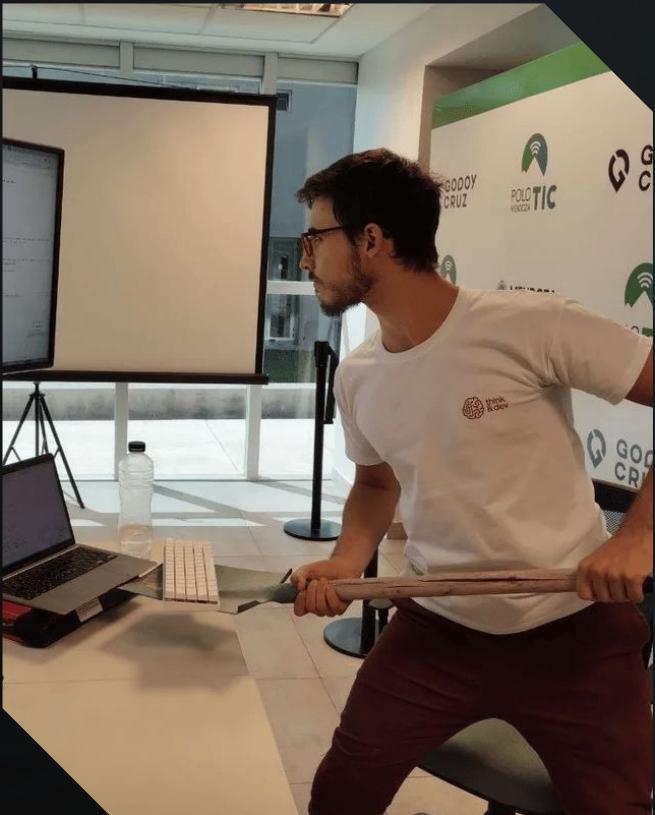


Construyendo Servidores de HTTP



Francesco Gentile (eyon.dev)

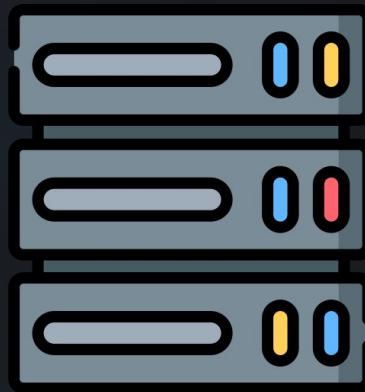
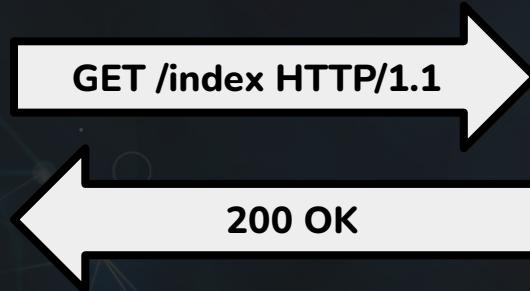
Desarrollador Fullstack
CTO en Eluter

¿Que es el HTTP?

HyperText

Transfer

Protocol



```
<h1>
    Qué querés tomar?
</h1>

<a href="/coffee">
    Café
</a>

<a href="/tea">
    Té
</a>
```

¿Que vamos a construir hoy?

2 Aplicaciones de Servidor



1. Servidor de Archivos

AKA: Sitio Web



Eyon

About Blog Contact

Hi! I'm Francesco Gentile

I use **Python** to solve real problems

Let's solve something together [Contact](#)



Name	Status	Type
eyon.dev	200	document
code.css	200	stylesheet
styles.css	200	stylesheet
css2?family=Open+Sans:ital,wght@0,3...	200	stylesheet
highlight.min.js	200	script
MainGH_low.jpeg	200	jpeg
DSC_2365_Low.JPG	200	jpeg
instagram.svg	302	text/plain ...
twitter.svg	302	text/plain ...
youtube.svg	302	text/plain ...
email-decode.min.js	200	script
twitch.svg	302	text/plain ...
github.svg	302	text/plain ...
linkedin.svg	302	text/plain ...
memvYaGs126MiZpBA-UvWbX2vVnXB...	200	font
twitch.svg	200	svg+xml
twitter.svg	200	svg+xml
github.svg	200	svg+xml
youtube.svg	200	svg+xml
instagram.svg	200	svg+xml
linkedin.svg	200	svg+xml
favicon.png	200	png
favicon.png	200	png

Request URL:	https://eyon.dev/
Request Method:	GET
Status Code:	200 OK
Remote Address:	172.67.140.205:443
Referrer Policy:	strict-origin-when-cross-origin

```
X Headers Preview Response Initiator Timing
1  |<!DOCTYPE html>
2  |<html>
3  |<head>
4  |<meta charset="utf-8">
5  |<meta name="viewport" content="width=device-width" , initial-scale="1.0">
6  |<title>
7  |    Eyon | Home
8  |</title>
9
10 |<link rel="stylesheet" href="/assets/css/code.css">
11 |<link rel="stylesheet" href="/assets/css/styles.css">
12 |<link rel="shortcut icon" type="image/png" href="/favicon.png">
13 |<link href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;1,400" href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;1,400" type="text/css">
14 |<meta name="google-site-verification" content="zG4KwhctHPDPuoQjjKFaP_KE7X_Te0eTE" href="https://www.google.com/search?rlz=1C1GCEU_enUS804US804&q=site%3Aeyon.dev+site+verification" type="text/html">
15 |<script src="//cdnjs.cloudflare.com/ajax/libs/highlight.js/11.5.0/highlight.min.js" type="text/javascript">
16
17 |<title>Home | Eyon</title>
18 |<meta name="generator" content="Jekyll v4.1.1" />
19 |<meta property="og:title" content="Home" />
20 |<meta name="author" content="Francesco Gentile" />
21 |<meta property="og:locale" content="en_US" />
22 |<meta name="description" content="Francesco Gentile's portfolio and blog about software development and technology." />
23 |<meta property="og:description" content="Francesco Gentile's portfolio and blog about software development and technology." />
24 |<link rel="canonical" href="https://eyon42.github.io/" />
25 |<meta property="og:url" content="https://eyon42.github.io/" />
26 |<meta property="og:site_name" content="Eyon" />
27 |<script type="application/ld+json">
28 |  {"name": "Francesco Gentile", "url": "https://eyon42.github.io/", "headline": "Home", "url": "https://eyon42.github.io/"}
29
30 |</head>
31 |<body>
32 |<div class="selection:text-white0 selection:bg-primaryRed">
33 |<header class="bg-white0 select-none md: absolute top-0 w-full z-50">
34 |<script>
35 |    function toggleMobileMenu() {
36 |        var x = document.getElementById("mobile-menu");
37 |        if (x.style.display === "block") {
38 |            x.style.display = "none";
39 |        } else {
```

2. Receptor de Datos



WIKIPEDIA
The Free Encyclopedia

Search

Create account

Username (help me choose)
Eyon.dev
Your username is public and cannot be made private later.

Password

It is recommended to use a unique password that you are not using on any other website.

Confirm password

Email address (recommended)
francescogentyle@gmail.com
Please look for an email from us to verify your address.

Email is required to recover your account if you lose your password.

CAPTCHA Security check

Captcha: percymoat
CRefresh
percymoat
Can't see the image? Request an account

Create your account

▼ General

Request URL: https://en.wikipedia.org/w/index.php?title=Special:CreateAccount&returnto=HTTP

Request Method: **POST**

Status Code: 302 Found

Remote Address: 208.80.154.224:443

Referrer Policy: origin-when-cross-origin

Headers Payload Preview Response Initiator Timing Cookies

▼ Query String Parameters view source view URL-encoded

title: Special:CreateAccount
returnto: HTTP

▼ Form Data view source view URL-encoded

title: Special:CreateAccount
wpName: Eyon.dev
wpPassword: REDACTED
retype: REDACTED
email: francescogentyle@gmail.com
captchaWord: REDACTED
wpCreateaccount: Create your account
wpEditToken: +\ REDACTED
authAction: create
force: REDACTED
wpCreateaccountToken: REDACTED
captchaId: REDACTED
campaign: REDACTED
geEnabled: -1
forceMentor: REDACTED

HTTP Mercado Pago / Crear Preferencia de Pago

POST {{BASE_URL}}/checkout/preferences

Params Authorization Headers (8) Body **JSON** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

```
1  {
2    "items": [
3      {
4        "title": "Dummy Title",
5        "description": "Dummy description",
6        "picture_url": "http://www.myapp.com/myimage.jpg",
7        "category_id": "car_electronics",
8        "quantity": 1,
9        "currency_id": "ARS",
10       "unit_price": 10
11     }
12   ],
13   "expires": true,
14   "date_of_expiration": "2023-06-03T00:00:00.000-03:00",
15   "external_reference": "W3M_VENDIMIA2023_GENERAL_test@test.com",
16   "payer": {
17     "email": "francescogentyle@gmail.com"
18   },
19   "back_urls": {
20     "failure": "https://web3makers.org/pagoW3MVendimia2023/failure",
21     "pending": "https://web3makers.org/pagoW3MVendimia2023/pending",
22     "success": "https://web3makers.org/pagoW3MVendimia2023/success"
23   },
24   "additional_info": "https://web3makers.org"
25 }
26
```

Webhooks

Recibe notificaciones en tiempo real a través de un HTTP POST cada vez que ocurra un evento relacionado con las transacciones de esta aplicación.

[Acceder a la documentación](#)

Configurar notificaciones

Definí los eventos y la dirección a dónde se transmitirán.

Simular

[Modo de prueba](#)

[Modo producción](#)

URL para prueba

`https://test.eyon.dev/webhooks/mp`

Eventos

Pagos ×

Planes y suscripciones ×

▼

Clave secreta ?

G

[Guardar](#)

[Restablecer](#)

**¿Cual es la forma más rápida
de tener una solución?**

No Code

Sin escribir código de aplicación

Servir Archivos Estáticos

Name	Status	Type
eyon.dev	200	document
code.css	200	stylesheet
styles.css	200	stylesheet
css2?family=Open+Sans:ital,wght@0,3...	200	stylesheet
highlight.min.js	200	script
MainGH_low.jpeg	200	jpeg
DSC_2365_Low.JPG	200	jpeg
instagram.svg	302	text/plain ...
twitter.svg	302	text/plain ...
youtube.svg	302	text/plain ...
email-decode.min.js	200	script
twitch.svg	302	text/plain ...
github.svg	302	text/plain ...
linkedin.svg	302	text/plain ...
memvYaGs126MiZpBA-UvWbX2vVnXB...	200	font
twitch.svg	200	svg+xml
twitter.svg	200	svg+xml
github.svg	200	svg+xml
youtube.svg	200	svg+xml
instagram.svg	200	svg+xml
linkedin.svg	200	svg+xml
favicon.png	200	png
favicon.png	200	png

Websites for you and your projects.

Hosted directly from your [GitHub repository](#). Just edit, push, and your changes are live.



Ready to get started? Build your own site from scratch or generate one for your project.

You get one site per GitHub account and organization,

Create a new repository

A repository contains all project files, including the revision history.

Owner Repository name 

Great repository names are short and memorable. Need inspiration? How about [fictional-winner](#)?

Description (optional)

 Public
Anyone can see this repository. You choose who can commit.

 Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.



2 Clone the repository

Go to the folder where you want to store your project, and clone the new repository:

```
- $ git clone https://github.com/username/username.github.io
```

3 Hello World

Enter the project folder and add an index.html file:

```
- $ cd username.github.io
- $ echo "Hello World" > index.html
```

4 Push it

Add, commit, and push your changes:

```
- $ git add --all
- $ git commit -m "Initial commit"
- $ git push -u origin main
```

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <http://eyon.dev>

Last deployed by  [Eyon42](#) last year

[Visit site](#) [...](#)

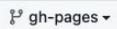
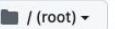
Build and deployment

Source

Deploy from a branch [▼](#)

Branch

Your GitHub Pages site is currently being built from the `gh-pages` branch. [Learn more about configuring the publishing source for your site.](#)

 `gh-pages`  `/ (root)` [Save](#)

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the `github-pages` environment by the [pages build and deployment](#) workflow.

[Learn more about deploying to GitHub Pages using custom workflows](#)

Custom domain

Custom domain

Custom domains allow you to serve your site from a domain other than `eyon42.github.io`. [Learn more about configuring custom domains.](#)

[Save](#) [Remove](#)

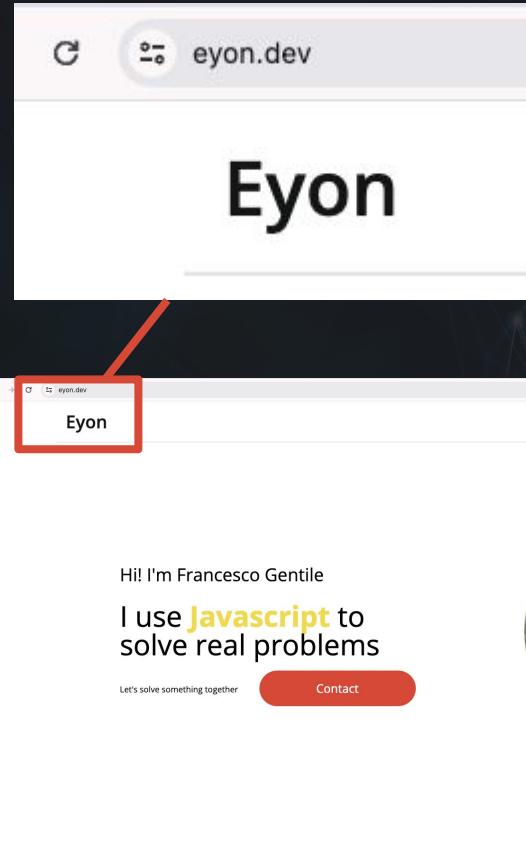
 DNS Check in Progress

[Enforce HTTPS](#) — Unavailable for your site because your domain is not properly configured to support HTTPS (`eyon.dev`) —

[Troubleshooting custom domains](#)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site.

When HTTPS is enforced, your site will only be served over HTTPS. [Learn more about securing your GitHub Pages site with HTTPS.](#)



Recibir Peticiones POST

Inspect webhooks and HTTP requests

Get a URL to collect HTTP or webhook requests and inspect them in a human-friendly way.

Optionally connect APIs, run code and return a custom response on each request.

Create Request Bin

Hosted on pipedream.com and private by default.

Create a [public bin](#) instead.

OR SUBSCRIBE TO EVENTS FROM A WEBHOOK OR REST API FOR ANY APP

Search by app...

1 - 24 of 2,000+ apps by most popular



HTTP / Webhook

Get a unique URL where you can send HTTP or webhook requests



Node

Anything you can do with Node.js, you can do in a Pipedream workflow. This includes using...



Python

Anything you can do in Python can be done in a Pipedream Workflow. This includes using...



OpenAI (ChatGPT)

OpenAI is an AI research and deployment company with the mission to ensure that...



Salesforce (REST API)

Web services API for interacting with Salesforce



HubSpot

HubSpot's CRM platform contains the marketing, sales, service, operations, and...



Zoho CRM

Zoho CRM is an online Sales CRM software that manages your sales, marketing, and...



Stripe

Stripe powers online and in-person payment processing and financial solutions for...



Shopify Developer App

Shopify is a user-friendly e-commerce platform that helps small businesses build a...

Your endpoint is

https://enohwbcag2dde.x.pipedream.net

copy

Generate Test Events

Send requests to this endpoint to inspect [webhooks](#), [custom events](#) and more:

- Events appear instantly
- Endpoints don't expire
- Easily delete a single event, or all

CURL

JAVASCRIPT

NODE

PYTHON 2

PYTHON 3

PHP

GO

RUBY

```
curl -d '{ "name": "Luke Skywalker" }' \
-H "Content-Type: application/json" \
https://enohwbcag2dde.x.pipedream.net/
```

HTTP <https://enohwbcag2dde.x.pipedream.net>

POST <https://enohwbcag2dde.x.pipedream.net>

Params Authorization Headers (7) **Body** • Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** ▾

```
1 {
2   ...
3 }
```

Untitled public

Endpoint <https://enohwbcag2dde.x.pipedream.net/> Copy + New

LIVE PAUSE Q Type to search...

Today

12:01:01 AM POST /

Details	POST	/
Headers	▶ (7) headers	copy
Body	RAW PRETTY STRUCTURED	copy
▼ "root": "hello": "world"		

Low Code

Escribiendo poco código



Google Sheets

HTTP SERVER   

File Edit View Insert Format Data Tools Extensions Help

Menus     100%  \$  %

F6  

	A	B	C	
1	Type "@Emoji" to insert an emoji			
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				

 Apps Script

 AppSheet

 GenerateMarkdownTable

 GPT for Sheets™ and Docs™

 Numerous.ai: ChatGPT for Sheets™

 Simple ML for Sheets



Apps Script

HTTP Server



Files

AZ



Run Debug doPost



Execution log



Code.gs



Libraries



Services



```
1  function doPost(e) {
2    const ss = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet()
3    ss.appendRow([e.postData.contents])
4 }
```

getActiveSpreadsheet()

New deployment

Select type	⚙️ Configuration	?
Web app	<p>Description</p> <p>New description</p> <input type="text" value="HTTP SERVER"/>	
Web app	<p>Execute as</p> <p>Me (francescogentyle@gmail.com)</p>	
	<p>The web app will be authorized to run using your account data.</p>	
Who has access	<p>Anyone</p>	
	<p>This can also be used as a library. Learn more</p>	

[Cancel](#) [Deploy](#)

getActiveSpreadsheet()

New deployment

Deployment successfully updated.

Version 1 on Apr 21, 2024, 12:25 AM

Deployment ID

AKfycbzwIJCZN24zmPUt7CzSEIzIG-Ys2Ngxu-0M1o9cirKMQaCzx7p74ocrmzMga_hPu9Y

[Copy](#)

Web app

URL

https://script.google.com/macros/s/AKfycbzwIJCZN24zmPUt7CzSEIzIG-Ys2Ngxu-0M1o9cirKMQaCzx7p74ocrmzMga_hPu9Y

[Copy](#)

[Done](#)

POST

https://script.google.com/macros/s/AKfycbwIJCZN24zmPUt7CzSEIzIG-Ys2Ngxu-0M1o9cirKMQaCZx7p74ocrmzMga_hPu9Y/exec

Params

Authorization

Headers (8)

Body

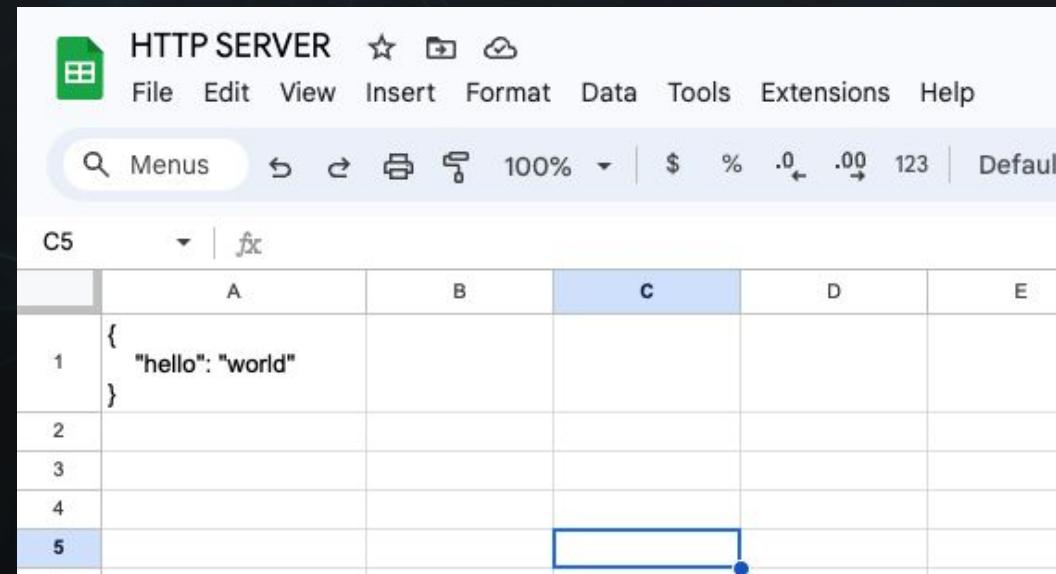
Pre-request Script

Tests

Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```
1 {  
2   ... "hello": "world"  
3 }
```



The screenshot shows a window titled "HTTP SERVER" with a menu bar including File, Edit, View, Insert, Format, Data, Tools, Extensions, and Help. The main area is a spreadsheet with columns A, B, C, D, and E. Row 1 contains the JSON object {"hello": "world"} in cell C1. Row 2 is empty. Row 3 is empty. Row 4 is empty. Row 5 is also empty. The cell C5 is highlighted with a blue border. The status bar at the bottom shows "Default..".

	A	B	C	D	E
1			{ "hello": "world" }		
2					
3					
4					
5					

Muy lindo todo

**¿Pero si necesito algo más allá de una
prueba de concepto o herramienta
interná?**

Serverless

(El servidor de alguien más)

Vercel Products Solutions Resources Docs Pricing

← Back to Templates

 [vercel/examples](#)

Python Hello World

Use Python on Vercel with Serverless Functions using the Python Runtime.

[Deploy](#) [View Demo](#)

Framework	Python
Use Case	Starter
CSS	CSS



Python Hello World

This example shows how to use Python on Vercel with Serverless Functions using the [Python Runtime](#).

Demo

<https://python-hello-world.vercel.app/>

Running Locally

```
npm i -g vercel
vercel dev
```

Create Git Repository

To ensure you can easily update your project after deploying it, a Git repository must be created. Every push to that Git repository will be deployed automatically.

Git Scope

Repository Name

 Eyon42

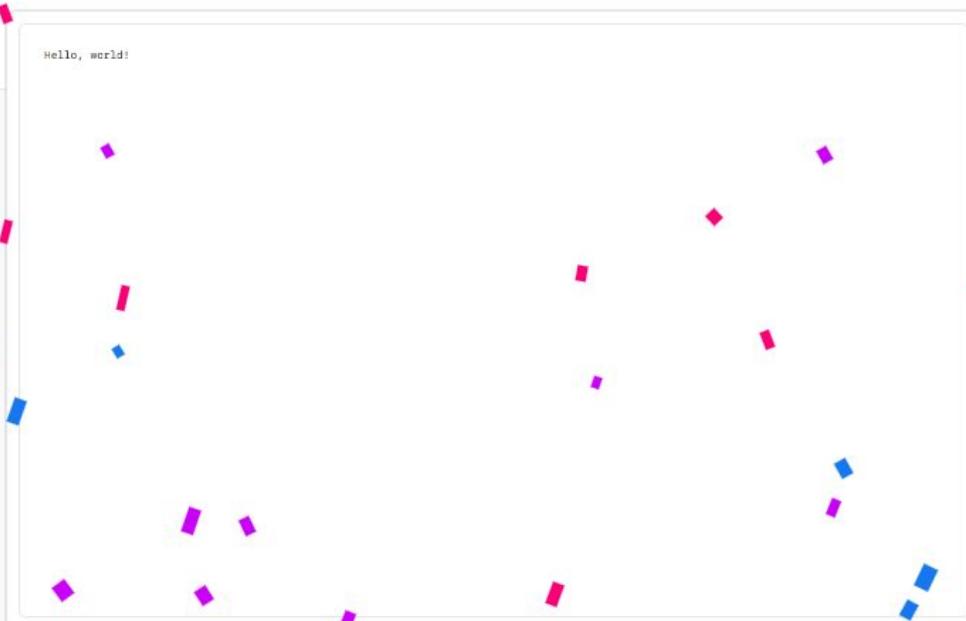
python-hello-world

 Create private Git Repository

Create

Congratulations!

You just deployed a new Project to Vercel.



Continue to Dashboard 

Next Steps

Instant Previews

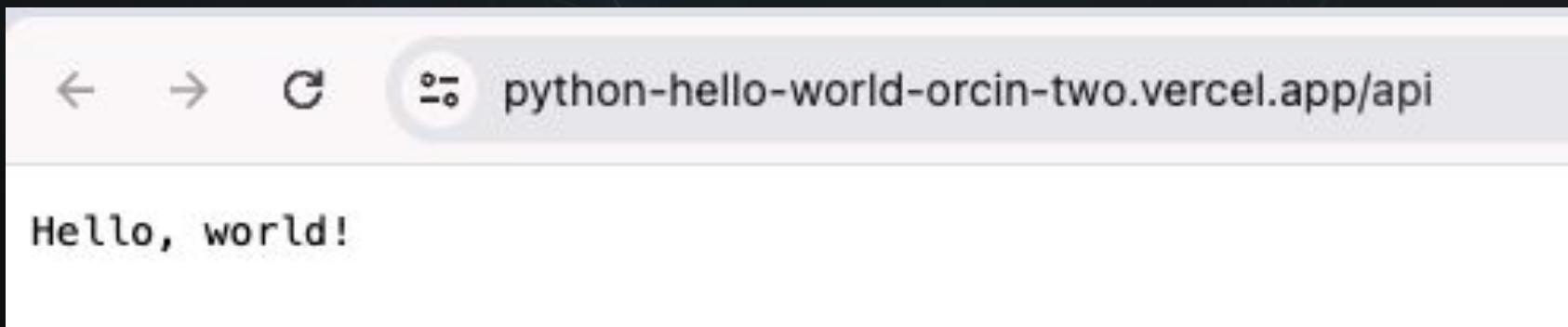
Push a new branch to preview changes instantly

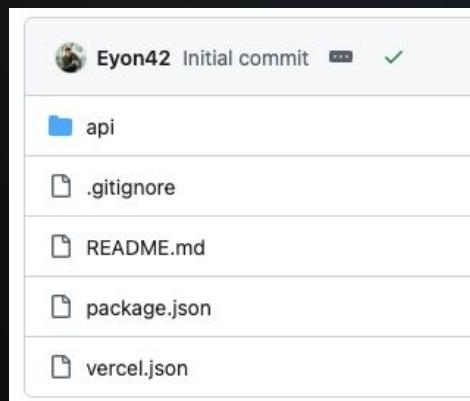
Add Domain

Add a custom domain to your project

Enable Speed Insights

Track how users experience your site over time





python-hello-world / api / index.py

Eyon42 Initial commit

Code Blame 10 lines (8 loc) · 296 Bytes ⚡ Code 55% faster with GitHub Copilot

```
1  from http.server import BaseHTTPRequestHandler
2
3  class handler(BaseHTTPRequestHandler):
4
5      def do_GET(self):
6          self.send_response(200)
7          self.send_header('Content-type','text/plain')
8          self.end_headers()
9          self.wfile.write('Hello, world!'.encode('utf-8'))
10         return
```



**Serverless es muy caro
o no se adapta a mis necesidades**

Levantando nuestro propio servidor



DigitalOcean



Google Cloud Platform



**No hay presupuesto para VM
o no es conveniente**

**Y no quiero tocar la configuración
de mi red**

Cloudflare Tunnel

Protect your web servers from direct attack

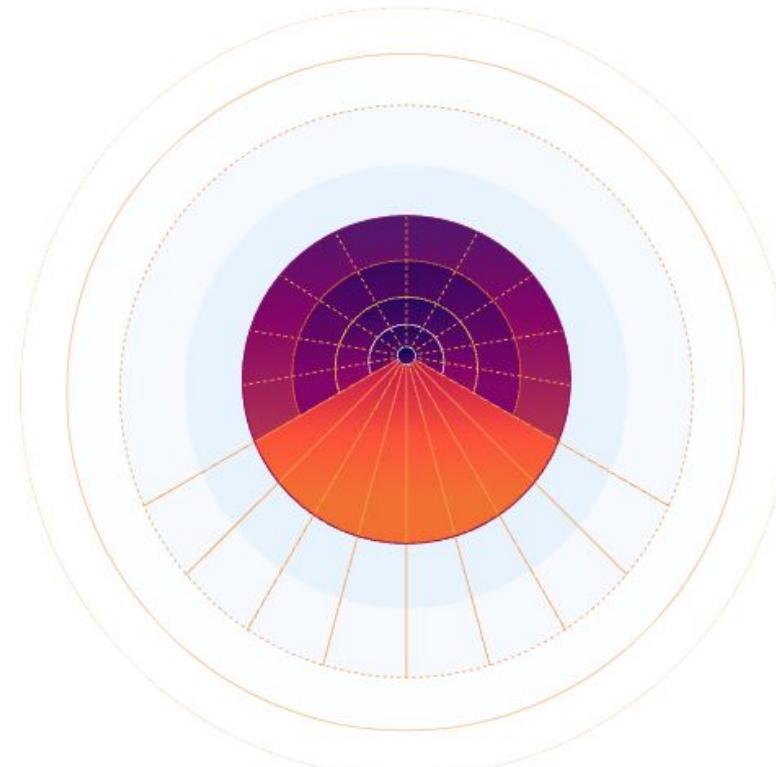
From the moment an application is deployed, developers and IT spend time locking it down — configuring ACLs, rotating IP addresses, and using clunky solutions like GRE tunnels.

There's a simpler and more secure way to protect your applications and web servers from direct attacks: Cloudflare Tunnel.

Ensure your server is safe, no matter where it's running: public cloud, private cloud, Kubernetes cluster, or even a Mac mini under your TV.

[View in Dashboard](#)

Looking for enterprise-grade solutions? [Contact Sales](#)



If you don't have cloudflared installed on your machine:

```
brew install cloudflared &&  
$ sudo cloudflared service install  
eyJhIjo1D...
```



Edit public hostname for Francesco Macbook Air

Public hostname

Subdomain

Domain (Required)

Path

Service

Type (Required)

URL (Required)

For example, <https://localhost:8001>

```
$ cloudflared tunnel --url http://localhost:8080
```





```
~$ sudo apt install nginx
```

```
fg_eyon@instance-20240425-203917:~$ sudo systemctl start nginx.service
fg_eyon@instance-20240425-203917:~$ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Fri 2024-04-26 00:23:32 UTC; 5min ago
    Docs: man:nginx(8)
  Process: 2124 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/S>
  Process: 2125 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 2150 (nginx)
    Tasks: 3 (limit: 1141)
   Memory: 2.2M
     CPU: 33ms
    CGroup: /system.slice/nginx.service
            └─2150 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              ├─2152 "nginx: worker process"
              ├─2153 "nginx: worker process"
              └─2154 "nginx: worker process"

Apr 26 00:23:32 instance-20240425-203917 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Apr 26 00:23:32 instance-20240425-203917 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server...
```

lines 1-17/17 (END)

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Hora de servir archivos

```
$ sudo mkdir /var/www/hello  
$ cd /var/www/hello
```

```
$ echo "Hello world" > index.html
```

```
fg_eyon@instance-20240425-203917:/var/www/hello$ ls -la  
total 12  
drwxrwxrwx 2 root      root      4096 Apr 26 00:58 .  
drwxr-xr-x  4 root      root      4096 Apr 26 00:55 ..  
-rw-r--r--  1 fg_eyon  fg_eyon    12 Apr 26 00:58 index.html  
fg_eyon@instance-20240425-203917:/var/www/hello$
```

```
:~$ cd /etc/nginx/sites-available/  
:/etc/nginx/sites-available$ sudo mv default default.bak  
:/etc/nginx/sites-available$ sudo vim default
```

```
server {  
    listen 80;  
    server_name _;  
    root /var/www/hello;  
}
```

```
$ sudo nginx -s reload
```



Not Secure

34.134.33.234

Hello world

Y una pequeña API

```
~$ vim app.py
```

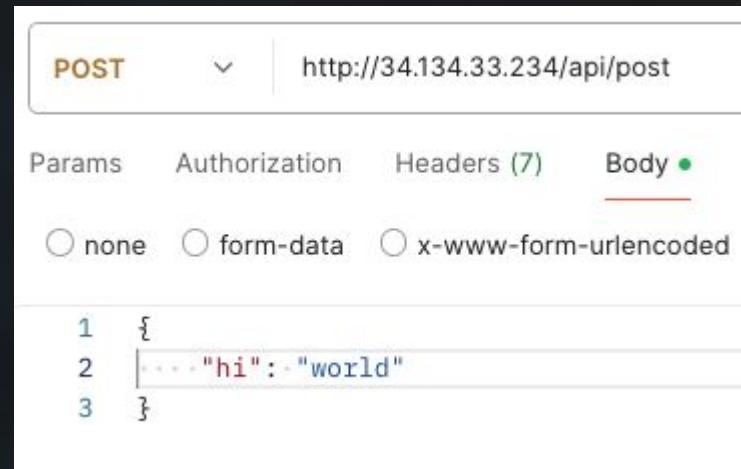
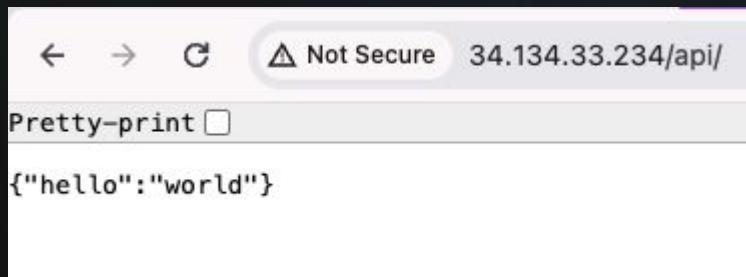
```
import json
from flask import Flask, request

app = Flask(__name__)

@app.route('/')
def hello_world():
    return { "hello": "world"}

@app.route('/post', methods = ['POST'])
def post():
    print(request.json)
    return "ok"
```

```
server {  
    listen 80;  
    server_name _;  
    root /var/www/hello;  
  
    location /api/ {  
        proxy_pass http://127.0.0.1:5000/;  
    }  
}
```



```
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [26/Apr/2024 03:09:38] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2024 03:16:54] "GET / HTTP/1.0" 200 -
{'hi': 'world'}
127.0.0.1 - - [26/Apr/2024 03:18:24] "POST /post HTTP/1.0" 200 -
```

Tengo un caso de uso MUY específico

(0 ganas de reinventar la rueda)

Implementación directa del protocolo HTTP

Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding
UC Irvine
J. Gettys
Compaq/W3C
J. C. Mogul
Compaq
H. Frystyk
W3C/MIT
L. Masinter
Xerox
P. Leach
Microsoft
T. Berners-Lee
W3C/MIT
June, 1999

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers [47]. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification defines the protocol referred to as "HTTP/1.1", and is an update to RFC 2068 [33].



The Original HTTP as defined in 1991

Click to go forward, hold to see history

This document defines the Hypertext Transfer protocol (HTTP) as originally implemented by the [World Wide Web](#) initiative software in the prototype released. This is a subset of the [full](#) HTTP protocol, and is known as HTTP 0.9.

No client profile information is transferred with the query. Future HTTP protocols will be back-compatible with this protocol.

This restricted protocol is very simple and may always be used when you do not need the capabilities of the full protocol which is backwards compatible.

The definition of this protocol is in the public domain (see [policy](#)).

The protocol uses the normal internet-style telnet protocol style on a TCP-IP link. The following describes how a client acquires a (hypertext) document from an HTTP server, given an HTTP document [address](#).

Connection

The client makes a TCP-IP connection to the host using the [domain name](#) or [IP number](#), and the [port number](#) given in the address.

If the port number is not specified, 80 is always assumed for HTTP.

The server accepts the connection.

Note: HTTP currently runs over TCP, but could run over any connection-oriented service. The interpretation of the protocol below in the case of a sequenced packet service (such as DECnet(TM) or ISO TP4) is that that the request should be one TPDU, but the response may be many.

Request

The client sends a document request consisting of a line of ASCII characters terminated by a CR LF (carriage return, line feed) pair. A well-behaved server will not require the carriage return character.

This request consists of the word "GET", a space, the [document address](#), omitting the "http:, host and port parts when they are the coordinates just used to make the connection. (If a gateway is being used, then a full document address may be given specifying a different naming scheme).

The document address will consist of a single word (ie no spaces). If any further words are found on the request line, they MUST either be ignored, or else treated according to the [full HTTP spec](#).

The search functionality of the protocol lies in the ability of the addressing syntax to describe a [search on a named index](#).

A search should only be requested by a client when the index document itself has been described as an index using the [ISINDEX tag](#).

Response

The response to a simple GET request is a message in hypertext mark-up language ([HTML](#)). This is a byte stream of ASCII characters.

Lines shall be delimited by an optional carriage return followed by a mandatory line feed character. The client should not assume that the carriage return will be present. Lines may be of any length. Well-behaved servers should restrict line length to 80 characters excluding the CR LF pair.

The format of the message is HTML - that is, a trimmed SGML document. Note that this format allows for menus and hit lists to be returned as hypertext. It also allows for plain ASCII text to be returned following the PLAINTEXT tag.

The message is terminated by the closing of the connection by the server.

Well-behaved clients will read the entire document as fast as possible. The client shall not wait for user action (output paging for example) before reading the whole of the document. The server may impose a timeout of the order of 15 seconds on inactivity.

Error responses are supplied in human readable text in HTML syntax. There is no way to distinguish an error response from a satisfactory response except for the content of the text.

Disconnection

The TCP-IP connection is broken by the server when the whole document has been transferred.

The client may abort the transfer by breaking the connection before this, in which case the server shall not record any error condition.

Requests are [idempotent](#). The server need not store any information about the request after disconnection.

Tim BL

<https://www.w3.org/Protocols/HTTP/AsImplemented>



Enlace a la interfaz TCP

```
use std::net::{TcpListener};  
let listener = TcpListener::bind("127.0.0.1:5001").unwrap();
```

Escuchar Streams

```
use std::net::{TcpListener};  
let listener = TcpListener::bind("127.0.0.1:5001").unwrap();  
  
for stream in listener.incoming() {  
    let mut stream = s.unwrap()  
    let mut buffer_reader = BufReader::new(&mut stream);  
}
```

Leyendo la primera línea

```
let mut first_line = String::new();
buffer_reader.read_line(&mut first_line).unwrap();
```

GET /index.html HTTP1.1

Procesando la petición

```
let mut info = first_line.split_whitespace();  
  
// extraer método  
let method = match info.next() {  
    Some("GET") => "GET",  
    Some(m) => panic!("Method {} not supported", m),  
    _ => panic!("shit"),  
};  
  
// extraer recurso  
let resource = match info.next() {  
    Some(r) => r,  
    _ => "/",  
};
```

Respondiendo

```
match resource {  
    "/" => {  
        stream.write_all(b"HTTP/1.1 200 OK\r\n\r\n").unwrap();  
        stream.write_all(b"Hello world").unwrap()  
    },  
    _ => {  
        stream.write_all(b"HTTP/1.1 404 Not Found\r\n\r\n").unwrap();  
        stream.write_all(b"404 - Not Found").unwrap()  
    }  
}
```

Todo en 30 líneas

```
use std::io::{prelude::*, BufReader};
use std::net::{TcpListener, TcpStream};
use std::path::PathBuf;
use std::{thread, time};

fn main() {
    let listener = TcpListener::bind("127.0.0.1:5001").unwrap();

    for s in listener.incoming() {
        let mut stream = s.unwrap();
        let mut buffer_reader = BufReader::new(&mut stream);
        let mut first_line = String::new();

        buffer_reader.read_line(&mut first_line).unwrap();
        let mut info = first_line.split_whitespace();
        // extraer método
        let method = match info.next() {
            Some("GET") => "GET",
            Some(m) => panic!("Method {} not supported", m),
            _ => panic!("shit"),
        };
        // extraer recurso
        let resource = match info.next() {
            Some(r) => r,
            _ => "/",
        };
        stream.write_all(b"HTTP/1.1 200 OK\r\n\r\n").unwrap();
        stream.write_all(b"Hello /var/birras!").unwrap();
    }
}
```



localhost:5001

Hello /var/birras !

Extra: HTTP implementado en un Videojuego

TURING COMPLETE



play campaign
sandbox
schematic hub
profile
options
credits
exit

0.1059 Beta



Contacto

✉️ francesco@eyon.dev

🔗 eyon.dev

🐦 [eyondev](https://eyondev.com)

