rpx-santanderdevweek2023

October 2, 2023

0.1 Extracao da API

1 Usando API e Chat Gpt para criar uma mensagem personalizada.

```
[3]: # Utilize sua própria URL se quiser ;)
     # Repositório da API: https://github.com/digitalinnovationone/
      ↪santander-dev-week-2023-api
     sdw2023_api_url = 'https://sdw-2023-prd.up.railway.app'
```

1.1 Extract

Extraindo a lista de IDs de usuário a partir do arquivo CSV. Para cada ID, faça uma requisição GET para obter os dados do usuário correspondente.

```
[4]: import pandas as pd

     df = pd.read_csv('SDW2023.csv')
     user_ids = df['UserID'].tolist()
     print(user_ids)
```

```
[4, 5, 6]
```

```
[5]: import requests
     import json

     def get_user(id):
       response = requests.get(f'{sdw2023_api_url}/users/{id}')
       return response.json() if response.status_code == 200 else None

     users = [user for id in user_ids if (user := get_user(id)) is not None]
     print(json.dumps(users, indent=2))
```

```
[]
```

1.2 Transform

Utilizando a API do OpenAI GPT-4 para gerar uma mensagem personalizada para cada usuário.

```
[6]: !pip install openai
```

```
Collecting openai
  Downloading openai-0.28.1-py3-none-any.whl (76 kB)
                            77.0/77.0 kB
1.4 MB/s eta 0:00:00
Requirement already satisfied: requests>=2.20 in
/usr/local/lib/python3.10/dist-packages (from openai) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from openai) (4.66.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-
packages (from openai) (3.8.5)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests>=2.20->openai) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai)
(2023.7.22)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp->openai) (23.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (4.0.3)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp->openai) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.3.1)
Installing collected packages: openai
Successfully installed openai-0.28.1
```

```python
[16]: #API Key da OpenAI, ela será salva como uma variável de ambiente.
      openai_api_key = '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'
```

```python
[17]: import openai

      openai.api_key = openai_api_key

      def generate_ai_news(user):
        completion = openai.ChatCompletion.create(
          model="gpt-4",
          messages=[
```

```
    {
        "role": "system",
        "content": "Você é um especialista financeiro, com trabalho de
↪aprovar ou não aumento de limite em 20% dependendo do saldo bancário de cada
↪cliente."
    },
    {
        "role": "user",
        "content": f"Crie uma mensagem para {user['name']} se o aumento de
↪limite dele foi aprovado ou não.. (máximo de 100 caracteres)"
    }
  ]
)
return completion.choices[0].message.content.strip('\"')

for user in users:
  news = generate_ai_news(user)
  print(news)
  user['news'].append({
      "icon": "https://digitalinnovationone.github.io/
↪santander-dev-week-2023-api/icons/credit.svg",
      "description": news
  })
```

1.3 Load

Atualize a lista de "news" de cada usuário na API com a nova mensagem gerada.

```
[18]: def update_user(user):
    response = requests.put(f"{sdw2023_api_url}/users/{user['id']}", json=user)
    return True if response.status_code == 200 else False

for user in users:
  success = update_user(user)
  print(f"User {user['name']} updated? {success}!")
```