

CLASE 5

HTTP Request: Consumir APIs

GET, POST, autenticación y manejo de respuestas JSON

■ Duración: 40 min | ■ Nivel: Intermedio | ■ 27/05/2026

■ OBJETIVO DE LA CLASE

Conectar n8n con cualquier API externa usando el nodo HTTP Request, manejar autenticación y procesar las respuestas.

El nodo HTTP Request

HTTP Request es el nodo más poderoso y universal de n8n. Si no existe un nodo específico para tu servicio, HTTP Request puede conectarse a cualquier API REST.

Soporta todos los métodos HTTP: GET, POST, PUT, PATCH, DELETE, HEAD. También soporta múltiples tipos de autenticación: API Key, Bearer Token, Basic Auth, OAuth2.

Ejemplo 1: API pública (sin auth)

1 Crear workflow: Clima actual

Vamos a obtener el clima de Buenos Aires usando Open-Meteo (sin API key)

2 Nodo HTTP Request

- Method: GET
- URL: <https://api.open-meteo.com/v1/forecast>

3 Agregar Query Parameters

Clic en 'Add Query Parameter':

- latitude: -34.61
- longitude: -58.38
- current_weather: true

4 Ejecutar y ver respuesta

El resultado será un JSON con temperatura, velocidad de viento, etc. Accedé con {{
\$json.current_weather.temperature }}

Ejemplo 2: API con API Key

1 Configurar Header con API Key

En HTTP Request:

- Authentication: Header Auth
- Name: X-API-Key (o el nombre que pida la API)
- Value: tu-api-key-aquí

2 Alternativa: Query Parameter

Algunas APIs quieren la key en la URL:

- URL: https://api.ejemplo.com/data?api_key=TU_KEY

3 Ejemplo con NewsAPI

- URL: <https://newsapi.org/v2/top-headlines>
 - Query params: country=ar, apiKey=TU_KEY
 - Trae las noticias principales de Argentina
-

Ejemplo 3: POST con cuerpo JSON

1 Método POST

- Method: POST
- URL: <https://api.ejemplo.com/usuarios>
- Body Content Type: JSON

2 Definir el body

En 'Body Parameters' → Switch a JSON Body:

```
{
  "nombre": "{{ $json.nombre }}",
  "email": "{{ $json.email }}",
  "plan": "premium"
}
```

3 Capturar la respuesta

La respuesta estará disponible como {{ \$json.id }}, {{ \$json.status }}, etc. en el siguiente nodo.

Manejo de errores HTTP

Error Handling

Activá 'Continue On Fail' para que el workflow no se rompa si la API falla. Los errores quedan en {{ \$json.error }}.

Retry on Fail

n8n puede reintentar automáticamente. En el nodo: Settings → Retry On Fail → hasta 5 veces.

Timeout

Configurá un timeout máximo para no esperar indefinidamente. Settings → Timeout: 10000 ms.

Código de respuesta

Verificá con IF: si {{ \$json.statusCode }} != 200, tomá acción alternativa.

■ TIP: Instalá la extensión 'Requestly' en Chrome para interceptar y modificar requests. Te ayuda a entender qué envían y reciben las APIs antes de replicarlo en n8n.

TAREA:

Crear un workflow que consulte la API de tipos de cambio (exchangerate-api.com), obtenga el valor del dólar y lo guarde en una Google Sheet.

➡■ *Próxima clase: Clase 6: Loops y procesamiento masivo - Cómo procesar listas grandes de datos eficientemente.*