

▾ Santander Dev Week 2023 (ETL com Python)

Contexto: Você é um cientista de dados no Santander e recebeu a tarefa de envolver seus clientes de maneira mais personalizada. Seu objetivo é usar o poder da IA Generativa para criar mensagens de marketing personalizadas que serão entregues a cada cliente.

Condições do Problema:

1. Você recebeu uma planilha simples, em formato CSV ('SDW2023.csv'), com uma lista de IDs de usuário do banco:

```
UserID
1
2
3
4
5
```

2. Seu trabalho é consumir o endpoint GET `https://sdw-2023-prd.up.railway.app/users/{id}` (API da Santander Dev Week 2023) para obter os dados de cada cliente.
3. Depois de obter os dados dos clientes, você vai usar a API do ChatGPT (OpenAI) para gerar uma mensagem de marketing personalizada para cada cliente. Essa mensagem deve enfatizar a importância dos investimentos.
4. Uma vez que a mensagem para cada cliente esteja pronta, você vai enviar essas informações de volta para a API, atualizando a lista de "news" de cada usuário usando o endpoint PUT `https://sdw-2023-prd.up.railway.app/users/{id}`.

```
# 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'
```

▾ Extract

Extraia 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.

```
import pandas as pd

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

[2543, 2544, 2545]

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))

[
  {
    "id": 2543,
    "name": "Mariana",
    "account": {
      "id": 2673,
      "number": "88888-1",
      "agency": "9999",
      "balance": 0.0,
      "limit": 200.0
    },
    "card": {
      "id": 2467,
      "number": "1231 1231 1231 4444",
      "limit": 1000.0
    },
    "features": [],
    "news": [
      {
        "id": 5431,
        "icon": "https://digitalinnovationone.github.io/santander-dev-week-2023-api/icons/credit.svg",
        "description": "Para o seu dinheiro crescer"
```

```

    }
  ]
},
{
  "id": 2544,
  "name": "Bruno",
  "account": {
    "id": 2674,
    "number": "88888-2",
    "agency": "9999",
    "balance": 0.0,
    "limit": 200.0
  },
  "card": {
    "id": 2468,
    "number": "1231 1231 1231 3333",
    "limit": 1000.0
  },
  "features": [],
  "news": [
    {
      "id": 5432,
      "icon": "https://digitalinnovationone.github.io/santander-dev-week-2023-api/icons/credit.svg",
      "description": "Quer saber como investir na sua conta bancaria? Conte com Santander"
    }
  ]
},
{
  "id": 2545,
  "name": "Henry",
  "account": {
    "id": 2675,
    "number": "88888-3",
    "agency": "9999",
    "balance": 0.0,
    "limit": 200.0
  }
}

```

▼ Transform

Utilize a API do OpenAI GPT-4 para gerar uma mensagem de marketing personalizada para cada usuário.

```
!pip install openai
```

```

Collecting openai
  Downloading openai-0.28.0-py3-none-any.whl (76 kB)
    76.5/76.5 kB 2.1 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)
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)
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-0.28.0

```

```

# Documentação Oficial da API OpenAI: https://platform.openai.com/docs/api-reference/introduction
# Informações sobre o Período Gratuito: https://help.openai.com/en/articles/4936830

```

```

# Para gerar uma API Key:
# 1. Crie uma conta na OpenAI
# 2. Acesse a seção "API Keys"
# 3. Clique em "Create API Key"
# Link direto: https://platform.openai.com/account/api-keys

```

```

# Substitua o texto TODO por sua API Key da OpenAI, ela será salva como uma variável de ambiente.
openai_api_key = 'sk-HULfQYgoD9DCH5N4CU6MT3B1bkFJ7XJ4AQJMGyVmB9SFRPlb'

```

```
import random
```

```

df = pd.read_csv('Frases.csv')
frases = df['Frases'].tolist()
print(frases)

```

```

for user in users:
    numFrase = random.randint(0, 2)
    news = frases[numFrase]

```

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

```
['Mariana, para o seu dinheiro crescer', 'Bruno, invista na sua conta bancaria digital! Faça transações com PIX sem taxas.', 'Henry
```

▼ Load

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

```
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}!")  
  
☞ User Mariana updated? True!  
   User Bruno updated? True!  
   User Henry updated? True!
```