# API betekent : Application programming interface

Een **application programming interface** (**API**) is een verzameling definities op basis waarvan een [computerprogramma](https://nl.wikipedia.org/wiki/Computerprogramma) kan communiceren met een ander programma of onderdeel (meestal in de vorm van [bibliotheken](https://nl.wikipedia.org/wiki/Bibliotheek_(informatica))). Vaak vormen API's de scheiding tussen verschillende lagen van abstractie, zodat applicaties op een hoog niveau van abstractie kunnen werken en het minder abstracte werk uitbesteden aan andere programma's. Hierdoor hoeft bijvoorbeeld een tekenprogramma niet te weten hoe het de [printer](https://nl.wikipedia.org/wiki/Printer) moet aansturen, maar roept het daarvoor een gespecialiseerd stuk [software](https://nl.wikipedia.org/wiki/Software) aan in een bibliotheek, via een afdruk-API.

## **Publieke website**

Het is steeds gebruikelijker dat ook [websites](https://nl.wikipedia.org/wiki/Website) over een publieke open API beschikken. Deze kan dan benaderbaar zijn via methodes als [REST](https://nl.wikipedia.org/wiki/REST), [SOAP](https://nl.wikipedia.org/wiki/SOAP_(protocol)), [GraphQL](https://nl.wikipedia.org/wiki/GraphQL" \o "GraphQL), [RSS](https://nl.wikipedia.org/wiki/Really_Simple_Syndication) of [XML](https://nl.wikipedia.org/wiki/Extensible_Markup_Language)/[RPC](https://nl.wikipedia.org/wiki/Remote_procedure_call). Het formaat dat wordt teruggestuurd kan dan bijvoorbeeld XML of [JSON](https://nl.wikipedia.org/wiki/JSON) zijn. Door een dergelijke API beschikbaar te stellen is het mogelijk om een [mashup](https://nl.wikipedia.org/wiki/Mashup_(internetapplicatie)" \o "Mashup (internetapplicatie)) te maken, zoals een combinatie van de kaarten van [Google Maps](https://nl.wikipedia.org/wiki/Google_Maps) met de foto's van [Panoramio](https://nl.wikipedia.org/wiki/Panoramio" \o "Panoramio). Dergelijke toepassingen zijn typisch voor de zogenaamde [Web 2.0](https://nl.wikipedia.org/wiki/Web_2.0)-gedachte. [Open data](https://nl.wikipedia.org/wiki/Open_data) worden doorgaans via een API ter beschikking gesteld.

Wanneer een website geen API aanbiedt, kan men nog trachten de data te verzamelen via [scrapen](https://nl.wikipedia.org/wiki/Scrapen).

**Representational state transfer (REST)** is een [software](https://nl.wikipedia.org/wiki/Software)-architectuur voor gedistribueerde mediasystemen zoals het [wereldwijde web](https://nl.wikipedia.org/wiki/Wereldwijde_web).

REST wordt veelal gebruikt voor het ontwerpen van [application programming interfaces](https://nl.wikipedia.org/wiki/Application_programming_interface" \o "Application programming interface) (API) voor het internet

In de TransIP REST API worden de volgende requests gebruikt:

* **GET:**Verzoekt om een 'resource' van een server, waarbij een resource een speciale variabele is (de resource representation) die een referentie tot een externe resource bevat. De data kan vervolgens worden aangepast en teruggestuurt met een PUT-request.
* **POST:**Een client voegt een nieuwe resource toe, bijvoorbeeld de bestelling van een product.
* **PUT:**Update een resource met nieuwe data, bestaande data wordt verwijderd. Dit gebruik je bijvoorbeeld als je met een GET request de specificaties van een VPS opvraagt, de description aanpast en het gehele array terug plaatst.
* **PATCH:**Wordt niet standaard gebruikt in API's, maar is wel aanwezig in onze REST API. PATCH wordt gebruikt voor een gedeeltelijke update van een resource, bijvoorbeeld het stoppen/starten van een VPS.
* **DELETE:**Gebruik je voor het verwijderen van resources.

De API gebruikt meestal JSON objecten en arrays voor alle requests en responses die via de API uitgevoerd worden.

Voorbeelden van een API :

1/ mopjes opvragen via [www.icanhazdadjoke.com](http://www.icanhazdadjoke.com)

Check of de site een API heeft!

JA

<https://icanhazdadjoke.com/api>

Wat is de enpoint? <https://icanhazdadjoke.com/>

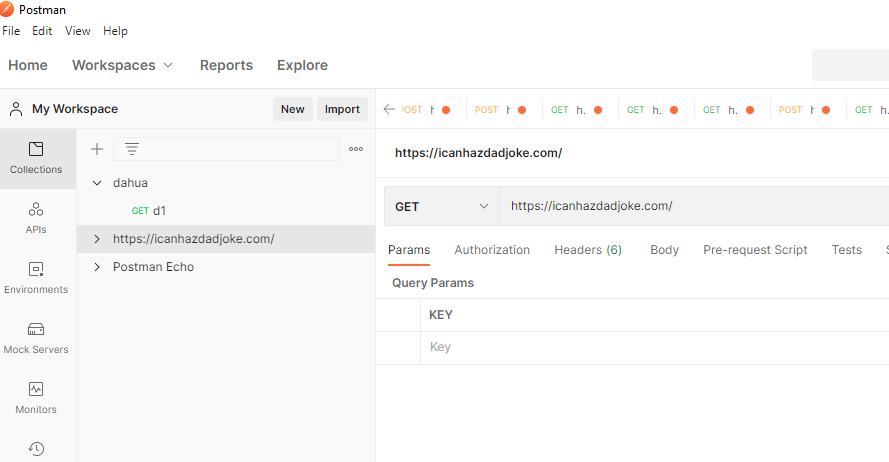
**Zijn er extra parameters en of headers? Optioneel of verplicht?**

**Testen met browser (JSON addon) of Postman ( download Postman)**

**Add JSON Viewer to chrome:**

<https://chrome.google.com/webstore/detail/json-viewer/gbmdgpbipfallnflgajpaliibnhdgobh>

**Je merkt nu dat een header “**Accept: application/json” **noodzakelijk is!**

****

Soms is een persoonlijke key, al of niet betalend, nodig!

Voorbeeld2 > weer-api op <https://openweathermap.org>

registreer en kopieer je key >> <https://home.openweathermap.org/users/sign_in>

Leese de API docs op >> <https://openweathermap.org/current>

Voorbeeld :

https://api.openweathermap.org/data/2.5/weather?q=lokeren,be&units=metric&APPID=cb7ec0.......

Test met Postman

Met TCP client low level code, mogelijk maar niet aangewezen..

Met Python requests package! >> <https://docs.python-requests.org/en/latest/>

GET requests:

payload = **{**'key1'**:** 'value1'**,** 'key2'**:** 'value2'**}**

r = requests.get**(**'https://httpbin.org/get'**,** params=payload**)**

**Hoe kan het op een andere manier zoals met forms?**

headers :

url = 'https://api.github.com/some/endpoint'

headers = **{**'user-agent'**:** 'my-app/0.0.1'**}**

r = requests.get**(**url**,** headers=headers**)**

POST requests:

payload = **{**'key'**:** 'value'**})**

r = requests.post**(**'https://httpbin.org/post'**,** data= payload)

# HTTP Status Codes

<https://httpstatuses.com/>

## 2×× Success

* [**200** OK](https://httpstatuses.com/200)

## 4×× Client Error

* [**400** Bad Request](https://httpstatuses.com/400)
* [**401** Unauthorized](https://httpstatuses.com/401)

## 5×× Server Error

* [**500** Internal Server Error](https://httpstatuses.com/500)

Oefeningen :

1/ Maak een Quiz met Trivia >> <https://opentdb.com/api_config.php>

Met Tkinter Trivia of Nederlandstalig alternatief stel je kwisvragen en hou je de score bij.

Er zijn ja/nee vragen, die zijn misschien het makkelijkst.

2/ Toon tijd, sunset, sunrise, buitentemperatuur op een OLED verbonden met je ESP32.

3/ Kan je een mopje laten voorlezen op 1 of andere manier?