

Yuan 的 HTTP Echo Server

專案介紹 與 開發心得

- Demo: <https://echo.yuaner.tw/> ...
- 專案原始碼 <https://github.com/chyuaner/cloudflare-echo-server>
- 本投影片的原始碼 <https://github.com/chyuaner/slide-cloudflare-echo-server>

Yuan Chiu chyuaner@gmail.com

主要核心功能

本專案是提供CDN Edge層級的http
回音伺服器

你對我發出Request，我的伺服器就
會把我從你這邊接到的資訊：

- 以什麼網址字串
- 帶了什麼Post Body
- 帶了什麼Request Header

會如實的以ResponseBody方式回應
給你。

同時也可以當作MyIP查詢使用，會顯
示在「Host」區塊。

Yuan 的 HTTP Echo Server

GEThttps://echo.yuaner.tw/assets/text/dict/textpar.html?ref=tablericons&utm_source=tablericons-mainsite&utm_medium=partner&utm_campaign=250k-more-icons&tab=all

MethodGET

BaseUrlhttps://echo.yuaner.tw

OriginalUrl/assets/text/dict/textpar.html?ref=tablericons&utm_source=tablericons-mainsite&utm_medium=partner&utm_campaign=250k-more-icons&tab=all

Protocolhttps

URL Params

1. assets

2. text

3. dict

4. textpar.html

URL Query

• ref: tablericons

• utm_source: tablericons-mainsite

• utm_medium: partner

• utm_campaign: 250k-more-icons

• tab: all

Host

• IP: 2001:b400:e2c2:c865:d68b:4031:6d82:80f3

• hostname: echo.yuaner.tw

• Colo: SJC

• Country: TW

• City: Kaohsiung

• Continent: AS

• Latitude: 22.61626

• Longitude: 120.31333

• Region: Takao

• Region Code: KHH

• Timezone: Asia/Taipei

Header

Key	Value
accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-encoding	gzip, br
accept-language	zh-TW,zh;q=0.8,en-US;q=0.5,en;q=0.3
cf-connecting-ip	2001:b400:e2c2:c865:d68b:4031:6d82:80f3
cf-ipcountry	TW
cf-ray	9a6201217b0cf94b
cf-visitor	{"scheme":"https"}
connection	Keep-Alive

專案特色

在CDN Edge層級直接提供完整服務

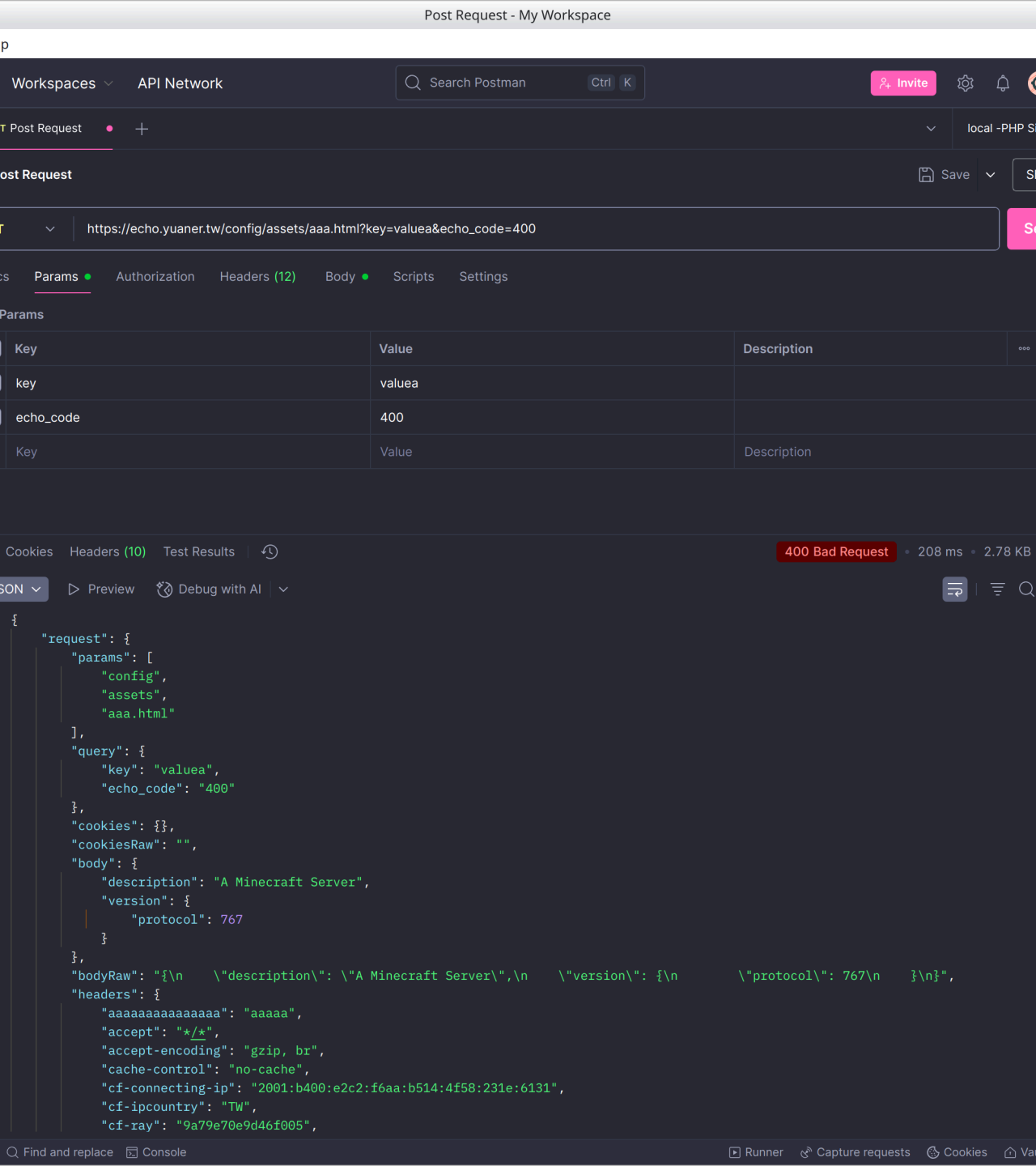
傳統後端架構（以我習慣的PHP為例）

公網 → CDN快取 → Reverse Proxy(管理vhost) → Nginx → php-fpm → 我的專案

 本專案採用的是：CDN層級**直接服務**架構

公網 → CDN直接服務

理論上極致效能低延遲



JSON格式作為預設 Response Body輸出

基本上以 [ealen/echo-server](#) 專案為基礎，重新撰寫復刻核心功能

分為三大塊：

- request
- http
- host

可用於Postman、Paw、Insomnia、Hoppscotch等HTTP API調試客戶端使用

有設計精美的 網頁UI界面

Yuan 的 HTTP Echo Server

GET

https://echo.yuaner.tw/assets/text/dict/textpar.html?ref=tablericons&utm_source=tablericons-mainsite&utm_medium=partner&utm_campaign=250k-more-icons&tab=all

Method

GET

BaseUrl

https://echo.yuaner.tw

OriginalUrl

/assets/text/dict/textpar.html?ref=tablericons&utm_source=tablericons-mainsite&utm_medium=partner&utm_campaign=250k-more-icons&tab=all

Protocol

https

URL Params

1. assets
2. text
3. dict
4. textpar.html

URL Query

- **ref:** tablericons
- **utm_source:** tablericons-mainsite
- **utm_medium:** partner
- **utm_campaign:** 250k-more-icons
- **tab:** all

Host

- IP: 2001:b400:e2c2:c865:d68b:4031:6d82:80f3
- hostname: echo.yuaner.tw
- Colo: SJC
- Country: TW
- City: Kaohsiung
- Continent: AS
- Latitude: 22.61626
- Longitude: 120.31333
- Region: Takao
- Region Code: KHH
- Timezone: Asia/Taipei

Header

Key	Value
accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-encoding	gzip, br
accept-language	zh-TW,zh;q=0.8,en-US;q=0.5,en;q=0.3
cf-connecting-ip	2001:b400:e2c2:c865:d68b:4031:6d82:80f3
cf-ipcountry	TW
cf-ray	9a6201217b0cf94b
cf-visitor	{\"scheme\":\"https\"}
connection	Keep-Alive

清楚明瞭的HTTP Method

網址列顏色會根據 GET, POST, PUT, DELETE改變顏色

GET `https://echo.yuaner.tw/config/assets/aaa.html?key=valuea`

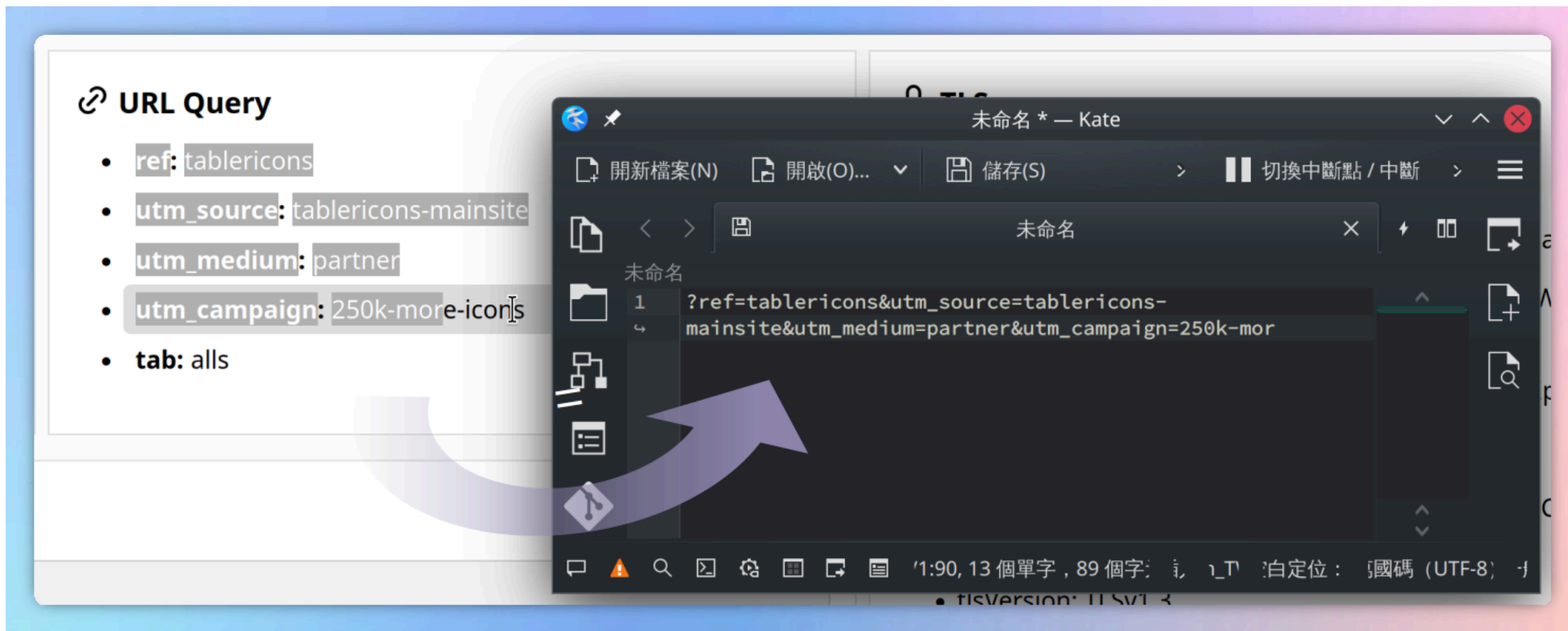
POST `https://echo.yuaner.tw/config/assets/aaa.html?key=valuea`

PUT `https://echo.yuaner.tw/config/assets/aaa.html?key=valuea`

DELETE `https://echo.yuaner.tw/config/assets/aaa.html?key=valuea`

顏色挑選有特別與Swagger、API Doc對齊

網址文字友善複製



有特別為「URL Params」、「URL Query」區塊特別設計友善文字複製。界面乍看下是ul li項目清單，但圈選文字後，會直接複製成可直接貼上網址列的字串

RWD手機友善 與 Dark Mode配色

RWD手機友善

Header

accept

text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

accept-encoding

gzip, br

accept-language

zh-TW,zh;q=0.8,en-US;q=0.5,en;q=0.3

cf-connecting-ip

2001:b400:e2c2:f6aa:b514:4f58:231e:6131

cf-ipcountry

TW

cf-ray

9a7a6e2dcfb5dc0d

Dark Mode配色

Yuan 的 HTTP Echo Server

GET https://echo.yuaner.tw/assets/text/dict/textpar.html?ref=tablericons&utm_source=tablericons-mainsite&utm_medium=partner&utm_campaign=250k-more-icons&tab=all

BaseUrlhttps://echo.yuaner.tw

OriginalUrl/assets/text/dict/textpar.html?ref=tablericons&utm_source=tablericons-mainsite&utm_medium=partner&utm_campaign=250k-more-icons&tab=all

ProtocolHTTP/2

URL Params

1. assets

2. text

3. dict

4. textpar.html

URL Query

ref: tablericons

utm_source: tablericons-mainsite

utm_medium: partner

utm_campaign: 250k-more-icons

tab: all

TLS

tlsCipher: AEAD-AES128-GCM-SHA256

tlsClientCiphersSha1: LnbVHP1R8ay/dUsahfir53h8aw4=

tlsClientExtensionsSha1: l41A6hAf8CFQan+yuADUWHCIneg=

tlsClientExtensionsSha1Le: //MhJMRn06+Eij53CchZp/zi/yQ=

tlsClientHelloLength: 1826

tlsClientRandom: MSQocOJYZ/ck6QgAZxhX9BKL+nCflpkBDs0uF9/Qm4w=

tlsVersion: TLSv1.3

Host

IP: 2001:b400:e2c2:f6aa:b514:4f58:231e:6131

hostname: echo.yuaner.tw

Header

Key	Value
accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-encoding	gzip, br
accept-language	zh-TW,zh;q=0.8,en-US;q=0.5,en;q=0.3
cf-connecting-ip	2001:b400:e2c2:f6aa:b514:4f58:231e:6131
cf-ipcountry	TW
cf-ray	9a7a6e2dcfb5dc0d
cf-visitor	{\"scheme\":\"https\"}

網頁界面仍有提供Raw Response Body

模擬本後端的JSON模式原始輸出，方便切換客戶端比對使用

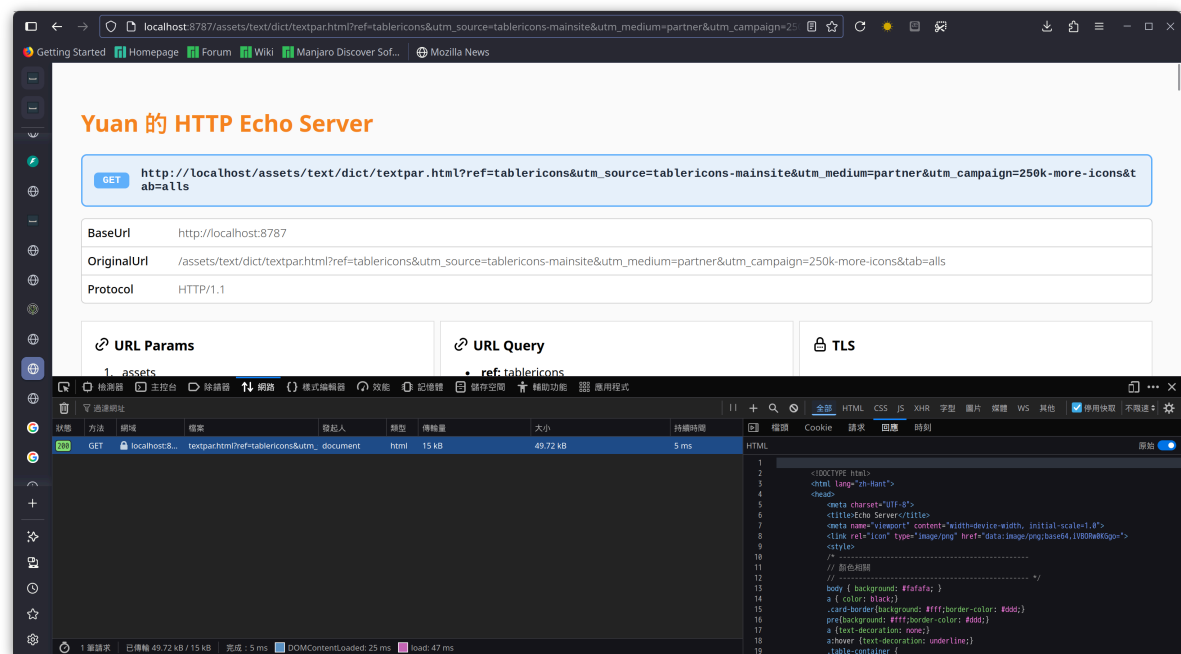
upgrade-insecure-request	1
user-agent	Mozilla/5.0 (X11; Linux x86_64; rv:139.0) Gecko/20100101 Firefox/139.0
x-forwarded-proto	https
x-real-ip	2001:b400:e2c2:f6aa:b514:4f58:231e:6131

- As Organization: HiNet Taiwan
- Region: Taiwan
- Region Code: 04
- Timezone: Asia/Taipei

</> Raw Response Body

```
{
  "request": {
    "params": [
      "assets",
      "text",
      "dict",
      "textpar.html"
    ],
    "query": {
      "ref": "tablericons",
      "utm_source": "tablericons-mainsite",
      "utm_medium": "partner",
      "utm_campaign": "250k-more-icons",
      "tab": "alls"
    },
    "cookies": {},
    "cookiesRaw": "",
    "body": {},
    "bodyRaw": "",
    "headers": {
      "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
      "accept-encoding": "gzip, br",
      "accept-language": "zh-TW,zh;q=0.8,en-US;q=0.5,en;q=0.3",
      "cf-connecting-ip": "2001:b400:e2c2:f6aa:b514:4f58:231e:6131",
      "cf-ipcountry": "TW",
      "cf-ray": "9a7a6e2dcfb5dc0d",
      "cf-visitor": "{\\\"scheme\\\":\\\"https\\\"}",
      "connection": "Keep-Alive",
      "host": "echo.yuaner.tw",
      "user-agent": "Mozilla/5.0 (X11; Linux x86_64; rv:139.0) Gecko/20100101 Firefox/139.0"
    }
  }
}
```

效能考量設計



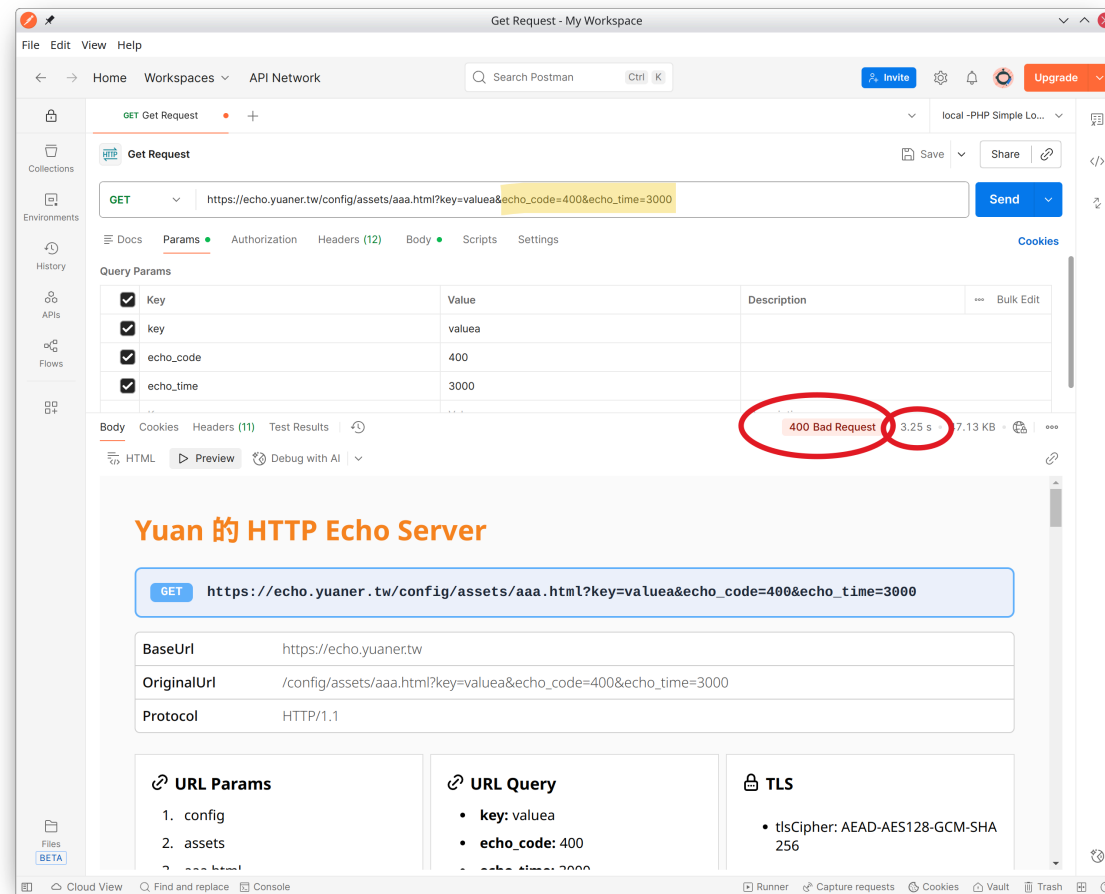
網頁版不會產生額外Request載入其他資源！（如:圖片、CSS、JS等等）

- 所有外部資源如Icons與Syntax Highlight JS都已直接內嵌在單一這個Request。
- 所有圖片都是SVG格式

➡ 附加功能

提供 URL Query 參數 來控制以下功能

- `echo_code=200` : 控制要回傳的 HTTP Status Code
- `echo_time=3000` : 控制伺服器要延遲多久才會送 Response , 模擬較差網路品質狀況使用



➡ 提供本地架設方式

本專案雖然是為Cloudflare量身打造的，但也有提供傳統的架設方式

- Docker

```
docker run -p 3000:3000
```

```
chyuaner/echo-server
```

- Node直接啟動

```
npm i && npm run start
```

The screenshot shows the Docker Hub interface for the repository **chyuaner/echo-server**. The page includes a header with the Docker Hub logo, navigation links (Explore, My Hub), a search bar containing "chyuaner/echo-server", and a "CtrlK" button. Below the header, the repository name is displayed with a "Manage Repository" button. The repository is described as "By chyuaner · Updated about 3 hours ago" and "提供輕量化的http回音伺服器 (Docker獨立執行版本)". It has 0 stars and 91 downloads. The "Overview" tab is selected, showing a description of the project as a "Yuan 的 HTTP Echo Server" and a link to the GitHub repository. A demo URL is provided: https://echo.yuaner.tw/assets/text/dict/textpar.html?ref=tablericons&utm_source=tablericons-mainsite&utm_medium=partner&utm_campaign=250k-more-icons&tab=all. A screenshot of the demo is shown, displaying a table with request details like Method (GET), BaseURL, OriginalURL, and Protocol. The "URL Query" section lists parameters like ref, utm_source, utm_medium, and utm_campaign. The "Host" section shows IP, hostname, color, country, city, continent, latitude, longitude, region, and timezone. The "Tag summary" section shows the "master" tag. The "Content type" is "Image" and the "Digest" is "sha256:1e5421ffd...". The "Size" is "45.7 MB" and the "Last updated" time is "about 3 hours ago". The "docker pull chyuaner/echo-server:master" command is shown at the bottom.

開發心得

本次採用語言

其實我本身比較擅長PHP，但...

- 本次目的就是想減少自有主機依賴，變成要遷就雲端提供商的限制
- Cloudflare Workers基本上就是以NodeJS語言為主
- 本次目的是盡可能減少線路「繞圈」與「轉換」，來換取效能

開發緣由

先前都是使用 ealen/echo-server

雖然已包好Docker，架設簡單、輸出結構簡潔漂亮，但是...

- 只有提供JSON格式回傳
- 需要有自己的主機架設
 - 然後我近期的主機資源很吃緊~~~
- 其他原因，下述...

近期對CDN優化有興趣

我以前使用Cloudflare Worker用途

- 網址搬遷時用白名單Mapping表概念，讓特定舊網址跳轉到新網址
- 短網址直接跳轉
- 偵測客戶端特定國家自動跳轉專用網址

後來發現Workers其實可以做到以下事情

- 拿到客戶端傳來的Request
 - **URL Params / Query**
 - **Request Post Body**
 - **Request Post Body**
- 可以控制Response內容
 - 跳轉導向到特定的網址
 - 改寫 Response Body內容
 - 可以直接輸出指定的字串，也包含整串HTML大字串

 評估下來，本次需求僅用以上功能就足夠了！！！！

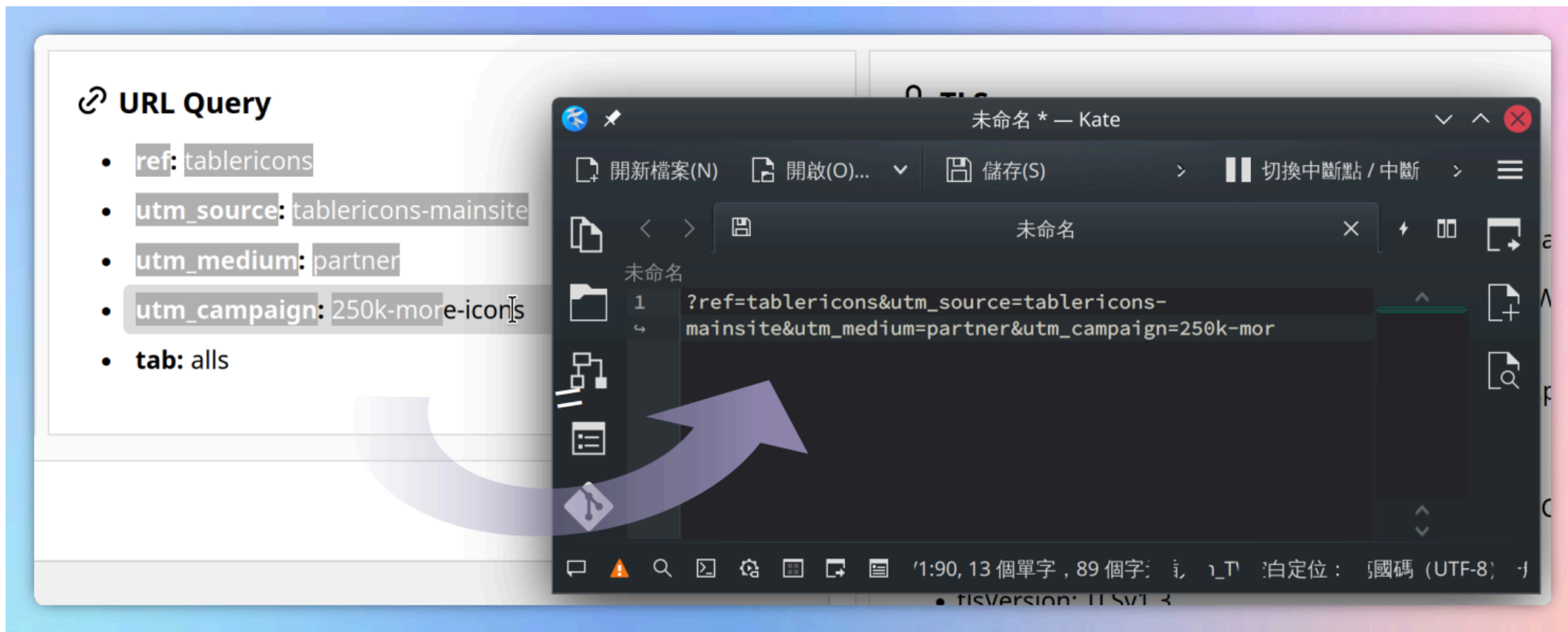
Cloudflare Worker執行能力有限

- 並不是完整的NodeJS環境，只能執行Cloudflare API名單內的function
- 沒有檔案概念，若需要額外資源檔案，需要另外加掛資源空間
 - 也直接導致很難使用ejs之類的模板框架（遇到底層有使用到 `eval` 與 `fs` 就會被拒絕執行）

對策：

- 所有資源全部Hard Code寫死內嵌，整理成JS陣列。
- 不使用模板系統，直接靠func整理HTML字串。
 - 但自己在規劃JS Function時仍然會盡量模組化。

網址文字友善複製



界面乍看下是url項目清單，但圈選文字後，會直接複製成可直接貼上網址列的字串

網址文字友善複製 實做方式

原始結構全部都是 ``

```
<span class="firstchar">?</span>
<span class="li">
  <span class="part">
    <span class="key">ref</span><span class="kvsplit">=</span>tablericons
  </span>
  <span class="split">&amp;</span>
</span>
<span class="li">
  <span class="part">
    <span class="key">utm_source</span><span class="kvsplit">=</span>tablericons-mainsite
  </span>
  <span class="split">&amp;</span>
</span>
<span class="li">
  <span class="part">
    <span class="key">utm_medium</span><span class="kvsplit">=</span>partner
  </span>
  <span class="split">&amp;</span>
</span>
<span class="li">
  <span class="part">
    <span class="key">utm_campaign</span><span class="kvsplit">=</span>250k-more-icons
  </span>
  <span class="split">&amp;</span>
</span>
<span class="li">
  <span class="part">
    <span class="key">tab</span><span class="kvsplit">=</span>alls
  </span>
</span>
```

全由CSS模擬 `` 的效果

0% JavaScript處理

```
.urltext {margin-left:1em;margin-bottom:-1em;}
.urltext .li {
  display: inline-block;
  vertical-align: middle;
  padding-left: 0.5em;
  border-radius: 5px;
}
.urltext .firstchar { padding-left:1em; }
.urltext .firstchar, .urltext .split, .urltext .kvsplit{
  opacity: 0;
}
.urltext .li {
  display: list-item;
  margin-left: 1em;
  position: relative;
  top: -2em;
  line-height: 2em;
}
.urltext .key{ font-weight: bold; }
.urltext .key::after {
  content: ": ";
  position: absolute;
}
```

提供本地架設方式

開發過程90%都是使用Cloudflare提供的wrangler模擬環境測試

但考量隨時會有獨立架設需求，仍有特別製作 `/src/server.js` 供獨立執行使用

`/src/index.js` 核心主功能流程與Cloudflare Workers直接進入點

`/src/server.js` 獨立執行專用 進入點

- 以NodeJS最原生的node:http為主
 - 本次需求沒有複雜的 router、session...等等，也沒有多頁面需求，所以不使用express這類框架減少無謂的依賴與效能損耗
- 會銜接回 `/src/index.js`，讓所有功能都與Cloudflare Workers模式對齊
 - 但... Cloudflare專屬功能提供的GeoIP資料則無法使用，需要用替代方式處理GeoIP問題。因為我暫時還沒有常駐用的獨立架設需求，就暫時不處理了。

搭配 Github Actions 自動更新到 Docker Hub

ner / cloudflare-echo-server

Q Type / to search



Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

New workflow

er Image

etrics

All workflows

Showing runs from all workflows

Q Filter workflow runs

12 workflow runs

Event

Status

Branch



將json輸出改成原文輸出不排版

Build & Push Docker Image #10: Commit [857a465](#) pushed by [chyuaner](#)

master

Tod: 51s



Fix Dockerfile & add package-lock

Build & Push Docker Image #6: Commit [e7921d0](#) pushed by [chyuaner](#)

master

Tod: 48s



Fix Dockerfile

Build & Push Docker Image #5: Commit [ceb1968](#) pushed by [chyuaner](#)

master

Tod: 24s



Fix Dockerfile

Build & Push Docker Image #4: Commit [9fd19c2](#) pushed by [chyuaner](#)

master

Tod: 28s



fix github actions: 先移除以tag觸發

Build & Push Docker Image #3: Commit [87483c3](#) pushed by [chyuaner](#)

master

Tod: 27s



github actions: 先移除以tag觸發

Build & Push Docker Image #2: Commit [ef04b36](#) pushed by [chyuaner](#)

master

Tod: 26s



add github action: docker 以 branch mast...

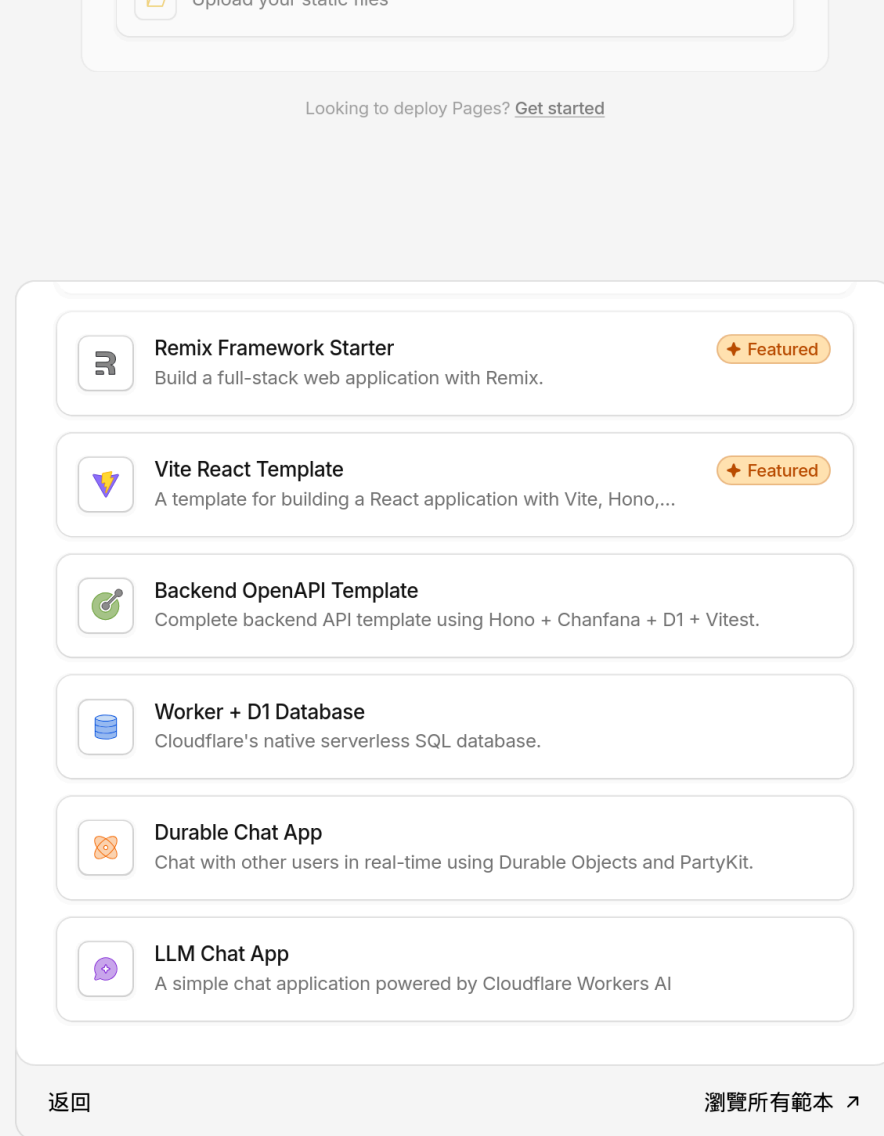
Build & Push Docker Image #1: Commit [1c9e5a0](#) pushed by [chyuaner](#)

master

Tod: 16s

後續新專案要不要用 Cloudflare Worker ?

- 他有提供介接DB的機能，但...
- Cloudflare Workers執行能力有限！
 - 本次專案已經用了大量的Wordaround方法，
尤其是有不少直接Hard Code寫死
- 如果是傳統電商需求，基本上不會考慮！
- Fake Images Please Workers形式復刻中 🤪



就醬，歡迎交流😊