

# Uso de API con Power Automate

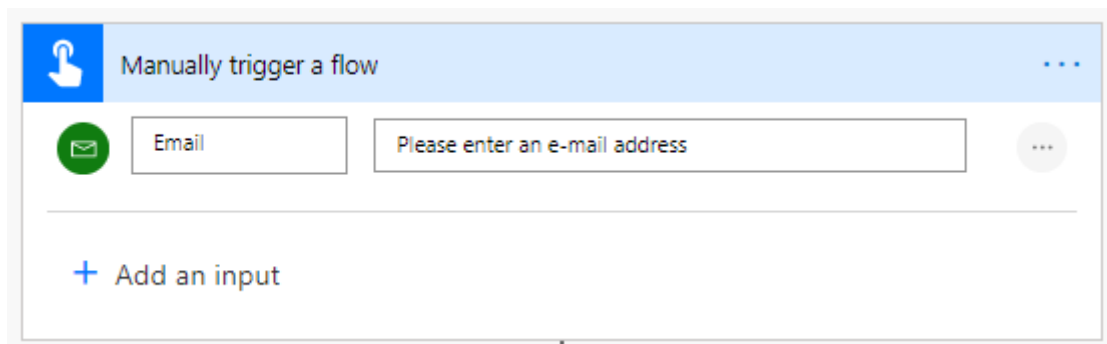
Internet está lleno de API públicas para jugar, y algunas de ellas no requieren ningún tipo de autenticación, por lo que es fácil y rápido jugar con ellas.

Entonces, construyamos un flujo simple para entender cómo usar una API en Power Automate.

Busqué en línea una API adecuada y encontré esta: [Official Joke API](#).

## Desencadenar

Cada flujo comienza con un disparador. Esta vez optaré por un disparador manual con un campo de entrada de correo electrónico



## Acción HTTP

Y ahora necesitamos una broma. La acción para esto en Power Automate es **HTTP**. La API oficial de Joke no requiere autenticación, por lo que solo podemos configurar el método y el URI.

La URL base de la API de Joke es <https://official-joke-api.appspot.com/>. No hay una ruta base en este caso particular.

Hay varios puntos finales disponibles para la API de Joke, pero en este ejercicio he decidido usar `/random_joke`.

La documentación de la API no dice qué método usar, pero en este caso podemos asumir con seguridad que el método es GET ya que solo queremos obtener una nueva broma.

The image shows a web-based HTTP client interface. At the top, there's a green header bar with a globe icon and the title "HTTP Get the joke". Below this, the interface is organized into sections for configuring an HTTP request:



- \* Method:** A dropdown menu currently set to "GET".
- \* URI:** A text input field containing the URL "https://official-joke-api.appspot.com/random\_joke".
- Headers:** A table with two columns, "Enter key" and "Enter value", and a trash icon to the right.
- Queries:** A table with two columns, "Enter key" and "Enter value", and a trash icon to the right.
- Body:** A text input field with the placeholder "Enter request content".
- Cookie:** A text input field with the placeholder "Enter HTTP cookie".

At the bottom left, there is a link that says "Show advanced options" followed by a downward arrow icon.

Método: OBTENER

TIPO: [https://official-joke-api.appspot.com/random\\_joke](https://official-joke-api.appspot.com/random_joke)

Ahora guarde el flujo y haga una prueba. Si todo funciona, debería tener una pequeña broma en el cuerpo de su acción HTTP en formato JSON. Aquí hay un ejemplo:

 HTTP 1s

INPUTS

Method

GET

URI

[https://official-joke-api.appspot.com/random\\_joke](https://official-joke-api.appspot.com/random_joke)

OUTPUTS

Status code

200

Headers

Key	Value
Vary	Accept-Encoding
Access-Control-Allow-Origin	*
X-Cloud-Trace-Context	921bad7b221649f40c7eb5ed...

Body

```
{
  "id": 162,
  "type": "general",
  "setup": "What did Michael Jackson name his denim store?",
  "punchline": "Billy Jeans!"
}
```

```
{
  "id": 162,
  "type": "general",
  "setup": "What did Michael Jackson name his denim store?",
  "punchline": "Billy Jeans!"
}
```

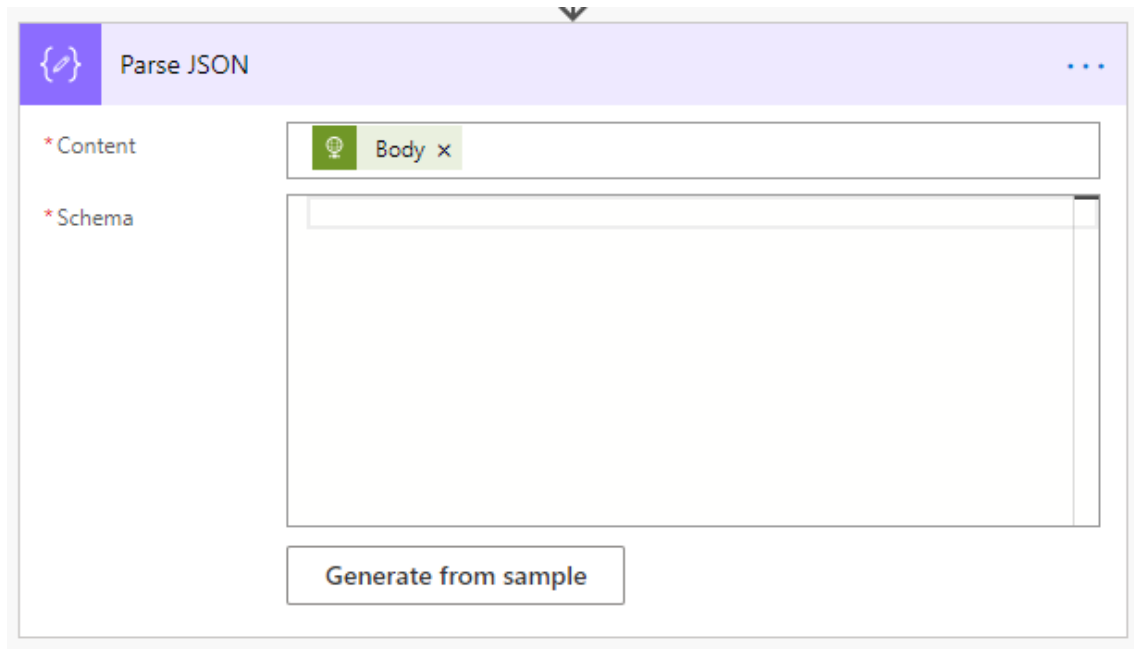
Asegúrese de copiar el contenido del texto del campo del cuerpo. ¡Lo necesitamos en el siguiente paso!

## Analizar JSON

Recibimos el chiste en formato JSON ordenado, pero para separar la configuración del remate, necesitamos la **Parse JSON** acción . Analizar JSON... analiza el JSON para que podamos usar los valores más adelante como contenido dinámico.

Entonces, como contenido, configuramos **el cuerpo de la acción HTTP anterior** . Y luego necesitamos un esquema para definir la estructura y el contenido de un objeto JSON.

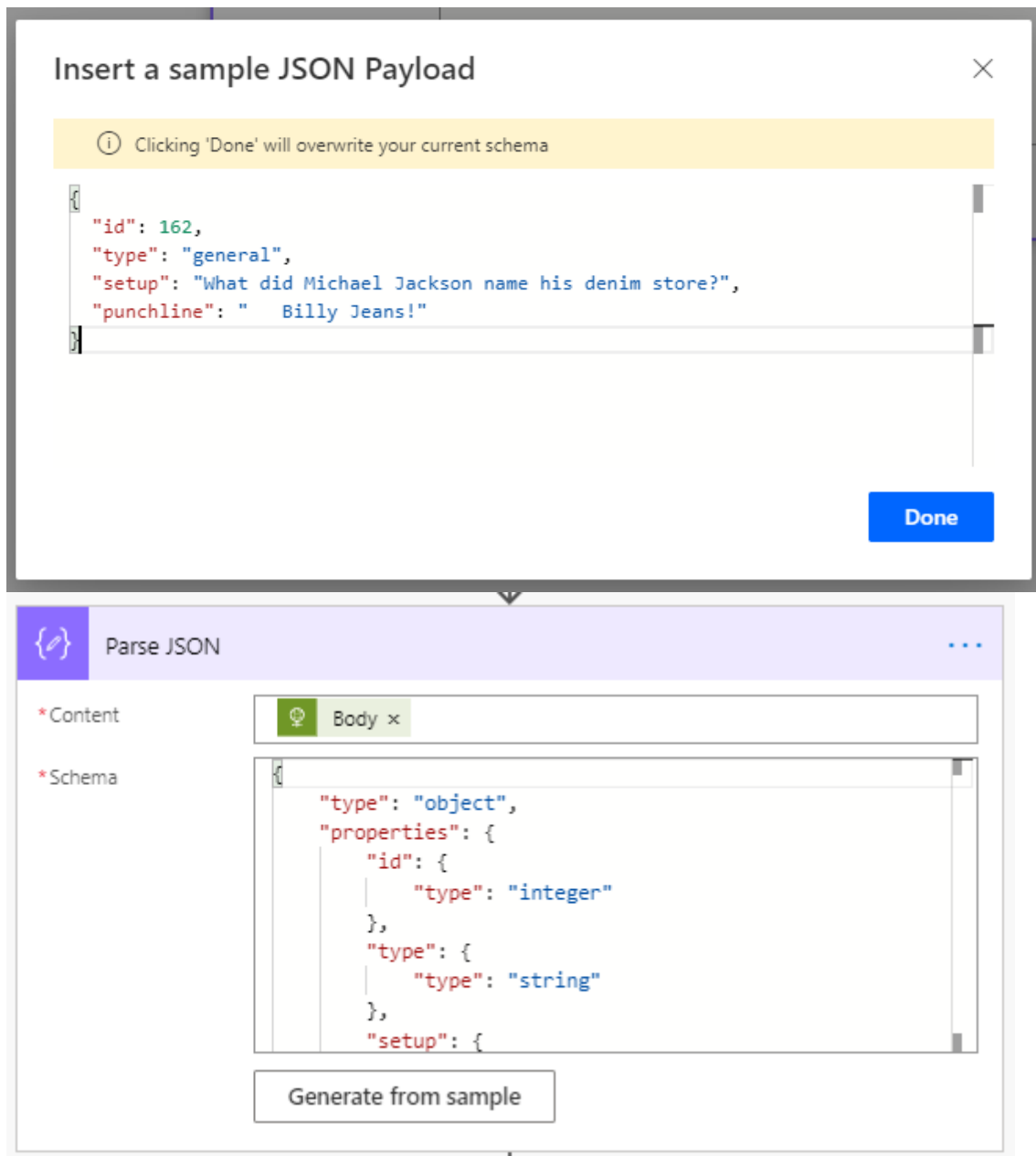
Pero ¿de dónde viene este esquema?



Si ha seguido mis instrucciones con cuidado, debe copiar el contenido JSON. Si no, adelante, hazlo ahora.

Elija **generar el esquema a partir de una muestra** (porque tenemos una muestra; el JSON que copiamos del campo del cuerpo de la solicitud HTTP).

Pegue el JSON del cuerpo de la solicitud HTTP en el editor de muestras y haga clic en "Listo". Generará el esquema automáticamente.



JSON Schema es un lenguaje gramatical para definir la estructura, el contenido, etc.

Por ejemplo aquí podemos ver que ID es un número entero (entero) y el chiste es texto (cadena).

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {
        "type": "integer"
      },
      "type": {
        "type": "string"
      },
      "setup": {
```

```

        "type": "string"
    },
    "punchline": {
        "type": "string"
    }
},
"required": [
    "id",
    "type",
    "setup",
    "punchline"
]
}
}

```

Ahora tenemos el chiste y lo hemos analizado con la acción Parse JSON. Ahora podemos enviar esta información de la forma que queramos, yo voy a utilizar notificaciones

