

テンプレート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テンプレート対話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、他の言語からで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。

-  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。

テンプレート対話APIとは

LLMを使った対話機能を利用するAPIです。あらかじめ設定されたプロンプトのテンプレートを用いて対話を行うことができます。テンプレート対話はサービス側で用意したシステムテンプレートを用いた対話と、システム管理者が作成したユーザテンプレートを用いた対話を行うことができます。

システムテンプレートは以下を提供しています

- 要約テンプレート
- 翻訳テンプレート（日本語へ翻訳）

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。

要約編

1. 要約テンプレートを利用する
2. (応用) requestsライブラリを利用したpythonプログラムで要約テンプレートを利用する

翻訳編

1. 翻訳テンプレートを利用する
2. (応用) requestsライブラリを利用したpythonプログラムで翻訳テンプレートを利用する

応用編

1. 管理ポータルでユーザテンプレートを作成する
2. ユーザテンプレートを利用する
3. (応用) requestsライブラリを利用したpythonプログラムでユーザテンプレートを利用する

要約編

要約テンプレートを利用する

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、要約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/templatechat -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>"
```

以下は要約を行なう場合に指定するリクエストパラメータの例です。

templateIdにsummaryを指定することで要約を指示します。

```
1 {  
2   "templateId": "summary",  
3   "userContent": "COVID-19パンデミックは、世界的なロックダウン措置と経済の混乱をもたらしました。...",  
4   "model": "cotomi-v3.0"  
5 }
```

応答がJson形式で返却されます。

```
1 {"answer": "COVID-19パンデミックは、世界的な影響をもたらし、教育や交流を制限する結果となった。..."}
```

requestsライブラリを利用したpythonプログラムで要約テンプレートを利用する

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。ファイル名を「summary_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 + '/templatechat'
12     key = 'Bearer ' + KEY
13     payload = { "templateId": "summary",
14                 "userContent": "COVID-19パンデミックは、世界的なロックダウン措置と経済の混乱をもたらしました",
15                 "model": MODEL }
16     headers = { 'content-type': 'application/json',
17                 'x-nec-genai-client-id': 'ABCDEF',
18                 'Authorization': key }
19     # 非ストリーム形式
20     response = requests.post(url, json=payload, headers=headers)
21     print(response.status_code)
22     print(response.text)
23     #print(response.headers)
24
25 if __name__ == '__main__':
26     func()
27
```

Pythonの実行

実行コマンド

```
1 python summary_main.py
```

実行結果の例

```
1 200
2 {"answer": "COVID-19パンデミックは、世界的な影響をもたらし、教育や交流を制限する結果となった。..."}
```

翻訳編

翻訳テンプレートを利用する

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、翻訳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/templatechat -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>"
```

以下は翻訳を行なう場合に指定するリクエストパラメータの例です。

templateIdにtranslationを指定することで翻訳を指示します。

```
1 {
2   "templateId" : "translation",
3   "userContent" : "The rise of artificial intelligence has revolutionized various industries, ...",
4   "model" : "cotomi-v3.0"
5 }
```

応答がJson形式で返却されます。

```
1 {"answer": "COVID-19パンデミックは、世界的な影響をもたらし、教育や交流を制限する結果となった。..."}
```

requestsライブラリを利用したpythonプログラムで翻訳テンプレートを利用する

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。ファイル名を

「translation_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 + '/templatechat'
12     key = 'Bearer ' + KEY
13     payload = { "templateId" : "translation",
14                "userContent" : "The rise of artificial intelligence has revolutionized various...",
15                "model" : MODEL }
16     headers = { 'content-type': 'application/json',
17                'x-nec-genai-client-id': 'ABCDEF',
18                'Authorization': key }
19     # 非ストリーム形式
20     response = requests.post(url, json=payload, headers=headers)
21     print(response.status_code)
22     print(response.text)
23     #print(response.headers)
24
25 if __name__ == '__main__':
26     func()
27
```

Pythonの実行

実行コマンド

```
1 python translation_main.py
```

実行結果の例

```
1 200
2 {"answer": "人工知能の発展は、医療、金融、運輸など、さまざまな産業に革命をもたらしています。..."}
```

応用編

管理ポータルでユーザテンプレートを作成する

システム管理者は、管理ポータル上で任意のテンプレートを作成・管理することができます。

詳細は「管理ポータル操作ガイド（テンプレート機能編）」をご確認ください。

なお、テンプレートのユーザープロンプトにパラメータ({userContent})を記載することで、テンプレート対話API実行時のリクエストパラメータ(userContent)で指定する任意の文字列にテンプレート内容を置換して対話することができます。パラメータの記載例は管理ポータルからシステムテンプレート（要約）の内容をご確認ください。

ユーザテンプレートを利用する

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、ユーザテンプレートでの対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/templatechat -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>"
```

以下はリクエストパラメータの例です。templateIdにはユーザテンプレートのIDを指定します。userContentには、テンプレートでパラメータ({userContent})を使用する場合は置換したい文字列を、使用しない場合は空文字を指定してください。

```
1 {
2   "templateId" : "template_12345678-90ab-cdef-ghij-zzz",
3   "userContent": "",
4   "model": "cotomi-v3.0"
5 }
```

応答がJson形式で返却されます。

```
1 {"answer": "こんにちは！..."}
```

requestsライブラリを利用したpythonプログラムでユーザテンプレートを利用する

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。ファイル名を「template_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 + '/templatechat'
12     key = 'Bearer ' + KEY
13     payload = { "templateId" : "template_12345678-90ab-cdef-ghij-zzz",
14                "userContent": "",
15                "model": MODEL }
16     headers = { 'content-type': 'application/json',
17                'x-nec-genai-client-id': 'ABCDEF',
18                'Authorization': key }
19     # 非ストリーム形式
20     response = requests.post(url, json=payload, headers=headers)
21     print(response.status_code)
22     print(response.text)
23     #print(response.headers)
24
25 if __name__ == '__main__':
26     func()
27
```

Pythonの実行

実行コマンド

```
1 python template_main.py
```

実行結果の例

```
1 200
2 {"answer": "こんにちは！..."}
```

