

Calculadora de moneda

IES CELIA VIÑAS - DWEC

Resumen

Vamos a realizar un intercambiador de moneda “real” haciendo uso de la API <https://www.exchangerate-api.com/> .

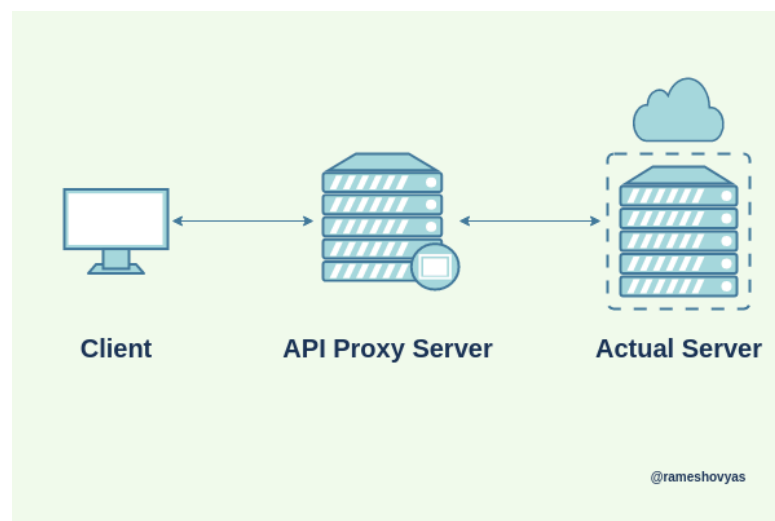
En el siguiente video puedes encontrar algunos aspectos del funcionamiento esperado:

📺 Tarea 2 - Calculadora de Moneda.mp4

Arquitectura del proyecto

Servidor Proxy

Vamos a construir un servidor proxy el cual sea capaz de recibir peticiones del cliente (página web) y realizar la petición a la API. Una vez recibida la información solicitada desde la API, se la enviaremos al cliente. Todo esto se realiza principalmente para saltarnos las restricciones CORS.



Página Cliente

Puede descargar el código fuente de la página web desde el siguiente enlace.

https://drive.google.com/file/d/13F-Wd39UI0z-uAXzh2uYyKkz4I3RE2BM/view?usp=drive_link

REQUISITOS

1. Servidor Proxy

- a. Debe de estar implementado en **nodejs** haciendo uso de las librerías:
 - i. **Express:** es para crear nuestro servidor proxy
 - ii. **Needle:** es un cliente HTTP ligero que permite a nuestro servidor realizar solicitudes HTTP en nuestro nombre al servidor real. También podríamos haber utilizado node-fetch, pero Needle es súper simple y ligero para esta tarea.
 - iii. **Dotenv:** nos permite crear archivos de variables de entorno donde almacenaremos nuestra clave de API y otros parámetros.
- b. Debe de implementar CORS.
- c. Debe de ejecutar llamadas a servidor teniendo en cuenta la asincronía de estas llamadas.
- d. Las variables de clase como URL, API_KEY, etc. deben de estar configuradas en un archivo **.env**.

2. Página Cliente:

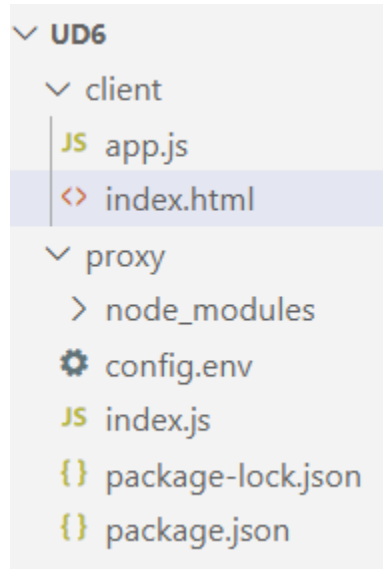
Debe de realizar las llamadas al servidor desde el código haciendo uso de **Fetch** y tener en cuenta la **asincronía** de las mismas.

IMPORTANTE: Existe una restricción nueva en Google Chrome (solo lo he probado en este navegador denominada **CSP** (Content Security Policy directive)). Podemos solventarlo añadiendo la siguiente línea en nuestro html:

```
<meta http-equiv="Content-Security-Policy" content="default-src 'self';  
script-src 'self' 'unsafe-inline'; connect-src 'self'  
http://127.0.0.1:3000;">
```

PASOS RECOMENDADOS

1. **Crear un nuevo proyecto NodeJS:** La estructura junto al **código cliente** debe ser parecida a la siguiente:



2. Debe de instalar las librerías: **express dotenv needle cors**. Puede usar **npm**, el gestor de paquetes de NodeJS.
3. Empiece a crear un EndPoint desde el cliente a nuestro servidor. Realice pruebas.
4. Puede continuar haciendo llamadas a la API para obtener datos sobre monedas.
5. Enlace el punto 3 y 4 para que la aplicación pueda funcionar.

6. Implemente el código necesario para que el cliente funcione apropiadamente.

Anexo:

En nodeJS existe la librería **nodemon**. Esta librería nos permite el autocompilado del código cada vez que hagamos un cambio. En este tutorial puede ver claramente como configurarlo:

https://fullstackopen.com/es/part3/node_js_y_express#nodemon

ENLACES

Express - https://fullstackopen.com/es/part3/node_js_y_express

CORS -

<https://dev.to/speaklouder/how-to-configure-cors-in-nodejs-with-express-11h>

DotEnv - <https://keepcoding.io/blog/sabes-que-es-el-dotenv-en-node-js/>

Needle - <https://www.npmjs.com/package/needle>

Tutorial en inglés -

<https://medium.com/@rameshovyas/step-by-step-guide-to-create-an-api-proxy-server-b1ec19bc36e1>