

Deploying API's

MoodBoost API

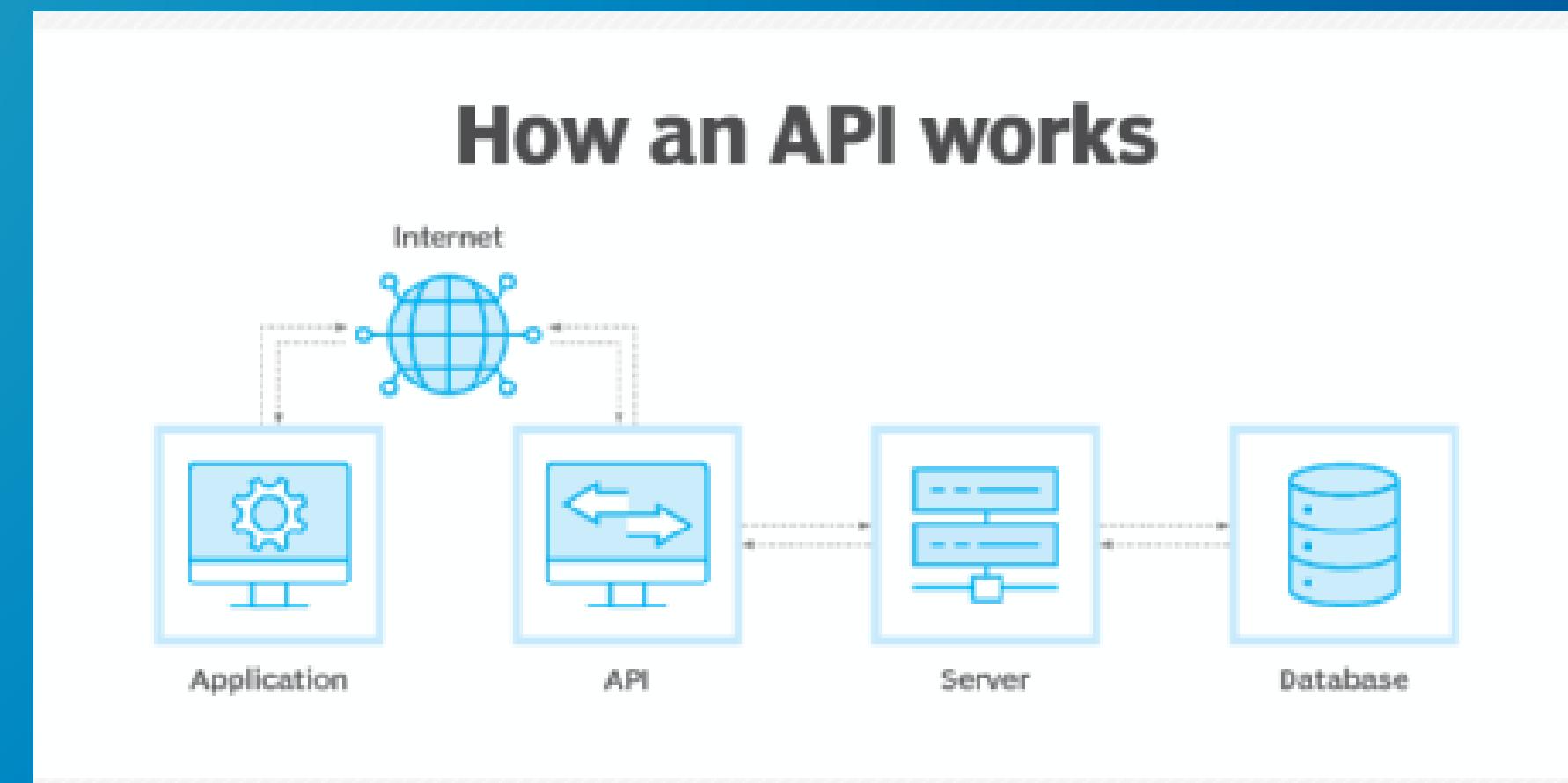


Anderson Aragon and Rocio Henriquez

What is an API?

An API (Application Programming Interface) is a set of rules and protocols that allows different software applications to communicate with each other.

- It allows one app to access functions or data from another without knowing how it works internally.
- It improves scalability and simplifies maintenance.



Summary Table

API	Common Uses	Apps That Use It	
Google Maps	Maps, routes, geolocation	Uber, Airbnb, Waze	
Twitter	Read/write tweets, user data	Hootsuite, TweetDeck	
Stripe / PayPal	Online payments	Shopify, Booking.com	
Spotify	Music and playlists	Discord, music apps	
OpenWeather	Real-time weather	Weather apps, phone widgets	
Firebase	Mobile backend (auth, data, notifications)	Mobile apps, games	

Name and logo of our API



What it does:

It's an API that analyzes a text (for example, a chat message or a social media post) and returns automatic suggestions to improve the user's mood.



Analyzes the sentiment of the submitted text (positive, negative, neutral).



Returns personalized motivational phrases.

Technologies



PYTHON

```
...nth[i], space); token_5 = strtok(token, space); strun...  
prature; strcpy(uv, token_5); index = 0; temp = temprature + index * humi...  
c of humidity, and the day is %f\n\n", day, month, temperature, comparis...  
prature_comparis...  
prature_comparison[i] < lowest){lowest = temperature_c...  
sword_only(){char arr[100]; *profile, *password, *user_n...  
, "r");fseek(ptr, 0, SEEK_CUR);fgets(arr, sizeof(arr), ...  
r_name);if (length > 5){strcpy(*user_name);}else{password...  
ng_forecast_data){FILE *ptr_1;char check[SIZE];char n...  
'};int temp, index, hum, wind, diff, checking, leaving
```

Python



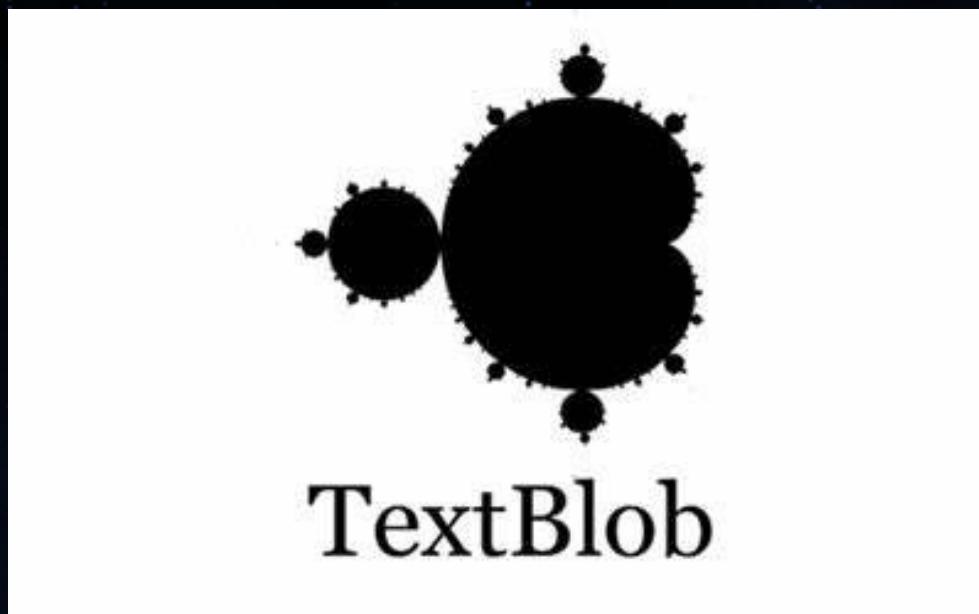
Flask

Flask for the API



ToolJet as a low-code interface to connect and test it.

Technologies



TextBlob

Textblob: Detects the state of being by polarities



Cloudflare: Used for publishing the API to the cloud



Postman: Used for testing.

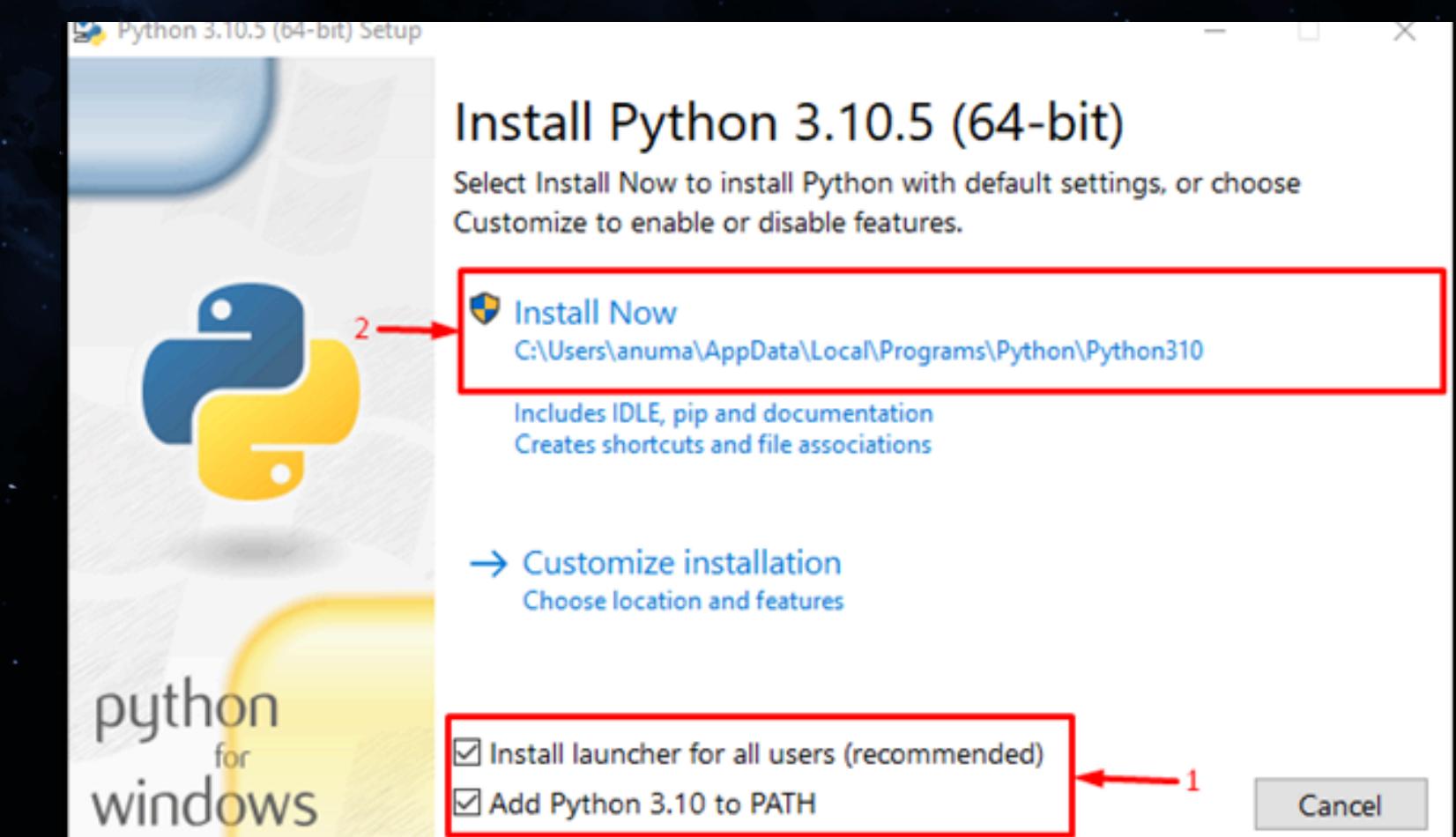
Steps to create it

1

Install Python and Flask

To perform this, install python from the website and install flask from pip

Remember: When installing python click on add to PATH



Steps to create it

2

Create the API

Remember to name the file so as to be easy to identify, make it as describable as possible

```
Tabnine | Edit | Test | Explain | Document
51 @app.route('/analizar', methods=['POST'])
52 def analizar():
53     data = request.get_json(silent=True)
54
55     if not isinstance(data, dict):
56         return jsonify({'sentimiento': 'Error: cuerpo inválido. Esperaba JSON'}), 400
57
58     texto = data.get('texto', '')
59     if not texto.strip():
60         return jsonify({'sentimiento': 'Error: el campo "texto" está vacío o no proporcionado'}), 400
61
62     sentimiento = detectar_sentimiento(texto)
63     frase = random.choice(frases_motivadoras.get(sentimiento, ["Sigue adelante, lo estás haciendo bien."]))
64
65     # Mantiene "sentimiento" para ToolJet y agrega "frase_motivadora"
66     return jsonify({
67         'sentimiento': sentimiento,
68         'frase_motivadora': frase
69     })
70
71 if __name__ == '__main__':
72     app.run(debug=True, port=5000)
```

Piece of code as representation

Steps to create it

3

Local test

You will have a terminal in python after creating the API, to test it you can use Postman as I personally used.

```
PS C:\Users\MINEDUCYT\Documents\API> & C:/Users/MINEDUCYT/AppData/Local/Programs/Python/Python313/python.exe c:/Users/MINEDUCYT/Documents/API/app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
```

Steps to create it

3

Local test

The screenshot shows the Postman application interface. On the left, the sidebar includes 'Collections' (with 'My first collection'), 'Environments', 'Flows', and 'History'. A central search bar at the top says 'Search Postman' with keyboard shortcuts 'Ctrl K'. The main workspace shows a 'Getting started' section with a 'POST https://depends-trusts-scottish-stupid.trycloudflare.com/analizar' request. The 'Body' tab is selected, showing a raw JSON payload:

```
1 {  
2   "texto": "I feel really happy"  
3 }  
4
```

The 'Headers' tab shows 8 items. Below the request, the response is displayed with a status of '200 OK', a duration of '1.74 s', and a size of '469 B'. The response body is:

```
1 {  
2   "frase_motivadora": ";Que gusto que te sientas tan bien! Recuerda: \"La alegría es la forma más  
3   sencilla de gratitud.\" - Karl Barth",  
4   "sentimiento": "muy alegre"
```

At the bottom, there are buttons for 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.

Steps to create it

4

Deploy to Cloudflare

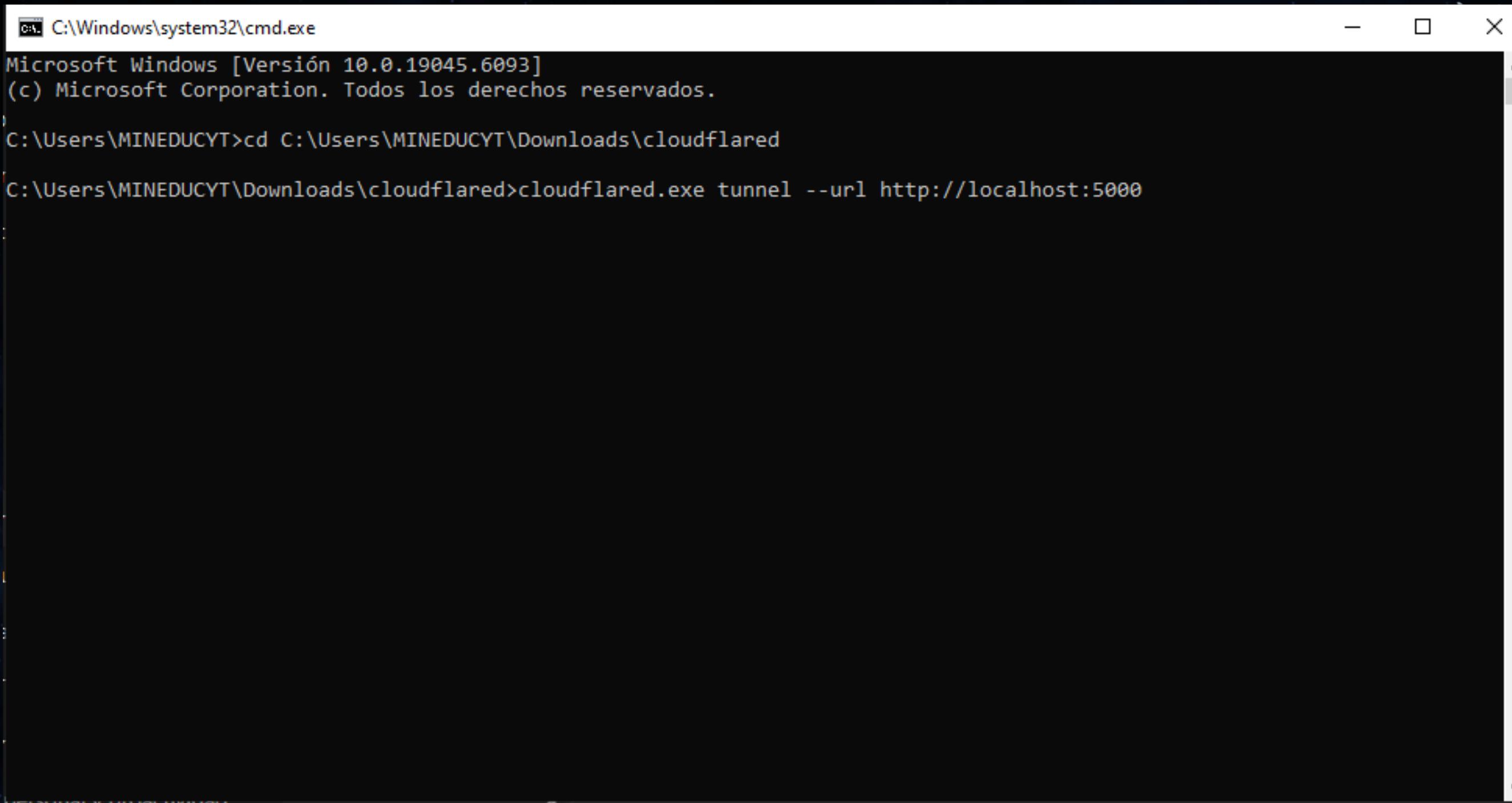
We have to install Cloudflare and place it in the right environment for connecting with python API.

After that, we have to deploy with cmd, as in the example of the following page.

Steps to create it

4

Deploy to cloudflared



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.6093]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\MINEDUCYT>cd C:\Users\MINEDUCYT\Downloads\cloudfared

C:\Users\MINEDUCYT\Downloads\cloudfared>cloudfared.exe tunnel --url http://localhost:5000
```

Steps to create it

4

Deploy to cloudflared

```
C:\Windows\system32\cmd.exe - cloudflared.exe tunnel --url http://localhost:5000
to experiment and try it out. However, be aware that these account-less Tunnels have no uptime guarantee, are subject to
the Cloudflare Online Services Terms of Use (https://www.cloudflare.com/website-terms/), and Cloudflare reserves the ri
ght to investigate your use of Tunnels for violations of such terms. If you intend to use Tunnels in production you shou
ld use a pre-created named tunnel by following: https://developers.cloudflare.com/cloudflare-one/connections/connect-app
s
2025-07-29T01:40:53Z INF Requesting new quick Tunnel on trycloudflare.com...
2025-07-29T01:40:58Z INF +-----+
2025-07-29T01:40:58Z INF | Your quick Tunnel has been created! Visit it at (it may take some time to be reachable):
2025-07-29T01:40:58Z INF | https://retro-deployment-genes-agree.trycloudflare.com
2025-07-29T01:40:58Z INF +-----+
2025-07-29T01:40:58Z INF Cannot determine default configuration path. No file [config.yml config.yaml] in [~/.cloudflare
d ~/.cloudflare-warp ~/cloudflare-warp]
2025-07-29T01:40:58Z INF Version 2025.7.0 (Checksum 9e37ce90110dc9e83108317273369641d2cab438cd9ca0458b30a6568fb09eac)
2025-07-29T01:40:58Z INF GOOS: windows, GOVersion: go1.24.2, GoArch: amd64
2025-07-29T01:40:58Z INF Settings: map[ha-connections:1 protocol:quic url:http://localhost:5000]
2025-07-29T01:40:58Z INF cloudflared will not automatically update on Windows systems.
2025-07-29T01:40:58Z INF Generated Connector ID: 0012ae75-05f4-4f23-8a4f-42c6f6f2d697
2025-07-29T01:40:58Z INF Initial protocol quic
2025-07-29T01:40:58Z INF ICMP proxy will use 192.168.1.9 as source for IPv4
2025-07-29T01:40:58Z INF ICMP proxy will use fe80::6165:2c3e:f0ba:8563 in zone Wi-Fi as source for IPv6
2025-07-29T01:40:59Z INF cloudflared does not support loading the system root certificate pool on Windows. Please use --
origin-ca-pool <PATH> to specify the path to the certificate pool
2025-07-29T01:40:59Z INF ICMP proxy will use 192.168.1.9 as source for IPv4
2025-07-29T01:40:59Z INF Tunnel connection curve preferences: [X25519MLKEM768 CurveP256] connIndex=0 event=0 ip=198.41.1
92.57
2025-07-29T01:40:59Z INF ICMP proxy will use fe80::6165:2c3e:f0ba:8563 in zone Wi-Fi as source for IPv6
2025-07-29T01:40:59Z INF Starting metrics server on 127.0.0.1:20241/metrics
2025-07-29T01:40:59Z INF Registered tunnel connection connIndex=0 connection=ec11ce0e-d1b4-4b04-ba37-3fb3cf6a6fda event=
0 ip=198.41.192.57 location=mia01 protocol=quic
```

Steps to create it

5

Test the public URL

Use Postman once again to make sure your API is working correctly after being published.

Steps to create it

6

Connect to ToolJet

The screenshot shows the ToolJet application interface. At the top, there's a navigation bar with tabs for 'MoodBoost Frontend' and 'Changes saved'. Below the navigation, the main area displays a user interface for mood analysis. On the left, there's an input field labeled 'inputTexto' containing the text 'I feel quite incredible'. Next to it is a button labeled 'Analizar'. To the right of the button, the text 'Muy Alegre' is displayed in large blue letters. Below this, a message reads: '¡tu energía es contagiosa! sigue brillando.' In the bottom left corner of the main area, there's a small icon with a red '62' badge. The bottom half of the screen is a configuration panel for a component named 'analizarSentimiento'. It has tabs for 'Setup', 'Transformation', and 'Settings', with 'Setup' currently selected. Under 'Setup', there are sections for 'Parameters' (with a '+ Add' button), 'Request' (Method: POST, URL: <https://depends-trusts-scottish-stupid.trycloudflare.com/analizar>), 'Headers', 'Params', 'Body' (which is active and shows a JSON object starting with '{'), and 'Cookies'. There's also a 'Raw' toggle switch. At the bottom of this panel, there's a 'Preview' section. On the far right of the interface, there's a sidebar titled 'Components' with sections for 'Commonly Used' (Table, Button, Form, Text Input, Date Picker, Text), 'Layouts' (Modal, Container, Tabs), and 'Forms' (List View). The entire interface has a clean, modern design with a dark background.

Steps to create it

6

Connect to ToolJet

The screenshot shows the ToolJet interface with a project named "analizarSentimiento". The "Setup" tab is selected, showing the "Request" configuration. The method is set to "POST" and the URL is <https://depends-trusts-scottish-stupid.trycloudflare.com/analizar>. Under the "Headers" tab, there is one entry: "Content-Type" set to "application/json". There is also a "Params" tab, a "Body" tab, and a "Cookies" tab. On the right side of the interface, there are "Run" and "Preview" buttons.

Create a REST API and name it as in image,
Method POST and paste URL provided by
cloudflared

Steps to create it

6

Connect to ToolJet

The screenshot shows the ToolJet interface with a component named "analizarSentimiento".

Component Preview:

- Input:** An input field labeled "inputTexto" containing the text "I feel quite incredible".
- Action:** A button labeled "Analizar".
- Output:** The result "Muy Alegre" is displayed.
- Message:** "¡tu energía es contagiosa! sigue brillando."

Component Configuration (Setup Tab):

- Request:** Method: POST, URL: <https://depends-trusts-scottish-stupid.trycloudflare.com/analizar>
- Body:** Raw mode, Body content:

```
1 {
2   "texto": "{{components.inputTexto.value}}"
3 }
```

Properties Panel:

- Data:** Label: "inputTexto", Placeholder: "inputTexto", Default value: "neutral".
- Events:** No event handlers.
- Validation:** Read documentation for TextInput →

Steps to create it

6

Connect to ToolJet

The screenshot illustrates the ToolJet interface for creating a sentiment analysis application. On the left, a card displays an input field labeled "inputTexto" containing the text "I feel quite incredible". Below it is a button labeled "Analizar" with the ID "BTNALANIZAR". To the right, the output text "Muy Alegre" is displayed, along with the message "¡tu energía es contagiosa brillando.". At the bottom, a "carSentimiento" card shows setup details: Method "POST", URL "https://depends-trusts", and Body tab selected.

Properties Panel (Right):

- Label:** Analizar
- Events:** On click (Run Query)
- Action Options:** New event handler
- Additional Actions:** Loading state (disabled), Visibility (enabled), Disable (disabled)

Raw Mode: Enabled

Steps to create it

6

Connect to ToolJet

The screenshot shows the MoodBoost Frontend interface. At the top, there's a navigation bar with 'MoodBoost Frontend', 'Changes saved', 'Env Development', 'ver v1', and a 'Promote' button. Below the navigation, there's a UI component where a user has typed 'I feel quite incredible'. This input is processed by a block labeled 'Analizar' (Analyze) which outputs the result 'Muy Alegre' (Very Happy). A tooltip for this result block contains the message: '¡tu energía es contagiosa! sigue brillando.' (Your energy is contagious! Keep shining.)

On the right side of the screen, the 'analizarSentimiento' query is being configured. It is set to a POST method with the URL <https://depends-trusts-scottish-stupid.trycloudflare.com/analizar>. The 'Preview' section shows the response in JSON format:

```
frase_motivadora: "¡Perfecto! Recuerda: \"La felicidad se encuentra en las pequeñas cosas.\""  
sentimiento: "alegre"
```

The 'Setup' tab is selected in the configuration panel. Other tabs include 'Transformation' and 'Settings'. There are also 'Run' and 'Preview' buttons. On the far right, there are sections for 'Data', 'Events', and 'Additional Actions'.

conclusion

THANKS