った形式で返却します。
サーバー送信イベントの使用 - Web API | MDN

streaming形式の返却仕様は以下の通りです。

- event:で返却データ種別を返却、data:で実際のデータを返却します。eventの種別はsystem(履歴などシステム上の情報返却)、error(異常系)、done(終了)の3つ。最初の応答メッセージにはeventはありません
- systemは全ての応答メッセージを返却し終えた後に送信します。全てのデータが送信したら正常系、異常系かかわらずevent: doneを送信する
- errorのJson定義詳細は「APIエラー」を参照
- 個々の通知は、2つの改行で終わるテキストのブロックとして送信されます

応答の一例は以下の通りです。

```
 data: {"answer" : "こ"}
 data: {"answer" : "ん"}
  :
 event: system

 data: {"historyId": "historyid_12345678-90ab-cdef-ghij-zzz"}

 event: done
```

Pythonプログラムによる API呼び出し

requestsライブラリを利用して新規に会話を開始する

requestsライブラリのインストール

インストールコマンドの例

```
pip install requests
```

Pythonコードを実装する

Pythonコードの例

Pythonのエディタを開き、下記のコードを記述します。KEYの値を適切なものに修正し、ファイル名を「chatapi_main.py」として保存します。

```
import requests

# KEYはAPIキー
KEY = 'abcdefg1234567890'
# BASEはサービスのURL
BASE = 'https://api.genai-api.nec-cloud.com/genai-api/v1'
# MODELはエイリアス名
MODEL = 'cotomi-fast-v2.0'

def func():
    url = BASE + '/chat'
    key = 'Bearer ' + KEY
```

```
    payload = { "userContent": "こんにちは、田中です",
                "systemContent": "あなたはAIアシスタントです",
                "historyId": "new",
                "model": MODEL }
    headers = { 'content-type': 'application/json',
                'x-nec-genai-client-id': 'ABCDEF',
                'Authorization': key }
    # 非ストリーム形式
    response = requests.post(url, json=payload, headers=headers)
    print(response.status_code)
    print(response.text)
    #print(response.headers)

if __name__ == '__main__':
    func()

```

Pythonの実行

実行コマンド

```
python chatapi_main.py
```

実行結果の例

```
200
{"answer":"こんにちは、田中さん！何かお手伝いできることはありますか？どんな質問にもお答えし、必要な情報を提供...",
 "historyId":"historyid_12345678-90ab-cdef-ghij-zzz"}
```