



Envio de SMS com Telesign API e a Linguagem Python

Isaac Felipe dos Santos Lima
ifdsl.lic20@uea.edu.br

Tópicos :


1. O que é API
2. Criando as Credencias da Telesign API
3. Integração com o Python Para envio de SMS

O que é uma API



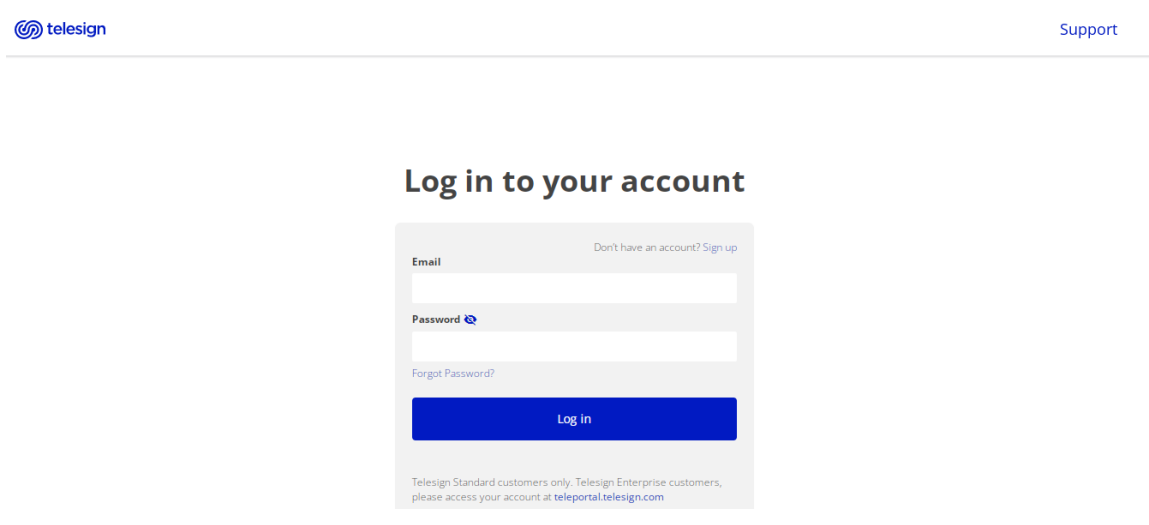
- As API (**Application Programming Interface**) são interfaces de comunicações entre dois serviços ou entre camadas de um mesmo serviços. É através dessa interface de comunicação que a gente consegue consumir funcionalidades de outros serviços, ou seja, é responsável por integrar sistemas.
- Sendo assim, Utilizaremos Telesign API para envio de SMS utilizando junto com a Linguagem Python.

Obtendo as credenciais da Telesign API

- 
- Para construir o programa para envio de SMS precisamos ter acesso a Telesign API para obtermos as credencias para utilizar em qualquer linguagem de programação. Pode-se obter no seguinte link :
 - <https://portal.telesign.com/login>
 - Caso você não estiver uma com sua conta no site, você precisará se cadastrar para poder obter acesso a API

Obtendo as credenciais da Telesign API

- Na imagem a baixo mostra a tela de login do site Telesign. Caso não tenha cadastro, deverá fazê-lo para poder ter as credenciais, após fazer o cadastro ele te redireciona passo a passo para criar suas credenciais de jeito tão simples pelo e-mail.



The screenshot displays the Telesign login interface. At the top left is the Telesign logo, and at the top right is a 'Support' link. The main heading is 'Log in to your account'. Below this, there is a login form with two input fields: 'Email' and 'Password'. The 'Password' field includes an eye icon for toggling visibility. Above the 'Email' field is a link that says 'Don't have an account? Sign up'. Below the 'Password' field is a link that says 'Forgot Password?'. A prominent blue 'Log in' button is positioned below the form fields. At the bottom of the form, a note states: 'Telesign Standard customers only. Telesign Enterprise customers, please access your account at teleportal.telesign.com'.

Obtendo as credenciais da Telesign API

- Na imagem a baixo mostra a tela inicial do site Telesign. Logo na tela Inicial ele mostra opção de escolher a Linguagem de sua preferencia para utilizar a API. Vamos escolher a Linguagem Python

The screenshot displays the Telesign API developer portal. On the left is a dark sidebar with the Telesign logo, a 'Trial Version' badge, an 'Upgrade' button, and the account balance '\$0.00'. Below this are navigation links: HOME, SETTINGS, BILLING, REPORTS, PRODUCTS, and LOGOUT. The main content area has a header with a message about the future portal and a link to 'My Telesign'. The 'Account' section shows the 'Customer ID' and 'API Keys' status, with a message indicating that credentials are being created. Below this, the 'Quick Starts' section includes a 'Send an SMS Message' button and a link to 'Add Two-Factor Authentication (SMS)'. The 'Send an SMS Message' guide is expanded, showing instructions to 'Install the SDK' (download the SDK) and 'Create Your Code' (create a Node-compatible file). A code snippet is provided:

```
from future import print function
```

. On the right, there are links for 'Documentation', 'API References', and 'Download SDK'. At the bottom right, there is a video thumbnail titled 'How to Send an SMS with Node.js' and a link to the 'Developer Center'. The footer contains copyright information for 2023 Telesign, links to 'Terms of Service' and 'Privacy Notice', and the 'Developer Center' link.

telesign

Check out your future portal! [My Telesign](#) is where you'll find product pages, API Tools and some financial tools to manage your account(s).

Trial Version
[Upgrade](#)
Account balance \$0.00

HOME
SETTINGS
BILLING
REPORTS
PRODUCTS
LOGOUT

Account

Customer ID

API Keys

We are creating your credentials. Please check again soon to see your keys.

[Manage API Key](#)

Quick Starts

[Send an SMS Message](#)

[Add Two-Factor Authentication \(SMS\)](#)

Send an SMS Message

Python

Install the SDK
Download the SDK

```
pip install telesign
```

[Copy Code](#)

Create Your Code

Create a Node-compatible file (e.g., "index.js"), add code displayed below and save it.

```
from future import print function
```

[Documentation](#)
[API References](#)
[Download SDK](#)

[How to Send an SMS with Node.js](#)

Developer Center

© 2023 Telesign / [Terms of Service](#) / [Privacy Notice](#)

Obtendo as credenciais da Telesign API

- Clicar em Manage API key para criar sua credencial

The screenshot displays the Telesign Developer Portal interface. On the left is a dark sidebar with the Telesign logo and navigation links: HOME, SETTINGS, BILLING, REPORTS, PRODUCTS, and LOGOUT. The main content area has a header with a message about the future portal and a section titled 'Account'. Under 'Account', there's a 'Customer ID' field and a section for 'API Keys' with a message: 'We are creating your credentials. Please check again soon to see your keys.' A red arrow points to the 'Manage API Key' link in this section. Below the API Keys section, there are 'Quick Starts' including 'Send an SMS Message' (highlighted), 'Add Two-Factor Authentication (SMS)', and 'Install the SDK'. The 'Send an SMS Message' section shows a code snippet: `pip install telesign` with a 'Copy Code' button. The 'Install the SDK' section shows a code snippet: `from future import print_function`. On the right side of the main content area, there are links for 'Documentation', 'API References', and 'Download SDK'. At the bottom, there's a 'Developer Center' link.

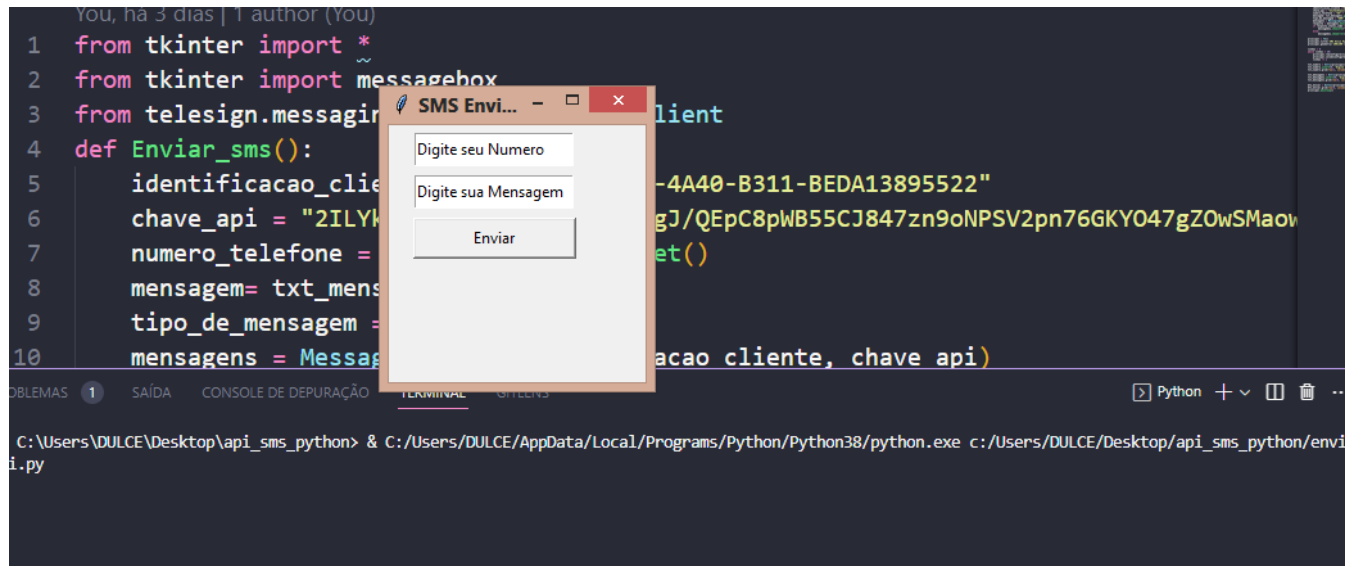
Obtendo as credenciais da Telesign API

- Após ter acesso a credencial vamos implementar o código

```
You, há 3 dias | 1 author (You)
1 from tkinter import *
2 from tkinter import messagebox
3 from telesign.messaging import MessagingClient
4 def Enviar_sms():
5     identificacao_cliente = "DA9CE9DF-13D6-4A40-B311-BEDA13895522"
6     chave_api = "2ILYkJE3aovlKzIzDiVwck6CgJ/QEpC8pWB55CJ847zn9oNPSV2pn76GKY047gZOwSMaov
7     numero_telefone = "55" + txt_numero.get()
8     mensagem= txt_mensagens.get()
9     tipo_de_mensagem = "ARN"
10    mensagens = MessagingClient(identificacao_cliente, chave_api)
11    resposta = mensagens.message(numero_telefone, mensagem, tipo_de_mensagem)
12    if resposta.status_code == 200:
13        messagebox.showinfo("SMS Enviado Com Sucesso")
14    else:
15        messagebox.showerror("Falha ao enviar SMS")
16
17 principal = Tk()
18 principal.title('SMS Envio Python')
19 principal.geometry('200x200')
20
```

Obtendo as credenciais da Telesign API

- Explicando brevemente ao código implementado:
- No código existe uma interface que é do próprio python que é o Tkinter, que é nativa da própria linguagem. Neste campo existe para digitar o número como também campo para digitar a msg



The screenshot displays a Python IDE with a dark theme. In the background, a Python script is visible, showing imports for Tkinter and Telesign, and a function `Enviar_sms()` that takes phone number and message as input. Overlaid on the script is a Tkinter window titled "SMS Envi...". This window contains two text input fields: the first is labeled "Digite seu Numero" and the second is labeled "Digite sua Mensagem". Below these fields is an "Enviar" button. The IDE's interface includes tabs at the top for "PROBLEMAS", "SAÍDA", "CONSOLE DE DEPUÇÃO", "TERMINAL", and "GIT/REPOS". The terminal at the bottom shows the command prompt for running the script: `C:\Users\DULCE\Desktop\api_sms_python> & C:/Users/DULCE/AppData/Local/Programs/Python/Python38/python.exe c:/Users/DULCE/Desktop/api_sms_python/envi...`

Obtendo as credenciais da Telesign API

- Existe um função que tem a missão de enviar a mensagem, pegando também as credenciais do cliente, no caso a credenciais da API, o número para quem for enviar a mensagem e a mensagem, através da função get

```
4 def Enviar_sms():
5     identificacao_cliente = "DA9CE9DF-13D6-4A40-B311-BEDA13895522"
6     chave_api = "2ILYkJE3aovlKzIzDiVwck6CgJ/QEpC8pWB55CJ847zn9oNPSV2pn76GKY047gZOwSMaov
7     numero_telefone = "55" + txt_numero.get()
8     mensagem= txt_mensagens.get()
9     tipo_de_mensagem = "ARN"
10    mensagens = MessagingClient(identificacao_cliente, chave_api)
11    resposta = mensagens.message(numero_telefone, mensagem, tipo_de_mensagem)
12    if resposta.status_code == 200:
13        messagebox.showinfo("SMS Enviado Com Sucesso")
14    else:
15        messagebox.showerror("Falha ao enviar SMS")
16
```

Obtendo as credenciais da Telesign API

- Outra configuração muito importante é as configurações da parte Interface da Aplicação

```
17 principal = Tk()
18 principal.title('SMS Envio Python')
19 principal.geometry('200x200')
20
21 linhas = 0
22 while linhas < 20:
23     principal.rowconfigure(linhas, weight=1)
24     principal.columnconfigure(linhas, weight=1)
25     linhas += 1
26
27 txt_numero = Entry(principal)
28 txt_numero.insert(0, 'Digite seu Numero')
29 txt_numero.grid(row=1, column=5, sticky='NS')
30
31 txt_mensagens = Entry(principal)
32 txt_mensagens.insert(0, 'Digite sua Mensagem')
33 txt_mensagens.grid(row=3, column=5, sticky='NS')
34
35 btn_enviar = Button(principal, text='Enviar', command=Enviar_sms)
36 btn_enviar.grid(row=5, column=5, sticky='NESW')
37 principal.mainloop()
```

Obtendo as credenciais da Telesign API

- 
- Coloquei brevemente as informações e as ações que meu programa faz com relação ao Desafio colocado em questão de envio de SMS via API. Nesse Desafio pude aprender bastante coisa.