

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、他の言語からの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。

-  本ガイドに記載する「https://<サーバのドメイン名>/」はGenerative AI FWがインストールされているサーバのドメイン名に置き換えてAPIを実行してください。
- 本サービスではHTTPSに既定では自己証明書を使用しています。そのためAPI利用時に考慮が必要です。詳細は「スタートアップマニュアル（導入準備編）」をご確認ください。

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 FW 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。

```
1 curl -X POST https://<サーバのドメイン名>/genai-api/v1/chat -H "Content-Type: application/json" ¥  
2 -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。

```
1 {  
2   "userContent": "こんにちは、田中です",  
3   "systemContent": "あなたはAIアシスタントです",  
4   "historyId": "new",  
5   "model": "cotomi-v3.0"  
6 }
```

応答がJson形式で返却されます。

```
1 {"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?",  
2  "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。

```
1 {  
2   "userContent": "私の名前をおぼえていますか",  
3   "systemContent": "あなたはAIアシスタントです",  
4   "historyId": "historyid_12345678-90ab-cdef-ghij-zzz",  
5   "model": "cotomi-v3.0"  
6 }
```

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。

```
1 {"answer": "田中さんですよ、覚えています。何かお手伝いすることはありますか?",  
2  "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。

```
1 {
2   "userContent": "こんにちは、田中です",
3   "systemContent": "あなたはAIアシスタントです",
4   "oneshot": true,
5   "model": "cotomi-v3.0"
6 }
```

応答がJson形式で返却されます。historyIdは返却されません。

```
1 {"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)。

```
1 {
2   "userContent": "こんにちは",
3   "systemContent": "あなたはAIアシスタントです",
4   "historyId": "historyid_12345678-90ab-cdef-ghij-zzz",
5   "model": "cotomi-v3.0",
6   "stream": true,
7   "streamNum": 1
8 }
```

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、異常系にかかわらず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。

```
1 data: {"answer" : "こ"}
2 data: {"answer" : "ん"}
3 :
4 event: system
5
6 data: {"historyId": "historyid_12345678-90ab-cdef-ghij-zzz"}
7
8 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

```
1 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。

```
1 import requests
2
3 # KEYはAPIキー
4 KEY = 'abcdefg1234567890'
5 # BASEはサービスのURL
6 BASE = 'https://<サーバのドメイン名>/genai-api/v1'
7 # MODELはエイリアス名
8 MODEL = 'cotomi-v3.0'
9
10 def func():
11     url = BASE + '/chat'
12     key = 'Bearer ' + KEY
13     payload = { "userContent": "こんにちは、田中です",
14                "systemContent": "あなたはAIアシスタントです",
15                "historyId": "new",
16                "model": MODEL }
17     headers = { 'content-type': 'application/json',
18                'x-nec-genai-client-id': 'ABCDEF',
19                'Authorization': key }
20     # 非ストリーム形式
21     response = requests.post(url, json=payload, headers=headers)
22     print(response.status_code)
23     print(response.text)
24     #print(response.headers)
25
26 if __name__ == '__main__':
27     func()
28
```

Pythonの実行

実行コマンド

```
1 python chatapi_main.py
```

実行結果の例

```
1 200
2 {"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
3  "historyId": "historyid_12345678-90ab-cdef-ghij-zzz"}
```