O que é uma API, sem complicação

Pense numa API como uma **porta de serviço** de um site. Em vez de abrir a página, ler e copiar a informação, o seu programa (ou a sua IA) bate nessa porta com um endereço e recebe o **dado pronto**: a temperatura agora, a cotação do dólar, o preço do bitcoin, o endereço de um CEP.

Você pede (um endereço/URL)→A API responde→Dado pronto em JSON

A resposta costuma vir em **JSON**: um texto organizado em pares `"nome": valor`, fácil para pessoas lerem e para programas usarem. Você vai ver um de verdade daqui a pouco.

- **Tempo:** cerca de 10 minutos até o primeiro resultado.
- **Nível:** iniciante. Se você sabe abrir um link, consegue fazer o primeiro teste.
- **Custo:** as APIs deste guia foram testadas sem chave e sem cadastro. Cada API tem suas regras de uso e limites; elas estão explicadas na seção de cuidados.

“Para de pagar API” não quer dizer que toda API é grátis para sempre e para qualquer uso. Quer dizer: antes de assinar um serviço, procure se já existe uma API aberta que resolve — e confira as regras dela.

O que é o public-apis (números de hoje)

O public-apis é um repositório no GitHub mantido pela comunidade com uma **lista de APIs gratuitas** organizada por tema. Não é uma API: é um catálogo para você descobrir qual porta bater.

2.190 APIs listadas nas categorias

51 categorias

1.051 marcadas sem chave (Auth = No)

491 mil+ estrelas no GitHub

- **Sem chave e com HTTPS:** 988 entradas. **Sem chave, HTTPS e CORS liberado:** 525 — as mais fáceis para começar (as colunas são explicadas no próximo passo).
- **Maiores categorias:** Development (220), Government (126), Geocoding (113), Games & Comics (111), Finance e Transportation (97 cada), Cryptocurrency (96).
- **GitHub:** 491.791 estrelas, 54.726 forks, licença MIT.

Como contamos: linhas das tabelas das 51 categorias do README (o topo tem uma área patrocinada e uma lista de servidores MCP, que ficaram de fora). Algumas APIs aparecem em mais de uma categoria — são 2.170 nomes diferentes. Conferido em **24/10/2026, às 10h36** (horário de Brasília), versão `82e763e` do README e API do GitHub. A lista muda quase todo dia.

Como achar uma API na lista

1. Abra o índice do public-apis e escolha uma **categoria** (Weather = clima, Currency Exchange = câmbio, Cryptocurrency = cripto, Government = governo, Geocoding = endereços e mapas…).
2. Em cada linha da tabela, leia as colunas:

Coluna	O que quer dizer	Para começar, prefira
API	Nome com link para a documentação oficial.	—
Description	O que ela entrega, em inglês.	—
Auth	Se precisa de identificação. `No` = sem chave; `apiKey` = cadastro e chave; `OAuth` = login autorizado (mais trabalhoso).	`No`
HTTPS	Se a conexão é criptografada. Sites atuais bloqueiam chamadas sem HTTPS.	`Yes`
CORS	Se a API aceita ser chamada direto de uma página no navegador. `Unknown` = ninguém conferiu; teste.	`Yes`

Exemplo real da categoria Weather:

```
| Open-Meteo | Global weather forecast API for non-commercial use | No | Yes | Yes |
```

Tradução: previsão do tempo, **sem chave**, com HTTPS, funciona no navegador — e o próprio nome já avisa: **uso não comercial**.

1. **Clique no nome** e abra a documentação. É lá que estão o endereço (endpoint), os parâmetros, os limites e os termos. A lista é um ponto de partida; a documentação é a regra.

Seu primeiro resultado em 10 minutos: o clima de Curitiba

A Open-Meteo está na lista (categorias Weather e Environment) e não pede chave. Teste de dois jeitos — escolha um:

Jeito 1 — no navegador: copie o endereço abaixo, cole na barra do navegador e aperte Enter. Aparece um texto em JSON. Pronto: você usou uma API.

Cole na barra do navegador:

```
https://api.open-meteo.com/v1/forecast?
latitude=-25.43&longitude=-49.27&current=temperature_2m,relative_humidity_2m,wind_speed_10m&timezone=America/
Sao_Paulo
```

Execute no terminal:

```
curl "https://api.open-meteo.com/v1/forecast?
latitude=-25.43&longitude=-49.27&current=temperature_2m,relative_humidity_2m,wind_speed_10m&timezone=America/
Sao_Paulo"
```

Trecho da resposta real (24/10/2026):

```
"current": {
  "time": "2026-10-24T10:30",
  "temperature_2m": 19.3,
  "relative_humidity_2m": 86,
  "wind_speed_10m": 12.0
}
```

Como ler: às 10h30 de 24/10/2026 fazia **19,3 °C** em Curitiba, com 86% de umidade e vento de 12 km/h. Quando você rodar, os números serão os de agora.

Para outra cidade, troque `latitude` e `longitude` (pesquise “latitude e longitude de [cidade]”). No Windows, use `curl.exe` no lugar de `curl` se o PowerShell reclamar.

Mais dois testes: dólar e bitcoin

Duas APIs da lista que responderam sem chave no nosso teste. Rode no terminal ou abra o endereço no navegador.

Dólar em reais — Frankfurter (categoria Currency Exchange; Auth = No, CORS = Yes). Cotações de bancos centrais e fontes oficiais, atualizadas diariamente — não é a cotação de cada segundo.

Execute no terminal:

```
curl "https://api.frankfurter.dev/v2/rate/usd/brl"
```

Trecho da resposta real (24/10/2026):

```
{"date":"2026-10-24","base":"USD","quote":"BRL","rate":5.0388}
```

Bitcoin em reais — CoinGecko (categoria Cryptocurrency; na lista, Auth = No).

Execute no terminal:

```
curl "https://api.coingecko.com/api/v3/simple/price?ids=bitcoin&vs_currencies=brl"
```

Trecho da resposta real (24/10/2026):

```
{"bitcoin":{"brl":401476}}
```

Respostas reais de 24/10/2026, entre 10h36 e 10h41 (horário de Brasília). **Atenção a um detalhe que acontece muito:** a CoinGecko respondeu sem chave, mas a documentação dela hoje orienta criar uma **chave gratuita (Demo)** e tem limites menores para quem não usa. Ou seja: a lista diz uma coisa, a documentação atualizada pode dizer outra. Vale sempre a documentação.

Outro exemplo: a Frankfurter já está na versão `v2`. Tutoriais antigos com `/v1/` ainda funcionam hoje, mas use o formato da documentação atual.

Use com IA: 5 pedidos prontos para copiar

Você não precisa programar a integração. Descreva o que quer e deixe o agente ler a documentação e escrever o código.

- **Claude Code ou Codex** (no seu computador, dentro da pasta do projeto): o agente cria e edita os arquivos e testa as chamadas. É o melhor lugar para os pedidos 1, 2 e 5.
- **Claude ou ChatGPT no navegador:** funcionam bem para os pedidos 3 e 4 e para gerar o painel do pedido 2 — você salva o arquivo que ele entregar e abre no navegador.

Troque o que está entre colchetes.

1. Painel com clima, dólar e bitcoin

```
Monte um painel simples em um único arquivo painel.html, que abra direto no navegador sem instalar nada.
Mostre três cartões:
1) Clima agora em Curitiba, pela Open-Meteo: https://api.open-meteo.com/v1/forecast?
latitude=-25.43&longitude=-49.27&current=temperature_2m,relative_humidity_2m,wind_speed_10m&timezone=America/
Sao_Paulo
2) Dólar em reais, pela Frankfurter: https://api.frankfurter.dev/v2/rate/usd/brl
3) Bitcoin em reais, pela CoinGecko: https://api.coingecko.com/api/v3/simple/price?
ids=bitcoin&vs_currencies=brl
Formate os valores em português do Brasil, mostre a hora da última atualização, atualize a cada 5 minutos e
mostre uma mensagem amigável se alguma API falhar. No rodapé, cite as fontes dos dados. Depois me diga como
abrir o arquivo.
```

2. Integrar uma API no meu projeto

```
Quero integrar a API [NOME DA API] (documentação: [LINK]) no meu projeto para mostrar [O QUE EU QUERO
MOSTRAR]. Antes de escrever código, leia a documentação e me diga: se precisa de chave, quais são os limites
de uso e se o uso comercial é permitido. Depois crie a integração seguindo o padrão do projeto, trate os
erros (sem internet, 401, 403, 429, resposta vazia) e me mostre como testar. Se precisar de chave, guarde em
variável de ambiente (.env) e nunca no código que vai para o navegador.
```

3. Achar a API certa para a minha ideia

```
Tenho esta ideia: [DESCREVA SUA IDEIA]. Procure na lista https://github.com/public-apis/public-apis APIs que
sirvam para isso. Prefira Auth = No, HTTPS = Yes e CORS = Yes. Para as 3 melhores, abra a documentação
oficial e confirme se ainda funciona, se hoje precisa de chave, quais são os limites e se permite uso
comercial. Entregue uma tabela comparando as três e um comando curl para eu testar cada uma no terminal.
```

4. Entender a resposta da API

```
Vou colar abaixo a resposta JSON de uma API. Explique em português simples o que significa cada campo e qual
unidade ele usa, e diga quais campos eu usaria para [MEU OBJETIVO]. Depois transforme os dados em uma tabela.

[COLE O JSON AQUI]
```

5. Revisar segurança e limites

```
Revise a integração com a API [NOME] neste projeto. Verifique: se alguma chave aparece no código que vai para
o navegador ou no histórico do Git; se existe tratamento para 401, 403, 429 e falta de internet; se
respeitamos o limite de chamadas (sugira cache se precisar); e se os termos da API permitem o meu uso. Liste
os problemas por gravidade e só corrija depois de me mostrar o plano.
```

Opcional: tenha a lista no seu computador

Não é obrigatório — dá para usar tudo pelo site do GitHub. Na própria página, use **Ctrl+F** (Windows) ou **Cmd+F** (Mac) e busque uma palavra em inglês: *weather*, *currency*, *bitcoin*, *brazil*, *CEP*.

- **Baixar ZIP, sem terminal:** no repositório, botão verde **Code** → **Download ZIP** (ou este link direto). Descompacte e abra o `README.md` em qualquer editor.
- **Clonar com Git** (se você já tem o Git instalado): use os comandos abaixo. O `--depth 1` baixa só a versão atual, mais rápido.

Execute no terminal:

```
git clone --depth 1 https://github.com/public-apis/public-apis.git
cd public-apis
```

Execute no terminal:

```
grep -i "weather" README.md | grep "| No |"
```

O segundo comando (Mac/Linux) mostra só as APIs de clima **sem chave**. Troque *weather* pelo tema que quiser. Quer ver quantas estão sem chave, com HTTPS e CORS liberado? `grep -c "| No | Yes | Yes |" README.md` — hoje deu 525. Para atualizar a lista depois: `git pull` dentro da pasta.

Atalho com IA: abra o Claude Code dentro da pasta `public-apis` e use o pedido 3 da seção anterior. O agente lê o README local e já te devolve as opções.

Cuidados antes de usar de verdade

- **Cotas e limites.** API gratuita tem limite. Exemplo: o plano aberto da Open-Meteo hoje aceita menos de 10.000 chamadas por dia, 5.000 por hora e 600 por minuto — e avisa que hospedagens como Vercel e Netlify podem bater o limite antes, porque muitos sites dividem o mesmo endereço. Guarde o resultado por alguns minutos (cache) em vez de chamar a cada clique.
- **Licença e uso comercial.** A licença MIT do public-apis vale para a *lista*, não para as APIs. Cada API tem seus termos. A Open-Meteo gratuita é só para uso não comercial (site com assinatura ou anúncio já conta como comercial) e pede crédito aos dados (licença CC BY 4.0). Para uso comercial, ela tem planos pagos — veja os valores no site oficial.
- **Nunca exponha chave no front-end.** Tudo que roda no navegador pode ser lido por qualquer visitante (é só abrir o DevTools). Se a API pede chave, a chamada deve sair de um servidor ou função serverless, com a chave em variável de ambiente (`.env`, que deve estar no `.gitignore`). Vazou? Apague a chave no painel da API e gere outra.
- **A lista pode estar desatualizada.** “Auth = No” hoje pode virar “precisa de chave” amanhã, e APIs saem do ar. Confira a documentação e tenha uma alternativa da mesma categoria.
- **Dados de pessoas.** Algumas APIs devolvem dados pessoais (CPF, nome, endereço). Usar esses dados exige base legal na LGPD. Na dúvida, não use.

Problemas comuns e como resolver

- **Funciona no navegador ou no terminal, mas a sua página dá erro “blocked by CORS policy”:** a API não aceita chamadas vindas de outro site. Escolha uma com CORS = Yes ou faça a chamada pelo seu servidor (uma função serverless resolve). Peça ao agente: “mova esta chamada para o servidor”.
- **401 Unauthorized:** falta a chave ou ela está errada/expirada. Confira o nome exato do cabeçalho ou parâmetro na documentação.
- **403 Forbidden:** a chave existe, mas não tem permissão para aquele recurso ou plano — ou a API bloqueou o acesso. Algumas exigem um cabeçalho `User-Agent` (a lista marca isso na coluna Auth).
- **429 Too Many Requests:** você passou do limite. Espere (veja se a resposta traz `Retry-After`), chame com menos frequência e use cache.
- **404 ou site fora do ar:** o endereço mudou ou a API foi desativada. Volte à documentação ou troque por outra da mesma categoria.
- **A resposta veio diferente do tutorial:** a API mudou de versão (como a Frankfurter `v1` → `v2`). Use o formato da documentação atual.
- **No Windows, o `curl` se comporta estranho:** use `curl.exe` no PowerShell.

Fontes e próximos passos

Números e testes conferidos em **24/10/2026**. Listas e APIs mudam: se algo estiver diferente, vale a documentação oficial.

- Repositório public-apis · README com a lista · como contribuir · dados do repositório na API do GitHub
- Open-Meteo: documentação · termos e limites
- Frankfurter: documentação
- CoinGecko: documentação
- MDN: o que é CORS · MDN: códigos de status HTTP

Seu próximo passo: rode o teste do clima agora e depois cole o pedido 1 no seu agente. Em 10 minutos você tem um painel com dados reais.

Curtiu? Siga @benites.ia no Instagram e comente nos vídeos o que você montou com uma API — ou qual tutorial quer ver depois.

Seguir @benites.ia